Readings in Medical Artificial Intelligence

The Addison-Wesley Series in Artificial Intelligence

Buchanan and Shortliffe (eds.): *Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project* (1984)

Clancey and Shortliffe (eds.): *Readings in Medical Artificial Intelligence: The First Decade* (1984)

Pearl: *Heuristics: Intelligent Search Strategies for Computer Problem Solving* (1984)

Sager: *Natural Language Information Processing: A Computer Grammar of English and Its Applications* (1981)

Wilensky: *Planning and Understanding: A Computational Approach to Human Reasoning* (1983)

Winograd: *Language as a Cognitive Process Vol. I: Syntax* (1983)

Winston: *Artificial Intelligence,* Second Edition (1984)

Winston and Horn: *LISP* (1981)

Readings in Medical Artificial Intelligence

The First Decade

Edited by

William J. Clancey
Department of Computer Science
Stanford University

Edward H. Shortliffe
Department of Medicine
Stanford University School of Medicine

Addison-Wesley Publishing Company
Reading, Massachusetts • Menlo Park, California
London • Amsterdam • Don Mills, Ontario • Sydney

To our friend Tim Beckett, an extraordinary physician and teacher, whose humanism and clinical skills inspired and moved his patients as well as his colleagues. His style and insights influenced us greatly, both in our research and in contemplating the nature of clinical reasoning as we prepared this book.

This book is in The Addison-Wesley Series in Artificial Intelligence.

Library of Congress Cataloging in Publication Data

Main entry under title:

Readings in medical artificial intelligence.

Bibliography: p.
Includes indexes.
1. Medicine—Data processing—Congresses.
2. Artificial intelligence—Congresses. I. Clancey, William J. II. Shortliffe, Edward Hance.
R858.A2R4 1984 610'.28'54 83-15560
ISBN 0-201-10854-2

ABCDEFGHIJ-MA-8987654

Contents

Contributors

JANICE S. AIKINS

Dr. Aikins received her Ph.D. in computer science from Stanford University in 1980. She is currently a research computer scientist at IBM's Palo Alto Scientific Center. She specializes in designing systems with an emphasis on the explicit representation of control knowledge in expert systems.

ROBERT L. BLUM

Dr. Blum received his M.D. from the University of California Medical School at San Francisco in 1973. From 1973 to 1976 he did an internship and residency in the Department of Internal Medicine at the Kaiser Foundation Hospital in Oakland, California, where he was chief resident in 1976. He received his Ph.D. in computer science and biostatistics at Stanford University in 1982. Currently a research associate in the Heuristic Programming Project at Stanford, Dr. Blum is principal investigator of the RX project. The goal of the RX project is the automated discovery and confirmation of medical knowledge from large time-oriented data bases.

BRUCE G. BUCHANAN

Professor Buchanan was a mathematics major at Ohio Wesleyan University and received his Ph.D. in philosophy from Michigan State University. He joined the Computer Science Department at Stanford University in 1966 and is now a professor of computer science research. He is co–principal investigator of the Heuristic Programming Project. His main research interest is artificial intelligence, specifically designing expert systems for scientists and physicians. He has helped develop several well-known programs, including DENDRAL, Meta-DENDRAL, MYCIN, and EMYCIN.

B. CHANDRASEKARAN

Professor Chandrasekaran received his Ph.D. from the Moore School of Electrical Engineering of the University of Pennsylvania in 1967. He spent the next two years as a research scientist at the Philco-Ford Corporation in Blue Bell, Pennsylvania, working on problems in the design of pattern-

recognition machines. He joined the faculty of Ohio State University in 1969, and is currently a professor of computer and information science. His current research interests span several areas in artificial intelligence, and include knowledge-based problem solving and vision.

WILLIAM J. CLANCEY

Dr. Clancey received his Ph.D. in computer science from Stanford University in 1979. He is currently a research associate in the Heuristic Programming Project at Stanford University. He specializes in computer-aided instruction in order to investigate, through student modeling and explanation in teaching, general principles of human learning and knowledge representation.

RANDALL DAVIS

Professor Davis received his Ph.D. in computer science from Stanford University in 1976. He spent two additional years at Stanford as a Chaim Weizmann Postdoctoral Scholar. In 1978 he joined the faculty at M.I.T. and held an Esther and Harold Edgerton Endowed Chair from 1979 to 1981. His current research focuses on systems that work from descriptions of structure and function and hence are capable of reasoning from "first principles" to support a wider range of robust problem-solving performance.

LAWRENCE M. FAGAN

Dr. Fagan received his B.S. degree from M.I.T. with an interdisciplinary program of computer science, psychology, and decision-making courses. He received his Ph.D. from Stanford in 1980, where he continued research in the Department of Medicine and the Department of Computer Science. He recently received his M.D. from the Ph.D.-to-M.D. program at the University of Miami. He has returned to Stanford as a senior research associate in the Department of Medicine. His research interests include computer-based therapy planning and knowledge representation of temporal events.

EDWARD A. FEIGENBAUM

Professor Feigenbaum is a professor of computer science at Stanford University. He is co–principal investigator of the Heuristic Programming Project at Stanford, a leading laboratory for work in knowledge engineering and expert systems. His work on the DENDRAL program, beginning

in 1965, initiated these fields of applied artificial intelligence. Dr. Feigenbaum also heads the SUMEX-AIM facility, the national computer facility for applications of artificial intelligence to medicine and biology, established by NIH at Stanford University.

FERNANDO GOMEZ

Professor Gomez was born in Arahal, Spain. He received his *licenciatura* in philosophy from the University of Valencia in 1972 and his M.A. in romance linguistics and his Ph.D. in computer science from Ohio State University in 1974 and 1981, respectively. He is currently an assistant professor of computer science at the University of Central Florida. His current research interest is the study of the comprehension of scientific discourse. In particular, he is interested in the acquisition of knowledge via natural language and in how commonsense knowledge and reasoning vary as a result of their interaction with new pieces of knowledge.

G. ANTHONY GORRY

G. Anthony Gorry received his Ph.D. in computer science from M.I.T. in 1967. From then until 1975 he was an associate professor at the Sloan School of Management and in the Department of Computer Science at M.I.T. At M.I.T. he conducted research in the use of computers to improve decision making, with particular emphasis on medical problems. He is now Vice-President for Institutional Development at Baylor College of Medicine. His broadly based interests continue to include the analysis of clinical cognition and the analysis of health policy.

PAUL E. JOHNSON

Professor Johnson received his Ph.D. from Johns Hopkins University in 1964. He is currently a professor of management sciences and psychology and a faculty member in the Center for Research in Human Learning at the University of Minnesota. He specializes in methodology for the study of expertise in complex decision environments. His recent work has focused on investigation of expert problem solving in several professional fields including medicine, science, law, engineering, and management.

JEROME P. KASSIRER

Dr. Kassirer received his M.D. from the University of Buffalo in 1957 and trained in internal medicine and nephrology in Buffalo and at New England Medical Center in Boston. He is a professor and associate chair of

the Department of Medicine at Tufts University School of Medicine and associate physician-in-chief at New England Medical Center. Dr. Kassirer's research interests include clinical applications of decision analysis and descriptive analysis of the problem-solving tactics of expert physicians.

CASIMIR A. KULIKOWSKI

Professor Kulikowski is a professor of computer science and associate director of the Laboratory for Computer Science Research at Rutgers University. Since 1972 he has also been a senior investigator and associate director of the Rutgers Research Resource on Computers in Biomedicine. His research is in the fields of artificial intelligence and pattern recognition, with emphasis on expert systems and their applications. He has directed several collaborative projects for the development of expert medical consultation systems.

JOHN KUNZ

Dr. Kunz worked as a biomedical engineer before entering the Ph.D. program in computer science at Stanford University, which he completed in 1984. He currently works for IntelliGenetics and continues his interests in using artificial intelligence as the basis for computer-assisted decision-making systems.

REED LETSINGER

Dr. Letsinger received his Ph.D. in philosophy from Stanford University in 1976 and his M.S. in artificial intelligence from the computer science program in 1981. He is currently working at Hewlett-Packard, where his present research focus is the application of expert systems technology to engineering problems.

RANDOLPH A. MILLER

Dr. Miller has been associated with the INTERNIST/CADUCEUS project since 1973. He received his M.D. from the University of Pittsburgh in 1976. After completing internal medicine house-staff training in 1979, he joined the University of Pittsburgh School of Medicine as an assistant professor of medicine. His research involves the further development of the INTERNIST/CADUCEUS computer-assisted medical diagnosis system.

JAMES H. MOLLER

Dr. Moller received his M.D. from Stanford University in 1959. He received house-staff training in pediatrics at the University of Minnesota Hospital and then served as a fellow in pediatric cardiology at that institution. He is currently Paul F. Dwan Professor of Pediatric Cardiology at the University of Minnesota. One of his current research interests is understanding the development of medical expertise, of which one component is the development of computer-assisted diagnostic programs in pediatric cardiology.

JACK D. MYERS

Dr. Myers received his M.D. from Stanford University in 1937 and had graduate training at Stanford and Harvard. He has served on the faculties of the schools of medicine at Emory, Duke, and Pittsburgh. During the past decade, as a professor-at-large at the University of Pittsburgh, he has cooperated in devising a computerized consultation system in internal medicine, INTERNIST/CADUCEUS.

RAMESH S. PATIL

Professor Patil received his Ph.D. in computer science from M.I.T. in 1981. He is currently an assistant professor in the Laboratory for Computer Science at M.I.T. His current research interests include the application of artificial intelligence techniques to medicine, with an emphasis on fundamental issues of representation and reasoning with causal knowledge and explanation of consultant program reasoning.

STEPHEN G. PAUKER

Dr. Pauker received his M.D. from Harvard Medical School in 1968 and trained in internal medicine and cardiology at New England Medical Center, Boston City Hospital, and Massachusetts General Hospital. He is currently a professor of medicine at Tufts University School of Medicine and chief of the Division of Clinical Decision Making at New England Medical Center. His research interests involve the applications of decision analysis to clinical medicine and the development of computer-based decision aids.

PETER POLITAKIS

Dr. Politakis joined Digital Equipment Corporation in 1982, as principal engineer, after completing his Ph.D. and SEEK research in the Computer Science Department at Rutgers University. Prior to that, he was at M.I.T.'s

Lincoln Laboratory for four years. His current research interest is the development of knowledge acquisition and validation techniques for expert systems.

HARRY E. POPLE

Dr. Pople studied electrical engineering at M.I.T. and did his graduate work under Herbert Simon in the systems and communications sciences program at Carnegie-Mellon University. A member of the University of Pittsburgh faculty since 1969, his principal research interests have been in the study of intelligent decision support systems, with particular emphasis on applications in medicine and management.

ARIN SAFIR

Dr. Safir began his career working as an electrical engineer, then became interested in medicine and received his M.D. from New York University in 1954. He trained in ophthalmology at the New York Eye and Ear Infirmary during 1956–1959. Dr. Safir also completed a fellowship in physiological optics at the University of Cambridge, in Cambridge, England, in 1962. He was the director of the Institute of Computer Science at Mount Sinai Medical Center in New York City for six years and is currently a professor and chief of ophthalmology at the University of Connecticut School of Medicine in Farmington.

WILLIAM B. SCHWARTZ

Dr. Schwartz, a professor of medicine and Vannevar Bush University Professor at Tufts School of Medicine, is a graduate of Duke University School of Medicine and did his postgraduate training in internal medicine at the University of Chicago and at Peter Bent Brigham Hospital. He was for many years head of the nephrology division at Tufts–New England Medical Center and from 1971 to 1976 was chairman of the Department of Medicine at Tufts. His subsequent work has been in the area of decision analysis and in the application of artificial intelligence techniques to clinical problem solving.

EDWARD H. SHORTLIFFE

Dr. Shortliffe received his Ph.D. in medical information sciences and his M.D. from Stanford University in 1975 and 1976, respectively. After completing his MYCIN research, he undertook house-staff training in internal medicine at Massachusetts General Hospital and Stanford University Hos-

pital. He is currently an assistant professor of medicine and computer science at Stanford University. His research interest is the development of computer-based clinical consultation systems for use by physicians.

DAVID B. SWANSON

Dr. Swanson received his Ph.D. in educational psychology from the University of Minnesota in 1978. He is currently an assistant director of the American Board of Internal Medicine in Philadelphia. His research interests are in the psychology of clinical decision making, the measurement of clinical competence, and computer applications in medical education.

WILLIAM R. SWARTOUT

Dr. Swartout received his Ph.D. in computer science from M.I.T. in 1981. He is currently a member of the research staff of the Information Sciences Institute of the University of Southern California. His research interest is the development of techniques that will allow programs to explain their reasoning, making them more understandable to both their users and their implementers.

PETER SZOLOVITS

Professor Szolovits received his Ph.D. in computer science from California Institute of Technology in 1975. He is currently an associate professor in the Department of Electrical Engineering and Computer Science at M.I.T. A specialist in artificial intelligence with an emphasis on medical applications, he is currently concerned with fundamental issues of representation and reasoning, including protocol analysis to discover how clinicians reason about probability and causality, and with programs that model human expert performance in some areas of medical diagnosis and care.

SHOLOM M. WEISS

Professor Weiss received his Ph.D. in computer science from Rutgers University in 1974. He is currently an associate research professor of computer science at Rutgers University and senior investigator in the medical modeling and decision-making group of the Rutgers Research Resource on Computers in Biomedicine. His current research interests include the development of generalized approaches to designing expert systems and the application of these systems to real-world problems in medicine and other domains.

Preface

In August of 1980, Stanford University was the site of the annual workshop on artificial intelligence in medicine (AIM). This specialized area of medical computer science research had been born almost ten years earlier with the near-simultaneous development of AIM research groups at Massachusetts Institute of Technology (in collaboration with physicians from the Tufts–New England Medical Center), the University of Pittsburgh, Rutgers University, and Stanford. These small groups of computer scientists working in the field were drawn together naturally by their common interests and by the establishment of the SUMEX-AIM network (Stanford University Medical EXperimental Computer for Artificial Intelligence in Medicine). This computing resource was established by the Biotechnology Resources Program of the NIH in 1974 and consisted of a pair of computers, one at Rutgers and one at Stanford, linked by a communications network. The funding for SUMEX-AIM not only provided computing power for researchers exploring the potential of artificial intelligence techniques in medicine but also established a series of annual workshops so that the investigators could gather to share their insight, results, and ideas regarding approaches to the difficulties they encountered.

The 1980 workshop was the first at Stanford; the five earlier sessions had been held at Rutgers University in New Jersey. Because of a growing interest in computers generally, and in artificial intelligence in particular, among local physicians and medical faculty, we decided to organize a public AIM tutorial to be held immediately following the small workshop. Most of the field's leaders were going to be there, and it seemed logical to extend their stay so that a public series of lectures could be held to acquaint a medical audience with the progress, current status, and potential of the emerging discipline. We were delighted by the interest in the program, by the excellent attendance at the two-day tutorial, and by the positive attitudes of the attendees (Teach and Shortliffe, 1981).

One of the lessons of that tutorial was the need for a readily available collection of readings to describe the first decade's work in the field. Those articles that had appeared were scattered in a number of publications, some from the medical literature and others from computer science journals. There had been no effective effort either to bring together the key articles or to describe the historical progression of work in the field.

As the word spread about the 1980 tutorial, we received increasing numbers of requests for such a collection, and the idea for this volume emerged. There was a clear need for a survey of key AIM activities, par-

ticularly in light of the success and visibility of the early efforts and because of the frequent failure to appreciate the significant barriers that remain to be overcome before widespread clinical impact will be achieved. This book is, accordingly, an attempt to address those issues. The authors and publishers of the original articles have kindly permitted us to reproduce them here, generally with only minor modifications to correct any inaccuracies that were discovered in retrospect. We have, in addition, included a new introductory chapter that defines the field of artificial intelligence, outlines its relevance to medicine, and identifies the key research issues that have guided AIM work and continue to do so. Each chapter is preceded by a brief introduction that outlines our view of its contribution to the field, the reason it was selected for inclusion in this volume, an overview of its content, and a discussion of how the work evolved after the article appeared and how it relates to other chapters in the book.

It is important to note that this book is by no means an exhaustive review of all AIM work during the period 1971–1981. Several fine pieces of work could not be included because of space limitations. We have accordingly tried to provide references to additional key articles throughout the volume. Those included were selected to provide a broad coverage of issues, as well as to exemplify what we consider to be some of the best and most influential work in the field.

The papers here tend to be more technical and detailed than those that appeared in a recent shorter volume on the subject (Szolovits, 1982). We have also provided a comprehensive index, a name index, and a bibliographical listing of all references cited throughout the volume. These additions have been designed to make the issues accessible to interested readers, particularly physicians, who may not have had previous experience with artificial intelligence. The chapters are organized in a loosely chronological way, with surveys and general system descriptions near the beginning and more recent work toward the end. The title of the volume has been selected to make it clear that we see the AIM field as a young and emerging discipline. Our views of the future, with an emphasis on the challenges as well as the promise that lies ahead, are the subject of the closing chapter.

This preface would be incomplete if we did not acknowledge and express our gratitude for the remarkable assistance we have had in preparing the book. The authors of the individual chapters dug through their archival records and provided us with copies of original manuscripts (often in electronic form) and the original figures that are reproduced in some of the chapters. Chapter introductions have been in large part adapted from the original abstracts; the authors helped us greatly by reviewing this material and correcting and augmenting the historical information (and we thank Paul Feltovich especially for providing his interesting sketch of the DIAGNOSER project).

We are also grateful to Michael Morgan of Addison-Wesley for his encouragement in bringing the volume to print, to Christopher Lane for

his patient assistance in writing special programs for document formatting, to Darlene Vian for her reliable managerial assistance, to Joan Differding for her help preparing new figures when the originals were not available, and to Jane Hoover for her meticulous care in reviewing and copyediting the submitted manuscript. It is to Barbara Elspas, however, that special thanks are due. She worked many long hours editing the chapters, compiling the bibliography, and generating an excellent manuscript that greatly facilitated the publication process. We are extremely grateful for the conscientious assistance she provided.

Stanford University W. J. C.
May 1984 E. H. S.

1

Introduction: Medical Artificial Intelligence Programs

William J. Clancey and Edward H. Shortliffe

1.1 Approaching Medical Artificial Intelligence

In approaching most fields of scientific inquiry, it is useful to consider two basic questions:

- What methodologies and assumptions do the researchers share?
- What issues and concerns distinguish the research projects from one another?

This introduction sketches out answers to these questions for the field of artificial intelligence applied to medicine (AIM), as viewed in the early 1980s, approximately a decade after the field's initial development. Our intent is to provide a common ground for appreciating what makes the work reported in this book special as a whole and for understanding the diverse terminology and research emphases of the individual chapters. In contrast with the history-oriented discussion one can find in Szolovits (1982), we discuss the dimensions by which one can recognize and study AIM programs. Thus we have two important goals: introducing the reader to a programming approach and relating the programs to one another through recurring research issues.

After a brief historical introduction, we define knowledge-based programming, provide dimensions for characterizing and comparing these programs, and outline the state of the art.

1.2 What Is Medical Artificial Intelligence?

> Artificial intelligence (AI) is the part of computer science concerned with designing intelligent computer systems, that is, systems that exhibit the characteristics we associate with intelligence in human behavior—understanding language, learning, reasoning, solving problems, and so on (Barr et al., vol. 1, 1981, p. 3).

Medical artificial intelligence is primarily concerned with the construction of AI programs that perform diagnosis and make therapy recommendations. Unlike medical applications based on other programming methodologies, such as purely statistical and probabilistic methods, medical AI programs are based on symbolic models of disease entities and their relationships to patient factors and clinical manifestations.

In the early 1960s researchers in AI had focused on problem solving in game playing, image recognition, speech understanding, and language understanding (Feigenbaum and Feldman, 1963). During this time some general problem-solving principles were formalized, such as reducing a complex problem to a network of subgoals. However, it was discovered that most of the difficulty in achieving intelligent behavior was in collecting and storing a large *knowledge base* of facts specific to the problem area. This result was confirmed by the success of a few large programs in scientific (Lindsay et al., 1980) and mathematical (MATHLAB, 1974) areas. At this time, the explosion in medical knowledge was forcing physicians to specialize increasingly and was often overwhelming to those who tried to remain generalists. Medicine was therefore a logical field in which to apply practically the developing knowledge-based techniques.

Large domain-specific problem solvers came to be known as *consultation programs*, for they fit the image of an expert-specialist who is asked to provide advice about some difficult problem. In medicine, the "problem" would typically be a patient with an illness to be diagnosed. By the late 1970s, these programs became known as *expert systems*, although that remains a somewhat generous characterization in light of the limitations of these programs (as described in the following chapters).

Four major systems were developed by 1975—PIP, CASNET, MYCIN, and INTERNIST—all described in this book. As early efforts, they are prototypes directed at two questions: What are the issues involved in designing a consultation program (e.g., what would make such a program acceptable to physician users)? What is the nature of the expertise to be formalized (e.g., how can factual and judgmental knowledge be integrated)?

Most AIM programs are directed at serious practical applications, but there are other reasons for doing the research. Some AIM researchers

have constructed programs for the sake of better understanding human problem solving in general. In this respect some of the most important results from this work are in the area of psychology (e.g., Chapter 12; Kassirer and Gorry, 1978; Kuipers and Kassirer, 1983; Swanson et al., 1979). On the other hand, constructing an AIM program is also fundamental medical research. The knowledge that is formalized in these programs does not come straight from textbooks; some is heuristic, developed through experience and passed down by apprenticeship. Thus, in their collaboration with physicians, medical AI researchers are helping to formalize medical knowledge. Importantly, the formalization process goes beyond accumulating facts to include new ways of structuring medical knowledge in general, such as formulating a language for describing diseases on multiple levels of detail (see Chapter 14). There is good reason to believe that such research will ultimately be of benefit to medical educators (Shortliffe, 1983).

Almost all AIM programs have been developed at universities, either in a medical school or in collaboration with a nearby school of medicine. The projects typically involve teams of collaborating computer scientists and physicians. On rare occasions, the computer scientist is also a physician and so can easily understand the medical issues or supply his or her own expertise (see Blum's work in Chapter 17). In many cases, the research has benefited by having computer scientists, initially unfamiliar with the medical field, approach the problem areas freshly, serving as "investigative reporters" who study what physicians know and how they solve problems.

How has the research advanced over the past decade? Most of the progress centers on the problem of *representing* medical knowledge. The first efforts concerned both the formalization of specific disease knowledge (e.g., how to distinguish between bacterial and viral meningitis) and the kinds of relations that physicians make among findings and diseases (e.g., that one disease is a complication of another). Two other areas of research are *knowledge acquisition*—interacting with an expert to formalize his or her knowledge and problem-solving procedures—and *explanation*—tracing back conclusions to the data and justifying the reasoning process. The capability of a program to perform these operations is now understood to depend very much on the adequacy of the underlying knowledge representation. Finding the right kinds of distinctions (for example, how to accurately model a causal process or how to state a diagnostic procedure) has been the focus of much of this research. Given finer-grained, more detailed models, the emphasis then shifts to formulating improved diagnostic operations for *manipulating* the knowledge to solve problems [the thrust of CADUCEUS (Pople, 1982) and developments from ABEL (Patil et al., 1982)].

We have briefly outlined how medical AI grew out of new directions in AI research. It is fair to say that the contributions have been at least as strong in the other direction. Efforts such as INTERNIST (Chapter 8) and

MYCIN (Chapter 5) led to a generalization of techniques that became known as *knowledge-based programming*. The idea of building expert systems like MYCIN has spread to almost every imaginable application (e.g., geology, structural analysis, and tax law), thereby making expert systems currently the largest subarea in the field of artificial intelligence. Moreover, even AI researchers specializing in subareas such as natural language understanding have come to realize that a knowledge base must be the foundation of any reasoning system (Carbonell, 1979). The wave of results is now flowing back to medical applications, as representation and design ideas developed in other scientific areas are picked up and adapted. Within AI, knowledge representation has become an area of study in its own right.

Finally, we should point out that just as AI takes in diverse areas such as signal understanding, image processing, and robotics, medical AI has parallel subfields corresponding to patient-monitoring systems, x-ray and ultrasound imaging systems, and prosthetic devices. In order to focus the collection of papers in this book, we have chosen to restrict the topic to systems concerned primarily with diagnosis and therapy—medical expert systems.

1.3 What Is a Knowledge Base?

The programs described in this book exemplify knowledge-based programming applied to medicine. The goals and techniques of knowledge-based programming are considerably different from other kinds of programming. Ways of analyzing and comparing traditional programs are not always relevant. Moreover, if the basic foundations of these programs are not understood, it is difficult to understand their limitations and potential.

One way to start is to understand that researchers in this field draw a distinction between *knowledge* and *data* (e.g., see Chapter 3). *Data* consist of records of information, such as patient records in a hospital, equipment maintenance records, or scientific measurements such as weather data. Data can be either symbolic (e.g., the names of patients) or numeric (e.g., temperatures). In computer science, the term *record* has generally become synonymous with the pattern that defines what might be recorded for each entry, such as the patient's name, his or her location in the hospital, date of entry, etc. A *data base* contains a set of such records.

By common usage, *knowledge* is anything you know, so it surely includes what we find in data bases. But knowledge also includes how things are related, what general patterns exist, why there are relations and patterns, as well as procedures for solving problems. For example, a data base might record information for a particular patient population from which correlations between drug therapy and adverse reactions or side effects could

be derived statistically. A related *knowledge base* might have the general rule "If the patient is a child without a full set of adult teeth, don't prescribe tetracycline," as well as a causal model that explains how the process of chelation occurs, and perhaps a procedure that says "Consider contraindication rules after making a diagnosis and before prescribing therapy." So in an important sense, a knowledge base is general. Its records are about disease processes, diagnosis, and therapy in general, not about particular patients. Some knowledge can be derived analytically from data bases, but some is based on experience and is judgmental or heuristic.

Certain concerns about data bases, such as organization and accuracy, carry over to knowledge-based programming. But a knowledge base is different because it is never complete. A knowledge base is a kind of model: it can be interpreted to predict or explain behavior in the world. Thus diagnosis is based on a causal explanation of what is happening to the patient, and therapy is based on predictions about how the disease process can be modified. As models, knowledge bases are incomplete in that they are *approximate* and *omit levels of detail.*

Simple knowledge bases, like simplified models, might apply only to simple versions of problems; for example, a medical system might not be designed to handle multiple diseases. Handling multiple diseases might require modeling how one disease could cause another. A medical model might also be incomplete because it is based on empirically observed correlations rather than on well-understood causal processes. Since medical science is continuously evolving, new understanding will modify the rules for interpreting symptoms and prescribing therapy. Finally, because medical knowledge bases contain judgmental knowledge relating to social costs and benefits, they always reflect the values of their designers, which might change over time. In general, knowledge bases are incomplete, approximate, and biased models of the world.

Knowledge bases are also incomplete with respect to level of detail as a model. We can always ask why a statement is true; in medicine we would then delve successively into biology, chemistry, and physics. Since we do not represent everything we know on all levels of explanatory detail (and at some level of detail *everyone* experiences a failure of detailed mechanistic understanding of biologic processes), the knowledge bases we build are necessarily incomplete. One reason for this incompleteness is that there is no practical way to build systems today that know more than a fraction of what any physician knows about the body and how it works. There is just too much knowledge, and we are still struggling to formalize even small portions of it. A second reason for leaving out levels of detail is that useful problem-solving performance can usually be produced even if we leave out pathophysiological knowledge about disease. However, when we push a program to resolve multiple diseases or to deal with an unusual presentation of a disease (one that tends to "violate the rules"), these simplistic models break down.

1.4 Design Criteria for Medical Knowledge Bases

The incompleteness of medical knowledge bases requires that they be built incrementally, both to allow for the difficulty of building a complete model at any time and to allow for improvements to the knowledge base as human experts learn and social judgments change. Perhaps the most important design feature that makes a knowledge base easy to maintain over time is *modularity*: ideally, there should be no side effects or complex interactions among parts of the knowledge base.

Modularity in a sense boils down to a matter of indexing. The level at which we wish to change the system should be easily accessible, so we do not have to wade through complex code to make a change. In knowledge-based systems, one solution is to index knowledge according to how it is used and modified during reasoning. For example, in a rule-based system, it is convenient to index rules by the disease diagnoses they support.

Modularity and indexing suggest that the knowledge base be *structured* according to dimensions that make it easy to use and maintain. To have an indexing scheme, you need primitives for the dimensions of indexing, just as we have the idea of alphabetic ordering for assembling phone books. In medical AI we are led to study and formalize primitives such as *subtype, cause, etiology,* and *specialization.* An important open question is to determine the set of primitives that could be used to describe the temporal properties of any disease.

Of course, creating a well-structured knowledge base is just part of building a consultation system. How will the knowledge be applied to solve problems? To give a very practical example, suppose you want to confirm the presence of a particular disease. Should you seek evidence for all of its manifestations? In what order should you consider them? When should you focus on another hypothesis? You need a *procedure* for doing diagnosis. This procedure is often called *control knowledge* because it controls how the specific knowledge about diseases is applied to solve problems.

The primitives for representing control knowledge are different from those for representing disease knowledge. Concepts like *iteration, steps, subroutines,* and *conditional actions* are useful. Stating control knowledge *explicitly* and *separately* from the disease knowledge offers a big bonus—the disease knowledge can be used in multiple ways; different procedures can be used to interpret it. For example, a knowledge base might be used both to provide consultative advice and to tutor a student (see Chapters 5 and 11). Such a separation also enhances the ability of a program to explain its reasoning, an important concern for the acceptability of the system to its ultimate users. But not all systems are designed this way.

To summarize the important points we have made about the design of knowledge bases:

- A knowledge base is inherently incomplete; it is common for only one level of knowledge to be represented.
- To allow for incremental development, easy maintenance is important.
- Maintenance and explanation are enhanced by modular, well-structured, explicit statements of disease relations and diagnostic procedures.
- A well-structured knowledge base can be used in multiple ways.

1.5 Basic Concepts of Knowledge Representation

In the study of knowledge-based programs, such as the dozen or so systems described in this book, it is useful to consider a number of representation issues.

First, it is important to make a distinction between *what kind of knowledge is represented* and the *representation language* itself. A researcher is indicating what kind of knowledge is represented when she says, "CENTAUR's knowledge base contains descriptions of prototypical patterns of diseases" (Aikins, 1980; 1983). She is describing the representation language when she says, "Disease manifestations are represented as 'prototype components,' as slots in a 'frame.' " In general, it is most helpful to understand what kind of knowledge is represented in a system before trying to grasp the representation language. For example, we can compare CASNET (Chapter 7), ABEL (Chapter 14), and RX (Chapter 17) in terms of the kinds of causal facts about diseases that each represents. The implications for problem solving can then be considered: how might RX use ABEL's multiple-hierarchical representation to better explain data base correlations, for example?

A representation language is just a notational device. The important properties of a representation language include the *brevity* and the *explicitness* with which certain kinds of facts can be stated. For example, when approaching a system for the first time, it is useful to ask how causal, taxonomic, and temporal relations of disease are represented. In a system like MYCIN, such facts are stored only implicitly, but problem-solving behavior can be modified in a direct, concise way. It is also important to ask how the diagnostic procedure is represented. Can the knowledge base be thought of as a network that is interpreted by a separate diagnostic procedure (as in ABEL, CASNET, INTERNIST, NEOMYCIN, PIP, RX, and XPLAIN)? Or is the procedure implicit, inseparable from the knowledge base (as in the original Digitalis Therapy Advisor, MDX, MYCIN, PUFF, and VM)?

It is important not to get confused about external representations (diagrams linking findings and diseases), technical arguments about formalism (rules versus causal networks), and the description of the kind of knowl-

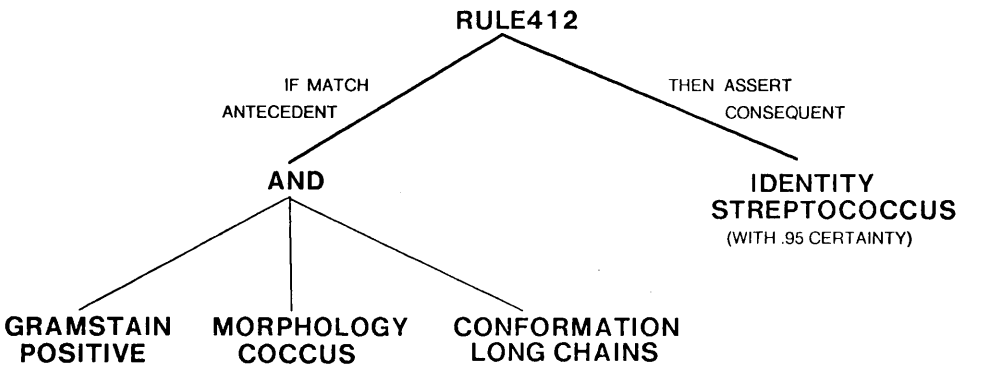


FIGURE 1-1 MYCIN rule viewed as nodes and links.

edge represented (e.g., disease prototypes). All of the jargon aside, what kinds of knowledge are factored out and represented explicitly? A beginner first studying this field should try to understand what the drawings and figures are showing about the kind of knowledge represented before worrying about the technical AI terms (such as production rule and frame). For example, Figures 7-1, 13-1, 14-7, 15-2, and 16-5 describe the kind of knowledge represented. When an internal name is mentioned (such as MYCIN's LABDATA), the important thing to understand is when the label is applied and how it is interpreted by the program (if a finding is marked as LABDATA, MYCIN will ask the user to supply a value before trying to make inferences from what it already knows).

In thinking about an internal knowledge representation, remember that at a basic level a knowledge base is completely describable in terms of nodes and links. You might first figure out what can be a node and how the nodes are linked, then try to pin down how the links are used by the control knowledge. To give a simple example, in a rule-based system such as MYCIN each rule is a conditional expression, which can be represented as a node linking an antecedent (IF part) to a consequent (THEN part) (Figure 1-1). (The rule represented in the figure allows MYCIN to conclude that an organism is almost certainly a streptococcus if it is a gram-positive coccus growing in long chains.) That is the static description. During problem solving, if MYCIN determines that the antecedent of a rule is satisfied, it asserts (adds to its data base) what is specified by the consequent. That is how a rule node and its links are interpreted. The next step is to understand when this operation would be performed on a particular rule node (i.e., when a rule is invoked), what else happens when a new assertion is made, and so on.

It is almost always useful in computer science to think in terms of processes. Ask yourself:

What is the input?
(In MYCIN, a rule);
What is the process?
(Determining if the antecedent is satisfied and making assertions);
What is the output?
(An updated data base of facts and beliefs about the patient).

In AIM knowledge bases, a *node* corresponds in general to medical concepts, for example, patient data and diseases. It is important to understand how the nodes are "marked" as a consultation proceeds. If a particular patient has a disease manifestation, for example, the internal representation is marked to indicate this fact. Furthermore, a link might be added to a particular disease node, indicating that, in this patient, this manifestation is believed to be caused by the disease. In this way a *patient-specific model* (see Chapter 14) is constructed as a constellation of possible findings, diseases, and connections among them. An example of a patient-specific model for the MYCIN domain is shown in Figure 1-2.

Another helpful consideration is to remember that a node might stand for an object (such as a CSF culture), a disease process (an infection), a patient-state description (increased brain pressure), an event (the onset of a headache), or a hypothesis (the belief that the patient has meningitis). Also, a node might stand for a *concrete* entity, such as a particular CSF culture, or an *abstract* one, such as the concept of cultures in general. In this way, general classes or categories can be described in a knowledge base.

A *link* is a relation between nodes. A *causal link* between two disease nodes indicates that one disease causes the other. From a simple perspective, links are labeled pointers that group findings and diseases into networks. For example, a taxonomy of diseases might be constructed by linking diseases with a *subtype link.* Links can also indicate spatial relations, levels of detail, examples of a general concept, etc. An issue of major concern is the interpretation of a link during problem solving. If a link indicates that a finding is "caused by" a disease, for example, does this mean that the program will list this disease as a diagnosis in its output? Is the link annotated in some way to indicate the conditions under which the causal process holds? If there are other causal links (and hence explanations) for this finding, how are they taken into account? The complexity of links and how belief about diseases is propagated through a network are major AIM issues.

1.6 Dimensions for Comparing Knowledge-Based Systems

Knowledge-based systems can be studied and compared along the following dimensions (with illustrative systems indicated in parentheses):

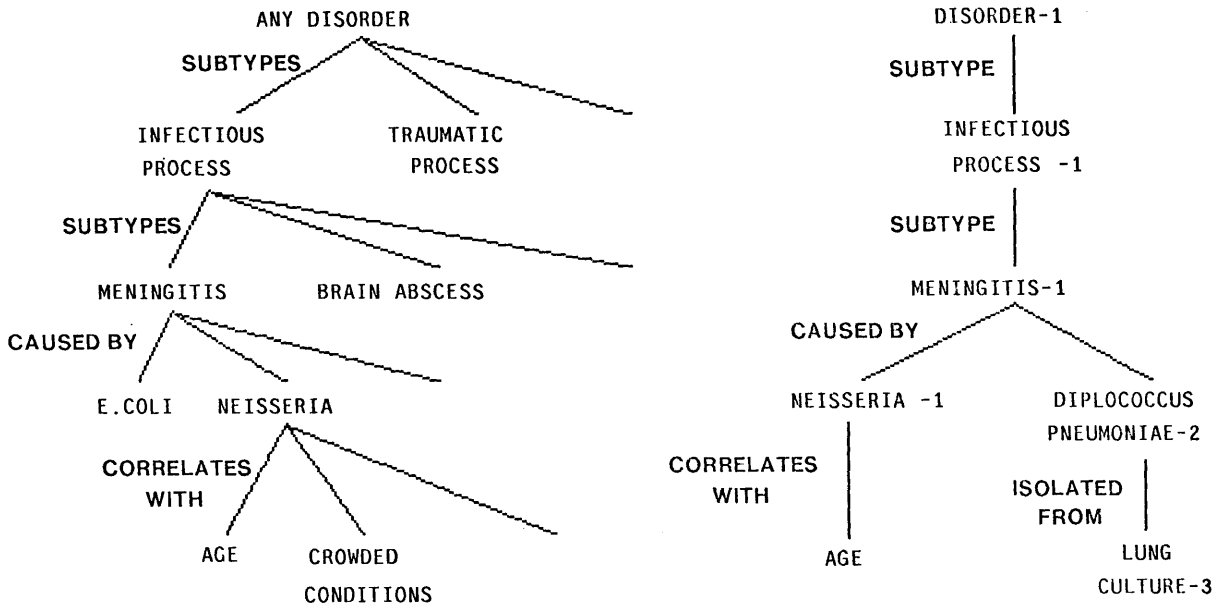


FIGURE 1-2 How a patient-specific model relates to general disease knowledge.

- *Content:* What kind of medical knowledge does the knowledge base contain? Programs typically contain (heuristic) links between findings and diseases (MYCIN). They sometimes contain pathophysiological descriptions of disease processes (CASNET). They rarely contain anatomical descriptions (CADUCEUS).
- *Structure:* How are the nodes and links organized? Programs typically contain hierarchies of various kinds (CENTAUR, INTERNIST, MDX, NEOMYCIN). They sometimes contain levels of abstraction (ABEL). No current programs contain multiple models or perspectives describing a single disease process.
- *Hypothesis formation and evaluation:* How does the program use links to make inferences (the *reasoning strategy*)? Most programs generate hypotheses from given data and use a hypothesis-directed *questioning strategy* (CASNET, INTERNIST, MDX, PIP). Many consider diagnoses in a focused, nonexhaustive way (INTERNIST, NEOMYCIN, PIP). A few attempt to model human reasoning (MDX, NEOMYCIN, PIP).
- *Management of uncertainty:* How does the program represent and cope with uncertain information? Most programs represent hypotheses with some degree of uncertainty, using a *scoring mechanism* for combining evidence and comparing hypotheses (CASNET, INTERNIST, MYCIN). No current programs cope with inconsistent evidence by reasoning about the justification for inferences.
- *Data collection:* How does the program acquire information about the problem? Most programs ask questions of the user, requiring keyboard input. No current programs allow a true mixed-initiative interaction. No current consultation programs can accept data from on-line medical data bases, although some data interpretation systems have been interfaced with patient monitoring devices (PUFF, VM) and other analytic devices (EXPERT/Electrophoresis).[1]
- *Explanation and knowledge acquisition:* What methods are available for building and testing the knowledge base? Some programs have a means of displaying the network in English form (MYCIN). Many have some form of "audit trail" so the reasoning can be traced back in debugging (MYCIN, NEOMYCIN, XPLAIN). Only a few programs have a user model to facilitate the interaction (GUIDON, XPLAIN). None of the programs truly learn from experience, but several detect patterns in the knowledge base (MYCIN), a patient data base (RX), or a case library (SEEK) as an aid in knowledge acquisition.
- *Meta-knowledge:* What is implicit in the knowledge base? What does the program know about its own design? Most programs attempt to separate

[1]The HELP system (Pryor et al., 1982) is a good example of a non-AI program that assists a physician by accessing an on-line data base. A recent AI system named ONCOCIN (Shortliffe et al., 1981) uses an on-line data base of patient information, but requires that *current* data be entered by the user.

out disease knowledge from the diagnostic procedure. Some programs have explicit knowledge about the principles underlying the network (ABEL). A few programs have abstract, nonmedical knowledge about their procedures and representation (MYCIN, NEOMYCIN).

The list above is meant to illustrate features to look for when studying medical systems. The progression from typical to rare parallels what many researchers would hold to be desirable and helps identify many of the key research areas for the next decade.

1.7 Are Knowledge-Based Systems Textbooks of the Future?

How is an ideal medical knowledge-based system different from a medical textbook? Understanding the differences might help the reader understand what researchers are trying to do in these programs, what makes their task difficult, and what they might potentially achieve for humanity.

We proceed by describing medical textbooks according to the dimensions for comparing knowledge-based systems given above:

- *Content:* A medical textbook typically contains all kinds of anatomic, disease process, and heuristic knowledge. As stated above, AIM systems currently tend to model only high-level associations between findings and diseases.
- *Structure:* Organization of textbooks is always of paramount concern. Different textbooks might organize the same knowledge along different dimensions (e.g., either by disease entity or by presenting complaints). However, sharp distinctions are generally not made about the kinds of knowledge being presented; diagnostic procedures tend to be interwoven with medical facts. Knowledge is usually not clearly stated on multiple levels of detail; a given textbook generally adopts one viewpoint. There is little discussion of the epistemological terms used, such as causality and subtype—they might even be used in a confused way. While this can also be true of programs, the requirements and trend for new systems is to articulate and be precise about these kinds of distinctions.
- *Hypothesis formation and evaluation:* Medical textbooks tend to give disease-specific relations for considering and confirming the presence of a disease. Consideration of the practical problems of diagnosing multiple diseases, separating complication from cause, ruling out diseases, weighing evidence—all of which must be formalized in a knowledge-based system—is generally not treated in a general way in textbooks. Typically, only tips and rules of thumb are given. Most importantly, knowledge-

based systems are *programs* that can solve problems. Textbooks sit there inertly, relying on you to search for the relevant facts and then to infer solutions on your own.

- *Management of uncertainty:* Books are often vague about how to interpret evidence, using words such as *often, suggest,* and *rarely seen,* without specifying a consistent interpretation for these terms or how to use them to solve problems. Programs require the discipline of at least some kind of "scoring function" for assigning weights and combining evidence, although this is often an *ad hoc* scheme that is adjusted until it works well enough. Resolving multiple diagnoses with a large number of findings requires information about importance, frequency, risk, cost, discomfort, etc., as well as the development of a more precise understanding of the conditions of causal processes (as in Chapter 14)—crucial information for successful knowledge-based programs, but often ignored in textbooks.
- *Data collection:* Some medical textbooks have very good discussions of the problem of interviewing a patient. But given all of the commonsense knowledge involved, current knowledge-based programs cannot help the consultation user to supply accurate information. For example, they cannot explain how to recognize lethargy in a patient or how to distinguish between coccus in long chains and rods. Certain assumptions about the user population are generally not stated explicitly, so the systems are missing meta-knowledge about their own design. But knowledge-based systems can actually use their knowledge to collect data from a data base or to conduct an interview; a textbook obviously must leave these tasks to the physician.
- *Explanation and knowledge acquisition:* Textbooks use graphic techniques, as well as prose, to explain complex relations and disease processes. While a textbook might have a good glossary, you cannot ask it for clarifications. You cannot pose hypothetical questions or give it new knowledge and see how the answer changes (see Chapter 11).
- *Meta-knowledge:* Some authors do a good job of describing the organization of their book. Arbitrary perspectives are possible for making a physician self-aware, for example, about how to speak to a patient, how to interpret culture results, and how to organize multiple problems on paper. However, medical AI is contributing new meta-knowledge about the structure of disease knowledge and the abstract character of diagnostic procedures. These topics are typically sacrificed in books to an emphasis on facts and are also often ignored in formal medical education.

So, while textbooks and knowledge-based systems are both knowledge repositories, a program has many potential advantages. Because knowledge can be represented independently from its use, a given knowledge base can be interpreted for multiple applications. Of course, in talking

about knowledge-based systems above, we are referring to the knowledge base plus the interpretive procedures. These include procedures for doing diagnosis, for teaching, for learning, etc. In most cases, developing such procedures is at least as difficult as developing the knowledge base itself. The construction process usually goes on in parallel, even when the knowledge is represented independently from how it is to be used. Developing these procedures involves substantial research. For example, consider the difficulties of developing the domain-independent teaching rules of GUIDON (Chapter 11) and how this involves basic research in education, and consider how developing the domain-independent diagnostic rules of NEOMYCIN (Chapter 15) involves basic research in psychology and medicine. Each of these is an expert system–building task in its own right.

Taking an optimistic point of view, a knowledge base might be the basis of any *intelligent agent*—a consultant, tutor, librarian, or decision analyst—responding actively to the needs of its user, capable of explaining itself, and, most importantly, capable of learning from experience (Clancey, 1983a). A well-designed knowledge-based program can be easily revised and cheaply copied. With the refinement of knowledge-based systems, textbooks might become like parchments, as antiquated as a mechanical calculator when compared to a personal computer.

1.8 Implementation of Medical AI Systems

Implementation includes the programming language (software) and the computer (hardware) upon which a system is constructed. With only one exception, the programs described in this book are written in the LISP (LISt Processing) programming language. A good introduction to LISP for the layperson is Winston (1977). LISP is chosen by AI researchers as much for its natural capabilities for representing knowledge networks (a *list* is a linked set of nodes) as for the powerful environment for building large programs that is offered by most LISP dialects [e.g., Interlisp (Teitelman and Masinter, 1981)].

The exception described here is EXPERT (Chapter 20), which is implemented in FORTRAN. FORTRAN was chosen because it is faster and runs on many different kinds of machines. However, it should be noted that FORTRAN cannot easily be used directly. Encoding meta-knowledge about the knowledge network design and interpretive procedures requires building a language on top of FORTRAN that makes it easier to reference a node by its name or by the kinds of links it has. Thus some key features of the LISP language must be added to FORTRAN to make it useful.

Medical AI programs generally run on machines with large address spaces. With one exception (INTERNIST, Chapter 8) it is currently the interpretive and maintenance software, not the knowledge bases, that takes

up the space. This is likely to change as AIM systems grow in size and complexity. The current situation reflects both the youth of the field and the tediousness of building large knowledge bases. The software for these systems includes not only the program for interacting with the user during a consultation, but also knowledge acquistion and explanation programs. The address limitation makes it impractical to develop these programs on a 16-bit machine, typical of the small personal computers of the early 1980s. However, a system might be "downloaded" after development, particularly if user interaction is minimized (see Chapter 19). Over the past decade hardware costs have plummeted, and new personal machines have become available. The 32-bit "professional workstations" that run LISP and are just appearing on the market provide insight into the kinds of computing environments that will bring AIM systems to physicians in a cost-effective manner in the decades ahead.

1.9 What Is the State of the Art?

The dimensions for comparing programs are descriptive, but can be adapted to characterize the best that programs can do today. The following list gives some dimensions of quality with short descriptions of what representative programs have accomplished:

- *Performance:* Several systems that have been formally evaluated in statistical studies of their performance are CASNET, INTERNIST, MYCIN, and PUFF (Duda and Shortliffe, 1983). A typical finding is that program behavior is acceptable to 80% of the evaluators, but evaluators usually disagree as much among themselves. INTERNIST currently has by far the best capability to deal with a wide variety of problems.
- *User interaction:* MYCIN set the standards for user interaction in terms of providing spelling correction, a nicely laid-out question-answer format, and English input of simple questions. The ONCOCIN program (Shortliffe et al., 1981; Bischoff et al., 1983) is adapting standard ways of filling out a patient's chart to the demands of a consultation program.
- *Explanation:* MYCIN was the first consultation program to provide an explanation facility. The therapy program was redesigned to enable it to present concise explanations of the optimization process (Buchanan and Shortliffe, 1984). The tutorial features of GUIDON (Chapter 11) provide more individualized display of reasoning. XPLAIN (Chapter 16) deals with explanation of procedures and methods for providing multiple levels of detail. Structuring the knowledge representation for the purpose of explanation is the focus of NEOMYCIN research (Chapter 15).

- *Representational adequacy:* While representation is at the heart of all medical AI research, some programs have been constructed specifically to solve problems with earlier representations. Perhaps the most complex examples are the multiple disease relations of CADUCEUS (Pople, 1982), the levels of causal detail of ABEL (Chapter 14), the abstract metarules of NEOMYCIN (Chapter 15), and the refinement structure of XPLAIN (Chapter 16).
- *Actual use:* Of the AIM systems with which we are familiar, only PUFF (Chapter 19), EXPERT/Electrophoresis (Chapter 20), and ONCOCIN (Bischoff et al., 1983) are developed to the point of being used routinely by physicians.
- *Psychological model:* Programs developed specifically as models of human reasoning include PIP (Chapter 6), MDX (Chapter 13), NEOMYCIN (Chapter 15), and CADUCEUS (Pople, 1982). The program reported by Johnson et al. (1981) has been evaluated to determine its accuracy as a model.
- *Knowledge acquisition:* The state of the art is represented by the packages of EMYCIN/TEIRESIAS (Davis, 1979; van Melle, 1980) and the interactive features of SEEK (Chapter 18).

1.10 Organization of This Book

We close this introductory chapter with a brief discussion of the papers we have selected for the book. Because many excellent projects and papers could not be included, we urge the reader to make use of the extensive bibliography we have provided. The papers cited there will provide valuable additional insights regarding many of the issues raised in this volume.

The first chapter, by Gorry, introduces the rationale and advantages of applying AI approaches to medical problem solving. This is followed by an extensive survey of computer-based clinical decision aids; of particular interest is the description of traditional algorithmic, statistical, pattern recognition, and decision-theory approaches.

Kulikowski then provides an introductory overview to the early AIM systems, placing them in the context of the expert systems subarea of AI that partially grew out of them. These classic systems are then described: MYCIN (known for its use of rules), PIP (use of frames), CASNET (use of a causal-associational network), and INTERNIST (handling of multiple problems). The chapter by Szolovits and Pauker makes detailed comparisons of representation issues handled by these four programs.

The next chapters describe two medical developments from the MYCIN program: VM and GUIDON. [Buchanan and Shortliffe (1984) provide a complete survey of the MYCIN project and its spinoffs.]

Representation issues in modeling physicians' reasoning are considered by Feltovich's psychological study, MDX, and ABEL.

NEOMYCIN and XPLAIN are contemporaneous, second-generation systems designed to enhance explanatory capability. The approaches are complementary, so it is useful to consider these programs together.

Knowledge acquisition in the form of partially automated learning is considered by RX (based on statistical analysis of a data base) and SEEK (based on analysis of case experience).

The development of two practical, routinely used programs is described: the implementation of PUFF in BASIC running on a minicomputer and of the EXPERT/Electrophoresis system running on a microprocessor.

In the last chapter, we take a step back to consider some of the practical issues regarding the AIM field. To what extent will the complicated systems we are developing ever be used? What are the principal research challenges that remain? Will medical expert systems be viewed as beneficial tools, or as threats to physicians or to the sanctity of the physician-patient relationship? Difficult questions such as these can be answered more realistically now that we have had a decade of solid experience with AIM research and have had more time to observe the evolution of society's attitudes toward computers and the remarkable revolution in hardware technology. Both of these developments are changing our predictions about the future, and they permit an optimistic view of medical AI and its potential for beneficial clinical use.

2

Computer-Assisted Clinical Decision Making

G. Anthony Gorry

In the early 1970s, a small number of medical computing research groups simultaneously realized that the field of artificial intelligence offered potential solutions to problems that had previously constrained the effectiveness and acceptance of medical decision-making programs. At Rutgers University, this arose when Kulikowski, a computer scientist who had previously worked with statistical pattern-recognition systems (Nordyke et al., 1971), noted that a consultation system for the diagnosis management of glaucoma would significantly benefit from enhanced knowledge of physiology and causality. At Stanford University, the new approaches arose from earlier work applying AI to chemistry in the DENDRAL program (Lindsay et al., 1980) and from Shortliffe's and Buchanan's disenchantment with the interactive features of traditional diagnostic programs that were based on statistical techniques (Shortliffe and Buchanan, 1975). At the University of Pittsburgh, Pople's previous work with computer models of neuroanatomy and abductive logic (Pople and Werner, 1972; Pople, 1973) led to the symbolic models used in INTERNIST. Meanwhile, at the Massachusetts Institute of Technology and Tufts–New England Medical Center, Gorry, Schwartz, and others had undertaken notable work applying formal decision theory to medical problems. A pair of landmark papers appeared in the American Journal of Medicine *in 1973 (Gorry et al., 1973; Schwartz et al., 1973). Those researchers were not totally satisfied with the decision-theory approach, however, and Gorry in particular was impressed by simultaneous work that was underway at M.I.T.'s Project MAC (now the Laboratory for Computer Science and the Artificial Intelligence Laboratory).*

From *Methods of Information in Medicine*, 12: 45–51 (1973). Used with permission.

We present here Gorry's insightful paper, which resulted from those dissatisfactions and from his observations about the potential utility of AI techniques. The paper discusses his group's experience with formal decision-theory models and their limitations. Gorry outlines briefly the motivation for the group's new work based on AI techniques, summarizes their early results, and outlines their plans for pursuing the research in the future. The Present Illness Program (Chapter 6) was the first result of this early work. Although the plan for future research was not completely clear at the time the article appeared in 1973, the issues outlined and the recognition that artificial intelligence techniques offered some potential solutions to the problems of representation and the human interface make this article an important early "bridge piece" between work using the traditional normative models and the newer approaches that are the subject of the rest of this book.

2.1 Motivation for the Research

In the past few years, there have appeared in the literature many discussions of the use of computers in the health care system and of the way in which they might improve the efficiency of that system. Such improvements are seen as arising from a wide variety of computer-based activities, such as scheduling of hospital admissions, control of laboratories, and maintenance of medical records. Although these activities (and others as well) can undoubtedly benefit from the introduction of well-designed computer systems, more fundamental problems remain. There is an increasing shortage of medical personnel and a geographical maldistribution because new doctors are reluctant to practice in rural or depressed urban communities. Also these discussions fail to indicate how a high level of physician competence can be maintained in the face of a continued expansion of medical knowledge. The gap between what a doctor should know and what can be retained and utilized is continually widening.

As Schwartz (1970) has noted, "The computer thus remains (in the light of conventional projections) as an adjunct to the present [health care] system, serving a palliative function, but not really solving the major problems of that system."

There is, in fact, little reason to believe that any of the current proposals for solving these problems, technological or other, will do more than mitigate their severity. Despite plans to reorganize patterns of medical care and efforts to enlarge medical school capacity and create new classes of "doctors' assistants," the physician shortage promises to be with us for decades and to pose a serious obstacle to health planning. The problem of maintaining and improving quality appears equally knotty since there is

little indication that current programs in postgraduate education will be adequate to the challenge.

If conventional remedies will not meet the demands imposed by society's broad commitment to extensions of health care, it is clear that new, even heretical strategies must be devised. One intriguing possibility is to use the computer as an "intellectual" or "deductive" instrument—a consultant that is built into the very structure of the health care system and augments or replaces many of the traditional activities of the physician. One can envision an ongoing dialogue between the physician and the computer with the latter continuously taking note of history, physical findings, laboratory data, and the like, alerting the physician to probable diagnoses, and suggesting possible courses of action. One may hope that the computer, well-equipped to store a large volume of information and ingeniously programmed to assist in decision making, will help free the physician to concentrate on the application of bedside skills, the management of the emotional aspects of disease, and the exercise of good judgment in the nonquantifiable aspects of clinical care.

The computer, used in this manner, might also open the way to quite different means of employing nonphysician personnel. Use of the computer as an intellectual resource in diagnosis and treatment might well be coupled to the development of new types of highly specialized allied health personnel who could perform functions of a scope well beyond that currently considered feasible for doctors' assistants. Computer-supported "health care specialists," aided by a variety of automated devices for history taking, blood analysis, and other procedures, and trained to perform a careful physical examination, might take over a large segment of the responsibility for the delivery of primary medical care. Guided by the computer, constrained from exceeding their capacities by instructions built into the computer programs, and linked to regional consulting centers by appropriate display devices, the new breed of "health care specialists" could make a major contribution to the resolution of the seemingly insoluble problem of maldistribution and shortage of physicians.

While such visions of the future are heady stuff, a serious consideration of the problems to be solved is immediately sobering. Clearly, considerable intellectual and technological resources must be marshaled and a long-term research commitment must be made if such a scenario is to become a reality.

The work discussed in the next section constitutes a very modest investigation of one aspect of this problem. The focus of this work is on the decision-making aspects of clinical medicine. The original hope was to embody in a computer program a normative procedure for diagnostic and therapeutic decision making that could be applied to a variety of clinical problems (Gorry, 1968). Although this work was only a partial success, it proved a very valuable exercise from which a number of new ideas were

gained. A discussion of these ideas will be postponed until the discussion of the new research plan. The discussion in the next section has not been "edited" to reflect the new (and hopefully better) view of the problem.

2.2 Review of Past Research

2.2.1 Introduction

The purpose of this section is to review our own research on the use of a computer to solve diagnostic and treatment problems in medicine. A major result of this research has been the development of a computer program that is intended to serve as a consultant in a number of medical problem areas. Here the considerations that underlie the program are discussed. The basic functions of the program are outlined in a nontechnical way, and an example of the use of the program is given. Then the results of the use of the program for several different medical problems are reviewed. Finally, an attempt is made to ascertain the potential of programs such as this in the delivery of appropriate medical care. Detailed reports on various aspects of this research are available in the literature (Gorry, 1967; 1968; Gorry and Barnett, 1968a; 1968b), and so the emphasis here will be on providing a general overview of the work and results obtained to date.

2.2.2 Modeling the Diagnostic and Treatment Problem

The use of digital computers in the selection of good diagnostic and treatment strategies has received increased attention in recent years. One reason for this interest is the general desire to improve the ability of the clinician to deal with the difficult problems that can arise in the management of a patient. A significant portion of the difficulty stems from the fact that the physician must sort out numerous possibilities and develop hypotheses about the state of health of the patient. The ability of the computer to store extremely large amounts of data, to enumerate many possibilities, and to perform complex logical operations suggests its potential value in this problem-solving process. Before a computer can be used to significant advantage in analyzing diagnostic and treatment strategies, however, precise procedures must be formulated for the means of inference required to deduce the clinical state of the patient from observed signs and symptoms, and a formalized capability must be developed for the prediction and assessment of possible therapeutic measures. In other words, the prob-

lem of performing diagnostic inference and weighing therapeutic strategies must be reduced to a problem of computation.

In order to better understand the requirements, a model of the diagnostic-treatment problem was formulated. The model is a mathematical one, but its principal characteristics can be discussed in terms of the way a physician deals with this problem, although it should be noted that the model was not developed as a description of the way in which physicians operate. The purpose of the model is to permit the exploitation of the particular capabilities of a computer. Hence, in the next several paragraphs, when I am discussing the way in which a physician or doctor deals with the problem, I am using *physician* or *doctor* instead of *model* for convenience, and I am not presenting a theory of human problem solving in the medical area. [The relationship of the model to the actual problem-solving behavior of the physicians is discussed in Gorry (1970).]

In general, a physician confronted with a potentially ill patient initially does not have sufficient information about the patient to decide on a diagnosis or on a therapeutic policy. The information the physician does have, however, in addition to his or her general medical knowledge and experience, enables formulation of some tentative hypotheses about the state of health of the patient. This opinion will exert a considerable effect on the strategy the physician will employ in dealing with the patient. For convenience, let us say that the options available to the physician are tests and treatments. By *test* we mean any means for obtaining additional information about the patient ranging from simple questions to laboratory procedures or certain surgical procedures. The physician employs those tests that are expected to provide results of significant value in improving the current view of the patient's problem. The term *treatment* will be used to refer to any means at the doctor's disposal to correct the health state of any patient. Treatments range from drugs to a variety of surgical procedures. The selection of an appropriate treatment for a given problem is strongly dependent on the correctness of the doctor's opinion about the patient's problem. The selection of the wrong treatment, for whatever reason, can have very serious consequences for the patient.

The value of the information obtained from a test is determined by the contribution this information makes to improving the doctor's current view of the patient's problem and hence to reducing the risk of misdiagnosis with its associated cost. Hence the doctor is inclined to perform many tests. On the other hand, the tests available generally are not without some cost in terms of patient discomfort, time of skilled persons, money, etc. Thus there is a conflicting tendency to hold the number of diagnostic tests to a minimum.

As is discussed in Gorry and Barnett (1968b), the doctor resolves these conflicting tendencies by performing sequential diagnosis. At a particular point in time, given the current view of the patient's problem, the physician can evaluate the choices available. The basic choice is to employ a test to

obtain more information or to select a treatment in the hopes of curing the patient.

If the physician elects to cease testing and to make a diagnosis, the choice of a treatment implies a certain risk of mistreatment through a misdiagnosis. On the other hand, the doctor can perform some test in the hopes of gaining additional information on which to base a diagnosis and the resulting choice of treatment. In this case, the doctor incurs the cost (in some terms) of the test selected. When the results of the test are known, and when they have been incorporated into the current view of the problem, the physician is faced with a decision problem of exactly the same form as the one just solved. Thus a doctor can be thought of as solving a sequence of similar decision problems. At each stage of the process, the cost of further testing is balanced against the expected reduction in the cost of treatment due to the test results. When, in the opinion of the physician, no test possesses the property that is expected to reduce the risk of treatment by an amount that exceeds its cost, the physician will cease testing, make a diagnosis, and treat the patient. If the physician repeatedly updates the current view of the problem in keeping with the latest information available, and if the physician has sufficient knowledge, effective diagnostic and therapeutic strategies may be developed.

Although this description of the manner in which a physician deals with diagnosis-treatment problems is simplified and somewhat artificial, it does emphasize the fundamental role that sequential decision making plays in the process. It seemed clear that it was necessary for a computer program to exploit an analogous capability (framed in terms suitable for a machine) in solving more general problems of the type.

2.2.3 The Development of the Computer Program

In this section, the basic components of a computer program to assess diagnostic and therapeutic strategies are discussed. These components directly reflect the view of the required problem-solving process outlined in the preceding section. The discussion of the program is nontechnical. Readers interested in the technical details are referred to Gorry (1967; 1968).

The program has three basic components. The first is called the *information structure,* and it constitutes the medical experience of the program. By changing the information structure, one can convert the program for use in a new problem area. This is the only part of the program that changes from one application to the next.

In addition to the diseases, signs, symptoms, tests, and treatments, the information structure contains two types of information: probabilities and utilities. The probabilities relate signs and symptoms to diseases. For example, one probability might be the conditional probability of red blood cell casts in the urine given that the patient has acute tubular necrosis. The

program's understanding of various diseases is entirely in terms of the conditional probabilities that relate to the variety of signs and symptoms and treatment consequences to those diseases.

The utilities of the tests, treatments, and treatment consequences are thought of as the subjective preferences of an expert. The utility of a test reflects the pain associated with the test, the cost of the test, the time of a skilled person required for the test, the risk of the test to the patient, etc. Similar factors are reflected in the utilities of the treatments and the treatment consequences. Utility can be thought of as the common denominator in terms of which all these diverse factors are measured. Utility assessment will be considered in more detail later. Here we simply note that if the program is to make comparisons of factors such as risk and cost, a common scale must be established for seemingly diverse outcomes.

The second major segment of the program is called the *inference function.* Basically the task of the inference function is to establish the diagnostic significance of a particular test result. In a typical situation, a doctor confronted with a particular diagnostic problem must interpret the available evidence (observed signs and symptoms, etc.) in terms of past personal medical experience. In other words, the doctor employs a method of deduction that can accommodate both a general understanding of diseases and the individual instance represented by the current patient. The inference function of the program is the analogue of this capability in the physician. It uses probabilistic inference based on Bayes' Rule (Gorry, 1967; Gorry and Barnett, 1968a) to obtain a probability distribution for the likelihood of each disease given the evidence to date and general medical experience. The latter is incorporated in the information structure of the program. It is this probability distribution, then, that constitutes the current view taken by the program of the given problem. This view is updated whenever any new evidence is made available to the program. The updated probability distribution is one of the major factors that influence the strategy chosen by the program for dealing with a given patient.

The third component of the program is called the *test/treatment selection function.* Its purpose is to select at each stage in the problem-solving process an appropriate test or treatment for use on the patient. By considering the probability distribution associated with the current view of the problem and the utilities of the various treatment consequences, this function can determine the best treatment to perform, assuming that no further tests are to be used. The treatment chosen is the one that minimizes the expected risk, and it provides the standard used in evaluating the potential value of further testing.

In evaluating the potential usefulness of a particular test, the program considers the current view, the utilities of the various tests, and the likelihood of the possible test results. For each possible result of a test, the program can simulate the change in the current distribution that would occur if this result were obtained. The expected risk of treatment can be estimated for this new distribution. For each result of a test, the expected

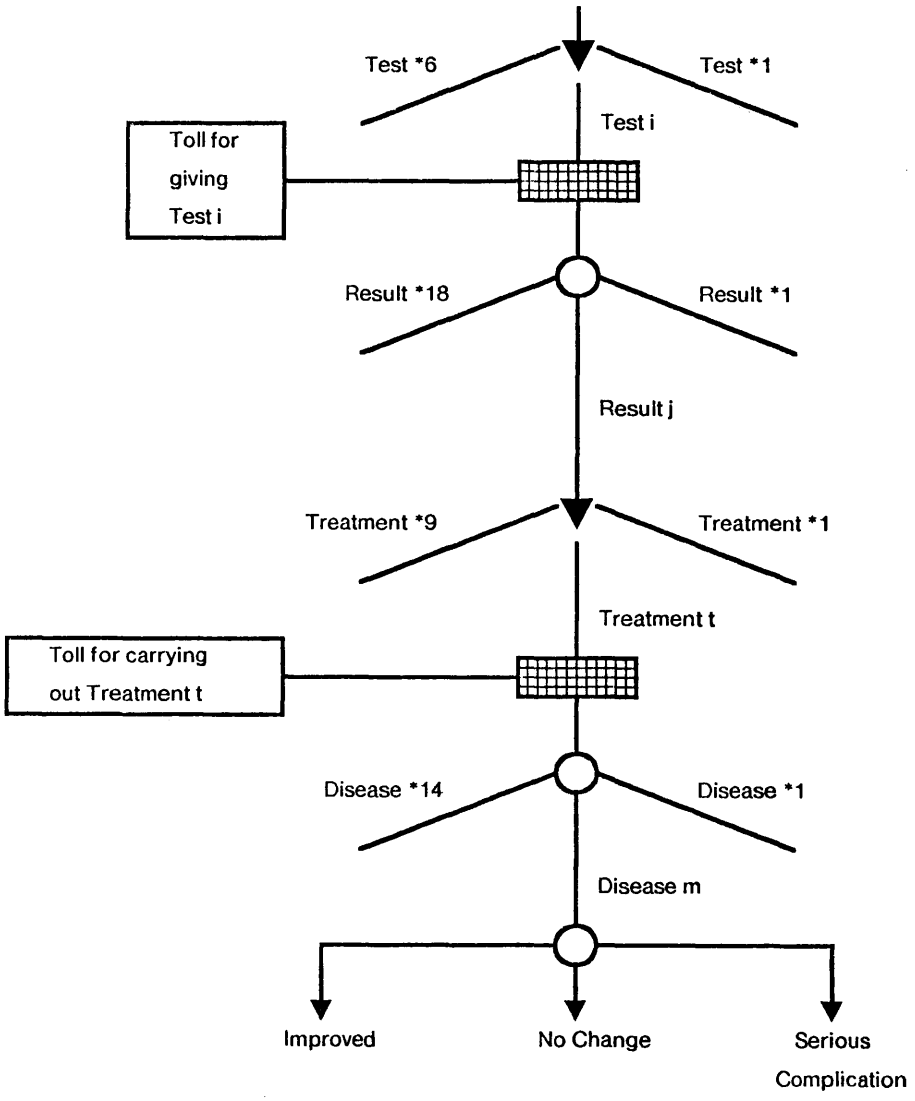


FIGURE 2-1 Example of a decision tree.

risk of treatment given the result is weighted by the likelihood of obtaining that result, and the sum of these products is added to the utility of the test to obtain the overall measure. A schematic representation of the factors considered in evaluating a test is presented in Figure 2-1. By analyzing decision trees such as the one shown, the program attempts to select the best test or treatment at each stage of the analysis.

In Figure 2-2, an actual dialogue between a user and the program is presented.[1] The problem being considered is the diagnosis of a case of congenital heart disease. At the outset of the discussion, the program is essentially passive, simply accumulating whatever evidence the user offers and using the inference function to update its current view of the problem. When the user has completed the initial description of the patient, the test/treatment function is invoked to determine the best diagnosis-treatment policy. In this case, no treatments were considered, and the problem was merely one of diagnosis. The example, however, does give a basic impression of the use of the program.

2.2.4 Experience with the Program

The program has performed extremely well in the medical problems to which it has been applied. It was used as a diagnostic program (no treatments were considered) in two areas: primary bone tumors and congenital heart disease. The bone tumor problem involved the classification of patients into one of nine disease groups based on radiographic evidence and other physical findings. Studies made with the program indicated that it performed at the level of an experienced radiologist given the same evidence (Gorry, 1967). The second medical problem considered was the diagnosis of congenital heart disease. Here there are 35 diseases in question. Again the results showed that the program performed at the level of an expert (Gorry, 1967; 1968). Both the bone tumor problem and the congenital heart disease problem had certain characteristics that made them unsatisfactory tests of the program's capability, however. First, in neither area were the costs of the tests sufficiently high to make the sequential aspect of the program particularly important. Because tests were so cheap, it mattered little how many were performed or in what order. Second, the treatment problem was not considered because the doctors needed to help define these treatments and their consequences were not available.

Because of these difficulties, a third medical area was considered, the diagnosis and treatment of acute renal failure. The management of the acute renal failure syndrome is an important medical problem. Although the incidence of the problem is relatively small in the context of all disease treatment problems, the potential risk to the patient's life is sufficiently great to give the problem special significance. In dealing with this problem, the physician must account for this risk factor as well as the possibility that the tests that could be employed to gain further information about the patient can contribute to the medical problem if improperly used.

The definition of the acute renal failure problem used in this study included 15 diseases. As in the previous problems of bone tumors and

[1]The ELIZA program, developed by Professor Joseph Weizenbaum of M.I.T., was used to facilitate user input.

PLEASE TELL ME ABOUT THE PATIENT.
Well he is about 65 years old.
PLEASE CONTINUE.
No cyanosis.
O.K. PLEASE GO ON.
What is your opinion of the 3 most probable diseases now.
2 WITH PROBABILITY .68
18 WITH PROBABILITY .10
24 WITH PROBABILITY .04
ALL RIGHT. GO ON PLEASE.
Name four diseases for me.
ATRIAL SEPTAL DEFECT WITH PULMONARY HYPERTENSION.
PLEASE CONTINUE.
I meant the name of disease 24.
VENTRICULAR SEPTAL DEFECT WITH PULMONARY FLOW LE. 1.4 SYSTEMIC FLOW.
O.K. PLEASE GO ON.
What more do you need to know.
CHECK THE HILAR VESSELS. ARE THEY INCREASED OR DECREASED.
That information is not available.
LOOK FOR A LEFT SECOND SYSTOLIC MURMUR.
What is your current view.
MY CURRENT VIEW IS

DISEASE	PROB	DISEASE	PROB
2	.68	34	.03
18	.10	5	.03
24	.04	12	.02
25	.04		

NOW LET'S GO ON.
LOOK FOR A LEFT SECOND SYSTOLIC MURMUR.
Say there is a left second systolic murmur.
I SEE.
IS THE AXIS IN THE ECG RIGHT, LEFT, OR NORMAL.
What is the probability of atrial septal defect now.
IT IS 85.
ALL RIGHT. GO ON PLEASE.
IS THE AXIS IN THE ECG RIGHT, LEFT, OR NORMAL.
Left.
ALL RIGHT.
CHECK FOR A LEFT FOURTH SYSTOLIC MURMUR.
What are the 3 most probable diseases now.
5 WITH PROBABILITY .43
2 WITH PROBABILITY .39
18 WITH PROBABILITY .05
PLEASE CONTINUE.
CHECK FOR A LEFT FOURTH SYSTOLIC MURMUR.
What is the medical name of disease 5.
ATRIO-VENTRICULAR COMMUNIS.
O.K. PLEASE GO ON.
CHECK FOR A LEFT FOURTH SYSTOLIC MURMUR.
Say there is one.
O.K.
IS THERE AN ACCENTUATED OR DIMINISHED P2.
What is the probability of disease 5 now.
IT IS 71.
ALL RIGHT. GO ON PLEASE.

FIGURE 2-2 Example of a session with the program. (Program's responses and questions are in capital letters.)

```
IS THERE AN ACCENTUATED OR DIMINISHED P2.
Not available.
I RECOMMEND A DIAGNOSIS OF DISEASE 5.
What are the 3 most probable diseases.
        5 WITH PROBABILITY .71
        2 WITH PROBABILITY .23
       24 WITH PROBABILITY .01
I'm finished. Thanks for your help.
I HOPE THAT I HAVE BEEN OF SERVICE. GOODBYE.
```

FIGURE 2-2 continued

congenital heart disease, the information structure for the program included the relevant probabilities. In this case, however, no attempt was made to obtain these probabilities from an analysis of historical data. Rather, the opinion of an expert was used in establishing each probability. In addition, special attention was paid to the assessment of the required utilities. Again, the opinion of an expert renal specialist served as the basis for these numbers.

The precise manner in which these judgments were obtained from the expert and the way in which they were converted to utilities are discussed in Betaque and Gorry (1971). Here I want to outline the procedure briefly. The renal expert was given a series of hypothetical decision problems. Each problem required that a choice be made between a particular event for certain (such as curing the patient by performing a certain operation) and accepting a chance in a *lottery*. If the expert chose the lottery, a given event would be chosen for him with probability P, and some other event would be chosen with probability $1 - P$. Before making a choice, the expert is told exactly what the two events in the lottery are and what the value of P is. With the theory discussed in Betaque and Gorry (1971), a series of these decision problems can be used to establish the utilities of tests, treatments, and consequences required by the program.

With the information structure for the renal failure problem developed in this way, the program duplicated the diagnostic-treatment decisions of expert renal specialists in over 90% of the cases tested. Furthermore, when the information structures from two experts were used, the program agreed more closely with the expert whose judgments it was using than did the other expert.

2.3 Plan for Further Research

To provide a context for a discussion of our plan for further research in this area, I want to offer a criticism of the work to date. Without going into detail, let me say that the evaluations of the program were strongly biased in favor of the program. The number of diseases, their rigid defi-

nitions, and the types of tests and treatments used all combined to make simple, exhaustive search an effective strategy. Thus the program did quite well compared to the experts, but the method it employed differed from the ones they used. Although we cannot characterize precisely the methods used by the experts, it is clear that these methods can accommodate the greater complexity of real clinical situations. The potential usefulness of exhaustive search as the primary decision procedure for the program, however, is open to question. In this regard, it is instructive to consider some of the failures of the program in the experiments described above.

One such case was a patient with acute glomerulonephritis (AGN), a common cause of acute renal failure. Patients with AGN seldom have severe hypertension, but the patient presented to the physicians and the program did. The program obtained the correct diagnosis, but the treatment it recommended differed from that proposed by the doctors. Although both the physicians and the program chose the same treatment for AGN, the physicians recognized the need to deal with the patient's hypertension and hence recommended a second treatment as well.

Clearly, the program could be modified to check for this problem and to make the appropriate decisions. The same could be done for several other problems of this type that were identified. Similar modifications would be required to obtain the appropriate interpretation of certain signs and symptoms. For example, hematuria (red blood cells in the urine) is an important diagnostic finding in acute renal failure. On the other hand, a patient with an indwelling catheter will generally have hematuria regardless of his or her intrinsic disease. Hence the interpretation of this finding should reflect this fact. Again, either the program or the data it uses must be changed. Although these particular problems could easily be solved within the context of the existing problem, they raise an important question. How many such "minor" modifications will be required for the program to have practical use in the clinical management of acute renal failure?

For a period of several months, we have investigated the amount and type of knowledge possessed by two acknowledged renal experts. Although much more work needs to be done, I can offer certain tentative conclusions. These conclusions provide motivation for a change of direction in this research.

1. Although detailed knowledge of physiology and pathophysiology is sometimes useful in clinical decision making, gross knowledge of this kind coupled with a large number of experiential facts and mini–decision procedures forms the primary basis of clinical judgment in renal disease.
2. The knowledge used by the experts is both factual and procedural. Their experience has provided them with a rich repertoire of ideas of the form "if x is present and y is absent, then a good trial hypothesis is

D." Such rules allow them to focus their attention on relatively few diagnoses or treatments. Of course, these rules are heuristics, but many of them are of considerable value in dealing with experts' decision-making problems. By remembering large numbers of such patterns or rules, they avoid search to a large extent.

3. This experiential knowledge is not framed in deterministic terms, but is associated with various degrees of certainty.
4. The renal experts can specify only part of this knowledge *a priori*. A large part of this knowledge can be elicited only in response to apparent misconceptions on my part (or as embodied in the program).
5. Although there are very many "pieces" of knowledge involved, these experts seem able to state them clearly when the occasion arises.

The physicians with whom I have been working are acknowledged experts in renal disease, and their performance in this field far surpasses that of a very large fraction of the doctors who treat patients with this problem.[2] It is important, then, to get as much of their knowledge as possible in distributable form (i.e., a program).

The original program was based on a particular normative view of clinical decision making. The judgments of experts could be added only to the extent that these judgments could be expressed as simple probabilistic relationships or as utilities. Procedural knowledge was added through reprogramming. Thus the addition of knowledge was either implicit (setting probabilities or utilities to cause the program to arrive at a conclusion that a physician could obtain more directly) or laborious (reprogramming). Unfortunately, I am convinced that, for the foreseeable future, the desire to add knowledge will be great, and an attempt to maintain the program (perhaps for its simple, aesthetic appeal) will prove frustrating at best.

Although this discussion has been brief, it indicates the general tenor of the problems I foresee with the approach we had been using. Decision analysis is a useful tool when the problem has been reduced to a small, well-defined one of action selection. It cannot be the sole basis of a program to assist clinicians generally in an area such as renal disease.

2.3.1 A New Program for Renal Disease

Several months ago, we began the development of a prototype program for use in the problem of acute renal disease. This program is currently in a most rudimentary form. Therefore I will be discussing here not so much an existing program as some goals toward which we are working. Our short-term goal is to produce a version of this prototype that can be

[2]This is not a condemnation of the latter group. It is a simply a reflection of the fact that most people with kidney disease do not have access to the experts and resources of a major teaching hospital.

used by renal specialists in an informal way as a means to assess the potential of the ideas on which it is based.

Recent developments by people in the Artificial Intelligence Laboratory at M.I.T. have opened the way for the exploration of new approaches to computer assimilation of knowledge. The developments comprise both a way of looking at the problem of machine knowledge and some very high-level programming systems (Sussman et al., 1971; Winograd, 1971). The prototype system incorporates some of these new ideas and as a result is better able to accept experiential knowledge directly from the user. The details of the new program are beyond the scope of this paper (and may change significantly over time). Here, I will restrict myself to the conceptual framework within which this program is being built.

A simple language has been implemented to permit renal experts to give advice to the program regarding facts or ways to proceed in a particular circumstance. Examples of such statements are the following:

a. In acute glomerulonephritis, if hematuria is gross then red blood cell casts are very likely;

b. If proteinuria is heavy and hematuria is gross and red blood cell casts are present and diagnosis is acute renal failure, then diagnosis of glomerulonephritis is very likely.

The basic functions of the program are (1) to accept such statements, (2) to note appropriate associations among various statements, and (3) to use the statements deductively when appropriate to draw conclusions about diagnosis or management.

It must be emphasized that the new program is very primitive as yet. The new technology mentioned above has greatly facilitated its development, however, and it seems likely that a much improved program can be implemented. The real question is whether sufficient improvement can be realized to make the program useful. At present, we cannot answer this question, but I can indicate the chief problem areas to be explored.

2.3.2 Problems for Investigation

Concept Identification

We intend to continue to try to identify the important concepts in renal disease. By this, I mean the identification of the central, problem-specific ideas in terms of which the experts organize their knowledge. One example is the concept of renal function. There are several approaches to inferring renal function and assessing whether it is stable or changing. This determination is very important in diagnosis and in choosing management strategies. It is possible to obtain from the experts the procedure by which they

infer a value for renal function. Further, many statements about the interpretation of changes in renal function can be made. To capture the knowledge embodied in these statements, some computer realization of the concept of renal function must be developed.

Already it is clear that there are many such concepts. We will be trying to identify the most important ones and to develop reasonable ways to represent them in the program. Needless to say, a major question will be how many such concepts are required in the program and the complexity of their realization. One possibility is that the number is so large as to be impossible to deal with at present. Another is that the individual concepts are based on an implicit assumption of enormous knowledge about the world. We believe that the number of important concepts is indeed large, but not beyond our capabilities. For example, a very large portion of the basic knowledge about kidney disease is contained in one book (admittedly a large one). Further, the expert clinicians believe that big chunks of that book are unnecessary for the support of *clinical* activities.

The issue of how much common sense is assumed in these concepts is also important. On the one hand, it could be argued that to understand these concepts a program must understand a tremendous amount about the world. On the other hand, the relatively precise language of medicine may be the key here. The program may know many facts about streptococcal infection and its role in acute renal failure without understanding the concept of germs. The physician using the program may have little need to ask the program for the latter. More generally, the user will have considerable knowledge organized in terms of fairly well-defined words and phrases. The knowledge of the program can be expressed in these terms to assist the physician. More detailed knowledge on the part of the program may be unnecessary.

Already it is clear that there are many concepts, but that not all are of great importance. We will be trying to identify the most important ones and to develop reasonable ways to represent them in a program.

Language Development

Because we believe that the continual addition of knowledge is critical, we will be working on the development of a language within which experts can express this knowledge to the program. An understanding of the important concepts in renal disease, of course, is a prerequisite for the design of such a language. In general terms, what we are seeking is an automatic programming capability so experts can *program* the machine directly. At present, we can envision three languages involved in this process.

First, at the lowest level there will be the computer language in which the concepts are realized. At a higher level will be a language in which statements concerning these concepts are made without explicit recogni-

tion of the details of the lower-level realization. Such a language may well be an extension of the simple IF/THEN-type language already implemented. Maintaining this separation may lessen the problems arising from changes in the particular realization of the concepts in the machine. The third-level language will be English. We are hoping to use Winograd's program (Winograd, 1971) to translate statements made by the experts (in a subset of English) into the intermediate language mentioned. The second-level language can be viewed as a canonical representation of the subset of English that can be accepted. Such a translation will require an interaction with both of the lower-level languages, but we can say little in detail about this process. We do believe, however, that, whatever the realization, language will be critical if the knowledge of experts is to be captured. Also, we believe that they must be given some form of English for input and inquiry. Hence the tasks of concept identification and language development will have highest priority.

One question is worth raising here, although at present we do not know the answer. This question concerns the necessity for English. With experts dedicated to the project being the sole source of knowledge input, there might be little need for English; they could be taught to use the second-level language. On the other hand, if interaction with other clinicians proves to be important (and we believe it will) then English may be very important. The question of how much is to be gained from English is one that will be considered carefully.

Explanation

The other side of the coin is explanation. If experts are to use and improve the program directly, then it must be able to explain the reasons for its actions. Furthermore, this explanation must be in terms the physicians can understand. The steps in a deduction and the facts employed must be identified for the expert so that he or she can correct one or more of them if necessary. As a corollary, the user must be able to easily find out what the program knows about a particular subject.

2.3.3 A Comment on Goals

The original aim of this research was to produce a decision-making program. Although this is still the long-term goal, we believe the time required to achieve this goal is sufficiently long to necessitate the establishment of some short-term goals. Presently, we consider a reasonable (but somewhat vague) goal to be the construction of a program that can accept knowledge and answer simple requests for parts of that knowledge. Because there will be many cases where the program will lack knowledge relevant to a par-

ticular clinical situation, it should make not pronouncements but rather suggestions of things to consider and the assumptions on which its suggestions are based.

ACKNOWLEDGMENTS

My colleagues, Dr. William B. Schwartz and Dr. Jerome P. Kassirer of the Tufts–New England Medical Center have made major contributions to the work discussed. Any inadequacies in the discussion, however, are my responsibility alone.

3

Knowledge Engineering for Medical Decision Making: A Review of Computer-Based Clinical Decision Aids

Edward H. Shortliffe, Bruce G. Buchanan, and Edward A. Feigenbaum

We now jump ahead to 1979 when Shortliffe, Buchanan, and Feigenbaum published a review article that more broadly surveys the field of computer-based medical decision making. Like Gorry's paper, this article focuses on the limitations of early work that had made artificial intelligence techniques and knowledge-engineering research particularly attractive. However, the coverage of other models is more detailed and comprehensive, and the discussion of AI benefits from another five years of work to which the authors were able to refer. We include this article early in this volume to help set the scene for the discussions of AI systems that follow. Many of the systems subsequently described in detail are referenced here in describing the evolution of computer-based approaches to medical advice giving.

The article reviews representative examples from each of several major medical decision-making paradigms: (1) clinical algorithms, (2) clinical data banks that include analytic functions, (3) mathematical models of physical processes, (4) pattern recognition, (5) Bayesian statistics, (6) decision analysis, and (7) the symbolic reasoning approaches of AI. Because the topic is too broad to provide exhaustive discussions of the techniques and systems in each category, the approach used here is to undertake case studies as a basis for analyzing general strengths and limitations. It should

 From *Proceedings of the IEEE*, 67: 1207–1224 (1979).

be noted that the authors do not claim that any one method is best for all applications and they stress that considerable basic research in medical computing remains to be done. They also suggest that powerful new approaches may lie in the melding of two or more established techniques, a trend that is already characterizing some of the AIM work of the 1980s.

3.1 Introduction

As early as the 1950s, physicians and computer scientists recognized that computers could assist with clinical decision making (Lipkin and Hardy, 1958) and began to analyze medical diagnosis with a view to the potential role of automated decision aids in that domain (Ledley and Lusted, 1959). Since that time a variety of techniques have been applied, accounting for at least 800 references in the clinical and computing literature (Wagner et al., 1978). In this article we review several decision-making paradigms and discuss some issues that account for both the multiplicity of approaches and the limited clinical success of most of the systems developed to date. Because other authors have reviewed computer-aided diagnosis (Jacquez, 1972; Schoolman and Bernstein, 1978; Wardle and Wardle, 1978) and the potential impact of computers in medical care (Schwartz, 1970), our emphasis here will be somewhat different. We will focus on the representation and use of knowledge, termed *knowledge engineering*, and the inadequacies of data-intensive techniques, which have led to the exploration of novel symbolic reasoning approaches during the last decade.

3.1.1 Reasons for Attempting Computer-Aided Medical Decision Making

Because of the accelerated growth in medical knowledge, physicians have tended to specialize and to become more dependent on assistance from other experts when presented with a complex problem outside their own area of expertise. The primary care physician who first sees the patient has thousands of tests available with a wide range of costs (both fiscal and physical) and potential benefits (i.e., arrival at a correct diagnosis or optimal therapeutic management). Even the experts in a specialized field may reach very different decisions regarding the management of a specific case (Yu et al., 1979a). Diagnoses that are made, on which therapeutic decisions are based, have been shown to vary widely in their accuracy (Garland, 1959; Prutting, 1967; Rosenblatt et al., 1973). Furthermore, medical students usually learn about decision making in an unstructured way, largely through observing and emulating the thought processes they perceive to be used by their clinical mentors (Kassirer and Gorry, 1978).

Thus the motivations for attempts to understand and automate the process of clinical decision making have been numerous (Wardle and Wardle, 1978). They are directed both at diagnostic models *and* at assisting with patient-management decisions. Among the reasons for introducing computers into such work are the following:

1. to improve the *accuracy* of clinical diagnosis through approaches that are systematic, complete, and able to integrate data from diverse sources;
2. to improve the *reliability* of clinical decisions by avoiding unwarranted influences of similar but not identical cases (a common source of bias among physicians), and by making the criteria for decisions explicit and hence reproducible;
3. to improve the *cost efficiency of tests and therapies* by balancing the expenses of time, inconvenience, or funds against benefits and risks of definitive actions;
4. to improve our *understanding of the structure of medical knowledge,* with the associated development of techniques for identifying inconsistencies and inadequacies in that knowledge; and
5. to improve our *understanding of clinical decision making,* in order to improve medical teaching and to make computer programs more effective and easier to understand.

3.1.2 The Distinction Between Data and Knowledge

The models on which computer systems base their clinical advice range from data-intensive to knowledge-intensive approaches. There are at least four types of knowledge that may be distinguished from pure statistical data:

1. knowledge derived from data analysis (largely numerical);
2. judgmental or subjective knowledge;
3. scientific or theoretical knowledge;
4. high-level strategic knowledge or "self-knowledge."

If there is a chronology to the field over the last 20 years, it is that there has been progressively less dependence on "pure" observational data and more emphasis on higher-level symbolic knowledge inferred from primary data. We include with domain knowledge a category of judgmental knowledge that reflects the experience and opinions of an expert regarding an issue about which the formal data may be fragmentary or nonexistent. Since many decisions made in clinical medicine depend on this kind of judgmental expertise, it is not surprising that investigators should begin to

look for ways to capture and use the knowledge of experts in decision-making programs. Another reason to move away from purely data-intensive programs is that in medicine the primary data available to decision makers are far from objective (Feinstein, 1970; Komaroff, 1979). They include subjective reports from patients and error-prone observations (Gill et al., 1973). Also, the terminology used in the reports is not standardized (Croft, 1972), and the classifications often overlap. Thus decision-making aids must be knowledgeable about the unreliability of the data as well as the uncertainty of the inference.

For example, data-intensive programs include medical record systems that accumulate large data banks to assist with decision making. There is little knowledge *per se* in the data bank, but there *are* large amounts of data that can help with decisions and be analyzed to provide new knowledge. A program that retrieves a patient's record for review or even one that retrieves the records of several patients (matching some set of descriptors) is performing a data-management task with little reasoning involved (Greenes et al., 1970; Rodnick and Wiederhold, 1977). Although there is statistical "knowledge" contained in the conditional probabilities generated from such a data bank and utilized for Bayesian analysis, it is all numeric. At the other extreme are systems that encode and use the kind of expert knowledge that cannot be easily gleaned from data banks or literature review (as described in subsequent chapters in this volume). Systems that model human reasoning or emphasize the education of users tend to fall toward this end of the data-knowledge continuum.

In addition to judgmental and statistical knowledge, there are other forms of information that can play an important role in computer-based clinical decision aids. For example, underlying scientific theories and relationships are often ignored by diagnostic programs but provide the foundation for decisions made by human experts. Consider, for example, the potential utility of techniques that could effectively represent and use the basic knowledge of biochemistry, biophysics, or detailed human physiology. Biomedical modeling research offers some mathematical techniques for encoding such knowledge in certain domains, but symbolic approaches and clinically useful applications are still largely unrealized.

Finally, there is another kind of knowledge used by human decision makers—an understanding of reasoning processes and strategies themselves. This kind of high-level or meta-level knowledge, if incorporated into computer programs, may not only heighten their decision-making performance but also augment their acceptability to users by making them appear to be more aware of their own power, strategies, and limitations.

We use the term *knowledge engineering*, then, to refer to computer-based symbolic reasoning issues such as knowledge representation, acquisition, explanation, and "self-awareness" or self-modification (Feigenbaum, 1977). It is along these dimensions that knowledge-based programs differ most sharply from conventional calculations. For example, such programs can solve problems by pursuing a line of reasoning; the individual inference

steps and the whole chain of reasoning may also form the basis for explanations of decisions. A major concern in knowledge engineering is clear separation of the medical knowledge in a program from the inference mechanism that applies that knowledge to the data of individual cases. One goal of this chapter is to identify the strengths and weaknesses of earlier work, those issues that have motivated several current researchers to investigate the automation of clinical decision aids through knowledge engineering.

3.1.3 Parameters for Assessing Work in the Field

Barriers to successful implementation of computer-based diagnostic systems have been analyzed on several occasions (Croft, 1972; Friedman and Gustafson, 1977; Startsman and Robinson, 1972) and need not be reviewed here. However, in assessing programs it is pertinent to examine several parameters that affect the success and scope of a particular system in light of its intended users and application. Unfortunately, the medical computing literature has few descriptions of systems for which all the following issues can be assessed:

1. How accurate is the program?[1]
2. What is the nature of the knowledge in the system, and how is it generated or acquired?
3. How is the clinical knowledge represented, and how does it facilitate the performance goals of the system described?
4. How are knowledge and clinical data used, and how does this impact on system performance?
5. Is the system accepted by the users for whom it is intended? Is the interface with the user adequate? Does the system function outside of a research setting, and is it suitable for dissemination?
6. What are the limitations of the approach?

An issue we have chosen not to address is the cost of a system, including the size of the required computing resource. Not only is information on this question scanty for most of the programs, but expenses generated in a research and development environment do not realistically reflect the costs one expects from a system once it is operating for service use.

[1]Although this is important, it is not the only measure of clinical effectiveness. For example, the effects on morbidity, mortality, and length of hospital stay may also be important parameters. As we shall show, few systems have reached a stage of implementation where these parameters can be assessed. Moreover, because of the complexity of the interacting influences that affect the usual measures of outcome, it may be difficult ever to define the marginal benefit of such systems.

3.1.4 Overview of This Chapter

An exhaustive review of computer-aided diagnosis will not be attempted in light of the vastness of the field, and we have therefore chosen to present the prominent paradigms by discussing representative examples. In separate sections we give an overview, example, and discussion of (1) clinical algorithms, (2) data bank analysis, (3) mathematical models, (4) pattern recognition, (5) Bayesian analysis, (6) decision theory, and (7) symbolic reasoning. We close each section by identifying the range of applications for which the approach appears most appropriate, the limitations of the approach, and the ways in which symbolic reasoning techniques may strengthen the approach by improving its performance or acceptability.

The seven principal examples we have selected are not necessarily the best nor the most successful; however, they illustrate the issues we wish to discuss within the major paradigms. We have also referenced other closely related systems, so the bibliography should guide the reader to more details on particular topics. Any attempt to categorize programs in this way is inherently fraught with problems in that several systems draw upon more than one paradigm. Thus we have occasionally felt obligated to simplify a topic for clarity in light of the overall purposes of this review and the limitations of the space available to us.

Because we are only interested here in decision-making tools for use by clinicians, we have chosen to disregard systems that are designed primarily for use by researchers (Groner et al., 1971; Johnson and Barnett, 1977; Mabry et al., 1977; Rubin and Risley, 1977). Furthermore, we shall not discuss biomedical engineering applications of computers, such as advanced automated instrumentation techniques [e.g., computerized tomography (Kak, 1979)] or signal processing techniques [e.g., programs for EKG analysis (Pipberger et al., 1975) or patient monitoring (Warner, 1968)]. Because they do not explicitly make inferences, we have also omitted programs designed largely for data storage and retrieval that leave the actual analysis and decision making to the clinician (Greenes et al., 1970; Korein et al., 1971; Weed, 1973). We have also chosen to discuss working computer programs rather than unimplemented theories or early reports of work in progress.

3.2 Clinical Algorithms and Automation

3.2.1 Overview

Clinical algorithms, or protocols, are flow charts to which a diagnostician or therapist can refer when deciding how to manage a patient with a specific clinical problem (Sherman et al., 1973). Such protocols usually allow

decisions to be made by carefully following the simple branching logic, although there are built-in safeguards whereby referrals to experts are made if a case is unusually complex. The value of a protocol depends on the infrequency with which such referrals are made, so it is important to design algorithms that reflect an appropriate balance between safety and efficiency. In general, algorithms have been designed by expert physicians for use by paramedical personnel who have been entrusted with the performance of certain routine clinical-care tasks.[2] The methodology has been developed in part because of a desire to define basic medical logic concisely so that detailed training in pathophysiology would not be necessary for ancillary practitioners. Experience has shown that intelligent high school graduates, selected in large part because of poise and warmth of personality, can provide excellent care guided by protocols after only four to eight weeks of training. This care has been shown to be equivalent to that given by physicians for the same limited problems and to be accepted by physicians and patients alike for such diverse clinical situations as diabetes management (Komaroff et al., 1974; McDonald et al., 1975), pharyngitis (Grimm et al., 1975), headache (Greenfield et al., 1976), and other disease categories (Sox et al., 1973; Vickery, 1974).

The role of the computer in such applications has been limited, however. In fact, several groups initially experimented with computer representation of the algorithms but have since abandoned the efforts and resorted to prepared paper forms (Komaroff et al., 1974; Vickery, 1974). In these cases the computer had originally guided the physician assistant's collection of data and had specified precisely what decisions should be made or actions taken, in accordance with the clinical algorithm. However, since the algorithmic logic is generally simple and can often be represented on a single sheet of paper, the advantages of an automated approach over a manual system have not been clearly demonstrated. In one study Vickery (1974) showed that supervising physicians could detect no significant difference between the performance of physicians' assistants using automated versus manual systems, although the computer system entirely eliminated errors in data collection (since it demanded all relevant data at the appropriate time). Furthermore, the computer could not, of course, decide whether the actual observations entered by the physician's assistant were correct; yet this kind of inaccuracy was one of the most common reasons why supervisors occasionally found an assistant's performance unsatisfactory.

There are two other ways in which the computer has been used in the setting of clinical algorithms. First, mathematical techniques have been used to analyze signs and symptoms of diseases and thereby to identify

[2]Clinical algorithms have also been prepared for use by physicians themselves, but Grimm has found that they are generally less well accepted by doctors (Grimm et al., 1975). He showed, however, that physician performance could improve when protocols were used in certain settings.

those that should most appropriately be referenced in corresponding clinical algorithms (Glesser and Collen, 1972; Knapp et al., 1977; Walsh et al., 1975). The process for distilling expert knowledge in the form of a clinical algorithm can be an arduous and imperfect one (Sherman et al., 1973); formal techniques to assist with this task may prove to be very valuable.

Some researchers in this area also use computers to assist with clinical care audit, comparing actual actions taken by a physician's assistant with those recommended by the algorithm itself. Sox et al. (1973) have described a system in which the assistant's checklist for a patient encounter was sent to a central computer and analyzed for evidence of deviation from the accepted protocol. Computer-generated reports then served as feedback to the physician's assistant and to the supervising physicians.

3.2.2 Example

We have selected for discussion a project that differs from those previously cited in that (1) computer techniques are still being utilized, and (2) the clinical algorithms are designed for use by primary care physicians themselves. This is the cancer chemotherapy system developed in Alabama by Mesel et al. (1976). The algorithms were developed in response to a desire to allow private practitioners, at a distance from the regional tertiary-care center, to manage the complex chemotherapy for their cancer patients without routinely referring them to the central oncologists. Mesel et al. have described a "consultant-extender system" that enables the primary physician to treat patients with Hodgkin's disease under the supervision of a regional specialist. Five oncologists developed a care protocol for the treatment of Hodgkin's disease, and this algorithm was placed on-line. Once patients had been entered in the study, their private physicians would prepare "encounter forms" at the time of each office visit. These forms would document pertinent interval history, physical findings, and lab data, as well as the chemotherapy administered. The form would then be sent to the regional center, where it was analyzed by the computer and a customized clinical algorithm was produced to assist the private physician with the management of *that* patient during the next appointment. Thus the computer program would take into account the ways in which the individual patient's disease might progress or improve and would prepare an appropriate clinical algorithm. This protocol was sent back to the physician in time for it to be available at the next office visit. The private practitioner was encouraged to call the regional specialist directly if the protocol seemed in some way inadequate or if additional questions arose. The authors present data suggesting that their system was well accepted by physicians and patients, and that excellent care was delivered.[3] Retrospective review of

[3]This is an interesting result in the light of Grimm's experience mentioned earlier. One possible explanation is that physicians were more accepting of the algorithmic approach in Mesel's case because it allowed them to perform tasks that they would previously not have been able to undertake.

cases that were treated at the referral center, but without the use of the protocols, showed a 16% rate of variance from the management guidelines specified in the algorithms; there was no such variance when the protocols were followed. Thus algorithms may be effective tools for the administration of complex specialized therapy in circumstances such as those described.[4]

3.2.3 Discussion of the Methodology

Although clinical algorithms are among the most widespread and accepted of the decision aids described in this chapter, the simplicity of their logic makes it clear why the technique cannot be effectively applied in most medical domains. Decision points in the algorithms are generally binary (i.e., a given sign or symptom is either present or absent), and there tend to be many circumstances that can arise for which the user is advised to consult the supervising physician (or specialist). Thus the difficult decision tasks are left to experts, and there is generally no formal algorithm for managing the case from that point on. It is precisely the simplicity of the algorithmic logic and the safeguard of the supervising expert that have permitted many algorithms to be represented on one or two sheets of paper and have obviated the need for direct computer use in most of the systems. The contributions of clinical algorithms to the distribution and delivery of health care, to the training of paramedics, and to quality care audit have been impressive and substantial. However, the approach is not suitable for extension to the complex decision tasks to be discussed in the following sections.

3.3 Data Bank Analysis for Prognosis and Therapy Selection

3.3.1 Overview

Automation of medical record keeping and the development of computer-based patient data banks have been major research concerns since the earliest days of medical computing. Most such systems have attempted to avoid direct interaction between the computer and the physician recording the data, with the systems of Weed (1968; 1973) and Greenes et al. (1970) being notable exceptions. Although the earliest systems were designed merely as record-keeping devices, there have been several recent attempts to create programs that could also provide analyses of the information

[4]More recently the Alabama group has reported similar success implementing a consultant-extender system for adjuvant chemotherapy in breast carcinoma (Wirtschafter, 1979).

stored in the computer data bank. Some early systems (Greenes et al., 1970; Karpinski and Bleich, 1971) had retrieval modules that identified all patient records matching a Boolean combination of descriptors; however, further analysis of these records for decision-making purposes was left to the investigator. Weed has not stressed an analytical component in his automated problem-oriented record (Weed, 1973), but others have developed decision aids that use medical record systems fashioned after his (Slamecka et al., 1977).

The systems for data bank analysis all depend on the development of a complete and accurate medical record system. Once such a system is developed, a number of additional capabilities can be provided: (1) correlations among variables can be calculated; (2) prognostic indicators can be measured; and (3) the response to various therapies can be compared. A physician faced with a complex management decision can look to such a system for assistance in identifying patients who had similar clinical problems in the past and can then see how those patients responded to various therapies. A clinical investigator who keeps the records of his study patients on such a system can use the program's statistical capabilities for data analysis. Hence, although these applications are inherently data-intensive, the kinds of "knowledge" generated by specialized retrieval and statistical routines can provide valuable assistance for clinical decision makers. For example, they can help avoid the inherent biases of anecdotal experience, such as those that occur when an individual practitioner bases decisions primarily on personal encounters with one or two patients having a rare disease or complex of symptoms.

There are many excellent programs in this category, one of which is discussed in some detail in the next section. Several others warrant mention, however. The HELP system at the University of Utah (Warner et al., 1972a; 1974; Warner, 1978) utilizes a large data file on patients from the Latter-Day Saints Hospital. Clinical experts formulate specialized "HELP sectors," which are collections of logical rules that define the criteria for a particular medical decision. These sectors are developed by an interactive process; the expert proposes important criteria for a given decision and is provided with actual data regarding each criterion (based on relevant patients and controls from the computer data bank). The criteria in the sector are thus adjusted by the expert until adequate discrimination is made to justify using the sector's logic as a decision tool.[5] The sectors are then used for a variety of tasks throughout the hospital.

Another system of interest is that of Feinstein et al. at Yale (1972), in which physicians interact with the system to request assistance in estimating prognosis and guiding management for patients with lung cancer. Similarly, Rosati et al. (1975) have developed a system at Duke University that

[5]This process might be seen as a tool to assist with the formulation of clinical algorithms as discussed in the previous section. Another approach using data bank analysis for algorithm development has also been described (Glesser and Collen, 1972).

uses a large data bank of patients who have undergone coronary arteriography. New patients can be matched against those in the data bank to help determine patient prognosis under a variety of management alternatives.

3.3.2 Example

One of the most successful projects in this category is the ARAMIS system (Fries, 1972). The approach was designed originally for use in an outpatient rheumatology clinic and then broadened to a general clinical data base system (TOD) (Weyl et al., 1975; Wiederhold et al., 1975) so that it could be transferred to clinics in oncology, metabolic disease, cardiology, endocrinology, and certain pediatric subspecialties. All clinic records are kept in a large tabular format in which a column indicates a specific clinic visit and the rows indicate the relevant clinical parameters that are being followed over time. These charts are maintained by the physicians seeing the patient in a clinic, and the new column of data is later transferred to the computer data bank by a transcriptionist; in this way time-oriented data on all patients are kept current. The defined data base (clinical parameters to be followed) is determined by clinical experts and in the case of rheumatic diseases has now been standardized on a national scale (Hess, 1976).

The information in the data bank can be used to create a prose summary of the patient's current status, and there are graphical capabilities that can plot specific parameters for a patient over time (Weyl et al., 1975). However, it may be in the analysis of stored clinical experience that the system has its greatest potential utility (Fries, 1976). In addition to performing search and statistical functions such as those developed in data bank systems for clinical investigation (Johnson and Barnett, 1977; Mabry et al., 1977), ARAMIS offers a prognostic analysis for a new patient when a management decision is to be made. Using the consultative services of the Stanford Immunology Division, an individual practitioner may select clinical indices for a patient and have them matched against those of other patients in the data bank. Based on two to five such descriptors, the computer locates relevant prior patients and prepares a report outlining their prognoses with respect to a variety of endpoints (e.g., death, development of renal failure, arthritic status, pleurisy). Therapy recommendations are also generated on the basis of a response index that is calculated for the matched patients. A prose case analysis for the physician's patient can also be generated; this readable document summarizes the relevant data from the data bank and explains the basis for the therapeutic recommendation.

The rheumatologic data bank generated under ARAMIS has now been expanded to involve a national network of immunologists who are accumulating time-oriented data on their patients. This national project

seeks in part to obtain enough data so that groups of retrieved patients will be sizable, thereby controlling for some observer variability and making the system's recommendations more statistically defensible.

3.3.3 Discussion of the Methodology

Data bank analysis systems have powerful capabilities to offer to the individual clinical decision maker. Furthermore, medical computing researchers recognize the potential value of large data banks in supporting many of the other decision-making approaches discussed in subsequent sections. There are important additional issues regarding data bank systems:

1. Data acquisition remains a major problem. Many systems have avoided direct physician-computer interaction but have then been faced with the expense and errors of transcription. The developers of one well-accepted record system still express their desire to implement a direct interface with the physician for these reasons, although they recognize the difficulties encountered in encouraging direct use of a computer system by doctors (Stead et al., 1977).[6]
2. Analysis of data in the system can be complicated by missing values that frequently occur, outlying values, and poor reproducibility of data over time and among physicians. Conversely, the system can itself be used to identify questionable values of tests or observations.
3. The decision aids provided tend to emphasize patient management rather than diagnosis. Feinstein's system (Feinstein et al., 1972) is only useful for patients with lung cancer, for example, and the ARAMIS prognostic routines, which are designed for patient management, assume that the patient's rheumatologic diagnosis is already known.
4. There is no formal correlation between the way expert physicians approach patient-management decisions and the way the programs arrive at recommendations. Feinstein and Koss felt that the acceptability of their system would be limited by a purely statistical approach, and they therefore chose to mimic human reasoning processes to a large extent (Koss and Feinstein, 1971), but their approach appears to be an exception.
5. Space requirements for data storage can be large since the decision aids of course require a comprehensive medical record system as a basic component.

Slamecka has distinguished between structured and empirical approaches to clinical consulting systems (Slamecka et al., 1977), pointing out

[6]Bischoff et al. (1983) have recently described ONCOCIN, an oncology decision advice system that has successfully required direct physician interaction and is based on the TOD patient record format.

that data banks provide a largely empirical basis for advice whereas structured approaches rely on judgmental knowledge elicited from the literature or from experts. It is important to note, however, that judgmental knowledge is itself based on empirical information. Even an expert's intuitions are based on observations and "data collection" over years of experience. Thus one might argue that large, complete, and flexible data banks *could* form the basis for large amounts of judgmental knowledge that we now have to elicit from other sources. Some researchers have indicated a desire to experiment with methods for the automatic generation of medical decision rules from data banks, and one component of the research on Slamecka's MARIS system is apparently pointed in that direction (Slamecka et al., 1977). Indeed, some of the most exciting and practical uses of large data banks may be found precisely at the interface with those knowledge-engineering tasks that have most confounded researchers in medical symbolic reasoning (Blum and Wiederhold, 1978).[7]

3.4 Mathematical Models of Physical Processes

3.4.1 Overview

Pathophysiologic processes can be well described by mathematical formulas in a limited number of clinical problem areas. Such domains have lent themselves readily to the development of computer-based decision aids since the issues are generally well defined. The actual techniques used by such programs tend to reflect the details of the individual applications, the most celebrated of which have been in pharmacokinetics (particularly digitalis dosing), acid-base/electrolyte disorders, and respiratory care (Menn et al., 1973).

It is important that cooperating experts assist with the definition of pertinent variables and the mathematical characterization of the relationships among them. The computer program requests the relevant data, makes the appropriate computations, and provides a clinical analysis or recommendation for therapy. Some of the programs have also incorporated branched-chain logic to guide decisions about what further data are needed for adequate analysis.[8]

Programs to assist with digitalis dosing have gradually introduced broader medical knowledge over the last ten years. The earliest work was

[7]See also Chapter 17.

[8]*Branched-chain logic* refers to mechanisms by which portions of a decision network can be considered or ignored depending on the data on a given case. For example, in an acid-base program the anion gap might be calculated and a branch point could then determine whether the pathway for analyzing an elevated anion gap would be required. If the gap were not elevated, that whole portion of the logic network could be skipped.

Jelliffe's (Jelliffe et al., 1970) and was based on his considerable experience studying the pharmacokinetics of the cardiac glycosides. His computer program used mathematical formulations based on parameters such as therapeutic goals (e.g., desired predicted blood levels), body weight, renal function, and route of administration. In one study he showed that computer recommendations reduced the frequency of adverse digitalis reactions from 35% to 12% (Jelliffe and Jelliffe, 1972). Later, another group revised the Jelliffe model to permit a feedback loop in which the digitalis blood levels obtained with initial doses of the drug were considered in subsequent therapy recommendations (Peck et al., 1973; Sheiner et al., 1975). More recently, a third group in Boston, noting the insensitivity of the first two approaches to the kinds of nonnumeric observations that experts tend to use in modifying digitalis therapy, augmented the pharmacokinetic model with a patient-specific model of clinical status (Gorry et al., 1978). Running their system in a monitoring mode, in parallel with actual clinical practice on a cardiology service, they found that each patient in the trial in whom toxicity developed had received more digitalis than would have been recommended by their program.

3.4.2 Example

Perhaps the best known program in this category is the interactive system developed at Boston's Beth Israel Hospital by Bleich. Originally designed as a program for assessment of acid-base disorders (Bleich, 1969), it was later expanded to consider electrolyte abnormalities as well (Bleich, 1971; 1972). The knowledge in Bleich's program is a distillation of his own expertise regarding acid-base and electrolyte disorders. The system begins by collecting initial laboratory data from the physician seeking advice on a patient's management. Branched-chain logic is triggered by abnormalities in the initial data so that only the pertinent sections of the extensive decision pathways created by Bleich are explored. The approach is therefore similar to the flowcharting techniques used by the clinical algorithms described earlier, but it involves more complex mathematical relationships than algorithms typically do. Essentially all questions asked by the program are numerical laboratory values or yes-no questions (e.g., "Does the patient have pitting edema?"). Depending on the complexity and severity of the case, the program eventually generates an evaluation note that may vary in length from a few lines to several pages. Included are suggestions regarding possible causes of the observed abnormalities and suggestions for correcting them. Literature references are also provided with the recommendations.

Although the program was made available at several east coast institutions, few physicians accepted it as an ongoing clinical tool. Bleich points out that part of the reason for this was the system's inherent educational impact; physicians simply began to anticipate its analysis after they had

used it a few times (Bleich, 1971).[9] The system's lack of sustained acceptance by physicians is probably due to more than its educational impact, however. For example, there is no feedback in the system; every patient is seen as a new case, and the program has no concept of following a patient's response to prior therapy. Furthermore, the program generates differential diagnosis lists but does not pursue specific etiologies; this can be particularly bothersome when there are multiple coexistent disturbances in a patient and the program simply suggests parallel lists of etiologies without noting or pursuing the possible interrelationships. Finally, the system is highly individualized in that it contains only the parameters and relationships that Bleich specifically thought were important to include in the logic network. Of course, human consultants also give personalized advice that may differ from that obtained from other experts. However, a group of researchers in Britain (Richards and Goh, 1977) who compared Bleich's program to four other acid-base/electrolyte systems, found total agreement among the programs in only 20% of test cases when these systems were asked to define the acid-base disturbance and the degree of compensation present. Their analysis does not reveal which of the programs reached the correct decision, however, and it may be that the results are more an indictment of the other four programs than a valid criticism of the advice from Bleich's acid-base component.

3.4.3 Discussion of the Methodologies

The programs mentioned in this section are very different in several respects, and each tends to overlap with other methodologies we have discussed. Bleich's program, for example, is essentially a complicated clinical algorithm interfaced with mathematical formulations of electrolyte and acid-base pathophysiology. As such, it suffers from the weaknesses of all algorithmic approaches, most importantly its highly structured and inflexible logic, which is unable to contend with unforeseen circumstances not specifically included in the algorithm. The digitalis dosing programs all draw on mathematical techniques from the field of biomedical modeling (Groth, 1977) but have recently shown more reliance on methods from other areas as well. In particular, these have included symbolic reasoning methods that allow clinical expertise to be encoded and used in conjunction with mathematical techniques (Gorry et al., 1978). The Boston group that developed this most recent digitalis program is interested in similarly developing an acid-base/electrolyte system so that judgmental knowledge of experts can be interfaced with the mathematical models of pathophysiology.[10]

---

[9]Subsequently, Bleich experimented with the program operating as a monitoring system, thereby avoiding direct interaction with the physician.

[10]See Chapter 14.

There is also a large research community of mathematicians who attempt to understand and characterize physical processes by devising simulation models (Groth, 1977). Although such models are largely empirical and have generally not found direct application in clinical medicine, their research role may eventually be broadened to provide practical decision aids through interfaces with the other paradigms described in this review.

The major strength of mathematical models is their ability to capture mathematically sound relationships in a concise and efficient computer program. However, the major limitation, as with most of the paradigms discussed here, is that few areas of medicine are amenable to firm, quantitative description. Because the accuracy of the results depends on correct identification of relevant parameters, the precision and certainty of the relationships among them, and the accuracy of the techniques for measuring them, mathematical models have limited applicability at present. Furthermore, those domains that *do* lend themselves to mathematical description may still benefit from interactions with symbolic reasoning techniques, as has been demonstrated in the Digitalis Therapy Advisor (Gorry et al., 1978).

3.5. Statistical Pattern-Matching Techniques

3.5.1 Overview

Pattern-recognition techniques define the mathematical relationship between measurable features and classifications of objects (Duda and Hart, 1973; Kanal, 1974). In medicine, the presence or absence of each of several signs and symptoms in a patient may be definitive for the classification of the patient as abnormal or into the category of a specific disease. Pattern-recognition techniques are also used for prognosis (Armitage and Gehan, 1974) or predicting disease duration, time course, and outcomes. These techniques have been applied to a variety of medical domains, such as image processing and signal analysis, in addition to computer-assisted diagnosis.

In order to find the diagnostic pattern, or discriminant function, the method requires a training set of objects for which the correct classification is already known, as well as reliable values for their measured features. If the form and parameters are not known for the statistical distributions underlying the features, then they must be estimated. *Parametric* techniques focus on learning the parameters of the probability density functions, while *nonparametric* (or "distribution-free") techniques make no assumptions about the form of the distributions. After training, then, the pattern can

be compared to new, unclassified objects to aid in deciding the category to which the new object belongs.[11]

There are numerous variations on this general methodology, most notably in the mathematical techniques used to extract characteristic measurements (the features) and to find and refine the pattern classifier during training. For example, linear regression analysis is a commonly used technique for finding the coefficients of an equation that defines a recurring pattern or category of diagnostic or prognostic interest. A class of patients can be described by a feature vector $X = [x_1, x_2, \ldots, x_n]$ (where x_i is one of n descriptive variables). The goal is to produce an equation relating the posterior probabilities[12] of each diagnostic class to the feature vector through a set of n coefficients (a_i):[13]

$$P(D_i|X) = a_1 x_1 + a_2 x_2 + \cdots + a_n x_n$$

Recent work emphasizes structural relationships among sets of features more than statistical ones.

Three of the best known training criteria for the discriminant function are the following:

a. *least squared error criterion:* choose the function that minimizes the squared differences between predicted and observed measurement values;

b. *clustering criterion:* choose the function that produces the tightest clusters;

c. *Bayes' criterion:* choose the function that has the minimum cost associated with incorrect diagnoses.[14]

Ten commonly used mathematical models based on these criteria have been shown to produce remarkably similar diagnostic results for the same data (Croft, 1972).

---

[11]It is possible to detect patterns, even without a known classification for objects in the training set, with so-called unsupervised learning techniques. Also, it is possible to work with both numerical and nonnumerical measurements.

[12]The posterior probability of a diagnostic class, represented as $P(D_i|X)$, is the probability that a patient falls in diagnostic category D_i *given* that the feature vector X has been observed.

[13]See Levi et al. (1976) for a study in which the coefficients are reported because of their medical import.

[14]This is one of many uses of Bayes' Theorem, a definitional rule that relates posterior and prior probabilities. For an overview of its use as a diagnostic rule (as opposed to a training criterion) and a definition of the formula, see Section 3.6.

3.5.2 Example

There are numerous papers on the use of pattern-recognition methods in medicine. Armitage and Gehan (1974) discuss three examples of prognostic studies, with an emphasis on regression methods. Goldwyn et al. (1971) discuss uses of cluster analysis. One diagnostic application by Patrick (1977) uses Bayes' criterion to classify patients having chest pains into three categories: D_1, acute myocardial infarction (MI); D_2, coronary insufficiency; and D_3, noncardiac causes of chest pain. The need for early diagnosis of heart attacks without laboratory tests is a prevalent problem, yet physicians are known to misclassify about one-third of the patients in categories D_1 and D_2 and about 80% of those in D_3. In order to determine the correct classification, each patient in the training set was classified after three days, based on laboratory data including electrocardiogram (ECG) and blood data (cardiac enzymes). There remained some uncertainty about several patients with "probable MI." Seventeen variables were selected from many: nine features with continuous values (including age, heart rates, white blood count, and hemoglobin) and eight features with discrete values (sex and seven ECG features).

The training data were measurements on 247 patients. The decision rule was chosen using Bayes' Theorem to compute the posterior probabilities of each diagnostic class given the feature vector X ($X = [x_1, x_2, \ldots, x_{17}]$). Then a decision rule was chosen to minimize the probability of error by adjusting the coefficients on the feature vector X such that for the correct class D_i:

$$P(D_i|X) = \max[P(D_1|X), P(D_2|X), P(D_3|X)]$$

The class conditional probability density functions must be estimated initially, and the performance of the decision rule depends on the accuracy of the assumed model.

Using the same 247 patients for testing the approach, the trained classifier averaged 80% correct diagnoses over the three classes, using only data available at the time of admission. Physicians, using more data than the computer, averaged only 50.5% correct over these three categories for the same patients. Training the classifier with a subset of the patients and using the remainder for testing produced results that were nearly as good.

3.5.3 Discussion of the Methodology

The number of reported medical applications of pattern-recognition techniques is large, but there are also numerous problems associated with the approach. The most obvious difficulties are choosing the set of features in the first place, collecting reliable measurements on a large sample, and verifying the initial classifications among the training data. Current tech-

niques are inadequate for problems in which trends or movement of features are important characteristics of the categories. Also the problems for which existing techniques are accurate are those that are well characterized by a small number of features ("dimensions of the space").

As with all techniques based in statistics, the size of the sample used to define the categories is an important consideration. As the number of important features and the number of relevant categories increase, the required size of the training set also increases. In one test (Croft, 1972) pattern classifiers trained to discriminate among 20 disease categories from 50 symptoms were correct 51–64% of the time. The same methods were used to train classifiers to discriminate between 2 of the diseases from the same 50 symptoms and produced correct diagnoses 92–98% of the time.

The *context* in which a local pattern is identified raises problems related to the issue of using medical knowledge. It is difficult to find and use classifiers that are best for a small decision, such as whether an area of an x-ray is inside or outside the heart, and to integrate those into a global classifier, such as one for abnormal heart volume.

Accurate application of a classifier in a hospital setting also requires that the measurements in that clinical environment be consistent with the measurements used to train the classifier initially. For example, if diseases and symptoms are defined differently in the new setting, or if lab test values are reported in different ranges, or if different lab tests are used, then decisions based on the classification are not reliable.

Pattern-recognition techniques are often misapplied in medical domains in which the assumptions are violated. Some of the difficulties noted above are avoided in systems that integrate structural knowledge into the numerical methods and in systems that integrate human and machine capabilities into single, interactive systems. These modifications will overcome one of the major difficulties seen in completely automated systems, that of providing the system with good "intuitions" based on an expert's *a priori* knowledge and experience (Kanal, 1974).

3.6 Bayesian Statistical Approaches

3.6.1 Overview

More work has been done on Bayesian approaches to computer-based medical decision making than on any of the other methodologies we have discussed. The appeal of Bayes' Theorem[15] is clear: it potentially offers an exact method for computing the probability of a disease based on observations and data regarding the frequency with which these observations

[15]Also often referred to as Bayes' Rule, discriminant, or criterion.

are known to occur for specified diseases. In several domains the technique has been shown to be exceedingly accurate, but there are also several limitations to the approach, which we discuss below.

In its simplest formulation, Bayes' Theorem can be seen as a mechanism to calculate the probability of a disease, in light of specified evidence, from the *a priori* probability of the disease and the conditional probabilities relating the observations to the diseases in which they may occur. For example, suppose disease D_i is one of n mutually exclusive diagnoses under consideration and E is the evidence or observations supporting that diagnosis. Then if $P(D_i)$ is the *a priori* probability of the ith disease:[16]

$$P(D_i|E) = \frac{P(D_i)\,P(E|D_i)}{\sum_{j=1}^{n} P(D_j)\,P(E|D_j)}$$

The theorem can also be represented or derived in a variety of other forms, including an odds/likelihood ratio formulation. We cannot include such details here, but any introductory statistics book or Lusted's volume (1968) presents the subject in detail.

Among the most commonly recognized problems with the use of a Bayesian approach is the large amount of data required to determine all the conditional probabilities needed in the rigorous application of the formula. Chart review or computer-based analysis of large data banks occasionally allows most of the necessary conditional probabilities to be obtained. A variety of additional assumptions must be made, for example: (1) the diseases under consideration are assumed mutually exclusive and exhaustive (i.e., the patient is assumed to have exactly one of the n diseases); (2) the clinical observations are assumed to be conditionally independent over a given disease;[17] and (3) the incidence of the symptoms of a disease is assumed to be stationary (i.e., the model generally does not allow for changes in disease patterns over time).

One of the earliest Bayesian programs was the system of Warner et al. (1964) for the diagnosis of congenital heart disease. They compiled data on 83 patients and generated a symptom-disease matrix consisting of 53 symptoms (attributes) and 35 disease entities. The diagnostic performance of the computer, based on the presence or absence of the 53 symptoms in a new patient, was then compared to that of two experienced physicians. The program was shown to reach

[16]Here, $P(D_i|E)$ is the probability of the ith disease *given* that evidence E has been observed; $P(E|D_i)$ is the probability that evidence E will be observed in the setting of the ith disease.

[17]The purest form of Bayes' Theorem allows conditional dependencies and the order in which evidence is obtained to be explicitly considered in the analysis. However, the number of required conditional probabilities is so unwieldy that conditional independence of observations and nondependence on the order of observations are generally assumed (see Chapter 9).

diagnoses with an accuracy equal to that of the experts. Furthermore, system performance was shown to improve as the statistics in the symptom-disease matrix stabilized with the addition of increasing numbers of patients.

In 1968 Gorry and Barnett (1968a) pointed out that Warner's program required making all 53 observations for every patient to be diagnosed, a situation that would not be realistic for many clinical applications. They therefore used a modification of Bayes' Theorem in which observations are considered sequentially.[18] Their computer program analyzed observations one at a time, suggested which test would be most useful if performed next, and included termination criteria so that a diagnosis could be reached, when appropriate, without a need to make all the observations. Decisions regarding tests and termination were made on the basis of calculations of expected costs and benefits at each step in the logical process.[19] Using the same symptom-disease matrix developed by Warner, they were able to attain equivalent diagnostic performance using only 6.9 tests on average.[20] They pointed out that, because the costs of medical tests may be significant (in terms of patient discomfort, time expended, and financial expense), the use of inefficient testing sequences should be regarded as ineffective diagnosis. Warner has also more recently included Gorry's and Barnett's sequential diagnosis approach in an application regarding structured patient history-taking (Warner et al., 1972b).

The medical computing literature now includes many examples of Bayesian diagnosis programs, most of which have used the nonsequential approach, in addition to the necessary assumptions of symptom independence and mutual exclusiveness of disease as discussed above. One particularly successful research effort has been chosen for discussion.

3.6.2 Example

Since the late 1960s de Dombal and associates, at the University of Leeds, England, have been studying the diagnostic process and developing computer-based decision aids using Bayesian probability theory. Their area of investigation has been gastrointestinal diseases, originally acute abdominal

---

[18]A similar approach was devised in the Soviet Union at approximately the same time by Vishnevskiy and associates. Their analyses and a summary of the impressive amount of statistical data they have amassed are contained in Vishnevskiy et al. (1973).

[19]See the decision theory discussion in Section 3.7.

[20]Tests for determining attributes were defined somewhat differently than they had been by Warner. Thus the maximum number of tests was 31 rather than the 53 observations used in the original study.

pain (de Dombal et al., 1972) with more recent analyses of dyspepsia (Horrocks and de Dombal, 1975) and gastric carcinoma (Zoltie et al., 1977).

Their program for assessment of acute abdominal pain was evaluated in the emergency room of their affiliated hospital (de Dombal et al., 1972). Emergency room physicians filled out data sheets summarizing clinical and laboratory findings on 304 patients presenting with abdominal pain of acute onset. The data from these sheets became the attributes that were subjected to Bayesian analysis; the required conditional probabilities had been previously compiled from a large group of patients with one of seven possible diagnoses.[21] Thus the Bayesian formulation assumed each patient had one of these diseases and selected the most likely on the basis of recorded observations. Diagnostic suggestions were obtained in batch mode and did not require direct interaction between physician and computer; the program could generate results within 30 seconds to 15 minutes depending on the level of system use at the time of analysis (Horrocks et al., 1972). Thus the computer output could have been made available to the emergency room physician, on average, within 5 minutes after the data form was completed and handed to the technician assisting with the study.

During the study (de Dombal et al., 1972), however, these computer-generated diagnoses were simply saved and later compared to (a) the diagnoses reached by the attending clinicians and (b) the ultimate diagnosis verified at surgery or through appropriate tests. Although the clinicians reached the correct diagnosis in only 65–80% of the 304 cases (with accuracy depending on the individual's training and experience), the program was correct in 91.8% of cases. Furthermore, in six of the seven disease categories the computer was shown to be more likely to assign the patient to the correct disease category than was the senior clinician in charge of a case. Of particular interest was the program's accuracy regarding appendicitis—a diagnosis that is often made incorrectly. In no cases of appendicitis did the computer fail to make the correct diagnosis, and in only six cases were patients with nonspecific abdominal pain incorrectly classified as having appendicitis. Based on the actual clinical decisions, however, more than 20 patients with nonspecific abdominal pain were unnecessarily taken to surgery for appendicitis, and in six cases patients with appendicitis were "watched" for more than eight hours before they were finally taken to the operating room.

These investigators also performed a fascinating experiment in which they compared the program's performance based on data derived from 600 real patients with the accuracy the system achieved using "estimates" of conditional probabilities obtained from experts (Leaper et al., 1972).[22]

---

[21]Appendicitis, diverticulitis, perforated ulcer, cholecystitis, small bowel obstruction, pancreatitis, and nonspecific abdominal pain were the seven possibilities.

[22]Such estimates are referred to as "subjective" or "personal" probabilities, and some investigators have argued that they should be utilized in Bayesian systems when formally derived conditional probabilities are not available (Lusted, 1968).

As discussed above, the program was significantly more effective than the unaided clinician when real-life data were utilized. However, it performed significantly *less* well than did clinicians when expert estimates were used. The results supported what several other observers have found, namely that physicians often have very little idea of the "true" probabilities for symptom-disease relationships.

Another study of note at the University of Leeds was an analysis of the effect of the system on the performance of clinicians (de Dombal et al., 1974). The trial we have mentioned involving 304 patients was eventually extended to 552 before termination. Although the computer's accuracy remained in the range of 91% throughout this period, the performance of clinicians was noted to improve markedly over time. Fewer negative laparotomies were performed, for example, and the number of acute appendices that perforated (ruptured) also declined. However, these data reverted to baseline after the study was terminated, suggesting that the constant awareness of computer monitoring and feedback regarding system performance had temporarily generated a heightened awareness of intellectual processes among the hospital's surgeons.

3.6.3 Discussion of the Methodology

The ideal matching of the problem of acute abdominal pain and Bayesian analysis must be emphasized; the technique cannot necessarily be as effectively applied in other medical domains where the following limitations of the Bayesian approach may have a greater impact:

1. The assumption of conditional independence of symptoms usually does not apply and can lead to substantial errors in certain settings (Norusis and Jacquez, 1975a). This has led some investigators to seek new numerical techniques that avoid the independence assumption (Cumberbatch and Heaps, 1976). If a pure Bayesian formulation is used without making the independence assumption, however, the number of required conditional probabilities becomes prohibitive for complex real-world problems (see Chapter 9).
2. The assumption of mutual exclusiveness and exhaustiveness of disease categories is usually false. In actual practice concurrent and overlapping disease categories are common. In de Dombal's system, for example, many of the abdominal pain diagnoses missed were outside the seven "recognized" possibilities; if a program starts with an assumption that it need consider only a small number of defined likely diagnoses, it will inevitably miss the rare or unexpected cases (precisely the ones with which the clinician is most apt to need assistance).
3. In many domains it may be inaccurate to assume that relevant conditional probabilities are stable over time (e.g., the likelihood that a par-

ticular bacterium will be sensitive to a specific antibiotic). Furthermore, diagnostic categories and definitions are constantly changing, as are physicians' observational techniques, thereby invalidating data previously accumulated.[23] A similar problem results from variations in *a priori* probabilities depending on the population from which a patient is drawn.[24] Some observers feel that these are major limitations to the use of Bayesian techniques (Edwards, 1972).

In general, then, a purely Bayesian approach can so constrain problem formulation as to make a particular application unrealistic and hence unworkable. Furthermore, even when diagnostic performance is excellent, such as in de Dombal's approach to abdominal pain evaluation, clinical implementation and system acceptance will generally be difficult. Forms of representation that allow explanation of system performance in familiar terms (i.e., a more congenial interface with physician users) will heighten clinical acceptance; it is at this level that Bayesian statistics and symbolic reasoning techniques may most beneficially interact.

3.7 Decision Theory Approaches

3.7.1 Overview

Bayes' Theorem is only one of several techniques used in the larger field of decision analysis, and there has recently been increasing interest in the ways in which decision theory might be applied to medicine and adapted for automation. Several excellent surveys of the field are available in basic reviews (Howard, 1968), textbooks (Raiffa, 1968), and medically oriented journal articles (McNeil et al., 1975; Schwartz et al., 1973; Taylor, 1976). In general terms, decision analysis can be seen as any attempt to consider *values* associated with choices, as well as probabilities, in order to analyze the processes by which decisions are made or should be made. Schwartz identifies the calculation of "expected value" as central to formal decision analysis (Schwartz et al., 1973). Ginsberg contrasts medical classification problems (e.g., diagnosis) with broader decision problems (e.g., "What should I do for this patient?") and asserts that most important medical decisions fall in the latter category and are best approached through decision analysis (Ginsberg, 1972).

[23]Although gradual changes in definitions or observational techniques may be statistically detectable by data base analysis, a Bayesian analysis that uses such data is inevitably prone to error.

[24]de Dombal has examined such geographic and population-based variations in probabilities and has reported early results of his analysis (de Dombal and Gremy, 1976).

The following topics are among the central issues in the field:

1. *Decision trees.* The decision-making process can be seen as a sequence of steps in which the clinician selects a path through a network of plausible events and actions. Nodes in this tree-shaped network are of two kinds: *decision nodes,* where the clinician must choose from a set of actions, and *chance nodes,* where the outcome is not directly controlled by the clinician but is a probabilistic response of the patient to some action taken. For example, a physician may choose to perform a certain test (decision node) but the occurrence or nonoccurrence of complications may be largely a matter of statistical likelihood (chance node). By analyzing a difficult decision process before taking any actions, it may be possible to delineate in advance all pertinent chance and decision nodes, all plausible outcomes, plus the paths by which these outcomes might be reached. Furthermore, data may exist to allow specific probabilities to be associated with each chance node in the tree.

2. *Expected values.* In actual practice physicians make sequential decisions based on more than the probabilities associated with the chance node that follows. For example, the best possible outcome is not necessarily sought if the costs associated with that "path" far outweigh those along alternate pathways (e.g., a definitive diagnosis may not be sought if the required testing procedure is expensive or painful and patient management will be unaffected; similarly, some patients prefer to "live with" an inguinal hernia rather than undergo a surgical repair procedure). Thus anticipated costs (financial expenditures, complications, discomfort, patient preference) can be associated with the decision nodes. Using the probabilities at chance nodes, the costs at decision nodes, and the "values"[25] of the various outcomes, an "expected value" for each pathway through the tree (and in turn each node) can be calculated. The ideal pathway, then, is the one that maximizes the expected value.

3. *Eliciting values.* Obtaining from physicians and patients the cost and values they associate with various tests and outcomes can be a formidable problem, particularly since formal analysis requires expressing the various costs in standardized units. One approach has been simply to ask for value ratings on a hypothetical scale, but it can be difficult to get physicians or patients to keep the values separate from their knowledge of the probabilities linked to the associated chance nodes. An alternate approach has been the development of lottery games. Inferences regarding values can be made by identifying the odds, in a hypothetical lottery, at which the physician or patient is indifferent regarding taking a course of action with certain outcome or betting on a course with preferable outcome but with a finite chance of significant negative costs if the "bet" is lost. In certain

[25]Also termed "utilities" in some references; hence the term "utility theory" (Raiffa, 1968).

settings this approach may be accepted and may provide important guidelines in decision making (Pauker and Pauker, 1977).

4. *Test evaluation.* Since the tests that lie at decision nodes are central to clinical decision analysis, it is crucial to know the predictive value of tests that are available. This leads to consideration of test sensitivity, specificity, disease prevalence, receiver operator characteristic curves, and sensitivity analysis (Komaroff, 1979; McNeil and Adelstein, 1977).

Many of the major studies of clinical decision analysis have not specifically involved computer implementations. Schwartz et al. examined the workup of renal vascular hypertension, developing arguments to show that for certain kinds of cases a purely qualitative theoretical approach was feasible and useful (Schwartz et al., 1973). However, they showed that for more complex, clinically challenging cases the decisions could not be adequately sorted out without the introduction of numerical techniques. Since it was impractical to assume that clinicians would ever take the time to carry out a detailed quantitative decision analysis by hand, they pointed out the logical role for the computer in assisting with such tasks and accordingly developed the system we discuss as an example below (Gorry et al., 1973).

Other colleagues of Schwartz at Tufts–New England Medical Center have been similarly active in applying decision theory to clinical problems. Pauker and Kassirer have examined applications of formal cost-benefit analysis to therapy selection (Pauker and Kassirer, 1975), and Pauker has also looked at possible applications of the theory to the management of patients with coronary artery disease (Pauker, 1976). An entire issue of the *New England Journal of Medicine* has also been devoted to papers on this methodology (Inglefinger, 1975).

3.7.2 Example

Computer implementations of clinical decision analysis have appeared with increasing frequency since the mid-1960s. Perhaps the earliest major work was that of Ginsberg at the Rand Corporation (Ginsberg, 1971), with more recent systems reported by Pliskin and Beck (1976) and Safran et al. (1977).

We will briefly describe here the program of Gorry et al., developed for the management of acute renal failure (Gorry et al., 1973). Drawing upon Gorry's experience with the sequential Bayesian approach previously mentioned (Gorry and Barnett, 1968a), the investigators recognized the need to incorporate some way of balancing the dangers and discomforts of a procedure against the value of the information to be gained. They divided their program into two parts: phase I considered only tests with minimal risk (e.g., history, examination, blood tests), and phase II considered procedures involving more risk and inconvenience. The phase I pro-

gram considered 14 of the most common causes of renal failure and used a sequential test selection process based on Bayes' Theorem and omitting more advanced decision theory methodology (Gorry and Barnett, 1968a). The conditional probabilities utilized were subjective estimates obtained from an expert nephrologist and were therefore potentially as problematic as those discussed by Leaper et al. (1972). The researchers found that they had no choice but to use expert estimates, however, since detailed quantitative data were not available either in data banks or in the literature.

It is in the phase II program that the methods of decision theory were employed because it was in this portion of the decision process that the risks of procedures became important considerations. At each step in the decision process, this program considers whether it is best to treat the patient immediately or to first carry out an additional diagnostic test. To make this decision the program identifies the treatment with the highest current expected value (in the absence of further testing) and compares this with the expected values of treatments that could be instituted if another diagnostic test were performed. Comparison of the expected values are made in light of the risk of the test in order to determine whether the overall expected value of the test is greater than that of immediate treatment. The relevant values and probabilities of outcomes of treatment were obtained as subjective estimates from nephrologists in the same way that symptom-disease data had been obtained. All estimates were gradually refined as Gorry and his colleagues gained experience using the program, however.

The program was evaluated on 18 test cases in which the true diagnosis was uncertain but two expert nephrologists were willing to make management decisions. In 14 of the cases the program selected the same therapeutic plan or diagnostic test as was chosen by the experts. For 3 of the 4 remaining cases the program's decision was the physicians' second choice and was, they felt, a reasonable alternative plan of action. In the last case the physicians also accepted the program's decision as reasonable, although it was not among their first two choices.

3.7.3 Discussion of the Methodology

The excellent performance of Gorry's program, despite its reliance on subjective estimates from experts, may serve to emphasize the importance of the clinical analysis that underlies the decision-theory approach. The reasoning steps in managing clinical cases have been dissected in such detail that small errors in the probability estimates are apparently much less important than they were for de Dombal's purely Bayesian approach (Leaper et al., 1972). Gorry suggests this may be simply because the decisions made by the program are based on the combination of large aggregates of such numbers, but this argument should apply equally for a Bayesian system. It seems to us more likely that distillation of the clinical domain in a formal

decision tree gives the program so much more *knowledge* of the clinical problem that the quantitative details become somewhat less critical to overall system operation. The explicit decision network is a powerful knowledge structure; the "knowledge" in de Dombal's system lies in conditional probabilities alone, and there is no larger scheme to override the propagation of error as these probabilities are mathematically manipulated by the Bayesian routines.

The decision theory approach is not without problems, however. Perhaps the most difficult problem is assigning numerical values (e.g., dollars) to a human life or a day of health, etc. Some critics feel this is a major limitation to the methodology (Warner, 1978). Overlapping or coincident diseases are also not well managed, unless specifically included in the analysis, and the Bayesian foundation for many of the calculations still assumes mutually exclusive and exhaustive disease categories. Problems of symptom-conditional dependence still remain, and there is no easy way to include knowledge regarding the time course of diseases.[26] Gorry points out that his program was also incapable of recognizing circumstances in which two or more actions should be carried out concurrently. Furthermore, decision theory *per se* does not provide the kind of focusing mechanisms that clinicians tend to use when they assume an initial diagnostic hypothesis in dealing with a patient, then discard it only if subsequent data make that hypothesis no longer tenable. Other similar strategies of clinical reasoning are becoming increasingly well recognized (Kassirer and Gorry, 1978) and account in large part for the applications of symbolic reasoning techniques to be discussed in the next section.

3.8 Symbolic Reasoning Approaches

3.8.1 Overview

In the early 1970s researchers at several institutions simultaneously began to investigate the potential clinical applications of symbolic reasoning techniques drawn from the branch of computer science known as artificial intelligence (AI). The field is introduced in a recent book by Winston (1977). The term *artificial intelligence* is generally accepted to include those computer applications that involve symbolic inference rather than strictly numerical calculation. Examples include programs that reason about mineral exploration, organic chemistry, or molecular biology; programs that converse in English and understand spoken sentences; and programs that generate theories from observations.

[26] *Ed. note*: More recently, Markov modeling techniques have been introduced to allow consideration of the temporal aspects of disease progression for decision analysis approaches.

Such programs gain their power from qualitative, experiential judgments, codified in so-called rules of thumb or heuristics, in contrast to numerical calculation programs whose power derives from the analytical equations used. The heuristics focus the attention of the reasoning program on parts of the problem that seem most critical and parts of the knowledge base that seem most relevant. They also guide the application of the domain knowledge to an individual case by deleting items from consideration as well as focusing on items. The result is that these programs pursue a line of reasoning, as opposed to following a sequence of steps in a calculation. Among the earliest symbolic inference programs in medicine was the diagnostic interviewing system of Kleinmuntz and McLean (1968). Other early work included Wortman's information processing system, the performance of which was largely motivated by a desire to understand and simulate the psychological processes of neurologists reaching diagnoses (Wortman, 1972).

It was the landmark paper by Gorry in 1973, however, that first critically analyzed conventional approaches to computer-based clinical decision making and outlined his motivation for turning to newer symbolic techniques (see Chapter 2). He used the acute renal failure program discussed above (Gorry et al., 1973) as an example of the problems arising when decision analysis is used alone. In particular, he analyzed some of the cases on which the renal failure program had failed but the physicians considering the cases had performed well. His conclusions from these observations include the following four points:

1. Clinical judgment is based less on detailed knowledge of pathophysiology than it is on gross chunks of knowledge and a good deal of detailed experience from which rules of thumb are derived.
2. Clinicians know facts, of course, but their knowledge is also largely judgmental. The rules they learn allow them to focus attention and generate hypotheses quickly. Such heuristics permit them to avoid detailed search through the entire problem space.
3. Clinicians recognize levels of belief or certainty associated with many of the rules they use, but they do not routinely quantitate or use these certainty concepts in any formal statistical manner.
4. It is easier for experts to state their rules in response to perceived misconceptions in others than it is for them to generate such decision criteria *a priori*.

In the renal failure program medical knowledge was embedded in the structure of the decision tree. This knowledge was never explicit, and additions to the experts' judgmental rules generally required changes to the tree itself.

Based on observations such as those above, Gorry identified at least three important problems for investigation:

1. *Medical concepts.* Clinical decision aids traditionally had no true "understanding" of medicine. Although explicit decision trees had given the decision theory programs a greater sense of the pertinent associations, medical knowledge and the heuristics for problem solving in the field had never been explicitly represented or used. So-called common sense was often clearly lacking when the programs failed, and this was often what most alienated potential physician users.
2. *Conversational capabilities.* Gorry argued that further research on the development of computer-based linguistic capabilities was crucial both for capturing knowledge from collaborating experts and for communicating with physician users.
3. *Explanation.* Diagnostic programs had seldom emphasized an ability to explain the basis for their decisions in terms understandable to the physician. System acceptability was therefore inevitably limited; the physician would often have no basis for deciding whether to accept the program's advice and might therefore resent what could be perceived as an attempt to dictate the practice of medicine.

Gorry's group at M.I.T. and Tufts developed new approaches to examining the renal failure problem in light of these observations (see Chapter 6).

Because of the limitations of the older techniques, it was perhaps inevitable that some medical researchers would turn to the AI field for new methodologies. Major research areas in AI include knowledge representation, heuristic search, natural language understanding and generation, and models of thought processes—all topics clearly pertinent to the problems we have been discussing. Furthermore, AI researchers were beginning to look for applications in which they could apply some of the techniques they had developed in theoretical domains. This community of researchers has grown in recent years, and a recent issue of *Artificial Intelligence* was devoted entirely to applications of AI to biology, medicine, and chemistry (Sridharan, 1978).

Among the programs using symbolic reasoning techniques are several systems that have been particularly novel and successful. At the University of Pittsburgh, Pople, Myers, and Miller have developed a system called INTERNIST that assists with test selection for the diagnosis of *all* diseases in internal medicine (Pople et al., 1975). This awesome task has been remarkably well attacked to date, with the program correctly diagnosing a large percentage of the complex cases selected from clinical pathologic conferences in the major medical journals (see Chapter 8). The program utilizes a hierarchical disease categorization, an *ad hoc* scoring system for quantifying symptom-disease relationships, plus some clever heuristics for focusing attention, discriminating between competing hypotheses, and diagnosing concurrent diseases (Pople, 1977). The system currently has an inadequate human interface, however, and is not yet implemented for clinical trials.

Weiss, Kulikowski, and Amarel (Rutgers University) and Safir (Mt. Sinai Hospital, New York City) have developed a model of reasoning regarding disease processes in the eye, specifically glaucoma (see Chapter 7). In this specialized application area it has been possible to map relationships between observations, pathophysiologic states, and disease categories. The resulting causal-associational network (termed CASNET) forms the basis for a reasoning program that gives advice regarding disease states in glaucoma patients and generates management recommendations. The system currently has a limited human interface, however, and is not yet implemented for clinical trials.

For AI researchers the question of how best to manage uncertainty in medical reasoning remains a central issue. All the programs mentioned have developed *ad hoc* weighting programs and avoided formal statistical approaches. Others have turned to the work of statisticians and philosophers of science who have devised theories of approximate or inexact reasoning. For example, Wechsler (1976) describes a program that is based on Zadeh's fuzzy set theory (Zadeh, 1965), and Shortliffe and Buchanan (1975) have turned to confirmation theory for their model of inexact reasoning.

3.8.2 Example

The symbolic reasoning program selected for discussion is the MYCIN system at Stanford University (Shortliffe, 1976; Buchanan and Shortliffe, 1984). The researchers cited a variety of design considerations that motivated the selection of AI methodologies for the consultation system they were developing (Shortliffe et al., 1974). They primarily wanted it to be useful to physicians and therefore emphasized the selection of a problem domain in which physicians had been shown to err frequently, namely the selection of antibiotics for patients with infections. They also cited human issues that they felt were crucial to make the system acceptable to physicians:

1. the system should be able to explain its decisions in terms of a line of reasoning that a physician can understand;
2. the system should be able to justify its performance by responding to questions expressed in simple English;
3. the system should be able to "learn" new information rapidly by interacting directly with experts;
4. the system's knowledge should be easily modifiable so that perceived errors can be corrected rapidly before they recur in another case; and
5. the interaction should be engineered with the user in mind (in terms of prompts, answers, and information volunteered by the system as well as by the users).

All these design goals were based on the observation that previous computer decision aids had generally been poorly accepted by physicians, even when they were shown to perform well on the tasks for which they were designed. MYCIN's developers felt that barriers to acceptance were largely conceptual and could be counteracted in large part if a system were perceived as a clinical *tool* rather than a dogmatic replacement for the primary physician's own reasoning.

Knowledge of infectious diseases is represented in MYCIN as production rules, each containing a "packet" of knowledge obtained from collaborating experts (Shortliffe, 1976).[27] A production rule is simply a conditional statement that relates observations to associated inferences that may be drawn. For example, a MYCIN rule might state that "*if* a bacterium is a gram-positive coccus growing in chains, *then* it is apt to be a streptococcus." MYCIN's power is derived from such rules in a variety of ways:

1. it is the program that determines which rules to use and how they should be chained together to make decisions about a specific case;[28]
2. the rules can be stored in a machine-readable format but translated into English for display to physicians;
3. by removing, altering, or adding rules, we can rapidly modify the system's knowledge structures without explicitly restructuring the entire knowledge base; and
4. the rules themselves can often form a coherent explanation of system reasoning if the relevant ones are translated into English and displayed in response to a user's question.

Associated with all rules and inferences are numerical weights reflecting the degree of certainty associated with them. These numbers, termed *certainty factors,* form the basis for the system's inexact reasoning (Shortliffe and Buchanan, 1975). They allow the judgmental knowledge of experts to be captured in rule form and then utilized in a consistent fashion.

The MYCIN system has been evaluated regarding its performance at therapy selection for patients with either septicemia (Yu et al., 1979b) or meningitis (Yu et al., 1979a). The program performs comparably to experts in these two task domains, but it has no rules regarding the other infectious disease problem areas. Further knowledge base development would therefore be required before MYCIN could be made available for clinical use; hence questions regarding its acceptability to physicians cannot be fully assessed. However, the required implementation stages have been delineated (Shortliffe and Davis, 1975), attention has been paid to all the design criteria mentioned above, and the program does have a powerful explanation capability (Scott et al., 1977).

---

[27]Production rules are a methodology frequently employed in AI research (Davis and King, 1977) and effectively applied to other scientific problem domains (Buchanan and Feigenbaum, 1978).

[28]The control structure utilized is termed *goal-oriented* and is similar to the consequent-theorem methodology used in PLANNER (Hewitt, 1972).

3.8.3 Discussion of the Methodology

Whereas the computations used by the other paradigms mostly involve straightforward application of well-developed computing techniques, artificial intelligence methods are largely experimental; new approaches to knowledge representation, language understanding, heuristic search, and the other symbolic reasoning problems we have mentioned are still needed. Thus the AI programs tend to be developed in research environments, where short-term practical results are unlikely to be found. However, out of this research are emerging techniques for coping with many of the problems encountered by other paradigms we have discussed. AI researchers have developed promising methods for handling concurrent diseases (Pople, 1977) (see also Chapter 8), assessing the time course of disease (Fagan et al., 1979), and acquiring adequate structured knowledge from experts (Davis and Buchanan, 1977). Furthermore, inexact reasoning techniques have been developed and implemented (Shortliffe and Buchanan, 1975), although they tend to be justified largely on intuitive grounds. In addition, the techniques of artificial intelligence provide a way to respond to many of Gorry's observations regarding the three major inadequacies of earlier paradigms described above: (1) the medical AI programs all stress the representation of medical knowledge and an "understanding" of the underlying concepts; (2) many of them have conversational capabilities that draw on language processing research; and (3) explanation capabilities have been a primary focus of systems such as MYCIN.

Szolovits and Pauker have recently reviewed some applications of AI to medicine and have attempted to weigh the successes of this young field against the very real problems that lie ahead (see Chapter 9). They identify several deficiencies of current systems. For example, termination criteria are still poorly understood. Although INTERNIST can diagnose simultaneous diseases, it also pursues all abnormal findings to completion, even though a clinician often ignores minor unexplained abnormalities if the rest of a patient's clinical status is well understood. In addition, although some of these programs now cleverly mimic some of the reasoning styles observed in experts (Elstein et al., 1978; Kassirer and Gorry, 1978), it is less clear how to keep the systems from abandoning one hypothesis and turning to another one as soon as new information suggests another possibility. Programs that operate this way appear to digress from one topic to another—a characteristic that decidedly alienates a user regardless of the validity of the final diagnosis or advice.

Still largely untapped is the power of an AI program to understand its own knowledge base, i.e., the structure and content of the reasoning mechanisms as well as of the medical facts. In effect, AI programs have the ability to "know what they know," the best working example of which can be found in the prototype system named TEIRESIAS (Davis, 1976). Because such programs can reason about their own knowledge, they have the power to encode knowledge about strategies, e.g., when to use and when to ignore specific items of medical knowledge and which leads to

follow up on. Such meta-level knowledge offers a new dimension to the design of "intelligent assistant" programs, which we predict will be exploited in medical decision-making systems of the future.

3.9 Conclusions

This review has shown that there are two recurring questions regarding computer-based clinical decision making:

1. *Performance:* How can we design systems that reach better, more reliable decisions in a broad range of applications?
2. *Acceptability:* How can we more effectively encourage the use of such systems by physicians or other intended users?

We shall summarize these points separately by reviewing many of the issues common to all of the paradigms discussed in this chapter.

3.9.1 Performance Issues

Central to ensuring a program's adequate performance is a matching of the most appropriate technique with the problem domain. We have seen that the structured logic of clinical algorithms can be effectively applied to triage functions and other primary care problems but would be less naturally matched with complex tasks such as the diagnosis and management of acute renal failure. Good statistical data may support an effective Bayesian program in settings where diagnostic categories are small in number, nonoverlapping, and well defined, but the inability to use qualitative medical knowledge limits the effectiveness of the Bayesian approach in more difficult patient management or diagnostic environments. Similarly, mathematical models may support decision making in certain well-described fields in which observations are typically quantified and related by functional expressions. These examples, and others, demonstrate the need for thoughtful consideration of the technique most appropriate for managing a clinical problem. In general, the simplest effective methodology is to be preferred,[29] but acceptability issues must also be considered, as discussed below.

[29]It is also always appropriate to ask whether computer-based approaches are needed at all for a given decision-making task. For all but the most complex clinical algorithms, for example, the developers have tended to discard computer programs. Similarly, Schwartz et al. pointed out that the decision analyses can often be successfully accomplished in a qualitative manner using paper and pencil (Schwartz et al., 1973).

As researchers have ventured into more complex clinical domains, a number of difficult problems have tended to degrade the quality of performance of computer-based decision aids. Significant clinical problems require large knowledge bases that contain complex interrelationships including time and functional dependencies. The knowledge of such domains is inevitably open-ended and incomplete, so the knowledge base must be easily extensible. Not only does this require a flexible representation of knowledge, but it encourages the development of novel techniques for the acquisition and integration of new facts and judgments. Similarly, the inexactness of medical inference must somehow be represented and manipulated within effective consultation systems. As we have discussed, all these performance issues are important knowledge-engineering research problems for which artificial intelligence already offers promising new methods.

It is also important to consider the extent to which a program's "understanding" of its task domain will heighten its performance, particularly in settings where knowledge of the field tends to be highly judgmental and poorly quantified. We use the term *understanding* here to refer to a program's ability to reason about, as well as reason with, its medical knowledge base. This implies a substantial amount of judgmental or structural knowledge (in addition to data) contained within the program. Analyses of human clinical decision making (Elstein et al., 1978; Kassirer and Gorry, 1978) suggest that as decisions move from simple to complex, a physician's reasoning style becomes less algorithmic and more heuristic, with qualitative judgmental knowledge and the conditions for evoking it coming increasingly into play. Furthermore, the performance of complex decision aids will also be heightened by the representation and utilization of high-level meta-knowledge that permits programs to understand their own limitations and reasoning strategies. In order to design medical computing programs with these capabilities, the designers themselves will have to become cognizant of knowledge-engineering issues. It is especially important that they find effective ways to match the knowledge structures that they use to the complexity of the tasks their programs are designed to undertake.

3.9.2 Acceptability Issues

A recurring observation as one reviews the literature of computer-based medical decision making is that essentially none of the systems has been effectively utilized outside of a research environment, *even when its performance has been shown to be excellent!* This suggests that it is an error to concentrate research primarily on methods for improving the computer's decision-making performance when clinical impact depends on solving other problems of acceptance as well. There are some data (Startsman and Robinson, 1972) to support the extreme view that the biases of medical

personnel against computers are so strong that systems will inevitably be rejected, regardless of performance.[30] However, we are beginning to see examples of applications in which initial resistance to automated techniques has gradually been overcome through the incorporation of adequate system benefits (Watson, 1974).

Perhaps one of the most revealing lessons on this subject is an observation regarding the system of Mesel et al. described in the section on clinical algorithms (Mesel et al., 1976). Despite documented physician resistance to clinical algorithms in other settings (Grimm et al., 1975), the physicians in Mesel's study accepted the guidance of protocols for the management of chemotherapy in their cancer patients. It is likely that the key to acceptance in this instance is the fact that these physicians had previously had no choice but to refer their patients with cancer to the tertiary care center some distance away where all complex chemotherapy was administered. The introduction of the protocols permitted these physicians to undertake tasks *that they had previously been unable to do.* It simultaneously allowed maintenance of close doctor-patient relationships and helped the patients avoid frequent long trips to the center. The motivation for the physician to use the system is clear in this case. It is reminiscent of Rosati's assertion that physicians will first welcome computer decision aids when they become aware that colleagues who *are* using them have a clear advantage in their practice (Rosati et al., 1973).

A heightened awareness of human-engineering issues among medical computing researchers will also make computers more acceptable to physicians by making the program easier and more pleasant to use. Fox has recently reviewed this field in detail (Fox, 1977). The issues range from the mechanics of interaction at the computer (e.g., using display terminals with such features as light pens, special keyboards, color, and graphics) to the features of the program that make it appear as a helpful tool rather than a complicating burden. Also involved, from both the mechanical and global design sides, is the development of flexible interfaces that tailor the style of the interaction to the needs and desires of individual physicians.

Adequate attention must also be given to the severe time constraints perceived by physicians. Ideally, they would like programs to take no more time than they currently spend when accomplishing the same task on their own. Time and schedule pressures are similarly likely to explain the greater resistance to automation among interns and residents than among medical students or practicing physicians in Startsman's study (Startsman and Robinson, 1972).

The issue of a program's "self-knowledge" impacts on the acceptance of consultation systems in much the same way as it does on program performance. Decision makers, in general, and physicians, in particular, will place more trust in systems that appear to understand their own limitations

---

[30] *Ed. note*: More recent studies have shown marked improvement in attitudes in the past decade, however (Teach and Shortliffe, 1981).

and capabilities and that know when to admit ignorance of a problem area or inability to support any conclusion regarding an individual patient. Moreover, physicians will have a means for checking up on these automated assistants if the programs have an ability to explain not only the reasoning chain leading to their decisions but their problem-solving strategies as well. High-level knowledge, including a sense of scope and limitation, may thus allow a program to know enough about itself to prevent its own misuse. Furthermore, since systems that are not easily modifiable tend not to be accepted, meta-level knowledge about representation and interconnections within the knowledge base may help overcome the problem of programs becoming tied too closely to a store of knowledge that is regionally or temporally specific. It is therefore important to stress that considerations such as those we have mentioned here may argue in favor of using symbolic reasoning techniques even when a somewhat less complex approach might have been adequate for the decision task itself.

3.10 Summary

In summary, the trend toward increased use of knowledge-engineering techniques for clinical decision programs stems from the dual goals of improving the performance *and* increasing the acceptance of such systems. Both acceptability and performance issues must be considered from the outset in a system's design because they indicate the choice of methodology as much as the task domain itself does. As greater experience is gained with these techniques and as they become better known throughout the medical computing community, it is likely that we will see increasingly powerful unions between symbolic reasoning and the alternative paradigms we have discussed. One lesson to be drawn lies in the recognition that much basic research remains to be done in medical computing, and that the field is more than the application of established computing techniques to medical problems.

ACKNOWLEDGMENTS

We wish to thank R. Blum, L. Fagan, J. King, J. Kunz, H. Sox, and G. Wiederhold for their thoughtful advice in reviewing earlier drafts of this paper. We are also grateful to Dr. Herbert Sherman and the original reviewers for their constructive suggestions regarding revisions.

4

Artificial Intelligence Methods and Systems for Medical Consultation

Casimir A. Kulikowski

Shortly after the preceding review article appeared, Kulikowski published the following more detailed analysis of the knowledge-engineering approach. Focusing mostly on the medical AI systems of the 1970s, he considers the major problems that arise in designing a consultation program. These problems center about choosing diagnostic interpretation and treatment-planning strategies and the knowledge representations for formalizing them. In choosing a knowledge representation, Kulikowski notes that explanation and knowledge acquisition are just as important as efficient and effective performance (Shortliffe, 1982b). Indeed, these concerns are interrelated: justifying decisions and updating the knowledge base, as the system is built incrementally or new information becomes available, place a premium on the modularity of a representation and the ease with which its reasoning procedures can be explained.

In both diagnosis and treatment decisions, schemes for quantifying the uncertainty of inferences raise difficult issues of both an empirical and a formal logical nature (see also Chapter 9). In addition, many specific practical problems of system design arise. Achieving robust performance despite uncertain relationships is a crucial requirement; an important insight resulting from the design of several systems is that robust performance can largely be achieved by a rich network of deterministic relationships that interweave the space of hypotheses.

Kulikowski also discusses several knowledge-based representational schemes that generalize the results of the early consultation programs

 From *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-2: 464–476 (1980).

[EMYCIN (van Melle et al., 1981), EXPERT (Weiss and Kulikowski, 1979), AGE (Nii and Aiello, 1979)]. By providing an environment for encoding knowledge, editing the evolving knowledge base, and testing programs, these systems provide techniques and tools that promise to be very versatile in helping to design new medical expert systems.

While the earlier chapters in this volume provide motivation for applying artificial intelligence techniques to medicine, comparing the methods to those of traditional algorithmic programming and statistics, in this paper Kulikowski presents the knowledge-based perspective as a whole. This serves as a prelude to detailed discussions of particular consultation systems (Chapters 5, 6, 7, and 8) and to Szolovits and Pauker's analysis of medical reasoning in the context of these programs (Chapter 9).

4.1 Introduction

4.1.1 The Need for Computer-Based Medical Consultation

Expert medical consultation is a scarce, expensive, yet critical component of any health care system. Making the knowledge and expertise of human experts more widely available through computer consultation systems has been recognized as an important mechanism for improving the access to high-quality health care (Schoolman and Bernstein, 1978; Schwartz, 1970). The simulation of clinical cognition by the computer raises important scientific questions about the structure, consistency, completeness, and uncertainty of medical knowledge. These considerations are of particular interest to researchers in artificial intelligence (Minsky, 1968; Newell and Simon, 1972; Nilsson, 1980), cognitive psychology (Elstein, 1976), and medical science and education (Feinstein, 1967; Komaroff, 1979; Schoolman and Bernstein, 1978). These matters are also important if we are to assess the performance and understand the role of computer consultation systems in medical practice.

In a recent bibliography of automated medical decision-making methods and systems (Wagner et al., 1978) over 800 references are cited, and these do not include many of the simplest state-of-the-art applications or the most complex AI methods. If all of these are taken into account, it is likely that closer to 2,000 articles have been written describing medical decision-making and consultation systems. Yet the effect of automated decision making on medical practice after 20 years of fairly intense activity has not been very dramatic. There have been some notable successes, such as automated EKG interpretation, which is now routinely available, and a few institutions have on-line consultative decision capabilities, but on balance, remarkably few systems have gone beyond the prototype stage.

There are many reasons for the slow introduction of computer-based decision systems into medical practice. Some are social, some technological, yet ultimately there is a simple pragmatic reason: such systems have rarely been shown to fill an indispensable need in the clinical setting. This picture may be beginning to change: with the proliferation of new special-purpose biochemical tests and the accelerated specialization of medicine, the demand for easy reference to up-to-date consultative advice and medical information is beginning to be increasingly recognized. Medical data bases that pool information from national networks of collaborating researchers (Fries, 1976), record-keeping systems with capabilities for retrieving general medical information and references (Schultz and Davis, 1979), and computer-based medical instruction and testing systems have gradually grown and spread during the past decade. The National Library of Medicine has recently moved in the direction of supporting research into the structure and organization of medical knowledge bases and the methodologies by which they can be kept up to date and disseminated to practitioners (Schoolman and Bernstein, 1978). This complements the ongoing support programs of research and computing resources for artificial intelligence in medicine (AIM) by the Biotechnology Resources Program of the Division of Research Resources of the NIH (Ciesielski, 1978; Freiherr, 1979).

Another technological impetus for change can be expected to come from the increased availability of microprocessors, which will make inexpensive computing readily available to practitioners in their own offices. Many are already experimenting with methods of encoding their decision logic in the form of simple algorithms, and there has been a notable proliferation of small medical-computing groups and societies in the past few years that have served to focus these activities. The automated interpretation of laboratory instrument results, particularly in clinical pathology, is also becoming more prevalent (Bieman, 1979; Speicher, 1978; Young, 1976). It is likely to stimulate a need for more extensive clinical decision systems that will back up and integrate the results from several different instruments, ranging over various systems of the body. The scope of an AI model of internal medicine, such as that developed for INTERNIST (Pople et al., 1975), the modularity and explanatory capabilities of MYCIN (Shortliffe, 1976), and the pathophysiological reasoning and efficiency of compiled expert knowledge available in EXPERT (Weiss and Kulikowski, 1979) will all be useful for such tasks.

Not all work on consultation methods and systems needs to be ultimately justified in terms of their application in clinical practice. Contributing to help organize medical knowledge and research and supporting medical education are two other important fundamental objectives. The AI systems are particularly relevant in both these regards, since they have concentrated not only on achieving good performance, but on justifying and explaining this performance based on models of diseases and patients.

Three recent reviews of medical decision methods and systems have included the artificial intelligence approaches (see Schoolman and Bernstein, 1978; and Chapters 3 and 9). An article by Szolovits and Pauker (Chapter 9) describes the four earliest AIM systems, contrasting categorical (deterministic) with probabilistic components of their reasoning strategies. A review by Shortliffe, Buchanan, and Feigenbaum (Chapter 3) emphasizes the symbolic reasoning nature of the AI programs and highlights the importance of explanation and updating facilities, as well as good conversational capabilities for interacting with consultation programs; the authors draw mainly on their experience with the MYCIN system for illustrative examples. The present paper takes a somewhat different approach in that it suggests a set of characteristic representational, reasoning, and control features for describing consultation programs, and then uses these as the basis for its comparisons.

4.1.2 Goals and Approaches of Artificial Intelligence in Medicine

In reviewing artificial intelligence approaches to medical consultation, it is important to characterize the concerns and goals of AI research that have influenced the work in this field.

The spectrum of research in AI can be described as ranging between two extreme approaches. The first stresses the development of theories of cognition through computer-based experimentation. Michie (1974) has given a definition of AI consonant with this view: "the development of a systematic theory of intellectual processes." In contrast, a more pragmatic concern of imitating or approximating the behavior of human problem solvers is expressed in the definition given by Minsky (1968): "The science of making machines do things that would require intelligence if done by men." The first approach shares many concerns with cognitive psychology. The major aspect of computer programs from this viewpoint is that their reasoning procedures must exhibit capabilities of understanding that imitate those used by human problem solvers. At the other extreme, the correspondence with human behavior can be viewed strictly in terms of the output performance of a computer system, regardless of whether the reasoning leading up to this performance simulates that of humans. Much of the problem solving done in robotics takes this approach (Winston, 1972). In a similar vein, research in pattern recognition, developing from engineering and the mathematical disciplines (Duda and Hart, 1973; Fukunaga, 1972), has stressed the importance of achieving accurate performance in detecting and classifying patterns, usually by mathematical and statistical techniques that do not attempt to parallel human reasoning.

Despite the contrast between the performance-oriented and understanding-oriented work in AI in the past, it can be recognized that these

represent complementary approaches that are open to researchers seeking to develop computer-based problem-solving programs. Recent work in the automated recognition of human speech (Lesser et al., 1975; Lesser and Erman, 1979) exemplifies the maturity of AI in developing systems that are oriented to both understanding and performance.

Expert medical consultation is a problem-solving process that draws on a rich, though incomplete, body of knowledge that is both empirical and conceptual in nature. Until the introduction of artificial intelligence methods, the reasoning of computer-based consultation programs relied primarily on normative knowledge (prescribed as norms or rules of reasoning) that is used directly in medical decision making. The major emphases of the AI approaches have been:

1. to clearly separate the domain-specific knowledge base of a consultation model from the reasoning and control strategies used by the consultation programs (this facilitates *modification* of the knowledge base, which is likely to require frequent changes for incorporating new results from medical research and practice);
2. to capture the expert medical knowledge about specific inferences or decisions in the form of modular rules that reference the concepts and facts of the medical domain, also organized in a modular fashion (this facilitates the *explanation* of a consultation program's reasoning processes, which is crucial to the acceptability of a computer-based system);
3. to develop logically powerful and expressive representations for describing medical concepts and facts (such as disease hierarchies and mechanisms and the corresponding courses of illness) that serve to *support* and *justify* the decision rules in terms of knowledge structures that are commonly used by physicians;
4. to experiment with a variety of reasoning and evaluation methods and to develop general strategies to control the reasoning (this introduces *flexibility* and the ability to recover from mistakes through alternative means of reasoning, hence giving a *fail-soft* capability);
5. to develop methods of facilitating *user interaction* with the programs, either by specialized natural language interpretation capabilities or by flexible command languages.

By incorporating many of the attributes described above, computer consultation programs are beginning to display some of the scope, depth, and flexibility of reasoning that characterizes expert human consultants. At the same time, the process of building these systems is uncovering new problems in the representation, application, and validation of medical knowledge.

4.2 Decision-Making Problems and Styles in Medical Consultation

4.2.1 Medical Consultation Tasks

The tasks involved in medical consultation depend on the nature of the advice that is being sought from the consultant. Whether it is feasible to capture some of the reasoning and problem-solving processes employed by a consultant within computer programs depends largely on the relative role of reasoning versus perceptual skills used by the human expert. If expertise in performing a specialized physical examination, involving the detection of subtle signs through visual, tactile, and other sensory cues, constitutes a crucial element of the expert's consultation, it is not reasonable to expect any current computer system to perform such a consultation. [Nevertheless, computer-based systems may provide valuable new modes of extracting perceptual information on the patient, such as by tomography (Kak, 1979).] If, on the contrary, the scope and definition of items in a review of systems and the elicitation of a medical history have been well determined for a given diagnostic or treatment selection problem and the major role of the human consultant is to provide a sophisticated interpretation of the findings, then it is not unreasonable to investigate such processes of interpretation and attempt to simulate their performance by computer-based systems. If, in addition, it is possible to build a knowledge base that incorporates both descriptive models of pathophysiological mechanisms as well as the normative components of expert reasoning, and if strategies of explanation can be formulated that permit the program to answer questions about its own reasoning, then it is not unreasonable to claim that such a system demonstrates certain elements of "understanding" not unlike those manifested by human problem solvers.

The major tasks of medical consultation that must be performed by a computer system can be summarized as follows:

1. the sequential elicitation of findings and the assessment of their reliability and internal consistency;
2. the interpretation of the findings in terms of a model of diagnostic classes and their relationships;
3. the extrapolation of the natural course that the illness is likely to follow (prognosis);
4. the formulation of various plans for therapeutic management and the selection of an initial treatment;
5. the explanation and justification of the above;

6. the reassessment of the patient's status on return visits and the reevaluation and possible modification of diagnostic, prognostic, and therapeutic conclusions.

At any given point in the course of a consultation, one of the tasks described above will be the main goal of the reasoning of the human or computer-based consultant. In a generalized consultation scheme these goals and their various subgoals (such as eliciting a specific finding or formulating a specific treatment for a given disease) must be explicitly represented if their sequencing is to be easily modifiable by the control strategy just as it is by the human consultant's strategy decisions. The principal types of medical facts and concepts and some of the reasoning links among them are shown in Figure 4-1.

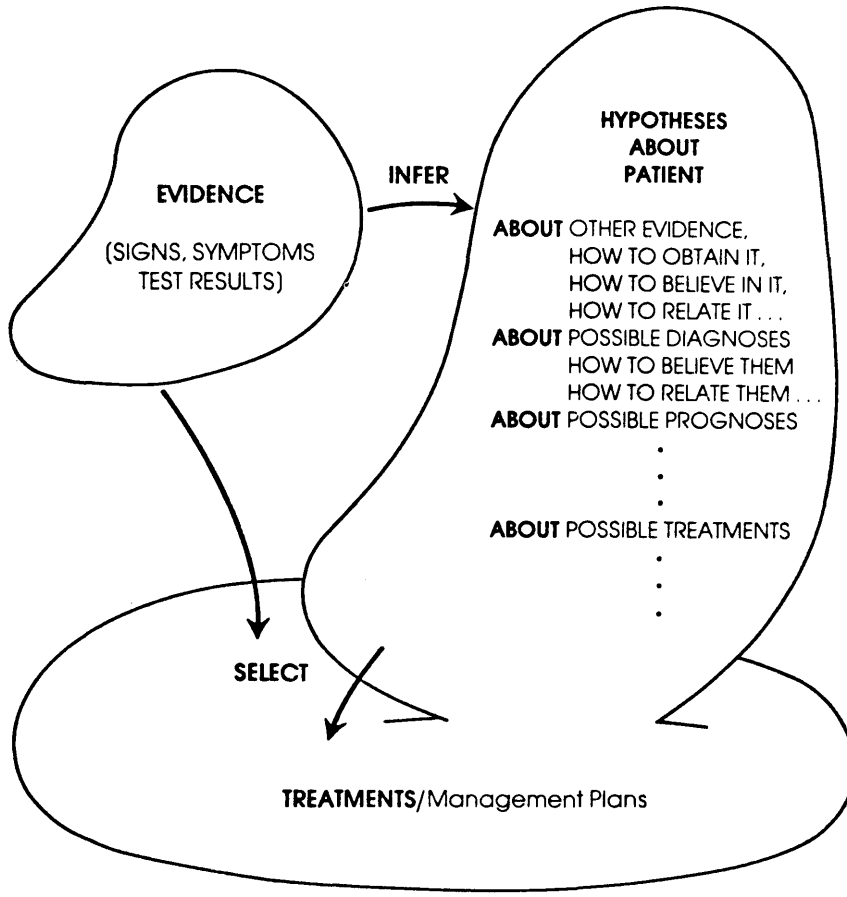


FIGURE 4-1 Problem solving in medical consultation.

The medical facts about an individual patient (findings) can be viewed as the direct *evidence* from which hypotheses about possible diagnoses, prognoses, and treatments are generated and tested. This evidence comprises the history and symptoms reported by the patient, the signs elicited by the physician during the course of an examination, and the results of specialized tests for detecting specific pathophysiological states or conditions. A data structure used for describing a finding can include details about its measurement technique, its range of values, its reliability, its timing, its cost, and its logical relation to other measurements. It will be associated by various relational links and rules of reasoning to the hypotheses.

Hypotheses usually require a very different descriptive structure. They stand for the major medical concepts used in reasoning, such as the diagnostic and prognostic categories applicable to the patient, but may also include a variety of intermediate constructs, such as syndromes, pathophysiological and clinical states, courses of illness, and clusters of clinical evidence. These intermediate concepts can be used to define the higher-level concepts. Although the major type of hypothesis is one that refers directly to the clinical condition of the patient, it is also possible that we may want to explicitly represent hypotheses that are assertions about related contexts (such as the environment of the patient, a relative of the patient, etc.). Some hypotheses may be subconcepts of others, in which case they may inherit properties of the parent concept; others may be causal antecedents, which implies that they must also occur in temporal sequence before their consequents.

A consultation system must also represent the various treatments that are potentially able to control the patient's illness. The treatments are interrelated in terms of applicability and risk/benefit factors: therapeutic effectiveness, toxicity, potential for undesirable interactions, and other constraints. To manage a patient with a complex or prolonged illness, a management plan must be formulated. The plan must consist of the various potential sequences of treatments that are available to control the alternative courses that may be followed by the illness after an initial treatment. In computer-based consultation schemes, it is important to represent a realistically large scope of alternatives and their relations to the hypotheses and findings of patients. On the basis of these relationships, rules for selecting treatments can be derived and explained.

A significant component of human consultative reasoning is often characterized as being judgmental. In designing computer-based systems, an immediate question arises as to how best to simulate such judgments, if indeed they are to be simulated at all. One school of thought holds that it would be best if they could be replaced by more objective methods, usually of a statistical decision-theory type (Grémy, 1976). But even with this approach, judgmental knowledge is needed to choose decision thresholds. Others have attempted to capture the expertise of human consultants in the form of reasoning rules that directly incorporate judgmental ele-

ments (Shortliffe and Buchanan, 1975). Regardless of the approach, the relative value of alternative reasoning outcomes (misdiagnoses, inadequate treatments, etc.) clearly enters into consultative reasoning. Thus computer-based schemes must include a representation of these values (also called *utilities*) to be used by their decision strategies. The exact manner in which such values are to be used depends on the structure of knowledge in the program, the overall strategies of reasoning, and the nature of the values involved. Values on outcomes will be very different if they are those of the patient rather than of the physician, and both will differ from any "average" or societal values for comparing outcomes. Pauker (1978) has recently discussed these problems from a decision-theory viewpoint. In addition, different experts may well disagree on how to treat a given patient, each giving a justification for his or her point of view. Such sources of variability ensure that in most situations there will be no single "correct" or "optimal" mode of treating a patient, and the role of a consultation system must be seen as one of presenting the alternatives, with a clear indication of the source for the value judgments that enter into each decision.

4.2.2 Types of Medical Consultation

The kinds of reasoning involved in medical consultation depend on the specific type of problem presented to the consultant. In the past, computer-aided methods have been used in the following consultative situations:

1. interpreting a single test and listing possible diagnoses;
2. screening the patient for a particular disease (or group of related diseases) from multiple tests and clinical findings;
3. performing some of the tasks of a primary care physician in acquiring information on the present illness of the patient, proceeding to a differential diagnosis, and making treatment recommendations if appropriate;
4. simulating the role of a specialist who is asked to provide interpretation and management suggestions for complex cases referred by primary care physicians.

The artificial intelligence consultation programs developed to date have simulated the last two types of consultation. They have been research prototypes, and it is not unreasonable to expect that if programs of this type are to become widely used in clinical practice, connections between them and the more basic types of single-test and screening programs will have to be developed.

4.2.3 Evolution of Formal Methods of Decision Making in Consultation

The applications of formal methods of decision making have concentrated on problems of diagnostic reasoning, though decision-analysis techniques have been applied to treatment-selection problems. The sequence in which different techniques have been introduced is approximately as follows:

mid-1940s: Statistical hypothesis-testing methods [mostly for screening and radiology (Yerushalmy, 1947); computations by calculator]

1954: Logical scheme for matching symptoms to diagnoses [slide rule (Nash, 1954) or hollerith cards used for sorting and matching]

1958: Statistical and logical techniques combined (Lipkin and Hardy, 1958) [computers introduced and used in most subsequent work]

1960: Bayesian and discriminant methods (Ledley and Lusted, 1959)

1968: Sequential Bayesian methods, and decision-theory approaches applied to treatment selection (McNeil et al., 1975; Schwartz et al., 1973) (also see Chapter 2)

1969: Pattern-recognition methods (Kulikowski, 1970; Patrick et al., 1977)

1970: Information-processing models for diagnosis (Wortman, 1972)

1971: Knowledge-based artificial intelligence systems (Kulikowski and Weiss, 1972; Pople et al., 1975; Shortliffe, 1976; see also Chapter 6)

Ledley and Lusted (1959) gave the first overview of the applicable methods from logic and probability, and the 1960s saw the introduction of various statistical, logical, and pattern-recognition techniques for diagnostic decision making. These methods, relying on large data bases of reliably diagnosed case histories, performed well in narrowly defined medical domains using a clearly specified (or standardized) set of patient findings. Lack of adequate statistics and problems of consistently introducing value judgments about possible misdiagnoses into the decision framework have proven to be important limitations of these methods.

A very different manner of encoding medical reasoning in a computer program has also been available: the sequence of decisions performed by a physician in reaching a diagnosis or choosing a treatment can be flowcharted and directly implemented as an algorithm. But insofar as the same conclusions may be reached by many different pathways and it is quite usual for experts to differ in their preferred sequences of tests and intermediate decisions for a given type of case, such a *flow chart algorithm approach* is usually too rigid and idiosyncratic to be widely accepted. However, characterizing the reasoning of an expert in a specialty can be useful for

teaching, for comparison with medical practice, and for guiding the decisions of physicians' assistants (Komaroff et al., 1974). Simple decision algorithms for patient self-help have been proposed recently as a technique of preventive medicine (Vickery and Fries, 1978), which may also reduce the burden on health care facilities. A mixed algorithm scheme is characteristic of one of the best-known consultation programs—Bleich's system for acid-base and electrolyte balance (Bleich, 1969). It intermingles the direct logical assessment of patient findings with calculations from mathematical formulas that describe the underlying biochemical changes.

To provide information about past experiences with prognosis and treatment, several different groups have relied on the *logical matching* of current patient profiles to prior stored cases in a large data base. The ARAMIS system in rheumatology at Stanford University (Fries, 1972; 1976) and similar ones in lung cancer at Yale University (Feinstein et al., 1972) and cardiovascular diseases at Duke University (Rosati et al., 1975) are well-known examples. The major methodological question for these systems is the form in which patient profiles are to be specified and the choice of query types that can be easily supported by the data base structure. Although they have not addressed the problems of how to incorporate their results into the broader interpretation of a patient's condition, they represent an important step in the direction of standardizing knowledge about the time course of diseases within a data base. And insofar as all interpretation is left to the physician using the system, they have been more readily accepted than many of the consultation programs.

In the late 1960s and early 1970s various *pattern-recognition methods* began to be applied to medical decision making (Kulikowski, 1970; Patrick et al., 1977). In some instances they provided the means of overcoming the limitations of small-sized statistical samples through the use of well-chosen heuristics; in others they enabled the summarization of large numbers of findings through synthetic "features" (Kulikowski, 1970), but in common with the statistical approaches, they suffered from being a "black box" approach to medical reasoning. That is, the patient's findings would be transformed mathematically into some heuristic score or weight, which would then become the sole basis for ranking diagnoses or treatment recommendations.

Figure 4-2 shows a schematic diagram of a typical pattern-recognition or statistical system for medical consultation. The sequence of operations specified by algorithm typically consists of a preprocessing, or filtering, to extract the set of patient findings relevant to the clinical problem under consideration, and the extraction of features (logical or mathematical transformations) that when selected for best discriminatory performance enable the classifier to be both simple and effective. The domain-specific knowledge base used by the algorithm is composed of various patterns of association between findings and hypotheses (for statistical methods), profiles of correctly diagnosed cases (for nonparametric sample-based methods, such as the nearest-neighbor technique), or explicit sequences of decisions (for the flowcharting methods). Most programs implementing these meth-

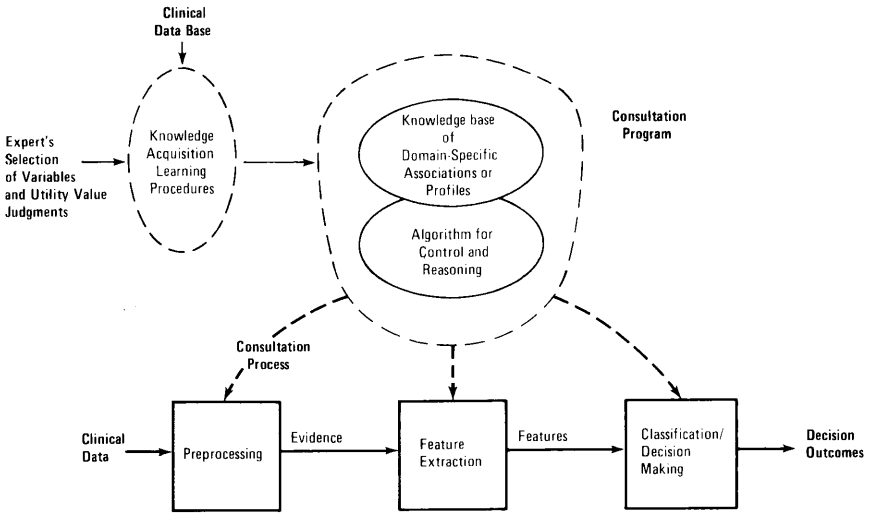


FIGURE 4-2 Statistical or pattern-recognition system for consultation.

ods intermingle elements of domain knowledge and reasoning mechanisms under algorithm control in a relatively fixed manner. The outcome, rather than the process of reasoning, is the main concern, so considerations of computational efficiency often override the possibility of introducing more flexible or general modes of reasoning that would come closer to imitating human expert behavior.

In designing such a system, the knowledge acquisition phase usually consists of analyzing the data base of clinical cases that have well-established diagnostic and treatment endpoints. The decision rules to be used by the classifier can be "learned" by various techniques (Chilanski et al., 1976; Duda and Hart, 1973; Fukunaga, 1972). The medical expert defines the scope of the problem by specifying the variables that are to be examined in the data base. If a decision-analysis method is to be used, the expert must also provide the utility or cost factors (and prior probabilities of hypotheses for subjectively estimated situations) to be used as part of the decision rule thresholds (McNeil et al., 1975).

The application of *artificial intelligence* methods sought to remedy the "black box" situation by introducing a structure of knowledge familiar to the physician into the decision-making schemes. The approach of using a computer-based model to study the decision making of clinicians was begun by researchers interested in cognitive processes. Kleinmuntz and McLean (1968) developed a program for simulating a consultation session in neurology, and Wortman (1972) developed an information-processing model for medical reasoning (including conceptual hierarchies and memory mechanisms). Initial prototype consultation programs using AI con-

cepts were developed in ophthalmology [CASNET (Weiss et al., 1978)], infectious diseases [MYCIN (Shortliffe, 1976)], internal medicine [INTERNIST (Pople et al., 1975)], and renal disease [PIP (Pauker et al., 1976)], while an article by Gorry (see Chapter 2) advocated the introduction of conceptual structures, language development, and explanation into medical decision-making systems.

All of the AI approaches use heuristic measures for scoring the weight of confidence or credibility that they assign to a hypothesis as an explanation of the patient's condition. These measures are typically computed from uncertainty weights attached by the human experts to the various reasoning rules in the consultation model. The reasoning strategies of all of the systems, however, rely as much on the structure of connectivities among concepts and between concepts and facts as on the scoring mechanisms themselves. This provides the systems with a natural way of supporting explanations, and often allows alternative and sometimes redundant lines of reasoning to be pursued, giving a measure of flexibility to their behavior.

Contemporary with the evolution of the AI approaches, several other investigators have introduced constraints and intermediate reasoning constructs into probabilistic frameworks. These include Bayesian approaches (Patrick, 1977; Warner, 1978) and a latent factor method (Woodbury and Clive, 1980). Fuzzy logic has also been applied to diagnostic problems (Wechsler, 1976).

The subsequent sections review the early AI systems and trace the evolution of the knowledge-based schemes that have been developed to the present.

4.3 Artificial Intelligence Methods in Consultation

4.3.1 A Comparative Overview of Early AI Consultation Systems

In this section we discuss the first major AIM systems—CASNET, MYCIN, INTERNIST, and PIP, each of which is described in greater detail in later chapters.

CASNET/Glaucoma Consultation System

A causal-associational network (CASNET) was developed as a means of representing the pathogenesis of a disease, in terms of which the patient's findings are interpreted. The causal relations, with associated degrees of

strength, express not only the mechanisms of a disease but also their modifications under various regimens of treatment. Different patterns over the causal network are associated with the various elements in a classification scheme of diagnostic hypotheses, which can include degrees of severity and progression of a disease. Appropriate treatment plans can be associated with the diagnostic hypotheses, and specific treatments within the plans are related to each other by constraints of how they cover for particular illnesses, how they may interact, etc. Normative knowledge is in the form of inferential rules linking patient findings to the intermediate hypotheses about pathophysiological states and preference rules linking findings to treatments. Uncertainty measures on these links range from +1 for full confirmation to −1 for full disconfirmation.

The reasoning control strategy of CASNET can be characterized as mainly event-driven: the incoming clinical data trigger the inference rules that assign weights to the pathophysiological states. A thresholding evaluation mechanism then yields a logical status of "confirmed," "disconfirmed," or "undetermined" to each causal state. The subgraph of confirmed and undetermined states forms a *patient-specific interpretation model* at every stage of the consultation. The system uses the causal model to constrain the search for possible hypotheses by guiding the requests for further patient data. This is carried out by first propagating direct and inverse causal weights throughout the net every time a data item is entered. Such a global assessment is made efficient by the partially ordered and precompiled nature of the causal net. Once the weights are computed, the choice of next question is hypothesis-driven: a criterion of maximal diagnostic information for a given cost range guides the selection that will add to the weight of evidence of the most likely intermediate hypothesis (state). This strategy may be superseded by domain-specific strategies for data acquisition, which can encode prespecified protocols given by experts; this was the case in the specialized CASNET/Glaucoma system. When all the data having a bearing on the consultation have been accumulated, the system carries out a final evaluation over the entire causal net, producing a weighting of the root nodes (primary causes). These trigger the higher-level diagnostic, prognostic, and treatment categories in a purely deterministic fashion. The choice of specific treatment, including the dosage, mode of administration, and time course, is then carried out by evaluation over the preference rules. These contain the various restrictions on the applicability of treatments, such as allergies, past history of treatment effectiveness, drug interactions, and so on.

A knowledge-base acquisition program for building CASNET-type models was developed at Rutgers University (see Chapter 20), and an in-depth model for consultation in the glaucomas was built incorporating the knowledge of clinical experts from five major ophthalmology research centers. The consultation model was tested with many cases of disease (from the U.S. and Japan) and participated in a national symposium on glaucoma, performing at an expert level (Lichter and Anderson, 1977).

MYCIN/Infectious Disease Therapy Consultant

A system of production rules with associated uncertainty weights serves to capture most of the expert knowledge in MYCIN (Shortliffe et al., 1973; Shortliffe, 1976). Rules are of the following form: IF premise assertions are true, THEN consequent assertions are true with confidence weight X. The assertions can be Boolean combinations of clauses, each of which consists of a predicate statement about an <attribute, object, value> triple. The triples represent medical facts and hypotheses about the patient and related objects or contexts, such as infections, cultures, and organisms. For example, <GRAMSTAIN, E.COLI, GRAMNEG> stands for "the gram stain of the *E. coli* organism is gram-negative." Goals and subgoals of the consultation process, such as "select therapies to cover for all diagnosed infections," can also be explicitly represented by the predicate structure of an assertion.

The uniformity of representation for both domain-specific inferences and reasoning goals makes it possible for MYCIN to use a very general and simple control strategy: a goal-directed backward chaining of rules. In this approach, the first rule to be evaluated is one containing the highest-level goal—to select treatments for all the infections of the patient. This requires that the infections be known. But since they are usually unknown, the system must then try to satisfy subgoals that will allow the infections to be inferred. Discovering the results of cultures or other clinical parameters of the patient would be the most direct subgoals. These in turn may be deduced from other rules, but eventually the attempt to satisfy rule premises will end with assertions that can only be confirmed by directly questioning the user for the appropriate information. Once this happens, the system can begin to reason deductively by successively satisfying subgoals that it had previously unwound. A hierarchical tree of contexts (patient-infections-cultures-organisms) anchors and constrains the order in which the rules are invoked. This, together with a network of links among clinical parameter values and the templates for the parameters, constitutes the descriptive component of the MYCIN knowledge base.

The reasoning evaluation mechanisms include a fuzzy logic function for combining the effect of uncertain assertions within a rule (a minimum for conjunctive and a maximum for disjunctive combinations) and a heuristic cumulative function to add the confidence weights from rules with different sources of evidence in their premises (Shortliffe and Buchanan, 1975). The confidence weights (or factors) are expressed on a scale from -1 for complete disbelief in an assertion to $+1$ for complete belief. Separate measures of belief and disbelief are used in updating hypothesis weights, because of the need to avoid the probabilistic constraint that an assignment of probability P to a hypothesis implies a probability of $1-P$ for its negation. Shortliffe developed his scheme of confidence factors to provide physicians with a means of expressing their belief or disbelief in a hypothesis independently of one another. Although the MYCIN reason-

ing strategy is almost entirely based on the rule evaluation procedures, the final selection of therapy is carried out by a specialized algorithm, which uses the deduced knowledge of the patient's infections, the causative organisms, and the ranking of drugs by sensitivities and preference categories (of effectiveness).

The MYCIN system places special emphasis on the modular nature of its knowledge and on the ease that this modularity entails for generating explanations. A question-answering program interacts with the performance program to find out about the reasoning sequences leading to a given conclusion and the reasons behind the latter's requests for patient data. The user interface of the system has been developed with careful attention to its "friendliness" and the capability to express its rules in English. The system is able to understand a domain-specific vocabulary of commands and descriptions of patient-related facts, using a keyword-recognition scheme. Various Interlisp facilities are used to advantage in giving the system a good "conversational style." There have been formal evaluations of the MYCIN system by a number of independent consultants that demonstrated that the program performed at a level comparable to that of experts (Yu et al., 1979a; 1979b).

INTERNIST/Diagnostic Consultant in Internal Medicine

One of the principal aims of INTERNIST system development has been to explore the manner in which expert clinicians reason about diagnosis when the space of possibilities is large and hierarchically structured, as in internal medicine (Pople, 1975; 1977; Pople et al., 1975). The program uses a knowledge base in the form of a hierarchy of diseases, from the general (liver disease, heart disease, etc.) to the specific (hepatocellular infection, aortic stenosis, etc.), with the typical findings linked to the most specific form of each disease group. Other links include finding-to-disease evocation and disease-to-disease causal connections. A cost-related specification (history questions, signs, or the more expensive tests) and global weights of import are attached to the findings. There are uncertainty weights associated with most of the links, expressed on a scale that ranges from 1 (for least confirmation) to 5 (for maximum confirmation). The weights are subjectively estimated by the medical expert.

The reasoning strategy of INTERNIST begins in an event-driven fashion: the initial data presented to the system evoke a set of related disease hypotheses. For each of the evoked diseases, the system builds a patient-specific model, consisting of four lists: observed findings consistent with the disease, those unexplained by the disease, findings as yet unobserved that would be consistent with the disease, and those that ought to be observed if the disease is the correct diagnosis. Each disease model is scored positively for explained findings and negatively for the unexplained ones, with the individual findings weighted according to their importance.

Bonuses are added to hypotheses that are linked causally to other confirmed diseases. A partitioning heuristic then splits the space of hypotheses into those that compete and those that complement the most highly ranked one. For example, if thyroid carcinoma is found to be the most likely disease from the first evaluation, diseases like a thyroid cyst would be competing hypotheses, whereas a heart disease would be complementary in that it accounts for other findings largely unrelated to the thyroid problem.

Once the partitioning is completed, a number of different strategies may be pursued by the system, depending on the size of the competing hypothesis set. If there are more than four competitors, the system will try to rule them out by asking questions about the findings that are expected to be present in the disease. If the number of alternatives ranges from two to four, a discriminatory strategy is followed that consists of seeking results that are strongly indicated by one disease but only weakly indicated by the other. Finally, if there are no competitors, the strategy will ask for data that will strongly confirm the highest-ranking hypothesis. When this process has been completed by the confirmation of the first major disease (or one of its competitors), the program repeats the cycle with the next most highly ranked hypothesis in order to account for findings that remain unexplained. This process continues until all findings have been accounted for. The reasoning of INTERNIST is therefore strongly focused around the highly ranked hypotheses once the initial phase of data entry is completed.

The INTERNIST system has been reported to cover a large proportion of the field of internal medicine and is routinely tested with complex cases from clinical-pathological conference case reports in the major medical journals (Pople, 1977). Once its knowledge base has been expanded sufficiently, it is expected to be tested outside the University of Pittsburgh in a formal manner. The system is also being used for educational purposes, and it is expected to be linked to other diagnostic systems (Freiherr, 1979).

PIP/ Present Illness Program

To develop an understanding of the problem-solving methods used by physicians for patients who present with a varied and potentially large set of complaints was the underlying motivation of the project in clinical cognition (see Chapter 6). The system, developed at M.I.T. and Tufts–New England Medical Center, evolved from Gorry's proposal to introduce conceptual structure to guide and support reasoning in diagnosis (see Chapter 2). The representation chosen for the system was the frame scheme developed by Minsky (1975). A *frame* is a prototypical description, which in PIP is centered around disease categories. Each frame is a structure with a name and a number of slots, which can be filled by various properties, logical and semantic relations, and associated inference rules. The disease

frames in PIP contain slots for descriptive relations (causal, complementary, complicational, etc.), logical conditions (necessary and sufficient findings), and reasoning rules of various types (suggestive, discriminatory, or conclusive rules). The most important slots are those containing a listing of evocative or triggering findings, and a listing of expected findings. Like CASNET and INTERNIST, PIP initiates its reasoning in an event-driven fashion: the initial data trigger a number of hypotheses, which are then considered to be "activated." PIP maintains a three-level status for its hypotheses during a consultation. All start out in long-term memory, with inactive status. Once a hypothesis is activated, it brings along all hypotheses that are directly complementary to it into "semiactive" status. A semiactive hypothesis is eligible to become active if any one of its typical findings is found to be true, whereas "inactive" hypotheses can only be activated by their triggering findings.

Once the reasoning process begins by triggering, the system attempts to "fill in the frame" by asking questions that will tend to confirm it or rule it out. This may be done categorically by matching findings that are logically sufficient or necessary (MUST-HAVE or MUST-NOT-HAVE relations) or probabilistically by thresholding a local score evaluated for the hypothesis. This score is computed from the uncertainty rules associated with the frames and has two components: a measure of the fit of observed-to-expected findings for the hypothesis and a ratio of the number of findings explained by the hypothesis to the total number of observed findings. PIP also propagates scores so that the effect of findings that are explained by lower-level hypotheses—the clinical or pathophysiological states, such as "nephrotic syndrome"—can be taken into account in the likelihood computations of hierarchically or causally related hypotheses (such as "glomerulonephritis"). The sequential questioning of the system is therefore hypothesis-directed in that the filling of a frame results in asking about its expected findings or those that will discriminate it from other hypotheses. Focus is shifted to other frames once the truth value of the original one has been established with a sufficiently high level of certainty. The process continues until all reported findings have been accounted for.

PIP was an experimental system, and it was tested with a knowledge base of about 70 hypothesis frames in renal disease and related disorders (see Chapter 9). Problems were uncovered in maintaining a sufficiently focused and clinically acceptable line of reasoning, and this contributed to a shift in emphasis toward more tightly structured and physiologically determined domains (acid-base balance and digitalis therapy) on the part of its developers (Gorry et al., 1978; Patil, 1979; Silverman, 1975). It has been suggested that one major reason for the difficulty of generating lines of reasoning that parallel those of clinical experts lies in the use of generalized scoring functions and in termination criteria that lead to exhaustive explanations of the observed findings (Szolovits and Pauker, 1979) (also see Chapter 9). When several top-ranking hypotheses have scores that are close in value, reflecting a very ambiguous case, the interpretation of additional

data may often result in rapid changes in the focus of the reasoning, as one piece of evidence pushes the score of one hypothesis above that of its competitors, and then another finding elevates the score of an alternative hypothesis above that of the first. To avoid an overdependence on scoring functions, all AIM systems have tried to incorporate into their knowledge bases as many categorical reasoning links as possible.

4.3.2 Characteristic Elements of AIM Consultation Systems

The four initial AIM systems and their successors all share certain characteristic properties. Figure 4-3 illustrates some of the principal components of the systems and the resulting consultation process.

In contrast to the pattern-recognition and statistical approaches, there is a deliberate separation of the domain-specific knowledge base, the general mechanisms of evaluation, and the control strategies of the system. The reasoning evaluation and control components are sometimes called the *inference engine* (Davis, 1979; Feigenbaum, 1977). The knowledge base is often also clearly divided between a *descriptive component* of data structures linked by domain-specific relations (hierarchical categorizations, subcomponent membership, causal precedence or antecedence, etc.) and a *normative component* of prescriptive reasoning rules that operates over the descriptive component using the evaluation mechanisms in a manner specified by the control strategies. This organization can be viewed as a specialized variant of the structure used in generalized production systems in AI (Newell and Simon, 1972; Nilsson, 1980).

In CASNET, INTERNIST, and PIP the reasoning process is centered around an explicit, structural descriptive component. The causal nets and hierarchical taxonomies can be viewed as special cases of *semantic networks* (Quillian, 1968), which were the first and most widely used means of representing knowledge for natural language interpretation. The *frame* (or unit) schemes offer a very natural alternative way of representing knowledge, which emphasizes the "chunking" or partitioning used by human experts to separate different topics, concepts, or hypotheses. The normative or reasoning knowledge in these systems is expressed as decision rules or procedures attached to the nodes of the semantic net, or as logical constraint conditions contained in the frames.

In contrast, MYCIN centers its knowledge around the normative component: the *production rules*. Its descriptive component is deemphasized, although the context tree and network for updating values of clinical parameters are crucial to the effective invocation of rules. This approach may facilitate the acquisition of the strictly inferential knowledge, but leaves open the question of how to relate the specific productions to prototypical concepts in the medical domain. The context tree does this, but in a very specific and understated manner. It has been suggested that the operation

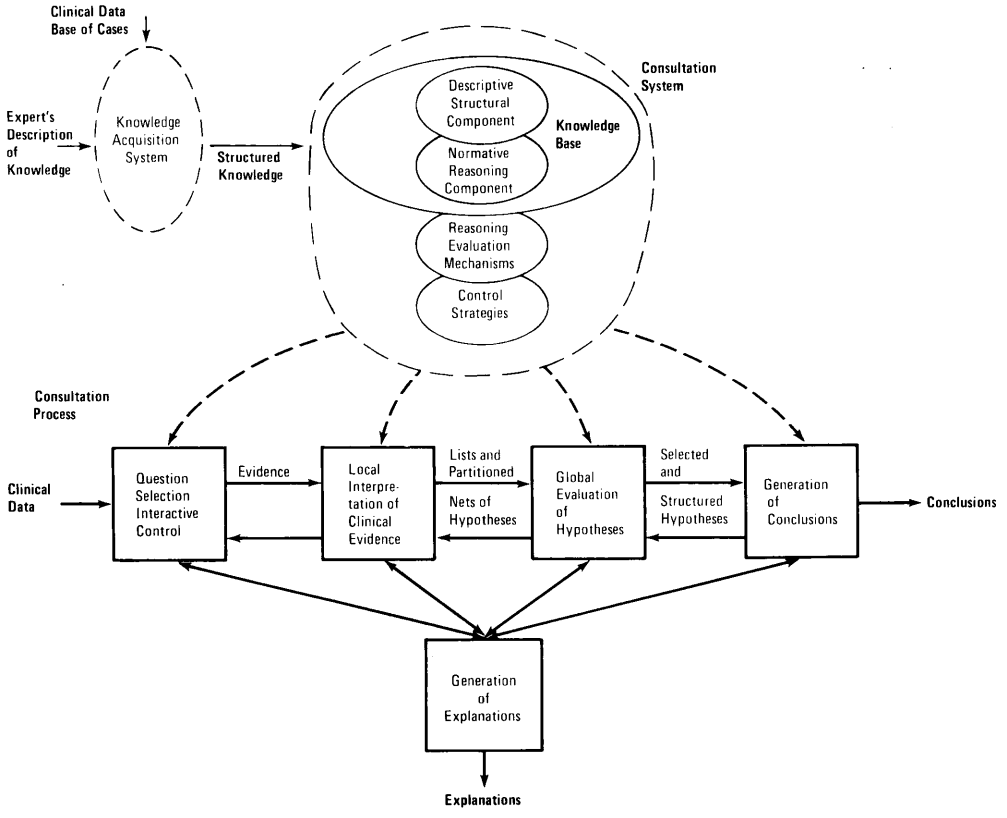


FIGURE 4-3 Artificial intelligence system for medical consultation.

of MYCIN could be turned "inside out" (see Chapter 9), with the contexts represented by frames, which will be filled up as the production rules that are attached to them are evaluated. The recent implementation of a mixed frame-and-production-rule representation [the CENTAUR system (Aikins, 1979; 1983)] has shown this to be a feasible approach.

Methods for quantifying uncertainty vary from system to system but share certain common properties: they treat confirmation and disconfirmation of hypotheses as independent processes (although combining functions are needed to produce measures of overall confidence for guiding the course of reasoning); the number of distinct uncertainty levels subjectively estimated by the experts is usually five or six; and they use fuzzy logic combining functions for evaluating the uncertainty of a Boolean combination of assertions.

Depending on the complexity of the consultation task, reasoning mechanisms may include: focus-of-attention heuristics to concentrate on a subspace of the space of possible hypotheses; pattern-matching mechanisms to actively scan incoming data for patterns that will trigger a hypothesis; goal generators to specify how sequences of subgoals ought to be pursued; global evaluation heuristics to piece together the results of several partial interpretations; and explanation mechanisms for tracing the reasoning. The control strategies specify how the different reasoning mechanisms are to be invoked, either automatically or in response to interactive commands given by the user.

The characteristic flow of information illustrated in Figure 4-3 shows that after an initial set of clinical data has been presented to the program, the control strategies can lead it to generate local interpretations (such as deciding on the normality, abnormality, or consistency of findings, or their interpretation in terms of directly related hypotheses), request more data as suggested by the initial interpretation, proceed to a global interpretation over the entire knowledge base (evaluating and comparing the partial interpretations, and selecting the most likely and coherently structured groupings of hypotheses), generate conclusions (integrating the various hypotheses into a final statement), and produce explanations for any of the preceding stages. The ability to recycle through previous stages of reasoning, allowing the user to request explanations and possibly changing the focus of reasoning by selectively introducing new data, introduces a significant degree of flexibility and generality that is characteristic of the AI approaches. It is interesting, however, that those consultation systems that give advice on treatment have done so without resorting to general methods of planning (Sacerdoti, 1977). This may reflect the fact that many treatment plans in medicine are short in length and center around the control of a limited number of clinical or physiological variables, making it possible to use relatively simple strategies of selection over prespecified alternative plans.

In building an AI consultation system, we rely more heavily on the knowledge of medical experts than in building probabilistic or pattern-recognition systems. The variety of structures employed by experts results in a much more complex knowledge acquisition process than must be faced by designers of the traditional systems, and a considerable effort has been devoted to these problems by subsequent AI system developers.

4.4 Evolution of AIM Systems and Knowledge Engineering

While the initial AIM systems were still evolving, several other systems were designed, taking advantage of the experiences and results obtained in the first cycle of development. The Digitalis Therapy Advisor (Gorry et al., 1978; Silverman, 1975) combined a single-compartment mathematical model for the effects of digitalis treatment with symbolic reasoning methods for the interpretation of patient-specific findings. After arriving at an initial determination of digitalis dosage based on the mathematical model, the system uses feedback information about the patient's clinical response to the dose (including both quantitative aspects, such as serum digoxin level, and qualitative cardiac signs and symptoms) to modify its recommendations for subsequent digitalis levels. The system was subjected to careful formal evaluation (Gorry et al., 1978), which demonstrated that its recommendations were comparable in effect to those of the clinical experts, suggesting that the system might be useful in health care situations where expert cardiac consultation is unavailable.

A generalization of the CASNET representational structures was included in the IRIS system, which used a semantic net to represent the descriptive knowledge of disease processes, reasoning primitives, and control states (Trigoboff and Kulikowski, 1977). IRIS was designed as a tool for experimenting with different reasoning and control strategies, rather than as a complete consultation system. It provided the user with a general mechanism for instantiating domain-specific facts and hypotheses and a mechanism for propagating inferences between them based on production rules. Specific control strategies could be written in Interlisp making use of the knowledge-base structure and reasoning elements of IRIS. Parts of the control strategies of MYCIN, INTERNIST, PIP, and CASNET were easily emulated in this manner. The MEDICO system, also applied in ophthalmology, used semantic and inference networks for knowledge acquisition (Walser and McCormick, 1976) and the design of a consultation system.

The PROSPECTOR system similarly combined the modularity of a rule-based scheme [using subjective Bayesian inferencing (Duda et al.,

1976) rather than the confidence-weight method of MYCIN] with a semantic network representation (Hart and Duda, 1977). Although this is a mineral exploration consultant rather than a medical consultant, PROSPECTOR is important in that it introduced the concept of a *partitioned semantic net* (Hendrix, 1975) to facilitate the attachment of rules to the appropriate set of semantic categories.

The facilitation of knowledge acquisition from experts and the updating of MYCIN-type models were the goals of the TEIRESIAS system (Davis, 1979). The system works mainly by analyzing mistakes of the consultation program, displaying the facts of the specific consultation case, the rules used by MYCIN, and its trace of reasoning. It then engages in a high-level dialogue (in a restricted set of natural language) with the expert builder of the knowledge base to try to discover the procedures by which the errors can be avoided. This knowledge is interpreted by TEIRESIAS so as to suggest possible changes in the rules of the consultation program. Taken together with the consultation model, TEIRESIAS represents an important example of a system that "knows what it knows," at least in the sense that one part of the representation can be used to represent properties and reasoning about another part. A different application of MYCIN techniques led to a consultant to help in the analysis of cases in the data base, which was implemented for use with ARAMIS (see Blum and Wiederhold, 1978; and Chapter 17).

The need to emulate the sequence of expert reasoning more accurately led to a new formulation of INTERNIST. The main concern was to develop a representation that would support strategies for handling multiple or composite hypotheses and would yield performance that converged more rapidly to the correct conclusions. Some of the elements introduced in the INTERNIST-II (Pople, 1977) system were *constrictor relationships* for describing very specific associations between findings and higher-level hypotheses, a multiproblem hypothesis generator with a modified scoring heuristic for taking advantage of the constrictor links, and control strategies for evaluating complexes of hypotheses rather than the individual hypothesis structures of the original system. Most recently, a knowledge-acquisition front end for INTERNIST has been adapted from the ZOG system, permitting the domain experts to enter their knowledge in a more natural manner (Freiherr, 1979). The problem of representing groups of related hypotheses in such a manner that they are "aggregated" in a natural way during inferencing has been a topic of concern for all of the researchers who deal with large hypothesis spaces. This question is a major consideration in the design of a new program for acid-base balance diagnosis and treatment (Patil, 1979).

A major problem that has not been adequately dealt with in the current consultation schemes is that of reasoning over temporal sequences of events and hypotheses. One approach to this problem, based on a real-time rule reevaluation within a MYCIN-like scheme, has been applied in the VM

system (Fagan, 1979) for ventilator management. In this application, the goal-directed strategy of MYCIN was not used, since the system must respond in an event-driven way to the changes in physiological status of the patient on the respirator. The inference of changes in hypotheses over the long-term course of chronic disease states was modeled in the CASNET/Glaucoma system by specialized time-dependent functions, and feedback of physiological parameter values is used in the reasoning of the Digitalis Therapy Advisor (Gorry et al., 1978). These examples represent specialized applications, and a general scheme for reasoning over time is still needed.

The *explanation* of reasoning has been a major concern of AIM systems, which has been extended recently to include tutorial advice in the GUIDON system (Clancey, 1979a; 1979c) for MYCIN-like consultants. An explanation scheme that is based on physiological and frame-based models has been developed for the Digitalis Therapy Advisor (Swartout, 1981).

A perennial problem for the designers of knowledge-based consultation programs has been to balance the mixture of declarative and procedural knowledge forms in their representations. In general, this has been alleviated by combining frames or semantic nets with production rules, as in IRIS (Trigoboff and Kulikowski, 1977), PROSPECTOR (Hart and Duda, 1977), CENTAUR (Aikins, 1979), NEUREX (Reggia, 1978), and NEUROLOGIST (Catanzarite and Greenburg, 1979), and in the knowledge-based schemata of EXPERT (Weiss and Kulikowski, 1979) and AGE (Nii and Aiello, 1979). Related to this are questions of modifying the control strategies so that the right kind of knowledge is applied to each problem-solving task, which have not as yet been explored in depth. A first attempt in this direction is the MDX system (Chandrasekaran et al., 1979), which develops a hierarchy of different "procedural experts" within a consultation system, with strict transfer of control protocols between them. The structure of experts in MDX directly parallels the links among the subspecialties of medicine. More research is needed to study not only this but other more flexible ways in which the control of concurrently operating experts can be coordinated.

As the number of examples of consultation programs and schemes increases, some common sets of techniques are beginning to emerge, which has led to the building of general tools for the construction of knowledge-based *expert systems*. This work has been characterized recently as *knowledge engineering* (Feigenbaum, 1978). Some of the general schemes for helping to build knowledge-based systems are EMYCIN, EXPERT, and AGE. The EMYCIN (van Melle, 1979) scheme is an outgrowth of MYCIN and permits the creator of a knowledge base to organize it so that it can be run with the MYCIN consultation control structure. Consultation programs in psychopharmacology (Brooks and Heiser, 1979) and structural analysis (Bennett and Englemore, 1979) illustrate the range of applications modeled with this representational scheme.

The EXPERT system (Kulikowski and Weiss, 1982; Weiss and Kulikowski, 1979) draws primarily on the CASNET experience and also provides a generalized consultation program that can be fitted with a knowledge base in any chosen medical specialty. Its representational scheme includes a hierarchical-causal network for hypotheses and treatments, a structured scheme for findings, and a set of production rules that permit the specification of contexts in terms of these elements. Models in rheumatology, neuro-ophthalmology, and endocrinology are being developed using this scheme (Freiherr, 1979). The system is designed so that physicians with some computer experience can construct models by writing them onto a file (with any system editor) using a simple descriptive language. The file is then compiled by a special program that checks for syntactical errors and produces a compiled model that can be efficiently run by the consultation program. Data-base updating and knowledge acquisition are also available to help in the process of debugging the model as it is tested against cases with reliable conclusions (Weiss and Kulikowski, 1979). A version of the system has been implemented on a minicomputer, thereby facilitating its dissemination to clinical environments.

The AGE (attempt to generalize) system (Nii and Aiello, 1979) provides a general set of technical tools for modeling consultative situations using the "blackboard" model (Lesser et al., 1975; Lesser and Erman, 1979), which was developed for handling the representation and processing of information from multiple sources of knowledge in speech understanding. Building a consultation model in AGE requires knowledge of Interlisp facilities, so this system is designed primarily for use by computer scientists working with medical specialists. Since the development of models that perform at an expert level has been shown to call for intensive interdisciplinary collaborations, such an approach is likely to continue as the main mode of research, at least until there are more experts who combine advanced training in both fields. Thus the current stage of development of knowledge engineering for medical consultation is one of constructive expansion in a number of varied applications. The next few years are likely to see many efforts at validation and application of these systems in realistic clinical environments.

The practical advances in developing knowledge-engineering tools continue to uncover new problems of a formal nature concerning representation, inference, and control in consultative problem solving. There is no lack of candidates for the title of "most difficult problem" when we attempt to study or emulate aspects of expert human reasoning on the computer. If a single set of problems qualifies for major attention, it might be those centered around the properties of concept abstraction and self-referencing that we associate strongly with "knowing what we know." Issues of concurrency in reasoning and related questions of whether and how to maintain logical and semantic consistency of the knowledge bases also present crucial open questions. These and other problems will continue to

offer sufficient challenges of an epistemological and formal nature and are likely to encourage active research that will parallel the engineering efforts for many years to come.

ACKNOWLEDGMENT

Part of this work was supported by a grant (RR 643) from the Biotechnology Resource Program, Division of Research Resources, National Institutes of Health.

5

Production Rules as a Representation for a Knowledge-Based Consultation Program

Randall Davis, Bruce G. Buchanan, and Edward H. Shortliffe

Among the early AIM systems, MYCIN was one of the most influential. Initially developed as a thesis project by Edward Shortliffe at Stanford University, the system spawned an active research group, which refined the program's capabilities and added some of the features described in this chapter. At Stanford, MYCIN served as a basis for several new experiments as well, some of which are described in other chapters in this book (viz., Chapters 10, 11, 15, and 19). Although MYCIN was never implemented for routine clinical use, its decision-making performance was validated in formal experiments, and it was shown to reach decisions at the level of an expert in the field (Yu et al., 1979a; 1979b). Its appeal, however, largely rests in the clarity of the representation and control techniques that it uses and in the human-engineering features that make it an easy system to learn to use and to demonstrate. The results of the MYCIN work and of its associated experiments have recently been described in a book about the project (Buchanan and Shortliffe, 1984).

Randall Davis joined the MYCIN group during its early days, and his own thesis research on knowledge acquisition, meta-level reasoning, and explanation evolved in that setting (Davis and Lenat, 1982). In 1977 Davis joined with Bruce Buchanan and Shortliffe to publish the following technical paper describing MYCIN and its capabilities. By the time this

From *Artificial Intelligence*, 8: 15–45 (1977). Used with the permission of North-Holland Publishing Company.

paper appeared, MYCIN had begun to exhibit a high level of performance as a consultant in the task of selecting antibiotic therapy for bacteremia. The report discusses issues of representation and design for the system. It also describes the basic task and discusses the constraints involved in the use of a program as a consultant. The control structure and knowledge representation of the system are examined in this light, and special attention is given to the impact of production rules as a representation. There is also brief discussion of the model of inexact reasoning developed for MYCIN, a numerical scheme that is further discussed in the review of AIM systems by Szolovits and Pauker (Chapter 9). Emphasis is also placed on the effort to maintain a separation between the knowledge in the system and its control mechanism, or inference engine. *The domain-independent portions of MYCIN became known as EMYCIN ("Essential MYCIN") and have been used to develop other expert systems, one of which is currently in use in a medical setting (PUFF—see Chapter 19).*

5.1 Introduction

Two recent trends in artificial intelligence research have been applications of AI to real-world problems and the incorporation in programs of large amounts of task-specific knowledge. The former is motivated in part by the belief that artificial problems may prove in the long run to be more a diversion than a base to build on and in part by the belief that the field has developed sufficiently to provide techniques capable of tackling real problems.

The move toward what have been called knowledge-based systems represents a change from previous attempts at generalized problem solvers (for example, GPS). Earlier work on such systems demonstrated that while there was a large body of useful general-purpose techniques (e.g., problem decomposition into subgoals, heuristic search in its many forms), these did not by themselves offer sufficient power for high performance. Rather than nonspecific problem-solving power, knowledge-based systems have emphasized both the accumulation of large amounts of knowledge in a single domain and the development of domain-specific techniques, in order to develop a high level of expertise.

There are numerous examples of systems embodying both trends, including efforts at symbolic manipulation of algebraic expressions (MATHLAB Group, 1974), speech understanding (Lesser et al., 1975), chemical inference (Buchanan and Lederberg, 1971), and the creation of computer consultants as interactive advisors for various tasks (Hart, 1975; Shortliffe et al., 1975), as well as several others.

In this paper we discuss issues of representation and design for one such knowledge-based application program—the MYCIN system devel-

oped over the past three years as an interdisciplinary project at Stanford University and discussed elsewhere (Shortliffe, 1976; Shortliffe et al., 1973; 1975; Shortliffe and Buchanan, 1975). Here we examine in particular how the implementation of various system capabilities is facilitated or inhibited by the use of production rules as a knowledge representation. In addition, the limits of applicability of this approach are investigated.

We begin with a review of features that were seen to be essential to any knowledge-based consultation system and suggest how these imply specific program design criteria. We note also the additional challenges offered by the use of such a system in a medical domain. This is followed by an explanation of the system structure and its fundamental assumptions. The bulk of the paper is then devoted to a report of our experience with the benefits and drawbacks of production rules as a knowledge representation.

5.2 System Goals

The MYCIN system was developed originally to provide consultative advice on diagnosis of and therapy for infectious diseases—in particular, bacterial infections in the blood.[1] From the start, the project has been shaped by several important constraints. The decision to construct a high-performance AI program in the consultant model brought with it several requirements. First, the program had to be *useful* if we expected to attract the interest and assistance of experts in the field. The task area was thus chosen partly because of a demonstrated need: in a recent year, for example, one of every four people in the U.S. was given penicillin and almost 90% of those prescriptions were unnecessary (Kagan et al., 1973). Problems such as these indicate the need for more (or more accessible) consultants to physicians selecting antimicrobial drugs. Usefulness also implies competence, consistently high performance, and ease of use. If advice is not reliable or is difficult to obtain, the utility of the program is severely impaired.

[1]We have recently begun investigating extensions to the system. The next medical domain will be the diagnosis and treatment of meningitis infections. This area is sufficiently different to be challenging and yet similar enough to suggest that some of the automated procedures we have developed may be quite useful. (*Ed. note:* This extension was successfully completed.) A paper by van Melle (1974) reports on an interesting effort at inserting an entirely different knowledge base into the body of the current system. A small part of an automobile repair manual was translated into production rules, and the appropriate attributes, objects, contexts, and vocabulary were provided. It then required relatively little effort to plug this new knowledge base into the standard system code, and a small but completely functional automobile consultant program resulted. [*Ed. note:* The general framework is now known as EMYCIN (van Melle et al., 1981).]

A second constraint was the need to design the program to accommodate a *large and changing body of technical knowledge.* It has become clear that large amounts of task-specific knowledge are required for high performance and that this knowledge base is subject to significant changes over time (Buchanan and Lederberg, 1971; Green et al., 1974). Our choice of a production rule representation was significantly influenced by such features of the knowledge base.

A third demand was for a system capable of handling an *interactive dialogue* and one that was not a "black box." This meant that it had to be capable of supplying coherent explanations of its results, rather than simply printing a collection of orders to the user. This was perhaps the major motivation for the selection of a symbolic reasoning paradigm, rather than one that, for example, relied totally on statistics. It meant also that the flow of dialogue (the order of questions) should make sense to a physician and not be determined by programming considerations. Interactive dialogue required, in addition, extensive human-engineering features designed to make interaction simple for someone unaccustomed to computers.

The choice of a medical domain brought with it additional demands (Shortliffe et al., 1974). Speed, access, and ease of use gained additional emphasis, since a physician's time is typically limited. The program also had to fill a need well recognized by the clinicians who would actually use the system, since the lure of pure technology is usually insufficient. Finally, the program had to be designed with an emphasis on its supportive role as a tool for the physician, rather than as a replacement for his or her own reasoning process.

Any implementation selected had to meet all these requirements. Predictably, some have been met more successfully than others, but all have been important factors in influencing the system's final design.

5.3 System Overview

5.3.1 The Task

The fundamental task is the selection of therapy for a patient with a bacterial infection. Consultative advice is often required in the hospital because the attending physician may not be an expert in infectious diseases, as, for example, when a cardiology patient develops an infection after heart surgery. Time considerations compound the problem. A specimen (of blood, urine, etc.) drawn from a patient may show some evidence of bacterial growth within 12 hours, but 24 to 48 hours (or more) are required for positive identification. The physician must therefore often decide, in the absence of complete information, whether or not to start treatment

and what drugs to use if treatment is required. Both of these may be difficult questions.

The task is made clearer by the initial and final parts of a sample dialogue with the MYCIN system, shown in Figures 5-1 and 5-2 (italicized comments toward the right are for explanatory purposes and are not part of the actual dialogue). A great deal of attention has been paid to making the interaction easy and natural. Questions have been carefully worded, for example, so as to set up expectations of responses without making them as uninteresting as multiple-choice questions. Currently, the dialogue is directed primarily by the system; patient information can be entered only by answering each question in turn.

There are several options available to make the interaction more informative. The user can, for instance, ask for a rephrasing of the question or a display of some (or all) of the acceptable responses. If a requested item of information is unavailable, he or she can respond with UNKNOWN (UNK or U, for short). The user can also modify any answer of which he or she is unsure by attaching a number indicating a degree of certainty (as, for example, in questions 13 and 15).

The main purpose of the system is to provide advice, and it does so via the *consultation program* (Figure 5-3). In addition to this performance program, there are three adjunct programs that increase system utility and flexibility: (a) a *question-answering program* that handles simple natural language inquiries about either the specific consultation or the general knowledge base of the program, (b) an *explanation program* for giving reasons why the performance program has requested an item of information or has drawn a particular conclusion, and (c) a *knowledge-acquisition program* for extending the knowledge base through a dialogue with an expert.

The system is written in Interlisp, runs on a DEC KI-10 with approximately 130K words of nonshared code, and is fast enough to permit convenient real-time interaction. The consultation program (with the required data structures) requires approximately 50K, the explanation program 10K, the question-answering program 15K, the rule-acquisition program 20K, and the rules themselves approximately 8K (the remainder includes a dictionary, information on drug properties, and various system utility features).

As is described below, each of these four interrelated programs aids in meeting one or more of the goals outlined above.

5.3.2 The Rules

The primary source of domain-specific knowledge is a set of some 450 rules, each with a premise and an action (Figure 5-4). The premise is a

--------PATIENT-1--------
1) Patient's name: (first-last)
**FRED SMITH

{*background data*}

2) Sex:
**MALE
3) Age:
**55
4) Have you been able to obtain positive cultures from a site
at which Fred Smith has an infection?
**YES
--------INFECTION-1--------
5) What is the infection?
**PRIMARY-BACTEREMIA
6) Please give the date and approximate time when signs or symptoms of
the primary-bacteremia (INFECTION-1) first appeared. (mo/da/yr)
**MAY 5, 1975
The most recent positive culture associated with the primary-
bacteremia (INFECTION-1) will be referred to as:
--------CULTURE-1--------
7) From what site was the specimen for CULTURE-1 taken?
**BLOOD
8) Please give the date and time when this blood culture
(CULTURE-1) was obtained. (mo/da/yr)
**MAY 9, 1975
The first significant organism from this blood culture (CULTURE-1)
will be referred to as:
--------ORGANISM-1--------
9) Enter the identity of ORGANISM-1:
**UNKNOWN
10) Is ORGANISM-1 a rod or coccus (etc.):
**ROD

{*typically the identity is not known and must be inferred*}

11) The gram stain of ORGANISM-1:
**GRAMNEG
12) Have there been POSITIVE cultures from sites which are NOT
associated with infections about which you are seeking
advice?
**NO
13) Did ORGANISM-1 grow aerobically?
**YES (8)
14) Did ORGANISM-1 grow anaerobically?
**YES
15) What is the suspected portal of entry of ORGANISM-1 into
this sterile site?
**GI(6)
.
.
.

{*more questions follow, to gather sufficient information on possible identities of the infecting organisms*}

FIGURE 5-1 Initial segment of a session with the consultation system. User responses are in boldface capital letters and follow the double asterisks. Italicized comments at the right are not part of the actual dialogue.

My therapy recommendation will be based on the following probable infection(s) and potential causative organism(s):

INFECTION-1 is PRIMARY-BACTEREMIA

{possible identities}

<Item 1> The identity of ORGANISM-1 may be PSEUDOMONAS-AERUGINOSA
<Item 2> The identity of ORGANISM-1 may be KLEBSIELLA-PNEUMONIAE
<Item 3> The identity of ORGANISM-1 may be E.COLI
<Item 4> The identity of ORGANISM-1 may be BACTEROIDES-FRAGILIS
<Item 5> The identity of ORGANISM-1 may be ENTEROBACTER
<Item 6> The identity of ORGANISM-1 may be PROTEUS-NON-MIRABILIS

46) On a scale of 0 to 4, where higher numbers indicate increasing severity, how would you rate Fred Smith's degree of sickness?
**3
47) Does Fred Smith have a clinically significant allergic reaction to any antimicrobial agent?
**NO
48) Fred Smith's weight in kilograms (or <number> POUNDS):
**70

{note that 2 drugs cover for 6 potential organisms}

[Rec 1] My preferred therapy recommendation is as follows:
In order to cover for Items <1 2 3 5 6>:
Give: GENTAMICIN
Dose: 119 mg (1.7 mg/kg) q8h IV [or IM] for 10 days
Comments: Modify dose in renal failure
In order to cover for Item <4>:
Give: CLINDAMYCIN
Dose: 595 mg (8.5 mg/kg) q6h IV [or IM] for 14 days
Comments: If diarrhea or other GI symptoms develop, patient should be evaluated for possible pseudomembranous colitis.

Do you wish to see the next choice therapy?

{alternative choices are available}

**Y
Which, if any, of the above drugs do you wish eliminated from consideration?
**CLINDAMYCIN

[Rec 2] Next best therapy recommendation:
In order to cover for Items <2 3 4 5 6>:
Give: CHLORAMPHENICOL
Dose: 1 gm (15 mg/kg) q6h IV for 14 days
Comments: Monitor patient's white count
In order to cover for Item <1>:
Give: GENTAMICIN
Dose: 119 mg (1.7 mg/kg) q8h IV [or IM] for 10 days
Comments: Modify dose in renal failure

FIGURE 5-2 Final segment of a consultation. Alternative choices for therapies are generated if requested.

Boolean combination of predicate functions on associative triples. Thus each clause of a premise has the following four components:

<predicate function> <object> <attribute> <value>

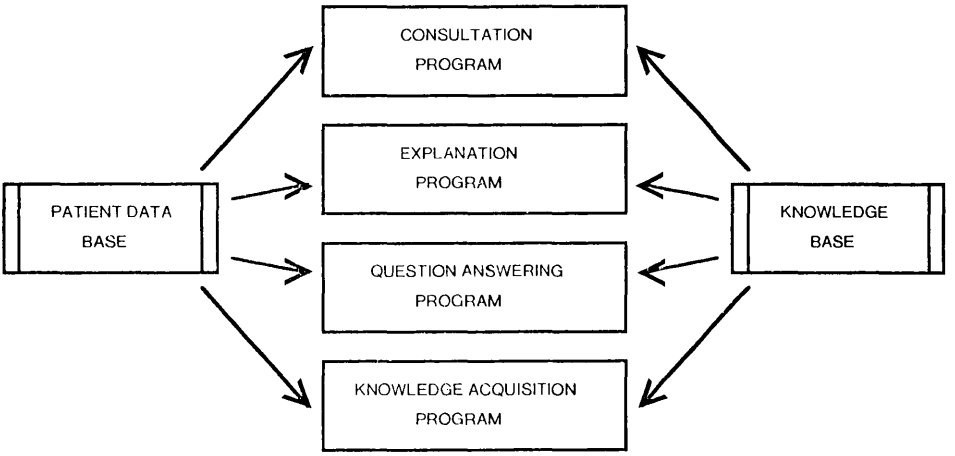


FIGURE 5-3 The six components of the system: Four programs, the knowledge base, and the patient data base. All of the system's knowledge of infectious disease is contained within the knowledge base. Data about a specific patient collected during a consultation are stored in the patient data base. Arrows indicate the direction of information flow.

There is a standardized set of 24 predicate functions (e.g., SAME, KNOWN, DEFINITE), some 80 attributes (e.g., IDENTITY, SITE, SENSITIVITY), and 11 objects (e.g., ORGANISM, CULTURE, DRUG) currently available for use as primitives in constructing rules. The premise is always a conjunction of clauses, but may contain arbitrarily complex conjunctions or disjunctions nested within each clause. Instead of writing rules whose premise would be a disjunction of clauses, we write a separate rule

```
PREMISE:  ($AND (SAME CNTXT INFECT PRIMARY-BACTEREMIA)
                (MEMBF CNTXT SITE STERILESITES)
                (SAME CNTXT PORTAL GI))
ACTION:   (CONCLUDE CNTXT IDENT BACTEROIDES TALLY .7)

IF:  1) The infection is primary-bacteremia, and
     2) The site of the culture is one of the sterile sites, and
     3) The suspected portal of entry of the organism is the gastro-intestinal tract,
THEN: There is suggestive evidence (.7) that the identity of the organism is bacteroides.
```

FIGURE 5-4 A rule from the knowledge base. $AND and $OR are the multi-valued analogues of the standard Boolean AND and OR.

for each clause. The action part indicates one or more conclusions that can be drawn if the premises are satisfied; hence the rules are (currently) purely inferential in character.

It is intended that each rule embody a single, modular chunk of knowledge and state explicitly in the premise all necessary context. Since the rule uses a vocabulary of concepts common to the domain, it forms, by itself, a comprehensible statement of some piece of domain knowledge. As will become clear, this characteristic is useful in many ways.

Each rule is, as is evident, highly stylized, with the IF/THEN format and the specified set of available primitives. While the LISP form of each is executable code (and, in fact, the premise is simply evaluated by LISP to test its truth, and the action evaluated to make its conclusions), this tightly structured form makes possible the examination of the rules by other parts of the system. This in turn leads to some important capabilities, to be described below. For example, the internal form can be automatically translated into readable English, as shown in Figure 5-4.

Despite this strong stylization, we have not found the format restrictive. This is evidenced by the fact that of nearly 450 rules on a variety of topics, only 8 employ any significant variations. The limitations that do arise are discussed below.

5.3.3 Judgmental Knowledge

Since we want to deal with real-world domains in which reasoning is often judgmental and inexact, we require some mechanism for being able to say "A *suggests* B" or "C and D *tend* to rule out E." The numbers used to indicate the strength of a rule (e.g., the .7 in Figure 5-4) have been termed *certainty factors* (CF's). The methods for combining CF's are embodied in a model of approximate implication. Note that while these are derived from and are related to probabilities, they are distinctly different [for a detailed review of the concept, see Shortliffe and Buchanan (1975)]. For the rule in Figure 5-4, then, the evidence is strongly indicative (.7 out of 1), but not absolutely certain. Evidence confirming a hypothesis is collected separately from that disconfirming it, and the truth of the hypothesis at any time is the algebraic sum of the current evidence for and against it. This is an important aspect of the truth model, since it makes plausible the simultaneous existence of evidence in favor of and against the same hypothesis. We believe this is an important characteristic of any model of inexact reasoning.

Facts about the world are represented as quadruples, with an associative triple and its current CF (Figure 5-5). Positive CF's indicate a predominance of evidence confirming a hypothesis; negative CF's indicate a predominance of disconfirming evidence.

Note that the truth model permits the coexistence of several plausible values for a single attribute, if they are suggested by the evidence. Thus,

```
(SITE CULTURE-1 BLOOD 1.0)
(IDENT ORGANISM-2 KLEBSIELLA .25)
(IDENT ORGANISM-2 E.COLI .73)
(SENSITIVS ORGANISM-1 PENICILLIN -1.0)
```

FIGURE 5-5 Samples of information in the patient data base during a consultation.

for example, after attempting to deduce the identity of an organism, the system may have concluded (correctly) that there is evidence both that the identity is *E. coli* and that it is *Klebsiella,* despite the fact that they are mutually exclusive possibilities.

As a result of the program's medical origins, we also refer to the attribute part of the triple as a *clinical parameter* and use the two terms interchangeably here. The object part (e.g., CULTURE-1, ORGANISM-2) is referred to as a *context.* This term was chosen to emphasize its dual role as both part of the associative triple and as a mechanism for establishing the scope of variable bindings. As explained below, the contexts are organized during a consultation into a tree structure whose function is similar to those found in "alternate world" mechanisms of languages like QA4.

5.3.4 Control Structure

The rules are invoked in a backward-unwinding scheme that produces a depth-first search of an AND/OR goal tree (and hence is similar in some respects to PLANNER's consequent theorems): given a goal to establish, we retrieve the (precomputed) list of all rules whose conclusions bear on the goal. The premise of each is evaluated, with each predicate function returning a number between -1 and 1. $AND (the multi-valued analogue of the Boolean AND) is a minimization operation, and $OR (similar) takes the maximum.[2] For rules whose premise evaluates successfully (i.e., greater than .2, an empirical threshold), the action part is evaluated, and the conclusion made with a certainty that is equal to:

<premise value> * <certainty factor>

[2]Note that, unlike standard probability theory, $AND does not involve any multiplication over its arguments. Since CF's are not probabilities, there is no *a priori* reason why a product should be a reasonable number. There is, moreover, a long-standing convention in work with multi-valued logics which interprets AND as min and OR as max. It is based primarily on intuitive grounds: if a conclusion requires all of its antecedents to be true, then it is a relatively conservative strategy to use the smallest of the antecedent values as the value of the premise. Similarly, if any one of the antecedent clauses justifies the conclusion, one is safe in taking the maximum value.

Those that evaluate unsuccessfully are bypassed, while a clause whose truth cannot be determined from current information causes a new subgoal to be set up, and the process recurses. Note that *evaluating* here means simply invoking the LISP EVAL function—there is no additional rule interpreter necessary, since $AND, $OR, and the predicate functions are all implemented as LISP functions.

Variations from the Standard Depth-First Search

Unlike PLANNER, however, the subgoal that is set up is a generalized form of the original goal. If, for example, the unknown clause is "the identity of the organism is *E. coli*," the subgoal that is set up is "determine the identity of the organism." The new subgoal is therefore always of the form "determine the value of the <attribute>" rather than "determine whether the <attribute> is equal to <value>." By setting up the generalized goal of collecting all evidence about an attribute, the program effectively exhausts each subject as it is encountered and thus tends to group together all questions about a given topic. This results in a system that displays a much more focused, methodical approach to the task, which is a distinct advantage where human-engineering considerations are important. The cost is the effort of deducing or collecting information that is not strictly necessary. However, since this occurs rarely—only when the <attribute> can be deduced with certainty to be the <value> named in the original goal—we have not found this to be a problem in practice.

A second deviation from the standard rule-unwinding approach is that every rule relevant to a goal is used. The premise of each rule is evaluated, and if successful, its conclusion is invoked. This continues until all relevant rules have been used or until one of them has given the result with certainty. This use of all rules is motivated in part by the model of judgmental reasoning and the approximate implication character of rules—unless a result is obtained with certainty, we should be careful to collect all positive and negative evidence. It is also appropriate to the system's current domain of application, clinical medicine, where a conservative strategy of considering all possibilities and weighing all the evidence is preferred.

If after trying all relevant rules (referred to as *tracing the subgoal*), the total weight of the evidence about a hypothesis falls between $-.2$ and .2 (again, empirically determined), the answer is regarded as still unknown. This may happen if no rule were applicable, if the applicable rules were too weak, if the effects of several rules offset each other, or if there were no rules for this subgoal at all. In any of these cases, when the system is unable to infer the answer, it asks the user for the value (using a phrase that is stored along with the attribute itself). Since the legal values for each attribute are also stored with it, the validity (or spelling) of the user's response is easily checked. (This also makes possible a display of acceptable answers in response to a ? input from the user.)

The strategy of always attempting to deduce the value of a subgoal and asking only when that fails would ensure the minimum number of questions. It would also mean, however, that work might be expended searching for a subgoal, arriving perhaps at a less than definite answer, when the user already knew the answer with certainty. In response to this, some of the attributes have been labeled as LABDATA, indicating that they represent entities that are often available as results of laboratory tests. In this case the deduce-then-ask procedure is reversed, and the system will attempt to deduce the answer only if the user cannot supply it. Given a desire to minimize both tree search and the number of questions asked, there is no guaranteed optimal solution to the problem of deciding when to ask for information and when to try to deduce it. But the LABDATA distinction used here has performed quite well and seems to embody an appropriate criterion.

Three other recent additions to the tree-search procedure have helped improve performance. First, before the entire list of rules for a subgoal is retrieved, the system attempts to find a sequence of rules that would establish the goal with certainty, based only on what is currently known. Since this is a search for a sequence of rules with CF = 1, we have termed the result a *unity path*. Besides efficiency considerations, this process offers the advantage of allowing the system to make "commonsense" deductions with a minimum of effort (rules with CF = 1 are largely definitional). Since it also helps minimize the number of questions, this check is performed even before asking about LABDATA attributes. Because there are few such rules in the system, the search is typically very brief.

Second, a straightforward bookkeeping mechanism notes the rules that have failed previously and avoids trying to reevaluate any of them. (Recall that a rule may have more than one conclusion, may accordingly conclude about more than a single attribute, and hence may get retrieved more than once).

Finally, we have implemented a partial evaluation of rule premises. Since many attributes are found in several rules, the value of one clause (perhaps the last) in a premise may already have been established, even while the rest are still unknown. If this clause alone would make the premise false, there is clearly no reason to do all the search necessary to try to establish the others. Each premise is thus "previewed" by evaluating it on the basis of currently available information. This produces a Boolean combination of TRUEs, FALSEs, and UNKNOWNs, and straightforward simplification (e.g., $F \wedge U \equiv F$) indicates whether the rule is guaranteed to fail.

Templates

The partial evaluation is implemented in a way that demonstrates the utility of stylized coding in the rules. It is also forms an example of what was alluded to earlier when we noted that the rules may be examined by various

Function	*Template*	*Sample function call*
SAME	(SAME CNTXT PARM VALUE)	(SAME CNTXT SITE BLOOD)

FIGURE 5-6 PARM is shorthand for clinical parameter (attribute); VALUE is the corresponding value; CNTXT is a free variable that references the context in which the rule is invoked.

elements of the system, as well as executed. We require a way to tell if any clause in the premise is known to be false. We cannot simply EVAL each individually, since a subgoal that had never been traced before would send the system off on its recursive search. However, if we can establish which attribute is referenced by the clause, it is possible to determine (by reference to internal flags) whether or not it has been traced previously. If so, the clause can be EVALed to obtain the value. A template (Figure 5-6) associated with each predicate function makes this possible.

The template indicates the generic type and order of arguments to the predicate function, much like a simplified procedure declaration. It is not itself a piece of code, but is simply a list structure of the sort shown above and indicates the appearance of an interpreted call to the predicate function. Since rules are kept in interpreted form (as shown in Figure 5-4), the template can be used as a guide to dissect a rule. This is done by retrieving the template for the predicate function found in each clause and then using that as a guide to examining the clause. In the case of the function SAME, for instance, the template indicates that the attribute (PARM) is the third element of the list structure that comprises the function call. The preview mechanism uses the templates to extract the attribute from the clause in question and can then determine whether or not it has been traced.

There are two points of interest here. First, part of the system is "reading" the code (the rules) being executed by another part; and second, this reading is guided by the information carried in components of the rules themselves. The ability to read the code could have been accomplished by requiring all predicate functions to use the same format, but this is obviously awkward. By allowing each function to describe the format of its own calls, we permit code that is stylized without being constrained to a single form and hence is flexible and much easier to use. We require only that each form be expressible in a template built from the current set of template primitives (e.g., PARM, VALUE, etc.). This approach also ensures that the capability will persist in the face of future additions to the system. The result is one example of the general idea of giving the system access to and an "understanding" of its own representations. This idea has been used and discussed extensively by Davis (1976).

We have also implemented antecedent-style rules. These are rules that are invoked if a conclusion is made that matches their premise condition.

```
PREMISE:  ($AND (MEMBF SITE CNTXT NONSTERILESITES)
                (THEREARE OBJRULES (MENTIONS CNTXT PREMISE SAMEBUG))
ACTION:   (CONCLIST CNTXT UTILITY YES TALLY -1.0)

IF:  1) The site of the culture is one of the nonsterile sites, and
     2) There are rules which mention in their premise a previous organism
        which may be the same as the current organism
THEN:  It is definite (1.0) that each of them is not going to be useful.
```

FIGURE 5-7 A meta-rule. A previous infection that has been cured (temporarily) may reoccur. Thus one of the ways to deduce the identity of the current organism is by reference to previous infections. However, this method is not valid if the current infection was cultured from one of the nonsterile culture sites. Thus this meta-rule says, in effect, "If the current culture is from a nonsterile site, don't bother trying to deduce the identity of the current organism from identities of previous organisms."

They are currently limited to commonsense deductions (i.e., CF = 1) and exist primarily to improve system efficiency. Thus, for example, if the user responds to the question of organism identity with an answer of which he or she is certain, there is an antecedent rule that will deduce the organism gram stain and morphology. This saves the trouble of deducing these answers later via the subgoal mechanism described above and allows rejection of rules using the preview mechanism described above.

5.3.5 Meta-Rules

With the system's current collection of 450 rules, exhaustive invocation of rules would be quite feasible, since the maximum number of rules for a single subgoal is about 30. We are aware, however, of the problems that may occur if and when the collection grows substantially larger. It was partly in response to this that we developed an alternative to exhaustive invocation by implementing the concept of *meta-rules*. These are strategy rules that suggest the best approach to a given subgoal. They have the same format as the clinical rules (Figure 5-7), but can indicate that certain clinical rules should be tried first, last, before others, or not at all. Thus before the entire list of rules applicable to any subgoal is processed, the meta-rules for that subgoal are evaluated. They may rearrange or shorten the list, effectively ordering the search or pruning the tree. By making them specific to a given subgoal, we can specify precise heuristics without imposing any extra overhead in the tracing of other subgoals.

Note, however, that there is no reason to stop at one level of meta-rules. We can generalize this process so that, before invoking any list of rules, we check for the existence of rules of the next higher order to use

in pruning or rearranging the first list. Thus, while meta-rules are strategies for selecting clinical rules, second-order meta-rules would contain information about which strategy to try, third-order meta-rules would suggest criteria for deciding how to choose a strategy, etc. These higher-order meta-rules represent a search by the system through *strategy space*, and appear to be powerful constraints on the search process at lower levels. (We have not yet encountered higher-order meta-rules in practice, but neither have we actively sought them.)

Note also that since the system's rule unwinding may be viewed as tree search, we have the appearance of a search through a tree with the interesting property that each branch point contains information on the best path to take next. Since the meta-rules can be judgmental, there exists the capability of writing numerous, perhaps conflicting, heuristics and having their combined judgment suggest the best path. Finally, since meta-rules refer to the clinical rules by their content rather than by name, the method automatically adjusts to the addition or deletion of clinical rules, as well as to modifications to any of them.

The capability of meta-rules to order or prune the search tree has proved to be useful in dealing with another variety of knowledge as well. For the sake of human engineering, for example, it makes good sense to ask the user first about the positive cultures (those showing bacterial growth) before asking about negative cultures. Formerly, this design choice was embedded in the ordering of a list buried in the system code. Yet it can be stated quite easily and explicitly in a meta-rule, yielding the significant advantages of making it both readily explainable and modifiable. Meta-rules have thus proved capable of expressing a limited subset of the knowledge formerly embedded in the control structure code of the system.

Meta-rules may also be used to control antecedent rule invocation. Thus we can write strategies that control the depth and breadth of conclusions drawn by the system in response to a new piece of information.

An overview of these mechanisms is shown in Figure 5-8, and indicates the way they function together to ensure an efficient search for each subgoal.

The final aspect of the control structure is the tree of contexts (recall the dual meaning of the term) constructed dynamically from a fixed hierarchy as the consultation proceeds (Figure 5-9). This serves several purposes. First, bindings of free variables in a rule are established by the context in which the rule is invoked, with the standard access to contexts that are its ancestors. Second, since this tree is intended to reflect the relationships of objects in the domain, it helps structure the consultation in ways familiar to the user. In the current domain, a patient has one or more infections, each of which may have one or more associated cultures, each of which in turn may have one or more organisms growing in it, and so on.

```
Procedure FINDVALUEOF (item GOAL)
    begin item X; list L; rule R; premise-clause P;
    if (X ← UNITYPATH(GOAL)) then return (X);
    if LABDATA(GOAL) and DEFINITE-ANSWER(X ← ASKUSER(GOAL)) then return(X);
    L ← RULES-ABOUT(GOAL);
    L ← APPLY-METARULES(GOAL,L,∅);

    for R ∈ L do
        unless PREVIEW(R) = false do
            begin "evaluate-rule"
            for P ∈ PREMISES-OF(R) do
                begin "test-each-premise-clause"
                if not TRACED(ATTRIBUTE-IN(P)) then FINDVALUEOF(ATTRIBUTE-IN(P));
                if EVALUATION-OF(P) < .2 then next(R);
                end "test-each-premise-clause";
            CONCLUDE(CONCLUSION-IN(R));
            if VALUE-KNOWN-WITH-CERTAINTY(GOAL) then
                begin MARK-AS-TRACED(GOAL); return(VALUEOF(GOAL)); end;
            end "evaluate-rule";

    MARK-AS-TRACED(GOAL);
    if VALUEOF(GOAL) = unknown and NOT-ALREADY-ASKED(GOAL)
                then return(ASKUSER(GOAL))
                else return(VALUEOF(GOAL));
    end;

Procedure APPLY-METARULES(item GOAL; list L; integer LEVEL);
    begin list M; rule Q;
    if (M ← METARULES-ABOUT(GOAL,LEVEL + 1))
                then APPLY-METARULES(GOAL,M,LEVEL + 1);
    for Q ∈ M do USE-METARULE-TO-ORDER-LIST(Q,L);
    return(L);
    end;

Procedure CONCLUDE(action-clause CONCLUSION);
    begin rule T; list L;
    UPDATE-VALUE-OF(ATTRIBUTE-IN(CONCLUSION), VALUE-IN(CONCLUSION));
    L ← ANTECEDENTRULES-ASSOCIATED-WITH(CONCLUSION);
    L ← APPLY-METARULES(ATTRIBUTE-IN(CONCLUSION),L,0);
    for T ∈ I do CONCLUDE(CONCLUSION-IN(T));
    end;
```

FIGURE 5-8 The control structure as it might appear in an ALGOL-like language.

5.4 Relation to Other Work

We outline briefly in this section a few programs that relate to various aspects of our work. Some of these have provided the intellectual basis from which the present system evolved, others have employed techniques that are similar, while still others have attempted to solve closely related

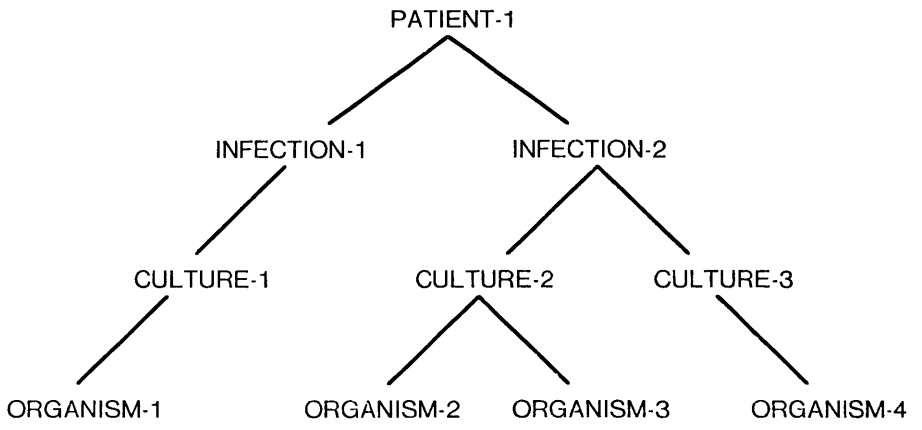


FIGURE 5-9 A sample of the contexts that may be sprouted during a consultation.

problems. Space limitations preclude detailed comparisons, but we indicate some of the more important distinctions and similarities.

There have been a large number of attempts to aid medical decision making (see Chapter 3 for an extensive review). The basis for some programs has been simple algorithmic processes, often implemented as decision trees (Meyer and Weissman, 1973; Warner et al., 1972a) or more complex control structures in systems tailored to specific disorders (Bleich, 1971). Many have based their diagnostic capabilities on variations of Bayes' Theorem (Gorry and Barnett, 1968a; Warner et al., 1964) or on techniques derived from utility theory in operations research (see Chapter 2). Models of the patient or disease process have also been used successfully (Silverman, 1975; Kulikowski et al., 1973) (see also Chapter 6). A few recent efforts have been based on some form of symbolic reasoning. In particular, the glaucoma diagnosis system described in Chapter 7 and the diagnosis system of Pople et al. (Chapter 8) can also be viewed as rule-based systems.

Carbonell's work (1970) represents an early attempt to make uncertain inferences in a domain of concepts that are strongly linked, much as MYCIN's are. Although the purpose of Carbonell's system was computer-aided instruction rather than consultation, much of our initial design was influenced by his semantic net model.

The basic production rule methodology has been applied in many different contexts, in attempts to solve a wide range of problems [see, for example, Davis and King (1977) for an overview]. The most directly relevant of these is the DENDRAL system (Buchanan and Lederberg, 1971), which has achieved a high level of performance on the task of mass spectrum analysis. Much of the initial design of MYCIN was influenced by the

experience gained in building and using the DENDRAL system, which in turn was based in part on the work of Waterman (1970).

There have been numerous attempts to create models of inexact reasoning. Among the more recent is LeFaivre (1974), which reports on the implementation of a language to facilitate fuzzy reasoning. It deals with many of the same issues of reasoning under uncertainty that are detailed in Shortliffe and Buchanan (1975).

The approach to natural language used in our system has thus far been quite elementary, primarily keyword-based. Some of the work reported by Colby et al. (1974) suggested to us initially that this might be a sufficiently powerful approach for our purposes. This has proven generally true because the technical language of this domain contains relatively few ambiguous words.

The chess-playing program of Zobrist and Carlson (1973) employs a knowledge representation that is functionally quite close to ours. The knowledge base of that system consists of small sequences of code that recognize patterns of pieces and then conclude (with a variable weighting factor) the value of obtaining that configuration. These workers report quite favorably on the ease of augmenting a knowledge base organized along these lines.

The natural language understanding system of Winograd (1972) had some basic explanation capabilities similar to those described here and could discuss its actions and plans.

As noted, part of our work has involved making it possible for the system to understand its own operation. Many of the explanation capabilities were designed and implemented with this in mind and it has significantly influenced design of the knowledge-acquisition system as well. These efforts are related in a general way to the long sequence of attempts to build program-understanding systems. Such efforts have been motivated by, among other things, the desire to prove correctness of programs [as in Waldinger and Levitt (1974) or Manna (1969)] and as a basis for automatic programming [as in Green et al. (1974)]. Most of these systems attempt to assign meaning to the code of some standard programming language like LISP or ALGOL. Our attempts have been oriented toward supplying meaning for the terms used in MYCIN's production rules (such as SAME). The task of program understanding is made easier by approaching it at this higher conceptual level, and the result is correspondingly less powerful. We cannot, for instance, prove that the implementation of SAME is correct. We can, however, employ the representation of meaning in other useful ways. It forms, for example, the basis for much of the knowledge-acquisition program (see Section 5.6.3) and permits the explanation program to be precise in explaining the system's actions [see Davis (1976) for details].

Finally, similar efforts at computer-based consultants have recently been developed in different domains. The work detailed by Nilsson (1975) and Hart (1975) has explored the use of a consultation system similar to

the one described here as part of an integrated vision, manipulation, and problem-solving system. Recent work on an intelligent terminal system (Anderson and Gillogly, 1977) has been based in part on a formalism that grew out of early experience with the MYCIN system.

5.5 Fundamental Assumptions

We attempt here to examine some of the assumptions that are explicit and implicit in our use of production rules. This will help to suggest the range of application for these techniques and to indicate some of their strengths and limitations.

There are several assumptions implicit in both the character of the rules and the ways in which they are used. First, it must be possible to write such judgmental rules. Not every domain will support this. Writing such rules appears to require a field that has attained a certain level of formalization, that includes perhaps a generally recognized set of primitives and at least a minimal understanding of basic processes. It does not seem to extend to one that has achieved a thorough, highly formalized level, however. Assigning certainty factors to a rule should thus be a reasonable task whose results are repeatable, but not a trivial one in which all rules are assigned a certainty of 1.0.

Second, we require a domain in which there is a limited sort of interaction between conceptual primitives. Our experience has suggested that a rule with more than about six clauses in the premise becomes conceptually unwieldy. The number of factors interacting in a premise to trigger an action therefore has a practical (but no theoretical) upper limit. Also, the AND/OR goal tree mechanism requires that the clauses of a rule premise can be set up as nonconflicting subgoals for the purposes of establishing each of them [just as in robot problem solving; see Fahlman (1974) and the comment on side effects in Siklossy and Roach (1973)]. Failure of this criterion causes results that depend on the order in which evidence is collected. We are thus making fundamental assumptions concerning two forms of interaction—we assume (a) that only a small number of factors (about six) must be considered simultaneously to trigger an action, and (b) that the presence or absence of each of those factors can be established without adverse effect on the others.

Also, certain characteristics of the domain will influence the continued utility of this approach as the knowledge base grows. Where there are a limited number of attributes for a given object, the growth in the number of rules in the knowledge base will not produce an exponential growth in search time for the consultation system. Thus, as newly acquired rules begin to reference only established attributes, use of these rules in a consultation will not produce further branching, since the attributes men-

tioned in their premises will have already been traced. In addition, we assume that large numbers of antecedent rules will not be necessary, thus avoiding very long chains of forward deductions.

There are essential assumptions as well in the use of this formalism as the basis for an interactive system. First, our explanation capabilities (reviewed in Section 5.6.2) rest on the assumption that display of either a rule or some segment of the control flow is a reasonable explanation of system behavior. Second, much of the approach to rule acquisition is predicated on the assumption that experts can be "debriefed," that is, that they can recognize and then formalize chunks of their own knowledge and experience and express them as rules. Third, the IF/THEN format of rules must be sufficiently simple, expressive, and intuitive that it can provide a useful language for expressing such formalizations. Finally, the system's mode of reasoning (a simple *modus ponens* chaining) must appear natural enough that a user can readily follow along.

There is an important assumption, too, in the development of a system for use by two classes of users. Since the domain experts who educate the system so strongly influence its conceptual primitives, vocabulary, and knowledge base, we must be sure that the naive users who come for advice speak the same language.

The approach we describe does not, therefore, seem well suited to domains requiring a great deal of complex interaction between goals, or to those for which it is difficult to compose sound judgmental rules. As a general indication of potentially useful applications, we have found that cognitive tasks are good candidates. In one such domain, antibiotic therapy selection, we have met with encouraging success.

5.6 Production Rules as a Knowledge Representation Scheme

In Section 5.2 we outlined three design goals for the system we are developing: utility (including competence), maintenance of an evolutionary knowledge base, and support of an interactive consultation. Our experience has suggested that production rules offer a knowledge representation that greatly facilitates the accomplishment of these goals. Such rules are straightforward enough to make feasible many interesting features beyond performance, yet powerful enough to supply significant problem-solving capabilities. Among the features discussed below are the ability for explanation of system performance and for acquisition of new rules, as well as the general "understanding" by the system of its own knowledge base. In each case we indicate the current performance levels of the system and evaluate the role of production rules in helping to achieve this performance.

5.6.1 Competence

The competence of the system has been evaluated in two studies in the past few years. In mid-1974, a semiformal study was undertaken, employing five infectious disease experts not associated with the project (Shortliffe, 1976). They were asked to evaluate the system's performance on 15 cases of bacteremia selected from current inpatients. We evaluated such parameters as the presence of extraneous questions, the absence of important ones, the system's ability to infer the identity of organisms, and its ability to select appropriate therapy. The principal problem discovered was an insufficient number of rules concerned with evaluating the severity of a patient's illness. Nevertheless, the experts approved of MYCIN's therapy recommendation in 72% of the evaluations. (There were also considerable differences of opinion regarding the best therapy as selected by the experts themselves.)

A more formal study is currently under way. Building on our experience gained in 1974, we designed a more extensive questionnaire and prepared detailed background information on a new set of 15 patients. These were sent to five experts associated with a local hospital and to five others across the country. This will allow us to evaluate performance and, in addition, to measure the extent to which the system's knowledge base reflects regional trends in patient care.[3]

Advantages of Production Rules

Recent problem-solving efforts in AI have made it clear that high performance of a system is often strongly correlated with the depth and breadth of the knowledge base. Hence the task of accumulation and management of a large and evolving knowledge base soon poses problems that dominate those encountered in the initial phases of knowledge-base construction. Our experience suggests that giving the system itself the ability to examine and manipulate its knowledge base provides some capabilities for confronting these problems. These are discussed in subsequent sections.

The selection of production rules as a knowledge representation is in part a response to this fact. One view of a production rule is as a modular segment of code (Winograd, 1975) that is heavily stylized (Waterman, 1970; Buchanan and Lederberg, 1971). Each of MYCIN's rules is, as noted, a simple conditional statement: the premise is constrained to be a Boolean expression, the action contains one or more conclusions, and each is completely modular and independent of the others. Such *modular, stylized coding* is an important factor in building a system that is to achieve a high level of competence.

---

[3]*Ed. note:* This formal evaluation of the bacteremia rules was subsequently published (Yu et al., 1979b), as was a third study of the system's meningitis performance (Yu et al., 1979a).

For example, any stylized code is easier to examine than is unstylized code. This is used in several ways in the system. Initial integration of new rules into the knowledge base can be automated, since their premise and action parts can be systematically scanned, and the rules can then be added to the appropriate internal lists. In the question-answering system, inquiries of the form "Do you recommend clindamycin for bacteroides?" can be answered by retrieving rules whose premise and action contain the relevant items. Similarly, the detection of straightforward cases of contradiction and subsumption is made possible by the ability to examine rule contents. Stylized code also makes feasible the direct manipulation of individual rules, facilitating automatic correction of such undesirable interactions.

The benefits of modularized code are well understood. Especially significant in this case are the ease of adding new rules and the relatively uncomplicated control structure that the modular rules permit. Since rules are retrieved because they are relevant to a specific goal (i.e., they mention that goal in their action part), the addition of a new rule requires only that it be added to the appropriate internal list according to the clinical parameters found in its action. A straightforward depth-first search (the result of the backward chaining of rules) is made possible by the lack of interactions among rules.

These benefits are common to stylized code of any form. Stylization in the form of production rules in particular has proved to be a useful formalism for several reasons. In the domain of deductive problems especially, it has proven to be a natural way of expressing knowledge. It also supplies a clear and convenient way of expressing modular chunks of knowledge, since all necessary context is stated explicitly in the premise. This in turn makes it easier to ensure proper retrieval and use of each rule. Finally, in common with similar formalisms, one rule never directly calls another. This is a significant advantage in integrating a new rule into the system—it can simply be "added to the pot" and no other rule need be changed to ensure that it is called (compare this with the addition of a new procedure to a typical ALGOL-type program).

Shortcomings of Production Rules

Stylization and modularity also result in certain shortcomings, however. It is, of course, somewhat harder to express a given piece of knowledge if it must be put into a predetermined format. The intent of a few of the rules in our system is thus less than obvious to the naive user even when translated into English. The requirement of modularity (along with the uniformity of the knowledge base) means all necessary contextual information must be stated explicitly in the premise, and this at times leads to rules that have awkwardly long and complicated premises.

Another shortcoming in the formalism arises in part from the backward-chaining control structure. It is not always easy to map a sequence

of desired actions or tests into a set of production rules whose goal-directed invocation will provide that sequence. Thus, while the system's performance is reassuringly similar to some human reasoning behavior, the creation of appropriate rules that result in such behavior is at times nontrivial. This may in fact be due more to programming experience that is oriented primarily toward ALGOL-like languages rather than to any essential characteristic of production rules. After some experience with the system we have improved our skill at "thinking backward."

A final shortcoming arises from constraining rule premises to contain "pure" predicates.[4] This forces a pure problem reduction mode in the use of rules: each clause of a premise is set up as an independent goal, and execution of the action should be dependent solely on the success or failure of the premise evaluation, without referencing the precise value of that evaluation. It is at times, however, extremely convenient to write what amounts to a "for each" rule, as in "for each organism such that . . . conclude . . . ." A few rules of this form are present in the system (including, for example, the meta-rule in Figure 5-7), and they are made to appear formally like the rest by allowing the premise to compute a value (the set of items that satisfy the premise) that is passed to the action clause via a global variable. While this has been relatively successful, the violation of the basic formalism results in other difficulties—in particular, in the explanation system, which produces somewhat murky explanations of such rules. We are working toward a cleaner solution of this problem.

5.6.2 Explanation

Augmentation or modification of any knowledge base is facilitated by the ability to discover what knowledge is currently in the system and how it is used. The system's acceptance (especially to a medical audience) will be strongly dependent on the extent to which its performance is natural (i.e., humanlike) and transparent. Lack of acceptance of some applications programs can be traced to their obscure reasoning mechanisms, which force the user to accept or reject advice without a chance to discover its basis. One of our original design criteria, then, was to give the system the ability to provide explanations of its behavior and knowledge. It soon became evident that an approach relying on some form of symbolic reasoning (rather than, for example, statistics) would make this feasible. This was one of the primary reasons behind the choice of the production rule representation, and it has continued to influence the program's development.

---

[4]That is, a predicate that returns a value indicating only success or failure. Since we use a multi-valued logic, the predicate functions in rule premises return a number between 0 and 1. The alternative approach is to allow any non-NIL value to indicate success (e.g., the MEMBER function in LISP).

(a) **** HOW DID YOU DECIDE THAT ORGANISM-1 MIGHT BE AN E.COLI?**
I used RULE021 to conclude that the identity of ORGANISM-1 is e.coli.
This gave a cumulative CF of (.47).
Then I used RULE084 to conclude that the identity of ORGANISM-1 is e.coli. This gave a cumulative CF of (.55).
Finally, I used RULE003 to conclude that the identity of ORGANISM-1 is e.coli. This gave a cumulative CF of (.74).

(b) **** DID YOU USE RULE 163 TO FIND OUT ANYTHING ABOUT ORGANISM-1?**
RULE163 was tried in the context of ORGANISM-1, but it failed because it is not true that the patient has had a genito-urinary tract manipulative procedure (clause 3).

(c) **** DID RULE 116 TELL YOU ANYTHING ABOUT INFECTION-1?**
No. RULE116 would be relevant to the primary-bacteremia when applied to ORGANISM-1.
But clause 2 ("the gram stain of the organism is not known") was already known to be false for ORGANISM-1, so the rule was never tried.

(d) **** WHY DIDN'T YOU CONSIDER STREPTOCOCCUS AS A POSSIBILITY FOR ORGANISM-1?**
The following rule could have been used to determine that the identity of ORGANISM-1 was streptococcus: RULE033
But clause 2 ("the morphology of the organism is coccus") was already known to be false for ORGANISM-1, so the rule was never tried.

(e) **** DO YOU RECOMMEND CLINDAMYCIN FOR BACTEROIDES?**
Yes.
RULE060
If the identity of the organism is bacteroides
Then I recommend therapy be chosen from among the following:
CHLORAMPHENICOL, CLINDAMYCIN, RIFAMPIN, ERYTHROMYCIN, CARBENICILLIN
TETRACYCLINE, LINCOMYCIN

(f) **** IS BLOOD A STERILE SITE?**
Yes, blood is one of those sites that are normally sterile.

(g) **** WHAT ARE THE ORGANISMS ASSOCIATED WITH THE GASTRO-INTESTINAL TRACT?**
The likely pathogens associated with the gastro-intestinal tract are: ENTEROCOCCUS, CLOSTRIDIUM-GANGRENE, BACTEROIDES, KLEBSIELLA, PSEUDOMONAS, E.COLI, ENTEROBACTER, PROTEUS.

FIGURE 5-10 Examples of natural language question-answering capabilities. Questions (a)–(d) reference a specific consultation, while (e)–(g) are general inquiries answered from the system's knowledge base.

Our initial efforts at explanation and question answering were based on three capabilities: (1) to display on demand during the consultation the rule currently being invoked, (2) to record rules that were invoked, and, after the consultation, to be able to associate specific rules with specific events (questions and conclusions) to explain why each of them happened, and (3) to search the knowledge base for a specific type of rule in answer to inquiries from the user. The first of these could be easily implemented via the single-word command format described below.

The latter two were intended for use after the consultation and hence were provided with a simple natural language front end. Examples are shown in Figure 5-10 [additional examples can be found in Shortliffe et al., (1975)]. Note that the capability for answering questions of type (2) has been extended to include inquiries about actions the program *failed* to

take [question (d), Figure 5-10]. This is based on the ability of the explanation system to simulate the control structure of the consultation system and can be extremely useful in deciphering the program's behavior. For questions of type (3) [question (e) in Figure 5-10] the search through the knowledge base is directed by a simple parsing of the question into a request for a set of rules, with constraints on premise and/or action contents. The retrieval of relevant rules is guided primarily by preestablished (but automatically generated) lists that indicate premise and action contents.

Some generalization of and extensions to the methodology of (1) and (2) have been motivated by two shortcomings. Displaying the current rule is not particularly informative if the rule is essentially definitional and hence conceptually trivial. The problem here is the lack of a good gauge for the amount of information in a rule. Recording individual rule invocations, questions, and conclusions is useful, but, as a record of individual events, it fails to capture the context and ongoing sequence. It is difficult therefore to explain any event with reference to anything but the specific information recorded with that event.

Two related techniques were developed to solve these problems. First, to provide a metric for the amount of information in a rule, we use (in a very rough analogy with information theory) the function $(-\log \mathrm{CF})$. Rules that are definitional ($\mathrm{CF} = 1$) have by this measure no information, while those that express less obvious implications have progressively more information. The measure is clearly imperfect, since, first, CF's are not probabilities, and there is thus no formal justification that $(-\log \mathrm{CF})$ is a meaningful measure. Second, any sophisticated information content measure should factor in the state of the observer's knowledge, since the best explanations are those that are based on an understanding of what the observer fails to comprehend. Despite these shortcomings, however, this heuristic has proved to be quite useful.

To solve the second problem (explaining events in context) the process of recording individual rule invocations has been generalized: all the basic control functions of the system have been augmented to leave behind a history of each of their executions. This internal trace is then read by various parts of the explanation system to provide a complete, in-context explanation of any part of the system's behavior.

Because the consultation process is essentially one of search through an AND/OR goal tree, inquiries during the course of a consultation fall quite naturally into two types: WHY a question was asked, and HOW a conclusion was (or will be) reached. The first of these looks "up" the goal tree, in examining higher goals, while the second looks "down" in examining rules that may help achieve a goal. This part of the system's explanation capability can thus be viewed in general as a process of tree traversal. By combining this concept with the information content metric, we make possible explanations in varying levels of detail: the tree may be traversed in steps whose information content size is specified by the user, rather than simply stepping from goal to goal. At the start of the traversal process,

In this series, how many of the cultures were positive for the e.coli (ORGANISM-1)? *{typical system prompt}*

**** WHY** *{request from user to determine motivation for question}*

[i.e. WHY is it important to determine the number of cultures which were positive for the organism?] *{system's rephrasing of the WHY question, and answer}*

[1.0] This will help to determine whether there is significant disease associated with this occurrence of the organism.
It has already been established that
[1.1] the culture was taken from a sterile source
[1.2] the number of cultures in the series is greater than 2
Therefore, if
[1.3] the number of cultures which were positive for ORGANISM-1 is greater than 2
Then there is suggestive evidence (.5) that there is significant disease associated with this occurrence of the e.coli (ORGANISM-1).
[RULE049]

**** WHY** *{repeated WHY to determine relevance of previous explanation}*

[i.e. WHY is it important to determine if there is significant disease associated with this occurrence of the organism?]

[2.0] This will help to determine a diagnosis for which the infection should be treated.
It has already been established that
[2.1] the infection is not one of: primary-bacteremia, meningitis
[2.2] for some current culture of the patient it is true that the site of the culture is blood
Therefore, if
[2.3] there is significant disease associated with this occurrence of the organism
Then
it is definite (1.0) that the diagnosis for which the infection should be treated is secondary-bacteremia
[RULE103]

FIGURE 5-11 Examples of explanation capabilities. User input is in boldface capital letters and follows the double asterisks. Expansion of each WHY (or HOW) question (enclosed in square brackets) is produced by the system to ensure that the user is aware of the system's interpretation of the query.

"information distance" from the current goal to the top of the tree is normalized to 9. The argument to the WHY command (an integer between 1 and 10, assumed to be 1 if absent) is then taken to indicate some part of that distance. Thus WHY indicates an explanation that encompasses approximately one-third of the total "conceptual" distance. Repeated WHY questions have the effect of stepping "up" the goal tree. Examples are shown in Figure 5-11; additional examples are found in Davis and Lenat (1982). Similarly, HOW commands step "down" the tree. This can result in examining either branches that have already been traversed (as in Figure 5-12) or those that have yet to be tried (in which case the question becomes "HOW will you determine . . .").

** **HOW 1.1** {*the 1.1 refers to the information in the line labeled [1.1] in Figure 5-11*}

[i.e. HOW was it determined that the culture was taken from a sterile source?]

[3.1] RULE001 was used to conclude that there is suggestive evidence (.7) that the culture was taken from a sterile source.

[3.2] RULE022 was used to conclude that there is strongly suggestive evidence (.8) that the culture was taken from a sterile source.

** **HOW 3.1** {*another request from the user*}

[i.e. HOW was RULE001 used?]

It has already been established that

[4.1] the site of the culture is one of: those sites from which the sterility of cultures depends upon the method of collection

[4.2] the method of collection of the culture is one of: the recognized methods of collection associated with the site of the culture, and

[4.3] it is not known whether care was taken in collecting the culture

Therefore

there is strongly suggestive evidence (.8) that the culture was taken from a sterile source

[RULE022]

FIGURE 5-12 Examples of explanation capabilities—HOW questions.

The system's fundamental approach to explanation is thus to display some recap of its internal actions, a trace of its reasoning. The success of this technique is predicated on the claim that the system's basic approach to the problem is sufficiently intuitive that a summary of those actions is at least a reasonable basis from which to start. While it would be difficult to prove the claim in any formal sense, there are several factors that suggest its plausibility.

First, we are dealing with a domain in which inference, and decision making in the face of uncertainty, is a primary task. The use of production rules in an IF/THEN format seems therefore to be a natural way of expressing things about the domain, and the display of such rules should be comprehensible. Second, the use of such rules in a backward-chaining mode is, we claim, a reasonably intuitive scheme. *Modus ponens* is a well understood and widely (if not explicitly) used mode of inference. Thus the general form of the representation and the way it is employed should not be unfamiliar to the average user. More specifically, however, consider the source of the rules. They have been given to us by human experts who were attempting to formalize their own knowledge of the domain. As such, they embody accepted patterns of human reasoning, implying that they should be relatively easy to understand, especially for those familiar with the domain. As such, they will also attack the problem at what has been

judged an appropriate level of detail. That is, they will embody the right size "chunks" of the problem to be comprehensible. We are not, therefore, recapping the binary bit–level operations of the machine instructions for an obscure piece of code. We claim instead to be working with primitives and a methodology whose substance, level of detail, and mechanism are all well suited to the domain and to human comprehension, precisely because they were provided by human experts. This approach provides what may plausibly be an understandable explanation of system behavior.

This use of symbolic reasoning is one factor that makes the generation of explanations an easier task. For example, it makes the display of a backtrace of performance comprehensible (as, for example, in Figure 5-11). The basic control structure of the consultation system is a second factor. The simple depth-first search of the AND/OR goal tree makes HOW, WHY, and the tree traversal approach natural (as in Figures 5-11 and 5-12). We believe several concepts in the current system are, however, fairly general in purpose and would be useful even in systems that did not share these advantages. Whatever control structure is employed, the maintenance of an internal trace will clearly be useful in subsequent explanations of system behavior. The use of some information metric will help to ensure that those explanations are at an appropriate level of detail. Finally, the explanation-generating routines require some ability to decipher the actions of the main system.

By way of contrast, we might try to imagine how a program based on a statistical approach could explain itself. Such systems can, for instance, display a disease that has been deduced and a list of relevant symptoms, with prior and posterior probabilities. No more informative detail is available, however. When the symptom list is long, it may not be clear how each of the symptoms (or some combination of them) contributed to the conclusion. It is more difficult to imagine what sort of explanation could be provided if the program were interrupted with interim queries while in the process of computing probabilities. The problem, of course, is that statistical methods are not good models of the actual reasoning process [as shown in the psychological experiments of Edwards (1968) and Tversky and Kahneman (1974)], nor were they designed to be. While they are operationally effective when extensive data concerning disease incidence are available, they are also for the most part "shallow," one-step techniques, which capture little of the ongoing process actually used by expert problem solvers in the domain.[5] We have found the presence of even the current

[5]However, the reasoning process of human experts may not be the ideal model for *all* knowledge-based problem-solving systems. In the presence of reliable statistical data, programs using a decision-theory approach are capable of performance surpassing those of their human counterparts. In domains like infectious disease therapy selection, however, which are characterized by judgmental knowledge, statistical approaches may not be viable. This appears to be the case for many medical decision-making areas. See Chapter 2 and Shortliffe and Buchanan (1975) for further discussion of this point.

basic explanation capabilities to be extremely useful, and they have begun to pass the most fundamental test: it has become easier to ask the system what it did than to trace through the code by hand. The continued development and generalization of these capabilities is one focus of our present research.

5.6.3 Knowledge Acquisition

Since the field of infectious disease therapy is both large and constantly changing, it was apparent from the outset that the program would have to deal with an evolving knowledge base. The domain size made writing a complete set of rules an impossible task, so the system was designed to facilitate an incremental approach to competence. New research in the domain produces new results and modifications of old principles, so that a broad scope of capabilities for knowledge-base management was clearly necessary.

As suggested above, a fundamental assumption is that the expert teaching the system can be "debriefed," thus transferring his or her knowledge to the program. That is, presented with any conclusion he or she makes during a consultation, the expert must be able to state a rule indicating all relevant premises for that conclusion. The rule must, in and of itself, represent a valid chunk of clinical knowledge.

There are two reasons why this seems a plausible approach to knowledge acquisition. First, clinical medicine appears to be at the correct level of formalization. That is, while relatively little of the knowledge can be specified in precise algorithms (at a level comparable to, say, elementary physics) the judgmental knowledge that exists is often specifiable in reasonably firm heuristics. Second, on the model of a medical student's clinical training, we have emphasized the acquisition of new knowledge in the context of debugging (although the system is prepared to accept a new rule from the user at any time). We expect that some error on the system's part will become apparent during the consultation, perhaps through an incorrect organism identification or therapy selection. Tracking down this error by tracing back through the program's actions is a reasonably straightforward process that presents the expert with a methodical and complete review of the system's reasoning. He or she is obligated to either approve of each step or correct it. This means that the expert is faced with a sharply focused task of adding a chunk of knowledge to remedy a specific bug. This makes it far easier for the expert to formalize his or her knowledge than would be the case if he or she were told, for example, "tell me about bacteremia."

This methodology has the interesting advantage that the context of the error (i.e., which conclusion was in error, what rules were used, what the facts of this case were, etc.) is of great help to the acquisition system in interpreting the expert's subsequent instructions for fixing the bug. The

error type and context supply the system with a set of expectations about the form and content of the anticipated correction, and this greatly facilitates the acquisition process [details of this and much of the operation of the acquisition system are found in Davis and Lenat (1982)].

The problem of educating the system can be usefully broken down into three phases: uncovering the bug, transferring to the system the knowledge necessary to correct the bug, and integrating the new (or revised) knowledge into the knowledge base. As suggested above, the explanation system is designed to facilitate the first task by making it easy to review all of the program's actions. Corrections are then specified by adding new rules (and perhaps new values, attributes, or contexts) or by modifying old ones. This process is carried out in a mixed-initiative dialogue using a subset of standard English [an early example is found in Shortliffe et al. (1975)].

The system's understanding of the dialogue is based on what may be viewed as a primitive form of "model-directed" automatic programming. Given some natural language text describing one clause of a new rule's premise, the system scans the text to find keywords suggesting which predicate function(s) are the most appropriate translations of the predicate(s) used in the clause. The appropriate template for each such function is retrieved, and the parsing of the remainder of the text is guided by the attempt to fill this in.

If one of the functions were SAME, the template would be as shown in Figure 5-6. CNTXT is known to be a literal, which should be left as is; PARM signifies a clinical parameter (attribute); VALUE denotes a corresponding value. Thus the phrase "the stain of the organism is negative" would be analyzed as follows: the word *stain* in the system dictionary has as part of its semantic indicators the information that it may be used in talking about the attribute *gram stain* of an organism. The word *negative* is known to be a valid value of gram stain (although it has other associations as well). Thus one possible (and in fact the correct) parse is

```
(SAME CNTXT GRAM GRAMNEG)
```

or "the gram stain of the organism is gram-negative."

Note that this is another example of the use of higher-level primitives to do a form of program understanding. It is the semantics of PARM and VALUE that guide the parse after the template is retrieved, and the semantics of the gram stain concept that allow us to ensure the consistency of each parse. Thus by providing semantics and treating such concepts as conceptual primitives at this level we make possible the capabilities shown, using relatively modest amounts of machinery.

Other, incorrect parses are, of course, possible and are generated, too. There are three factors, however, that keep the total number of parses within reasonable bounds. First, and perhaps most important, we are dealing with a very small amount of text. The user is prompted for each clause of the premise individually, and while he or she may type an arbitrary

amount of text at each prompt, the typical response is less than a dozen words. Second, there is a relatively small degree of ambiguity in the semiformal language of medicine. Therefore a keyword-based approach produces only a small number of possible interpretations for each word. Finally, ensuring the consistency of any given parse (e.g., that VALUE is indeed a valid value for PARM) further restricts the total number generated. Typically, between 1 and 15 candidate parses result.

Ranking of possible interpretations of a clause depends on expectation and internal consistency. As noted above, the context of the original error supplies expectations about the form of the new rule, and this is used to help sort the resulting parses to choose the most likely.

As the last step in educating the system, we have to integrate the new knowledge into the rest of the knowledge base. We have only recently begun work on this problem, but we recognize two important general problems. First, the rule set should be free of internal contradictions, subsumptions, or redundancies. The issue is complicated significantly by the judgmental nature of the rules. While some inconsistencies are immediately obvious (two rules that are identical except for differing certainty factors), indirect contradictions (resulting from chaining rules, for example) are more difficult to detect. Inexactness in the rules means that we can specify only an interval of consistent values for a certainty factor.

The second problem is coping with the secondary effects that the addition of new knowledge typically introduces. This arises primarily from the acquisition of a new value, clinical parameter, or context. After the information required to specify the new structure has been requested, it is often necessary to update several other information structures in the system, and these in turn may cause yet other updating to occur. For example, the creation of a new value for the site of a culture involves a long sequence of actions: the new site must be added to the internal list ALLSITES; it must then be classified as either sterile or nonsterile and then be added to the appropriate list; if the site is nonsterile, the user has to supply the names of the organisms that are typically found there, and so forth. While some of this updating is apparent from the structures themselves, much of it is not. We are currently investigating methods for specifying such interactions and a methodology of representation design that minimizes or simplifies the interactions to begin with.

The choice of a production rule representation does impose some limitations in the task of knowledge transfer. Since rules are simple conditional statements, they can at times fail to provide power sufficient to express more complex concepts. In addition, while expressing a single fact is often convenient, expressing a larger concept via several rules is at times somewhat more difficult. As suggested above, mapping from a sequence of actions to a set of rules is not always easy. Goal-directed chaining is apparently not currently a common human approach to structuring larger chunks of knowledge.

Despite these drawbacks, we have found the production rule formalism a powerful one. It has helped to organize and build, in a relatively

short period, a knowledge base that performs at an encouraging level of competence. The rules are, as noted, a reasonably intuitive way of expressing simple chunks of inferential knowledge, and one that requires no acquaintance with any programming language. While it may not be immediately obvious how to restate domain knowledge in production rule format, we have found that infectious disease experts soon acquire some proficiency in doing this with relatively little training. We have had experience working with five different experts over the past few years, and in all cases had little difficulty in introducing them to the use of rules. While this is a limited sample, it does suggest that the formalism is a convenient one for structuring knowledge for someone unfamiliar with programming.

The rules also appear capable of embodying appropriately-sized chunks of knowledge and of expressing concepts that are significant statements. They remain, however, straightforward enough to be built from relatively simple compositions of conceptual primitives (the attributes, values, etc.). While any heavily stylized form of coding of course makes it easier to produce code, stylizing in the form of production rules in particular also provides a framework that is structurally simple enough to be translatable into simple English. This means that the experts can easily comprehend the program's explanation of what it knows, and can equally easily specify knowledge to be added.

5.7 Conclusions

The MYCIN system has begun to approach its design goals of competence and high performance, flexibility in accommodating a large and changing knowledge base, and ability to explain its own reasoning. Successful applications of our control structure with rules applicable to other problem areas have been (a) fault diagnosis and repair recommendations for bugs in an automobile horn system (van Melle, 1974), (b) a consultation system for industrial assembly problems (Hart, 1975), and (c) part of the basis for an intelligent terminal system (Anderson and Gillogly, 1977).

A large factor in this work has been the production rule methodology. It has proved to be a powerful, yet flexible, representation for encoding knowledge and has contributed significantly to the capabilities of the system.

ACKNOWLEDGMENTS

The work reported here was funded in part by grants from the Bureau of Health Sciences Research and Evaluation (grant HS01544) and NIH (grant GM 29662), from the Advanced Research Projects Agency under ARPA

contract DAHC15-73-C-8435, and from the Medical Scientist Training Program (NIH grant GM-81922).

The MYCIN system has been developed by the authors in collaboration with: Drs. Stanley Cohen, Stanton Axline, Frank Rhame, Robert Illa, and Rudolpho Chavez-Pardo, all of whom provided medical expertise; William van Melle, who made extensive revisions to the system code for efficiency and to introduce new features; Carlisle Scott, who (with William J. Clancey) designed and implemented the expanded natural language question-answering capabilities.

6

Towards the Simulation of Clinical Cognition: Taking a Present Illness by Computer

Stephen G. Pauker, G. Anthony Gorry, Jerome P. Kassirer, and William B. Schwartz

Remarkably little is known about the cognitive processes employed in the solution of clinical problems. This paucity of information is probably accounted for in large part by the lack of suitable analytic tools for the study of the physician's thought processes. In the following early work, which arose from Gorry's observations outlined in Chapter 2, Pauker and his colleagues report on the use of the computer as a laboratory for the study of clinical cognition.

Their experimental approach consisted of several elements. First, cognitive insights gained from the study of clinicians' behavior were used to develop PIP, a computer program designed to take the present illness of a patient with edema. The program was then tested with a series of prototypical cases, and the present illnesses generated by the computer were compared to those taken by the clinicians in their group. Discrepant behavior on the part of the program was taken as a stimulus for further refinement of the evolving cognitive theory of the present illness. Corresponding refinements were made in the program, and the process of testing and revision was continued until the program's behavior closely resembled that of the clinicians.

The advances in computer science that made this kind of effort possible included goal-directed programming, pattern matching, and a large associative memory, all of which were products of research in the AI field.

From *American Journal of Medicine*, 60: 981–996 (1976). Used with permission.

The information used by the program is organized in a highly connected set of associations, which are then used to guide such activities as checking the validity of facts, generating and testing hypotheses, and constructing a coherent picture of the patient. As the program pursues its interrelated goals of information gathering and diagnosis, it uses knowledge of diseases and pathophysiology, as well as limited "common sense," to assemble dynamically many small problem-solving strategies into an integrated history-taking process.

Although the work was preliminary and aimed more at understanding cognitive processes and the related computer science issues than at short-term development of a clinical tool, PIP provided important new insights regarding the links among cognitive psychology, computer science, and the expertise of clinical problem solving. The article is also noteworthy because it represented the first time that the concepts of artificial intelligence appeared in a clinical medical journal. In addition, the research challenges that grew out of the PIP work have to a large extent defined the research directions of the AIM researchers at Tufts–New England Medical Center and M.I.T. in subsequent years.

6.1 Introduction

During the last decade there has been increasing interest in the use of the computer as an aid to both clinical diagnosis and management. Programs have been written that can carry out a review of systems (Slack et al., 1966), guide in the evaluation of acid-base disorders (Bleich, 1969; 1972), recommend the appropriate dose of digitalis (Peck et al., 1973; Jelliffe et al., 1972), and weigh the risks and benefits of alternative modes of treatment (Gorry et al., 1973). Some of these programs have been used to a limited extent in clinical practice, whereas others are prototypes that, although not yet of practical value, offer promise for the future. All, however, have the underlying characteristics that they are highly structured and that they deal with well-defined, sharply constrained problems. In nearly all instances, the use of a formalism, such as a flow chart (Slack et al., 1966; Bleich, 1969; 1972), decision analysis (Gorry et al., 1973), or a mathematical algorithm (Peck et al., 1973; Jelliffe et al., 1972), is the guiding principle used to capture clinical expertise in the computer.

There are, however, aspects of clinical medicine that cannot be reduced to formalisms, that is, situations in which a fixed recipe cannot provide the skilled guidance of the experienced clinician. To deal with this class of problems, new and more flexible strategies are under development, but work on such strategies is still in its embryonic phase (Shortliffe et al., 1973; Kulikowski et al., 1973; Pople and Werner, 1972).

In this paper, we report on the development of a computer program that uses unstructured problem-solving techniques to take the history of the present illness of a patient with edema.[1] We have chosen the problem of present illness for investigation because it is prototypical of clinical problems that demand complex problem-solving strategies. The present illness is, furthermore, the keystone on which a physician builds his or her diagnosis and bases many subsequent management decisions. Although we have examined only a limited range of issues in the present program, we believe that our effort is a first step toward a full understanding of the way in which a physician carries out the history-taking process.

6.2 Computer Science in the Study of Clinical Cognition

Our attempt to simulate the unstructured problem-solving processes of the present illness falls into the domain of computer science known as *artificial intelligence*. Research in this field is concerned with producing computer programs that exhibit behavior that would be termed *intelligent* if such behavior were that of a person. Examples of such work are programs that, to a limited extent, understand English, make sense of certain kinds of visual scenes, and control the operations of robots (Winston, 1974). Such research has been underway for 20 years (Feigenbaum, 1963) and, during this time, some major lessons have been learned. Perhaps the most important discovery has been that formalisms alone, for example, cybernetics (Bell, 1962), mathematical logic (McCarthy, 1968), and information theory (Shannon and Weaver, 1949), cannot produce intelligent behavior in complex, real-world situations. It has become abundantly clear that no single, formal approach can accommodate the knowledge of first principles and the experience, common sense,[2] and guesswork (Minsky, 1975) required for "intelligent" activities.

Because of the obvious competence of people in carrying out activities that formalisms cannot, artificial intelligence researchers have turned more recently to the study of human problem solving (Winston, 1974; Minsky, 1968). The study of natural intelligence, in fact, has become the central activity of artificial intelligence, and the experimental method of the field now emphasizes the use of computer systems as laboratories in which the-

[1]The program described here should be contrasted with the well-bounded "present illness algorithms" (Stead et al., 1972), which rely on flow charts for their implementation and which refer the patient to the physician for further questioning whenever the situation appears to be complex or serious.

[2]By common sense, we mean all the ordinary, rather pedestrian knowledge about everyday occurrences that is possessed by reasonably intelligent people.

ories of human problem solving can be represented and tested (Newell and Simon, 1972).

Conventional computer-programming concepts and structures have proven inadequate to express complex theories of human problem solving; however, new techniques have been developed that ameliorate these technological difficulties. Greatly improved systems have been created for managing very large collections of facts, and new goal-directed programming languages have been designed for utilizing these facts in the solution of difficult problems. Through the appropriate statement of goals, it is possible to construct a program that brings knowledge to bear *when it is required.* As new facts are obtained, such programs can dynamically organize many small problem-solving techniques into a coherent strategy that can respond flexibly to the changing picture of the world.[3] Equally important, as we shall discuss, is that these new languages provide means for giving a program *advice* as to when a particular piece of knowledge may be useful and how that knowledge should be applied to particular situations.

We believe that the ideas and technology now emerging from artificial intelligence research should make possible realistic simulations of human problem-solving strategies. In assessing the feasibility of building an "intelligent" program, however, some vital questions must be answered:

What is expert knowledge?
How much knowledge is required?
How should it be organized and how should it be applied?

The answers to these questions will come only from the careful study of real problem domains, and the success of such studies will be determined in large part by the *boundedness* of the problem domain under consideration. We believe that medicine, with its highly developed taxonomy, its codified knowledge base, the generally repetitive nature of the problem-solving encounters, and the existence of acknowledged experts, constitutes a promising problem domain because of its relatively well-bounded character. We therefore believe that, building on the technology at hand, acceptable progress can be made toward the development of sophisticated

[3]Expressed in technical terms, these languages do not require a detailed, rigid program because of pattern-directed invocation. Each subroutine contains a statement of what it potentially can accomplish, so the programmer need not specify which subroutines (or even that any subroutine) should carry out a desired action. Rather, he or she can specify the desired effect or goal and ask the computer to identify and use those subroutines that appear relevant. This type of program organization has many applications, such as offering heuristic advice and generating hypotheses. As an example of one class of problem that is very difficult to solve with conventional techniques, but that is trivial with this type of language, consider the problem of logical deduction. The program is told "All Greeks are poets" and "Anyone born in Athens is a Greek." We then tell the program that "Constantine was born in Athens" and ask "Is Constantine a poet?" The program automatically deduces the answer, basically using the same process we would. That is, it sets about to find out if Constantine is a poet. It realizes that the way to answer this question is to determine if he is a Greek, and therefore it asks if Constantine was born in Athens. When it discovers that he was, the original question is answered.

systems that can deal competently with complex clinical problems. To achieve such progress, however, the essential first step is to examine in depth the nature of the clinician's cognitive processes.

6.3 Methods of Procedure

Our first efforts were directed toward elucidating a number of the problem-solving strategies that physicians use in taking the history of the present illness of a patient with edema. This analytic effort was carried out through introspection and through direct observations of clinicians' problem-solving behavior. The insights gained in this way were represented as a computer program [using the CONNIVER programming system (Sussman and McDermott, 1972)] that incorporates the goal-directed techniques described in Section 6.2. The program was then tested with a series of prototypical cases in which edema was the presenting problem, and the questioning strategy followed by the program was compared to that of the physicians whom it was intended to simulate.

It immediately became apparent that the program's behavior differed markedly from that of the physicians, but, by examining specific discrepancies, we were able to recognize components of the clinicians' reasoning process that had been misunderstood or neglected in our initial analysis. With these new insights, we revised the program and evaluated its history-taking performance again. With each iteration of this process, the performance of the program improved and our insights into the cognitive process deepened. The study was terminated when the program closely simulated the manner in which the physician members of the team take the present illness of a patient with edema.[4]

6.4 Examples of Computer-Generated Analyses of Present Illnesses

Figure 6-1 presents a portion of a typical dialogue between a user (a physician) and the program. The language of both the questions and the computer-generated summaries (Figures 6-2 through 6-5) is rather stilted because most of our effort has been devoted to examining the history-taking process, not to producing a polished output. Each case demonstrates the program's use of somewhat different overall strategies of history taking.

[4]Our program simulates the behavior of the particular physicians in our group. The question of differences in problem-solving behavior among physicians is one we intend to pursue further.

PRESENTING PROBLEM: A YOUNG MAN WITH MASSIVE PEDAL EDEMA

1A IS PEDAL EDEMA, WHICH IS MASSIVE, 1. FOR-DAYS? 2. FOR-WEEKS? 3. FOR-MONTHS? 4. FOR-YEARS?	*>>1*
1B IS IT, 1. FIRST-TIME? 2. INFREQUENT? 3. OCCASIONAL? 4. FREQUENT?	*>>1*
1C IS IT, 1. WITHOUT-TEMPORAL-PATTERN? 2. WORSE-IN-MORNING? 3. WORSE-IN-EVENING?	*>>3*
1D IS IT, 1. SYMMETRICAL? 2. ASYMMETRICAL?	*>>1*
1E IS IT, 1. PITTING? 2. NON-PITTING?	*>>1*
IF IS IT, 1. ERYTHEMATOUS? 2. NOT ERYTHEMATOUS?	*>>2*
1G IS IT, 1. PAINFUL? 2. NOT PAINFUL?	*>>2*
2 WELL, IS THERE DYSPNEA?	*>>NO*
3 BY THE WAY, IS THERE HEAVY ALCOHOL CONSUMPTION?	*>>NO*
3A IS ALCOHOL CONSUMPTION, 1. NONE? 2. RARE? 3. SOCIAL?	*>>3*
4 TELL ME, IS THERE JAUNDICE?	*>>NO*
5 TELL ME, WAS THERE PROTEINURIA IN THE PAST?	*>>UNKNOWN*
5A IS THERE A LARGE-POLICY LIFE INSURANCE?	*>>NO*
5A1 IS LIFE INSURANCE, 1. NONE? 2. UNABLE-TO-OBTAIN? 3. SMALL-POLICY?	*>>3*
5B IS THERE SERVED-IN ARMED FORCES?	*>>YES*
6 WELL, IS THERE VARICOSE VEINS?	*>>NO*
7 WAS THERE EDEMA IN THE PAST?	*>>NO*
8 WELL, IS THERE ANY HEMATURIA?	*>>NO*
8A IS THERE . . .	*>>NO*

FIGURE 6-1 Case 1. Initial portion of a representative present illness dialogue. The questions were asked by the computer, and the responses (preceded by >>) are those of the user. The complete present illness analysis, as generated by the computer, is shown in Figure 6-2.

PRESENTING PROBLEM: A YOUNG MAN WITH MASSIVE PEDAL EDEMA

THIS CASE CAN BE SUMMARIZED AS FOLLOWS:

THIS IS A YOUNG MAN WHO HAS PEDAL EDEMA WHICH IS NOT-PAINFUL, NOT-ERYTHEMATOUS, PITTING, SYMMETRICAL, WORSE-IN-EVENING, FIRST-TIME, FOR-DAYS AND MASSIVE. HE DOES NOT HAVE DYSPNEA. HE HAS SOCIAL ALCOHOL CONSUMPTION. HE DOES NOT HAVE JAUNDICE. IT IS NOT EXPLICITLY KNOWN WHETHER IN THE PAST HE HAD PROTEINURIA, BUT HE HAS SMALL-POLICY LIFE INSURANCE, AND HE HAS SERVED IN ARMED FORCES. HE DOES NOT HAVE VARICOSE VEINS. IN THE PAST HE DID NOT HAVE EDEMA. HE DOES NOT HAVE HEMATURIA. HE HAS NORMAL BUN. HE HAS NORMAL CREATININE. HE HAS PERI-ORBITAL EDEMA, WHICH IS WORSE-IN-MORNING, FIRST-TIME, FOR-DAYS AND SYMMETRICAL. HE HAS LOW ALBUMIN CONCENTRATION. HE HAS HEAVY PROTEINURIA, WHICH IS >5GRAMS/24HRS. HE HAS MODERATELY-ELEVATED, RISING WEIGHT. IN THE RECENT PAST HE DID NOT HAVE PHARYNGITIS. IN THE RECENT PAST HE HAD NOT-ATTENDED SCHOOL. IN THE RECENT PAST HE HAD NOT-ATTENDED SUMMER CAMP. IN THE RECENT PAST HE HAD NOT BEEN EXPOSED TO STREPTOCOCCI. IN THE RECENT PAST HE DID NOT HAVE FEVER. IT IS SAID, BUT HAS BEEN DISREGARDED, THAT HE HAS RED-CELL-CASTS-IN URINARY SEDIMENT. HE DOES NOT HAVE JOINT PAIN. HE DOES NOT HAVE RASH. HE HAS NEGATIVE ANA. HE DOES NOT HAVE FEVER. HE HAS NOT-RECEIVED ANTIBIOTIC. HE DOES NOT HAVE ANEMIA. IN THE PAST HE DID NOT HAVE HEMATURIA.

DIAGNOSES THAT HAVE BEEN ACCEPTED ARE: NEPHROTIC SYNDROME AND SODIUM RETENTION.

THE LEADING HYPOTHESIS IS IDIOPATHIC NEPHROTIC SYNDROME.

HYPOTHESES BEING CONSIDERED:

	A. fit of case to hypothesis	B. fraction of findings explained by hypothesis	average of A and B
IDIOPATHIC NEPHROTIC SYNDROME	0.80	0.37	0.58
ACUTE GLOMERULONEPHRITIS	0.22	0.27	0.24
HENOCH-SCHOENLEIN PURPURA	0.07	0.10	0.09

FIGURE 6-2 Case 1. Computer-generated summary of present illness of a patient with idiopathic nephrotic syndrome. Note that the diagnosis was not available to the computer; the program was provided only with the description of the presenting problem. The ranking at the bottom of the figure is based on the average of the "fit" of the case of the hypothesis (column A) and the fraction of the findings explained by the hypothesis (column B). For details of the evaluation (scoring) procedure, see text.

Case 1. Figure 6-2 shows the computer-generated summary of Case 1, a patient with idiopathic nephrotic syndrome. The computer was given as the chief complaint "a young man with massive pedal edema." The behavior of the program can be briefly summarized as follows. The computer characterized the edema in detail and, in light of the specific findings, turned to questions designed to elucidate etiology. After quickly determining that there was no history suggestive of congestive heart failure, alcoholic cirrhosis, varicosities, or renal failure, it noted that the patient had several findings strongly suggestive of nephrotic syndrome. The program then initiated a search for causes of the nephrotic syndrome, first exploring the possibility that the patient was suffering from poststrepto-

PRESENTING PROBLEM: A MIDDLE-AGED WOMAN WITH PEDAL EDEMA

THE CASE CAN BE SUMMARIZED AS FOLLOWS:

THIS IS A MIDDLE-AGED WOMAN, WHO HAS PEDAL EDEMA, WHICH IS NOT-PAINFUL, NOT-ERYTHEMATOUS, PITTING, SYMMETRICAL, 4+, WITHOUT-TEMPORAL-PATTERN, OCCASIONAL AND FOR-WEEKS. SHE DOES NOT HAVE DYSPNEA. SHE HAS HEAVY ALCOHOL CONSUMPTION. SHE HAS JAUNDICE. SHE HAS PAINFUL HEPATOMEGALY. SHE HAS SPLENOMEGALY. SHE HAS ASCITES. SHE HAS PALMAR ERYTHEMA. SHE HAS SPIDER ANGIOMATA. SHE DOES NOT HAVE PAROTID ENLARGEMENT. SHE HAS PROLONGED PROTHROMBIN TIME. SHE HAS MODERATELY-ELEVATED SGPT. SHE HAS MODERATELY-ELEVATED SGOT. SHE HAS MODERATELY-ELEVATED LDH. SHE HAS NOT RECEIVED BLOOD TRANSFUSIONS. SHE HAS NOT EATEN CLAMS. SHE DOES NOT HAVE ANOREXIA. SHE HAS MELENA. SHE DOES NOT HAVE HEMATEMESIS. SHE HAS LOW SERUM IRON. SHE HAS ESOPHAGEAL VARICES.

DIAGNOSES THAT HAVE BEEN ACCEPTED ARE: ALCOHOLISM AND GI BLEEDING.
THE LEADING HYPOTHESIS IS CIRRHOSIS.

HYPOTHESES BEING CONSIDERED:

	A. fit of case to hypothesis	B. fraction of findings explained by hypothesis	average of A and B
CIRRHOSIS	0.72	0.78	0.75
HEPATITIS	0.75	0.30	0.53
PORTAL HYPERTENSION	0.72	0.17	0.45
CONSTRICTIVE PERICARDITIS	0.17	0.13	0.15

FIGURE 6-3 Case 2. Computer-generated summary of the present illness of a patient with cirrhosis of the liver. The format is identical to that of Figure 6-2.

coccal glomerulonephritis and then looking for evidence of a systemic disease such as lupus erythematosus. Finding no evidence of a systemic disorder, the program made the diagnosis of nephrotic syndrome, probably idiopathic in character, but indicated that acute glomerulonephritis remained as a second, albeit much less likely, possibility. Note, incidentally, that the program disregarded the statement that red cell casts had been seen because it concluded that in the absence of hematuria the report of red cell casts was almost certainly in error. Also note that the questions about life insurance and military service were utilized because normal earlier physical examinations can suggest that proteinuria had not been present in the past.

Case 2. Figure 6-3 summarizes the present illness of a patient with Laennec's cirrhosis. The computer was given as the chief complaint "a middle-aged woman with pedal edema." In response, it obtained a detailed description of the character of the edema and then undertook an exploration of possible etiologies. On finding that the patient drank large quantities of alcohol, it turned to cirrhosis as a working hypothesis and quickly uncovered many stigmata of liver disease. The program also briefly ex-

PRESENTING PROBLEM: A YOUNG MAN WITH PEDAL EDEMA AND OLIGURIA

THE CASE CAN BE SUMMARIZED AS FOLLOWS:

THIS IS A YOUNG MAN, WHO HAS OLIGURIA. HE HAS PEDAL EDEMA, WHICH IS NOT-PAINFUL, NOT-ERYTHEMATOUS, PITTING, SYMMETRICAL, WITHOUT-TEMPORAL-PATTERN, FIRST-TIME AND FOR-DAYS. IT HAS BEEN DENIED THAT HE HAS RECENT SCARLET FEVER. IN THE RECENT PAST HE DID NOT HAVE PHARYNGITIS. IN THE RECENT PAST HE HAD NOT-ATTENDED SUMMER CAMP. IN THE RECENT PAST HE HAD NOT-BEEN-EXPOSED-TO STREPTOCOCCI. HE HAS NOT-RECEIVED RADIOGRAPHIC CONTRAST MATERIAL. HE HAS NOT-RECEIVED NEPHROTOXIC DRUGS. IN THE RECENT PAST HE DID NOT HAVE HYPOTENSION. HE HAS MODERATELY-ELEVATED URINE SODIUM. HE HAS URINE SPECIFIC GRAVITY WHICH IS ISOSTHENURIC. HE HAS NO-RED-CELLS-IN, NO-WHITE-CELLS-IN, RENAL-CELLS-IN, NO-RENAL-CELL-CASTS-IN, HYALINE-CASTS-IN URINARY SEDIMENT. IT IS NOT EXPLICITLY KNOWN WHETHER HE HAS BEEN-EXPOSED-TO A CLEANING FLUID. HE DOES NOT HAVE HYPOTENSION. HE DOES NOT HAVE TACHYCARDIA. HE HAS NORMAL TURGOR-AND-PERFUSION-OF SKIN. HE HAS MODERATELY-ELEVATED, RISING WEIGHT.

DIAGNOSES THAT HAVE BEEN ACCEPTED ARE: SODIUM RETENTION, EXPOSURE TO NEPHROTOXINS, EXPOSURE TO HEPATOTOXINS AND ACUTE RENAL FAILURE.

THE LEADING HYPOTHESIS IS ACUTE TUBULAR NECROSIS.

HYPOTHESES BEING CONSIDERED:

	A. fit of case to hypothesis	B. fraction of findings explained by hypothesis	average of A and B
ACUTE TUBULAR NECROSIS	0.50	0.37	0.43
ACUTE GLOMERULONEPHRITIS	0.20	0.21	0.20
IDIOPATHIC NEPHROTIC SYNDROME	0.18	0.16	0.17
CHRONIC GLOMERULONEPHRITIS	0.19	0.11	0.15

FIGURE 6-4 Case 3. Computer-generated summary of the present illness of a patient with acute tubular necrosis. The format is identical to that of Figure 6-2.

plored other etiologies of liver disease, such as the hepatitis induced by transfusions or by the ingestion of raw shellfish, but could find no evidence in support of these diagnoses. It then returned to the primary hypothesis of cirrhosis and, in searching for possible complications, noted the presence of both esophageal varices and chronic gastrointestinal bleeding. It concluded that the patient had alcoholic cirrhosis and that hepatitis was an alternative, but much less likely, possibility.

Case 3. Figure 6-4 shows the computer-generated summary of a patient with acute tubular necrosis produced by carbon tetrachloride exposure. The computer was given as the chief complaint "a young man with edema and oliguria." The program immediately undertook a search for causes of acute renal failure. It first focused on the diagnosis of acute glomerulonephritis but could find no evidence of streptococcal exposure. It next explored the possibility of acute tubular necrosis but was unable to find an etiological factor. When the program later assessed the characteristics of the urine sediment, however, it noted many hallmarks of tubular

PRESENTING PROBLEM: A MIDDLE-AGED MAN WITH ASCITES AND PEDAL EDEMA

THE CASE CAN BE SUMMARIZED AS FOLLOWS:

THIS IS A MIDDLE-AGED MAN WHO HAS ASCITES. HE HAS PEDAL EDEMA, WHICH IS NOT-PAINFUL, NOT-ERYTHEMATOUS, PITTING, SYMMETRICAL, WORSE-IN-EVENING, OCCASIONAL AND FOR-MONTHS. HE HAS SOCIAL ALCOHOL CONSUMPTION. HE HAS HEPATOMEGALY. HE DOES NOT HAVE JAUNDICE. HE DOES NOT HAVE PALMAR ERYTHEMA. HE DOES NOT HAVE SPIDER ANGIOMATA. HE DOES NOT HAVE PAROTID ENLARGEMENT. HE DOES NOT HAVE GYNECOMASTIA. HE DOES NOT HAVE TESTICULAR ATROPHY. HE HAS NORMAL BILIRUBIN. HE HAS NORMAL PROTHROMBIN TIME. HE HAS NORMAL SGPT. HE HAS NORMAL SGOT. HE HAS CHEST PAIN WHICH IS RELIEVED-BY-SITTING-UP, WITHOUT-RADIATION, MODERATE, OCCASIONAL, FOR-SECONDS AND SHARP. HE HAS EXERTIONAL DYSPNEA. HE HAS ORTHOPNEA. HE DOES NOT HAVE PAROXYSMAL NOCTURNAL DYSPNEA. HE HAS ELEVATED NECK VEINS. HE HAS KUSSMAUL'S SIGN. HE HAS PERICARDIAL KNOCK. HE HAS DISTANT HEART SOUNDS. HE HAS PERICARDIAL-CALCIFICATION-ON, NORMAL-HEART-SIZE-ON, CLEAR-LUNG-FIELDS-ON CHEST XRAY.

THE LEADING HYPOTHESIS IS CONSTRICTIVE PERICARDITIS.

HYPOTHESES BEING CONSIDERED:

	A. fit of case to hypothesis	B. fraction of findings explained by hypothesis	average of A and B
CONSTRICTIVE PERICARDITIS	0.78	0.50	0.64
CONGESTIVE HEART FAILURE	0.44	0.21	0.32

FIGURE 6-5 Case 4. Computer-generated summary of the present illness of a patient with constrictive pericarditis. The format is identical to that of Figure 6-2.

injury. Pursuing this lead, it soon uncovered an exposure to a cleaning fluid that it presumed contained carbon tetrachloride. It then explored the possibility that acute hypotension had also contributed to the development of the oliguria but could obtain no evidence in support of this hypothesis. Finally, it determined that body weight was increasing, and from this fact concluded that the patient was retaining sodium. Because the data base does not currently include the distinction between the retention of salt and the retention of free water, the program could not arrive at the correct interpretation of the weight gain, namely, that the overhydration was due to water retention *per se*.

Case 4. Figure 6-5 gives the summary of the present illness of a man with constrictive pericarditis secondary to tuberculosis. The program was given as the chief complaint "a middle-aged man with ascites and pedal edema." After further characterizing the edema, the computer focused on a hepatic etiology and found that the patient had an enlarged liver. Although subsequent questioning revealed only social alcohol consumption, the program persisted in its search for stigmata of cirrhosis. When none was found, it turned to a possible cardiac etiology and noted that chest pain was a prominent complaint; the pain was, however, more characteristic of pleural or pericardial than of myocardial disease. It next found that

there was both neck vein distention and orthopnea, but that there was no paroxysmal nocturnal dyspnea. These clinical findings, in combination with the ascites, suggested the diagnosis of pericardial disease. Further questioning then revealed many of the stigmata of constrictive pericarditis.

Because even the experienced clinician often confuses constrictive pericarditis with cirrhosis, it is understandable why the diagnosis of cirrhosis was pursued with such vigor. Note, however, that the program was deficient in that it failed to explore other etiologies of predominantly right-sided cardiac failure, such as cor pulmonale and multiple pulmonary emboli; this shortcoming is explained by the fact that the current knowledge base does not include information about these latter diagnoses.

6.5 Nature of the Underlying Computer Programs

In this section we shall first discuss the overall behavior of the program in terms of its major components and the way that these components interact. We shall then consider in detail the underlying processes used by the program.

6.5.1 An Overview of the Present Illness Program

In taking a history of the present illness, the program, much like the physician, tries to develop a sufficient "understanding" of the patient's complaints to form a reasonable basis on which to evaluate the clinical problem and to lay the groundwork for subsequent management decisions. It accomplishes this goal by undertaking two processes: information gathering and diagnosis. Although these two threads of the problem-solving process are interwoven, for clarity of exposition we shall consider them separately.

By *information gathering,* we mean the accumulation of a profile of data concerning the patient. Because there are innumerable facts that could be gathered, one needs a sharp focus for this activity. This focus is obtained through the pursuit of a small set of diagnostic hypotheses that are suggested by the presenting complaints.

The process of *diagnosis,* in contrast, is an attempt to infer the meaning of a constellation of given findings and does not involve the acquisition of additional information about the patient; rather, it is concerned with the processing of the available facts. When additional findings are required, the diagnostic process turns again to the information-gathering process. Thus the history-taking process is directed both at establishing *what the facts are* and at establishing *what the facts mean* (Feinstein, 1967).

In taking a present illness, our program uses the chief complaint to generate hypotheses about the patient's condition. It also actively seeks additional clinical information to accomplish a number of different tasks, including testing hypotheses and eliminating unlikely ones. Any of these activities may spawn further tasks, such as checking the validity of a newly discovered fact or asking about related findings. As will become evident, however, this brief description understates both the complexity of the program's behavior and the differences between this program and others previously reported.

6.5.2 The Basic Components of the Program

The complexity of the program's behavior is the result of the interaction of the four factors schematically shown in Figure 6-6: (1) the patient-specific data, (2) the supervisory program, (3) the short-term memory, and (4) the long-term (associative) memory.

1. *The patient-specific data.* These are the facts provided by the user either spontaneously or in response to questions asked by the program. These data comprise the computer's knowledge about the patient.

2. *The supervisory program.* The supervisory program guides the computer in taking the present illness and oversees the operation of various subprocesses, such as selecting questions, seeking and applying relevant advice, and processing algorithms (such as flow charts). The principal goal of this supervisor is to arrive at a coherent formulation of the case, by quickly generating and testing hypotheses and by excluding competing hypotheses. At the present time, there are about 300 potential questions that relate to over 150 different concepts that the program can employ in its information-gathering activities.

3. *The short-term memory.* The short-term memory is the site in which data about the patient interact with general medical knowledge that is kept in long-term memory (see below). The supervisory program determines which aspects of this general knowledge enter the short-term memory and how such knowledge is melded with the patient-specific data that are under consideration. The amount of information in short-term memory is quite variable, depending on the complexity of the case and the number of active hypotheses. For a simple case, the short-term memory might contain only two or three hypotheses and the knowledge and deductions associated with them. In a complex or puzzling case, it might contain five or ten hypotheses.

4. *The long-term (associative) memory.* The long-term memory contains a rich collection of knowledge, organized into packages of closely related

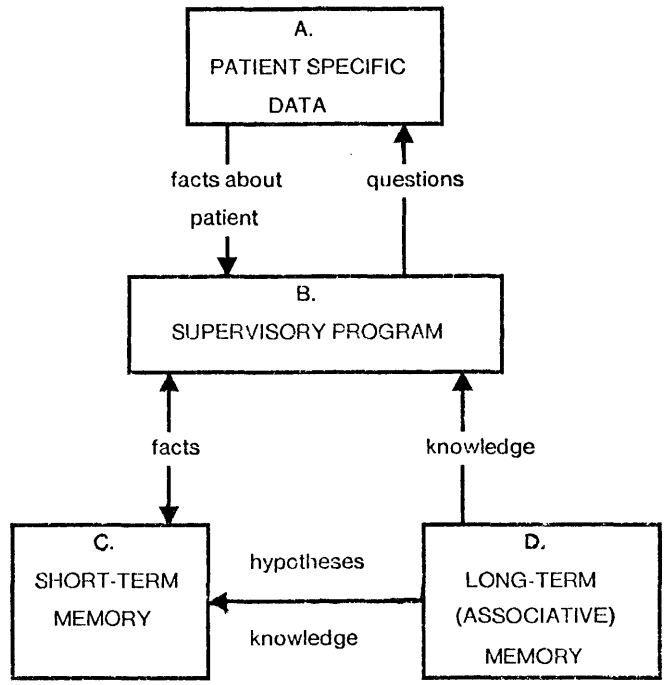


FIGURE 6-6 Overview of program organization. Clinical data (A) are presented to the supervisory program (B), which places them in short-term memory (C). The supervisory program, after consulting both short-term (C) and long-term memories (D), generates hypotheses and moves the information associated with these hypotheses from long-term to short-term memory. The supervisory program then asks for additional patient-specific data relevant to its hypotheses. At every stage, each hypothesis is evaluated (scored) by the program to determine whether it should be rejected, accepted, or considered further.

facts called *frames* (Minsky, 1975). Frames are centered around diseases (such as acute glomerulonephritis), clinical states (such as nephrotic syndrome), or physiologic states (such as sodium retention). Within each frame is a rich knowledge structure that includes prototypical findings (signs, symptoms, laboratory data), the time course of a given illness, and rules for judging how closely a given patient might match the disease or state that the frame describes. A typical example of a frame (nephrotic syndrome) is shown in Figure 6-7.

As shown in Figure 6-8, the frames are linked into a complex *network.* In the figure each frame is represented as a shaded sphere (diseases are

NAME: NEPHROTIC SYNDROME
IS-A-TYPE-OF: CLINICAL STATE
FINDING: LOW SERUM ALBUMIN CONCENTRATION
FINDING: HEAVY PROTEINURIA
FINDING: >5GRAMS/24HRS PROTEINURIA
FINDING: MASSIVE SYMMETRICAL EDEMA
FINDING: EITHER FACIAL OR PERI-ORBITAL AND SYMMETRICAL EDEMA
FINDING: HIGH SERUM CHOLESTEROL CONCENTRATION
FINDING: URINE LIPIDS PRESENT
MUST-NOT-HAVE: PROTEINURIA ABSENT
IS-SUFFICIENT: BOTH MASSIVE PEDAL EDEMA AND >5GRAMS/24HRS PROTEINURIA
MAJOR SCORING:
    SERUM ALBUMIN CONCENTRATIONS
        LO: 1.0
        HIGH: -1.0
    PROTEINURIA:
        >5GRAMS/24HRS: 1.0
        HEAVY: 0.5
        EITHER ABSENT OR LIGHT: -1.0
    EDEMA:
        MASSIVE AND SYMMETRICAL: 1.0
        NOT MASSIVE BUT SYMMETRICAL: 0.5
        ERYTHEMATOUS: -0.2
        ASYMMETRICAL: -0.5
        ABSENT: -1.0
MINOR SCORING:
    SERUM CHOLESTEROL CONCENTRATION:
        HIGH: 1.0
        NOT HIGH: -1.0
    URINE LIPIDS:
        PRESENT: 1.0
        ABSENT: -0.5
MAY-BE-CAUSED-BY:
    ACUTE GLOMERULONEPHRITIS,
    CHRONIC GLOMERULONEPHRITIS,
    NEPHROTOXIC DRUGS,
    INSECT BITE,
    IDIOPATHIC NEPHROTIC SYNDROME,
    SYSTEMATIC LUPUS ERYTHEMATOUS, OR
    DIABETES MELLITUS
MAY-BE-COMPLICATED-BY:
    HYPOVOLEMIA
    CELLULITIS
MAY-BE-CAUSE-OF: SODIUM RETENTION
DIFFERENTIAL DIAGNOSIS:
    IF NECK VEINS ELEVATED, CONSIDER: CONSTRICTIVE PERICARDITIS
    IF ASCITES PRESENT, CONSIDER: CIRRHOSIS
    IF PULMONARY EMBOLI PRESENT, CONSIDER: RENAL VEIN THROMBOSIS

FIGURE 6-7 A typical frame. Information about a disease, a physiologic state, etc., is stored in the form of a frame within the long-term memory. Included in a typical frame, as shown here for nephrotic syndrome, are descriptions of typical findings, numerical factors to be used in scoring, and links to other frames (e.g., MAY-BE-CAUSED-BY, MAY-BE-COMPLICATED-BY). There are also rules for excluding (MUST-NOT-HAVE) and satisfying (IS-SUFFICIENT) the fit of the frame to the case at hand. For further details, see text.

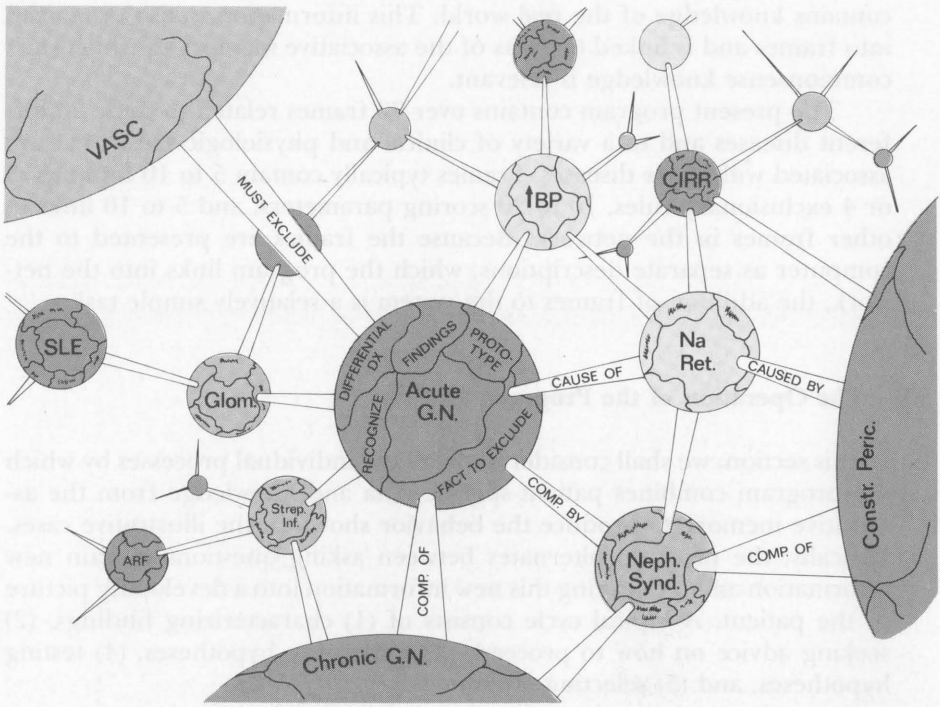


FIGURE 6-8 The long-term (associative) memory. The long-term memory consists of a rich collection of knowledge about diseases, signs, symptoms, pathologic states, real-world situations, etc. Each point of entry into the memory allows access to many related concepts through a variety of associative links shown as rods. Each rod is labeled to indicate the kind of association it represents. Note that the dark gray spheres denote disease states, medium gray spheres denote clinical states (e.g., nephrotic syndrome) and light gray spheres denote physiologic states (e.g., sodium retention). Abbreviations used in this figure are Acute G.N. = acute glomerulonephritis, Chronic G.N. = chronic glomerulonephritis, VASC = vasculitis, CIRR = cirrhosis, Constr. Peric. = constrictive pericarditis, ARF = acute rheumatic fever, Na Ret. = sodium retention, SLE = systematic lupus erythematosus, ↑BP = acute hypertension, Glom. = glomerulitis, Strep. Inf. = streptococcal infection, Neph. Synd. = nephrotic syndrome.

dark gray, clinical states are medium gray, and physiologic states are light gray), and the links between the frames are represented as labeled rods. These links depict a variety of relations, such as MAY-BE-CAUSED-BY and MAY-BE-COMPLICATED-BY.

In addition to information about diseases and physiology, the network contains knowledge of the real world. This information is also organized into frames and is linked to areas of the associative memory in which such commonsense knowledge is relevant.

The present program contains over 70 frames related to some 20 different diseases and to a variety of clinical and physiologic states that are associated with these diseases. Frames typically contain 5 to 10 findings, 3 or 4 exclusionary rules, 10 to 20 scoring parameters, and 5 to 10 links to other frames in the network. Because the frames are presented to the computer as separate descriptions, which the program links into the network, the addition of frames to the system is a relatively simple task.

6.5.3 The Operation of the Program

In this section, we shall consider in detail the individual processes by which the program combines patient-specific data and knowledge from the associative memory to produce the behavior shown in the illustrative cases. Basically, the program alternates between asking questions to gain new information and integrating this new information into a developing picture of the patient. A typical cycle consists of (1) characterizing findings, (2) seeking advice on how to proceed, (3) generating hypotheses, (4) testing hypotheses, and (5) selecting questions.

Characterizing Findings

After being presented with the chief complaint, the supervisor retrieves from the associative memory a procedure that characterizes that complaint in detail. This procedure is a flow chart that follows a set pattern in eliciting such features as the location, severity, and duration of the complaint. The program uses this detailed description of the complaint to limit the number of hypotheses that it will later have to consider.

Seeking Advice on How to Proceed

One of the most important features of our program is its ability to assemble small history-taking strategies into an overall approach that is tailored to the case at hand. This ability is critically dependent on the availability of appropriate advice about efficient methods for the exploration and organization of the case. Here we shall present three examples of the program's use of this facility:

1. Advice can be given that alerts the supervisor to ask one or more questions that will "zero in" on the presenting problem and thus, at the stage

of hypothesis generation (see below), limit the number of diagnostic possibilities that must be evaluated.

2. Advice can be given that guides the supervisor in its evaluation of information that is being presented. Such validity checks can be of several types. First, the program might point out that a finding itself is clearly in error, e.g., a weight gain of 50 pounds in 48 hours. Second, it might note that new information is inconsistent with other facts known about the patient, e.g., the presence of red cell casts in the absence of hematuria. Finally, it might indicate that a new finding contradicts a conclusion already drawn about the case.[5]
3. Advice can be given that alerts the supervisor to errors that might stem from a patient's misinterpretation of a particular sign or symptom. For example, if a patient complains of "blood in the urine," the supervisor is told that dark urine, which is attributed by the patient to blood, may be caused by the presence of bile, myoglobin, or anthocyanins (from beets).

Hypothesis Generation

After the complaint has been characterized and all relevant advice has been acted upon, the supervisory program proceeds to generate working hypotheses. Hypothesis generation consists of moving frames from long-term memory to short-term memory, where each frame plays a special role in guiding further exploration of the patient's problem. Frames can exist in one of four states: dormant, semiactive, active, and accepted. Initially, the short-term memory contains no frames; all frames are in the long-term memory and are said to be in the *dormant state.* In this nascent condition, however, some of the findings in the frames are associated with small, independent computer programs called *daemons.* A few of these daemons extend like tentacles from the frame into the short-term memory (see Figure 6-9, BEFORE); these are primarily the daemons of those findings that are strongly suggestive of their associated frames. When the matching fact for a daemon is added to the short-term memory, the entire frame attached to the daemon is added to the short-term memory (see Figure 6-9, AFTER). As pointed out, this process is synonymous with forming a hypothesis. Those frames that have entered short-term memory as hypotheses are called *active.* As is reflected in the AFTER half of Figure 6-9, frames one link away from an active frame are also affected in that during the activation process they are pulled closer to short-term memory. Consequently, more tentacles from such frames can reach into memory where they can now watch for their matching facts. These related frames,

[5]The latter two kinds of advice would not be provided in the initial cycle, which deals with the chief complaint, because, at such an early stage, the short-term memory would not contain any detailed information about the patient.

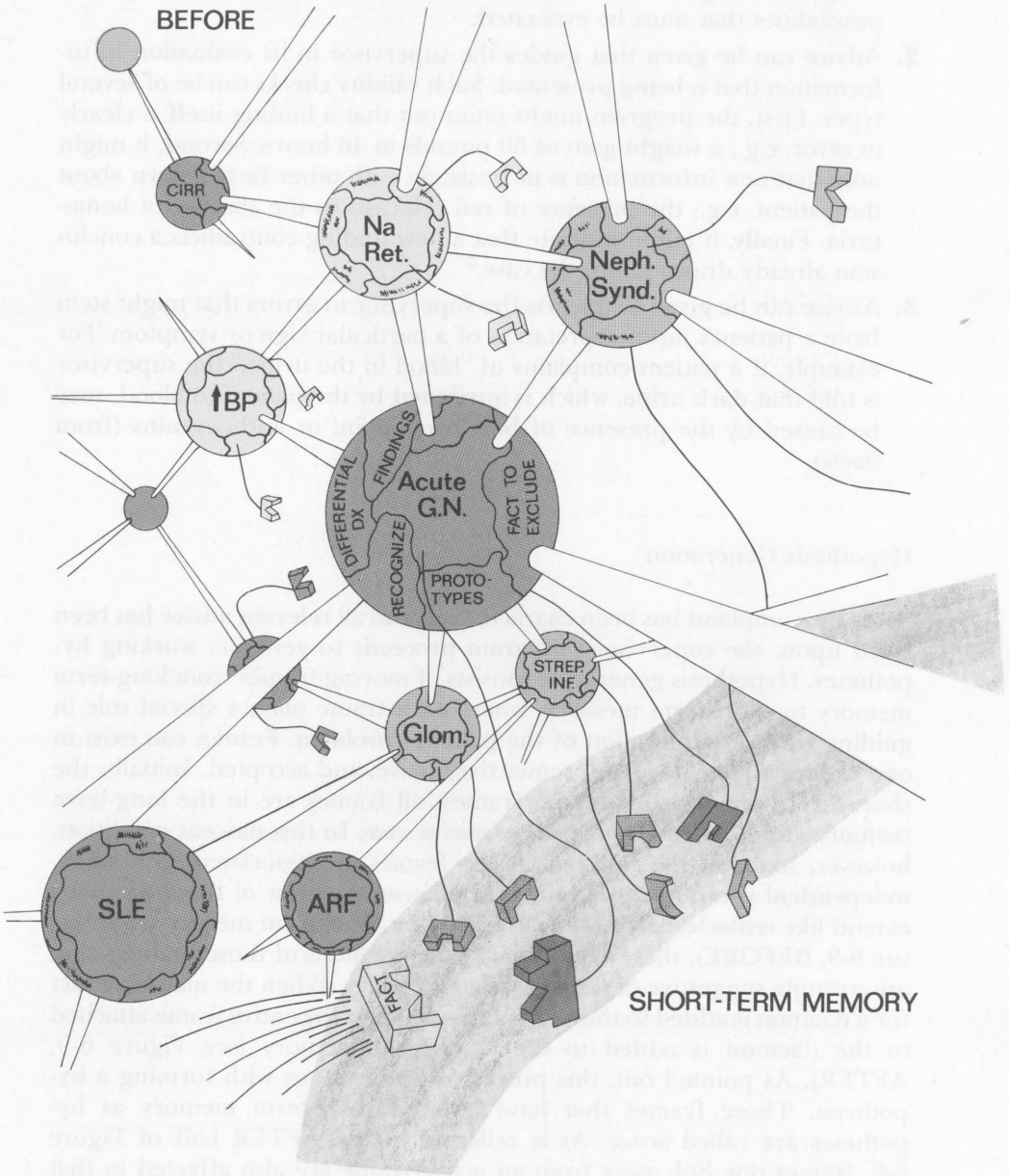
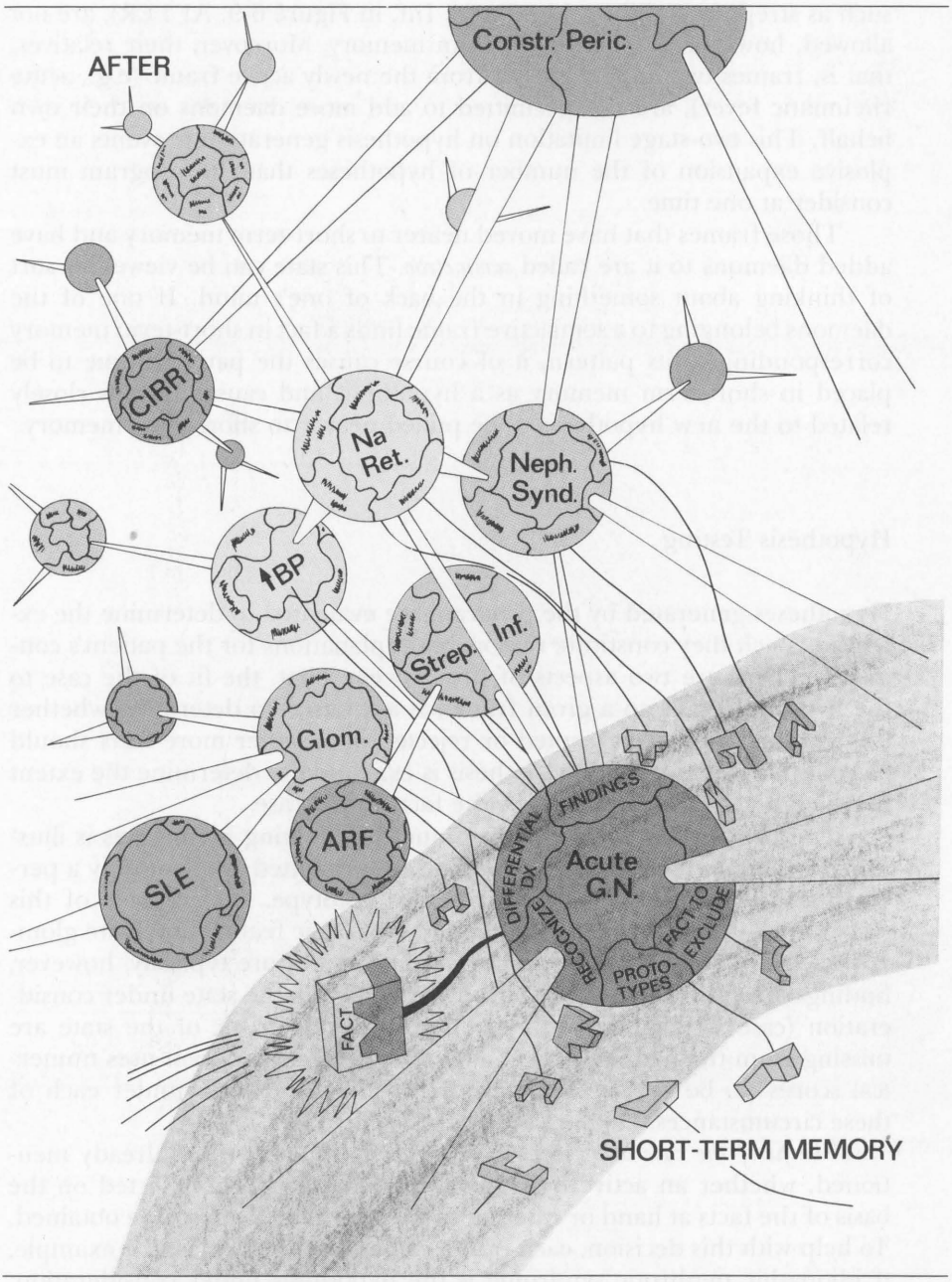


FIGURE 6-9 Hypothesis generation. (The abbreviations are the same as those used in Figure 6-8.) BEFORE: in the nascent condition (when there are no hypotheses in short-term memory), tentacles (daemons) from some frames in long-term memory extend into the short-term memory, where each constantly searches for a matching fact.



AFTER: the matching of fact and daemon causes the movement of the full frame (in this case, acute glomerulonephritis) into short-term memory. As a secondary effect, frames immediately adjacent to the activated frame move closer to short-term memory and are able to place additional daemons therein. Note that, to avoid complexity, the daemons on many of the frames are not shown.

such as streptococcal infection (Strep. Inf. in Figure 6-9, AFTER), are not allowed, however, to enter short-term memory. Moreover, their relatives, that is, frames two links removed from the newly active frame (e.g., acute rheumatic fever), are not permitted to add more daemons on their own behalf. This two-stage limitation on hypothesis generation prevents an explosive expansion of the number of hypotheses that the program must consider at one time.

Those frames that have moved nearer to short-term memory and have added daemons to it are called *semiactive*. This state can be viewed as sort of thinking about something in the back of one's mind. If one of the daemons belonging to a semiactive frame finds a fact in short-term memory corresponding to its pattern, it of course causes the parent frame to be placed in short-term memory as a hypothesis and causes frames closely related to the new hypothesis to be pulled nearer to short-term memory.

Hypothesis Testing

Hypotheses generated by the program are evaluated to determine the extent to which they constitute reasonable explanations for the patient's condition. There are two aspects of this process. First, the fit of the case to the hypothesis (i.e., to a given frame) is appraised to determine whether the hypothesis can be accepted or rejected or whether more facts should be collected. Second, each hypothesis is examined to determine the extent to which it can account for all of the facts in the case.

The problem faced by the program in evaluating hypotheses is illustrated in Figure 6-10. In case A, we have represented schematically a perfect match between patient and disease prototype. An example of this situation would be a patient who has all the classic features of acute glomerulonephritis and no other abnormal findings. More typically, however, findings are present that are not ordinarily seen in the state under consideration (case B, Figure 6-10), or findings characteristic of the state are missing from the patient (case C, Figure 6-10). The program uses numerical scores (to be discussed) to measure the degree of fit under each of these circumstances.

The fit of the case to the hypothesis serves to determine, as already mentioned, whether an active hypothesis can be accepted or rejected on the basis of the facts at hand or whether more information should be obtained. To help with this decision, each frame contains specific rules. For example, if idiopathic nephrotic syndrome is the hypothesis under consideration, and the program then learns that the patient has had gross hematuria, an *exclusionary rule* rejects the hypothesis and permanently removes the nephrotic syndrome from short-term memory. On the other hand, if the patient has both edema and massive proteinuria (protein excretion of

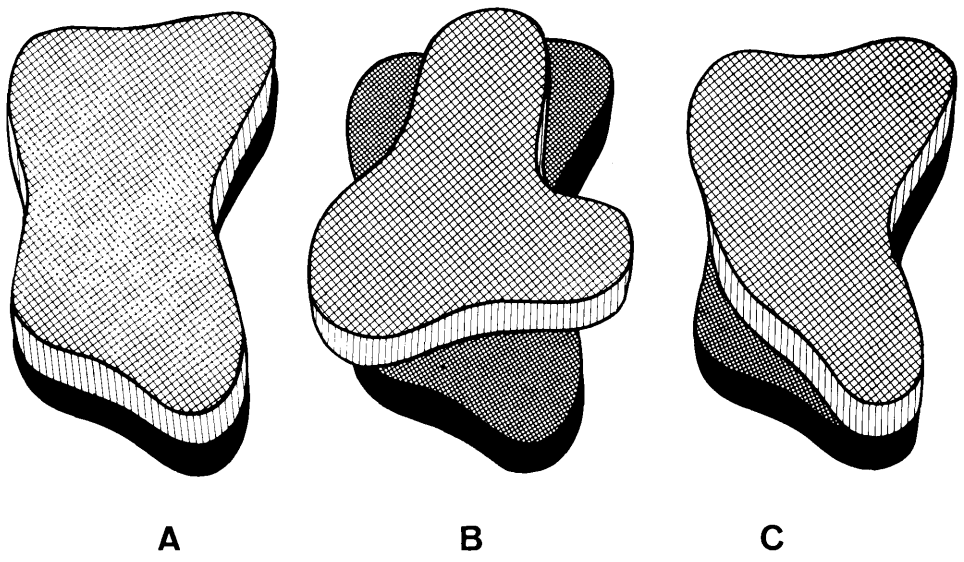


FIGURE 6-10 Schematic representation of pattern matching. Two wafers are shown in each instance, the lower one denoting the prototype being sought and the upper one denoting the case being tested. Case A: an exact match. Every important feature of the prototype is found in the case, and there are no features of the case that are not explained by the prototype. Case B: there are features of the case that are not explained by the prototype. Case C: there are features in the prototype that are not found in the case.

greater than 5 g/24 hours), a *sufficiency rule* immediately accepts the nephrotic syndrome hypothesis.[6]

In this accepted state, the hypothesis is asserted as if it were a fact. This new "fact" then is added to the short-term memory where it, in turn, can be found by daemons belonging to other frames. We should emphasize, however, that if later facts contradict the original conclusion, the acceptance is revoked.

In many instances, of course, there is no simple rule that can serve either to exclude or to establish a given hypothesis, and a *scoring process* is required. This scoring process uses numerical values (contained in the frame) that reflect the likelihood that various clinical findings will occur in

[6]Not only can disease frames be accepted or rejected, but frames corresponding to physiologic and clinical states can be similarly accepted or rejected.

the given disorder.[7] Major features are given more weight in the final scoring process than are the minor features.

Consider, for example, the nephrotic syndrome frame shown in Figure 6-7. Those features of the frame that can be readily identified as present or absent (e.g., low serum albumin concentration or heavy proteinuria) are given a numerical value. The remaining features are not initially assigned values; instead their contribution to the scoring process is determined by affiliated frames, e.g., hypovolemia is evaluated by means of a specific hypovolemia frame. Once such an affiliated frame has carried out its scoring function, the resulting value is passed on to the central frame. For example, acute glomerulonephritis receives a score for "elevated blood pressure in the absence of signs of chronic hypertension" from the acute hypertension frame to which it is related by a "may be complicated by" link. Similarly, the acute hypertension frame itself depends on affiliated frames (e.g., hypertensive encephalopathy) for its own score. In some instances, such propagation of scoring proceeds through several levels.

If the score for a hypothesis exceeds a defined threshold, the frame is accepted by the supervisor. Similarly, if the score falls below a given threshold (i.e., if the hypothesis no longer fits the patient "well enough"), the supervisor forces the hypothesis into a semiactive state. *The ability of the hypothesis to account for the findings of the case* is the extent to which all the facts of the case are explained by the hypothesis and its affiliated frames. The hypothesis of acute glomerulonephritis can explain, for example, both "low serum complement" and "oliguria"; the former finding is a part of the acute glomerulonephritis frame itself, and the latter is a part of a closely linked frame, "acute renal failure." It cannot, however, account for the finding of long-standing hypertension. The program computes for each frame a value equal to the fraction of all findings in the patient profile that are explained by the hypothesis. This value and the measure of the fit of the case to the hypothesis are averaged, and the hypotheses are assigned a rank order based on the average.

Selecting Questions

After the supervisor has ranked the hypotheses, it seeks to gather more information about the patient in order to improve its understanding of the clinical problem. The hypothesis that has received the highest overall score is explored first, with the initial inquiries directed to the classic findings of the disorder. The answer to each question that is posed causes the reevaluation of all hypotheses; as new information is obtained, the supervisor determines whether the leading hypothesis being pursued is still plausible, should now be accepted, or should be discarded from active consideration.

---

[7]The weights associated with those features of the frame known to be present in the patient are summed and then normalized by the maximum attainable score.

After the program has gathered information on prototypical features of the frame, it turns to questions about minor features of the disorder and then to inquiries about complications, etiological factors, and differential diagnoses. A change in the train of questioning usually indicates that, as the result of the continuous process of reevaluation, a new hypothesis has moved into the leading position.[8]

Repeating the Cycle

Any new finding obtained in the course of the questioning process sets into motion a cycle that is the same as that just described for the chief complaint: each new sign or symptom is characterized, advice is sought, hypotheses are generated and tested, and additional questions are asked. In its cycle of response to new findings, the program will, however, make use of the information acquired earlier (and hypotheses already generated) and will thus focus its questioning more sharply than if such a context were not available.

Controlling the Proliferation of Hypotheses

As discussed, the information gathering and the diagnostic competence of the present illness program depend critically on its ability to quickly generate hypotheses to account for the patient's condition. For this reason, "aggressive" hypothesis generation occurs even when only a few rather isolated facts are available. To avoid the excessive computational burden that is often produced by such an aggressive strategy, the program employs several methods to restrict the number of hypotheses under active consideration.

Two of these methods have already been mentioned—the "zeroing-in" on a complaint and the two-stage process of hypothesis generation. A third method is the application of the *principle of parsimony*. Let us take, as an example, a patient with edema and massive proteinuria who is hypothesized to have nephrotic syndrome with sodium retention. If the program discovers that the patient has a positive test for antinuclear antibody, it does not simply add a new hypothesis, "systematic lupus erythematosus." Instead, it incorporates the hypothesis of "nephrotic syndrome with sodium retention" into a new, overall hypothesis of "lupus erythematosus

[8]Sometimes the program considers a new hypothesis because of advice stored in the frame currently under consideration. For example, if nephrotic syndrome is the current hypothesis and symptoms suggestive of pulmonary emboli are reported, advice in the nephrotic syndrome frame will suggest that attention be shifted to renal vein thrombosis. The supervisor will then call up the questions designed to explore this latter possibility.

with nephrotic syndrome." If at a later time the more parsimonious hypothesis is rejected, the subhypotheses are again given independent status as active frames or, alternatively, are returned to a dormant state.

6.6 Comments

The present report demonstrates that insight derived from the study of clinical cognition can be combined with advanced techniques for computer simulation to create computer programs that possess a powerful problem-solving capability. The major technological advance embodied in our program is the capacity to retrieve and apply knowledge *when that knowledge is required,* thus freeing the programmer from the virtually impossible task of specifying all contingencies in advance [as would be necessary, for example, in a branching flow chart (Slack et al., 1966; Bleich, 1969; 1972; Stead et al., 1972)]. The key to implementation of the present system lies in the goal-directed nature of its operation. It is this goal-directed character that permits the supervisory program to select pertinent medical and real-world knowledge from the computer's memory and to dynamically assemble many small problem-solving techniques that efficiently guide the acquisition of additional clinical information. Another central feature of the program is the organization of its data base into an associative memory in which clusters of closely related facts about diseases and clinical states are stored in a fashion analogous to a richly cross-referenced encyclopedia. These groups of facts, called *frames* (Figure 6-7), are further organized into a *network* (Figure 6-8) that facilitates efficient retrieval of closely related blocks of information. When the supervisory program is presented with a clinical problem (that is, a chief complaint), it generates hypotheses about the case by moving frames from the long-term (associative) memory into short-term memory, where the frames interact with a profile of the patient's clinical data. When a hypothesis is generated, the supervisory program becomes "aware" of all the etiologies, complications, and other features of the hypothesized condition because the frames describing such related facts are drawn into close proximity to short-term memory.

The goal of the program is to arrive at the best possible diagnostic appraisal by evaluating these hypotheses. To accomplish this purpose, the computer characterizes each finding in detail, seeks relevant advice from the associative memory, and tests the hypotheses. The set of hypotheses under consideration also provides a framework within which additional information, both medical and real-world, is sought and interpreted. Throughout the questioning process, the supervisory program searches for inconsistencies in the information that it has obtained. When such inconsistencies arise, the program consults the memory for specific advice on how to deal with the conflicting information.

As new facts are obtained, additional hypotheses may be generated. From time to time, the supervisory program may also combine several hypotheses to form a more coherent diagnostic picture. As the questioning proceeds, all hypotheses under consideration are repetitively tested and scored to measure the "goodness of fit" between the description of the disease or physiologic state and the profile of facts about the patient. This testing provides the basis for either the acceptance or rejection of each hypothesis. At the termination of the questioning, the accepted hypotheses are listed, and the other hypotheses are rank ordered on the basis of the final score calculated for each.

The fundamental principles embodied in the present illness program are, we believe, applicable not solely to the problem of edema but are broadly relevant to the history-taking process. It is obvious, however, that many strategies other than those we have employed must be uncovered if a system such as we have described is to deal effectively with a wide range of clinical problems. In addition, numerous medical and real-world facts must be added to the program, and ingenious new techniques must be devised that can deal with multiple coexisting diseases, that can draw appropriate inferences about the temporal aspects of a patient's history, and that can choose the point at which the questioning process should be terminated. The solutions to such problems obviously will require many years of intensive work.

6.6.1 The Problem of Scale

We believe that, over the long term, computer programs can be developed that should be capable not only of taking the present illness but also of assisting in virtually all aspects of patient management. If this view is correct, one must then ask whether the existing technology will be able to cope with the volume of information that might be required by such a system; that is, will it be possible to store the requisite number of facts at a reasonable cost and to retrieve them in an efficient and effective manner? To answer this question we must first ask how much a computer program must "know" before it knows all of general internal medicine. Obviously, any calculation of this sort must be highly speculative, but it seems certain that the program must have available at least that body of information that is contained in a standard textbook of medicine. As shown in section A of Table 6-1, it appears that each of the two most widely used textbooks of medicine contains on the order of 200,000 facts.[9] This estimate far un-

[9]This estimate was arrived at by the crude technique of estimating the number of facts on several pages (not only basic facts, but the relationships between them) and multiplying this average by the total number of pages in the book. The major source of variability in such an estimate is the definition of what constitutes a single fact, because such a definition is to a certain extent arbitrary. In our calculations, we have used as a yardstick the amount of information that is treated as a single fact by our program; however, the choice of any other reasonable yardstick would not have changed our results appreciably.

TABLE 6-1 Estimate of total number of facts contained in standard textbooks of medicine and in representative subspecialty texts*

Title	*Pages (approx.)*	*Facts per page (approx.)*	*Total facts*
A. GENERAL INTERNAL MEDICINE			
Principles of Internal Medicine (Wintrobe, 1974b)	2,035	100	200,000
Textbook of Internal Medicine (Beeson and McDermott, 1975)	1,892	100	190,000
B. SPECIALTY TEXTS			
Diseases of the Kidney (Strauss, 1971)	1,456	40	60,000
The Heart (Hurst, 1974)	1,755	50	90,000
Clinical Hematology (Wintrobe, 1974a)	1,788	40	70,000

*Estimated as described in Footnote 9.

derstates, however, the total amount of information that is relevant to the practice of internal medicine. It is clear, for example, that there is a fund of basic science information used by the clinician that does not appear in such a textbook of medicine. To account for this body of data, we will double our estimate to a total of 400,000 facts. Finally, there is a considerable body of information about the real world (life insurance examinations, army physicals, time of day, seasons of the year), which, we will estimate, requires knowledge of still another 100,000 facts.[10] This brings us to a total of 500,000 facts. If we now double this value to take cognizance of possible underestimates, we arrive at an upper bound of approximately 1 million facts as the core body of information in general internal medicine.

The core knowledge embodied in the approximately ten separate subspecialties of internal medicine is, of course, considerably larger. To estimate the volume of clinical information basic to the entire domain of subspecialty medicine, we first have estimated the number of facts in textbooks of nephrology, cardiology, and hematology. As shown in section B of Table 6-1, each of the subspecialty treatises contains on the order of 60,000 facts. From this we estimate that the core body of information in all medical subspecialty texts combined is about 600,000 facts. If we assume that approximately one-third of this information represents duplications among the specialty fields, we arrive at a total body of 400,000 facts, a value approximately twice that estimated for general internal medicine. Using the same ratio between facts and other kinds of relevant information as we used in the case of general medicine, we calculate, correcting again for any possible underestimate, that the core of information in the subspecialties of internal medicine does not exceed 2 million facts.

[10]Note that we are only concerned with real-world knowledge that is relevant to medicine, not with all such knowledge possessed by the average person.

6.6.2 Can the Knowledge Base of Internal Medicine Be Stored at a Reasonable Cost?

Can 2 million facts be stored in a computer system at a reasonable cost? If we assume that each fact requires for its representation an average of 10 words of computer memory,[11] a computer storage capacity of 20 million words would be necessary. A memory of this size is certainly large, and, if core storage were required, the cost would at present be prohibitive. Because only a small part of the data is used at any one time, an inexpensive mass storage device (such as a magnetic disc or drum) would be a practical alternative storage medium; even at present prices, the cost would probably be no more than $20,000.[12]

We should note, furthermore, that even with rapid progress in the development of "expert" consulting programs, it is unlikely that a system could reach the size we have envisioned in a period of less than ten years. By that time, given the rapid evolution of computer technology, it is almost certain that a memory capable of storing 2 million facts could be purchased at a very low cost. From these considerations, it can reasonably be concluded that data storage will not be the limiting factor in the development of consulting capability within the computer. Indeed, even the storage of an additional large body of specialized information drawn from the literature, should, some years hence, pose no great technological difficulties.

6.6.3 Can the Data Base of Internal Medicine Be Efficiently Managed?

The problems of organizing, retrieving, and applying the relevant data are far more formidable than is the problem of data storage. We believe, however, that the task is probably not insuperable, because in any given case only a very small fraction of the available knowledge needs to be retrieved. Furthermore, the retrieval of whatever information is required will be greatly eased by the fact that pertinent information can be dealt with in the highly organized clusters known as frames. Assuming that the average frame will contain on the order of 100 facts, only 20,000 frames would be required for the postulated data base of 2 million facts.

Probably the most difficult aspect of data management will be the problem of *coding*, the process of ensuring that each fact is properly associated with other facts. Only if the large data base of internal medicine

---

[11] We are considering this representation in a computer language such as LISP, which is quite efficient at storing and retrieving the type of symbolic data that we envision will be used.

[12] *Ed. note:* This cost estimate, accurate in 1976 when this article appeared, is excessive in 1984 due to technological advances.

can be transferred automatically from English to the appropriate representation within the computer is there hope that serious errors, omissions, and contradictions can be avoided. Current efforts to develop computer programs that understand English give promise that this fundamental problem will eventually be solved (Schank and Colby, 1973).

The arguments we have considered here have led us to the conclusion that, over the long term, there are not likely to be intrinsic technological constraints on the realization of a system capable of coping with all of internal medicine. In fact, the availability of increasingly powerful technology suggests a future in which computer programs may well "know" far more than any individual physician. For the short term, however, we look toward the development of programs that know a great deal, but not all, of internal medicine.

6.6.4 Some Reflections on the Cognitive Process

As discussed earlier, the present illness simulation described here is based on insight derived from introspection and from observation of the problem-solving behavior of experienced clinicians. Here we offer a brief discussion of certain key ideas that we believe merit further study by investigators interested either in computer-aided decision making or in clinical cognition.

Our study clearly illuminates an important difference between the expert in practice and the expert as often pictured in literature or folklore. The epitome of the expert in fiction is the detective who, through superior deductive powers and by sheer force of logic, organizes the facts at hand in such a way that they lead to a single, inevitable conclusion. By contrast, the real-world clinician seems to rely much more heavily upon "guessing," the initial hypothesis typically being based on precious little data. These guesses are apparently prompted by patterns of clinical findings or by specific complaints that bring to mind particular diseases. The physician then tries to demonstrate the correctness of his or her guesses, moving to new hypotheses only if the initial impressions prove untenable.

The rapidity with which the initial hypotheses are generated and the ostensibly fragile basis of the guessing process together constitute the most striking feature of the behavior of experienced clinicians. Often with only the age, sex, and presenting complaint of the patient, the clinician unhesitatingly selects a single working hypothesis. Even in ambiguous situations, he or she rarely begins with more than a few hypotheses.

Another characteristic of the experienced physician is the fashion in which he or she continually pares the list of diagnostic possibilities. The physician discards some, accepts others, and often combines individual possibilities into a single, new, integrated hypothesis. In this way, he or she is generally able to limit sharply the number of diagnoses which must actively be considered. We can understand the value of such a sharp focus

when we consider that, in taking a present illness, the physician can gather only a small fraction of the potential set of facts concerning the patient and must therefore seek information very selectively. In consequence, the clinician must find a context within which to properly focus his or her questioning and to organize the information that is obtained.

Because the initial hypotheses are usually generated on the basis of relatively few facts, they will often later prove to be incorrect. In such cases, how does the experienced clinician proceed to undo any "damage" done by aggressive hypothesis generation? Our observations suggest that he or she often employs the rather efficient strategy of associating one hypothesis with others with which it may be readily confused (e.g., "multiple pulmonary emboli are often confused with cardiomyopathy"). By explicitly remembering such situations, the physician can move directly from a hypothesis that has become suspect to one that offers another plausible explanation for the presenting findings.

Unlike the seasoned clinician, the medical student or young physician does not have an extensive knowledge of such relations and so is unlikely to move from one hypothesis to another in such a skillful fashion. Therefore, the novice who acts aggressively in hypothesis generation risks making serious errors. We have observed that the student or house officer, apparently to counter this problem, often approaches the diagnostic process in a highly structured, methodical fashion. Similarly we have noted that the experienced physician performing outside his or her area of expertise uses a far more structured approach than is his or her usual custom. The seasoned clinician's expertise in taking a present illness thus appears to derive in considerable part from a complex set of associations and from a familiarity with many alternative scenarios within that individual's "frames."

We believe that the experimental methods utilized in the present study, if extensively employed, will provide important new insights into the process of clinical problem-solving. Furthermore, as our understanding of problem-solving processes grows, it seems likely that the study of clinical cognition will assume a significant place in the medical curriculum. Such increased attention to this neglected aspect of medical education should eventually make an important contribution to improving the quality of physician performance.

ACKNOWLEDGMENTS

This research was supported in part by the Health Resources Administration, U.S. Public Health Service, under grant 1 R01 MB 00107-01 from the Bureau of Health Manpower and under grant HS 00911-01 from the National Center for Health Services Research.

7

A Model-Based Method for Computer-Aided Medical Decision Making

Sholom M. Weiss, Casimir A. Kulikowski, Saul Amarel, and Aran Safir

While MYCIN and PIP were under development at Stanford and Tufts/M.I.T., a group of computer scientists at Rutgers University was developing a system to aid in the evaluation and treatment of patients with glaucoma. The group was led by Professor Casimir Kulikowski, a researcher with extensive background in mathematical and pattern-recognition approaches to computer-based medical decision making (Nordyke et al., 1971), working within the Rutgers Research Resource on Computers in Biomedicine headed by Professor Saul Amarel. Working collaboratively with Dr. Arin Safir, Professor of Ophthalmology, who was then based at the Mt. Sinai School of Medicine in New York City, Kulikowski and Sholom Weiss (a graduate student at Rutgers who went on to become a research scientist there) developed a method of computer-assisted medical decision making that was based on causal-associational network (CASNET) models of disease. Although the work was inspired by the glaucoma domain, the approach had general features that were later refined in the development of the EXPERT system-building tool (see Chapters 18 and 20).

A CASNET model consists of three main components: observations of a patient, pathophysiological states, and disease classifications. As observations are recorded, they are associated with the appropriate intermediate states. These states, in turn, are typically causally related, thereby forming a network that summarizes the mechanisms of disease. It is these patterns of states in the network that are linked to individual disease classes. Strat-

From *Artificial Intelligence*, 11: 145–172 (1978).

egies of specific treatment selection are guided as much by the individual pattern of observations and diagnostic conclusions as they are by the disease classification itself.

Unlike mathematical models of disease processes, a CASNET model is inherently symbolic and focuses on causality and temporal sequences of events. Although not all medical topics are well understood at this level, CASNET demonstrated that there are areas of medicine in which explicit model representations permit powerful reasoning strategies that go beyond simple matching of treatments with diseases. It is this ability to match treatment plans with the patient's current stage in the progression of a disease process and with expectations of future events that set CASNET apart from the other early AIM systems. More recently ABEL (Chapter 14) and VM (Chapter 10) have extensively studied similar issues, and Pople has discussed at length the need to incorporate causal reasoning and a sense of temporal progression into future versions of INTERNIST (Pople, 1982).

7.1 Introduction

In the present paper, a general approach to structuring medical knowledge for computer-aided diagnosis and therapy is presented. We have developed a representation that models disease processes as a causal-associational network (CASNET). This model-based method has been used successfully in designing a consultation program for the diagnosis and long-term treatment of the glaucomas. The consultation program uses a set of general decision-making strategies in conjunction with a class of causal-associational models (Kulikowski and Weiss, 1971; Weiss, 1974). In this paper, examples will be given from a CASNET model of glaucoma. However, the model representation and decision-making procedures are generalizable to other medical domains.

Diagnostic problems have often been cast into a pattern-recognition or statistical decision-theory framework. Computer representation is not difficult, and as a result many well-known methods such as those based on Bayes' Theorem have been used (Brodman et al., 1959; Warner et al., 1964; Gorry and Barnett, 1968a). The difficulties with applying these methods (such as scarcity of statistics and the use of invalid approximations) are also sufficiently persistent that alternative approaches have been sought. In many medical areas, existing knowledge could enhance the decision-making capabilities of a diagnostic system. There are many useful decision rules specific to a given medical application that the physician directly applies in his or her reasoning.

In the past few years, there has been increased interest in the application of artificial intelligence (AI) techniques to medical decision making.

AI techniques attempt to capture decision-making rules explicitly, while statistical methods may extract them implicitly from accumulated sample experience. The AI approaches intend to overcome some of the limitations of purely statistical methods by developing a more structured representation of the diagnostic and therapy selection problems. A program that uses decision strategies based on explicit representations of medical knowledge can more easily incorporate evolving changes in its knowledge base, independently of the reasoning strategies. It can also incorporate the results of clinical experience by matching the more explicit patterns of reasoning to the decisions and opinions of physicians. Such systems are more likely to be accepted because they are expressed in a decision-making context familiar to the clinician. A structured representation can also permit the formulation of complex hypotheses that express progression and severity of disease. Some researchers have attempted to increase the scope, accuracy, and explanation capabilities of their systems by increasing structure, while still preserving a statistical framework (Patrick et al., 1974). Others have relied on logical and semantic encodings of contextual knowledge within an artificial intelligence framework (Pople et al., 1975; Shortliffe et al., 1973; Wortman, 1972) (see also Chapter 6).

Several fundamental AI issues are raised by medical decision-making problems. One important issue concerns the development of representations that are powerful enough to capture a complex and changing knowledge base in a realistic task domain. There has been an increased interest in recent years in developing AI systems that use expert knowledge in a variety of application areas (Buchanan et al., 1969; Duda et al., 1977; Reddy, 1977; Sridharan and Schmidt, 1977). Methods of acquiring knowledge from experts, the choice of appropriate levels of abstraction and resolution for describing a given problem, and the choice of computer representation of the knowledge base are all problems that immediately arise in developing such systems. They are closely linked to the control strategies or methods used to produce interpretations for individual cases. Fundamental to most such control strategies is the capability of approximate reasoning. This is needed to manage the multiple hypotheses that can be generated from a large and complex knowledge base, which includes statements at different levels of uncertainty. Once decisions are reached, producing explanations becomes an important task if the acceptability of the system is to be enhanced. Practical issues of implementation for these large knowledge-based systems include ease of knowledge management (updating), efficiency, choice of languages, and transferability into practical use in both the original domain and other similar ones.

The present paper describes the methods of representation and interpretation developed while building a knowledge-based system for medical consultation. In the course of describing these methods, specific solutions to some of the issues raised above are offered.

7.2 Causal-Associational Network (CASNET) Models

A causal-associational network is a particular type of semantic network (Woods, 1975) designed to:

a. describe dynamic processes in terms of (loop-free) causal relationships among a set of internal variables;
b. relate this description to external variables that are considered to be manifestations of the internal processes; and
c. describe various classifications imposed on the dynamic processes.

CASNET models can be used to describe many different complex processes, but we have developed them to describe pathophysiological processes of disease (Weiss, 1974). Knowledge, in our scheme, is represented by three types of data elements, corresponding to the three kinds of description outlined above: observations of the patient; pathophysiological states; and diagnostic, prognostic, and therapeutic categories. Observations are the direct evidence obtained about a patient. Pathophysiological states are intermediate constructs that describe internal conditions assumed to take place in the patient; they summarize results from many different observations. Categories of disease are conceptually at the highest level of abstraction, summarizing patterns of states and observations. In Figure 7-1 we summarize this three-level description of disease processes. Considerations of all three levels enter into the recommendation of therapy. Bonner et al. (1964) developed a single-level model with causal and associational relations intermingled. When diagnosis is to be modeled in a domain of knowledge where mechanisms of disease are understood, the cause-and-effect model can be used to significantly improve the basis on which decisions are made. When, however, less information is available, associations between findings must be relied on to a greater extent, and the goals of reaching structured and well-explained conclusions and recommendations may not be fully satisfied.

7.2.1 Causal Network of States

In our model of disease, the pathogenesis and mechanisms of a disease process are described in terms of cause-and-effect relationships between pathophysiological states. States are summary descriptions of events that are deviations from normality. Strict causality (Bunge, 1963) is not as-

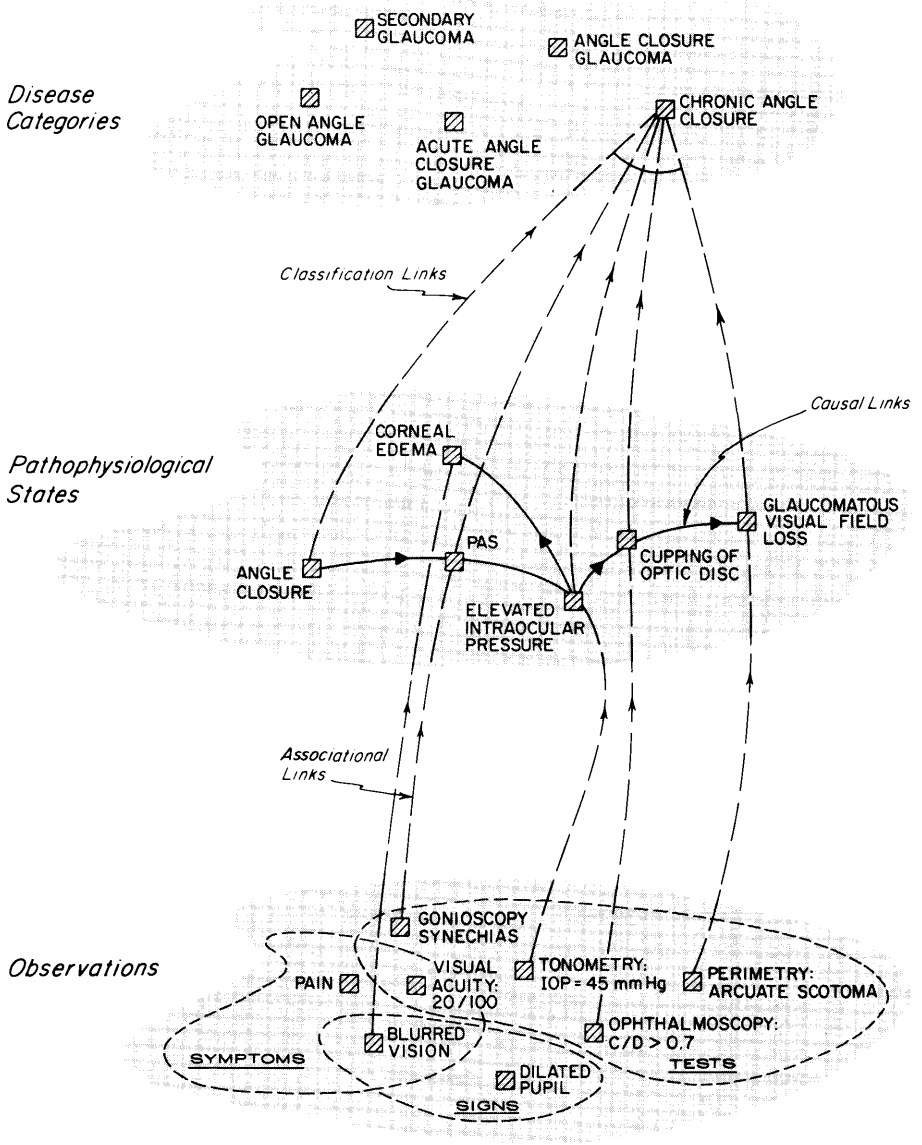


FIGURE 7-1 Three-level description of a disease process.

sumed—there may be multiple causes and effects, and in a given patient, a cause may be present without any of its effects occurring at the same time. Various effects can follow from a given cause, each produced with a different strength of causation. Examples of states would be "increased

intraocular pressure" or "glaucomatous visual field loss." Many such states may occur simultaneously in any disease process. A state thus defined may be viewed as a set of values of a state variable as used in control systems theory. It does not correspond to one of the mutually exclusive states that could be used to describe a probabilistic system. This definition was chosen to correspond to the basic entities physicians use when they describe disease mechanisms. A somewhat simplified graph model of glaucoma is illustrated in Figure 7-2, where each node, n_i, is a pathophysiological state, and each edge is a causal connection. Disease processes may be characterized by pathways through the network. A complete pathway from a starting to a terminal node usually represents a complete disease process, while partial pathways, from starting to nonterminal nodes, represent various degrees of evolution within the disease process. Progression along a causal pathway is usually associated with increasing seriousness of the disease. For example, in Figure 7-2 a complete pathway is traversed from n_{35} (PRIMARY OPEN ANGLE MECHANISM) to n_{31} (GLAUCOMATOUS VISUAL FIELD LOSS): (n_{35} n_{25} n_{26} n_{27} n_{28} n_{29} n_{30} n_{31}). A partial pathway is traversed from n_{35} (PRIMARY OPEN ANGLE MECHANISM) to n_{26} (ELEVATED INTRAOCULAR PRESSURE): (n_{35} n_{25} n_{26}).

When a set of cause-and-effect relationships between states is specified, the resulting structure is a network, or directed acyclic graph of states. The state network is defined by (S, F, N, X), where S is the set of starting states, those states with no antecedent causes; F is the set of final states, those states with no effects; N is the total set of states; X is the set of mappings between states indicating causal relationships.

The mappings are of the form

$$n_i \xrightarrow{a_{ij}} n_j$$

where a_{ij} is the strength of causation (interpreted in terms of frequency of occurrence) and n_i and n_j are states. This rule is interpreted as follows: state n_i causes state n_j, independently of other events, with frequency a_{ij}. Starting states are also assigned a frequency measure indicating a prior or starting frequency. The strengths of causation are represented by numerical values, fractions between 0 and 1 that correspond to qualitative ranges such as sometimes, often, usually, or always.

States are summary statements. Many events and many complex relationships may be summarized by a single state. For example, in Figure 7-2, "neural tissue loss and cupping of the nerve head" is a summary of a much more complex situation. If a higher-resolution description is desired, several different types of nerve loss and cupping could be specified. The resolution of states should be maintained at a level consistent with the objective of efficient decision making. A state network can be thought of as a streamlined model of disease that unifies several important concepts and guides us in our goal of diagnosis. It is not meant to be a complete model of disease.

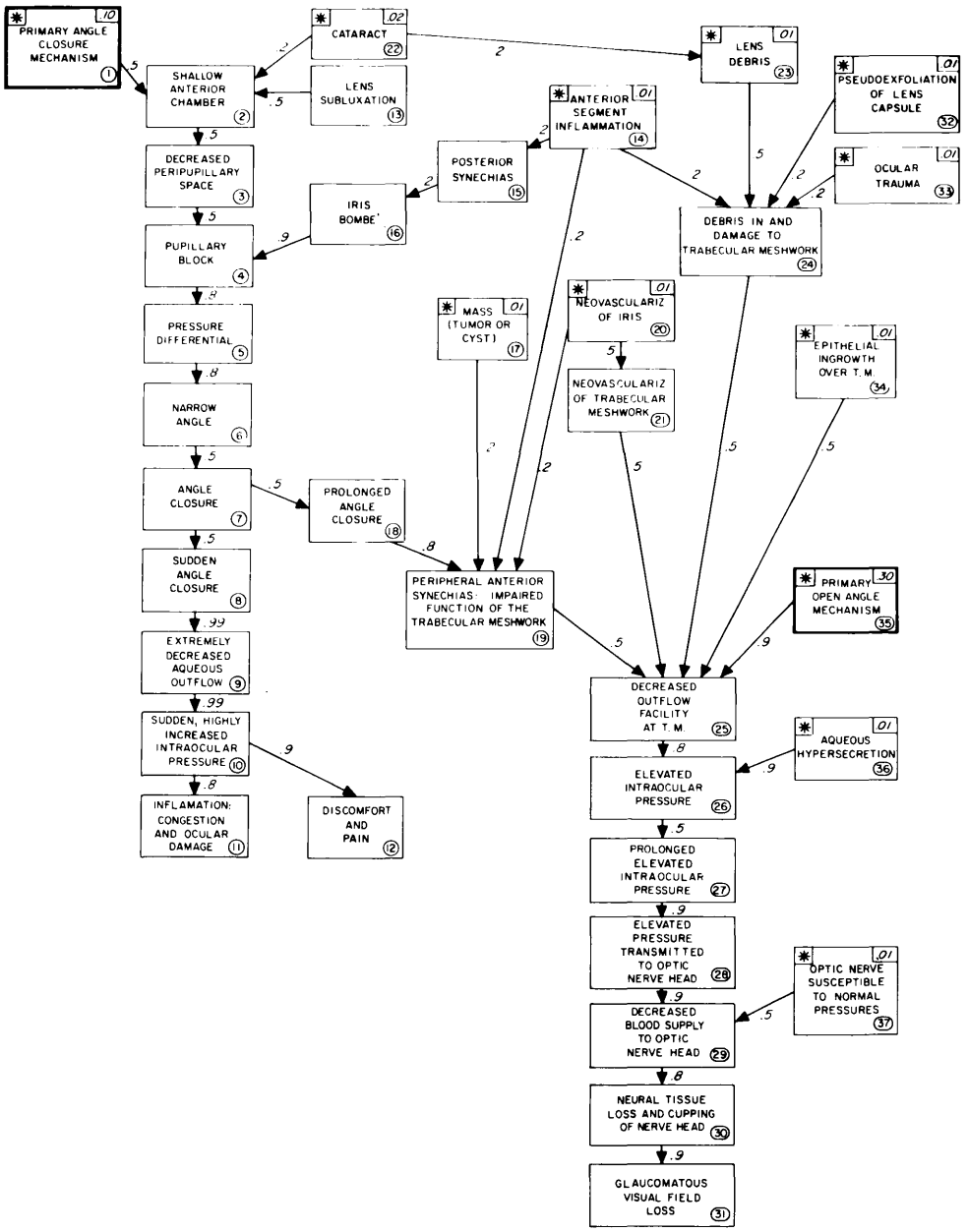


FIGURE 7-2 Partial causal network for glaucoma. States with no antecedent causes are indicated by asterisks (*). The circled numbers correspond to the state labels (n_i) used in examples in the text.

7.2.2 Rules for Associating States with Observations

Observations (tests)—the history, signs, symptoms, and laboratory tests—are the form in which information about a patient is presented. These clinical features, however, must be unified into some coherent framework for explanation and diagnosis. Observations about a patient are used to confirm or deny certain states in the network that describe the disease process. A single state may be associated with many observations. These states can then be related by causal pathways that explain the mechanisms of disease in a patient. The relationship between tests and states is noncausal; it is associational. For a given observation, confidence measures are used to indicate a degree of belief in the presence of specific states.

The rules for associating tests with states are represented as

$$t_i \xrightarrow{Q_{ij}} n_j$$

where t_i is the ith observation (or Boolean combination of observations), n_j is the jth node, and Q_{ij} $(-1 \leq Q_{ij} \leq +1)$ is the confidence in n_j given that t_i is observed to be true. Positive values of Q_{ij} correspond to an increased confidence in n_j, and negative values correspond to a decreased confidence in n_j when t_i is observed. Associated with each observation are costs $C(t_i)$ that reflect the cost of obtaining the result t_i.

Example 1. Two different instruments may be used to measure intraocular pressure (tension): a Schiotz tonometer and an applanation tonometer. A high Schiotz tension reading may indicate an elevated intraocular pressure with a confidence of 0.5. A high applanation tension reading, which is usually more reliable, may be assigned a higher confidence, such as 0.7. If by ophthalmoscopy it is further demonstrated that the appearance of the optic disc indicates damage to the optic nerve (with a confidence of 0.3), these results may be combined and assigned a confidence of 0.8 that the pressure has been and is truly elevated. Figure 7-3 illustrates these relationships, with circular nodes standing for states and square nodes for observations. The number on the link that connects a test to a state is the confidence with which a test supports a state.

7.2.3 Rules for Associating Disease Categories with States

Diagnostic and prognostic categories of disease are defined in terms of ordered patterns of rules, which we refer to as classification tables. The tables contain rules of the form

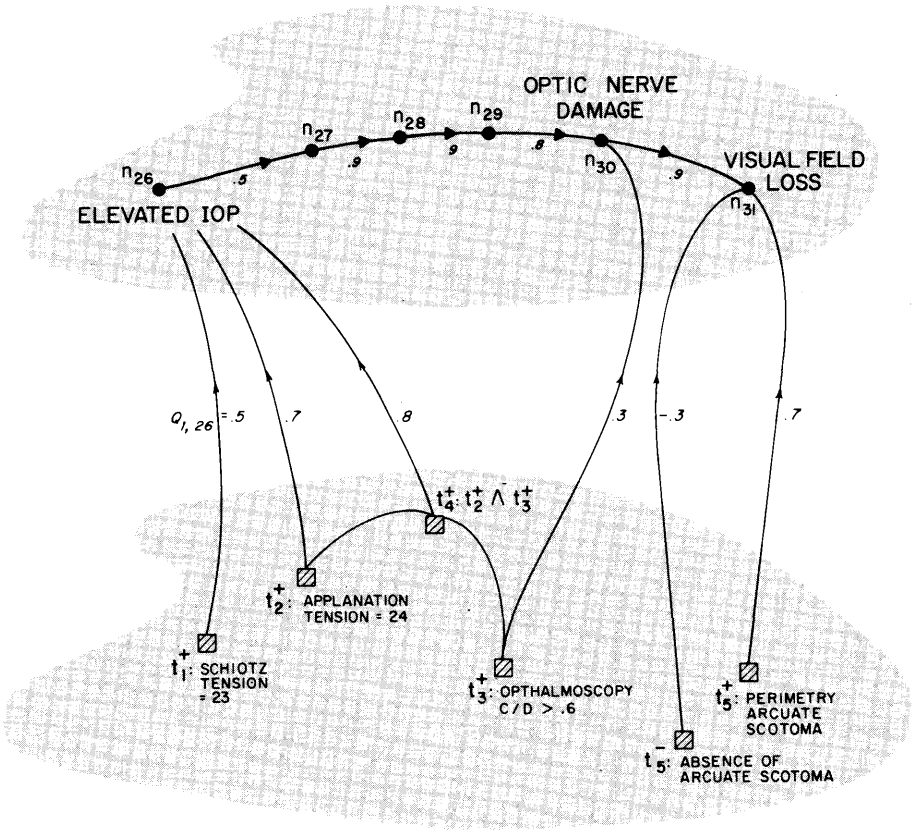


FIGURE 7-3 Examples of rules for associating states with observations.

$$
\begin{array}{llllll}
n_1 & \wedge & \neg n_2 & & \rightarrow & D_1 \\
n_1 & \wedge & n_2 & \wedge \quad \neg n_3 & \rightarrow & D_2 \\
 & & & \vdots & & \\
n_1 & \wedge & n_2 & \wedge \ldots \quad \neg n_i & \rightarrow & D_{i-1} \\
n_1 & \wedge & n_2 & \wedge \ldots \quad n_i & \rightarrow & D_i
\end{array}
$$

where D_i is the ith diagnostic and prognostic category, which is implied by the given ordered pattern of states, $n_1 \wedge n_2 \ldots \wedge n_i$. For this chosen form and ordering of rules, the tables can be referred to by using an abbreviated notation of ordered pairs:

$$(n_1, D_1), (n_2, D_2), \ldots (n_{i-1}, D_{i-1}), (n_i, D_i)$$

The classification tables can be augmented to include therapy recommendations. These tables are ordered triples of the form

$$(n_1, D_1, T_1), (n_2, D_2, T_2), \ldots (n_{i-1}, D_{i-1}, T_{i-1}), (n_i, D_i, T_i)$$

where T_i are treatments (or treatment plans) for patients falling into particular diagnostic categories.

In the following sections, clinical decision making will be considered as a problem of using a CASNET model for (a) selecting and interpreting observations, (b) analyzing and resolving conflicts and contradictions in the observations, (c) selecting diagnostic and prognostic categories, and (d) recommending treatments.

7.3 Test Result Interpretation

A test result has the following form: observation t_i is true, false, or uncertain. Based on a given result for t_i, a measure of confidence, Q_{ij}, may be assigned to state n_j. More than one test may confirm or deny a single state with varying degrees of confidence. The total confidence in the presence or absence of a state is derived from all local mappings from tests to states occurring for a given patient. Each node, n_j, in the state network is assigned a measure, $\mathrm{Cf}(n_j)$. Initially, the Cf of all nodes is undetermined; i.e., $\mathrm{Cf}(n_j) = 0$.

Rule 1. When a test result is received and a rule $t_i \xrightarrow{Q_{ij}} n_j$ is found applicable, the $\mathrm{Cf}(n_j)$ is affected as follows:

a. If $|\mathrm{Cf}(n_j)| < |Q_{ij}|$, then $\mathrm{Cf}(n_j)$ is reset to Q_{ij}.

b. If $Cf(n_j) = -Q_{ij}$, then $Cf(n_j)$ is set to 0 (and the conflict is noted) until another result t_k is received such that $|Q_{kj}| > |Q_{ij}|$.

c. Otherwise, $Cf(n_j)$ is unchanged.

Thus, of all the test results that are evidence for a given state, we choose the result in which we have the greatest confidence. When a new test result is received with a confidence measure equal but counter to the previously accumulated evidence, the conflict is noted, and the status of the node is reset to be undetermined.

A Cf measure is used to evaluate whether the status of a node is assumed to be confirmed, denied, or undetermined. Let Θ be a nonnegative integer that serves as a threshold fixed in advance for a specific model. (The threshold for test selection may be fixed at a level different from that used for classification.)

Rule 2.

a. If $Cf(n_j) > \Theta$, then n_j is assumed confirmed.

b. If $Cf(n_j) < -\Theta$, then n_j is assumed denied.

c. Otherwise, the status of n_j is assumed undetermined.

In this way, the designer of a model can assign confidence to the test results. Whenever the status of a node n_j exceeds (or is less than) a uniform and consistent threshold, node n_j is assumed confirmed (or denied). At some point there is enough confidence in these findings to draw at least tentative conclusions about some specific aspects of the disease, which are summarized in the states. These conclusions can change when other test results, in which we have greater confidence, are received.

The initial state network graph is a static structure. However, based on a series of observations, a configuration, or labeled subnet, of the state network can be generated that is applicable to a given patient. For a given patient, a configuration of the state network is described by assigning each node either a confirmed, denied, or undetermined status. The state network dynamically evolves into different configurations, each determined by the interpretation of the test results. Tentative diagnostic conclusions and decisions can be reached for each configuration of the state network.

7.4 Strategies for Test Selection

A configuration of the state network can be used not only to reach conclusions, but also to select questions. An interactive sequential questioning procedure that is guided by the results of previously asked questions can usually reduce the number of questions that must be asked, often elimi-

nating irrelevant and redundant questions. Asking the right questions in an intelligent order is an important aspect of the diagnostic process.

The strategies for test selection that have been developed for CASNET-type models can be categorized as those emphasizing (a) local logical constraints among questions, (b) categories of causal pathways, and (c) likelihood measures over the states.

These strategies are not mutually exclusive, and all three may be combined into a single overall strategy. The simplest strategy, yet perhaps the most effective for a well-circumscribed domain of application, is the strategy that emphasizes local logical constraints among the questions. For this strategy, questions on related topics are organized into small local tree structures. Each group of questions is asked only when a fixed set of logical conditions is satisfied.

The second strategy depends on isolating the causal pathways that potentially explain the observations that have already been recorded. The strategy would then pursue observations that are related to the states found on these pathways. The identification of pathways that may explain the current observations and related processes of disease is discussed in Section 7.5.

A likelihood strategy for the CASNET model is based on the assignment of weights to each of the nodes in the state network. Tests that can produce results having greater measures of confidence than are currently held for the states are considered possible candidates for further testing. Of these tests, the one that relates to the highest-weighted node is selected.

A number of characteristics of the state network are important for the specification of inference strategies:

a. No loops may exist in the network because all transitions between nodes are unidirectional under the assumption of causal production.

b. Starting nodes have no antecedent causes (or predecessors in the network) and represent events taken as the starting events in the causal chains. These nodes are assigned (prior) weights, a_i, based on their relative frequency of occurrence.

c. Each transition weight has a maximum value of unity. The sum of transition weights leaving node n_i is not necessarily unity, because the successors of n_i are not necessarily mutually exclusive. In addition, the model incorporates only consequences of events that are of interest to the process being described, leaving unspecified any other possible outcomes.

d. The transition weight, a_{ij}, in a link $n_i \xrightarrow{a_{ij}} n_j$ *is assigned on the assumption that* n_j is caused by n_i with frequency a_{ij}, independently of the way in which n_j was entered from other nodes of the network. For a given model, a consistent interpretation must be given to the transition weights throughout the network.

During a procedure of sequential test selection, a given node of the network can be in one of three status conditions: confirmed, denied, or undetermined. Initially, all nodes are undetermined, but as tests are selected and results obtained, some of the nodes will be confirmed and others denied.

At every stage of the selection procedure, each node in the network is assigned a weight that is determined by the current configuration of confirmed and denied nodes of the network. The derivation of these weights is given below. The weights serve as an index for the selection of further nodes to be queried. In a model of disease, where the nodes represent states of the disease process, the weights are used to choose the sequence of states to be tested. The assignment of weights in a causal network has a superficial similarity to a Markov chain (Gheorghe et al., 1976). The important differences are found in the lack of mutual exclusivity between successors of a node and in the assumption of causal production between successor nodes. A model could be designed as a Markov network, but it would require the specification of a much larger number of nodes and transitions for the many possible combinations of events that can occur in a complex process.

7.4.1 Forward Weights for Test Selection

An *admissible pathway* is said to exist from node n_i to node n_j when none of the intermediate nodes in the pathway are denied. For the remainder of the discussion on the calculation of weights, a reference to a pathway refers to an admissible pathway. Also, it is assumed that successive nodes on a pathway are numbered consecutively.

The weight of entering node n_j from a single admissible pathway starting at node n_i is defined as the product of the transition weights between all pairs of successive nodes (n_k, n_{k+1}) in the pathway:

$$w_F(j|i) = \prod_{k=1}^{j-1} a_{k,k+1}$$

The total forward weight of node n_j is computed as the sum of the weights $w_F(j|i)$ for those admissible pathways entering n_j, starting at the nearest confirmed or starting nodes, n_i, of the network. A nearest confirmed node within a pathway is a node such that there are no other confirmed nodes in the pathway between it and n_j. In the case that n_i is an unconfirmed starting node, the weight of this pathway is multiplied by the starting weight.

$$\text{Let } \mu_i = a_i \text{ when } n_i \text{ is an unconfirmed starting state,}$$
$$= 1 \text{ otherwise.}$$

The total weight of n_j is then

$$w_F(j) = \sum_i w_F(j|i) \cdot \mu_i$$
$$= a_j \text{ when } n_j \text{ is a starting node}$$

where i ranges over the set of nearest confirmed or starting nodes.

Example 2. (See Figure 7-2.) Assume that n_7 is confirmed, n_{17} and n_{20} are denied, and the other nodes are of undetermined status. The forward weight of $n_{19} = (0.5)(0.8) + (0.01)(0.20) = 0.402$. The weight is calculated from pathways beginning at n_7 (the nearest confirmed node) and at n_{14} (the only undenied starting node that leads to n_{19}).

The rationale for choosing the product of successive transition weights in a pathway lies in the assumption that each transition weight, a_{ij}, is independent of all preceding transitions. If n_{j-1} is a confirmed node, the weight $w_F(j-1|i) = 1$, so all previous transition weights within the pathway need not be computed. Hence the weight of the end node of a pathway is calculated only from the nearest confirmed or starting node.

When there is only one admissible pathway leading to a node, ignoring the possibility of overlapping causal events introduces no error in the computation of weights. If the network is defined with mutual exclusivity between all successors of any given node, the problem can be operationally treated as a Markov chain calculation. That is, all confirmed nodes do in reality lie on a unique pathway. In our representation, when overlap between pathways does occur, some nodes in the network may be given a greater weight by the above computations than would result from exact frequency assignments over disjointly defined pathways. Yet, because a pathway to a node, n_j, represents the manner in which n_j is produced, this greater weighting is acceptable and even helpful. The tendency toward overweighting is related to the number of pathways that lead to n_j and the strength of transitions between the nodes in these pathways. But it is precisely those nodes that have many possible ways of occurring and that have strong causal and frequency connections that are the most likely to occur for the patient. Since a product of fractions not greater than unity is employed for the computation of weights, weights computed on the basis of few observations will result in relatively small weight assignments to the nodes. For some nodes, this weight assignment may be an accurate measure of frequency. Even without exact frequency assignments, the manner in which confirmation or denial of nodes is included in the weight calculation can be quite effective in guiding the selection of tests. The confirmation of a given node, n_i, will usually greatly increase the weight of all of its effects. Many fewer fractional multiplications will be used in computing the necessary pathways from n_i, since for the successors of n_i the weight of n_i can be assumed to be unity. Similarly, the weights resulting from a

denied node, n_i, will decrease, since n_i cannot lie on any admissible pathway to another node. This is precisely the response needed to guide the search for information to topics suggested by the accumulated evidence.

In general, the overall effect of forward weight calculation is to increase the weights of those nodes resulting from confirmed nodes while decreasing the weights of those from denied nodes.

7.4.2 Inverse Weights for Test Selection

The forward weight of a node, n_i, summarizes the weight of evidence carried from the causes of n_i. The weight of n_i can also take into account the confirmed nodes that are effects of n_i. For this we must define some inverse weight of confirmed effect n_j on the cause n_i. In analogy to Bayes' formula for inverse probability, an inverse weight can be defined as

$$w_I(i|j) = [w_F(j|i) \cdot w_F(i)]/w_F(j)$$

where confirmed nodes are ignored in the pathways (and forward weights are, therefore, computed from starting states). Because an admissible pathway cannot contain a denied node, an inverse weight is proportional to the weight of pathways passing through n_j that also pass through n_i divided by the weight of all currently possible pathways to n_j.

Example 3. (See Figure 7-2.) Assume that all pathways are denied, except those beginning with n_{35} and n_{37}. Let n_{31} be confirmed and the remaining nodes undetermined. The inverse weights for n_{35} and n_{37} are calculated as follows:

$$w_F(35) = 0.30,\ w_F(37) = 0.01 \text{ (starting weights)}$$

$$\begin{aligned} w_F(31) &= (0.3)(0.9)(0.8)(.05)(0.9)(0.9)(0.8)(0.9) + (0.01)(0.5)(0.8)(0.9) \\ &= 0.067 \end{aligned}$$

$$w_F(31|35) = (0.9)(0.8)(0.5)(0.9)(0.9)(0.8)(0.9) = 0.210$$

$$w_F(31|37) = (0.5)(0.8)(0.9) = 0.360$$

$$\begin{aligned} w_I(35|31) &= w_F(35) \cdot w_F(31|35)/w_F(31) \\ &= (0.30)(0.210)/0.067 = 0.940 \end{aligned}$$

$$\begin{aligned} w_I(37|31) &= w_F(37) \cdot w_F(31|37)/w_F(31) \\ &= (0.01)(0.36)/0.067 = 0.054 \end{aligned}$$

Since several effects may follow from a single cause, it is desirable to choose some function of all the inverse weights to represent the overall

inverse weight of a node, n_i. A reasonable choice is the maximum of the inverse weights for each n_i:

$$w_I(i) = \max_j \{w_i(i|j)\}$$

This function was selected because we are searching the network for strong evidence that n_i is present. For the important situations where nodes lie on a single pathway to a confirmed node or nodes lie on every pathway to a confirmed node, the inverse weight for those nodes will be correctly assigned as unity. Where there are two or more mutually exclusive pathways to a confirmed node, the inverse weight remains a relatively accurate frequency measure. However, the pathways to a confirmed node need not be mutually exclusive. Therefore, the maximum of the inverse weights is used as an overall measure of inverse weight. The inverse weight for a confirmed node assigns weight on a fractional basis to the node's potential causes, even when more than one cause is strongly indicated. The maximum weight compensates for the lack of mutual exclusivity by considering evidence other than a single confirmed node. Since several confirmed nodes may in fact be unrelated, an average or sum would appear to be less effective. The maximum is effective because it preserves the weight of a strong piece of confirmatory evidence without dilution from other nonexclusive causes, because it recognizes the possible multiplicity of confirmed effects from a single cause, and because it generally provides a reasonable basis of comparison with the forward weights.

The calculation of inverse weights is strongly influenced by evidence for the confirmation or denial of nodes. The weight of a node may be increased when its effects are confirmed. Initially, a pathway may be an unlikely alternative, but after some testing it may become the only feasible pathway to a particular confirmed node. This results in increased weight assignments to the remaining causes of the confirmed node.

7.4.3 Overall Weight for Test Selection

In order to choose a node for testing, a single function of the forward and inverse weights of the node is needed as an overall measure. The maximum of these two weights has been chosen:

$$W_i = \max \{w_F(i), w_I(i)\}$$

This choice reflects the need to have a measure of strong confirmatory evidence for the potential presence of a node, n_i. Evidence of the denial of n_i is included in both $w_F(i)$ and $w_I(i)$. These forward and inverse weights represent the contribution from different parts of the network toward the likelihood of confirmation of n_i. The maximum is thus a measure of strong

confirmatory evidence toward n_i throughout the network. It should usually provide good testing candidates. Relatively efficient algorithms can be specified for the computation of weights (Weiss, 1974). These algorithms take advantage of the acyclic nature of the state network so that the states may be topologically sorted.

7.4.4 Test Selection with Cost Assignment

A weight is a measure of likelihood, based on the evidence gathered for the possible causes of a node. The weight does not take into account the cost of performing a test that may confirm or deny the node. Let t_i be a test for node n_j and C_i be the cost of t_i. W_j is the currently assigned (nonzero) weight of node n_j.

Two cost strategies have been used for test selection:

a. *Maximum weight-to-cost ratio:* select t_i such that $W_j/C_i = \max_{m,n}(W_m/C_n)$.

b. *Maximum weight within a certain range of costs:* select t_i such that $W_j = \max_m(W_m)$ for all t_n with $C_n < C$.

A strategy of maximum weight selection is a special case of strategy a when the costs are equal or are ignored. A minimum cost strategy is a special case of strategy b where C_n is taken as the minimum cost (for the remaining tests) and W_j is any nonzero weight.

The stopping rule for the likelihood strategy consists of terminating test selection when no weight exceeds a fixed threshold. For an in-depth consultation in an application such as glaucoma, where all topics must be covered thoroughly, all questions are asked that have not been logically excluded by prior responses. This corresponds to setting the threshold to zero.

A form of hypothesis-driven test selection has also been formulated. A hypothesis corresponds to a class of likely causal pathways that explain the patient's observations and related but as yet unknown findings. The strategy then selects tests that support the hypotheses. The following section describes methods of identifying the likely hypotheses (classes of pathways) for a patient.

7.5 Interpretation of Disease Processes Within the State Network

The state network is a general structure that implicitly contains large numbers of both complete and partial causal state pathways, representing processes of disease. Several general classes of pathways can be described that

are useful for decision making and explanation. These classes of pathways are characterized by (a) their starting nodes, and (b) their terminal nodes.

Starting nodes or states are those states in the network for which no causes have been defined. The starting nodes are explicitly determined by the structure of the state network; the complete set of possible starting states is independent of any configuration of confirmed states. In Figure 7-2, n_{14} is a starting state and n_{19} is not; n_{19} will never be a starting state, even when all of its causes (n_{18}, n_{17}, n_{14}, n_{20}) are denied. Within the model, a starting state is the most antecedent cause of further progression of disease in a patient. It represents a basic causal mechanism that characterizes a disease process. Any causal pathway that explains the disease process involved in a particular patient can be characterized by its starting state. When a nonstarting state has all of its antecedent clauses denied, this state will not appear on any pathway that attempts to explain the manifestation of disease in a patient. The nonstarting states represent events that should be explained by the events that cause them.

The clinician is usually most concerned with the most likely causes of disease found in a patient. The most likely starting node is taken as the node that explains the greatest number of states of disease. This is the starting state from which pathways (containing no denied nodes) are generated that traverse the greatest number of confirmed nodes. If two or more starting states are found, a likelihood measure is computed for the states, and the starting node with the greatest weight is selected. If a single starting state does not explain all of the confirmed nodes, then another starting state is found that explains the greatest number of remaining states. The procedure is continued until all of the confirmed nodes are explained, and the complete set of most likely starting nodes is identified. The pathways generated from these nodes represent the most likely explanations of the disease processes manifested in the patient.

The physician may also wish to discover alternative though less likely causes that potentially explain the disease mechanisms present in a patient. Potential explanations of the disease processes for a patient can be found by generating all pathways that reach confirmed states, without traversing any denied states. In addition, since a state network is usually designed for a restricted domain of diseases, the clinician may wish to determine those causes of disease that have not yet been eliminated. These may be observed by generating all undenied pathways in the state network.

Observations of a patient are often gathered sequentially. History questions are asked before the physical examination, which precedes the laboratory tests. For a given configuration of the state network, pathways may be generated that, by necessity, are based on an incomplete set of observations. For a specific patient the physician is often interested in determining those disease processes that have not yet been ruled out and may be uncovered by additional observations. Pathways that explain disease processes for a specific patient are usually terminated at a confirmed node. This provides the direct explanation of the events that have been observed. By continuing causal pathways beyond this usual termination point of a con-

firmed node and extending them to include all nodes with an undetermined status, those aspects of the disease process that remain possible can be indicated.

Many diseases are (irreversibly) progressive. Once the particular processes are determined, the physician is concerned with identifying the stages of the disease to which the patient may subsequently proceed. The particular pathways that have been generated to explain observations for a patient may be continued to the terminal nodes of the state network, even if they traverse currently denied nodes. These pathways will give an indication of possible future events, and provide the basis for prognostic assessment.

Example 4. (See Figure 7-2.) Assume that nodes n_{15} and n_{19} are confirmed and the remaining nodes are undetermined; n_{14} will be selected as the most likely mechanism, because it explains both n_{15} and n_{19}. The pathways emanating from n_{14} ($n_{14}\, n_{15}\, n_{16}\, n_4\, n_5\, n_6\, n_7\, n_{18}\, n_{19}$ and $n_{14}\, n_{19}$) directly explain the current observations of the patient. However, for future examinations, more observations may be recorded, and one will probably be interested in continuing the pathways, to check for elevations of intraocular pressure (n_{10} or n_{26}). There are also other mechanisms that are less likely, but that may potentially explain n_{19} (e.g., n_{20}).

7.6 Conflicts and Contradictions

The diagnostician is sometimes faced with the task of interpreting test results that are seemingly conflicting and in some cases contradictory. It is possible to recognize and resolve many conflicts and contradictions because the test results for a patient are interpreted through a model of disease that expresses the meaning of these observations. The model may be viewed as containing an implicit set of consistency conditions that must be satisfied for each patient.

The procedures for interpreting test results have been designed to resolve explicit conflicts in these results. As described earlier, the test result that is held with greatest confidence is taken as the accepted result. If conflicting results are received with equal confidence, then the conflict is noted, and the status of the state of disease remains undetermined until additional results, with greater confidence, resolve the conflict.

A typical contradictory situation occurs when a state is confirmed, yet all of its potential causes in the network are denied. For example, in Figure 7-2, a contradiction would result if n_{19} is confirmed, and n_{18}, n_{17}, n_{14}, and n_{20} are all denied. There is not an admissible pathway to confirmed node n_{19}, because all of the pathways contain a denied node. One potential explanation for this difficulty is that the model of disease may be incom-

plete and some causes (of confirmed node n_{19}) are missing from the network. For example, although it is not indicated in Figure 7-2, n_{33} (OCULAR TRAUMA) may in fact cause n_{19} (PERIPHERAL ANTERIOR SYNECHIAS). The model designer may intentionally not specify all potential causes; instead, he or she may indicate that for some nodes no contradiction should be assumed because the model of causes for these nodes is incomplete. Either the model is incomplete or a contradiction has been found.

Based on a configuration of confirmed and denied nodes in the state network, pathways of disease are generated to explain the processes of disease found in a given patient. Some of the nodes in these pathways may have an undetermined status, with $|\mathrm{Cf}(n_i)| < \Theta$. When the Cf of a node generated in a pathway is undetermined but in the direction of denial, i.e., $-\Theta < \mathrm{Cf}(n_i) < 0$, then the explanation of disease is inconsistent. The explanation provided by the model may be valid, but it indicates that further, more conclusive evidence is needed. If any inconsistencies are found in these pathways, it is important to check for any alternative explanation that, while not the most likely, is entirely consistent with the states that are explained. This can be accomplished by changing the threshold Θ to zero and then finding the most likely starting node. Now all nodes that have been tested with any degree of confidence will be assumed either confirmed or denied. Either the same most likely starting nodes will be selected or alternative mechanisms will be found.

Example 5. (See Figure 7-2.) Assume that n_{35} is the most likely starting node and the pathway $n_{35}\ n_{25}\ n_{26}\ n_{27}\ n_{28}\ n_{29}\ n_{30}\ n_{31}$ is generated. If, however, the status of n_{25} is undetermined in the direction of denial, an inconsistency is indicated. If a search is made for alternative but consistent explanations and n_{25} is assumed denied, then n_{36} is selected as the most likely starting node, and the consistent pathway $n_{36}\ n_{26}\ n_{27}\ n_{28}\ n_{29}\ n_{30}\ n_{31}$ is generated.

7.7 Classification of Diseases

Recognition of the basic mechanisms of disease for a patient often is insufficient for diagnostic classification. An evaluation of the status of a patient must also determine the degree of progression and severity of disease. Patients with the same disease may exhibit different degrees of dysfunction. For example, glaucoma may lead to total blindness, but many cases will be encountered with little or no loss of vision, and these cases must be treated quite differently.

The CASNET system differentiates between two important categories of classification: (a) the mechanism of disease, and (b) the severity and the

degree of progression of disease. The cause or mechanism of disease is described in terms of the state network by the starting nodes. For a given patient, a set of most likely starting nodes will be found that identifies the underlying causal mechanism of disease. Implicit in the most likely pathways that follow from these starting nodes is a description of the progression of the disease. Statements are needed to summarize significant findings that take into account such factors as the current severity of disease and the prognosis for the patient. Additionally, specific and well-established disease labels often exist to give diagnostic descriptions. While each name may directly correspond to a specific mechanism of disease, several mechanisms of disease are frequently summarized by a single name.

The classification tables, (n_1, D_1), (n_2, D_2), ... (n_i, D_i), enable us to produce such descriptions of the status of the patient. These tables contain ordered sets of diagnostic statements interpreting the significance of the various findings and pathways of disease. When the processes of disease found in the patient are known, as displayed by the most likely pathways generated for the patient, classification tables will be searched to determine the appropriate statements.

Each starting state has pointers to the particular classification tables that contain diagnostic statements that evaluate this disease mechanism. Several starting states may refer to the same table, since several causal mechanisms may be included in the same diagnostic category. For a given patient, the most likely starting states point to the appropriate tables. The classification tables contain a series of rules ordered by seriousness of disease. The appropriate diagnostic statement corresponds to the single rule that is satisfied in the table. This rule will correspond to the deepest confirmed state in the pairs (n_k, D_k) that is reached from any of the most likely pathways that refer to this table. In most instances, an additional constraint will be added to the search of the classification table: when a state, n_i, within a table is confirmed, it must be traversed by a pathway generated from a most likely starting node that refers to this table. Otherwise, the statement for n_i is inappropriate; other pathways may refer to n_i in a different table. The deepest state is appropriate since any statement that is found earlier in the table is for a less serious stage of disease and can be ignored.

Example 6. A classification table for the primary open angle mechanism, n_{35}, from Figure 7-2, is given as

$$(n_{25}, D_1), (n_{26}, D_2), (n_{30}, D_3), (n_{31}, D_4)$$

where

D_1 = mild risk of open angle glaucoma
D_2 = high risk of open angle glaucoma
D_3 = very high risk of open angle glaucoma; significant risk of visual field loss
D_4 = open angle glaucoma

If n_{35} is selected as a most likely starting state, and n_{25} and n_{26} are confirmed but not n_{30}, then D_2 is appropriate. If n_{25}, n_{26}, n_{30}, and n_{31} are confirmed, then D_4 is appropriate.

Within a table, differing intensities of a disease process can be determined by differences in the magnitude or intensity of the states. In glaucoma, different intensities of pressure may be distinguished by defining states of moderately elevated pressure or extremely elevated pressure. These states, when found in classification tables, may then lead to different conclusions.

In some instances, it is necessary to have classification rules that indicate that specific states are denied. The same notation and interpretation for a classification table is used, where each entry in the table is not a confirmed state, but rather the required truth value (confirmed, denied, or undetermined) for that state. Multiple causes for a particular patient's disease may be either independent or related. If they are independent, separate classification tables are required. If they are related, the same classification tables are referenced for each of the multiple causes. Rules that are based on truth values (and not confirmation alone) are used to distinguish situations where multiple causes cannot be classified independently.

We can now summarize our diagnostic method as a series of transformations. As test results are received, they are related to individual states. These states are then organized into pathways inferred from configurations of a state network. The generated pathways are then related to classification tables containing the detailed diagnostic categories.

7.8 Treatment Recommendations

In some cases a therapy recommendation can be explicitly linked to a specific diagnostic conclusion. There may be a unique treatment for a given condition. In other instances, a category of treatments may be described (for example, the class of miotic medications) without an indication of a specific medication. For these simplified situations, the recommendation of a treatment is a continuation of the diagnostic statement found in a classification table.

Example 7. The classification table of Example 6 may be augmented to include treatment recommendations as follows:

$$(n_{25}, D_1, T_1), (n_{26}, D_2, T_2), (n_{30}, D_3, T_3), (n_{31}, D_4, T_4)$$

where the treatment recommendations are

T_1 = return visit in 6 months
T_2 = careful follow-up with repeated tension readings
T_3 = careful follow-up or a therapeutic trial with pilocarpine 1% QID
T_4 = miotic therapy (or, if medically uncontrolled, surgery)

The pairs (D_i, T_i) are linked together for this table; they may not always be found together for other tables and other mechanisms of disease.

A recommendation for therapy is usually a more complex problem than is described above. While the number of potential treatments that are applicable to a patient may be greatly reduced by the precision of the diagnosis, many treatments may still remain feasible. One of these treatments must then be selected. In addition, once a treatment is recommended and given to the patient, it is important to evaluate and monitor the effectiveness of that treatment.

While the purpose of therapy is to control and if possible to cure disease, the recommendation of treatment often introduces factors that are external to the original diagnosis. Specific treatments may be contraindicated because of particular conditions of the patient that do not relate directly to the diseases that are modeled. These factors must be considered before a treatment is recommended. For example, age, allergies, and history of other illnesses may all play an important role in the recommendation of a medication. A treatment for disease may in itself cause new processes of dysfunction that are unrelated to the original diagnosis. Many medications are known to cause side effects, and unwanted complications may ensue from surgical procedures.

A plan of action can be designed to select treatments for patients who fall into a particular diagnostic category. A strategy of treatment selection adapts the general treatment plan to the specific circumstances of a patient. The treatment plan must take into account (a) the effectiveness of the current treatment, and (b) indications or contraindications for various therapies.

Diagnostic conclusions for a patient are found by interpreting the specific observations within the model of disease. These diagnostic conclusions will consider severity and progression of disease. The diagnostic statements may then point to one or more treatment plans. These are shown in Figure 7-4. Each plan consists of an ordered list of treatments, $T_{i1}, T_{i2}, \ldots, T_{in}$. The list is ordered by preference: treatment 1 is tried before treatment 2, which will be tried before treatment 3, etc. This plan represents a prototypical sequence of treatments for patients in the appropriate diagnostic categories, as agreed to in advance by the experts in the domain. A strategy for recommending treatment for an individual will usually follow the order

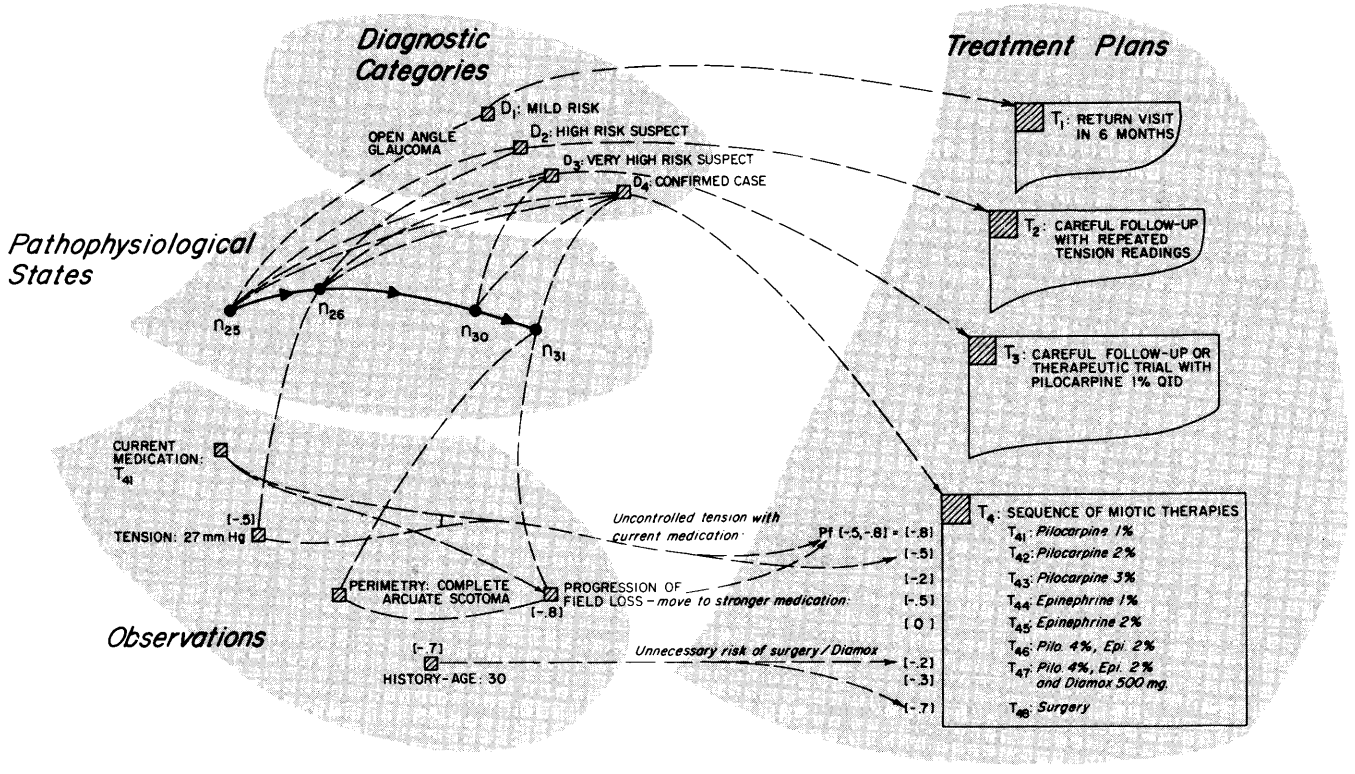


FIGURE 7-4 Examples of treatment plans.

of treatment preferences. However, the ordering may be changed in response to particular observations noted in the individual patient. Within a treatment plan, deviations from the prototype result from changes in the degrees of preference or contraindications for specific treatments. In certain situations, no well-established set of preferences exists, and the selection of a treatment from within the general plan is almost completely determined by the pattern of observations for each patient.

The strategy for treatment selection is described as follows: within a treatment plan, T_k, each specific treatment, T_{kj} has associated with it a preference measure, $\mathrm{Pf}(T_{kj})$, which is assigned from direct observations of the patient. Each observation, t_i, that affects the preference of T_{kj} contributes a measure, $\mathrm{Pf}_{ij}(-1 \leq \mathrm{Pf}_{ij} \leq +1)$, which is assigned in a manner similar to the Q_{im} for relating observations to the disease states, n_m. For example, a drug intolerance may associate a negative preference with a particular treatment. For glaucoma, a very high tension reading after treatment would indicate ineffective control of the disease and contribute a negative preference to the current treatment. Being in a particular age group may increase the preference measure of one treatment over another. The overall preference measure, Pf, is computed by the same rules used to compute the confidence measure, Cf, for the disease states. Once the Pf values have been computed, the rule for selecting a specific treatment, T_{kj}, from within its plan, T_k, can be summarized as follows:

a. Select T_{kj} such that $\mathrm{Pf}(T_{kj}) = \max_i [\mathrm{Pf}(T_{ki})]$.

b. If there is more than a single treatment with maximum Pf, select the one with smallest index j in the *a priori* prototypical ordering.

Example 8. A treatment plan T_4 corresponding to a confirmed case of open angle glaucoma (D_4), as indicated in Example 7, is shown in Figure 7-4. The $\mathrm{Pf}(T_{4j})$ are computed from the observations of a patient, some of which are illustrated in the figure. The patient shown is currently under treatment T_{41}, yet the observed tension of 27 mm of Hg indicates an uncontrolled intraocular pressure. This assigns a decreased preference measure of -0.5 to the current treatment T_{41} and the related treatment T_{42}. The patient also showed progression of field loss, which decreases the preference (-0.8) for the current medication T_{41} even more strongly. Because the patient is under 30, a systemic medication such as Diamox is less preferred (-0.3). The relatively higher risk of surgery versus medication results in the assignment to T_{48} of a decreased preference, -0.7. As a result of comparing these and other Pf's derived from the observations, the treatment with the maximum preference for this patient is T_{45}, which is recommended.

7.9 Results and Discussion

A general method for solving a class of diagnostic and therapy selection problems has been presented. These ideas on model-based interpretation have been put into practice through the implementation of a computer system for medical decision making. Much experience has been gained in the development of a model for the diagnosis and treatment of glaucoma, which has led to the design of a system with a high level of "expertise." This has influenced our general approach toward knowledge representation and reasoning procedures. The consultation system, however, is not specific to glaucoma. Other models of disease have been developed for the anemias, thyroid dysfunction, diabetes, and hypertension. Glaucoma, however, is the one application that has been pursued in depth and has undergone clinical testing.

The design of a consultation system can be broken down into two important tasks. These are the design and representation of models and the design of general problem-solving algorithms that use a suitably defined model for decision making. In the general CASNET system, a medical expert describes or modifies a model, but does not alter the reasoning procedures that select diagnostic interpretation and treatment plans. Two separate computer programs have been developed: the modeling program for designing application models (Kulikowski and Weiss, 1973b), and the consultation program that uses models for reaching diagnoses and recommending therapies (Weiss et al., 1978).

The current glaucoma consultation system has more than 100 states, 400 tests, 75 classification tables, and 200 diagnostic and treatment statements. Results must be interpreted for each eye, so that, in effect, twice the number of rules are involved in any ophthalmological model. There are also many special rules for binocular comparisons of states, tests, and diagnostic and treatment statements. A set of the program's conclusions for a sample case is given in Figure 7-5. This session illustrates the level of performance that the program has attained in reasoning about complex cases of glaucoma.

The consultation program has been designed for efficient performance. Human-engineering aspects of program design have also been emphasized. The program has been developed primarily as a tool for the research of medical decision making by computer. However, our approach to program development involved the collaboration of a network of physicians with minimal prior experience in the use of computers. Their active participation in the project required careful attention to programming details that would allow our collaborators and other ophthalmologists to use the programs with little difficulty. This implies that only limited typing

VISIT 1:

RIGHT EYE:

(1) PRESENT DIAGNOSTIC STATUS:

PIGMENTARY GLAUCOMA. OPEN ANGLE GLAUCOMA. CHARACTERISTIC VISUAL FIELD LOSS WITH CORRESPONDING DISC CHANGES. EARLY FIELD LOSS.

(2) TREATMENT RECOMMENDATIONS:

PILOCARPINE 2% QID.

RESEARCH STUDIES

ALTERNATIVE INTERPRETATIONS OF PIGMENTARY GLAUCOMA:
. SECONDARY GLAUCOMA
. PRIMARY OPEN ANGLE GLAUCOMA

REFERENCES:

1. "WHEN PIGMENTARY GLAUCOMA WAS FIRST DESCRIBED IT WAS THOUGHT TO BE A FORM OF SECONDARY GLAUCOMA CAUSED BY PLUGGING OF THE TRABECULAR MESHWORK BY THE SAME PIGMENT THAT FORMED THE KRUKENBERG'S SPINDLES. HOWEVER, AN INCREASING NUMBER OF OBSERVERS NOW BELIEVE THAT IT IS A VARIANT OF PRIMARY OPEN ANGLE GLAUCOMA..." (WILENSKY, PODOS 1975, TRANSACTIONS NEW ORLEANS ACAD. OPTH.)
2. "MORE RECENT EVIDENCE SUGGESTS THAT PIGMENTARY GLAUCOMA IS A SEPARATE ENTITY..." (ZINK, PALMBERG, ET AL, A.J.O., SEPT. 1975)

VISIT 7:

RIGHT EYE:

(1) PRESENT DIAGNOSTIC STATUS:

PIGMENTARY GLAUCOMA. OPEN ANGLE GLAUCOMA. CHARACTERISTIC VISUAL FIELD LOSS WITH CORRESPONDING DISC CHANGES. ADVANCED FIELD LOSS. CURRENT MEDICATION HAS NOT CONTROLLED IOP IN THE EYE. (AS INDICATED BY PROGRESSION OF CUPPING) (AS INDICATED BY VISUAL FIELD LOSS PROGRESSION)

(2) TREATMENT RECOMMENDATION:

FILTERING SURGERY IS INDICATED. AS AN ALTERNATIVE, PHOSPHOLINE MAY BE TRIED (BUT NOT USED 2 WEEKS BEFORE SURGERY).

FIGURE 7-5 Examples of program-generated decisions for a case of pigmentary glaucoma, abstracted from a sequence of seven visits.

would be required and that quick response time, even for complex diagnostic interpretations, would be essential.

Initially, we designed and built a prototype model that was demonstrated to a select audience of ophthalmologists. At this point, the program was far from being expert. However, rapid progress in the development of a decision-making system can be made by building a small simplified prototype and modifying and improving the prototype.

A very significant event in the development of the program has been the formation of ONET—the Ophthalmological Network. Using the SUMEX-AIM computer,[1] we have put together a nationwide group of ophthalmological clinician-researchers who have participated in the development of the program's knowledge base. They enter cases and suggest improvements. Their suggestions have not been based on a comprehensive review of the logical rules contained in the program. Rather, we have concentrated on entering realistic cases and comparing the program's questioning sequence and conclusions with those of the experts.

Within a period of approximately a year and a half of ONET collaboration, the program achieved an expert level in the long-term diagnosis and treatment of many types of glaucoma. The program's performance has been validated by our group of experts and by the system's participation in panel discussions of glaucoma cases at ophthalmological symposia. In November 1976 a scientific exhibit of the program was presented at the annual meeting of the American Academy of Ophthalmology and Otolaryngology. Ophthalmologists were invited to present difficult cases to the computer. The program did well, with 77% of the ophthalmologists who entered cases describing the program as performing at an expert or very competent level (Weiss et al., 1978).

In comparing the experiences of modeling glaucoma and other diseases, we have obtained some insight into the advantages and limitations of the CASNET representation and its associated decision-making methods. When an understanding of the mechanisms of disease serves as a basis for decision making, the CASNET approach is most valuable. When reasoning is mostly judgmental and based more on empirical information than knowledge of the disease mechanisms, other decision models may prove more appropriate (Patrick et al., 1974; Shortliffe et al., 1973).

In the MYCIN system (see Chapter 5), descriptive domain knowledge is implicitly contained within the system of production rules that encode the clinical judgment of an expert consultant. Therapy selection for infectious diseases is a medical domain in which empirical knowledge plays a predominant part in the problem-solving process, and it is not surprising that this domain has been successfully modeled in terms of judgmental rules alone. In glaucoma, as in other diseases where mechanisms of dysfunction are reasonably well known and have an important effect on the selection of treatments, we have developed a more structured representation for causal knowledge. And yet, since strict Aristotelian causality is hardly applicable in medicine, the causal representation is embedded within an associational structure of observations that accounts for the uncertainties of clinical findings.

In questions of hypothesis generation and approximate reasoning, the

[1]This computer was established at Stanford University with NIH support to provide a national shared resource for research in AIM (AI in medicine).

CASNET approach is quite distinctive in its use of the causal-associational structuring of knowledge. An overall diagnostic hypothesis for a patient is usually a composite of several hypotheses. It is not uncommon to find five or six hypotheses included in the final diagnostic statement. Many of these hypotheses include statements of uncertainty within them, as, for example, "very high risk of glaucoma" or "mild risk of glaucoma."

Approximate reasoning takes place at several levels. Measures of uncertainty are used to interpret observations in terms of the most elementary subhypotheses: the pathophysiological states of the causal network. A thresholding of the measures of uncertainty for all observations that are relevant to a given state determines whether that state is to be considered a "confirmed," "denied," or "undetermined" subhypothesis for the patient. At this level the method corresponds to the usual approach of assigning a likelihood or degree of belief to a hypothesis. At a higher level of abstraction these subhypotheses of states are grouped together in a more deterministic fashion. Measures of uncertainty are less important at this stage because the hypotheses themselves include qualifying statements as described above. Thus the greatest reduction of uncertainty takes place between the observations and the states, which serve as local and relatively simple summaries of events in the course of a disease. The detailed structural relationships among states allows a fine-resolution encoding of the possible patterns of the disease. Because statements of uncertainty are associated with these patterns, they can be related to final hypotheses in a deterministic logical manner without losing the soundness of the outcome. It is often advantageous to do this, in so far as it corresponds more closely to the conclusions expressed by an expert physician.

The explanations produced by the CASNET/Glaucoma system also appear to correspond more closely to those of the physician. Instead of tracing all the rules involved in arriving at the final diagnosis, the composite hypothesis includes certain key subhypotheses that the physician recognizes as necessary elements in justifying the conclusions or recommendations. For example, in glaucoma, a typical subhypothesis would be "corresponding disc and visual field changes," which is both explanatory and supportive of a higher-level hypothesis of "open angle glaucoma." The subhypothesis is itself the summary of many different observations. In building the CASNET system, we have found that exhaustive tracing of rules is much more valuable as a debugging tool than as an explanation for the physician.

The CASNET/Glaucoma system has proved to be highly efficient and sufficiently expert to be accepted as such by many ophthalmologists. Its solutions to many of the representational and strategy questions have been shown to be effective in a realistic problem domain. Nevertheless, the role of such large knowledge-based consultation systems in routine clinical practice remains an open question.

ACKNOWLEDGMENTS

Supported in part by grant RR-643 from the Biotechnology Resources Program, Division of Research Resources, NIH, and in part by grant R01-MB-00161 of the Health Resources Administration.

8

INTERNIST-1, An Experimental Computer-Based Diagnostic Consultant for General Internal Medicine

Randolph A. Miller, Harry E. Pople, Jr., and Jack D. Myers

One of the best-known AIM systems is the large diagnostic program constructed by researchers at the University of Pittsburgh during the 1970s. The work developed out of a collaboration between Harry Pople (a computer scientist with an interest in AI, logic programming, and medical applications) and Jack Myers, university professor (medicine) and prominent clinician, who was eager to try to encode some of his diagnostic expertise in a high-performance computer program. Rather than selecting a small subtopic in medicine for the work, Pople and Myers decided to consider the entire field of internal medicine. This necessarily required approaches that quickly narrowed the search space of possible diseases and also permitted case analyses in which two or more diseases could coexist and interact. The resulting program, now known as INTERNIST-1 (or INTERNIST, for short), is capable of making multiple and complex diagnoses in internal medicine. It differs from other programs for computer-assisted diagnosis in the generality of its approach and in the size and diversity of its knowledge base.

The knowledge base was developed over several years by Myers and medical student assistants. One of these students, Dr. Randolph Miller, became involved in the programming as well and, as a clinical faculty

member at the University of Pittsburgh, continues as a principal collaborator on the project. Those building the knowledge base would study the major diseases in medicine one by one, identifying both their major and minor clinical manifestations and developing weights that link each finding with the diseases in which it can occur. The resulting ad hoc scoring scheme proved to be capable of guiding excellent diagnostic reasoning. To test the program during its development, Myers and his students would select especially difficult cases for consideration, often ones drawn from published clinical pathological conferences in medical journals.

After several years of testing and refinement of the knowledge base, the study outlined in the following chapter was performed. To document the strengths and weaknesses of the program, the group performed a systematic evaluation of the program's capabilities. Its performance on a series of 19 clinicopathological exercises ("Case Records of the Massachusetts General Hospital"), published in the New England Journal of Medicine, *appeared qualitatively similar to that of the hospital clinicians but inferior to that of the case discussants. As a result, Miller, Pople, and Myers believe that the evaluation demonstrated that the present form of the program is not sufficiently reliable for clinical applications. They cite specific deficiencies that must be overcome before the program is ready for clinical use: an ability to construct differential diagnoses spanning multiple problem areas, new methods to avoid occasional attribution of findings to improper causes, and human-engineering enhancements to allow the program to explain its "thinking." A more detailed discussion of the serious limitations in the underlying representation and control methods used in INTERNIST-1 has recently been presented by Pople (1982). In that article Pople explains the contemplated enhancements that will be the basis for the next version of INTERNIST, to be known as CADUCEUS.*

8.1 Introduction

INTERNIST-1, an experimental program for computer-assisted diagnosis in general internal medicine, differs considerably in scope from other medical diagnostic computer programs. In the past, techniques including mathematical modeling, use of Bayesian statistics, pattern recognition, and other approaches (Wardle and Wardle, 1978; Wagner et al., 1978) (see also Chapter 3), have been shown to be useful in circumscribed areas such as the differential diagnosis of abdominal pain (deDombal et al., 1972) and the diagnosis and treatment of meningitis (Yu et al., 1979a). However, no program developed for use in a limited domain has been successfully adapted for more generalized use. From its inception, INTERNIST-1 has addressed the problem of diagnosis within the broad context of general internal medicine (Pople et al., 1975; Myers et al., 1982; Pople, 1982). Given a patient's

initial history, results of a physical examination, or laboratory findings, INTERNIST-1 was designed to aid the physician with the patient's work-up in order to make multiple and complex diagnoses. The capabilities of the system derive from its extensive knowledge base and from heuristic computer programs that can construct and resolve differential diagnoses.

The INTERNIST-1 program represents an example of applied symbolic reasoning (artificial intelligence). A variety of such techniques have been developed by computer scientists in an attempt to model the thought processes and problem-solving methods employed by human beings (Winston, 1977; Nilsson, 1980). An important aspect of the INTERNIST-1 approach to computer-assisted diagnosis is that the program attempts to form an appropriate differential diagnosis in individual problem areas. A *problem area* is defined as a selected group of observed findings, the differential diagnosis of which forms what is assumed to be a mutually exclusive, closed (i.e., exhaustive) set of diagnoses. Physicians routinely construct such closed differential diagnoses on the basis of causal considerations (e.g., bacterial pneumonias) or pathoanatomic considerations (e.g., causes of obstructive jaundice). By constructing specific differential diagnoses to address identified problem areas, a physician or computer program can narrow the set of possible diagnoses from all known diseases to well-defined collections of competing diagnoses in a small number of categories. Heuristic principles, such as diagnosis by exclusion, can then be employed to resolve each differential diagnosis. The use of such strategies in INTERNIST-1 represents an attempt to model the behavior of physicians.

Reported below is the first systematic evaluation of INTERNIST-1. The purpose of the study was to illustrate the strengths and weaknesses of the program and to provide a rough estimate of its clinical acumen. The trial was conducted with clinicopathological conferences (CPC's) that had been published in the *New England Journal of Medicine* (NEJM) but had not previously been analyzed by the system. The CPC's fulfill the criteria of being diagnostically challenging cases and of containing sufficiently detailed information to allow computer analysis. The evaluation was not intended to validate INTERNIST-1 for clinical use. CPC's should not be used for such a purpose, and as the trial demonstrated, the program does not yet possess sufficient reliability for clinical application. Nevertheless, INTERNIST-1 performed remarkably well, considering the simple, *ad hoc* nature of its algorithms.

8.2 The INTERNIST-1 Knowledge Base

A medical knowledge base must meet the needs of any associated diagnostic programs. In particular, the INTERNIST-1 knowledge base was designed to permit the consultant program to construct and resolve differential diagnoses. The knowledge base incorporates individual disease

profiles, which list findings that can occur in patients with each illness. By inverting the disease profiles with use of a computer program, an exhaustive differential diagnosis for each finding is obtained; these manifestation-based differential-diagnosis lists are retained as part of the knowledge base. The diagnostic program can use these lists to construct differential diagnoses in clinical cases.

How to group potential diagnoses into relevant problem areas is a separate consideration. The individual diseases in the INTERNIST-1 knowledge base are part of a *disease hierarchy* that is organized from the general to the specific. For example, acute viral hepatitis is classified as an hepatocellular infection, hepatocellular infection is a subclass of diffuse hepatic parenchymal disease, and diffuse hepatic parenchymal disease falls into the category of hepatic parenchymal disease, which is a major subclass of diseases of the hepatobiliary system. Initially, it was thought that access to the disease hierarchy would allow INTERNIST-1 to construct appropriate differential diagnoses (i.e., problem areas) based on higher-level concepts such as hepatocellular infection. If several diagnoses representing types of hepatocellular infection were under consideration, it would be simple to create a problem area for hepatocellular infection. However, early experience with the system showed that a rigid hierarchical classification scheme was inadequate, since a single disease often merits simultaneous categorization under more than one heading. Infectious mononucleosis is both a hepatocellular infection and a type of infectious lymphadenopathy. Hierarchical classification would require that it be listed as one or the other, but not both. An additional concern is that diseases may present differently in different patients. For example, alcoholic hepatitis may occur with predominance of intrahepatic cholestasis in one patient and with massive hepatocellular necrosis in another. Solution of the classification problem entailed development of algorithms (discussed below) that permit INTERNIST-1 to construct problem areas in an *ad hoc* manner.

The building block for the INTERNIST-1 data base is the individual disease. For each diagnosis entered into the system, a disease profile is constructed. The disease profile consists of findings (historical items, symptoms, physical signs, and laboratory abnormalities) that have been reported to occur in association with the disease, including demographic data and predisposing factors. Two clinical variables are associated with each manifestation in an INTERNIST-1 disease profile: an evoking strength and a frequency. The evoking strength answers the question "Given a patient with this finding, how strongly should I consider this diagnosis to be its explanation?" The frequency is an estimate of how often patients with the disease have the finding. In addition, each manifestation is assigned a disease-independent import. The import is the global importance of the manifestation—that is, the extent to which one is compelled to explain its presence in any patient. Although the evoking strengths, frequencies, and imports are expressed as numbers (on a scale of 0 to 5 or 1 to 5) in the INTERNIST-1 knowledge base, it is important to remember that they rep-

resent a shorthand for judgmental information, as their suggested interpretations in Tables 8-1 through 8-3 indicate. True quantitative information does not exist in the medical literature in most cases; the numbers used by INTERNIST-1 are judgmental in that they are compiled after a review of the available knowledge.

The current INTERNIST-1 knowledge base, which represents 15 person-years of work, encompasses over 500 individual disease profiles (an example appears in Figure 8-1) and approximately 3550 manifestations of disease. The disease profiles have been generated by review of the literature and by consultation with expert clinicians. In addition to the disease profiles, the knowledge base details relations among diagnoses and among manifestations. Within INTERNIST-1, important high-level pathophysiologic states (such as acute left ventricular failure, chronic congestive left heart failure, prerenal azotemia, and chronic uremia) are profiled as if they were diseases. The knowledge base contains links between such "diseases" and other diseases. The links are used to express causality or a predisposition of patients with one disease to have another. Because INTERNIST-1 formulates and resolves problem areas serially, it can piece together interdependent components of a multisystem illness one by one, using the links in the data base to promote consideration of diseases related to previously concluded diagnoses. The total number of links among the 500 diagnoses in the data base is about 2600. The 3550 manifestations in the INTERNIST-1 knowledge base are not independent. Men do not have oligomenorrhea, and a patient with oligomenorrhea must be presumed to be female. The knowledge base includes the properties of each manifestation that specify how its presence or absence may influence the presence or absence of other manifestations. There are roughly 6500 such interrelationships detailed in the knowledge base.

8.3 The Diagnostic Algorithms

The problem-solving algorithms represent the intellectual core of the INTERNIST-1 system. Although the scoring mechanism described below manipulates probabilistic data (evoking strengths, frequencies, and imports), it must be emphasized that the behavior of INTERNIST-1 results primarily from application of two heuristic principles: formation of problem areas via a partitioning algorithm, and conclusion of diagnoses within problem areas using strategies such as diagnosis by exclusion.

The steps on pages 197–200 are taken during an INTERNIST-1 diagnostic consultation. [Please refer to Section 8.6 for an annotated sample case analysis taken from a CPC published in the *New England Journal of Medicine* (Castleman, 1969).]

TABLE 8-1 Interpretation of Evoking Strengths

Evoking strength	*Interpretation*
0	Nonspecific—manifestation occurs too commonly to be used to construct a differential diagnosis.
1	Diagnosis is a rare or unusual cause of listed manifestation.
2	Diagnosis causes a substantial minority of instances of listed manifestation.
3	Diagnosis is the most common but not the overwhelming cause of listed manifestation.
4	Diagnosis is the overwhelming cause of listed manifestation.
5	Listed manifestation is pathognomonic for the diagnosis.

TABLE 8-2 Interpretation of Frequency Values

Frequency	*Interpretation*
1	Listed manifestation occurs rarely in the disease.
2	Listed manifestation occurs in a substantial minority of cases of the disease.
3	Listed manifestation occurs in roughly half the cases.
4	Listed manifestation occurs in the substantial majority of cases.
5	Listed manifestation occurs in essentially all cases—i.e., it is a prerequisite for the diagnosis.

TABLE 8-3 Interpretation of Import Values

Import	*Interpretation*
1	Manifestation is usually unimportant, occurs commonly in normal persons, and is easily disregarded.
2	Manifestation may be of importance, but can often be ignored; context is important.
3	Manifestation is of medium importance, but may be an unreliable indicator of any specific disease.
4	Manifestation is of high importance and can only rarely be disregarded, as, for example, a false-positive result.
5	Manifestation absolutely must be explained by one of the final diagnoses.

```
DISPLAY WHICH MANIFESTATION LIST?
ALCOHOLIC HEPATITIS

AGE 16 TO 25 ...0 1
AGE 26 TO 55 ...0 3
AGE GTR THAN 55 ...0 2
ALCOHOL INGESTION RECENT HX ...2 4
ALCOHOLISM CHRONIC HX ...2 4
SEX FEMALE ...0 2
SEX MALE ...0 4
URINE DARK HX ...1 3
WEIGHT LOSS GTR THAN 10 PERCENT ...0 3
ABDOMEN PAIN ACUTE ...1 2
ABDOMEN PAIN COLICKY ...1 1
ABDOMEN PAIN EPIGASTRIUM ...1 2
ABDOMEN PAIN NON-COLICKY ...1 2
ABDOMEN PAIN RIGHT UPPER QUADRANT ...1 3
ANOREXIA ...0 4
DIARRHEA ACUTE ...1 2
MYALGIA ...0 3
VOMITING RECENT ...0 4
ABDOMEN BRUIT CONTINUOUS RIGHT UPPER QUANDRANT ...1 2
ABDOMEN TENDERNESS RIGHT UPPER QUADRANT ...2 4
CONJUNCTIVA AND/OR MOUTH PALLOR ...1 2
FECES LIGHT COLORED ...1 2
FEVER ...0 4
HAND(S) DUPUYTRENS CONTRACTURE(S) ...1 2
JAUNDICE ...1 3
LEG(S) EDEMA BILATERAL SLIGHT OR MODERATE ...1 2
LIVER ENLARGED MASSIVE ...1 2
LIVER ENLARGED MODERATE ...1 3
LIVER ENLARGED SLIGHT ...1 2
PAROTID GLAND(S) ENLARGED ...1 2
SKIN PALLOR GENERALIZED ...0 2
SKIN PALMAR ERYTHEMA ...1 3
SKIN SPIDER ANGIOMATA ...2 3
SKIN TELANGIECTASIA ...1 1
ALKALINE PHOSPHATASE BLOOD GTR THAN 2 TIMES NORMAL ...1 2
ALKALINE PHOSPHATASE BLOOD INCREASED NOT OVER 2 TIME NORMAL ...1 4
BILIRUBIN BLOOD DECREASED ...2 2
BILIRUBIN URINE PRESENT ...2 4
CHOLESTEROL BLOOD DECREASED ...2 2
CHOLESTEROL BLOOD INCREASED ...1 2
HEMATOCRIT BLOOD LESS THAN 35 ...1 3
HEMOGLOBIN BLOOD LESS THAN 12 ...1 3
KETONURIA ...1 2
PROTEINURIA ...1 2
SGOT 120 TO 400 ...2 3
SGOT 40 TO 119 ...2 3
SGOT GTR THAN 400 ...1 2
UREA NITROGEN BLOOD LESS THAN 8 ...2 2
UROBILINOGEN URINE ABSENT ...1 1
UROBILINOGEN URINE INCREASED ...2 4
```

FIGURE 8-1 A sample manifestations list. The first number after each manifestation is its evoking strength for the diagnosis; the second is the frequency of the manifestation in the disease.

WBC 14000 TO 30000 ...0 3
WBC 4000 TO 139000 PERCENT NEUTROPHIL(S) INCREASED ...0 3
WBC LESS THAN 4000 ...1 1
ACTIVATED PARTIAL THROMBOPLASTIN TIME INCREASED ...1 3
ANTIBODY MITOCHONDRIAL ...1 1
ANTIBODY SMOOTH MUSCLE ...2 3
BSP RETENTION INCREASED ...1 5
ELECTROPHORESIS SERUM ALBUMIN DECREASED ...2 4
ELECTROPHORESIS SERUM GAMMA GLOBULIN INCREASED ...2 4
FACTOR VII PROCONVERTIN DECREASED ...1 2
LDH BLOOD INCREASED ...1 3
MAGNESIUM BLOOD DECREASED ...2 2
PROTHROMBIN TIME INCREASED ...2 3
SGPT 200 TO 600 ...1 2
SGPT 40 TO 199 ...2 3
SGPT GTR THAN 600 ...1 1
LIVER BIOPSY BILE PLUGGING ...1 2
LIVER BIOPSY FATTY METAMORPHOSIS ...2 4
LIVER BIOPSY FOCAL NECROSIS AND INFLAMMATION ...2 5
LIVER BIOPSY HEPATOCELLULAR NECROSIS MARKED ...2 3
LIVER BIOPSY MALLORY BODIES ...3 3
LIVER BIOPSY PERIPORTAL FIBROSIS MILD ...1 3
LIVER BIOPSY PERIPORTAL INFILTRATION NEUTROPHIL(S) ...3 5
LIVER BIOPSY PERIPORTAL INFILTRATION ROUND CELL(S) ...1 2
LIVER BIOPSY SMALL BILE DUCT(S) PROMINENT ...1 2

LINKS FOR ALCOHOLIC HEPATITIS:

Predisposes to	MALLORTY WEISS SYNDROME ...1 1
Causes	SINUSOIDAL OR POSTSINUSOIDAL PORTAL HYPERTENSION ...1 2
Causes	HEPATIC ENCEPHALOPATHY ...2 2
Causes	RENAL FAILURE SECONDARY TO LIVER DISEASE <HEPATORENAL SYNDROME> ...2 2
Coincident with	PANCREATITIS ACUTE ...2 2
Precedes	MICRONODAL CIRRHOSIS <LAENNECS> ...2 3

FIGURE 8-1 continued

1. Initial positive (present) and negative (absent) patient findings are entered by the user. As each new positive manifestation is encountered, the program retrieves its complete differential diagnosis from the inverted disease profiles in the knowledge base. A *disease hypothesis* is created for each item on the manifestation's differential-diagnosis list. A master list of all such disease hypotheses is maintained. Higher-level concepts from the classification hierarchy are retained on the differential-diagnosis list as long as the diagnoses that they subsume are indistinguishable in their ability to explain the observed data. The master differential list therefore comprises all possible diagnoses that can explain any of the observed findings (taken either individually or in groups).
2. For each disease hypothesis, four lists are maintained: all positive manifestations in the patient that are explained by the disease hypothesis (i.e., findings matching the disease profile stored in the data base); all manifestations that might occur in a patient with the disease but are

known to be absent in the patient being considered; all manifestations present in the patient but not explained by the disease hypothesis, that is, not found on the disease profile (these manifestations represent either "red herrings" or items that would have to be explained by a second disease present in the patient); and manifestations on the disease's profile about which nothing is known (this list is used in determining which questions to ask).

3. Each hypothesis on the master list of diagnoses is given a score. Scores are calculated as the sum of a positive and a negative component as follows. The positive component includes the weights of all manifestations explained by the hypothesis, based on the evoking strengths of the observed manifestations for the diagnosis. A nonlinear weighting scheme is used: an evoking strength of 0 counts as 1 point; a strength of 1 counts as 4 points; a 2 counts as 10 points; a 3 counts as 20; a 4 as 40; and a 5 as 80. Any disease hypothesis related to a previously concluded diagnosis (through links in the data base) is given a bonus score. The bonus awarded is 20 points times the frequency number listed for the hypothesized diagnosis in the disease profile of the concluded diagnosis. The negative component includes the weight of all manifestations that are expected to occur in patients with the disease but are absent in the patient under consideration. A nonlinear scale based on the expected frequency of the manifestation in the disease is used: a frequency of 1 counts as -1 point; a 2 as -4 points; a 3 as -7 points; a 4 as -15 points; and a 5 as -30 points. Also included are the weights of all manifestations present in the patient but not explained by the hypothesized diagnosis. The import (clinical significance) of each manifestation is used to assess this penalty: an import of 1 counts as -2 points; a 2 as -6 points; a 3 as -10 points; a 4 as -20 points; and a 5 as -40 points. The net score for any disease hypothesis is thus the sum of the above four component weights.
4. After all disease hypotheses have been scored, the master list of all hypotheses is sorted by descending score. Diagnoses whose scores fall a threshold number of points below the topmost diagnosis are temporarily discarded as unattractive. They may be reconsidered, however, if further evidence obtained during the case analysis raises their scores above the threshold (relative to the topmost diagnosis).
5. At this point, the sorted master differential-diagnosis list is a heterogeneous grouping of many disease hypotheses. A critical step in the diagnostic logic of INTERNIST-1 is to delineate a set of competitors for the topmost diagnosis (i.e., to create a problem area containing the topmost disease hypothesis). Only one of the set of diseases in a properly defined problem area is likely to be present in a patient. Problem area construction is carried out by the INTERNIST-1 partitioner, which employs a remarkably powerful yet simple heuristic rule. The rule states, "Two diseases are competitors if the items not explained

by one disease are a subset of the items not explained by the other; otherwise, they are alternatives (and may possibly coexist in the patient)." To paraphrase, if Disease A and Disease B taken together explain no more observed manifestations than does either one taken alone, then the diseases are classified as competitors. Competitors for the likeliest diagnosis are identified from the master differential list using the partitioning rule; including the topmost diagnosis, they constitute the *current problem area.* Because INTERNIST-1 defines problem areas in this *ad hoc* manner, its differential diagnoses will not always resemble those constructed by clinicians.

6. Once the problem area containing the most attractive diagnosis has been selected, criteria for establishing a definitive diagnosis can be applied. If the problem area contains only the topmost diagnosis, INTERNIST-1 will immediately decide on (conclude) that diagnosis. If there is more than one diagnosis in the problem area, INTERNIST-1 directly concludes the leading diagnosis when its score is 90 or more points higher than the nearest competitor. The value of 90 was chosen because it slightly exceeds the weight carried by a pathognomonic finding (80 points). This method of concluding a diagnosis is a hallmark of INTERNIST-1. The absolute score of the diagnosis does not matter. The only point of importance is whether the diagnosis is sufficiently higher in score than its reasonable competitors (other diagnoses that explain the same set of findings).
7. If it is not possible to conclude a diagnosis (which by default means that the current problem area contains more than one hypothesis), one of three questioning strategies is selected: pursuing, ruling out, or discriminating. The pursuing mode is selected if the second-best contender is 46 to 89 points behind the topmost diagnosis. In the pursuing mode, questions are asked to establish the topmost diagnosis, since it is close to fulfilling criteria for conclusion. The questions asked are those that are most specific for the leading diagnosis (i.e., those with high evoking strengths). If there are five or more diagnoses within 45 points of the topmost diagnosis, the ruling-out mode is used. Questions that have high frequency numbers under the contenders are asked, with the expectation that several negative responses will remove some of the diagnoses from contention. The discriminating mode is used when there are two to four diagnoses within 45 points of the leading diagnosis. The questions asked attempt to maximize the spread in scores.
8. In order to improve the efficiency of computations, questions are asked in small groups. The level of questioning is escalated (from history to physical-examination findings to gradations of laboratory results) only after the useful questions in a previous category have been exhausted. After the answers are processed, the disease hypotheses are again scored and partitioned. A new differential diagnosis

is formed on the basis of the (possibly) new topmost diagnosis. This *ad hoc* method for constructing a differential diagnosis gives INTERNIST-1 seemingly intelligent behavior, since the program will often change focus from one problem area to another when questioning in the first area has been counterproductive.

9. When a diagnosis is concluded, all observed manifestations explained by the diagnosis are removed from future consideration. The program then recycles using the remaining unexplained positive findings. Subsequent findings are marked as explained when a previously concluded diagnosis can account for them. However, it is not possible to undo a previous diagnostic conclusion when contradictory evidence becomes available.
10. When a problem area contains more than one disease hypothesis and all useful lines of questioning have been exhausted (without meeting criteria for concluding the topmost diagnosis), the program will defer making a diagnosis in that problem area. Diagnoses in the problem area are then displayed by descending score, along with an explanation that the differential diagnosis cannot be resolved.
11. When all remaining manifestations have an import of 2 or less, the program stops.

8.4 An Evaluation of INTERNIST-1

We have completed a preliminary evaluation of INTERNIST-1. The program was evaluated to compare its clinical acumen to that of human experts and to highlight its strengths and weaknesses. CPC's published in the *New England Journal of Medicine* (NEJM) as "Case Records of the Massachusetts General Hospital" were used for the computer analysis. During the trial, only the published findings available to the case discussant were presented to INTERNIST-1 (i.e., only findings mentioned before the presentation of the pathological findings). The knowledge base of INTERNIST-1 was not altered during the course of the evaluation.

During the development of INTERNIST-1, hundreds of miscellaneous individual cases, both simple and complex, have been presented to the system in order to evaluate and improve the data base and the diagnostic computer program. Since many of these test cases included NEJM CPC's, cases for the trial were selected from 1969, a year from which no previous NEJM cases had been presented to INTERNIST-1. Before entering any cases, project members serially reviewed the published final anatomic diagnoses. All cases in which one or more of the major diagnoses were not represented in INTERNIST-1's still incomplete knowledge base were rejected. The diagnostic program cannot conclude a diagnosis that is

missing from the knowledge base; such a case would not be a fair test for the system. The excluded diagnoses were neither more rare nor more complex than the diagnoses chosen for analysis. Cases 1-1969 through 42-1969 (inclusive) were reviewed, and 19 cases were obtained in which all major CPC diagnoses were included in the data base. That only 19 of the 42 cases reviewed qualified for the study is not unexpected. It is estimated that the current INTERNIST-1 knowledge base includes roughly 70–75% of the major diagnoses of internal medicine. If each case on the average contained three major diagnoses, the probability that all three diagnoses would be included in the knowledge base is (0.75) × (0.75) × (0.75) or 42%.

In establishing criteria for evaluating performance on the NEJM CPC's, one must classify final anatomic or clinical diagnoses as major or minor. Major diagnoses are defined as those central to the problem. Classified as minor diagnoses are diseases that were present in the patient but were clinically less relevant, including those diseases only partially described in the published case protocol, as well as conditions that were successfully managed and that subsequently resolved. Diagnostic decisions made by the clinicians at the Massachusetts General Hospital (MGH), by the case discussants, and by INTERNIST-1 were classified as correct when they were confirmed by the pathologists or when a clinical syndrome was universally agreed to be present. When either the physicians or INTERNIST-1 introduced an incorrect diagnosis, a separate notation was made because an incorrect diagnosis has a different meaning from that of a failure to make a correct diagnosis. We recognize two ways for a program or a clinician to make a correct diagnosis in the setting of a CPC: to state unequivocally that the patient has the disease *(definitive diagnosis)* or to offer an unresolved differential diagnosis that includes the correct diagnosis as its topmost element *(tentative diagnosis)*. INTERNIST-1 makes definitive diagnoses by conclusion and tentative diagnoses by deferral (see above). The hospital clinicians and the case discussants also made both types of diagnoses. A tentative diagnosis was counted as incorrect if its topmost element was not the correct diagnosis, even if the associated differential diagnosis included the correct diagnosis.

Table 8-4 summarizes the results for the 19 trial cases. There were 43 possible correct major diagnoses. INTERNIST-1, the clinicians at the MGH, and the case discussants made 17, 23, and 29 correct definitive diagnoses, respectively. A correct tentative diagnosis was offered 8, 5, and 6 times, respectively. Thus, of 43 anatomically verified diagnoses, INTERNIST-1 failed to make a total of 18, whereas the clinicians failed to make 15 such diagnoses, and the discussants missed only 8. Of the 18 situations in which INTERNIST-1 failed to make an anatomically correct diagnosis, the clinicians or the discussant or both failed to make the correct diagnosis 11 times. INTERNIST-1 made a correct diagnosis in 7 circumstances in which the clinicians or the case discussant failed to do so. INTERNIST-1 made 5 incorrect definitive diagnoses and 6 incorrect tentative

TABLE 8-4 Summary of results for major diagnoses in 19 cases used in the INTERNIST-1 evaluation

	No. of instances		
Category	*INTERNIST-1*	*Clinicians*	*Discussants*
Definitive, correct	17	23	29
Tentative, correct	8	5	6
Failed to make correct diagnosis	18	15	8
Definitive, incorrect	5	8	11
Tentative, incorrect	6	5	2
Total no. of incorrect diagnoses	11	13	13
Total no. of errors in diagnosis	29	28	21
Total possible diagnoses	43	43	43

diagnoses (naming diseases that were not present in the patients). The MGH clinicians made 8 incorrect definitive diagnoses and 5 incorrect tentative diagnoses. The case discussants made 11 incorrect definitive diagnoses and 2 incorrect tentative diagnoses. Of the 5 situations in which INTERNIST-1 made an incorrect definitive diagnosis, 4 were situations in which the discussants also made a wrong diagnosis.

The shortcomings of the program, which were highlighted by the evaluation, fall into two general categories. The first type are limitations due to the structure or content of the knowledge base. Examples include the absence of a manifestation required to describe an important finding; the use of overly simplistic manifestations for some circumstances; the inadvertent omission of a finding from a disease profile; the assignment of an incorrect evoking strength, frequency, or import; and the failure of a manifestation to convey adequate anatomic information. The second type of limitation resulted from deficiencies in the design or implementation (or both) of the computer program. Included in this category were failure to incorporate temporal reasoning capabilities; problems resulting from use of the scoring algorithm; the inability to take a broad overview in attacking a complex problem; and the improper attribution of findings to concluded diagnoses (i.e., invoking the wrong explanation for a finding). Specific reasons for INTERNIST-1's incorrect diagnoses (made both by omission and by commission) are listed in Table 8-5.

8.5 Discussion

Experience with INTERNIST-1 has reinforced our impression of medical diagnosis as a complex process. Diagnosis consists of two fundamental activities: the generation of one or more differential diagnoses (each for a separate problem area), and the resolution of individual differential di-

TABLE 8-5 Classification of errors made by INTERNIST-1 during the evaluation

Type of error	*No. of occurrences*
Knowledge-base errors	
Data base incomplete/omission	2
Data base incorrect	2
Lack of anatomic knowledge	1
Failure to represent degree of severity	2
Computer-program faults	
Lack of temporal reasoning	3
Failure of scoring algorithm	3
Failure to seek global overview	1
Improper attribution of finding to a concluded diagnosis	6

agnoses. The surprising ability of the program to make multiple and complex diagnoses in the broad field of internal medicine emphasizes the power of its underlying heuristic methods.

Several important shortcomings of the INTERNIST-1 approach to diagnosis merit further investigation. Feinstein (1977b) has emphasized the importance of explanation as part of diagnostic reasoning. INTERNIST-1's greatest failing during the evaluation (occurring in 6 instances) was its inability to attribute findings to their proper causes. Because of the *ad hoc,* serial nature of INTERNIST-1's formation of problem areas, the program cannot synthesize a general overview in complicated multisystem problems. The structure of the knowledge base, especially the form of the disease profiles, limits the program's ability to reason anatomically or temporally. The program cannot recognize subcomponents of an illness, such as specific organ-system involvements or the degree of severity of pathologic processes.

A diagnostic program must be able to recognize the appropriate cause or causes of observed findings in a patient. A justification for each diagnosis must be developed on a pathophysiologic or causal framework that is consistent with established medical knowledge. To its detriment, INTERNIST-1's handling of explanation is shallow. When the program concludes a diagnosis, that diagnosis is allowed to explain any observed manifestations that are listed on its disease profile. Once explained, a manifestation is no longer used to evoke new disease hypotheses or to participate in the scoring process. This situation is compounded by the inadequate representation of causality in the INTERNIST-1 knowledge base. Disease profiles contain, in an undifferentiated manner, factors predisposing to the

illness as well as findings that result from the disease process itself. An example of this problem occurred in analysis of Case 17-1969, when INTERNIST-1 allowed hepatic encephalopathy to explain the finding of hypokalemia. The program should have recognized hypokalemia as a predisposing factor for hepatic encephalopathy and initiated a search for an independent cause of the finding. At present, the limitations of the knowledge base prohibit such activity.

What is required is a restructuring of the knowledge base to include intermediate-level pathophysiologic states and the segregation of predisposing factors from findings actually caused by a disease. Diseases should be profiled in terms of their intermediate states, rather than as exhaustive lists of manifestations. If the program had such a feature, the presence or absence of each state would be independently determined, and a disease would be allowed to explain a finding only when the state causing the finding was confirmed.

A related problem not handled well by INTERNIST-1 is the interdependency of manifestations. For example, persons with elevated conjugated bilirubin levels in their blood usually have bilirubinuria. At present, the evoking strengths of each finding count redundantly toward any diagnosis that can explain them. This phenomenon causes INTERNIST-1 to favor disproportionately the most common explanation for a set of findings. A solution would be the creation of an intermediate-level state, "abnormal bilirubin metabolism and transport," which would explain both conjugated hyperbilirubinemia and bilirubinuria. Appropriate weight for the intermediate state (rather than for the interdependent manifestations) could be given to any diseases that cause it. Thus creation of a causal network of pathophysiologic states, interposed between observable manifestations and final diagnoses, would allow a diagnostic program to attribute findings to causes accurately and would help to diminish the influence of interdependent manifestations of disease.

INTERNIST-1 constructs differential diagnoses in an *ad hoc* manner, using a scoring algorithm to define the topmost (best) diagnosis and another program, the partitioner, to define reasonable competitors for the topmost diagnosis. By formulating and focusing attention on only one problem domain at any given time, the program is able to disregard "red herrings" and to set aside—temporarily—findings caused by disease processes falling outside the selected problem domain. By creating and processing problem domains serially, the program is able to make multiple diagnoses. But INTERNIST-1 cannot formulate a broad perspective in complicated multisystem patient problems. It is constrained to working with tunnel vision, discriminating among diagnoses within each problem area, unable to look at several problem areas simultaneously. Only after a specific diagnosis is concluded can INTERNIST-1 use the links in its data base to give bonus weight to interrelated diagnoses in separate problem domains. New programming approaches to complex reasoning processes have been developed (Pople, 1982) to enable CADUCEUS, the successor

to INTERNIST-1, to synthesize a broad overview incorporating causal relationships into an approach to a patient's problems.

INTERNIST-1 is unable to reason anatomically or temporally. The program could not differentiate gastric compression due to pancreatic mass effect from that due to hepatic mass effect in Case 23-1969, and as a result it erroneously concluded that the patient had a hepatoma rather than pancreatitis. Nor can INTERNIST-1 recognize the degree of severity of a finding or process in all instances. Two of INTERNIST-1's failures during the evaluation resulted from its inadequate recognition of the degree of severity of an individual manifestation (a decreased blood potassium level) and of an organ-system involvement by a pathologic process (disseminated vasculitis). Reorganization of the data base to allow representation of these concepts is also being undertaken.

INTERNIST-1 is only one of many computer-based tools with the purpose of extending the capabilities of the physician. Such programs can broaden the clinician's scope and awareness of data for the diagnosis and treatment of illness. For the present, INTERNIST-1 remains a research tool. After refinement of the knowledge base and diagnostic programs, a prospective clinical trial will be required to compare the program's behavior with that of clinicians in terms of diagnostic accuracy, cost effectiveness, and danger to the patient.

8.6 A Sample Case Analysis

The transcript of an INTERNIST-1 case analysis given in Figure 8-2 illustrates the operation of the diagnostic programs. The case was taken from a CPC published in the *New England Journal of Medicine* in 1969 (Castleman, 1969). The laboratory values are reported as measured in 1969. The bracketed paragraphs labeled "Comment" have been interpolated for clarification; they are not part of the actual consultation. Places where the transcript has been abridged are indicated by ellipses.

INTERNIST-1 consultation SUMEX-AIM Version
15-May-81 07:31:39
ENTER CLASS NAME: NEJM-CASE-30-1969-ADMISSION-1

[Comment: Here the user enters the initial positive findings (present in the patient) and negative findings (absent). The specialized INTERNIST-1 vocabulary of some 3550 manifestations must be used in describing the case. The plus (+) prompt precedes each positive finding entered by the user. Because INTERNIST-1 has no mechanism for the representation of time, all findings have been collapsed into a single list, independently of their order of appearance in the patient.]

FIGURE 8-2 Transcript of an INTERNIST-1 case analysis.

INITIAL POSITIVE MANIFESTATIONS:
+AGE GTR THAN 55
+ARTHRITIS HX
+DEPRESSION HX
+SEX FEMALE
+THYROIDECTOMY HX
+ULCER PEPTIC HX
+URINE DARK HX
+WEIGHT INCREASE RECENT HX
+ANOREXIA
+CHEST PAIN LATERAL EXACERBATION WITH BREATHING
+CHEST PAIN LATERAL SHARP
+DYSPNEA ABRUPT ONSET
+ABDOMEN DISTENTION
+ABDOMEN FLUID WAVE
+ASTERIXIS
+FECES LIGHT COLORED
+JAUNDICE
+JOINT(S) PERIARTICULAR THICKENING
+JOINT(S) RANGE OF MOTION DECREASED
+LIVER ENLARGED MODERATE
+PLEURAL FRICTION RUB
+PULSE PRESSURE INCREASED
+SKIN PALMAR ERYTHEMA
+SKIN SPIDER ANGIOMATA
+SPLENOMEGALY SLIGHT
+TACHYCARDIA
+TACHYPNEA
+THYROID ENLARGED ASYMMETRICAL
+ALKALINE PHOSPHATASE BLOOD GTR THAN 2 TIMES NORMAL
+BILIRUBIN BLOOD CONJUGATED INCREASED
+BILIRUBIN URINE PRESENT
+FECES GUAIAC TEST POSITIVE
+GLUCOSE BLOOD 130 TO 300
+SGOT 120 TO 400
+AMMONIA BLOOD INCREASED
+ASCITIC FLUID PROTEIN 3 GRAM(S) PERCENT OR LESS
+ASCITIC FLUID WBC 100 TO 500
+ELECTROPHORESIS SERUM ALBUMIN DECREASED
+ELECTROPHORESIS SERUM BETA GLOBULIN INCREASED
+ELECTROPHORESIS SERUM GAMMA GLOBULIN INCREASED
+GLUCOSE TOLERANCE DECREASED
+IMMUNOELECTROPHORESIS SERUM IGA INCREASED
+IMMUNOELECTROPHORESIS SERUM IGM INCREASED
+IMMUNOELECTROPHORESIS SERUM IGG INCREASED
+LDH BLOOD INCREASED
+LE TEST POSITIVE
+LIVER RADIOISOTOPE SCAN IRREGULAR UPTAKE
+PROTHROMBIN TIME INCREASED
+RHEUMATOID FACTOR POSITIVE
+STOMACH BARIUM MEAL HIATAL HERNIA
+*DENY*

[Comment: The keyword *DENY* indicates to INTERNIST-1 that the user now wants to enter negative findings. The minus (-) prompt precedes each negative finding entered.]

FIGURE 8-2 continued

INITIAL NEGATIVE FINDINGS:
-ALCOHOLISM CHRONIC HX
-DIARRHEA CHRONIC
-FECES BLACK TARRY
-FEVER
-PRESSURE VENOUS INCREASED ON INSPECTION
-HEMATOCRIT BLOOD LESS THAN 35
-UREA NITROGEN BLOOD 30 TO 59
-URIC ACID BLOOD INCREASED
-ASCITIC FLUID AMYLASE INCREASED
-ASCITIC FLUID CYTOLOGY POSITIVE
-ASCITIC FLUID LDH GTR THAN 500
-ESOPHAGUS BARIUM MEAL VARICES
-STOMACH BARIUM MEAL ULCER CRATER
-T3 RESIN UPTAKE INCREASED
-T4 TOTAL BLOOD INCREASED
-GO

[Comment: The user enters *GO* to indicate that all relevant positive and negative findings have been entered. The INTERNIST-1 consultant programs guide the user through the rest of the diagnostic work-up by asking questions. For clarity, all reponses typed by the user from this point will be in italics.]

DISREGARDING: DEPRESSION HX, WEIGHT INCREASE RECENT HX, CHEST PAIN LATERAL EXACERBATION WITH BREATHING, CHEST PAIN LATERAL SHARP, DYSPNEA ABRUPT ONSET, ABDOMEN FLUID WAVE, ASTERIXIS, JOINT(S) PERIARTICULAR THICKENING, PLEURAL FRICTION RUB, THYROID ENLARGED ASYMMETRICAL, FECES GUAIAC TEST POSITIVE, GLUCOSE BLOOD 130 TO 300, AMMONIA BLOOD INCREASED, ASCITIC FLUID WBC 100 TO 500, GLUCOSE TOLERANCE DECREASED

CONSIDERING: AGE GTR THAN 55, SEX FEMALE, URINE DARK, ANOREXIA, FECES LIGHT COLORED, JAUNDICE, LIVER ENLARGED MODERATE, SKIN PALMAR ERYTHEMA, SKIN SPIDER ANGIOMATA, SPLENOMEGALY SLIGHT, ALKALINE PHOSPHATASE BLOOD GTR THAN 2 TIMES NORMAL, BILIRUBIN BLOOD CONJUGATED INCREASED, BILIRUBIN URINE PRESENT, SGOT 120 TO 400, ELECTROPHORESIS SERUM ALBUMIN DECREASED, ELECTROPHORESIS SERUM GAMMA GLOBULIN INCREASED, IMMUNOELECTROPHORESIS SERUM IGA INCREASED, IMMUNOELECTROPHORESIS SERUM IGG INCREASED, IMMUNOELECTROPHORESIS SERUM IGM INCREASED, LDH BLOOD INCREASED, LE TEST POSITIVE, LIVER RADIOISOTOPE SCAN IRREGULAR UPTAKE, PROTHROMBIN TIME INCREASED, RHEUMATOID FACTOR POSITIVE.

DISCRIMINATE: HEPATITIS CHRONIC ACTIVE, BILIARY CIRRHOSIS PRIMARY

[Comment: At this point, INTERNIST-1 has constructed a master differential-diagnosis list, ordered its members with a scoring algorithm, and then focused attention on the most promising problem area, which appropriately contains two liver diseases. The DISREGARDING list consists of all findings that are inconsistent with the topmost (first-listed) diagnosis; the CONSIDERING list includes all findings explained by the topmost diagnosis (i.e., chronic active hepatitis). The number of plausible contenders in the problem area determines the strategy for questioning. Questions are asked to discriminate between the two diagnoses.]

Please Enter Findings of LIPID(S) BLOOD
GO
CHOLESTEROL BLOOD DECREASED?
N/A

[Comment: INTERNIST-1 asks questions in two forms. A question of the style "Please Enter Findings of . . . " asks about a generic class of findings, such as serum lipids or chest x-ray films. The user enters specific (positive or negative) findings and when finished types *GO*. If there are no positive or negative findings to enter, typing *GO* will cause the programs to ask the original single item that prompted the generic question.

FIGURE 8-2 continued

For example, an elevated cholesterol level would favor primary biliary cirrhosis over chronic active hepatitis, whereas the converse situation would apply if the cholesterol were decreased. Thus, INTERNIST-1 asked a generic question about serum liquids. The second form of questioning is direct. INTERNIST-1 simply prints a manifestation name followed by a question mark. The user can then respond by typing *YES* (present), *NO* (absent), or *N/A* (not available—either unknown or not performed). The published CPC did not mention the serum cholesterol value, so the user entered *N/A*.]

Please Enter Findings of INSPECTION AND PALPITATION SKIN
GO
SKIN XANTHOMATA?
NO
Please Enter Findings of BLOOD COUNT AND SMEAR
GO
RBC TARGET(S)?
NO

[Comment: INTERNIST-1 uses information about the type of each manifestation—history, symptoms, signs, and laboratory tests, both routine and two gradations of more expensive or invasive laboratory procedures. The types of manifestations determine which questions are asked. Although the greatest discriminating power for resolving a differential diagnosis often resides in an invasive test, such as a liver biopsy, INTERNIST-1 has been programmed to exhaust the useful history and physical-examination findings before escalating its questioning to more expensive or dangerous laboratory tests.]

Please Enter Findings of KIDNEY FUNCTION TEST(S)
GO
UREA NITROGEN BLOOD 60 TO 100?
NO
UREA NITROGEN BLOOD GTR THAN 100?
NO
Please Enter Findings of LIVER FUNCTION TEST(S)
GO
UROBILINOGEN URINE ABSENT?
N/A

[Comment: INTERNIST-1 did not seek to investigate renal function itself in the passage above. It was programmed to recognize that the urinary urobilinogen value may be unreliable in the presence of marked azotemia, so it first asked about the blood urea nitrogen.]

CONCLUDE: HEPATITIS CHRONIC ACTIVE

[Comment: It is likely that INTERNIST-1 had nearly made the diagnosis of chronic active hepatitis before it asked any questions. The negative answers favored chronic active hepatitis, and therefore INTERNIST-1 concluded the diagnosis.]

DISREGARDING: DEPRESSION HX, CHEST PAIN LATERAL EXACERBATION WITH BREATHING, CHEST PAIN LATERAL SHARP, DYSPNEA ABRUPT ONSET, ASTERIXIS, JOINT(S) PERIARTICULAR THICKENING, PLEURAL FRICTION RUB, THYROID ENLARGED ASYMMETRICAL, FECES GUAIAC TEST POSITIVE, GLUCOSE BLOOD 130 TO 300, AMMONIA BLOOD INCREASED, GLUCOSE TOLERANCE DECREASED

CONSIDERING: WEIGHT INCREASE RECENT HX, ABDOMEN DISTENTION, ABDOMEN FLUID WAVE, ASCITIC FLUID OBTAINED BY PARACENTESIS, ASCITIC FLUID WBC 100 TO 500, ASCITIC FLUID PROTEIN 3 GRAM(S) PERCENT OR LESS ...
CONCLUDE: TRANSUDATIVE ASCITES
DISREGARDING: ...
CONSIDERING: ASTERIXIS, PULSE PRESSURE INCREASED, TACHYCARDIA, AMMONIA BLOOD INCREASED
PURSUING: HEPATIC ENCEPHALOPATHY

FIGURE 8-2 continued

[Comment: The links in the INTERNIST-1 knowledge base between chronic active hepatitis and hepatic encephalopathy have resulted in a bonus weight's being given to hepatic encephalopathy here; previously, links had promoted the consideration of transudative ascites, since it can also be caused by chronic active hepatitis.]

CSF FLUID OBTAINED?
N/A

[Comment: Here INTERNIST-1 was about to ask about the glutamine level in the cerebrospinal fluid. Since no lumbar puncture was performed, the result is not available.]

CONCLUDE: HEPATIC ENCEPHALOPATHY

[Comment: In the above situation, there were no diagnostically helpful tests remaining for INTERNIST-1 to ask. INTERNIST-1 has been programmed to relax its criteria for concluding a diagnosis when all useful lines of questioning have been blocked. Since INTERNIST-1 had been close to making the diagnosis of hepatic encephalopathy, the program now concludes the diagnosis. The case analysis was intentionally stopped at this point, because all relevant major diagnoses had been covered. Without such intervention, INTERNIST-1 would try to explain any remaining important findings, such as the arthritis and pleurisy.]

FIGURE 8-2 continued

ACKNOWLEDGMENTS

We are indebted to Craig Dean, Charles Oleson, and Kenneth Quayle for their contributions in writing the INTERNIST-1 computer programs; to Zachary Moraitis for his assistance in the conceptual design of the project and in the development of the knowledge base; to a large number of medical students and several fellows in computer medicine for their assistance in the development of the INTERNIST-1 knowledge base; and to the staff of the SUMEX-AIM computing facility of the National Institutes of Health for providing expert assistance and a friendly environment for programming.

The INTERNIST-1/CADUCEUS project is supported by grants from the Division of Research Resources (R24 RR 01101) and the National Library of Medicine (R01 LM 03710 and R23 LM 035789), National Institutes of Health. The SUMEX computing project is supported by a grant (RR 00785) from the Biotechnology Resources Program, National Institutes of Health.

9

Categorical and Probabilistic Reasoning in Medical Diagnosis

Peter Szolovits and Stephen G. Pauker

In the mid-1970s, when Gorry left M.I.T. to go to Baylor College of Medicine, Peter Szolovits took over as head of the Clinical Decision-Making Group at Project MAC (now known as the Laboratory for Computer Science). He renewed ties with the collaborators at Tufts University with whom Gorry had previously worked (Pauker, Schwartz, and Kassirer). The following chapter is an early result of those developing ties. It was written for a special issue of Artificial Intelligence *that dealt solely with applications of AI in biomedicine (Sridharan, 1978). In the article Szolovits and Pauker review the lessons of the major four AIM programs of the early 1970s.*

The review begins by noting that medical decision making can be viewed along a spectrum, with categorical (or deterministic) reasoning at one extreme and probabilistic (or evidential) reasoning at the other. The authors discuss classical flow charts as the prototype of categorical reasoning and decision analysis as the prototype of probabilistic reasoning. Within that context they compare MYCIN, PIP, CASNET, and INTERNIST—the four systems described in Chapters 5 through 8. They note that, although all four systems can exhibit impressive expertlike behavior, none of them is capable of truly expert reasoning. They argue that a program that can demonstrate expertise in the area of medical consultation will have to use a judicious combination of categorical and probabilistic reasoning—the former to establish a sufficiently narrow context and the latter to make comparisons among hypotheses and eventually to recommend therapy. We include the paper here because it nicely summarizes and integrates the

From *Artificial Intelligence*, 11: 115–144 (1978).

discussions of the systems in the four preceding chapters. By citing the limitations of the early systems, this article helped define and clarify some of the research issues that evolved later in the decade and are discussed in subsequent chapters.

9.1 Introduction

How do practicing physicians make clinical decisions? What techniques can we use in the computer to produce programs that exhibit medical expertise? Our interest in these questions is motivated by our desire:

1. to provide (by computer) expert medical consultation to general practitioners or paramedical personnel in communities where such consultation is normally unavailable;
2. to come to understand the reasoning processes of expert doctors so that we may improve the teaching of their skills to medical students; and
3. to advance the techniques of artificial intelligence, especially as applied to medicine (AIM), to support our other goals.

In other publications, we have described research by our group on programs to take the history of the present illness of a patient with renal disease (Pauker and Gorry, 1976; Szolovits and Pauker, 1976) and to advise the physician in the administration of the drug digitalis to patients with heart disease (Gorry et al., 1978; Silverman, 1975; Swartout, 1977). Here, we would like to review the reasoning mechanisms[1] used by our own programs, by other AI programs with medical applications, and, by inference, by physicians.

9.2 Categorical and Probabilistic Decisions

Most decisions made in medical practice are straightforward. Whether the physician is taking a history of a patient's illness, performing a routine physical examination, or ordering a standard battery of laboratory tests, he or she makes few real decisions. To a large extent his or her expertise

[1]In this discussion, we take *reasoning* to be synonymous with *decision making*. Although the former is a broader term, we are specifically concerned with that aspect of reasoning that yields medical decisions. An earlier review of work in this area was made by Pople et al. (1975).

consists of mastery of the appropriate set of *routines* with which he or she responds to typical clinical situations.

This view is corroborated, in part, by the observed differences between the diagnostic approach of a medical student or newly minted doctor and that of a practicing expert. The novice struggles "from first principles" initially to propose plausible theories and then to rule out unlikely ones, whereas the expert simply recognizes the situation and knows the appropriate response. We might say that the expert's knowledge is *compiled* (Rubin, 1975; Sussman, 1973). Similar differences have even been noted among expert consultants in different specialties when they are presented the same case, and even between the performance of the same consultant on cases within compared to cases outside his or her specialty. The expert doctor dealing with a case within his or her own specialty approaches the case parsimoniously; the expert less familiar with the case resorts to the more general diagnostic style associated with the nonexpert (Miller, 1975).

An important characteristic of expert decision making, then, is the use of an appropriate set of routines or rules that apply to the great majority of clinical situations. We shall identify this as *categorical reasoning*.[2] A categorical medical judgment is one made without significant reservations: if the patient's serum sodium is less than 110 mEq./l., administer sodium supplements; if the patient complains of pain on urination, obtain a urine culture and consider the possibility of a urinary tract infection. These rules, as applied by the physician, are not absolutely deterministic. Although their selection and use do not involve deep reasoning, the doctor may withhold his or her full commitment from conclusions reached by even such categorical rules. The doctor thereby establishes the flexibility to modify his or her conclusions and rethink the problem if later difficulties arise.

A categorical decision typically depends on a relatively few facts; its appropriateness is easy to judge, and its result is unambiguous. A categorical decision is simple to make, and the rule that forms its basis is usually simple to describe (although its validity may be complicated to justify). Physicians most often work with categorical decisions, and, to whatever extent possible, computer experts should do the same.

Unfortunately, not every decision can be categorical. No simple rule exists for deciding whether to perform a bone marrow biopsy or when to discharge a patient from the cardiac intensive care unit. Those decisions must be made by carefully weighing all the evidence. Although we know that doctors do so, we do not understand just how they weigh the evidence that favors and that opposes various hypotheses or courses of action; this is an important unsolved problem for both AI and cognitive psychology (Newell and Simon, 1972; Tversky and Kahneman, 1974).

---

[2]Webster's defines *categorical* as "unqualified; unconditional; absolute; positive; direct; explicit; . . . "

A number of formal schemes for the weighing of evidence are used, and we shall concentrate on one of them, the *probabilistic,* to contrast with the categorical mode of reasoning[3] discussed above. We do not believe or suggest that formal probabilistic schemes are naturally used in decision making by physicians untrained in the use of such schemes. Indeed, there is convincing evidence that people are very poor at probabilistic reasoning (Tversky and Kahneman, 1974). Yet we believe that, with appropriate limitations as discussed below, probabilistic reasoning can be an appropriate component of a computerized medical decision-making system, especially for the difficult decisions for which categorical reasoning is inappropriate.[4]

In this paper we examine prototypical categorical and probabilistic reasoning systems, their limitations, and their successful applications, and then describe and analyze the reasoning mechanisms of some current AIM programs in terms of these schemes. We conclude with some comments and speculations on the requirements for reasoning mechanisms in future AIM programs.

9.2.1 Purely Categorical Decision Making—The Flow Chart

Categorical reasoning is exemplified by the simplest *flow chart* programs for guiding frequent decisions based on a well-accepted rationale. The flow chart is a finite state acceptor in which every nonterminal node asks a question whose possible answers are the labels of the arcs leaving that node. The machine has a unique initial state corresponding to initial contact with the user and a number of possible terminal states, each labeled by an outcome—a diagnosis, patient referral, selected therapy—relevant in its domain of application.[5] Every answer to every question is decisive; the formalism is simple and attractive.

---

[3]Other potentially appropriate schemes include the theory of *belief functions* (Shafer, 1976) and the application of *fuzzy set theory* (Gaines, 1976; Zadeh, 1965). All share the characteristic that arithmetic computations are performed to combine separate beliefs or implications to determine their joint effect. We are not convinced of the uniform superiority of any of these formalisms. Because we are most familiar with the probabilistic scheme, we have chosen to examine it in detail.

[4]Although our approach to the construction of expert medical systems has been, in general, to follow the way we think expert physicians reason, the known deficiencies in people's abilities to make correct probabilistic inferences suggest that this is one area in which the computer consultant could provide a truly new service to medicine. However, it is not universally accepted in medicine that probabilistic techniques are a valid way to make clinical decisions (Feinstein, 1977b).

[5]In some flow chart schemes, the structure of the acceptor is a tree. In that case, every terminal node can be reached only by a unique path. In other flow charts, the acceptor is augmented to retain information collected during questioning (e.g., in history-taking systems). Even in those systems, it is uncommon for a piece of information to be used to select a branch in the flow chart in any place except where it is determined. Thus that augmentation does not provide the program with any additional state information.

Perhaps the most successful use of categorical decision-making programs is in patient-referral triage.[6] Nurse-practitioners using standardized information-gathering and decision-making protocols can effectively handle routine orders for noninvasive laboratory tests and the scheduling of emergency or routine visits with a doctor. Such a system is now used in the walk-in clinic at the Beth Israel Hospital in Boston (Perlman et al., 1974), actually employing pen and printed forms rather than computer-generated displays and keyboard input.

Although every decision in a flow chart is categorical, the development of that flow chart may have been based on extensive probabilistic computations. Optimal test selection studies (Peters, 1976) and treat versus no-treat decision models (Pauker and Kassirer, 1975) are examples of probabilistic means of generating categorical decision models.

Whereas patient referral deals with a broad problem domain that may require only shallow knowledge, the problem of providing the physician with advice about the administration of digitalis requires a great deal of knowledge about a narrow medical domain. That domain is, in fact, sufficiently well understood at the clinical (although not the physiological) level that a reasonably straightforward program has been implemented (Silverman, 1975) that gathers relevant clinical parameters about the patient, projects digitalis absorption and excretion rates, adjusts for patient sensitivities, and monitors the patient's clinical condition for signs of therapeutic benefit or toxic effect. Although the numerical models used by the program are complex, its data-gathering strategy and its heuristic techniques for adjusting dosages are simple enough that most parts of the program can be explained to the user by simply translating the computer's routines into English (Swartout, 1977). This program relies largely on categorical reasoning.

Why are categorical decisions not sufficient for all of medicine? Because the world is too complex! Although many decisions may be made straightforwardly, many others are too difficult to be prescribed in any simple manner. When many factors may enter into a decision, when those factors may themselves be uncertain, when some factors may become unimportant depending on other factors, and when there is a significant cost associated with gathering information that may not actually be required for the decision, then the rigidity of the flow chart makes it an inappropriate decision-making instrument.[7]

---

[6]Triage is "the medical screening of patients to determine their priority for treatment; the separation of a large number of casualties, in military or civilian disaster medical care, into three groups: those who cannot be expected to survive even with treatment; those who will recover without treatment; and the priority group of those who need treatment in order to survive" (Stedman, 1961).

[7]Of course, one could, in principle, anticipate every complication and degree of uncertainty to every answer in the flow chart. If medical diagnosis is a finite process, then a gigantic flow chart could capture it all. This is, however, the equivalent of playing chess by having precomputed every possible game; it is probably equally untenable. It suffers similarly from losing all of the parsimony of the underlying model that the physician must have, from which the giant flow chart would be produced.

9.2.2 Purely Probabilistic Decision Making—Bayes' Rule and Decision Analysis

In a typical probabilistic decision problem,[8] we are to find the true state of the world, H_T, which is one of a fixed, finite set of exhaustive and mutually exclusive hypotheses, $H_1, H_2, \ldots, H_n$. We start with an initial estimate of the probability that each H_i is the true state. We then perform a series of tests on the world and use the results to revise the probability of each hypothesis. Formally, we have a probability distribution, P, that assigns to each H_i a prior probability, P_{H_i}. The available tests are $T_1, T_2, \ldots, T_m$, and for each test, T_i, we may obtain one of the results, $R_{i,1}, R_{i,2}, \ldots, R_{i,r_i}$.

Consider the case where we perform a series of the tests. We define the *test history* of the patient after the ith test to be the list of <test, result> pairs performed so far:

$$Q_i = (<T_{\text{sel}(1)}, R_{\text{sel}(1),k_{\text{sel}(1)}}>, \ldots, <T_{\text{sel}(i)}, R_{\text{sel}(i),k_{\text{sel}(i)}}>) \tag{1}$$

where sel is the test selection function.

If for every H_j and for every possible testing sequence, Q_i, we can assess how likely we would be to observe Q_i in the situation where H_j were known to be the true state, then we may apply Bayes' Rule to estimate, after any possible test history, the likelihood that H_j is H_T. In other words, if we know the conditional probability of any test history given any hypothesis, $P_{Q_i|H_j}$, for each j and Q_i, then we can apply Bayes' Rule to compute the posterior probability distribution over H:

$$P_{H_j|Q_i} = \frac{P_{Q_i|H_j} \cdot P_{H_j}}{\sum_{k=1}^{n} P_{Q_i|H_k} \cdot P_{H_k}} \tag{2}$$

A straightforward application of the above methodology would be to perform every test for every patient in a fixed order, obtaining Q_n, and then to use formula (2) to compute the posterior probabilities. Less naive applications of the methodology involve sequential diagnosis, in which the order of tests selected depends on previous results and in which diagnosis may terminate before all tests are performed. In sequential diagnosis, the next test to be performed may be selected by an expected information-maximizing function (Gorry et al., 1973) or a classical decision analysis that maximizes expected utility. The diagnostic process may terminate when the likelihood of the leading hypothesis exceeds some threshold[9] or when

[8]Here we follow Gorry (1967). This is the Bayesian approach to probabilistic decision problems.

[9]Sometimes, it is the ratio of the likelihood of the leading hypothesis to that of the next hypothesis that must exceed a threshold.

the expected cost of obtaining further information exceeds the expected cost of misdiagnosis due to missing those further data. Each of these techniques has been applied in diagnosis.

The failure of the pure probabilistic decision-making schemes lies in their voracious demand for data. Consider the size of the data base that would be needed for a direct implementation of the Bayesian methodology described above. In performing i of m possible tests, we can choose ${}_mP_i$ $(= m!/(m - i)!)$ possible test sequences. If every test has r possible results, then there will be $r^i \cdot {}_mP_i$ possible test histories after i tests. If we want to know the probability distribution over the H_i after each test (to help to select the next one), then we need to sum over test histories of every length and to multiply by the number of hypotheses, n, to get a total of

$$n \cdot \sum_{i=1}^{m} r^i \cdot {}_mP_i \tag{3}$$

conditional probabilities. For even a relatively small problem—e.g., $n = 10$ hypotheses, $m = 5$ binary tests $(r = 2)$—the analysis requires 63,300 conditional probabilities.[10]

Although the methodology described above is a complete view of medical diagnosis, it is certainly not an efficient one. To improve the scheme's efficiency, researchers typically make a series of assumptions about the problem domain that permit the use of a more parsimonious version of this decision method. First, it is usually assumed that two tests will yield the same results if we interchange the order in which they are performed.[11] That assumption reduces the number of conditional probabilities needed to

$$n \cdot \sum_{i=1}^{m} r^i \cdot {}_mC_i = n \cdot ((1 + r)^m - 1) \tag{4}$$

(2,420 in our example), which is still unwieldy.

A second assumption often made is that test results are conditionally independent—i.e., given that some hypothesis is the true state of the world, the probability of observing result $R_{i,k}$ for test T_i does not depend on what results have been obtained for any other test. This assumption allows all

[10]We are actually underestimating the amount of data required for such an analysis. In addition to the conditional probabilities, we also need other values to construct an optimal test-selection function. For example, we might use the costs of performing each test (possibly different after each different test history) and the costs and benefits of each possible treatment.

[11]Although this seems very reasonable, it is not strictly true. The effect of one test may be to interfere with a later one. For example, the upper GI series can interfere with interpretation of a subsequent intravenous pyelogram (IVP). The situation is even more complex since the effect of the former test on the latter often depends on the time that elapses between them. Even so, the assumption is so useful that it is worth making.

information from previous tests to be summarized in the revised probability distribution after the ith test, and the data requirements are reduced to approximately $m \cdot n \cdot r$ conditional probabilities (100 in our example), which is reasonable for some applications (Flehinger and Engle, 1975).

Unfortunately, three serious problems arise with the above scheme and its simplifications. The assumption of conditional independence is usually false, and the basic premises of the applicability of Bayes' Rule, that the set of hypotheses is exhaustive and mutually exclusive, are often violated. These may all lead to diagnostic conclusions that are wrong.

In a small study of the diagnosis of left-sided valvular heart disease, we have found that assuming conditional independence between observations of systolic and diastolic heart murmurs leads (not surprisingly) to erroneously reversed conclusions from those obtained by a proper analysis. To the extent that anatomical and physiological mechanisms tie together many of the observations that we can make of the patient's condition and to the extent that our probabilistic models are incapable of capturing those ties, simplifications in the computational model will lead to errors of diagnosis.

A similar error is introduced when conditional probabilities involving the negation of hypotheses are used. $P_{R|\sim H}$, being the probability of a test result R given that hypothesis H is *not* the true state of the world, cannot be assessed without knowing the actual probability distribution over the other hypotheses (unless, of course, there is only one other hypothesis). In fact, in our formalism,

$$P_{R|\sim H_i} = \sum_{j \neq i} P_{H_j} \cdot P_{R|H_j} / (1 - P_{H_i}) \tag{5}$$

which obviously depends on the probability distribution over the hypotheses. Even if we make the usual assumption of conditional independence, the practice of considering $P_{R|\sim H_i}$ to be a constant is unjustified and leads to further errors. Formalisms that employ a constant likelihood ratio implicitly commit this error, often without recognizing it (Duda et al., 1976; Flehinger and Engle, 1975). The likelihood ratio is defined as $P_{R|H_i} / P_{R|\sim H_i}$. Assuming conditional independence of the test results guarantees only that the numerator is constant, while, in general, the denominator will vary according to formula (5) as new results alter the probability distribution over the hypotheses. Using a constant likelihood ratio evaluates the current result in the context of the *a priori* probabilities, wrongly ignoring the impact of all of the evidence gathered up to that point.

A far more serious objection to the use of pure probabilistic decision making is that in most clinical situations the hypotheses under consideration are neither exhaustive nor mutually exclusive. If we perform a Bayesian calculation in the absence of exhaustiveness within the set of hypotheses, we will arrive at improperly normalized posterior probabilities. Their use in assessing the relative likelihoods of our possible hypotheses

is appropriate, but we may not rest absolute prognostic judgments or compute expected values on the basis of such calculations.

The absence of mutual exclusivity is a more serious flaw in this methodology. Doctors find it useful to describe the clinical situation of a patient in terms of abstractions of disorders. When a patient is described as having acute poststreptococcal glomerulonephritis (AGN), for example, no one means that this patient exhibits every symptom of the disease as described in a textbook or that every component of the disease and its typical accompaniments is present. Having accepted such a description of the patient with AGN, diagnosis may then turn to consideration of whether such common (but not necessary) complications as acute renal failure and hypertension are present as well. Mapping this process into the view imposed by classical probabilistic methods requires the creation of independent hypotheses for every possible combination of diseases. That technique leads to a combinatorial explosion in the data collection requirements of the system and at the same time destroys the underlying view the practicing physician takes toward the patient.

Because of the distortions that the pure probabilistic scheme imposes on the problem and because of the enormous data requirements it implies, it tends to be used successfully only in small, well-constrained problem domains.

9.3 Reasoning in Current AIM Programs

Medical judgment, by the physician and by computer programs, must be based on both categorical and probabilistic reasoning. The focus of research in applying artificial intelligence techniques to medicine is to find appropriate ways to combine these forms of reasoning to create competent programs that exhibit medical expertise. In this section, we will outline in brief the central reasoning strategy of four major AIM programs and compare their methods to the two "pure cases" presented above.

9.3.1 The Present Illness Program

Perhaps the best way to explain the reasoning of our program is to describe the data that are available to it. The Present Illness Program (PIP) (Szolovits and Pauker, 1976) (also see Chapter 6) can deal with a large set of possible *findings* and a separate set of *hypotheses*. Findings are facts about the patient that are reported to the program by its user. Hypotheses represent the program's conjecture that the patient is suffering from a disease or manifesting a clinical or physiological state. Associated with hypotheses are sets of prototypical findings that can either support or refute the hypothesis.

Relation to Findings

TRIGGERS	<findings>
FINDINGS	<findings>

Logical Decision Criteria

IS-SUFFICIENT	<findings>
MUST-HAVE	<findings>
MUST-NOT-HAVE	<findings>

Complementary Relation to Other Hypotheses

CAUSED-BY	<hypotheses>
CAUSE-OF	<hypotheses>
COMPLICATED-BY	<hypotheses>
COMPLICATION-OF	<hypotheses>
ASSOCIATED-WITH	<hypotheses>

Competing Relation to Other Hypotheses

DIFFERENTIAL-DIAGNOSIS
(<condition 1> <hypotheses>) . . . (<condition k> <hypotheses>)

Numerical Likelihood Estimator

SCORE
((<condition 1,1><score 1,1>) . . . (<condition 1,n_1 > <score 1,n_1 >))
. . .
((<condition m,1> <score m,1>) . . . <condition m,n_m > <score m,n_m >))

FIGURE 9-1 Structure of a hypothesis frame in PIP.

Findings reported by the user are matched against these prototypical findings and, if a match occurs,[12] PIP's belief in the hypothesis is reevaluated. Figure 9-1 shows the structure of a hypothesis in PIP.

Presentation

Both TRIGGERS and FINDINGS are often associated with the hypothetical disorder. If a reported finding matches one of the triggers of a hypothesis, that hypothesis is immediately *activated.* If it matches a nontrigger

[12]The details of this matching process are not relevant to the questions addressed here and will not be discussed. The prototype finding can express either the presence or absence of a sign, symptom, laboratory test, or historical finding. For example, it is possible to use the *absence* of increased heart muscle mass (which takes months to develop) to argue in favor of acute rather than chronic hypertension. In general, many possible findings may match a prototype finding pattern. Thus, within each frame, only those aspects of a finding that are important to the hypothesis at hand need be mentioned, and any of the category of possible findings thus defined will match successfully.

finding, its relevance to that hypothesis is only noticed if the hypothesis is already under consideration. The logical decision criteria are used by the program to make categorical decisions about the likelihood of the patient's suffering from the currently considered hypothesis. IS-SUFFICIENT covers the case of pathognomonic findings, in which the presence of a single finding is in itself sufficient to confirm the presence of the hypothesized disorder; logical combinations (by NOT, AND, and OR) may also be used to specify more complex criteria. MUST-HAVE and MUST-NOT-HAVE specify necessary conditions, in the absence of which the hypothesis will not be accepted as confirmed.[13]

The *complementary* hypotheses identify other disorders that may be necessary in addition to the hypothesis under consideration to account for the condition of the patient.[14] The relationship may be known as *causal* if the physiology of the disorders is well understood, may be *complicational* if one disorder is a typical complication of the other, or may be *associational* if the two may be related by some known but incompletely understood association. Although all noncomplementary hypotheses are competitors, medical practice specifically identifies those that may often be confused—that is the role of the DIFFERENTIAL-DIAGNOSIS relationships in the frame.

The complementary and competing relations to other hypotheses are used in controlling the activation of hypotheses. In an anthropomorphic analogy, we think of an *active* hypothesis as corresponding to one about which the physician is consciously thinking. Active hypotheses offer the possible explanations for the patient's reported condition and are the basis from which the program reasons to select its next question. *Inactive* hypotheses are all those possible disorders that play no role in the program's current computations; they may be inactive either because no findings have ever suggested their possibility or because they have been considered and rejected by evaluation in light of the available evidence. *Semiactive* hypotheses bridge the gap between active and inactive ones and allow us to represent hypotheses that are not actively under consideration but that may be "in the back of the physician's mind." As mentioned above, if a trigger of any hypothesis is reported, that hypothesis is made active. When a hypothesis is activated, all of its closely related complementary hypotheses are semiactivated. Whereas nontrigger findings of inactive hypotheses do not lead to consideration of those hypotheses, any reported finding of a semiactive hypothesis causes it to be activated (i.e., each of its findings is treated as a trigger). This models the observation that physicians are more likely to pay attention to the minor symptoms of a disease related to the diagnosis that they are already considering than to the minor symptoms

[13]For logical completeness, we could have an IS-SUFFICIENT-NOT-TO-HAVE criterion, which would confirm a hypothesis in the absence of some finding, but this is just not useful.

[14]Note that we use the word *complement* in the sense of completion, not as implying negation or something missing. This is the sense of the word used in Pople (1975).

of an unrelated disorder. Each of the complementary hypotheses identifies another disorder that may be present along with the one under consideration and that is therefore to be semiactivated. The DIFFERENTIAL-DIAGNOSIS relation identifies a set of competing hypotheses that are to be semiactivated if the appropriate condition holds.

We need to assign to every hypothesis some estimate of its likelihood. In PIP, that estimate forms one basis for deciding whether the hypothesis ought to be *confirmed,* if the estimate is sufficiently high, or *inactivated,* if it is sufficiently low. Further, PIP bases its questioning strategy in part on the likelihood of its leading hypothesis. That likelihood is estimated by combining a function that measures the fit of the observed findings to the expectations of the hypothesis with a function that is the ratio of the number of findings that are accounted for by the hypothesis to the total number of reported findings. These two components of the likelihood estimate are called the *matching score* and the *binding score.*

PIP allows us to define clinical and physiological states (not only diseases) as hypotheses. Thus it is not necessary to list every symptom of a disease with that disease hypothesis; commonly co-occurring symptoms can be made symptoms of a clinical state hypothesis, and their relation to the disease derives from the causal relation of the disease to the clinical state. This is an appropriate structure that is consistent with medical practice. It does, however, raise a problem in computing the matching and binding scores for a hypothesis. If a finding is accounted for by a clinical state that is related to a disease, then the binding score of the disease hypothesis should reflect that relation, and its matching score should also reflect that the finding has improved the fit of the facts of the case to the hypothesis. To effect this behavior, PIP uses a *score propagation* scheme, described below. A similar argument can be made to extend score propagation to disease hypotheses as well: if a disease is made more likely by the observation of one of its symptoms, causally related diseases should also be seen as more likely.

The numerical likelihood estimator (see Figure 9-1) is used to compute the *local score* part of the matching score. The local score reflects the degree to which the facts found support the hypothesis directly. It consists of a series of clauses, each of which is evaluated as a LISP COND.[15] The local score of a hypothesis is the sum of the values of the clauses, normalized by the maximum possible total score. Thus it ranges from a maximum of 1 (complete agreement) downward to arbitrarily large negative numbers (complete disagreement).

---

[15]That is, for clause *i*, first <condition *i*,1> is evaluated, and if it is true, the value of clause *i* is <score *i*,1>. If that first condition is false, then each other condition in the clause is evaluated in turn, and the value of the clause is the score for the first true condition. Prototypical finding patterns in the condition that have not yet been asked about—thus, whose truth is not yet known—are treated as false, unless the pattern requests a negative or unknown finding. If none of the conditions is true, the value of the clause is zero.

PIP now computes the matching score by revising the local score to include the effects of propagated information deriving from related hypotheses. Consider the case when PIP is trying to compute the score for the hypothesis, H_i. First we identify all those other hypotheses, H_j, that are possibly complementary to H_i.[16] PIP then computes the MATCHING-SCORE by adding up the contributions of every scoring clause of H_i and each H_j and normalizing by the maximum possible total for this virtual scoring function. The effect here is to mechanically undo the organization imposed by the use of clinical and physiological states, since we could achieve a similar effect by merely listing with each hypothesis the exhaustive set of symptoms to which it might lead. Figure 9-2 shows, as an example, the PIP frame for acute glomerulonephritis.

Discussion

PIP uses both categorical and probabilistic[17] reasoning mechanisms. We shall identify the various forms of reasoning that it undertakes and whether they are accomplished by categorical or probabilistic means. When a finding is reported to PIP, whether as a fact volunteered by the user or in response to the program's questions, it tries to characterize fully the finding in terms of all the descriptors known to apply to that finding. For example, if edema is reported, PIP will try to establish its location, severity, temporal pattern, and whether or not it is symmetrical, painful, and erythematous. Rather specific rules capture some of the physician's common sense: if the question of past proteinuria is raised, PIP can conclude its absence if the patient passed a military physical examination at that time. These inferences are purely categorical.

The main control over PIP's diagnostic behavior resides in the list of active and semiactive hypotheses. Recall that only these hypotheses are "under consideration"—only they are evaluated or used to select the pro-

[16] H_j may be directly linked as a complementary relation to H_i, or it may be linked by a causal path going through some other hypotheses. In the latter case, we insist that the flow of causality along such a linking path be unidirectional, for we do not want, for example, two independent causes of some disease to reinforce each other's likelihood merely by being possible causes of the same disorder. We also compute a LINK-STRENGTH between the hypotheses, which is the product of each LINK-STRENGTH along the component links. Those component link strengths are identified in the data base and reflect the strength of association represented by the links.

[17] As should be clear from the above discussion, we do not think of the score computations as representing a true probability (either objective or subjective). We have sometimes tried to think of our scores as log-transformed probabilities, but the analogy is weak. Rather, we must think of them as an arbitrary numeric mechanism for combining information, somewhat analogous to the static evaluation of a board position in a chess-playing program. It is useful, however, to contrast the scoring computations with a correct probabilistic formulation, because that analogy suggests an explanation for various deficiencies of the scoring scheme (Szolovits, 1976).

```
TRIGGERS      (EDEMA with LOCATION = FACIAL or PERI-ORBITAL,
                   PAINFULNESS = not PAINFUL,
                   SYMMETRY = not ASYMMETRICAL,
                   ERYTHEMA = not ERYTHEMATOUS)
FINDINGS      (COMPLEMENT with RANGE = LOW), (MALAISE), (WEAKNESS),
              (ANOREXIA), (EDEMA with SEVERITY = not MASSIVE),
              (PATIENT with AGE = CHILD or YOUNG, SEX = MALE)
CAUSED-BY      (STREPTOCOCCAL-INFECTION in RECENT-PAST)
CAUSE-OF      SODIUM-RETENTION, ACUTE-HYPERTENSION, NEPHROTIC-SYNDROME,
              GLOMERULITIS
COMPLICATED-BY     ACUTE-RENAL-FAILURE
COMPLICATION-OF      CELLULITIS

DIFFERENTIAL-DIAGNOSIS

   (CHRONIC-HYPERTENSION implies CHRONIC-GLOMERULITIS)
   (EDEMA with RECURRENCE = not FIRST-TIME
       implies NEPHROTIC-SYNDROME, CHRONIC-GLOMERULONEPHRITIS,
       FOCAL-GLOMERULONEPHRITIS)
   (ABDOMINAL-PAIN implies HENOCH-SCHOENLEIN-PURPURA)
   (RASH with PURPURA = PURPURIC implies HENOCH-SCHOENLEIN-PURPURA)
   (RASH with (either LOCATION = MALAR or PHOTOSENSITIVITY = PHOTOSENSITIVE)
       implies SYSTEMIC-LUPUS)
   (JOINT-PAIN implies HENOCH-SCHOENLEIN-PURPURA, SYSTEMIC-LUPUS)

SCORE

(((PATIENT with AGE = CHILD or YOUNG) → 0.8)
 ((PATIENT with AGE = MIDDLE-AGED) → −0.5)
 ((PATIENT with AGE = OLD) → −1.0))
(((COMPLEMENT with RANGE = LOW) → 1.0)
 ((COMPLEMENT with RANGE = NORMAL or MODERATELY-ELEVATED) → −0.7)
 ((COMPLEMENT with RANGE = VERY-HIGH) → −1.0))
(((EDEMA with LOCATION = FACIAL or PERI-ORBITAL, SYMMETRY = not ASYMMETRICAL,
     DAILY-TEMPORAL-PATTERN = WORSE-IN-MORNING, PAINFULNESS = not PAINFUL,
     ERYTHEMA = not ERYTHEMATOUS) → 1.0)
 ((EDEMA with LOCATION = FACIAL or PERI-ORBITAL, SYMMETRY = not ASYMMETRICAL,
     PAINFULNESS = not PAINFUL, ERYTHEMA = not ERYTHEMATOUS) → .5)
 ((EDEMA with SEVERITY = not MASSIVE) → 0.1)
 ((EDEMA with SEVERITY = MASSIVE) → −0.1)
(((PATIENT with SEX = MALE) → 0.3)((PATIENT with SEX = FEMALE) → −0.3))
(((ANOREXIA) → 0.3) ((ANOREXIA absent) → −0.3))
(((WEAKNESS) → 0.3) ((WEAKNESS absent) → −0.3))
```

FIGURE 9-2 The PIP hypothesis frame for acute glomerulonephritis.

gram's further questions. The activation (but not the evaluation) of all hypotheses is purely categorical. A hypothesis can come up for consideration only if one of its prototype findings is matched by a reported finding, if a complementary hypothesis is activated, or if a competing hypothesis is active and a finding matches a condition among its differential diagnosis clauses.

Once a hypothesis is under consideration, both categorical and probabilistic mechanisms exist to decide its merit. In 18 of the 38 fully developed hypothesis frames in the current PIP, we find categorical IS-SUF-

FICIENT rules to establish the presence of the hypothesized disorder.[18] By contrast, all frames have a scoring function by which a pseudoprobabilistic threshold test may confirm hypotheses. Similarly, 9 of the frames have necessary conditions that may be used categorically to rule out a hypothesis, whereas all may be inactivated if their scores fall below another threshold. In our experience, the program performs best when presented with cases decided on categorical grounds. Too often, small variations in a borderline clinical case can push a score just above or just below a threshold and affect the program's conclusions significantly. Of course, in a textbook case, even the probabilistic mechanism will reach the right conclusion because the evidence all points in a consistent direction. Perhaps it should not disappoint us when the program flounders on tough, indeterminate cases where we have neither certain logical criteria nor a consensus from the evidence.

Once the reevaluation of all hypotheses affected by the last finding introduced is done, PIP selects an appropriate question to ask the user. That selection depends on the probabilistic evaluation of each active hypothesis. PIP identifies the highest-scoring active hypothesis, and if one of its expected findings has not yet been investigated, that finding is asked about. If all its expected findings have already been investigated, then PIP pursues expected findings of hypotheses complementary to the leading one.

To its user, PIP's reasoning is discernible from the conclusions it reaches and the focus of its questioning. PIP appears unnatural when its focus frequently shifts, as the probabilistic evaluator brings first one and then another competing hypothesis to the fore. This major deficiency relates to the lack of categorical reasoning. Such reasoning might impose a longer-term discipline or diagnostic style (Miller, 1975) on the diagnostic process.

In summary, PIP proposes categorically and disposes largely probabilistically.

9.3.2 INTERNIST—The Diagnostic System of Pople and Myers

INTERNIST (Oleson, 1977; Pople, 1975; Pople et al., 1975) is a computerized diagnostic program that emphasizes a very broad coverage of clinical diagnostic situations. The INTERNIST data base currently covers approximately 80% of the diagnoses of internal medicine (Pople, 1976), and thus is the largest of these AIM programs. Although INTERNIST is close to its goal of covering most of internal medicine, other problems lie down-

[18]Currently, PIP contains a total of 69 hypothesis frames, but 31 of them are so skeletal that they can never be confirmed. They are there to maintain the appropriate complementary relationships, and they anticipate a future extension of our data base.

Portal-vein-occlusion

Manifestation	*L*	*F*
Hepatic-vein-wedge-pressure-normal	0	4
Splenomegaly	1	4
Gastro-intestinal-hemorrhage	1	4
Varices-esophageal	2	4
Portal-vein-obstruction-by-radiography	5	3
Anemia	1	3
Appendicitis-history	1	2
Ascites	1	2

FIGURE 9-3 A diagnosis and its manifestations in INTERNIST. *L* indicates evoking strength; *F* indicates frequency.

stream for these researchers, including human-engineering issues centered on usability of the program's interface, possibly significant costs of running the program and maintaining the data base, introducing some model of disease evolution in time, and dealing with treatment, as diagnosis is hard to divorce from therapy in any practical sense.

Presentation

The INTERNIST data base associates with every possible diagnosis, D_i, a set of manifestations, $\{M_j\}$. A manifestation is a finding, symptom, sign, laboratory datum, or another diagnosis that may be associated with the diagnosis. For every M_j listed under D_i, two likelihoods are entered. $L_{D_i|M_j}$, the *evoking strength*, is the likelihood that if manifestation M_j is seen in a patient, its cause is D_i. It is assessed on a scale of 0 to 5, where 5 means that the manifestation is pathognomonic for the diagnosis and 0 means that it lends virtually no support. $F_{M_j|D_i}$, the *frequency*, is the likelihood that a patient with a confirmed diagnosis, D_i, would exhibit M_j.

Although INTERNIST's developers resist identifying these numbers as probabilities, $F_{M_j|D_i}$ is clearly analogous to the conditional probability $P_{M_j|D_i}$. The evoking strength is like a posterior probability, $P_{D_i|M_j}$, that includes a population-dependent prior, P_{D_i}, that is not explicit in the data base. If we were to take such a probabilistic interpretation, all the usual complaints about the failure of Bayesian assumptions would be appropriate. The INTERNIST scoring function that computes with these numbers is, however, in no sense probabilistic, and the rough granularity of the data is undoubtedly equally significant. It is reported that small random perturbations of the frequencies and evoking strengths in the data base do not significantly alter the program's behavior. A small example of a diagnosis, its associated manifestations, and the evoking strengths and frequencies connecting them are shown in Figure 9-3 (Pople, 1976).

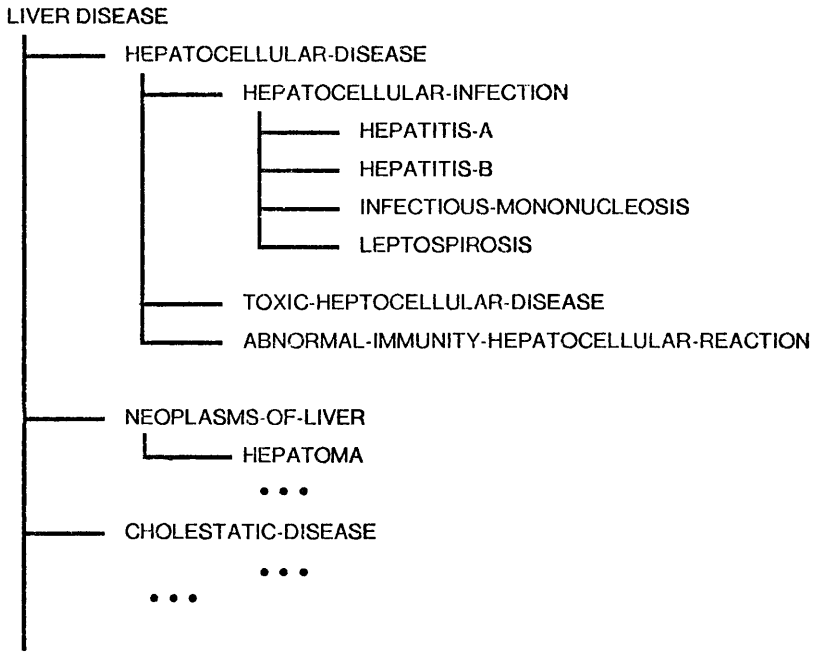


FIGURE 9-4 A small portion of INTERNIST's diagnosis hierarchy.

INTERNIST also classifies all its diagnoses into a disease hierarchy, a small part of which is shown in Figure 9-4 (Oleson, 1977). The use of hierarchy is an important mechanism for controlling the proliferation of active hypotheses during the diagnostic process because it allows a single general diagnosis to stand for all its possible specializations when no discriminating information is yet available to choose among them. This occurs, however, only when *all* specializations of the chosen general diagnosis have in common the same set of observed manifestations. Because INTERNIST wants to evaluate general as well as specific diagnoses, it *computes* for each general diagnosis a list of manifestations and their corresponding evoking strengths and frequencies. The manifestations for the general diagnosis are those common to each of its specializations, and the evoking strength and frequency of each are, respectively, the maximum evoking strength and minimum frequency of that manifestation among the specializations.

Borrowing the term from PIP, we will call a diagnosis *active* if at least one of its manifestations with a nonzero evoking strength has been observed, unless the diagnosis is a general one and must be replaced by its specializations (for example, because a manifestation occurring in one but not another of the more specific diagnoses has been reported). For each

active hypothesis, a score is computed by summing the scaled evoking strengths of all its manifestations that have been observed, adding "bonus" points for confirmed causally consequent diagnoses, subtracting the sum of frequencies of those of its manifestations that are known to be absent, and also subtracting a weight of *importance* for each significant finding that is reported to be present but that is not explained by either the diagnosis or some other confirmed diagnosis. Thus evocative findings and confirmed consequences of a diagnosis count in its favor, while expected findings that are known to be absent and reported findings that are unexplained count against it.

Discussion

Drawing an analogy with PIP, INTERNIST's diagnoses are PIP's hypotheses, the manifestations are the findings and causally related hypotheses, and the evoking strengths are like the triggers—they and the frequencies play the role of the scoring function. INTERNIST's use of the importance measure for unexplained findings is superior to PIP's simple fractional binding score. Because the scoring function in PIP is explicit in each hypothesis frame, it requires more effort to create but provides a more general means of evaluating the significance of present and absent findings. Also, because PIP provides some logical criteria for confirming or denying a hypothesis, it provides a data base with the option of categorical hypothesis evaluation.

The lumping together of findings with causally consequent diagnoses, both as manifestations, leads INTERNIST to some difficulties. For it, any manifestation is either present, absent, or unobserved. This may be appropriate for findings, but when imposed on the evaluation of diagnoses, ignores the arguably real support of a strongly suspected though not confirmed causally consequent diagnosis for its antecedent. As Pople has pointed out, this effect may prevent INTERNIST from diagnosing a syndrome of connected hypotheses if no one of them is definitely provable even though the circumstantial evidence of their combined high likelihood is convincing to a physician. A similar deficiency arises because reported findings are explained only by confirmed diagnoses. Again, a strongly suspected but not confirmed complementary hypothesis will not be able to explain its significant findings, and so the correct diagnosis may have its score strongly penalized. As discussed above, PIP addresses these problems by dealing more explicitly with complementary disorders and accepting that a hypothesis accounts for a finding if one of its *active* complementary hypotheses accounts for it. We will argue below, however, that both of these solutions are weakened by not having a sufficiently explicit model of the hypothesis they are pursuing.

The most interesting part of INTERNIST is its focusing mechanism. After scoring all its active diagnoses, INTERNIST chooses to concentrate

on the highest-ranking diagnosis. It partitions the others into two lists: the *competing* and the *complementary* diagnoses. A diagnosis is complementary to the chosen one if the two, together, account for more findings than either alone; otherwise the diagnosis is competing. The complementary list is then temporarily set aside, and a *questioning strategy* (one of RULE-OUT, NARROW, DISCRIMINATE, or PURSUE) is selected, depending on the number of high-scoring competitors and whether the information to be requested is low or high in cost. The complete scoring, partitioning, and strategy-selection processes are repeated after each new fact is reported. Confirmation is by numerical threshold. The partitioning heuristic is credited by Pople with having a very significant effect on the performance of the program, focusing its questioning on appropriate alternative diagnoses.

Because its intended coverage of disorders and findings is universal, INTERNIST relies on a uniform processing strategy and a simply structured data base. Much of its decision making falls under our probabilistic designation. The use of a hierarchic tree of diagnoses and of the rule for moving from a general to more specific diagnoses is categorical and captures an important part of a clinician's diagnostic behavior. The selection of questioning strategy is also categorical, although, interestingly, it depends on a probabilistic computation of the likelihood of each diagnosis.

9.3.3 CASNET—A Model of Causal Connectives

In a domain where normal and diseased states are well understood in physiological detail, it is sensible to build diagnostic models in which the basic hypotheses are much more detailed than the disease-level hypotheses of PIP and INTERNIST. Kulikowski, Weiss, and their colleagues have built such a system based on the causal modeling of the disease glaucoma. Their system is called CASNET, and it is in principle a general tool for building causal models with which well-known diseases may be diagnosed and treated (Weiss, 1974).

Presentation

CASNET defines a causal network of *dysfunctional states* and a set of *tests* that provide evidence about the likelihood of the existence of those states in the patient under consideration. States represent detailed dysfunctions of physiology, not complete diseases; thus the determination of disease is separated from the question of what, in detail, is going wrong in the patient.

The network consists of a set of nodes, some of which are designated as *starting states,* meaning that they are etiologically primary, and some as *final states,* meaning that they have no dysfunctional consequences. All

causal relationships are represented by a link between two nodes, with a link strength that is interpreted as the frequency with which the first node causes the second. Starting states are given a prior frequency. No cycles are allowed in the network. Almost all nodes are representations of real physiological disorders. Although logical combinations of physiological states may be represented by a single node (for example, to express joint causation), this technique is discouraged. Further, "the resolution of states should be maintained only at a level consistent with the decision-making goal. A state network can be thought of as a streamlined model of disease that unifies several important concepts and guides us in our goal of diagnosis. It is not meant as a complete model of disease" (Weiss, 1974).

Two separate probabilistic measures are computed for every state in the network. A node's *status* is an estimate of its likelihood from the results of directly relevant tests. The status determines whether a node is *confirmed* or *disconfirmed.* A node's *weight* is an essentially independent estimate of its likelihood that derives from the strength of causal association between the node and its nearest confirmed and disconfirmed relatives. The weight computation ignores test results that affect the node's own status but is sensitive to results that establish the confirmation status of its causal relatives.

All tests are binary and are entered with an evaluation of the cost of each. If a positive or negative test result is reported, a set of links from the test to nodes of the network implies the presence or absence, respectively, of the corresponding nodes. Each link is labeled with a confidence measure for both positive and negative results, separately. A test may represent a simple observation of the patient, or it may be a logical combination of specific results of other tests. Only the results of simple tests are directly asked of the user of the program—the others are computed from the results of simple tests.

The status of each node is measured in the same units that are used to report the confidence measures of the implications of tests. Every time the result of a test is reported, the status of every node to which that test is linked is recomputed: if the result of the test has less confidence (i.e., is smaller in magnitude) than the status of the node, no change occurs. If the test result has greater confidence, the node's status is changed to that value. If they are equal, but of opposite sign, the node's status is set to zero, and a contradiction is noted for the user. One threshold, T, is defined such that if the status of a node is less than $-T$, the node is *denied,* and if the status exceeds $+T$, the node is *confirmed.*

The use of a maximum-confidence value for status and the ability to define a high-confidence test as the conjunction of two lower-confidence tests are in the fuzzy set tradition. This approach sidesteps the problem of the interpretation of mutually dependent test results, as they arise in a Bayesian formulation, by requiring the designer of the data base to define explicitly a new test for any combination of tests that jointly support the same node. Weiss argues that in his application domain this is perfectly

appropriate, because when tests of varying confidence are available, only the results of the strongest should be counted (Weiss, 1974). One may question, however, whether this approach could be extended to wider medical areas, especially where many tests are available but only a consistent reading on most of them is enough to confirm a hypothesis.

Both for selecting a "most informative" test and for interpreting the pattern of status values among nodes of the network as a coherent disease hypothesis, CASNET defines an *acceptable path* in the network as a sequence of nodes that includes no denied nodes. A *forward weight* is computed for every node in the network, which represents the likelihood of that node when considering the degree to which its confirmed causal antecedents should cause it. Consider each admissible path that leads to node n_j and starts either at a starting node or at a closest confirmed node. CASNET computes the likelihood of causation along each such path by multiplying the link strengths along it (and the prior frequency for a starting state). The forward weight, w_j, of node n_j is defined to be the sum of the weights along each such path.

An *inverse weight,* representing the degree to which the presence of a node is implied by the presence of its causal consequents, is also computed.[19] CASNET then takes the maximum of the forward and inverse weights as the *total weight,* which is interpreted as a frequency measure of the degree to which the node is expected to be confirmed or disconfirmed from circumstantial causal evidence. Obviously, nodes with a high total weight and a status score near zero are excellent candidates for testing, since we might expect them to be confirmed. Conversely, nodes with low total weight are also candidates for testing, since we expect them to be denied. CASNET permits a number of different testing strategies to be used, based in part on the expected information implied by the weights and in part on the costs of the various tests.[20]

One should interpret the status of various nodes in the network as measures of the likelihood of subparts of a coherent disease. Based on the notion of the acceptable path, CASNET defines a number of different kinds of disease pathways, depending on which starting nodes are acceptable for such a path and on what criteria are used to terminate the path. It can compute those paths that are *most likely* to account for all the *confirmed* nodes in the network, all those that are *potential* explanations, and those that are not contradicted by a denied starting node (called *global*). Once the start of a disease path is selected, its termination criterion determines the type of path. An acceptable path that ends on a confirmed node is *confirmed.* An acceptable path ending on an undenied node is *possible.* A

[19]We cannot describe all of the computational mechanisms of CASNET here. An excellent presentation of the algorithms and a thorough justification for the particular choices made are in Weiss's thesis (1974).

[20]At present, the program is used with a fixed sequence of tests because an attempt is being made to gather a large, uniform data base about glaucoma patients. Thus the test selection function and this interesting weighting function are not in use (Weiss, 1976).

path that ends on a final state, even if it includes denied nodes, is *predictive*. Depending on the intent of the user, any combination of starting and termination criteria for a disease path may be selected. For example, the most likely starting criterion taken with the confirmed termination criterion will yield the "best estimate" diagnosis of the patient's current state. Selecting the global starting criterion and the predictive stopping criterion produces essentially all pathways through the network.

The most likely starting nodes are used to establish the probable causal mechanisms (the diseases) that account for the patient's difficulties. The ends of disease pathways give an estimate of the extent of the diseases. Together, these can be used to identify the primary disorder, to select a therapy for it, and to make prognostic judgments.

In a very clever manner, the determination of the effectiveness of therapy is handled by application of the same techniques used for diagnosis. A new causal network is constructed, in which the various therapies are the starting states and other nodes represent either complications of the treatments themselves or disorders not alleviated by the treatments. All of the above techniques are then available to assess whether any confirmed disorders are left after treatment and, if so, by what causal paths they could come about.

Discussion

At the level of testing, confirmation, and denial of nodes of the causal network, virtually all of CASNET's reasoning is probabilistic, based on the fuzzy set formalism for test interpretation and a probability interpretation for propagating causal frequency. The ability to define a hierarchy of tests (where higher tests summarize logical combinations of results of lower ones) and the simple confidence interpretation of node status provide a mechanism in which categorical rules for deciding node status are easily embedded.

The selection of a diagnosis and an associated therapeutic plan depends principally on the network designer's categorical understanding of the possible causal pathways through the net and on his or her definition of just which paths are subsumed by a given disease. In fact, if forward and inverse weights were not calculated, the elimination of any causal links that are not part of an identified disease path would result in no net effect on the operation of the program.

Weiss emphasizes that perfect accuracy in diagnosis by his program is not an unrealistic goal (presumably, without significant cost limitations on its testing strategy). This is to be contrasted to statistical classification schemes that would likely remain imperfect even with the addition of large quantities of new data. In CASNET, this confidence is justified because an error in the program's classification of a patient must ultimately indict some part of the causal model. In response, it may be necessary to add more

tests to help distinguish the erroneous case, or the network may need to be disaggregated in selected places to give a more detailed model of some aspect of the disease. In the typical statistical approach, where the unit hypothesis is the disease, such local refinement is less feasible.

The glaucoma program works so well because its domain is narrow and the pathophysiology is well understood. Especially when compared with the domain of all of internal medicine (INTERNIST) or renal disease (PIP), the level of detail that is medically known and that it is practical to include in the glaucoma program is great. In fact, we speculate that the program could be recast as a categorical reasoning program. Given a fixed flow chart for test selection, we might consider in turn each of the roughly 50 starting states. From each, we might imagine a discrimination network that traces those diseases that start with that starting node. The discrimination net would branch, based on the crudely quantized confidence measure (status) of each successor node. That same measure could be used to determine the end of the disease path and thus the degree of progression of the disease and its possible therapies. Of course, such a technique may be too rigid to use in a changing environment or may not capture some capabilities of the original program (e.g., it could not compute all possible causes of some dysfunction). We hasten to mark this as pure speculation, but it suggests that perhaps more powerful categorical decision-making techniques could equally well solve the glaucoma problem, and thus that the probabilistic appearance of the CASNET solution is perhaps unnecessary.

A causal model is, nevertheless, attractive. We have seen physicians create (occasionally incorrectly) causal explanations for phenomena that they associate with diseases even though such a causal model played no important role in their interpretation of the phenomena. People seem happier if they understand why something happens than if they merely know that, under given circumstances, it does. Causal models for diagnosing dysfunction have been implemented for simple physical devices (Rieger, 1975) and proposed for medicine (Smith, 1978). In both these approaches, causality is taken as a categorical, not a probabilistic, connection. Reasoning about likelihood is often quantified only in the very fuzzy sense of IMPOSSIBLE, UNLIKELY, POSSIBLE, PROBABLE, and CERTAIN, and distinctive rules rather than a uniform numerical computation are used to combine data with different degrees of likelihood.

9.3.4 Production Rules—MYCIN and Inference Nets

The final AIM program whose reasoning component we shall describe is MYCIN, which is being developed to advise physicians and medical students in the appropriate treatment of infections (Shortliffe and Buchanan, 1975) (see also Chapter 5).

IF: 1) The stain of the organism is gram positive and
2) The morphology of the organism is coccus and
3) The growth confirmation of the organism is chains
THEN: There is suggestive evidence (.7) that the identity of the organism is streptococcus

FIGURE 9-5 A typical MYCIN rule.

Presentation

MYCIN's knowledge is expressed principally in a number of independently stated rules of deduction, a typical example of which is shown in Figure 9-5. MYCIN's highest-level goal is to determine if the patient is suffering from a significant infection that should be treated, and if he or she is, to select the appropriate therapy. It uses a backward-chaining deduction scheme in which all applicable rules are tried: if a condition in the IF *(antecedent)* part of a rule is decidable from the data base, that is done; if the condition can be asserted by the THEN *(consequent)* part of some other rules, they are applied; otherwise, MYCIN asks the user. Thus the rule of Figure 9-5 might be applied in the following chain of reasoning:

1. To decide if the patient needs to be treated, we must decide if he or she has a significant infection.
2. We must know the likely identity of the infecting organism to decide if the infection is significant.
3. The rule of Figure 9-5 can determine the identity of the organism.

Because conditions in the rules may include logical disjunctions as well as conjunctions, the deduction forms an AND/OR tree.

When the methodology of MYCIN was applied to the simple domain of bicycle troubleshooting, a small set of categorical rules of this type was sufficient to give the program some interesting behavior. The complication in MYCIN arises from the uncertainty with which a medical rule implies its consequences, the applicability of several uncertain rules to suggest the same consequence, and the need to apply rules even when their antecedents are to some degree uncertain.

MYCIN associates a *certainty factor* (CF) with each rule, which is a number between 0 and 1, representing the added *degree of belief* that the rule implies for its consequent. With each fact in the data base is a *measure of belief* (MB) and a *measure of disbelief* (MD), both numbers between 0 and 1 that summarize all the positive and negative evidence that has been imputed for this datum by the application of rules that conclude about the

datum. The measures of belief and disbelief are maintained separately for each item, and the certainty factor of the fact is their difference. Thus the CF of a fact is a number between -1 and 1.

Arguing that the rule "A implies B with probability X" should not be inverted in the traditional probabilistic sense to entail "A implies not B with probability $(1-X)$," Shortliffe defines a *confirmation* formalism for computing the certainty of facts (Shortliffe and Buchanan, 1975). In its simplest form, it says the following: assume that we are told (perhaps by some rule $S1$) the fact H with certainty $\mathrm{MB}_{H|S1}$. Later, we discover that another source of information, $S2$, tells us H again, this time with certainty $\mathrm{MB}_{H|S2}$. Instead of using a maximum, as CASNET would, we would like to feel more confident in H after having received two reports in its favor than after having received either one by itself. MYCIN's scheme means that every new report of the truth of H reduces the difference between 1 and H's measure of belief by the fraction that is the certainty of the new report. For example, if $\mathrm{MB}_{H|S1} = 0.4$ and $\mathrm{MB}_{H|S2} = 0.6$, then the combined result is $\mathrm{MB}_{H|S1,S2} = 0.76$. This process is defined separately for positive and negative reports, and we have

$$\begin{aligned} \mathrm{MB}_{H|S1,S2} &= 0 \quad \text{if } \mathrm{MD}_{H|S1,S2} = 1 \\ &= \mathrm{MB}_{H|S1} + \mathrm{MB}_{H|S2}(1 - \mathrm{MB}_{H|S1}) \quad \text{otherwise} \end{aligned} \tag{6}$$

and

$$\begin{aligned} \mathrm{MD}_{H|S1,S2} &= 0 \quad \text{if } \mathrm{MB}_{H|S1,S2} = 1 \\ &= \mathrm{MD}_{H|S1} + \mathrm{MD}_{H|S2}(1 - \mathrm{MD}_{H|S1}) \quad \text{otherwise} \end{aligned} \tag{7}$$

where $S1$ and $S2$ are the two reports. The measures of belief and disbelief combine to give a certainty factor for each fact:

$$\mathrm{CF}_H = \mathrm{MB}_H - \mathrm{MD}_H$$

This, then, defines MYCIN's method of summarizing the certainty of a hypothesis when the application of several rules has contributed evidence for it.

To compute the measure of belief (or disbelief) contributed by a particular rule, MYCIN multiplies the CF of the rule by the MB (or MD) of the rule's antecedent. A fuzzy set strategy of maximizing for OR and minimizing for AND is adopted to compute the belief measures of the antecedent from the belief measures of its components. This approach is presented and justified in Shortliffe and Buchanan (1975) and Shortliffe (1976). An alternative formulation of separate measures of belief and disbelief is to be found in Shafer (1976).

Discussion

In MYCIN, the question of just what connections exist among different facts in the data base is not explicitly addressed. In addition to the rules that we have mentioned above, MYCIN also includes a *context hierarchy*, which plays a smaller but still important role in the program's operation. For example, the facts that "there are cultures associated with infections" and that "cultured organisms are associated with cultures" are embedded in no rules, but rather in this additional mechanism.[21] Turning MYCIN inside out, that context mechanism could be viewed as the principal organizational facility of the diagnosis program. In such a view, the underlying reasoning activity is filling in a *frame* for the patient by directly asking for some information (e.g., age and sex) and by instantiating and recursively filling in other frames (e.g., cultures and operations). The productions and their associated certainty factors are then seen as a set of procedurally attached heuristics to help fill in those frames. We conjecture that this methodology, which underlies the operation of the GUS program (Bobrow et al., 1977), would provide a reasonable alternative way of implementing the MYCIN system.

MYCIN's categorical knowledge is encoded in three ways. First, the presence of each rule implicitly establishes a categorical, inferential connection between those facts in its consequent and those it uses in its antecedent. The MYCIN control structure, which is a nearly purely categorical backward-chaining deduction scheme, is based on these relationships. Second, the context tree explicitly defines what objects may exist in MYCIN's universe of discourse and how they may relate. Such categorical information would underlie a GUS-like implementation of MYCIN. Third, many other relationships, which record such data as how to ask a question and what answers are acceptable, are also categorical in nature. MYCIN's probabilistic reasoning resides in its use of the measures of belief and disbelief about each fact and the certainty factors associated with each rule. Although this probabilistic method has important consequences for the assessment of the relative likelihoods of the various infecting organisms under consideration, it appears that it affects the program's questioning behavior only slightly. Except in the case where a line of reasoning is pursued because of the joint effect of several very weak independent inferences, which we suspect is rare, the particular numbers used make little difference except in the final diagnosis (and thus therapy). We note that the context tree that is built for each patient depends for its structure mainly on information that is always asked of the patient, such as what cultures have been taken, what operative procedures have been performed,

[21]Note that, because of interposed levels of complexity such as the existence of cultures, the example "traceback" we presented above of how MYCIN would decide to apply the rule of Figure 9-5 is overly simplistic.

and what drugs are being used in treatment. Even dramatic changes in the probabilistic component of MYCIN's reasoning strategy would not alter this behavior.

MYCIN has also inspired the creation of a more uniform inference scheme, in which every potential fact in the data base is viewed as a node in a large inferential network.[22] In such a network, the reasoning rules form the connections among the fact nodes, and we think of propagating some measures of likelihood among the nodes so that the impact of directly observable facts may be reflected on the diagnostic consequences of ultimate interest. This is the approach taken by Duda, Hart, and Nilsson (1976) in their *inference net* formalism. The propagation scheme used there is Bayesian in its heritage, but suffers from the typical distortions (see above) that the Bayesian methodology can introduce.

Of course, it is natural to compare the inference net to the causal net. The difference is primarily in the semantic interpretation of what a node and a link represent. In CASNET, the node is a dysfunctional state, and the link represents causality in the application domain. In the inference net, nodes are essentially arbitrary facts about the world, and rules are arbitrary implications among those facts. Much of Weiss's reasoning in justifying the particular propagation algorithms he has chosen rests on his specific interpretation of the network. Because the semantics of the inference net are less clearly (or constantly) defined, we must be more skeptical when evaluating the acceptability of the approximations introduced by the propagation formulas.

9.4 Another Look at the Problems of Diagnosis

Compared to the expert physician, our best AIM programs still have many deficiencies. We catalog a few of the more significant ones:

1. Programs that deal with relatively broad domains, such as INTERNIST and PIP, have inadequate criteria for deciding when a diagnosis is *complete.* There is no sense of when the major diagnostic problems have been resolved and only the "loose ends" remain: the programs continue exploring less and less sensible additional hypotheses until the user tires of the consultation. For example, PIP only stops if no active hypotheses remain or if every finding of every active hypothesis has been explored already.

[22]Uniformity is not necessarily an advantage for a reasoning scheme. For example, the particular structures used by MYCIN are cleverly exploited by Davis in building an interesting knowledge-acquisition module (Davis, 1976). In a uniform system of representation, it would be more difficult for his programs to decide just where new knowledge is to be added.

2. Because the initial strategy of the programs is to use every significant new finding as a clue to raise the possibility of associated disorders and because this strategy remains throughout the programs' operation, new hypotheses are continually being activated. Thus, when the program asks about an expected finding for one of its leading hypotheses and the finding is present, that finding often suggests new hypotheses as well, even though it is perfectly consistent with the diagnosis being pursued. Obviously, some such sensitivity is necessary or the program would remain committed to its first hypothesis, but we now feel that it would be preferable if new hypotheses were triggered only by evidence that contradicts a current belief.
3. Part of the routine developed by clinicians is an appropriate order for acquiring information systematically. Computer diagnosticians tend to enforce such an order either too strictly (e.g., the flow charts and MYCIN, which cannot accept out-of-sequence information in any useful way) or not at all (e.g., INTERNIST or PIP, where a global computation after the report of each fact may, in the worst case, change the program's focus to an entirely new topic for each question).
4. The programs rely on a global likelihood assessment scheme, but they use a semantics that is too weak for the states over which they try to compute approximate probabilities. For example, none of the programs can dynamically distinguish among the aggregate hypotheses
   a. A and B, both together, when in fact A has caused B,
   b. A and B co-occurring but apparently unrelated, and
   c. A or B but not both.

   Yet there are therapeutic and strategic decisions that hinge on just such distinctions. For example, it may be sufficient to treat only for A in the first case, but not in the second; trying to discriminate between A and B makes sense in the third case, but not in the others. PIP and INTERNIST might eliminate some of these hypotheses by noting those causal or associational links that are disallowed by the data base, but in no sense are these hypotheses generally distinguishable. MYCIN might include some rules that could, for example, reduce the possibility of hypothesis c, but it also lacks any mechanism to take up the problems of dependence. Although CASNET does allow the proper handling of this problem, it must do so by the creation of joint states, which is its weakest semantic ability.

9.4.1 Possible Improvements

The practice of clinical medicine offers some clues to the proper solution of some of these difficulties. Questions of the appropriate termination of the diagnostic process and control over the proliferation of hypotheses may be resolved by considering two factors. First, the diagnosis needs to be only

as precise as is required by the next decision to be taken by the doctor. Thus, if all the remaining possible diagnoses are irrelevant or equivalent in their implications for therapy or test selection, then nothing is lost by postponing their consideration. Bayesian programs that explicitly compare the cost of new information to its expected benefit will achieve this saving (Gorry et al., 1973), but none of the programs discussed here includes such a computation.

Second, the simple passage of time, "creative indecision," often provides the best diagnostic clues because the evolution of the disorder in time adds a whole new dimension to the other available information. Whereas MYCIN, CASNET, and the Digitalis Therapy Advisor all use changes over time as diagnostic clues, none of the programs exploits the possibility of deferring its own decisions with a deliberate eye to waiting for disease evolution. Such a strategy is also applicable on the much shorter time scale of the diagnostic session. In taking the present illness, for example, the doctor knows that a physical examination and a review of symptoms will soon provide additional information. Therefore, consideration of unlikely leads and small discrepancies can be deferred, leaving a coherent structure of problems to work with at the moment.

The ability to lay aside information that does not fit well with the current hypotheses is also a good mechanism for limiting the rapid shifts of focus caused by consideration of newly raised but unrelated hypotheses. In addition, however, the programs must have a sense of the orderly process by which information is normally gathered. The attempts in PIP to characterize a finding fully before proceeding and the attempts in INTERNIST and CASNET to ask summarizing questions (not described here) before launching on a series of similar, detailed questions are attempts to reflect such an order. We might, as Miller suggests (1975), go much further. We could, for example, incorporate a strategy that says, "When investigating a suspected chronic disease, insist on a chronological description of all the patient's relevant history." If such a strategy were followed, the program would not quickly jump at a "red herring" uncovered during the acquisition of those historical data. For example, consider a patient with a long history of sickle cell anemia who now complains of acute joint pain. Although that complaint would ordinarily raise the issue of rheumatoid arthritis, in this case we (and the program) should realize that the joint pain is a reasonable consequence of an already known disease process and should not evoke an immediate attempt to create elaborate additional explanations. Maintaining a richer semantic structure of just what the current hypothesis is and allowing that structure to control the program's focus of attention should also stabilize the program's behavior.

Another possible mechanism for controlling the logic of diagnosis is suggested by the following example. Consider the earliest stages in the diagnosis of chest pain, a symptom of potentially grave consequence. With a disaggregated structure of relationships between findings and hypotheses, chest pain might suggest angina pectoris, aortic stenosis, pneu-

monia, tuberculosis, pericarditis, costo-chondritis, depression, hiatus hernia, pancreatitis, esophagitis, gastric ulcer, fractured rib, pulmonary embolism, etc.—a long list of significantly different low-level hypotheses. Once those are all active, we must evaluate and compare all of them to choose a best hypothesis. On the other hand, we can say that, initially, we will only use the finding of chest pain to choose a somewhat specific diagnostic area for our further focus; specifically, we would like to choose one of these generic hypotheses: the pain is due to cardiac, pulmonary, gastrointestinal, psychogenic, or muscular-skeletal causes. We ask only the age and sex of the patient and three of the most important descriptors of the chest pain, its character, provocation, and duration. Obtaining a rank order for the five categories from each descriptor and combining them by a very simple arithmetic formula, we get a reasonably robust estimate of what is the best diagnostic area to pursue.

No simple scheme like the one suggested here is, of course, a panacea. However, we have been surprised at how effective rather crude heuristic techniques can be when they are tailored to a specific problem. To illustrate the necessity of that tailoring, it should be pointed out that the same technique appears *not* to be effective at the next level of diagnosis, for example, in sorting out the various possible cardiac causes of chest pain.

In summary, our analysis of the reasoning mechanisms of current AI programs leads us to these conclusions:

1. If possible, a carefully chosen categorical reasoning mechanism that is based on some simple model of the problem domain should be used for decision making. Many such mechanisms may interact in a large diagnostic system, with each being limited to its small subdomain. Many of the intuitively appealing observations made above can probably be implemented by the use of such techniques.
2. When complex problems need to be addressed—which treatment should be selected, how much of the drug should be given, etc.—then causal or probabilistic models are necessary. The essential key to their correct use is that they must be applied in a limited problem domain where their assumptions can be accepted with confidence. Thus it is the role of categorical methods to discover what the central problem is and to limit it as strongly as possible; only then are probabilistic techniques appropriate for its solution.

9.4.2 Postscript

As we interact with our medical colleagues at work, we are sometimes amazed by two observations:

1. They are often extremely reluctant to engage in any numerical computation involving the likelihood of a diagnosis or the prognosis for a

treatment. Even when official blessing is bestowed upon Bayesian techniques, we have seen both experienced and novice physicians acknowledge and then ignore them. Doctors certainly have a strong impression of their confidence in the diagnosis or treatment, but that impression must arise more from recognizing a typical situation or comparing the present case to their past experiences rather than from any formal computation of likelihoods.

2. An experienced physician can be pushed, in his or her domain of expertise, to give arbitrarily many complex potential explanations for a patient's condition. Especially in the teaching hospital environment with which we are most familiar, this serves the useful pedagogical purpose of discouraging pat answers from students. Because so many diagnostic possibilities appear to be available for the expert to consider, we suspect that the rapid generation and equally rapid modification or elimination of many explicit hypotheses play a significant role in his or her reasoning.

These observations reinforce our beliefs that somewhat more careful approaches to diagnosis are needed, ones that apply the most successful available techniques to each component of the diagnostic process. Although probabilistic techniques will be best in some well-defined domains, they should not be applied arbitrarily to making other decisions where the development of precise categorical models could lead to significantly better performance. The development and aggregation of a number of different approaches, both categorical and probabilistic, into a coherent program that is well suited to its application area remains a fascinating and difficult challenge.

When thinking about the effectiveness of a computerized medical consultant, it is essential to recognize the difference between impressive expertlike and truly expert behavior. A vehement critic of early work in artificial intelligence accused the practitioners of this "black art" of trying to reach the moon by climbing the tallest tree at their disposal (Dreyfus, 1972). We must be somewhat concerned that the initial successes of the current programs should not turn out to be merely the improved view from a lofty branch.

ACKNOWLEDGMENTS

This research was supported by the Department of Health, Education and Welfare (Public Health Service) under grant no. 1 R01 MB 00107-03 and by Dr. Pauker's Research Career Development Award (1 K04 GM 00349-01) from the General Medical Sciences Institute, National Institutes of Health.

10

Computer-Based Medical Decision Making: From MYCIN to VM

Lawrence M. Fagan, Edward H. Shortliffe, and Bruce G. Buchanan

We mentioned in the introduction to Chapter 5 that MYCIN provided a starting point for several additional research projects. The Ventilator Manager (VM) project of Larry Fagan had its beginnings in the MYCIN project but quickly diverged because of the dynamic nature of the intensive care unit (ICU) setting for which it was designed. MYCIN required a "snapshot" approach to patient assessment—temporal trends were poorly handled and advice was generally provided on the basis of a patient's situation at a single point in time. In dynamic settings like an ICU, however, decisions may be dependent on frequent sequential assessments of the patient's status. Fagan's work was accordingly also influenced by another earlier Stanford project known as HASP/SIAP (Nii et al., 1982). That system was not concerned with medical issues, but did develop iterative techniques for the ongoing analysis of signals.

Although Fagan was a graduate student at Stanford at the time, much of his work was based at Pacific Medical Center in San Francisco. The director of the postsurgical ICU there, Dr. John Osborn, had developed an elaborate monitoring system that was in routine use. However, the amount of data generated was sometimes overwhelming, particularly for physicians in training. It was clear that there was expertise involved in learning how to interpret the data, and the idea developed to build an expert system that could monitor the various physiological parameters and give advice accordingly. The following chapter discusses the resulting evolution from

From *Automedica*, 3: 97–106 (1980).

MYCIN to VM and explains how the differing requirements of the two clinical settings affected the ultimate design of the newer system. VM's specific approach is to use production rules for interpreting the physiological data, thereby permitting VM to aid in the management of patients being "weaned" from ventilators. This work is included here largely because of Fagan's insights regarding temporal reasoning in medicine. Particularly noteworthy is the development of techniques to allow the system to interpret patient data by comparing current findings with explicit expectations generated using production rules during earlier time periods.

10.1 Introduction

Since the early 1970s, researchers in computer-based medical reasoning have begun to recognize the potential benefits of applying symbolic reasoning techniques in clinical domains (see Chapter 3). One such research group is the Heuristic Programming Project at Stanford University. The first medical reasoning program developed by the project, known as the MYCIN system (Shortliffe, 1976), adopted symbolic processing techniques largely in response to a conviction that computer-based consultation systems, in order to be accepted by physicians, should be able to explain how and why a particular conclusion has been derived. Such systems should also be able to incorporate, organize, manipulate, and update large quantities of medical knowledge. Subsequently, a series of additional medical application programs using MYCIN's techniques has been created. In this paper we compare MYCIN, a program for infectious disease diagnosis and therapy, with a newer system, the Ventilator Manager (VM) program for measurement interpretation in the intensive care unit (ICU). Each of these programs uses a representation scheme, known as production rules (Davis and King, 1977), to encode the medical knowledge used for decision making. Each production rule is stated in the form "situation implies conclusion." Production rules may be chained together to form a line of reasoning leading from observed patient data to diagnostic and therapeutic conclusions. This report discusses the strengths of this form of knowledge representation and shows how production rules can be applied in two somewhat different clinical applications.

We begin by presenting the reasons that symbolic processing has been utilized for medical decision making. A brief discussion of the MYCIN program and a more detailed discussion of the VM program are included to demonstrate the use of the symbolic processing techniques. The design criteria for the two programs are compared. Differences in design criteria, plus experience with the MYCIN program, led to the extensions to the methodology described in the final section.

10.2 The Rationale for Using Symbolic Processing Techniques

There is increasing evidence that computer-based diagnosis and therapy programs will be accepted by physicians only if they meet a stringent set of design criteria. Several sets of design requirements have been suggested (Shortliffe et al., 1974) (see also Chapter 2). Although the overriding goal for any computer-based consultation program is, of course, that it be accurate, Gorry has suggested (see Chapter 2) that clinical decision systems should ideally have three additional capabilities: (1) the ability to maintain and manipulate a set of symbolic concepts, rather than mere numbers, (2) the ability to interact with clinicians using natural language, and (3) the ability to explain the reasoning process used to make conclusions. These goals were derived from his experience with a program that used decision analysis for the management of acute renal failure (Gorry et al., 1973). He concluded that detailed knowledge of medical concepts and the relationships between concepts would be required to reach reasonable conclusions reflecting a sense of the clinical context of the patient's problems. This could provide the program with a pragmatic view of the situation being analyzed. He encouraged the development of natural language communication in order to expedite the transfer of expertise, both from the expert to the program during the creation and expansion of the knowledge base and from the program to the user once the program becomes a clinical tool.

These criteria imply that the same piece of knowledge must be used in many different ways. The knowledge should be represented in a fashion that does not limit the manner in which it can be used. In many programming languages, one part of a program cannot access or modify another part. Thus incorporating the knowledge directly into the program's procedures limits the possible utilization of that knowledge. Facts must be in a form that can be manipulated as easily as numerical data are manipulated in conventional programming tasks.

The subfield of computer science known as artificial intelligence (AI) (Winston, 1977) has concentrated on using computers for symbolic reasoning rather than for calculating with numbers. One goal of our project has been to determine the strengths and limitations of the production rule methodology drawn from AI. Production rules offer the advantage of containing a small "packet" of knowledge. These packets can be combined to create a knowledge base of facts and relations known to the system. Using current symbolic processing languages, these rules can be translated from an external English-like syntax into an internal form that can be examined and interpreted by a task-independent control program. Because they can be displayed in English for communication with the user

RULE209

IF:
1) The site of the culture is blood, and
2) There is significant disease associated with this occurrence of the organism, and
3) The portal of entry of the organism is GI, and
4) The patient is a compromised host

THEN:
It is definite (1.0) that bacteroides is an organism for which therapy should cover

FIGURE 10-1 Example of a MYCIN rule. This is the English translation of a rule used to determine which organism may be causing the patient's infection.

and because they also facilitate the development of simple techniques for understanding natural language, production rules have allowed us to respond effectively to the design criteria outlined above.

10.3 Overview of MYCIN

MYCIN selects antimicrobial therapy for patients with severe infections (Shortliffe, 1976). The program uses knowledge obtained from infectious disease specialists; this knowledge was captured in the form of heuristics or "rules of thumb" that relate microbiological data and clinical signs and symptoms to possible pathogenic organisms. The details of the MYCIN program have been outlined in several other publications (referenced below) and will be described only briefly here.

10.3.1 Knowledge Representation

The MYCIN system is built around a set of medical concepts, such as the surgical history of the patient and the identity of infecting organisms. Each of these concepts is called a *clinical parameter.* Relations between the clinical parameters are used to build production rules of the form "IF premise THEN action." The premise of the rule is formed by the conjunction of statements about clinical parameters, for example, "The age is greater than 8" or "The patient has had recent neurosurgery." The action portion of the rule states what conclusions can be drawn from the premise with an associated measure of certainty. The English translation of a MYCIN rule is shown in Figure 10-1.

To perform a consultation, the rules must be combined together to form a line of reasoning (see Chapter 5). MYCIN uses a goal-directed

approach to integrate the knowledge, a process known as *backward chaining*. Starting with the top-level goal (i.e., to prescribe appropriate therapy), the program selects the set of rules that make a conclusion about this goal in their action part. The premise of each of these rules is evaluated to determine if a rule can be applied. If a fact needed to evaluate this premise is not available, then the program identifies other rules that make conclusions about the needed fact (or asks the user if no rules exist). In this manner, only the portion of the rule set that is relevant to the particular patient is examined. The number of questions asked is also minimized by this goal-directed search through the knowledge base.

The consultation program manipulates the rules as described above, but itself contains no knowledge about infectious diseases. The system also contains explanation and question-answering facilities that interact with both the knowledge in the rule set and an ongoing record of how rules were applied during a consultation (Scott et al., 1977). The definition and propagation of the measure of uncertainty *(certainty factor)* associated with each rule have also been a major area of concentration (Shortliffe and Buchanan, 1975). Evaluations (Yu et al., 1979a; 1979b) have shown that the performance of the system approaches that of a subspecialist in the two areas (bacteremia and meningitis) for which the knowledge base has been developed.

10.4 Overview of VM

The VM program is designed to interpret on-line quantitative data in the intensive care unit (ICU). These data are used to manage postsurgical patients receiving mechanical ventilatory assistance. VM is an extension of a physiologic monitoring system (Osborn et al., 1969) and is designed to perform five specialized tasks in the ICU: (1) to detect possible measurement errors, (2) to recognize untoward events in the patient/machine system and suggest corrective action, (3) to summarize the patient's physiologic status, (4) to suggest adjustments to therapy based on the patient's status over time and long-term therapeutic goals, and (5) to maintain a set of patient-specific expectations and goals for future evaluation by the program. The program produces interpretations of the physiologic measurements over time, using a model of the therapeutic procedures in the ICU and clinical knowledge about the diagnostic implications of the data.

Most medical decision-making programs, including the MYCIN system described above, have based their advice on data available at one particular time. In actual practice, the clinician receives additional information from tests and observations over time and reevaluates the diagnosis and prognosis of the patient. Both the progression of the disease and the response

```
STATUS RULE: STABLE-HEMODYNAMICS
DEFINITION: Defines stable hemodynamics based on blood pressures and heart rates
APPLIES to patients on VOLUME, CMV, ASSIST, T-PIECE
COMMENT: Look at mean arterial pressure for changes in blood pressure and
systolic blood pressure for maximum pressures
IF
  HEART RATE is ACCEPTABLE
  PULSE RATE does NOT CHANGE by 20 beats/min. in 15 min.
  MEAN ARTERIAL PRESSURE is ACCEPTABLE
  MEAN ARTERIAL PRESSURE does NOT CHANGE by 15 torr in 15 min.
  SYSTOLIC BLOOD PRESSURE is ACCEPTABLE
THEN
  The HEMODYNAMICS are STABLE
```

FIGURE 10-2 Sample VM interpretation rule. The meaning of ACCEPTABLE varies with the clinical context—for example, the type of ventilatory assistance. VOLUME, CMV, ASSIST, and T-PIECE refer to types of ventilation therapies.

to prior therapeutic interventions are important for assessing the patient's situation.

Data are collected in different therapeutic contexts. In order to interpret the data properly, VM includes a model of the stages that a patient follows from ICU admission through the end of the critical monitoring phase. Correct interpretation of physiologic measurements depends on knowing which stage the patient is in. The goals for patient management are also stated in terms of these clinical contexts. The program maintains descriptions of the current and optimal ventilatory therapies for any given time.

Knowledge is represented in VM by production rules of the following form:

```
IF:   Relations about one or more parameters hold
THEN: 1) Make a conclusion based on these facts,
      2) Make appropriate suggestions to clinicians, and
      3) Create new expectations about the future values of parameters
```

Additional information associated with each rule includes the symbolic name, the rule group (e.g., rules about instrument faults), the main concept (definition) of the rule, and all of the therapeutic states in which it makes sense. Figure 10-2 shows a sample rule for determining hemodynamic stability.

The VM knowledge base includes rules to support five reasoning steps that recur whenever a new time segment begins: (1) characterizing measured data as reasonable or spurious; (2) determining the therapeutic state of the patient (currently the mode of ventilation); (3) adjusting expectations of future values of measured variables when patient state changes; (4) checking physiologic status, including cardiac rate, hemodynamics, ventilation, and oxygenation; and (5) checking compliance with long-term

INITIALIZING RULE: INITIALIZE-CMV
DEFINITION: Initialize expectations for patients on controlled mandatory ventilation (CMV) therapy
APPLIES to all patients on CMV
IF ONE OF:
  PATIENT TRANSITIONED FROM VOLUME TO CMV
  PATIENT TRANSITIONED FROM ASSIST TO CMV
THEN EXPECT THE FOLLOWING:

		[----------acceptable range ---------]				
	very low	low	[----ideal ----] min	max	high	very high
Mean pressure	60	75	80	95	110	120
Heart rate		60			110	
Expired pCO2	22	28	30	35	42	50

FIGURE 10-3 Portion of an initializing rule. This rule establishes initial expectations of acceptable and ideal ranges of variables. Not all ranges are defined for each measurement. pCO2 is a measure of the percentage of carbon dioxide in expired air measured at the mouth.

therapeutic goals. Each reasoning step is associated with a collection of rules sorted by the type of conclusions made in the action portion of the rule, for example, all rules that determine the validity of the data.

10.4.1 Treating Measurement Ranges Symbolically

Most of the rules represent the measurement values symbolically, using the term ACCEPTABLE or IDEAL to characterize the appropriate ranges. The actual meaning of ACCEPTABLE changes as the patient moves from state to state, but the statement of the relation between the physiologic measurements remains constant. The use of symbolic statements (e.g., "HEART RATE is ACCEPTABLE") allows for the exposition of common principles of physiologic interpretation in different contexts. In addition, it minimizes the number of rules needed to describe the complexity of the diagnostic situation.

The meaning of the symbolic range is determined by rules that establish expectations about the value of measured data. For example, when a patient is taken off the ventilator, the upper limit of acceptability for the expired carbon dioxide measurement is raised. The actual numeric calculation of "expired pCO2 high" in the premise of any rule will change when the context switches (removal from ventilatory support), but the statement of the rules remains the same. An example of a rule that creates these expectations is shown in Figure 10-3.

10.4.2 Rule Interpretation

The VM rule interpreter is based on the MYCIN interpreter. The major changes include (1) forward-chaining (data-driven) rule invocation as opposed to backward chaining, (2) checking to see that information acquired in a previous time frame is still valid for making conclusions, and (3) cycling through appropriate parts of the rule set each time new information is available.

A data-driven approach is necessary to take advantage of the small set of measurement values available in each time frame. This means that the reasoning process works forward from the available information as opposed to working backward from a goal and obtaining information as necessary. Because of the demanding nature of the ICU environment, the system must acquire and interpret data with minimal staff intervention.

Each of the rule groups corresponding to the five reasoning steps mentioned above is considered in order. Each rule is examined to determine if it applies to the current context. The premise of the rule is examined to determine validity, and the appropriate conclusions are recorded by the program, as well as expectations on the future ranges of measurement values. Suggestions to clinicians are also printed out.

Often the examination of the rule premise requires the utilization of a value acquired earlier, for example, the temperature measurement, which is volunteered to the patient-monitoring system on an episodic basis. The reliability of the stored value is determined by evaluating either a time constant (for variables that predictably change over time) or a rule (for cases in which the assessment of a value's reliability is dependent on context-specific information). Associated with each parameter in the system is a specific mechanism for determining its reliability over time. If a measurement is concluded to be spurious or outdated, then it is treated as if it were unknown, requiring alternative methods for determining the status of the patient. The rule invocation process is repeated each time a new set of measurements is available (currently every 2 to 10 minutes).

Identical conclusions made in contiguous time frames are represented by the interval specified by the times of the first and last assertion. A list of these intervals summarizes the history of a particular conclusion. The evaluation of a rule clause such as "Patient hyperventilating for the past 30 minutes" is made by direct examination of the time intervals stored along with the conclusions, as opposed to looking at the original measurements. Expectations are associated with the appropriate measurement and are classified by duration and type, such as the upper limit of the acceptable range. Expectations can persist for a fixed interval, such as "for 20 minutes starting in 10 minutes," or for the duration of one or more clinical situations, for example, "while the patient is on the ventilator."

10.5 Comparison of Design Goals for MYCIN and VM

MYCIN was designed to serve in the ward setting as an expert consultant for antimicrobial therapy selection. A typical interaction might take place after the patient has been diagnosed and preliminary cultures have been drawn but little microbiological data are available. In critical situations, a tentative decision about therapy must often be made pending actual culture results. In return for assistance in making this decision, the clinician is asked to spend the small amount of time required to seek a consultation. As we have discussed, there are numerous challenges involved in the effort to motivate clinicians to use such a resource. The environment of the intensive care unit is quite different, however. Continuous surveillance and evaluation of the patient's status are required. The problem is one of making therapeutic adjustments over a long period of time, many of which are minor, such as adjusting the respiratory rate on the ventilator. The main reasons for interacting with VM would be to obtain status information or to investigate an unusual event. The program must therefore be able to interpret measurements with minimal human participation. When an interaction does take place, for example, when an unexpected event is noted by the program, it must be terse and concise.

This difference in the timing and style of the user/machine interaction has considerable impact on system design. For example, the VM system must (1) presume that the clinician's input into the system will be brief, (2) use historical data to determine the clinical situation, (3) be able to provide advice at any point in the hospital course of the patient, (4) be able to follow up on the outcomes of previous therapeutic decisions, and (5) be able to provide summaries of conclusions made over time. VM's environment thus differs from MYCIN's in that typed natural language input is an unlikely modality for communication with the clinician.

A consultation program should also be able to model the changing medical environment so that the program can interpret the available data in the appropriate context. Of course, areas like infectious diseases often have critical points where a consultation is most necessary. In the development of the meningitis section of the MYCIN knowledge base, the concept of "partially treated meningitis" (prior treatment with an antibiotic) was handled quite distinctly from the untreated case, even though the laboratory findings might be identical.

It was also necessary for VM to contain knowledge that could be used to evaluate the results of its therapeutic advice, just as a human consultant follows a case over a period of time. This is complicated by the fact that the user of the system may not follow the recommended therapy regimen. If the patient does not react as expected to the given therapy, then the program has to determine what alternative therapeutic steps may be required.

10.6 Extending the MYCIN Design

The VM program has been used as a test-bed to investigate methods for increasing the capabilities of symbolic processing approaches by extending the production rule methodology. The main area of investigation has been in the representation of knowledge about dynamic clinical settings. There are two components to representing a situation that changes over time: (1) providing the mechanism for accessing and evaluating data in a new time frame, and (2) building a symbolic model to represent the ongoing processes in the medical environment.

Another aspect of VM development has been to experiment with more general extensions to production rules based on observations of the use of the MYCIN system. These changes can be described by two research directions: (1) expanding the level of detail in the knowledge base, and (2) increasing the global structure of the knowledge base. The problem of designing an advice-giving program with limited user/machine interaction has also been explored.

10.6.1. Representing Knowledge About Dynamic Clinical Settings

With VM we have begun to experiment with mechanisms for providing MYCIN-like systems with the ability to represent the dynamic nature of the diagnosis and therapy process. The original MYCIN system was designed to produce therapeutic decisions for one critical moment in the patient's hospital course. This was extended with a "restart mechanism" that allows for selectively updating those parameters that might change in the interval between consultations. MYCIN can start a new consultation with the updated information, but the results of the original consultation are lost. In VM three requirements are necessary to support the processing of new time frames: (1) examining the values of historical data and conclusions, (2) determining the validity of those data, and (3) combining new conclusions with previous conclusions.

New *premise functions*, which define the relationships about parameters that can be tested when a rule is checked for validity, were created to examine the historical data. Premise functions used in MYCIN include tests to see if: (a) any value has been determined for a parameter, (b) the value associated with a parameter is in a particular numerical range, or (c) there is a particular value associated with a parameter. VM includes a series of time-related premise functions. The first function examines trends in input data over time, for example, "The mean arterial pressure does not rise by 15 torr in 15 minutes." A second function determines the stability of a series of measurements by examining the variation of measurements over a specific time period. Other functions examine previously deduced con-

clusions, as in "The patient has been on the T-piece for greater than 30 minutes" or "The patient has never been on the T-piece." Functions also exist for determining changes in the state of the patient, for example, "The patient has transitioned from assist mode to the T-piece." When VM is required to check whether a parameter has a particular value, it must also check to see if the value is "recent" enough to be useful.

The notion that data are reliable for only a given period of time is also used in the representation of conclusions made by the program. When the same conclusion is made in contiguous time periods (two successive evaluations of the rule set), then the conclusions are coalesced. The result is a series of intervals that specify when a parameter assumed a particular value. In the MYCIN system this information is stored as several different parameters. For example, the period during which a drug was given is represented by a pair of parameters corresponding to the starting and ending times of administration. In MYCIN, if a drug was again started and stopped, a new entity—DRUG-2—would have to be created. The effect of the VM representation is to aggregate individual conclusions into "states" whose persistence denotes a meaningful interpretation of the status of the patient.

10.6.2 Building a Symbolic Model

A sequence of states recognized by the program represents a segmentation of a time line. Specifying the possible sequences of states in a dynamic setting constitutes a symbolic model of that setting. The VM knowledge base contains a model of the ventilatory therapies. This model is used in three ways by the program: (1) to limit the number of rules examined by the program, (2) to provide a basis for comparing actual therapy with potential therapies, and (3) to provide the basis for the adjustment of expectations used to interpret the incoming data.

Attached to each rule in VM is a list of the clinical situations in which the rule makes sense. When rules are selected for evaluation, this list is examined to determine if the rule is applicable. This provides a convenient filter to increase the speed of the program. A set of rules is utilized to specify the conditions for suggesting alternative therapeutic contexts. Since these rules are examined every few minutes, they serve both to suggest when the patient's condition has changed sufficiently for an adjustment in ventilatory therapy and to provide commentary concerning clinical maneuvers that have been performed but are not consistent with the embedded knowledge for making therapeutic decisions. The model also provides mechanisms for defining expectations about reasonable values for the measured data. Much of the knowledge in VM is stated in terms of these expectations, and they can be varied in response to changes in the patient's situation.

RULE 236
(This rule applies to organisms from positive cultures, and is tried in order to find out about the infection that requires therapy or whether there is significant disease associated with this occurrence of the organism.)

IF:
1) The site of the culture is urine, and
2) The method of collection of the culture is voided, and
3) The colony count (in thousands) of the organism is greater than or equal to 100

THEN:
1) There is suggestive evidence (0.5) that the infection that requires therapy is cystitis, and
2) There is suggestive evidence (0.7) that there is significant disease associated with this occurrence of the organism

Author: Yu.
Comments: This definition of significance differs from E. Kass's original definition (Am. J. Med., 18:764, 1955) where two consecutive cultures are required. However, for practical purposes, if the patient is symptomatic, physicians generally start treatment on the basis of only one culture.
Created: 19 May 1977, 13:43.
Last edited: 1 June 1977, 11:50.

FIGURE 10-4 Example of a MYCIN rule with justificatory comments.

10.6.3 Expanding the Level of Detail in the Knowledge Base

Those who implement production rule systems often assume that the knowledge to be represented will be broken into small pieces corresponding to individual rules. What would happen in MYCIN if this assumption were violated? At one extreme there would be a single rule that weighed all of the clinical inputs in order to conclude the presence or absence of a single organism, say *E. coli,* but this would be too large and complicated to understand. The other extreme would be to base the deductive steps on the most minute details of physiologic knowledge, for example, knowledge of the cell wall properties of each species of bacteria. Explanation and modification would be very difficult in either situation. The approach taken in the development of MYCIN has been between these two extremes. Although no fixed criteria have been established, an examination of the rule set shows that intermediate steps have been left out when they appeared to be definitional in nature. Since the major performance requirements of a consultation system, that is, reaching correct hypotheses, revolve around propagation of the uncertainty associated with each piece of knowledge, definitional facts affect the outcome primarily by providing "commonsense" domain knowledge. Currently each of MYCIN's rules is augmented with a free-text justification or rationale that discusses some of the intervening steps that were used in formulating the particular content of that rule. The text justifications are available to the user if the basis for the knowledge in a rule is not clear from the translation of the rule itself (Figure 10-4).

Our representation of medical knowledge has been particularly stereotyped so that the programs we write can examine and manipulate the knowledge in many different ways. For example, in the middle of a MYCIN consultation the user can ask for an explanation of why a particular question is asked, resulting in the description of the chain of reasoning leading to the current rule under consideration (Davis, 1976; Scott et al., 1977). However, because a rule's justification is stored as unformatted text, it is unavailable for dissection and manipulation by the program as it gives explanations. It has become clear to us that the development of more formal mechanisms for encoding the basic knowledge that underlies a single rule (in a form that a computer can manipulate) will improve the educational and explanatory features of the program by providing an additional level of detail that can be explored and utilized programmatically. The detailed justifications could also be used for consistency checking since they represent the same knowledge but are stated in terms of "first principles." The requirements for augmenting the knowledge base in this way for the purpose of tutoring medical students have been described by Clancey (1979a).

The approach taken in VM is to introduce additional rules that are often definitional in nature (e.g., the rule in Figure 10-2 that defines hemodynamic stability). We have found that these additional rules act to form a convenient method for introducing abstract concepts into the rule base. This, in turn, has provided a basis for separating out the portion of the knowledge that was independent of the current context, for example, the physiology, from the knowledge that must adjust to the changing medical situation.

10.6.4 Increasing the Structure of the Knowledge Base

In addition to the need for more highly formalized justifications associated with each rule, we have observed the potential value of a more global organization of the rule base. In the development of a set of rules for the treatment of meningitis, we identified a situation in which a series of very similar rules were used to represent a "case analysis" of patient findings. The development of the meningitis knowledge base also included the need to represent *default decision rules* that applied to the majority of the patients considered but could still be customized for individual patient histories. The problem was broken up into a master rule that would make a preliminary set of conclusions and more specific rules that could modify the preliminary conclusions in response to unusual items from the patient's history. These more specific rules, therefore, cannot be understood without first considering the default rule. Two different methods can be used to handle this dependence. The first would be to rewrite each of the specific rules in order to incorporate all of the information in the default rule (and the default rule would then have to be changed to specifically exclude each

of the special cases). Then each of the rules would be more complex but somewhat more independent. However, it would be difficult to relate the differences in conclusions based on one special situation versus another. An alternative solution would be to recognize the inherent structure in the segment of knowledge that has been distributed across several rules. A technique used in other symbolic processing approaches (Pople, 1977) (see also Chapter 6) is to promote prototypical situations (and their exceptions) as the basic unit of knowledge representation. Information in these systems is often organized around individual diagnoses and groups together all of the knowledge pertaining to a particular disease. This method has the disadvantage that the size of each of these prototypical units, known as *frames* or *schemas* (Minsky, 1975), can become too large to comprehend. These organizational structures can also be used to provide for a more coherent consultation by supplying a larger context for the question-asking mechanism.

We have experimented with another representation for structuring the rule base: the creation of a rule set containing knowledge about the medical knowledge of the system (*meta-knowledge*) (Davis, 1976). These *meta-rules* can be used as "strategy rules" to order the application of rules in the knowledge base. They provide a heuristic mechanism for taking into account the facts that some information may be more relevant for making a specific conclusion and that other rules, although potentially applicable, can likely be ignored.

Another use for a global structure overlaid on the knowledge base would be to provide for anatomical models. Reggia (1978) suggests that this would have been useful in the development of a production rule system for neurological localization. Aikins (1979) has explored the combination of production rules and frames using the MYCIN methodology for the interpretation of pulmonary function tests.

The designers of future rule-based systems should consider some of the above methods for providing a global structure for the knowledge. Not all rules can be considered independently, and when rules are related, the connections should be available for manipulation by the computer.

10.6.5 Handling Limited User Input

In the intensive care unit, the lack of communication is partly solved by the availability of a large mass of on-line computer-processed data. Another approach to solving the communication problem is to display for the clinician conditional conclusions that require clinical observation before being carried out. For example, rather than asking whether a patient is sweating, VM might display a recommendation such as "If the patient is diaphoretic I suggest . . . , otherwise. . . ."

One additional solution to the problem of limited user-to-machine communication would be to anticipate the key questions that might be posed by the clinician at the bedside and provide a "menu" of likely ques-

tions for exploring the conclusions generated by the program. During the development of part of the meningitis knowledge base, the MYCIN program was modified to generate automatically the answers to a few key questions specified in advance by the medical expert. Such a key question for the ICU setting is "What is the status of ventilatory therapy?" The program, by the evaluation of several of the rules, can produce the following type of explanation: "Before transition to the T-piece can be suggested, hemodynamic stability must be present, which requires systolic blood pressure to be acceptable (current systolic blood pressure value is 170)."

10.7 Summary

Several years of experience with the MYCIN program have led to an understanding of additional requirements for symbolic processing approaches to medical decision making. These include extending the knowledge base beyond the facts necessary for high performance, providing an organizing structure for a large number of production rules, and extending the decision-making aids to include assistance throughout the patient's clinical course. For decision aids in the intensive care unit or other equally dynamic situations, programs cannot depend on interaction with the clinical users. Furthermore, they must handle data that are changing over time, but might be missing or spurious. They must also be able to provide tracking of the patient's status during the course of the underlying disease or in response to therapeutic intervention. A more complete description of the VM program can be found in Fagan (1980).

ACKNOWLEDGMENTS

We wish to thank Jonathan King for his comments on an earlier draft of this paper. The following people have also contributed to the work described here: J. S. Aikins, S. G. Axline, W. J. Clancey, S. N. Cohen, R. Davis, E. A. Feigenbaum, J. C. Kunz, J. J. Osborn, B. J. Rubin, A. C. Scott, S. M. Wraith, and V. L. Yu. This work has been supported by the National Institutes of Health (general medical sciences grant GM-24669) and by the Bureau of Health Sciences (research and evaluation grant HS-01544). Dr. Shortliffe is the recipient of a research career development award (LM-00048) from the National Library of Medicine. Computing resources have been provided by the SUMEX-AIM facility, under NIH grant RR-00785.

11

Intelligent Computer-Aided Instruction for Medical Diagnosis

William J. Clancey, Edward H. Shortliffe, and Bruce G. Buchanan

As AIM researchers began to develop techniques for allowing systems to explain their reasoning, some researchers became intrigued by the potential educational role of the developing methods. It became clear that advanced computer-aided instruction (CAI) programming techniques could be applied and extended in the medical setting. Intelligent computer-aided instruction (ICAI) differs from traditional CAI in its use of AI techniques for representing both subject material and teaching strategies.

Among ICAI programs, Clancey's GUIDON system described in this chapter is one of the largest and most complex. It contains all of the knowledge of MYCIN (Chapter 5) and uses a variety of techniques for mixed-initiative dialogue, student modeling, and response to partial student solutions. As a Stanford graduate student, Clancey had been involved in much of the early work on MYCIN and also became interested in ICAI and the possibility of adapting MYCIN for educational purposes. Thus GUIDON reflects the tremendous effort that went into building MYCIN's knowledge base of infectious disease rules, as well as nearly a decade of research in building ICAI systems. MYCIN's good performance in reaching decisions and giving explanations made a tutoring application of the knowledge base attractive. GUIDON also demonstrates the value of representing knowledge so that it can be applied in multiple settings, here for both consultation and teaching. This is the main advantage of separating

 From *Proceedings of the Third Annual Symposium on Computer Applications in Medical Computing*, Silver Springs, Md., October 1979, pp. 175–183.

the medical knowledge from the inference engine and encoding the medical knowledge in a stylized, program-readable form.

This chapter briefly outlines the difference between traditional instructional programs and ICAI. It then illustrates how GUIDON makes contributions in areas important to medical CAI: interacting with the student in a mixed-initiative dialogue (including the problems of feedback and realism), teaching problem-solving strategies, and assembling a computer-based curriculum.

In evaluating GUIDON's performance, one can see the value in the basic idea of formalizing teaching knowledge in procedures that are separate from the knowledge to be taught. However, the program is inherently limited by the MYCIN knowledge base. The rule set is poorly structured, does not contain pathophysiological knowledge for justifying the diagnostic associations, and does not explicitly state the strategies for gathering information and focusing on hypotheses. Thus the teaching perspective puts MYCIN's rules into sharp relief, revealing how they are crafted for good problem-solving, at the expense of making certain forms of common medical knowledge implicit (Clancey, 1983b).

GUIDON research evolved into a reconsideration of what a medical student needs to be taught about diagnosis. What are the diagnostic strategies of the primary care physician (as opposed to MYCIN's specialized top-down approach)? How are causal and subtype relations used to index medical knowledge during problem solving? This study of expertise (described briefly in Chapter 15) is complementary to Feltovich's psychological experiments, which reveal expert knowledge that is not formalized in medical textbooks (Chapter 12) and Gomez's and Chandrasekaran's emphasis on the interrelation of disease knowledge (Chapter 13). The other side of knowing what to teach is developing techniques for representing procedures in a way that makes explanation possible. Swartout's methods (Chapter 16) nicely complement the analysis and improvements to MYCIN that evolved from GUIDON research.

11.1 Introduction

Computer programs designed as aids for teaching medicine have been under development since the early 1960s. While some programs have been used for managing the use of conventional instructional material and grading tests, the predominant application has involved using the computer as a device that interacts with the student directly (Trzebiakowski and Ferguson, 1973). This application is generally called *computer-aided instruction* (CAI).

The goal of CAI research is to construct instructional programs that incorporate well-prepared course material in lessons that are optimized for

each student. Early programs were either electronic "page-turners" that printed prepared text and simple, rote drills or practice monitors that printed problems and responded to the student's solutions using prestored answers and remedial comments. In the intelligent CAI (ICAI) programs of the 1970s, course material is represented independently of teaching procedures so that problems and remedial comments can be generated differently for each student. Research today focuses on the design of programs that can construct a truly insightful model of the student's strengths, weaknesses, and preferred style of learning. It is believed that AI techniques will make possible a new kind of learning environment.

In this paper, we outline traditional CAI techniques and discuss the advantages of ICAI programs. GUIDON, an ICAI program for teaching medical diagnosis, is introduced. We then characterize the design issues of past medical CAI programs and illustrate how GUIDON makes contributions to these areas of concern.

11.1.1 Traditional CAI

In traditional systems (Harless et al., 1971; Weinberg, 1973), a course material author attempts to anticipate every wrong student response and prespecifies branching to specific teaching material based on the underlying misconceptions that he or she associates with each wrong response. Branching on the basis of response was the first step toward individualization of instruction (Crowder, 1962). This style of CAI has been dubbed *ad hoc, frame-oriented* (AFO) CAI by Carbonell (1970) to stress its dependence on author-specified units of information.

11.1.2 Intelligent Computer-Aided Instruction

In spite of the widespread application of AFO CAI to many problem areas, many researchers believe that most AFO courses do not make the best use of computer technology. Carbonell has pointed out that a programmed text can do much of what is required in CAI systems of the AFO type (Carbonell, 1970). In this pioneering paper, Carbonell goes on to define a second type of CAI that is known today as knowledge-based or intelligent CAI. Early CAI systems did, of course, have representations of the subject matter they taught, but ICAI systems also carry on a natural language dialogue with the student and use the student's mistakes to diagnose misunderstandings. ICAI has also been called *generative* CAI (Wexler, 1970) because it is typified by programs that present problems by generating them from a large knowledge base representing the subject material to be taught (Koffman and Blount, 1973).

However, the kind of program that Carbonell was describing in his paper was to be more than just a problem generator. Rather, it was to be

a computer-tutor that had the inductive powers of its human counterparts and could offer what Brown et al. (1976) call a *reactive learning environment,* in which the student is actively engaged with the instructional system and his or her interests and misunderstandings drive the tutorial dialogue.

The realization of the computer-tutor has involved increasingly complicated computer programs and has prompted CAI researchers to use artificial intelligence techniques. Artificial intelligence (AI) work in natural language understanding, the representation of knowledge, and methods of inference, as well as specific applications such as algebraic simplification, calculus, and theorem proving, have been applied by various researchers toward making CAI programs that are more intelligent and more effective. Early research on ICAI systems focused on representation of the subject matter (Carbonell, 1970; Suppes and Morningstar, 1972; Brown et al., 1974). The high level of domain expertise in these programs permitted them to be responsive in a wide range of problem-solving interactions.

In the mid-1970s, a second phase in the development of generative tutors has augmented knowledge representation techniques with expertise regarding the student's learning behavior, as well as tutorial strategies (Brown and Goldstein, 1977). AI techniques are used to construct models of the learner that represent his or her knowledge in terms of *issues* (Burton and Brown, 1976) or *skills* (Barr and Atkinson, 1975) that should be learned. These models then control tutoring strategies for presenting the instructional material. Finally, some ICAI programs are now using AI techniques to represent explicitly tutoring strategies themselves, gaining the advantages of flexibility and modularity of representation and control (Brown et al., 1976; Goldstein, 1977).

11.1.3 What Medical CAI Programs Attempt to Teach

Medical problem-solving skills can be categorized into three types: manipulative, interpersonal, and cognitive (Hoffer et al., 1975; Feinstein, 1977a). Manipulative skills involve acquisition of data and treatment by instrumentation. Interpersonal skills are involved in taking a patient history and discussing a diagnosis and alternative therapies. Cognitive skills comprise judgmental knowledge for managing a case: collecting data, reaching and testing hypotheses, and prescribing therapy. Most medical CAI programs are designed to teach cognitive skills. These skills are generally presented in two stages: acquisition of facts (e.g., properties of organisms, typical development of an infection) in preclinical years, and application of this knowledge to solve clinical problems (Hoffer et al., 1975). Most medical CAI programs present specific clinical problems that give the student an opportunity to apply his or her knowledge of facts, while following some diagnostic strategy for collecting data and forming hypotheses.

RULE507

IF: 1) The infection which requires therapy is meningitis,
2) Organisms were not seen on the stain of the culture,
3) The type of the infection is bacterial,
4) The patient does not have a head injury defect, and
5) The age of the patient is between 15 years and 55 years
THEN: The organisms that might be causing the infection are diplococcus-pneumoniae (.75) and neisseria-meningitidis (.74)

FIGURE 11-1 A typical MYCIN rule.

11.2 An Overview of the GUIDON System

The purpose of GUIDON research has been to develop a case method tutorial program that combines knowledge encoded in production rules [rules about infectious disease diagnosis provided by the MYCIN consultation system (Shortliffe, 1976) (see also Chapter 5)] with explicit tutorial discourse knowledge, while keeping the two distinct. GUIDON engages a student in a dialogue about a patient (a case) suspected of having an infection, and helps the student consider the relevant clinical and laboratory data for reaching a hypothesis about the causative organism(s). MYCIN's 450 diagnostic rules, one of which is shown in Figure 11-1, provide the underlying expertise that is used by the tutorial program in selecting topics to be discussed. MYCIN's methods provide a problem-solving approach for understanding the student's behavior and for defining skills to be taught. In addition, GUIDON has 200 tutorial rules, which include methods for guiding the dialogue economically, presenting diagnostic strategies, constructing a student model, and responding to the student's initiative.

A MYCIN rule consists of a set of preconditions (called the *premise*) that, if true, justifies the conclusion made in the *action* part of the rule. Conclusions are modified by *certainty factors* (Shortliffe and Buchanan, 1975), numbers that indicate how certain the rule's author is that the given conclusion is correct when the premise is true.

MYCIN's rules have not been modified for the tutoring application, but they are used in additional ways, for example, for forming quizzes, guiding the dialogue, summarizing evidence, and modeling the student's understanding. Flexible use of the rule set is made possible by the existence of *representational meta-knowledge* (Davis and Buchanan, 1977), which enables a program to take apart rules and reason about the components.

Two formal evaluations of MYCIN's performance have demonstrated that MYCIN's competence in selecting antimicrobial therapy for meningitis and for bacteremia is comparable to that of the infectious disease faculty at Stanford University School of Medicine (where MYCIN was developed)

(Yu et al., 1979a; 1979b). From this we conclude that MYCIN's rules capture a significant part of the knowledge necessary for demonstrably high performance in this domain.

11.3 GUIDON's Capabilities

The literature for medical CAI systems is extensive. Not all of the programs reported have a classic AFO design. For example, some programs use probability tables to generate "cases" (a patient with a specific problem) and use differential diagnosis to analyze the student's response and provide assistance (Entwisle and Entwisle, 1963; Steele et al., 1978). GUIDON is the first medical tutorial program we know of that is based on AI techniques. What contributions does it make to medical CAI? Most researchers address the following set of issues in the setting of GUIDON: (1) the nature of the dialogue interaction (including feedback and realism), (2) pedagogy, and (3) the problem of assembling a variety of cases.

We believe that GUIDON's main contribution lies in its capability to carry on a flexible dialogue with the student, allowing for problem-solving assistance in context, providing feedback for partial solutions at any time, and coping with the student's initiative in choosing topics and detail of discussion. Of secondary interest is the ease with which a library of cases can be assembled with minimal human intervention. Finally, current methods by which GUIDON provides assistance demonstrate that it has the potential for explicitly teaching strategies for doing medical diagnosis and perhaps for detecting which strategy the student is using.

11.3.1 Nature of the Dialogue Interaction

Medical CAI programs vary greatly in the nature of the dialogue that the program has with the student. Relevant issues considered here are

1. the form of input entered by the student,
2. the freedom of the student to direct the dialogue,
3. feedback for partial student solutions,
4. assistance provided for solving the problem, and
5. the realism of the interaction.

Input

Some programs restrict the student to key words or even numerical codes for diagnostic tests (Diamond et al., 1974), and others provide a humanlike interaction (by *ad hoc* means) that would tax the resources of any state-of-

Option type	*Examples*
Get case data	BLOCK, ALLDATA
Information retrieval	PENDING, DETAILS
Dialogue context	RULE, TOPIC
Convey what you know	IKNOW, HYPOTHESIS
Request assistance	HINT, TELLME
Change the topic	DISCUSS, STOP
Special	JUSTIFY, PROFILE

FIGURE 11-2 Some of the 30 options available in GUIDON dialogues.

the-art AI program (Swets and Feurzeig, 1965; Feurzeig et al., 1964). Some programs have borrowed AI techniques, for example, keyword analysis (Harless et al., 1971) and anaphoric resolution (Weber and Hageman, 1972). The main issue here is that it should be easy for the students to express themselves by using constructs that the program will be able to understand. This has been an important concern in ICAI in general. Some of the best results have been achieved by Burton (1976).

GUIDON, like most ICAI programs, accepts student input in the form of simple sentences. However, given the range of initiative we would like to allow (more than just collecting data), we are experimenting with the use of short-form options (Figure 11-2). This has the advantage that input is terse, and there is less chance of entering statements that the program cannot understand. In addition, the student is provided with a hard-copy listing of parameter designations that are recognized by the program, for example, BURNED, ALCOHOLIC, and AGE.

Some researchers believe that providing the student with a list of possible clinical factors to consider is unrealistic (a physician does not carry around a "crib sheet"). It serves as a crutch by cueing the student and, by the organization of the list, artificially distorts the student's approach (Hoffer et al., 1975; Harless et al., 1971). Other researchers claim that the list of relevant factors is helpful to a beginning student who would otherwise not know what questions to ask (Entwisle and Entwisle, 1963). We observed in one GUIDON tutorial that the student went down the list of factors in the hard-copy handbook once he had exhausted his own knowledge. This usage may not be harmful; problems may arise when students use the handbook inappropriately.

Student Initiative

An essential part of tutorial *dialogue management* (Clancey, 1979c) is allowing the student self-expression. In a mixed-initiative program, provision must be made for every potential *kind* of initiative that the student may

**** FACTORS**
The following factors will be useful to determine the type of the infection:
3a. whether the meningitis is partially treated
3b. whether a lumbar puncture has been performed on Pt538
3c. the duration of the neurological signs
3d. the CSF protein
(*)3e. whether Pt538 has a rash consistent with Herpes Zoster

.
.
.

Factors marked with * have already been stated. Based on what we have already mentioned, you should be able to conclude about: 3b.
**** USE 3C**
The duration of the neurological signs is 7.2 hours. Using other information available to you now, the duration of the neurological signs enables you to conclude that the type of the infection is bacterial (.2) viral (.2) fungal (-.4) tb (-.4) [RULE524].
**** IKNOW 3B**
Good. MYCIN has also made a decision.
The site of CULTURE-1 is evidence that a lumbar puncture has been performed on Pt538 [RULE112].
Back to our discussion of the type of the infection . . . .

.
.
.

Summary of evidence that the type of the infection is bacterial (.98) viral (-.76) fungal (-.83) tb (-.83):
29a. The fact that the meningitis is partially treated (.95) is evidence that the type of the infection is bacterial (.19) [RULE148]
29b. The WBC from the CSF and the percent of PMNs in the CSF is evidence that the type of the infection is bacterial (.9) viral (-.6) fungal (-.7) tb (-.7) [RULE501]

.
.
.

**** DETAILS 29A**
For which of the following factors would you like to see a review of the evidence? (select from the following):
1) the infection which requires therapy
2) the fact that the meningitis is partially treated (.95)
**** 2**
Summary of evidence that the meningitis is partially treated (.95):
32a. The infection which requires therapy, the fact that organisms were not seen on the stain of the pending CSF culture and the time since therapy with the cephalothin was started is evidence that the meningitis is partially treated (.95) [RULE145]
Do you want to see RULE148?
**** NO**

.
.
.

FIGURE 11-3 Sample use of options.

want to make. This includes being able to refer back to an earlier topic and provide more details, allowing the student to change the topic, and so on. We might summarize this by saying that we must allow the student to specify what he or she knows, wants to know more about, and wants to ignore. Figure 11-3 illustrates GUIDON's flexibility in responding to a student's initiative. Notice that tutorial remarks are indexed so that the student can easily refer to them later (by using them as arguments to options).

We allow the student to explore the reasoning of the underlying expert program, but we do not want the tutor to be simply a passive information retrieval system. In addition to laying out data and inferences clearly, the tutor has to reason about what constitutes reasonable, expected elaboration on the basis of what has been previously discussed. In the excerpt shown in Figure 11-3, GUIDON provided details for an inference (RULE148) by offering to support necessary preconditions that were not considered in the dialogue up to this point, though they could be inferred from known data.

Similarly, when the student takes the initiative by saying he or she knows something (see Figure 11-3), the tutor needs to determine what response makes sense, based on what it knows about the student's knowledge and shared goals for the tutorial session. The tutor may want to hold a detailed response in abeyance, simply acknowledge the student's remark, or probe for a proof. Selection among these *alternative dialogues* might require determining what the student could have inferred from previous interactions and the current situation. In the excerpt shown here, GUIDON decides that there is sufficient evidence that the student knows the solution to a relevant subproblem, so detailed discussion and probing are not necessary.

In many AFO systems, the flow of the dialogue is permanently fixed by the author of the course material. The student cannot change topics as he or she might wish, discussing subproblems and offering hypotheses to be evaluated. Systems like ATS (Weber and Hageman, 1972) have limited ability to reason with author-provided material (by indexing material with keywords), but it is still necessary for a course author to "sit down and play the role of the student for each major step in his tutorial." Thus it is still necessary to anticipate possible contingencies in each case individually.

Decoupling domain expertise from the dialogue program, an approach used by all ICAI systems, is a powerful way to provide flexible dialogue interaction. In GUIDON, *discourse procedures* (Clancey, 1979a) formalize how the program should behave in general terms, not in terms of the data and outcome of a particular case. A discourse procedure is a sequence of actions to be followed under conditions determined by the complexity of the material, the student's understanding of the material, and tutoring goals for the session. Each option available to the student generally has a discourse procedure associated with it. These procedures invoke other procedures for carrying on the dialogue, depending on circumstances of the particular situation.

For example, the procedure for the IKNOW option invokes the procedure for requesting and evaluating a student's hypothesis if the expert program has not yet made a final decision (so the tutor does not believe that the student can know the result). Otherwise, if the expert program has a final result, the procedure for discussing a completed topic is followed. Whether or not the student will be probed for details will depend

T-RULE5.02 [Directly state single, known rule]

IF: 1) There are rules having a bearing on this goal that have succeeded and have not been discussed, and
2) The number of rules having a bearing on this goal that have succeeded is 1, and
3) There is strong evidence that the student has applied this rule

THEN: Simply state the rule and its conclusion

FIGURE 11-4 T-rule for deciding how to complete discussion of a topic.

on the model that the tutor is building of the student's understanding (considered below).

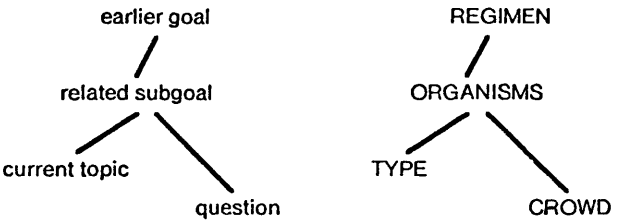
Conditional actions in discourse procedures are expressed as tutoring rules. Figure 11-4 shows the tutoring rule that caused GUIDON to acknowledge the student's statement about what he or she knew, rather than to ask for details.

As a final example of the problem of providing for and coping with the student's preferences, we will briefly consider the problem of focusing on topics during the dialogue. GUIDON allows a student to explicitly change the topic by using the DISCUSS option. However, student requests for data can also (implicitly) change the topic if the datum requested is not relevant to the current topic (cannot be used directly in any inference). In this respect, GUIDON enforces a goal-directed dialogue, so it will tell the student when he or she appears to be changing the topic. For example, if requested information is relevant to a previous, shallower subgoal (in the tree of topics by which the expert structures the problem solution), the tutor states this relation so that it is clear to the student what topic is currently being pursued (Figure 11-5).

Feedback

Nearly every discussion of medical CAI points to the importance of providing feedback to the student—primarily an evaluation of the student's solution, including mention of unnecessary and missed diagnostic questions. Programs vary from providing feedback at the end of the solution (Harless et al., 1971), to a step-by-step report that is inherent in AFO CAI (Feurzeig et al., 1964). Indeed, it is widely believed that the immediate correction of errors is an important capability of CAI (Hoffer et al., 1975). In a more general sense, the feedback that a CAI continuing education program offers provides a valuable tool for experienced physicians to evaluate their practices in light of new techniques (Brandt, 1974).

Providing feedback to the student is one problem that ICAI systems seem directly designed to resolve. A frame-oriented system is inherently unable to deal with unanticipated student errors; this would require that



**DOES THE PATIENT LIVE IN A CROWDED ENVIRONMENT?
Pt538 does not live in a crowded environment.
Whether the patient does live in a crowded environment is not relevant to determining the type of the infection. It is a consideration we can use later when we return to our discussion of the organisms that might be causing the infection.

FIGURE 11-5 Coping with an indirectly relevant question.

the author prepare for all possible contingencies, a combinatorially unattainable goal. Part of the problem is that these programs cannot flexibly articulate diagnostic reasoning, however realistic or humanlike the responses may appear. That is, they cannot tolerate a random interruption of the dialogue in which the student poses a partial solution to the problem and asks for a response. Programs based on statistical tables (Kirsch, 1963; Entwisle and Entwisle, 1963; Steele et al., 1978) certainly have more flexibility to evaluate a given student's questions, but they have trouble characterizing the typicality of a case or what are the reasonable, important questions to ask. Of course, probabilistic programs have no way to justify correlations or to provide principles for remembering them.

MYCIN's diagnostic rules make it possible to provide feedback to the student at any time, regardless of the order and completeness of the student's collection of case data. In contrast with the frame-oriented approach, we prepare for contingencies abstractly, for example, what to do *in general* when the student says he or she knows something but the expert program does not know it, as opposed to what to do if the student says he or she believes X after questions Q1 and Q2 have been asked but not Q3.

In order to provide feedback, the tutor must be able to account for the student's behavior in terms of problem-solving methods and strategies. This may involve interpreting wrong, but consistent, behavior in terms of misconceptions. In correcting the student, the tutor may need to refer back to past student actions and compare alternative approaches. Allowing for elaboration in a mixed-initiative exchange poses additional technical problems in dialogue management, such as maintaining context while each participant varies the topic.

In the excerpt that follows (Figure 11-6), the student's hypothesis for a subproblem is correct, but after analyzing the previous history of interaction with the student, GUIDON is not sure that the student really knows

What is the type of the infection?
** **BACTERIAL**
What facts about this case tell you that the type of the infection is bacterial? (Enter one factor per line.)
** **PMNS**
The percent of PMNs in the CBC is not relevant for determining the type of the infection.
** **PMNS IN THE CSF**
** **CEREBRAL-FLUID GLUCOSE VALUE**
** **PROTEIN OF THE CSF**
**
You're right about the CSF protein, the percent of PMNs in the CSF and the CSF glucose value, however, you should have mentioned the fact that the meningitis is partially treated, the WBC from the CSF, and the fact that the simultaneous blood glucose value is not known.

.
.
.

MYCIN rejected some possibilities you did not mention:

What facts about this case tell you that the type of the infection is not viral?

.
.
.

FIGURE 11-6 Responding to a student hypothesis.

why this hypothesis is correct. Therefore, instead of accepting this answer, GUIDON requests relevant factors to support the hypothesis and probes further to see if the student knows why an alternative hypothesis is discredited.

Note that GUIDON does more than say which student answers are right and which are wrong: the program actually reasons about the expertise that is required to solve the problem. Referring to the rules that MYCIN uses to solve subproblems (such as determining whether a meningitis infection is bacterial, fungal, viral, or tuberculous), GUIDON decides which of these rules, if any, are known by the student. That is, what inference chains are consistent with the student's behavior?

As the student inquires about the patient and receives more case data, this same information is given, piece by piece, to the MYCIN program. MYCIN applies rules, as far as it can, to carry its reasoning about the case forward. Thus, at any one time, some of the rules MYCIN follows for determining, say, the type of the infection, have made a conclusion, while others will require more information about the patient before they can be applied. Then, when the student offers an hypothesis, GUIDON compares the student's partial solution to MYCIN's, and, referring to what is known about the student's level of expertise and records from past interactions with him or her, the program attempts to explain the student's solution in terms of MYCIN rules. This analysis is complicated by the fact that a particular hypothesis about the problem may be indicated by more than one rule or that negative evidence may outweigh positive evidence.

A student model constructed in this way is called an *overlay model* because the tutorial program attempts to explain the student's behavior solely in terms of a fixed, single set of "expert knowledge," in this case, MYCIN's

** HINT
Hint: Look for strong, positive evidence that Diplococcus-pneumoniae is one of the organisms that might be causing the infection.

.
.
.

** HELP
Let's consider whether J.Smith has a head injury.

We already know that the patient has not had an injury to the central nervous system; this is evidence that he does not have a head injury [RULE509].

We now have strong evidence that Diplococcus-pneumoniae and Neisseria-meningitidis are organisms that might be causing the infection (considering the age of J.Smith and the fact that he does not have a head injury [RULE507]).

It remains for us to consider other factors for determining the organisms that might be causing the infection.

.
.
.

FIGURE 11-7 Providing assistance in context.

rules. Overlay models were first used by Burton and Brown (1976); the technique was elaborated further by Carr and Goldstein (1977). Limitations of this approach are considered in Section 11.4.

Assistance

Another basic property of a tutorial dialogue is the extent to which the program is able to provide assistance for solving the problem. Ideally, the tutor's guidance should be based on the student's partial solution. In general, this is a difficult problem because it requires that the tutor be sensitive to the student's current problem-solving strategy and the kind of advice he or she prefers (a hint? full details?). It must also be able to articulate problem-solving methods that might be applied (a problem of knowledge representation).

Using its overlay model of the student, GUIDON is able to provide assistance by once again reasoning about the rules that MYCIN has been able to apply at the time that the student requests help. In the example shown here (Figure 11-7), GUIDON provides assistance by applying a solution method (RULE507) that suggests evidence contrary to that which has been discussed to this point of the dialogue. In this case the selected method was alluded to in an earlier hint.

The program has many ways to present a rule to the student, such as forming a question or discussing each clause of the rule explicitly. Here GUIDON demonstrates the applicability of the solution method by showing how the truth of the single precondition that remains to be considered can be inferred from known evidence (RULE509). The inference is trivial, so it is given directly rather than opened up for discussion. GUIDON then

applies the original method (RULE507) and comments about the status of the current subproblem.

Thus providing assistance can involve applying a teaching strategy that carries the solution of the problem forward. This in turn requires being able to articulate reasoning on the basis of what the student knows, according to principles of economical presentation.

Observe that, to provide feedback and assistance, it is not sufficient simply to have a model of what the student knows: the program needs methods for presenting new material to the student. In a knowledge-based tutor, presentations are generated solely from the knowledge base of rules and facts. This requires that the tutor have presentation methods that opportunistically *adapt material to the needs of the dialogue.* In particular, the tutor has to be sensitive to how a tutorial dialogue fits together, including what kinds of interruptions and probing are reasonable and expected in this kind of discourse. GUIDON demonstrates its sensitivity to these concerns when it corrects the student before quizzing him or her about "missing hypotheses," chooses between terse and lengthy discussions of inferences, follows up on previous hints, and comments on the status of a subproblem after an inference has been discussed ("other factors remain to be considered . . .").

Realism of Course Material

Implicit in the design of most medical CAI programs is the assumption that similarity of the tutorial problem-solving environment to actual conditions in actual practice (e.g., the timing and sequence of events, interactions with assistants) is important to ensure transferability of learning to the clinical setting. Furthermore, when the purpose of the tutorial is to make the student familiar with his or her responsibilities on the ward, realism is an integral part of the course material.

Some medical CAI systems attempt to present the student with a "simulated patient" who can be interviewed and given therapy (Harless et al., 1971). Others place the student in a simulated hospital setting in which the student, as attending physician, orders tests, comes back "the next day" to reevaluate the patient, etc. (Feurzeig et al., 1964). The majority of programs, like GUIDON, simulate the kind of tutorial discussion that the student might have on the hospital wards with a resident physician or classroom instructor (Diamond et al., 1974; Weber and Hageman, 1972).

Compared to the investigation of discourse, modeling, and pedagogy, the simulation of a particular real-world problem-solving environment has not been a major focus of ICAI research. However, it seems probable that AI research dealing with the importance of knowledge about prototypical problem situations in everyday reasoning will be useful for generating realistic cases to be solved by the student, as well as for simulating moment-by-moment patient events.

11.3.2 Pedagogy

The main pedagogical question in CAI programs concerns what diagnostic strategy, if any, should be conveyed to the student and how this should be done. For example, one program is specifically designed to teach Weed's "problem-oriented approach" (Benbassat and Schiffmann, 1976); it imposes a fixed logical order on the kinds of questions that the student asks. Other researchers believe that a completely uninterrupted, "free-form" style is an essential part of teaching independent thinking and responsible problem solving (Harless et al., 1971).

GUIDON attempts to allow for a free-form style while still conveying problem-solving strategies. The student is free to gather case data in any order, but is told when he or she is wandering from the topic under consideration. Hints and help are based on a problem-solving strategy (Figure 11-7) that could be altered (nontrivially) to reflect Weed's approach.

CAI programs, including ICAI ones, have generally not focused on teaching problem-solving strategies because it is difficult to represent them internally in a way that allows the program to use them for teaching material (e.g., mentioning the strategy when posing a hint based on it) as well as for modeling the student (i.e., knowing that the student is following a particular strategy). Technical problems aside, medical CAI programs have probably focused on teaching facts and decision rules rather than strategies because "there is little agreement among medical educators about an explicit and detailed model of clinical competence" (Hoffer et al., 1975). Only recently have physicians developed scientific descriptions of alternative problem-solving strategies (Kassirer and Gorry, 1978), which, interestingly enough, have been based on AI research.

It is possible that the expert modules of ICAI systems (for example, the role MYCIN plays in the GUIDON program) will provide useful testbeds for formalizing and experimenting with problem-solving strategies. Meta-rules (Davis and Buchanan, 1977) and strategies for revising hypotheses provide a language by which GUIDON can be used to formalize and measure diagnostic competency. AI alone cannot provide the missing physiological, chemical, and physical knowledge that will provide a deeper understanding of medical problems, but AI approaches to search and hypothesis confirmation may provide suitable information-processing models for talking about different approaches to diagnosis.

11.3.3 Case Generation

A major advantage of CAI over other forms of medical instruction is that it has the potential to expose a student to a variety of cases that might far exceed what actual hospital experience would provide. However, to achieve this potential, it has been necessary in traditional medical CAI to spend many days designing and debugging each case. Various estimates are given

for the ratio of design time to course time, and 1 week of design for a 20-minute course is not atypical (Bitzer and Bitzer, 1973). Researchers emphasize the ease with which their frame-oriented systems may be changed, but it must be remembered that each clever addition in one case must be repeated in others, a clearly untenable situation if the science of instruction is to advance. GUIDON offers an improvement over the traditional approach: experience is cumulative, so that modifications made on the basis of one tutorial interaction will automatically show up in similar situations during discussion of any other case.

By coupling GUIDON to the patient library that has been accumulated during the testing of the MYCIN consultation program, formal course preparation is unnecessary. Given that MYCIN can work out the reasoning for solving a case and GUIDON can selectively discuss it with a student, preparation time for a new case is reduced to less than an hour for each hour of course time, allowing for the provision of some annotations that point out the pedagogical value of the case. (Patient cases are entered into the MYCIN system for the purpose of receiving a consultation or for testing the program, so the case library is available to GUIDON at no cost.) Eventually, given case-selection strategies (based on knowledge about the spectrum of cases), even these annotations would be unnecessary, and course preparation time would be eliminated.

11.4 Limitations of the Approach

A potential weakness of the GUIDON program is that it attempts to explain the student's behavior solely in terms of MYCIN's rules. When the student is basing his or her questions and hypotheses on incorrect rules, GUIDON is not able to detect these rules and correct them directly. It is also possible that the student's concepts are different from MYCIN's, so the student's conclusions might be correct, but he or she will want to support them using reasoning that is different from MYCIN's. This could involve something as simple as wanting to refer to the patient's age in general terms (infant, adolescent), while MYCIN only recognizes precise numerical ages.

Representing a hierarchy of common concepts is a relatively straightforward change to MYCIN. However, modeling medical reasoning in terms of an alternative rule set (not just a subset of MYCIN's) is a theory-formation problem that goes beyond the current capabilities of AI. Brown has investigated methods for modeling improper procedures that students follow when doing subtraction (Brown and Burton, 1978). Unfortunately, this technique does not generalize easily to semantically complex domains like medicine. The approach followed by Stevens for meteorology is to collect data about kinds of student misconceptions and then incorporate

**** TOPIC**
We are discussing the degree of reversibility of airway obstruction of PATIENT-7. Hypotheses you should be considering are: none, insignificant, slight, good and significant.

**** WHAT IS THE CHANGE IN FORCED EXPIRATORY VOLUME?**
The change in Forced-Expiratory-Volume (1sec.) after bronchodilation of M.Smith is 31.

**** WHAT IS THE CHANGE IN RESISTANCE?**
The change in Resistance after bronchodilation of M.Smith is 20.

We have substantially completed our discussion of the degree of reversibility of airway obstruction of M.Smith. Would you like to make a hypothesis now (as opposed to gathering more information)?

**** NO**

We will now consider evidence that leads to a conclusion about the degree of reversibility of airway obstruction of M.Smith.

Here is a conclusion that you may not have considered . . . .

The fact that the change in Forced-Expiratory-Volume (1sec.) after bronchodilation of M.Smith and the change in Resistance after bronchodilation are known is evidence that there are post bronchodilation test results [RULE028].

Given the post bronchodilation test results, does this tell you that the degree of reversibility of airway obstruction is slight?

**** YES**
No, given this fact, we can conclude that the degree of reversibility of airway obstruction of M. Smith is significant [RULE020].

The degree of reversibility of airway obstruction would be slight when the change in Forced-Expiratory-Volume (1sec.) after bronchodilation of the patient is between 1 and 5.

FIGURE 11-8 Excerpt from a PUFF tutorial.

these variations in the modeling process (Stevens et al., 1978). We believe that GUIDON tutorials will provide the opportunity for furthering this study.

11.5 Experimentation with Other Domains

Besides being able to use a fixed set of teaching strategies (the discourse procedures) to tutor different cases, GUIDON is able to provide tutorials in any problem area for which a MYCIN-like knowledge base of decision rules and fact tables have been formalized (van Melle, 1979). This affords an important perspective on the generality of the discourse and pedagogical rules. At this time two other medical consultation programs have been developed using MYCIN's rule formalism: PUFF (Kunz et al., 1978) provides diagnoses about pulmonary disease; HEADMED (Heiser and Brooks, 1978) advises about the use of psychopharmaceuticals.

The example shown in Figure 11-8 is taken from a GUIDON tutorial that uses PUFF's knowledge base for the problem of pulmonary function analysis. This example shows the program taking the initiative to present

new information to the student. GUIDON first interrupts the student's data collection to suggest that the student make an hypothesis; but the student does not do so. The program then observes that there is a particular problem-solving method that can be applied and that is probably known to the student (RULE020). However, the student probably cannot apply the method to this case because he or she does not know how to verify a necessary precondition. GUIDON presents the inference that it believes is unknown to the student (RULE028), and then asks him or her to take this evidence forward.

Experimental tutorials with other knowledge bases have revealed that the effectiveness of discourse strategies for carrying on a dialogue economically is determined in part by the depth and breadth of the reasoning tree for solving the problem. When a solution involves many rules at a given level (for example, when there are many rules to determine the organism causing the infection), the tutor and student will not have time to discuss each rule in the same degree of detail. Similarly, when inference chains are long, then an effective discourse strategy will entail summarizing evidence on a high level, rather than considering each subgoal in the chain.

11.6 Conclusions

In traditional medical CAI, as well as in some ICAI programs, teaching expertise is "compiled" into the program, combining all kinds of problem-solving, communication, and pedagogical strategies. In GUIDON we make the important step of explicitly codifying teaching expertise within the program as a body of rules to follow in various situations. In fact, the rules *are* the program. By decoupling medical expertise from dialogue strategies, we are able to focus more directly on rules of conversation and communication or "kibitzing" strategies (Burton, 1979). This is one of the special advantages of GUIDON's framework of discourse knowledge. GUIDON's tutoring rules never mention cultures or disease or any application area. Instead, the rules state how to teach, how to reply to a student, and how to guide a student. With these explicit principles before us, we are in a much better position to say what we are evaluating when we test the program.

The key to GUIDON's contributions lies in the flexibility of its representation of teaching and problem-solving knowledge. MYCIN's domain rules can be reasoned about to construct a student model, to provide assistance, and to select presentation methods. GUIDON's tutoring rules, wholly separated from the domain rules, constitute general procedures that can be followed any time in a dialogue, giving the program the capability to cope with arbitrary student initiative within the considerable range of expression the program's options allow. Finally, these tutoring

rules are problem- and domain-independent, allowing flexibility for teaching any case formalized in a MYCIN-like consultation system.

With respect to the issues of dialogue interaction, pedagogy, and case generation, GUIDON's primary contributions to medical CAI are greater individualization of tutorials, a framework for expressing and accumulating tutorial dialogue expertise, and a language for diagnostic problem-solving strategies. By constructing a model of problem-solving strategies in a student model, something not possible with traditional technology, ICAI systems could provide a basis for critiquing and teaching diagnosis in terms that even go beyond classroom or clinical experience.

ACKNOWLEDGMENTS

This research was supported in part by an ONR contract (N00014-79C-0302) and a National Library of Medicine career development award (LM00048) to E. H. Shortliffe.

12

LCS: The Role and Development of Medical Knowledge in Diagnostic Expertise

Paul J. Feltovich, Paul E. Johnson, James H. Moller, and David B. Swanson

After a visit by Herbert Simon to the University of Minnesota in 1972, Paul Johnson (then a professor of educational psychology at the Center for Research in Human Learning) and associated graduate students began applying information-processing concepts to the study of expertise and problem solving. This investigation was consistent with their view that psychology is the study of contextually dependent phenomena. That is, the psychology of human behavior is most fully understood in domains of use and practice.

Johnson then met James Moller (a professor of pediatrics) who had similar interests in problem solving within medicine and medical education, and the collaboration started. David Swanson was Johnson's graduate student and wrote a simulation program called DIAGNOSER as part of his Ph.D. dissertation. Paul Feltovich also studied with Johnson, and this chapter reports on his dissertation research, a formal psychological study. The development of DIAGNOSER and the design of Feltovich's study took place in tandem, and each contributed to the other, although the simulation was completed first. The whole group at Minnesota, over this period of time, evolved a conception of expertise in terms of the organization and

This chapter is based on the doctoral dissertation of Paul J. Feltovich, which was submitted to the graduate school of the University of Minnesota under the advisorship of Paul E. Johnson. The first version of this paper was presented at the annual meeting of the American Educational Research Association, Boston, 1980. Used with permission.

manipulation of knowledge and the adaptation of inner environment (knowledge and reasoning) to task environments.

The roots of Feltovich's study are interesting and illustrate the changing nature of psychological investigations over the past decade. The major empirical studies of clinical expertise (Elstein et al., 1978; Barrows et al., 1978) had focused on the process of clinical reasoning and found no differences between experts and novices. At the same time, psychological studies of expertise [e.g., Chase and Simon (1973)] had also found no differences at process levels (e.g., number of moves considered, depth of search). They were pointing to elements of the quality of reasoning and knowledge as the main contributors to expertise. The work by Barrows et al. in medicine also cited quality of reasoning as the only discriminator they could find. This previous work, in conjunction with the Minnesota group's view of expertise as the adaption of a knowledge base to a task environment, led Feltovich, Johnson, and Swanson to study the organization and representation of knowledge in medicine, focusing on the determinants of quality. In this pursuit they were influenced by related AI work in knowledge representation, including early writings about frames [e.g., Minsky (1975)] and collections [e.g., Bobrow and Collins (1975)].

Thus, in sharp contrast with traditional psychological studies, Feltovich and his colleagues attempted to ferret out how the structure of an individual's knowledge affects his or her problem solving. This level of analysis asks how particular hypotheses *come to mind, not just how many hypotheses are considered at once or how soon the first one is vocalized. The experiments reported here are of considerable value as scientific support for the structuring schemes that have been derived more intuitively by AI researchers. These include schemes for articulating strategies and principles in program explanations (Chapters 11 and 16) and factoring a knowledge base into "specialists" (Chapter 13). Such an analysis also provides a basis for eliciting knowledge from an expert and for teaching students (Chapter 15).*

The reported study investigates the contribution of case-related medical knowledge to clinical diagnosis. Subjects, varying in their training and clinical experience in pediatric cardiology, diagnosed four cases of congenital heart disease while thinking aloud. Each case was designed to assess a different aspect of the subjects' medical knowledge. Consistent differences in performance among diagnosticians at different levels of experience were found, and inferences were made to sources of medical knowledge responsible for performance. Recurrent sources of error were identified for the less experienced diagnosticians.

Unlike the other chapters in this volume, this chapter does not report on a working computer program. In a narrow sense, this is not a report of medical AI research. However, the contribution to AIM research is evident in the kinds of questions asked and in the form of the model of reasoning. In this respect Feltovich's work is distinguished in the depth and controlled nature of his investigation. Moreover, research that followed (Johnson et

al., 1981) made good use of the DIAGNOSER simulation model for testing and experimenting with conjectures about knowledge structures and reasoning.

The approach taken by Feltovich and colleagues in this study continues to evolve. Besides seeking generality in diverse areas such as law and physics, they are investigating the implications of their findings for the assessment of clinical competence and expertise, as well as the implications for teaching basic science for clinical problem solving.

12.1 Overview: Studies on the Nature of Knowledge and Reasoning

Knowledge influences reasoning and other cognitive skills. In recent years distinctions between knowledge and reasoning have blurred. That knowledge influences the quality and nature of reasoning that can occur has been suggested. That reasoning uses knowledge as a substrate is evident, and even the idea that reasoning constitutes a form of knowledge has been entertained.

Recent laboratory research has indicated that knowledge contributes to even the most fundamental cognitive skills. The knowledge base possessed by an individual has been shown to influence fundamental intellectual skills such as induction and analogy (Glaser and Pellegrino, 1980), basic memory mechanisms such as grouping and rehearsal (Chi, 1978), and even the functional size of short-term memory (Chi, 1976). Voss and his colleagues (Chiese et al., 1979; Spilich et al., 1979) have extended work of this sort beyond basic laboratory environments into domains of complex subject-matter learning. Within a given subject matter, high-knowledge individuals have greater recognition and recall of new material than do low-knowledge individuals, can make useful inferences from smaller amounts of partial information, and are better able to integrate new material within a coherent and interconnected framework of knowledge (organized, for example, around a common goal structure).

Reasoning itself has been shown to be highly dependent on the individual's knowledge base for the task environment in which the reasoning occurs. Subjects show dramatic improvement in testing the implications of logical inference rules (e.g., if p then q) when these are couched in terms of a familiar setting, as opposed to when the expression is stated in a more purely symbolic form (Rumelhart, 1979; Wason and Johnson-Laird, 1972). This content-constrained conception of formal reasoning is in contrast to structural developmental theories (Piaget, 1972) that claim cross-situational, content-free, and maturationally determined general reasoning skills. Yet even within these theories, evidence is emerging for the import of accumulated knowledge as a contributor to these abilities (Carey, 1973).

Artificial intelligence research has also shown an evolution from systems in which knowledge (declarative) and reasoning (procedures) were clearly separated to systems in which these components strongly interact. Early systems such as Green's QA3 (Green, 1969) and Quillian's TLC (Quillian, 1969) relied on data bases of uniformly formatted declarative knowledge and a few general-purpose reasoning algorithms for operating on these knowledge bases. These systems have given way to ones in which the separation between knowledge and reasoning components is much less distinct and in which general reasoning algorithms have considerably less status in comparison to specific (local) reasoning strategies associated with specific domains of knowledge (Norman et al., 1975; Sacerdoti, 1977; vanLehn and Brown, 1979). Reasoning is seen not so much as a general but as a task-specific skill.

The role of knowledge and its organization have been emphasized in recent work on expertise and expert/novice differences in problem solving in complex domains. The findings of groupings in expert perception of a chess board is taken as evidence that guidance in the choice of chess moves is provided by knowledge representations for configurations in the board (Chase and Simon, 1973). Similarly, Larkin (1978) has proposed a construct of "chunked procedures" for expert physics problem solvers, whereby expert categorization of a problem leads to a relatively integrated problem plan and associated "bursts" of equations applied in solution. Feltovich and colleagues have shown that differences in problem-solving processes among expert and novice physics problem solvers result both from differences in the structure of knowledge representations for problem types and from differences in memory organization among these types (Chi et al., 1981). Simon and colleagues (Hinsley et al., 1978; Paige and Simon, 1966) have shown that schemata, which are knowledge structures representing problem types, strongly influence the nature of the problem-solving process in algebra.

In light of developments such as those outlined in this section, Greeno (1979) has proposed that knowledge and its effects on problem solving constitute a relatively neglected and important direction for research. Others have turned attention to the problem of how knowledge bases change and develop with experience so as to become better suited to problem-solving demands (Anderson et al., 1979; Lenat et al., 1979; Rumelhart, 1979; Rumelhart and Norman, 1977). Among implications from this work important to the present study are that knowledge bases change in the directions of: (1) accretion or, simply, augmentation of knowledge, (2) knowledge reorganization, and (3) changes and refinements in the condition tests by which knowledge is judged applicable to situations.

The present study investigates the effects of medical knowledge on the clinical reasoning process and the changes in such knowledge as individuals gain experience with the task of medical diagnosis and with the subject matter of a subspecialty of medicine.

12.2 Introduction: Clinical Diagnostic Reasoning and Expert/Novice Studies

Recent research in clinical diagnosis (Barrows et al., 1978; Elstein et al., 1978; McGuire and Bashook, 1978) contributed to a consensus about the *general* form of the process of clinical diagnostic reasoning. Cues in patient data suggest hypotheses, which are, in turn, tested against subsequent data of the case. The basic hypothetico-deductive process is shared by experienced and inexperienced diagnosticians alike, as are numerous parametric characteristics of the process, such as the percentage of data items to first hypotheses, the average number of hypotheses maintained in active consideration, etc.

These studies, however, have generally neglected the content of diagnostic reasoning, that is, the knowledge base of medical subject matter involved in the diagnostic process. Yet, despite prevalent findings of lack of differences in the form of diagnostic reasoning as a function of experience, the few differential findings from these research efforts implicate the importance of the knowledge base. The Michigan State group (Elstein et al., 1978) found that expert and less expert physicians differ in the "accuracy of interpretation" of patient data with respect to the hypotheses they consider, a finding that shows the importance of the physician's knowledge of patient data that present in particular diseases. Barrows's group (Barrows et al., 1978) found that experience can be discriminated by the *actual* hypotheses (as opposed to the number of hypotheses) that physicians use. This suggests that experienced and less experienced physicians differ in their knowledge store of diseases or the conditions by which they judge that particular diseases are likely to apply to a case. The same projects have also confirmed the case-specificity of skill in diagnostic reasoning. The same physician may show different profiles of competence depending on his or her particular experiential history with different types of cases, a further indication that clinical reasoning is not a general skill, but rather a process that is strongly dependent on the contents of knowledge to which it is applied.

Research at the University of Minnesota has concentrated on diagnostic problem solving in the medical subspecialty of pediatric cardiology and has resulted in a theory of diagnosis in this field that attempts to explicate the knowledge and knowledge organization necessary for expert diagnostic performance (Johnson et al., 1979b). Extensive experimentation and consultation with an expert pediatric cardiologist has resulted in a computer-runnable instantiation of the theory for this expert that represents knowledge explicitly and shows strong correspondence to the subject's performance over a broad range of cases (Swanson, 1978; Swanson et al., 1979).

Within the constructs of this theory, the present experimental study investigates the development of the knowledge base, as exemplified by individuals with different levels of experience with pediatric cardiology, and the implications of developmental differences for diagnostic performance. The particular theoretical construct of focus is prototype, or disease knowledge (Johnson et al., 1979b).[1] Disease knowledge refers to a memory store of disease models, each of which specifies, for a particular disease, the pathophysiology of the disease and the set of clinical manifestations that a patient with the disease should present [see also Rubin, (1975), disease "templates," and Pople (1977), "disease entities"]. In the theory of the expert, this set of disease models is extensive [see also deGroot (1965) and Simon and Chase (1973)] and organized hierarchically [see also Wortman (1972) and Pople (1977)]. At upper (more general) levels of the hierarchy are *disease categories*, sets of diseases that present similarly because of physiologic or clinical similarity. Particular diseases occupy middle ranks of the hierarchy, and these, in turn, are differentiated at the lowest hierarchical levels into numerous variants of each disease, each of which presents slightly differently in the clinic for reasons of subtle underlying difference in anatomy, physiology, severity, or age of presentation in a patient.

Speculations about characteristics of novices' disease knowledge can be garnered from analysis of the training experiences that novices encounter, the training materials they use, as well as psychological theory pertaining more generally to the development of knowledge bases. The first postulate for the novice's knowledge base of diseases is that it is *classically centered.* Initial training materials (Moller, 1978), as well as the probability distribution of diseases presenting in the hospital, accentuate the most common versions of diseases that constitute "anchorage points" for subsequent elaboration of the store of diseases [see also Rosch et al. (1976), "basic objects"]. A second postulate for novices is that the disease store is *sparse* in the sense that it lacks extensive cross referencing and connection among the diseases in memory (Elstein et al., 1971; Shavelson, 1972; Thro, 1978). It is with experience that the starting-point set of diseases is augmented and both generalized into categorical clusters, as similarities among diseases are discovered, and discriminated into finer distinct entities, as differentiation points among and within diseases are learned (Reed, 1978; Wortman and Greenberg, 1971). A third postulate about novice disease knowledge refers to the internal structure of the disease models themselves; this involves *imprecision* in the clinical expectations associated with diseases. Given that there is a range of natural variability associated with the clinical findings that can occur with any disease, large sampling,

[1]The term *disease knowledge* will be used in the present paper instead of the term *prototype knowledge.* It was decided to abandon the latter designation because of its suggestion of entities particularly typical of a class (Rosch, 1975). While some disease models are prototypic, not all of them are.

through clinical experience or other training devices, is probably necessary to "tune" (Rumelhart and Norman, 1977; Anderson et al., 1979) clinical expectations in disease models to the naturally occurring range. Novice expectations may be either overly general, allowing clinical findings that should not occur, or overly specific, not allowing the legitimate range.

In contrast to the novice, whose disease store is assumed sparse, imprecise, and classical, the expert's knowledge base of disease models, by converse arguments as well as by our prior research findings, is assumed *dense*, *precise*, and *penumbral*. The device for studying these claims in the present study is the careful selection of naturally occurring patient cases, each of which, through the structure of patient data it contains, provides a focused test of a different aspect of disease knowledge. In a laboratory setting, these cases were diagnosed by subjects at different levels of experience with pediatric cardiology.

12.3 Method

12.3.1 Materials

Stimulus materials for the study were sets of patient data, each representing a different patient case, extracted from medical records of clinical cases seen at the University of Minnesota Hospitals. Clinical and laboratory findings from the medical record for each case were assembled in a typed "patient file." The file arranged these data in the typical clinical order of history findings, followed by those from physical examination, x-ray, and electrocardiogram (EKG).

Four cases were used, each of which was chosen to assess a different characteristic of subjects' disease knowledge, for example, the differentiation of a disease into subtypes. In addition, the case design employed a "garden path" methodology; some chosen cases showed early strong cues for erroneous diseases but had later critical, disconfirmatory evidence for these same diseases. This device had two functions. First, it brought all subjects to a common starting point in their thinking about possible explanations for the case. Second, because the true diseases were physiologically and clinically similar to the initially induced diseases, it provided a test of the precision in a subject's model of the initial disease (if it was to be rejected), and it established an environment for assessing the diseases that subjects considered as plausible competitors to the original disease. Hence the "garden path" is a means for studying subjects' "conceptual competitor" sets (Elstein et al., 1971).

Case 1. The operative (true) disease in this case is subvalvular aortic stenosis, an uncommon variant of aortic stenosis, the "classic" or most com-

mon version of which is valvular aortic stenosis. The case was meant to assess subjects' differentiation of diseases into subtypes and the precision in their models of the classical variant.

Case 2. The operative disease in this case is total anomalous pulmonary venous connection (TAPVC). The case contains classic auscultatory findings for atrial septal defect (other findings are discrepant), a highly common congenital heart disease, findings that are also perfectly consistent with TAPVC, and, in fact, also consistent with any disease in the category of diseases with volume overload in the right side of the heart (including, in addition to the diseases mentioned, partial anomalous pulmonary venous connection and some forms of endocardial cushion defect). The case was designed to assess subjects' knowledge of and use of disease clusters corresponding to disease categories.

Case 3. This case is a straightforward presentation of the operative disease, patent ductus arteriosus, a highly common congenital heart disease. The case was intended to assess the relationships of this disease to other similar diseases within a subject's disease knowledge and the diagnostic use of these related diseases in a case where the correct diagnosis seems clear.

Case 4. The operative disease in this case is pulmonary atresia, one of a group of physiologically similar diseases (including, in addition, tricuspid atresia and Ebstein's malformation) that constitute a category of "cyanotic diseases with decreased pulmonary blood flow." Like Case 2, this case was designed to assess subjects' knowledge and use of disease clusters corresponding to categories.

12.3.2 Subjects

Subjects were 12 individuals from the University of Minnesota Medical School and were chosen to span a dimension of training and clinical experience in the diagnosis and management of congenital heart disease. Except for faculty experts, so few subjects existed at the prespecified experience levels that the subjects chosen comprised nearly all of them. There were four subjects from each of the following three groups:

- *Students.* These were fourth-year medical students who had just completed a six-week course in pediatric cardiology. As part of this training, each had held primary responsibility for diagnosis and management of 25–30 patients with congenital heart disease.

- *Trainees.* Subjects in this group were either in the third year of a general pediatrics residency or were beginning their first year of fellowship in pediatric cardiology. Subjects in this group estimated that they had held primary responsibility for about 150 patients with congenital heart disease. Residents and fellows did not differ in their estimates.
- *Experts.* This group was composed of two faculty members in the division of pediatric cardiology with upwards of 20 years of active practice as pediatric cardiologists and two fourth-year fellows in pediatric cardiology, one of whom was board-certified at the time of the study. The two fellows estimated that they had held primary responsibility for about 400 patients with congenital heart disease. The best estimates the faculty subjects could give were somewhere between 5,000 and 10,000. The experience discrepancy in this group enabled assessment of the effects of very long-term experience in the faculty members.

12.3.3 Procedure

Each subject diagnosed all four cases, and every subject diagnosed the cases in the same order. The subject was presented the patient file for each case and was instructed to read aloud each numbered data segment in the order in which data were given in the file.[2] The subject could review findings but could not skip ahead. The subject was instructed to report *aloud* any thoughts he or she had at any time toward formulating a diagnosis for the patient's condition. At four points in the case, after history, physical examination, x-ray, and EKG, the subject was asked for an explicit reporting of any "hunches" he or she might have about the patient's condition. After EKG, the subject was also asked for a primary diagnosis and as many as two alternatives.

12.3.4 Data and Analyses

Basic data from the study were typed transcriptions (protocols) of tape recordings made while subjects diagnosed the cases and reported aloud their thinking toward a diagnosis for each. Particular analyses of these data vary somewhat according to the objectives of each case. In general, analyses are organized according to a concept of *logical competitor sets* (LCS), which are sets of diseases targeted as important from the choice of cases for the study (see Section 12.3.1). Diseases in the competitor set for each case share

[2]Order and content of patient findings presented to subjects were fixed in order to compare inferences, interpretations, and evaluations by subjects in a uniform "stimulus" environment. While this deemphasized some components of the diagnostic process, primarily those associated with data collection (e.g., strategy) and first-order interpretations of patient data (e.g., reading x-rays), "fixing" of the input was important to the control needed to investigate the knowledge-based issues of interest in the study.

major underlying physiology with the operative or true disease in the case and hence have similar clinical presentation.

In concentrating analyses on the logical competitor set for each case, a commitment was made to focus analyses on diseases specified in advance to be plausible but easily confused alternatives for the case. Hence they constitute a set of *good* hypotheses to be considered in a case. There were two major motivations for concentrating on LCS diseases. One motivation comes from prior work on expert/novice differences, which suggests that unless a dimension of *quality* is built into the "dependent variables" measured, expert/novice differences are not likely to be revealed (Chase and Simon, 1973; Barrows et al., 1978). The second motivation was the case design itself (see Section 12.3.1).[3] It was assumed that the structure contained within the cases (e.g., garden-path notions) would greatly control and delimit subjects' performance so that the important dynamics of each case would center around the prespecified hypotheses (the LCS) and their management. (This turned out not always to be true for some subjects/cases—as will be noted.)

The LCS for each case was developed from two major sources. First, for the operative disease in each case, an expert in pediatric cardiology and collaborator on the project (the third author, a faculty member in pediatric cardiology at the University of Minnesota) was asked to specify the set of alternative diseases most similar to the true disease and likely to be confused with it. Because these are diseases that are highly similar in clinical presentation, he was also asked to specify the items of patient data that, if interpreted correctly, could be used to discriminate among diseases in the LCS. These judgments were then cross-checked against a major disease reference for pediatric cardiology (Moss et al., 1977). Specifically, for each disease described in this reference, the authors provide a "differential diagnosis" section that discusses diseases similar to and difficult to discriminate from the target disease, as well as differential data points. Based on the reference, no diseases were deleted from the consultant's list, although some were added.

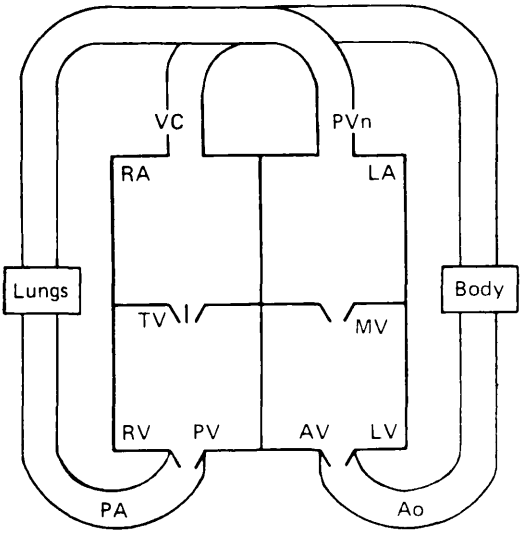
For each case, protocols were coded for two general kinds of uses of the logical competitor set. The first of these is the use of LCS members as hypotheses by subjects at each patient data point of the case. To the extent LCS members are used together, this is taken as evidence that these diseases are being used as competitors and are clustered in memory. The second is the evaluations of LCS members with respect to a set of selected data items. These evaluations yield evidence of the precision in subjects' individual disease models, and also can be used to discern characteristic kinds of errors among the subjects and the loci of these errors in disease knowledge.

---

[3]"Design" was through selection and not construction. Cases in the study are naturally occurring clinical cases and should not be considered oddities. According to the logic of the study, most cases, say, of TAPVC will have atrial septal defect as a naturally occurring garden-path foil.

12.4 Results

In this section, the results from the study will be presented in a case-by-case manner. The presentation of results from each case will follow the same general format. First, there is an introduction to each case that discusses the knowledge-based issue of interest and introduces the operative disease, its logical competitors, and key data points of the case. Since these discussions of congenital heart diseases refer to abnormal modifications to the normal heart and cardiovascular system, a depiction of the normal cardiovascular system is given for comparison as Figure 12-1. After the case discussion, two kinds of results are presented for each case. The first involves the use by subjects of LCS members as hypotheses during the course of the case. The second addresses diagnostic errors and their possible loci in disease knowledge.



LEGEND

Ao	= Aorta	PV	= Pulmonary Valve
AV	= Aortic Valve	PVn	= Pulmonary Veins
LA	= Left Atrium	RA	= Right Atrium
LV	= Left Ventricle	RV	= Right Ventricle
MV	= Mitral Valve	TV	= Tricuspid Valve
PA	= Pulmonary Artery	VC	= Vena Cavae

FIGURE 12-1 The normal heart and cardiovascular system.

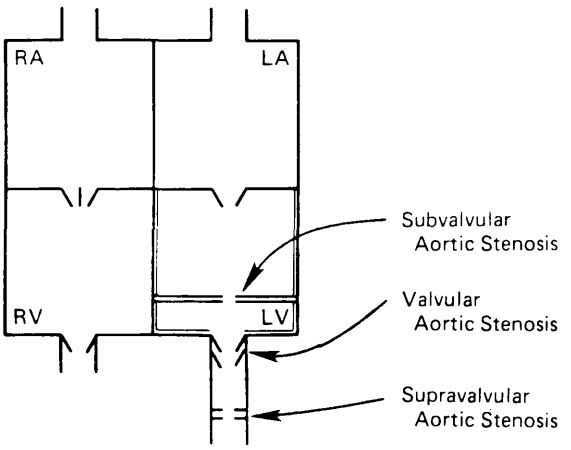


FIGURE 12-2 Logical competitor set for Case 1: three types of aortic stenosis.

12.4.1 Case 1: Subvalvular Aortic Stenosis

The purpose of this case is to investigate subjects' differentiation of a disease into subtypes. The vehicle for doing this is a diagnostic problem that encourages subjects to display, in a diagnostic setting, their working knowledge of a set of disease variants.

The logical competitor set for Case 1 includes three variants of aortic stenosis: valvular aortic stenosis (ValvAS), subvalvular aortic stenosis (SubAS), and supravalvular aortic stenosis (SupAS). Figure 12-2 depicts the anatomical abnormalities within the heart that define each of these disease variants. All involve obstruction to left ventricular outflow with different variants defined by slight differences in the locus of obstruction: ValvAS is obstruction at the aortic valve itself; SubAS is an obstruction slightly "upstream" from the valve; SupAS is obstruction slightly "downstream" from the valve. Because these disease variants are only subtly different anatomically and physiologically, they differ only slightly in clinical presentation. ValvAS is the most common of the three and receives the greatest amount of exposition in introductory training materials of pediatric cardiology (Moller, 1978). Hence it might be expected that subjects' knowledge for ValvAS would develop more rapidly than for the others and that ValvAS may function as a "foil" for some subjects. SubAS, however, is the operative disease in the case and the correct diagnosis.

In the patient file presented to subjects for Case 1, patient data items 17 and 19, a "thrill" and a "murmur" respectively, are strong cues for valvular aortic stenosis, although they are compatible with other variants. Hence it was suspected that all subjects would raise at least ValvAS as a

hypothesis by the time of these data points. Data item 18, a finding of "no systolic ejection click," is very strong evidence against ValvAS. Data items 10, "normal facies," and 22, "prominent aorta," are evidence against SupAS. All data of the case are compatible with the operative disease, SubAS.

Use of the Logical Competitor Set in Case 1

Table 12-1 shows the variants of aortic stenosis that were used as hypotheses by individual subjects at all patient data points where any variant was mentioned by any subject and at the four points of the case where "hunches" were actively solicited from the subjects.[4] Numbers representing data from the patient file are listed across the top in the left-to-right order in which they were presented to subjects. An X in this table simply indicates that the subject mentioned a particular aortic stenosis variant in the protocol at the data point where the X appears.

Table 12-1 shows an increase in the use of variants of aortic stenosis, other than ValvAS, from medical students to experts in pediatric cardiology. In particular, only one student (S2), ever raised both of the less classic variants of aortic stenosis at all, during the entire course of the case, and he mentioned SubAS and SupAS only once each. Two trainees (T1, T3) and three experts (E1, E3, E4) used all three variants at some time during the case. If one considers the number of subjects in each group who not only used all three variants but used each more than once, no students, one trainee (T1), and, again, three experts meet this criterion.

While simple mention (as reflected in Table 12-1) of the aortic stenosis variants as hypotheses is one indication of whether these were considered by subjects, a measure of how *actively* these hypotheses were considered is the prevalence with which they were evaluated with respect to data items. Table 12-2 shows all evaluations by subjects of the aortic stenosis variants with respect to the set of data items that are central to successful solution of the case. A mark (+, −, 0) under a disease variant and data item in this table indicates that the data item was judged to be positive, negative, or ambivalent evidence for the disease variant as a hypothesis.[5] For ex-

---

[4]Subjects E3 and E4 are the faculty subjects with upwards of 20 years of experience. They are noted with asterisks in this and all subsequent tables.

[5]There is no absolute correspondence between the use of a hypothesis at the point of a particular data item (Table 12-1) and the evaluation of the hypothesis with respect to that data item (Table 12-2). Subjects could evaluate a hypothesis with respect to a data item long past (e.g., evaluate with respect to data item 10 having reached, say, data point 17 of the case) and could also mention a hypothesis at a data point without necessarily evaluating the hypothesis with respect to that data item. Hence, for example, even though subject S2 mentioned all three variants at data point 10, he only ever evaluated one of these (SupAS) with respect to data item 10. The mention of the other variants at 10 was part of a puzzled attempt to recall the variants of aortic stenosis.

TABLE 12-1 Case 1: Subjects' Use of LCS Hypotheses in Response to Patient Data Items

		Patient data items																
		History						*Physical exam*								*X-ray*		*EKG*	
Subjects/hypotheses		*1*	*3*	*4*	*7*	*8*	*HHx*	*10*	*13*	*14*	*17*	*18*	*19*	*20*	*HPEx*	*22*	*Hxray*	*23*	*HEKG*
S1	ValvAS			X	X		X		X			X	X		X	X	X	X	X
	SupAS							X											
S2	ValvAS				X		X				X	X	X	X	X	X	X	X	X
	SubAS							X											
	SupAS							X											
S3	ValvAS						X		X	X	X	X	X	X	X	X	X	X	X
S4	ValvAS												X		X	X	X	X	X
T1	ValvAS						X				X	X	X		X	X			X
	SubAS												X	X	X	X	X	X	X
	SupAS												X						X
T2	ValvAS												X		X	X	X	X	X
T3	ValvAS				X	X	X		X		X	X	X	X		X		X	
	SubAS				X		X					X		X	X	X	X	X	X
	SupAS											X							
T4	ValvAS										X	X	X		X	X	X	X	X
E1	ValvAS	X	X		X		X				X	X	X	X	X	X	X	X	X
	SubAS										X		X		X	X	X		
	SupAS										X		X		X	X			
E2	ValvAS								X			X	X	X	X	X	X	X	X
	SupAS												X		X		X	X	
E3*	ValvAS		X				X				X	X	X	X	X	X	X		X
	SubAS										X	X	X	X	X	X	X		X
	SupAS							X			X		X		X	X			
E4*	ValvAS										X	X	X			X	X	X	X
	SubAS											X	X	X	X	X	X		X
	SupAS											X	X						

Note: X indicates a subject's use of a hypothesis at the time of a patient data item. HHx, HPEx, etc. refer to points in the case where subjects are asked for hunches.

* The two experts with more than 20 years of experience.

TABLE 12-2 Case 1: Evaluations of Target Data Items in Relation to LCS Hypotheses

	Target patient data items																	
	10 Normal facies			*17 Thrill*			*18 No click*			*19 Murmur*			*20 Aortic insuff.*			*22 Prominent aorta*		
Hypotheses	*ValvAS*	*SubAS*	*SupAS*	*ValvAS*	*SubAS*	*SupAS*	*ValvAS*	*SubAS*	*SupAS*	*ValvAS*	*SubAS*	*SupAS*	*ValvAS*	*SubAS*	*SupAS*	*ValvAS*	*SubAS*	*SupAS*
Subjects																		
S1			−				−			+						+		
S2			−	+			+			+			+			+		
S3				+			+			+			+			+		
S4										+						+		
T1				+			−			+	+	+		+		+	+	
T2										+						+		
T3			0	+	+	+	−	+		+			+				+	
T4				+			−			+						+		
E1			−	+	0	+	−	+	+	+	0	+	+	−	−	+	−	−
E2			−				−		+	+			+					
E3*			−	+	+	+	−	+		+	+	+	+	+		+	+	
E4*				+			−	+	+	+	+	+		+		+	+	

Note: + indicates subject judged data item as confirmatory for a hypothesis. − indicates subject judged data item as disconfirmatory for a hypothesis. 0 indicates subject judged data item as ambivalent in relation to a hypothesis.

* The two experts with more than 20 years of experience.

ample, a negative evaluation of "no click" with respect to ValvAS would be "The lack of a systolic ejection click is against valvular aortic stenosis."

Table 12-2 shows an increase, from students to experts, in the active evaluation of data items as evidence for or against the variants of aortic stenosis. In particular, no student evaluated all three of the variants with respect to a data item (of course, only one student, S2, ever mentioned all three variants at all). The two trainees (T1, T3) and three experts (E1, E3, E4) who used all three variants in the case also evaluated all three variants with respect to at least one data item. While this suggests activeness in the evaluation of variants by more experienced subjects, it does not necessarily reflect comparative evaluation. However, when a subject evaluates all variants with respect to the *same* data item, this is an indication that the subject is actively attempting to weigh the variants against each other to determine which is the best explanation for the data item and case. In this regard, no students, the two trainees (T1, T3), and again, the three experts (E1, E3, E4) evaluated all three variants with respect to a common (the same) data item. These same experts, but *not* the trainees, evaluated all variants in relation to *more* than one data item in common (E1, 5 items; E3, 2 items; E4, 2 items).

The analysis thus far suggests that with increasing diagnostic experience subjects know and actively utilize nonclassical variants of a disease as hypotheses in a diagnostic setting. Examination of the protocols of the two most experienced subjects, E3 and E4, yields some clue as to the knowledge structure that supports this performance. Figure 12-3 shows the protocols of these subjects at two data points: 17, which is the first strong evidence for valvular aortic stenosis and other variants; and 18, which is the strongest evidence against ValvAS. E3 raises all three variants together at the time of the first *strong* evidence. These hypotheses are then available to be evaluated comparatively against subsequent data, in particular, data item 18. This same form characterizes expert E1 (see Table 12-1). Expert E4, however, aggressively focuses on the "classic" member of the competitor set at 17, but immediately expands to the *full set* upon receiving strong negative evidence at 18. This form is shared by subject T3 and, less clearly, by subject T1 (see Table 12-1) and suggests that for these subjects LCS hypotheses other than the classic disease may have undergone at least partial activation earlier.

One explanation for these patterns is that in the expert a disease and its set of subtle variations come to constitute an interconnected memory unit, a kind of category; when one of the members is strongly activated in memory, the category and other members are also activated. The expert can then choose to consider category members in two modes. In the first mode, he or she tests all members simultaneously. This first mode might be termed *precautionary* since if any hypothesis encounters disconfirmatory evidence, alternative explanations for which the same evidence might be compatible are already under consideration. In the second mode, the expert tests only the most likely (in his or her current judgment) member.

(17) There is systolic thrill felt below the right clavicle, along the mid-left sternal border and in the suprasternal notch.

E3: This thrill is most consistent with a diagnosis of bicuspid aortic valve or aortic valvular stenosis. It would also be consistent with supravalvular stenosis and discrete subaortic stenosis.

E4: Until proved otherwise, now, he must have valvular aortic stenosis.

(18) The first heart sound is normal, and there is no systolic ejection click.

E3: The absence of a systolic ejection click in the presence of what I would consider to be an aortic outflow thrill makes aortic valvular stenosis and bicuspid aortic valve less likely. Aortic valvular stenosis of a very severe degree might be associated without a click. On the other hand, uh, it makes us think more seriously of discrete membranous subaortic stenosis.

E4: Absence of the click is against valvular aortic stenosis. Then perhaps instead he has subvalvular or supravalvular aortic stenosis.

FIGURE 12-3 Protocols from experts E3 and E4 at data points 17 and 18 in Case 1.

This mode might be termed one of *extraction* because its general success depends heavily on rejection of the target disease when appropriate, which, in turn, depends heavily on the precision in the diagnostician's model for the disease. In instances where the target disease is rejected, other category members provide a ready back-up set of alternative hypotheses. Further evidence for these speculations will be addressed as results from other cases are presented.

Diagnostic Errors in Case 1

A final analysis of the results of this case involves an attempt to discern the causes for subjects' errors in final diagnosis. Table 12-3 gives the final primary diagnosis for each subject. Among unsuccessful subjects, six subjects (S1, S3, S4, T2, T4, E2) never considered subvalvular aortic stenosis at all (see Table 12-1), although all generated and concluded valvular aortic stenosis. At least three explanations could apply to this lack of activation. First, and most basically, it could be that subjects do not know about SubAS at all. However, postexperimental interviews with all these subjects confirmed that they had some knowledge of this disease and could describe it. A second possible explanation is that these subjects have built up no

TABLE 12-3 Case 1: Subvalvular Aortic Stenosis —Final Diagnoses

Subjects		*Final diagnosis*
Students	S1	Valvular aortic stenosis
	S2	Valvular aortic stenosis
	S3	Valvular aortic stenosis
	S4	Valvular aortic stenosis
Trainees	T1	Subvalvular aortic stenosis
	T2	Valvular aortic stenosis
	T3	Subvalvular aortic stenosis
	T4	Valvular aortic stenosis
Experts	E1	Valvular aortic stenosis
	E2	Valvular aortic stenosis
	E3*	Subvalvular aortic stenosis
	E4*	Subvalvular aortic stenosis

*E3 and E4 are the two experts with more than 20 years of experience.

strong "bottom-up" association in memory between any data item of the case and the subvalvular disease. Even lacking such a "trigger" or recognition rule for SubAS itself, it would have been possible for subjects to generate SubAS as a side effect of their activation of ValvAS, if these two diseases were related in a memory unit, through a process of "spreading activation" (Anderson, 1976) or "top-down" activation (Rumelhart and Ortony, 1977; Bobrow and Norman, 1975). This suggests the third explanation—that for these subjects knowledge representations for the variants of aortic stenosis exist more in isolation than they do in the more experienced subjects. This is the issue of sparseness in disease knowledge.

For those subjects who generated ValvAS as a hypothesis but failed to abandon it in the face of strong negative evidence, examination of their handling of this disconfirmatory evidence yields insight into the nature and precision of their disease models for ValvAS. Discussion will focus on data item 18, the strongest evidence against ValvAS. Two students (S2, S3) evaluated 18, "no click," as confirmatory for ValvAS (Table 12-2). This appears to reflect, simply, an error in important factual knowledge about this disease. Two subjects (S4, T2) did not evaluate 18 at all with respect to ValvAS (Table 12-2). Significantly, they also did not generate *any* variant of aortic stenosis until after data item 18 (Table 12-1). This suggests that for these subjects the memory store of bottom-up associations between data items and aortic stenosis variants is not as extensive as for other subjects and, in particular, that data item 17 is not recognized as a strong cue for aortic stenosis–type diseases. A further implication is that the physical examination finding of a "systolic ejection click" and its import in ValvAS are not represented in the ValvAS disease models of these subjects, since, if

(18) The first heart sound is normal, and there is no systolic ejection click.

S1: Ah, well this, the fact that there is no systolic ejection click present, tells us that there is probably not a poststenotic dilation of the aorta, which one would expect with the presence of aortic stenosis and some aortic insufficiency. However, this does not necessarily rule it out.

T4: Love it. Um, well, okay. I wonder if there is..., no click, that's funny. I would expect it if he has AS. I wish they had said whether the murmur went up into his neck, okay.

(22) The chest x-ray shows normal cardiac size and contour and normal vascularity, but prominence of the ascending aorta.

S1: Ah, well this is what one would expect with ah, aortic stenosis with secondary aortic insufficiency. One would expect that the aorta, ascending aorta distal to the ah, to the stenosis, would be dilated due to the changes in the wall tension across the gradient. Therefore, ah, the fact that ah, a click was not heard on physical exam, may have been a subjective finding of the person examining. But, the x-ray does indeed suggest that there is some poststenotic dilation.

T4: Ha ha! AS-AI.

FIGURE 12-4 Protocols from subjects S1 and T4 at data points 18 and 22 in Case 1.

they were, the model itself should have led the subjects to reexamine this finding.

Finally, there were four subjects (S1, T4, E1, E2) who, although evaluating 18 as negative for ValvAS, still maintained ValvAS as a final diagnosis. The protocols of subjects S1 and T4 yield some insight into an explanation for these subjects. Figure 12-4 shows the protocols for these two subjects at data points 18 and 22, the latter consisting primarily of the finding of a "prominent aorta" on x-ray. Both subjects question ValvAS at 18, but are much more satisfied with this diagnosis at 22 and thereafter. Why might this be?

Figure 12-5 shows the causal relationship between a "tight" or stenotic aortic valve and an enlarged or prominent aorta. To open the tight valve, the left ventricle (LV) of the heart must generate abnormally high pressure. Blood expelled under this high pressure forces against the aortic wall and expands it. For the two subjects under discussion, it appears that their causal knowledge attributes the "systolic ejection click" in ValvAS to the enlarged aorta itself; that is, the click is caused by the large chamber into which the valve is opening, perhaps some kind of resonance phenomenon. For these subjects the causal chain from the valve to the click is as follows:

tight valve → big aorta → click

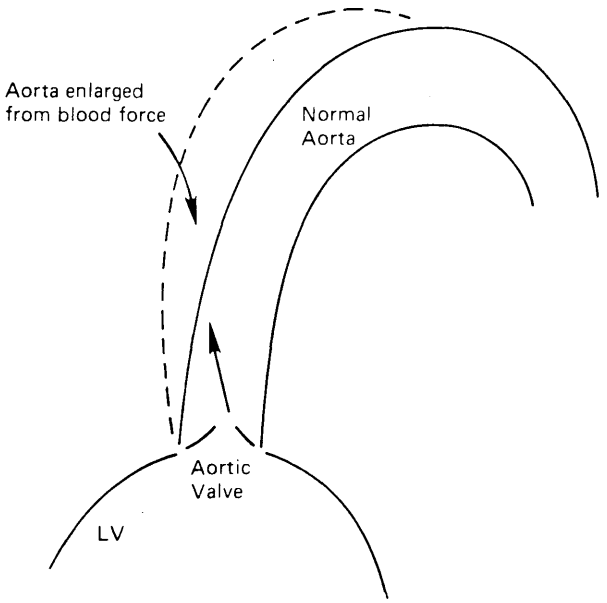


FIGURE 12-5 Aorta enlarged from the force of blood ejecting from a stenotic aortic valve.

Hence, for these subjects, the big aorta itself is predominant over the click as evidence for ValvAS, with the click just additional evidence for a big aorta. Once they receive their best evidence for a big aorta, data item 22, they are no longer worried about the lack of a click.

The true state of affairs appears to be that a tight valve causes both the click and the enlarged aorta at the same level of cause (Friedman and Kirkpatrick, 1977, p. 180). The systolic ejection click is associated with the opening of the tight valve itself as shown below:

tight valve → click
↳ big aorta

Hence both of these effects must be proved. Why might a number of subjects have misconstrued this relationship? One need look no farther than the introductory textbook these subjects use (Moller, 1978, p. 96) where the erroneous causal relationship is stated or at least strongly implied.

The subjects just discussed raise two important issues. First, they demonstrate how "small" knowledge errors can have major repercussions for the handling of a case, and they shed some insight into the case-specificity of a clinician's diagnostic performance found elsewhere (Elstein et al.,

1978). Second, they suggest a sensitivity in less experienced clinicians to specific training experiences, for example, training materials, particular patient cases, etc. As experience increases, so does the sample of "inputs" and the effects of particular experience might be expected to lessen.

12.4.2 Case 2: Total Anomalous Pulmonary Venous Connection

The purpose of this case is to investigate the aggregation by subjects of a set of physiologically similar diseases into a memory grouping or category. The case is different from Case 1 in that while Case 1 dealt with a set of variants of one disease, Case 2 is concerned with a set of diseases.

The logical competitor set for Case 2 includes four diseases: total anomalous pulmonary venous connection (TAPVC), partial anomalous pulmonary venous connection (PAPVC), atrial septal defect (ASD), and endocardial cushion defect (ECD). Figure 12-6 shows the anatomical and physiologic abnormalities within the heart that define each of these diseases.

In TAPVC, all four pulmonary veins (PVn in Figure 12-6) connect to the right atrium (RA) of the heart rather than to the left atrium (LA), their normal site of connection. All oxygenated blood coming back to the heart from the lungs mixes with deoxygenated blood coming back to the heart from the body. Hence, all blood subsequently pumped back to the body is a mixture of oxygenated and deoxygenated blood, which causes the patient to appear cyanotic, that is, to take on a mildly "blue" skin coloration.

In PAPVC, only a subset of the pulmonary veins connect abnormally to the right atrium, with the remainder connecting, as they should, to the left atrium. A result is that some already oxygenated blood is recirculated through the lungs. Blood pumped to the body, however, is oxygenated, and the patient retains a normal "pink" coloration.

Both ASD and ECD consist of a defect (a hole) in the atrial septum of the heart. They differ in the particular site of defect; ASD is a defect in the upper portion of the septum (the ostium secundum) while ECD is a defect in the lower portion of the septum (the ostium primum). In both diseases, the presence of the hole in the septum allows blood to shunt from the left atrium to the right atrium. While some oxygenated blood shunts to the right side to be recirculated to the lungs, blood expelled to the body is oxygenated, and the patient is pink.

A feature common to all four diseases in the LCS is an increased volume of blood in the right-sided chambers of the heart. This common element is a candidate feature on which diagnosticians might base a disease category, for example, "diseases with right-sided volume overload." A clinical manifestation related to volume overload that all these diseases produce in common is a set of three auscultation findings. One is a murmur

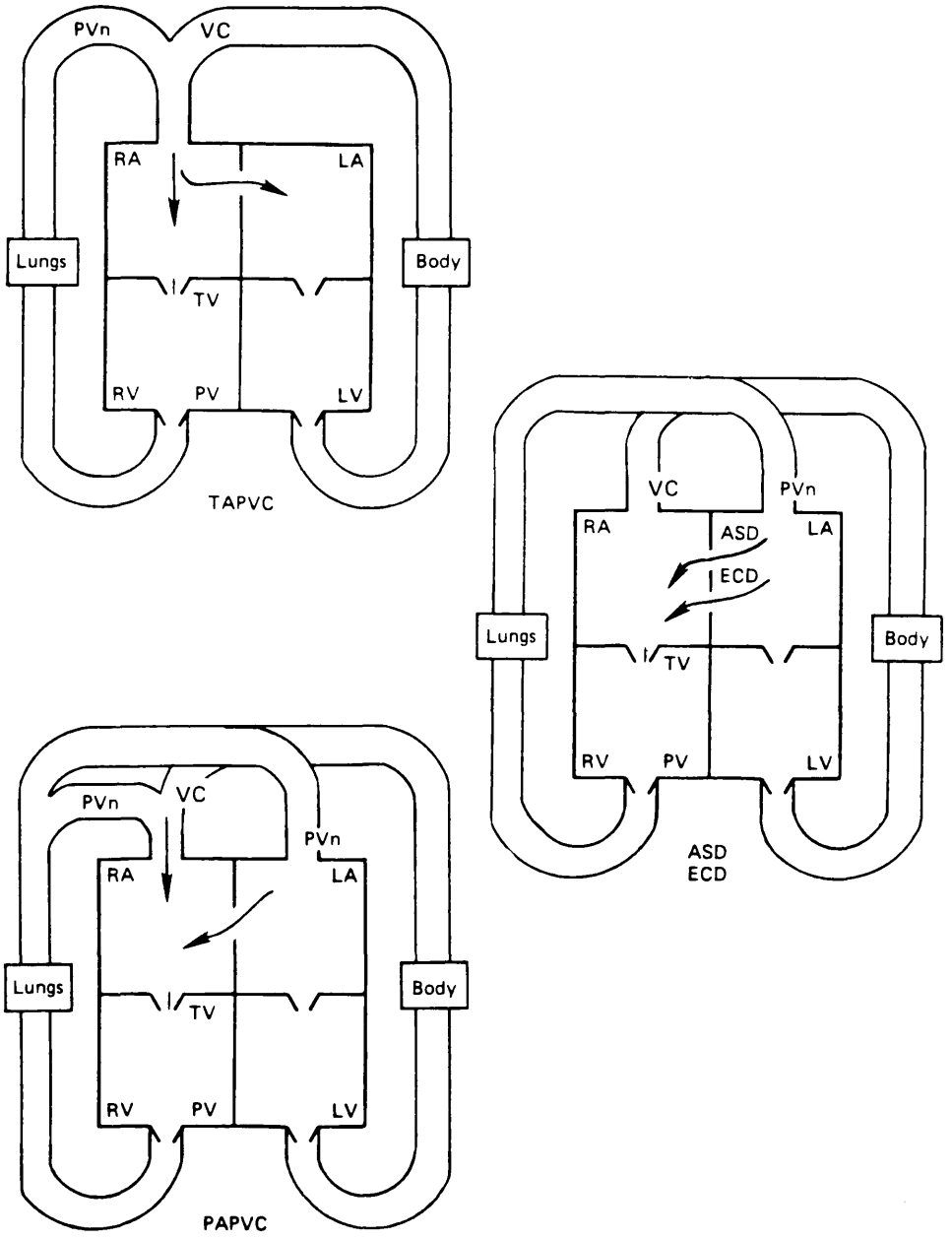


FIGURE 12-6 Logical competitor set for Case 2: total anomalous pulmonary venous connection, partial anomalous pulmonary venous connection, atrial septal defect, and endocardial cushion defect.

associated with increased blood flow across the tricuspid valve (TV). The second is a murmur associated with increased flow across the pulmonary valve (PV). The third is wide, fixed splitting of the second heart sound. The third finding is nearly pathognomonic for conditions of this type.

Of the four diseases, ASD is more common than the others. Hence it might be expected that subjects' knowledge for this disease would develop more rapidly than for the others. More importantly, ASD is the disease that is used instructionally to introduce the concepts of atrial level left-to-right shunting of blood in the heart and right-sided volume overload. Therefore, it might be expected that the three auscultation findings (especially the splitting) reflecting overload would be more strongly associated with ASD than with the other diseases. TAPVC, however, is the operative disease in the case.

There are six particularly important data items in the patient file presented to subjects for Case 2. Data items 17, 18, and 19 contain the set of three findings discussed above that are salient results of increased right-sided heart flow. Item 17 contains the "wide, fixed, split second heart sound." Hence, it was expected that all subjects would raise at least ASD, the classic instance of this type of disease, by the time of these data points. Data item 7 (also 11), which reports that the patient is mildly cyanotic, represents disconfirmatory evidence for all members of the LCS except TAPVC. Data item 21, which contains an x-ray description of "an unusual vascular shadow on the right side," is evidence against ASD and simultaneously constitutes a classic cue for PAPVC. In fact, one variant of PAPVC, scimitar syndrome, derives its name from its presentation of such a finding on x-ray (Lucas and Schmidt, 1977, p. 442). The EKG, item 22, contains a finding of "right-axis deviation" on the EKG and constitutes strong disconfirmatory evidence for ECD. All data of the case are compatible with the operative disease, TAPVC.

Use of the Logical Competitor Set in Case 2

Table 12-4 shows all uses by all subjects of the four diseases in the logical competitor set for Case 2 at all patient data points where any of the four was mentioned by any subject.

For reasons discussed above, it was assumed that most subjects would consider ASD in relation to the three data items, 17, 18, and 19. The use of other LCS members at these points is taken as evidence that the other diseases are associated in memory with ASD and this set of cues. Table 12-4 shows a decrease from students to experts in the number of subjects who considered only ASD at these points. All of the students considered only ASD, the disease we presume to be the classic exemplar of right-sided volume overload, at data items 17–19. Three of four trainees (T1, T2, T3) and the two least experienced experts also considered only ASD. Of the

TABLE 12-4 Case 2: Subjects' Use of LCS Hypotheses in Response to Patient Data Items

		Patient data items													
		History					*Physical exam*					*X-ray*		*EKG*	
Subjects/ hypotheses		*1*	*3*	*5*	*7*	*HHx*	*17*	*18*	*19*	*20*	*HPEx*	*21*	*Hxray*	*22*	*HEKG*
S1	ASD					X	X				X				
	PAPVC											X	X	X	X
S2	ASD						X		X		X				
	ECD											X	X	X	
	PAPVC											X			
S3	ASD						X				X		X		X
	ECD													X	X
	PAPVC											X			X
S4	ASD						X	X	X			X			X
	ECD													X	
	TAPVC											X			
T1	ASD						X		X						
	PAPVC											X	X	X	X
T2	ASD										X				
	ECD								X		X				
	TAPVC												X	X	X
T3	ASD			X	X		X	X	X		X	X		X	
	PAPVC													X	
	TAPVC													X	X
T4	ASD						X			X				X	X
	ECD									X	X	X	X	X	
	PAPVC													X	
	TAPVC						X					X			X
E1	ASD	X	X	X		X	X		X	X		X			
	PAPVC											X	X		X
E2	ASD								X		X			X	
	PAPVC										X				X
	TAPVC														X
E3*	ASD	X					X	X	X						
	ECD					X					X		X	X	
	PAPVC						X	X							
	TAPVC					X	X	X	X		X	X	X	X	X
E4*	ASD						X		X		X				
	ECD								X			X			
	PAPVC											X			
	TAPVC											X			X

Note: X indicates a subject's use of a hypothesis at the time of a patient data item. HHx, HPEx, etc. refer to points in the case where subjects are asked for hunches.

*The two experts with more than 20 years of experience.

two highly experienced experts, E3 utilized three LCS members (ASD, PAPVC, TAPVC) and E4 used two (ASD, ECD) at these points.

From the point of view of the entire case, no students, one trainee (T4), and two experts (E3, E4) generated *all four* members of the LCS during the course of the case. While this shows no obvious general trend toward increased use of the LCS with experience, it is perhaps significant that the full competitor set was used by the two high-level experts, E3 and E4.

In utilizing the full logical competitor set, the two most experienced subjects, E3 and E4, demonstrated the same patterns of *precaution* and *extraction* respectively as they did in Case 1. E3 considered three of the four LCS members (ASD, PAPVC, TAPVC) at item 17, the *first* strong cue for right-sided volume overload. E4 raised only ASD at this point and maintained this hypothesis until data item 21, which contains strong evidence against ASD. At this point, he expanded to the remainder of the LCS.

Diagnostic Errors in Case 2

Table 12-5 gives the final primary diagnoses for all subjects on Case 2. Only four subjects (trainees T2 and T3 and the two most experienced experts, E3 and E4) diagnosed the case correctly. Subjects who diagnosed the case incorrectly demonstrate informative types of errors.

Student S3 diagnosed the case as endocardial cushion defect (ECD). The strongest evidence against this disease is the finding of right-axis deviation on the EKG (data item 22). ECD uniformly presents with *left*-axis deviation and, in fact, is one of a very few congenital heart diseases that does; hence left-axis deviation is a nearly pathognomonic finding for ECD. S3 not only evaluated the *right axis* as positive evidence for ECD, but, in addition "triggered" or proposed ECD for the first time at this point (see Table 12-4). This is, simply, imprecision in the subject's disease model for ECD. It is as though the subject remembered that the EKG axis is important in ECD but could not remember the details.

The final diagnosis of subject T4 was ASD, even though she had considered TAPVC during the case. She correctly evaluated cyanosis (blueness—items 7 and 11) as negative for ASD, but maintained ASD nonetheless. Her primary difficulty was that she did not believe that TAPVC could present in a child as old as the one in the case (5 years old), although it certainly can—as the case itself, a real case, attests. This suggests that the allowable age range specified in the subject's disease model for TAPVC is overly restrictive, probably reflecting a limited sample of experiences with this disease.

Four subjects (S1, T1, E1, E2) diagnosed the case as PAPVC. Three of these subjects (S1, T1, E1) show a pattern in which only ASD (among the LCS members) is considered prior to data item 21, a classic x-ray cue for PAPVC, and only PAPVC is considered at that point and thereafter

TABLE 12-5 Case 2: Total Anomalous Pulmonary Venous Connection—Final Diagnoses

	Subjects	*Diagnosis*
Students	S1	Partial anomalous pulmonary venous connection
	S2	Transposition of the great vessels + pulmonary stenosis + atrial septal defect + partial anomalous pulmonary venous connection
	S3	Endocardial cushion defect
	S4	Pulmonary stenosis + atrial septal defect + ventricular septal defect
Trainees	T1	Partial anomalous pulmonary venous connection
	T2	Total anomalous pulmonary venous connection
	T3	Total anomalous pulmonary venous connection
	T4	Atrial septal defect
Experts	E1	Partial anomalous pulmonary venous connection
	E2	Partial anomalous pulmonary venous connection
	E3*	Total anomalous pulmonary venous connection
	E4*	Total anomalous pulmonary venous connection

*The two experts with more than 20 years of experience.

(see Table 12-4). This indicates a strong data-driven dependence in the diagnosis by these subjects; that is, the subjects are pushed from hypothesis to hypothesis depending on the most recent strong disease cue in the data, and when new hypotheses are generated, these are not strongly enough associated in memory with other LCS members to activate these other diseases. Some support for this claim can be seen in subject T1's protocol, taken from the point in the case where he offers his final diagnosis:

> *T1:* I am sort of drawing a blank on how to fit all this information together. And ah, I am just sort of guessing right now. I would say just scimitar syndrome [PAPVC] primarily based on the chest x-ray, and ah, I'm not really sure whether the whole thing fits together well. That is all I can say.

Of the four subjects, student S1 never evaluated PAPVC with respect to cyanosis; hence this finding had no opportunity to detract from his PAPVC hypothesis. Subject T1 evaluated cyanosis as *confirmatory* evidence for PAPVC, and this erroneous evaluation reinforced this disease interpretation. Expert subjects E1 and E2 evaluated cyanosis appropriately as negative evidence for PAPVC, but this evaluation was probably overridden by the strength of the cue for PAPVC on the x-ray.

Finally, two students (S2, S4) proposed configurations of multiple diseases as explanation for the case. Both of these composite diagnoses in-

TABLE 12-6 Case 2: Interpretations of Data Item 18

		Interpretation	
Subject		*Pulmonary stenosis*	*Increased flow pulmonary valve*
Students	S1	+	
	S2	+	
	S3	+	+
	S4	+	
Trainees	T1		
	T2	+	
	T3	+	+
	T4		+
Experts	E1	+	+
	E2		
	E3*		+
	E4*	+	+

Note: + indicates that a subject interpreted the murmur of data item 18 as pulmonary stenosis or increased flow over the pulmonary valve.
*The two experts with more than 20 years of experience.

cluded the disease pulmonary stenosis (PS), and it is this component of the final diagnosis that is the key to understanding the performance of these two subjects. Table 12-6 shows the interpretations by all subjects of data item 18, a systolic murmur in auscultation of the heart. Such a murmur results whenever there is too much flow over the pulmonary valve, relative to its orifice size. This situation prevails in either of two conditions:

1. When there is normal amount of flow but an abnormally small orifice. This is the disease pulmonary stenosis, which refers to an abnormally tight valve.
2. When there is a normal-sized orifice but abnormally high flow, the situation that prevails in the diseases of the LCS. A + under one of these two interpretations in Table 12-6 indicates that a subject attributed this interpretation to the murmur of data item 18.

Table 12-6 shows that most of the students (three of four) interpreted the murmur only as pulmonary stenosis, while most of the expert group (three of four) interpreted the murmur as increased flow *or* a tight valve. While student S1 (and subject T2) was eventually able to extract himself from his interpretation, students S2 and S4 were not. Once these students introduced PS into their diagnoses, they were forced to propose rather unusual combinations of multiple diseases to account for some of the findings of the case. For example, subject S2, in order to reconcile PS with other data of the case indicating increased blood flow in the lungs, simply transposed the great vessels of the heart; that is, he detached the pulmo-

nary artery from its normal mooring at the pulmonary valve and reattached it at the aortic valve and did the opposite with the other great vessel, the aorta. While this rather creative causal explanation represents a congenital heart disease, transposition of the great vessels, it is highly unlikely that a child with the combination of abnormalities proposed by the subject could have lived for five years untreated.

The interpretations of the systolic murmur by the students in Case 2 is another example of error, or at least limitation, in causal knowledge. It represents a situation where there are multiple causes for a finding and the novice considers only a subset. This is not unlike what has been shown at the disease and disease variant levels; that is, when multiple diseases in the logical competitor set can produce a finding, the novice seems limited to the most salient members. This suggests the import of grouped or clustered memory organization not only for diseases but also for "low-level," pathophysiologic interpretations for data.

12.4.3 Case 3: Patent Ductus Arteriosus

The purpose of this case is to test the robustness of expert grouping of hypotheses in a straightforward case in which there are no data discrepant with an initially induced disease interpretation. Interest is in whether subjects, even in a case with a very common disease, strong cues for this disease, and no data discrepant with this interpretation, still investigate a related set of physiologically similar alternatives.

The operative disease in the case is patent ductus arteriosus (PDA), a schematic for which is shown in Figure 12-7. This disease is an extracardiac shunt, that is, an abnormal communication between vessels, the aorta (Ao) and the pulmonary artery (PA), outside the heart. There are four other "disease" conditions in the logical competitor set. The congenital heart diseases arterio-venous fistula (AVF) and aorto-pulmonary window (APW) are other extracardiac shunts. Venous hum (VH) is a benign condition that presents a murmur similar to PDA, and ruptured sinus of valsalva (RSV) is a heart condition that has a clinical presentation similar to that of PDA. In the patient file presented to subjects for Case 3, the most important patient data item is number 19, a classic murmur of patent ductus arteriosus. It was assumed that all subjects would generate PDA as a hypothesis no later than this point. No data of the case are incompatible with PDA.

Use of the Logical Competitor Set in Case 3

Table 12-7 shows all uses of members of the logical competitor set by all subjects during the course of the case. It is clear that only one subject, E3, one of the two high-level experts, considered the full competitor set, al-

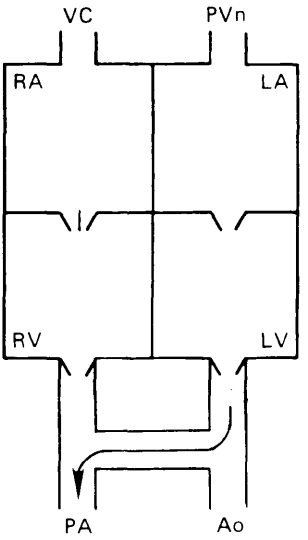


FIGURE 12-7 Patent ductus arteriosus.

though expert E2 considered three of the five—more than any of the remaining subjects. Since it was assumed that all subjects would consider PDA, a criterion far less stringent than "full use" for the LCS is the number of subjects in each group who considered even one additional LCS member and used it more than once. This condition holds for only one student (S3), one trainee (T1), but three of the experts (E1, E2, E3).

Expert E3 considered the full LCS in a precautionary pattern consistent with his performance on other cases (see Table 12-7). He used three of the five LCS members as hypotheses at data item 19, a strong cue for PDA. The remainder of the LCS was filled out two items later, after an intervening, uninformative data item, at the point where the subject was asked for "hunches." The other high-level expert, E4, looks in all respects like a novice in this case, in that he considered only PDA. However, if our earlier interpretations of an extraction method are correct for this subject, we would not expect him to expand to other members of the competitor set unless he encountered data discrepant with his target hypothesis; of course, there are none in this case.

The diseases in this case constitute a category of extracardiac communications and related conditions. An interpretation of the results from this case is that with high-level experience, it is this category, and not isolated individual members, that is generated and tested when a strong cue for a category member is encountered. No subject diagnosed this case incorrectly; hence analysis of subject errors is uninformative.

TABLE 12-7 Case 3: Subjects' Use of LCS Hypotheses in Response to Patient Data Items

		Patient data items												
		History					*Physical exam*				*X-ray*		*EKG*	
Subjects/ hypotheses		*3*	*4*	*5*	*7*	*HHx*	*14*	*19*	*20*	*HPEx*	*21*	*Hxray*	*22*	*HEKG*
S1	PDA					X		X		X	X	X	X	X
	APW												X	
S2	PDA					X		X		X	X	X	X	X
S3	PDA		X	X	X	X		X		X	X	X	X	X
	AVF									X				X
S4	PDA					X	X	X		X	X	X	X	X
	AVF						X							
T1	PDA						X	X	X	X	X	X	X	X
	APW						X	X		X		X		X
T2	PDA							X		X	X	X	X	X
T3	PDA							X	X	X	X	X	X	X
T4	PDA	X					X	X	X	X	X	X	X	X
E1	PDA						X	X		X	X	X		X
	AVF							X				X		X
E2	PDA						X	X		X	X	X	X	X
	AVF						X	X						
	VH							X		X				
E3*	PDA							X		X	X		X	
	AVF									X				
	VH							X						
	APW							X		X	X			
	RSV									X				
E4*	PDA							X		X		X	X	X

Note: X indicates a subject's use of a hypothesis at the time of a patient data item. HHx, HPEx, etc. refer to points in the case where subjects are asked for hunches.

*The two experts with more than 20 years of experience.

12.4.4 Case 4: Pulmonary Atresia

The objective of this case is similar to that of Case 2, that is, to assess subjects' aggregation of physiologically similar diseases into categories. Case 4 is different from Case 2 in that no single cue serves to distinguish the members of the logical competitor set from diseases outside it (as did "wide, fixed, split second heart sound" in Case 2). In Case 4 the diagnostician must arrive at the LCS by partitioning the space of diseases, using multiple data items from widely separated parts of the case.

The logical competitor set for Case 4 includes three diseases: pulmonary atresia (PAT), tricuspid atresia (TAT), and Ebstein's malformation of

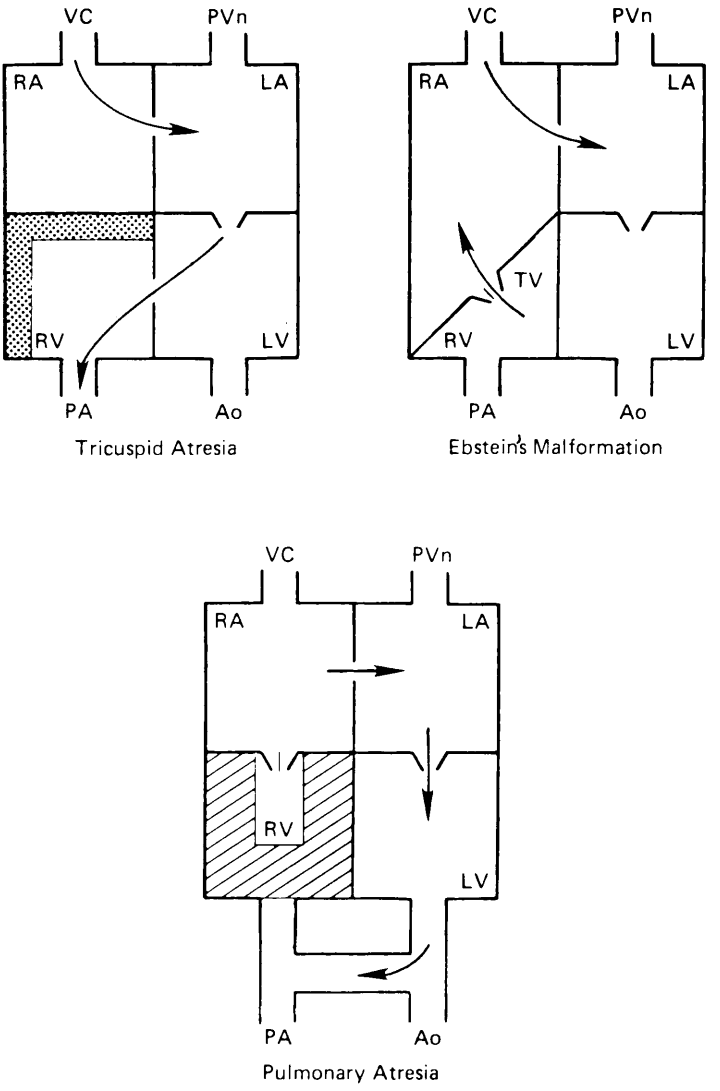


FIGURE 12-8 Logical competitor set for Case 4: pulmonary atresia, tricuspid atresia, and Ebstein's malformation

the tricuspid valve (EBST). Figure 12-8 depicts the anatomical abnormalities within the heart that define each of these diseases. In pulmonary atresia and tricuspid atresia, the pulmonary and tricuspid valves respectively are "shut" (only tissue exists where the valves should be). In Ebstein's malformation, a diminutive and noncompliant right ventricle (RV) restricts inflow of blood to that ventricle. The net physiology of all these diseases

is one of obstruction to blood flow on the right side of the heart, resulting in reduced blood flow to the lungs and right-to-left shunting of blood at the atrial level within the heart. The right-to-left shunting and diminished blood flow to the lungs cause the patient to be cyanotic (blue skin coloration). In short, these diseases constitute a physiologic category of "cyanotic diseases with decreased pulmonary blood flow."

Pulmonary atresia is the operative (or true) disease in the case. The three members of the LCS are best discriminated on the EKG. Tricuspid atresia produces a finding of left-axis deviation on the EKG, while pulmonary atresia produces a normal EKG axis. Ebstein's, unlike the other two, produces an EKG finding of right bundle branch blocking. All other clinical manifestations of the three diseases are quite similar.

There are several key data items in the patient file presented to subjects for Case 4. The subject receives evidence of cyanosis during history and early physical examination (items 1, 3, and 8). The x-ray, item 17, contains evidence of diminished blood flow to the lungs and, with the cyanosis evidence, could enable the subject to narrow diagnosis to the three members of the LCS. The EKG, item 18, contains information to discriminate among these.

Use of the Logical Competitor Set in Case 4

Table 12-8 shows all uses of members of the logical competitor set as hypotheses by all subjects during the course of the case. Table 12-8 shows a clear increase in the use of the full LCS from students to trainees, but no clear difference in this regard between trainees and experts. In particular, no student considered the full LCS, and two students (S1, S3) considered only one member. All four trainees and three experts (E1, E2, E3) used all of the diseases in the LCS. Two experts (E2, E3) used all three diseases more than once, while no trainee did—suggesting somewhat more active consideration of the LCS by these experts.

While both trainees and experts considered the full LCS, their patterns of use of these diseases were different. Three of the four experts used all members of the LCS at data point 17 (the x-ray) or at the immediately succeeding point where subjects reported hunches. Since item 17 is the data item that allows specification of the category "cyanotic heart diseases" into the category "cyanotic diseases with decreased pulmonary blood flow," this pattern suggests that the expert subjects were using this category. In contrast, *no* trainees used all three LCS members at either of these points, suggesting that these three diseases do not, at least to the same extent, constitute a functional diagnostic category for these subjects.

Regarding the expert diagnostic modes of precaution and extraction, expert E3 again considered all three LCS members together before the onset of data useful for discriminating among them. Expert E4 considered explicitly only pulmonary atresia, the correct disease, at data item 17. How-

TABLE 12-8 Case 4: Subjects' Use of LCS Hypotheses in Response to Patient Data Items

		Patient data items								
		History	*Physical exam*				*X-ray*		*EKG*	
Subjects/hypotheses		*HHx*	*14*	*15*	*16*	*HPEx*	*17*	*Hxray*	*18*	*HEKG*
S1	TAT						X	X	X	
S2	TAT			X				X	X	X
	PAT			X	X	X		X	X	
S3	EBST								X	X
S4	TAT						X	X	X	X
	PAT			X		X	X	X		X
T1	EBST						X			
	TAT			X		X	X	X	X	X
	PAT					X				
T2	EBST								X	
	TAT									X
	PAT			X						
T3	EBST						X			
	TAT			X						X
	PAT			X	X	X	X	X	X	X
T4	EBST									X
	TAT						X		X	
	PAT						X		X	X
E1	EBST						X			
	TAT						X	X	X	X
	PAT	X		X		X	X	X	X	X
E2	EBST						X	X		
	TAT						X	X	X	X
	PAT						X	X		X
E3*	EBST						X	X	X	X
	TAT		X				X	X	X	X
	PAT							X	X	X
E4*	PAT						X	X	X	X

Note: X indicates a subject's use of a hypothesis at the time of a patient data item. HHx, HPEx, etc. refer to points in the case where subjects are asked for hunches.

*The two experts with more than 20 years of experience.

ever, his protocol from the immediately succeeding data point, Hxray (hunches after x-ray), shows explicit consideration of the category of "cyanotic disease with decreased pulmonary blood flow" with targeting for active consideration of the particular LCS member he judged most likely:

> *E4:* At this point the picture would be more likely that of cyanotic heart disease involving decreased pulmonary blood flow. The specific defect would seem to be pulmonary atresia with intact septum.

TABLE 12-9 Case 4: Pulmonary Atresia—Final Diagnoses

	Subjects	*Diagnosis*
Students	S1	Truncus arteriosus
	S2	Hypoplastic right ventricle
	S3	Truncus arteriosus
	S4	Pulmonary atresia
Trainees	T1	Tricuspid atresia
	T2	Tricuspid atresia
	T3	Tricuspid atresia
	T4	Pulmonary atresia
Experts	E1	Pulmonary atresia
	E2	Pulmonary atresia
	E3*	Ebstein's malformation
	E4*	Pulmonary atresia

*The two experts with more than 20 years of experience.

Since no succeeding data are discrepant with this target hypothesis, his performance is consistent with the extraction mode as we have proposed it. In addition, E4's overt consideration of the LCS category here lends credence to a speculation we have made about the extraction mode in Case 1 and Case 2, that is, that the subject covertly considered the LCS category in those cases before he overtly articulated the members.

Diagnostic Errors in Case 4

Table 12-9 gives the final primary diagnoses for all subjects. The final diagnoses of the students on this case are outside the logical competitor set, and the full explanation for their performance is not transparent. However, a partial explanation can be given.

Two students (S1, S3) gave a final diagnosis of truncus arteriosus. Truncus is a congenital heart disease in which the aorta and pulmonary artery, the two great vessels that normally lead out of the heart, are merged into one large outlet vessel with one outlet valve. The single valve results in a patient finding of "single second heart sound" on auscultation as presented in Case 4. While truncus produces a single heart sound, so do a number of other diseases, including all members of the logical competitor set. It is not even necessary that only one valve exist for only a "single sound" to be produced; the same finding is produced when there are two outlet valves but the blood flow across one of them is substantially diminished—the situation in Ebstein's malformation and tricuspid atresia.

(15) The second heart sound is single and perhaps slightly increased in intensity. There is no gallop or diastolic murmur.

S1: Well, this is a significant finding because ah, the fact that the second heart sound is not split ah, suggests that ah, we could be dealing with a truncus.

S3: It could be ah, ah. There is a single outflow tract, ah. It could be truncus arteriosus. Ah, that would fit with the single S2 [second heart sound] . . . So, I'll go with number one on my list as ah, truncus arteriosus, and I'm not sure what type. I'd have to do an angio, I guess, or I mean arteriography.

FIGURE 12-9 Protocols from subjects S1 and S3 showing interpretation of "single second heart sound"—Case 4.

One explanation for the performance of students S1 and S3 is that they judged the "single sound" to be more discriminating for truncus than it really is; in particular, they did not consider the multidimensional nature of this finding—number of valves *and* flow. Some evidence for this explanation can be seen in protocols from these two subjects showing interpretations of the patient finding of a "single sound" (Figure 12-9). It is clear that this finding had a substantial influence on the final diagnoses of these subjects. If our interpretation for these subjects is correct, it would be another example of how the beginning practitioner is restricted in the number of alternative explanations he or she can bring to bear on a finding, at the level of either alternative pathophysiological causes or alternative disease explanations. In addition, the restricted explanations of novices are the highly salient or "classic" ones, since the "common trunk" that *defines* truncus greatly highlights the single sound as an expected finding in that disease.

S2, the other student who misdiagnosed Case 4, gave as a final diagnosis (hypoplastic right ventricle) one of the patient findings presented in the case (the EKG); that is, the subject used one of the patient data items as a final diagnosis. This subject suggests a kind of constraint relaxation that interacts with interpretive restrictiveness in the novice. The usual or preferred constraint on a good diagnostic explanation is that it account for much of the case data. When the novice encounters severe difficulty in meeting this constraint, he or she relaxes to accounting for a few key data items (S1, S3 above) or, in the extreme, to a data item itself, which embodies a level of physiological/disease interpretation.

The trainees and experts were nicely split on this case with most trainees (three of four) judging tricuspid atresia and most experts (three of four) judging pulmonary atresia, the correct disease. Recall that TAT and PAT are distinguishable on the axis of the EKG where TAT presents left-axis deviation and PAT presents a normal, undeviated axis. It is in the subjects' evaluations of this particular data item that we might expect to find an explanation for the performance of these two groups.

TABLE 12-10 Case 4: Evaluations of EKG Axis in Relation to Tricuspid Atresia and Pulmonary Atresia

		Hypotheses	
Subjects		*Tricuspid atresia*	*Pulmonary atresia*
Students	S1		
	S2		
	S3		
	S4	+	
Trainees	T1	+	
	T2		
	T3		
	T4	+	
Experts	E1	0	+
	E2	−	
	E3*	−	
	E4*		

Note: +, −, or 0 indicate that the subject evaluated the EKG axis as confirmatory, disconfirmatory, or ambivalent evidence, respectively, in relation to the hypothesis.

*The two experts with more than 20 years of experience.

Table 12-10 shows all explicit evaluations by subjects of the EKG axis as confirmatory (+), disconfirmatory (−), or ambivalent (0) evidence with respect to pulmonary atresia and tricuspid atresia. All subjects below the expert level who explicitly evaluated the axis with respect to either of these two diseases evaluated the axis as confirmatory evidence for tricuspid atresia. All expert subjects who explicitly evaluated the axis evaluated it as either disconfirmatory for tricuspid atresia or confirmatory for pulmonary atresia.

The EKG axis as presented in the case is +50 degrees, which technically represents left-axis deviation [for a four-day-old child, as presented in the case (Moller, 1978, p. 24)] as one would expect in tricuspid atresia. So that if one were using the textbook rule for discriminating PAT and TAT (Moller, 1978, p. 137), tricuspid atresia *would* be the diagnosis of choice in the case. However, the expert evaluations of this finding, as well as postexperimental discussions with these subjects, confirmed that the experts judged +50 degrees to be "just not far enough leftward" for tricuspid atresia and that these subjects would require the axis to be "down around zero or negative" before they would choose TAT over PAT. We see here a nice example of overly general, textbooklike rules of evaluation and clinical expectations in less experienced subjects (imprecise disease models) and pinpoint refinement of these in more experienced diagnosticians, probably just reflecting their greater clinical experience with the two diseases and the contextually dependent manifestations.

12.5. Summary

For the cases of the study, an expert *form* and an expert *substance* for diagnosis were identified. The expert form involves the full, active use of a set of physiologically similar diseases (the logical competitor set) for each case, diseases that have similar physiological structure and clinical presentation. The use of this set by the experts, generally in close proximity to the strongest cues for any member of the set, is interpreted here as evidence that these diseases constitute a unit or category in memory. Since diseases in the LCS are likely to be confused with each other, it would seem that as a "long-run" strategy of diagnosis it would be adaptive for a diagnostician to consider (give a "hearing" to) other members of the set whenever there is reason to believe any one of them is a good candidate in a case. It appears that this is what the experts do. Expert substance refers to correct data evaluations, within the logical competitor set of diseases, necessary to isolate the correct member. This is taken as evidence for precision in these subjects' models for diseases.

For the two high-level experts in the study, two distinct methods of utilizing the LCS were also identified:

1. *Precaution.* This involves the generation and use *together* as hypotheses of the full set of logical competitors, enabling them to be weighed against each other and the data.
2. *Extraction.* This method involves more aggressive focus on a member of the set, with full expansion to the remainder of the set as disconfirmatory evidence for the target member is found.

Medical students, after six weeks of training and clinical practice in the field represented by the cases, generally showed neither expert form nor expert substance. Students hardly ever considered the full LCS and focused on the "classic" members in cases that encouraged this. This suggests that LCS members, when they exist at all, are represented in a more isolated form in memory. Errors of evaluation (shared at times with intermediate-level subjects) included several types:

1. *Mundane factual errors.* These are just factual errors about which findings "go with" which diseases.
2. *Causal errors.* These are errors concerning how observable data are related to underlying physiology.
3. *Imprecise tests.* These are either overly general or overly restrictive tolerances on the range of variability allowed in an expected clinical finding for a disease.

4. *Interpretive restrictiveness.* This refers to restriction in the number of interpretations that are made of a finding. In some instances, these errors can be interpreted simply as reflecting imprecision in subjects' models for diseases, but other errors suggest a deficiency in integrating disease models or data with their underlying causal or physiological mechanisms.

The trainees in the study showed performances that at times looked very much expertlike and at other times could not be distinguished from the students. The number of trainees in each case who used the full LCS generally fell between the number of students and the number of experts. Moreover, depending on subject and case, trainees at times exhibited the types of errors discussed above for the students. The ultimate diagnoses of the trainees, unlike those of the students, were generally at least within the LCS, if not correct. This suggests that for these subjects the main problems were lack of connectedness in memory among LCS members or imprecision in knowledge necessary for discriminating LCS members correctly.

12.6 Discussion

The study demonstrates that diagnosticians' disease knowledge, a memory store of disease models and the memory organization among them, is crucial to successful diagnosis and does discriminate expert from less expert performance. The major differences that have been demonstrated among subjects concern their handling of a set of "good moves," that is, the logical competitor sets. More experienced subjects tend to consider more of the good hypotheses in a case, consider them in groups, and evaluate them correctly.

The study did not set out to show that highly experienced practitioners are better diagnosticians than novices; this should go without saying. The intent was to learn something about the medical knowledge that diagnosticians use, the way this knowledge influences performance, and the ways this knowledge changes as people acquire experience in a field. Medical students, after only six weeks of training in the field of interest, were included because these individuals represent the "starting point" in a long learning process.

12.6.1 The Nature of Knowledge Change

What has been learned about the nature of knowledge change? It seems clear that the whole learning process starts with a small set of "classic" training concepts where these include particular diseases, descriptions of

expected patient findings under these diseases, and rules for disambiguating diseases in this starting set. The learning of these training concepts is encouraged by the selection of content for inclusion in introductory training materials, that is, introductory textbooks and classroom instruction. The diseases are the common ones, the patient data descriptions are prototypic, or average, and the rules of evaluation are overly simplified. We have seen several instances where the locus of novice errors could be traced fairly directly to such statements in the introductory textbooks to which the subjects had been exposed. Although students' initial exposure is limited, it provides the cognitive "anchorage points" to enable them to benefit from the experience to follow.

With experience, the practitioner is exposed to and adds to memory additional diseases beyond the starting point set. Within psychology, the expert's "large vocabulary" of discriminable instances is now assumed (Chase and Simon, 1973). Concurrently with the addition of disease models to memory, there is an embellishment of the compositional features of a disease that are encoded in each disease model. These are features representing the disease's internal physiology and clinical presentation. The expert simply knows more defining characteristics of a disease (Rosch and Mervis, 1975). In some of our own work, we have found that expert physics problem solvers actively use "transformed" or "abstracted" features of a physics problem statement that novices do not even seem to recognize (Chi et al., 1981).

In Case 1 of the present study, there were some inexperienced subjects who did not "pick up" any aortic stenosis hypothesis until after the presentation of the critical finding of "no click." The fact that they did not return to this finding after the aortic stenosis model was engaged suggests they may have had *no* expectation regarding a click. Recall that in Case 2 of the present study some inexperienced subjects seemed to view the pulmonary stenosis issue (Table 12-6) as involving only one dimension, that is, orifice size, when in fact the problem involves the two interacting feature dimensions of size and flow. This is highly reminiscent of the "dimensional restrictiveness" or paucity of encoded problem features reported by Siegler (1976; 1978) for inexperienced problem solvers.

As an individual encodes more features of a disease, this provides opportunity for discriminating the disease into subtypes, that is, variants that differ on a particular feature (Anderson et al., 1979). As an illustration of what we mean, if a person encodes only the features of height and weight for people, he or she is quite limited in the discriminations he or she can make among people. It is clear that the disease knowledge of the highly experienced diagnostician is highly differentiated within a disease type. In the present study the case explicitly designed to assess this was Case 1, where the increasing differentiation was demonstrated. It can be noted that for Case 2, TAPVC, expert E3 raised and considered no fewer than ten different subvarieties of TAPVC, where each of these was distinguished by slight anatomical difference.

The differentiation of disease knowledge aids the development of precision in the clinical expectations associated with any particular disease model. If possible distinctions among versions of a disease are not made, that is, if they are in a sense all seen as the same thing, then the associated variability in clinical manifestations among patients will be great. However, when an expert represents in memory, say, ten different versions of TAPVC, with each of these perhaps differentiated into more specific versions by severity and age of presentation in a child, then the clinical expectations associated with each of these "micro-models" can be highly specific.

Precise clinical expectations, in turn, contribute to precise rules of evaluation for patient data. This is the difference between the "left-axis deviation" rule used by less experienced subjects in Case 4 and the experts' "down around zero or slightly negative" rule used in evaluating the EKG axis in that case with respect to tricuspid atresia (see Table 12-10 and the discussion about it). Again, in Case 1, one can see a nice example of how differentiation of a disease contributes to correct evaluation. In the protocol given in Figure 12-4, expert E3 raises the one micro-version of valvular aortic stenosis in which a click is not expected. This is the version with a pressure gradient between the left ventricle and aorta (over the valve) of greater than 100 mm, that is, "aortic stenosis of a very severe degree." Under this version, other data of the case would have been different from those presented. The expert was able to bring the appropriate (i.e., moderate severity) version of valvular aortic stenosis to bear on the evaluation and to reject it.

The embellishment of the feature set in disease models aids generalization as well as discrimination. Every additional feature represented for a disease is a potential feature of similarity with another disease; hence the potential of a generalization to "diseases that share feature *x*" exists (Anderson et al., 1979). The LCS analyses throughout this paper are taken as evidence that such groupings are pervasive in the more experienced knowledge base.

Students and novices learn some disease groupings directly (Moller, 1978, p. 46). These, like other teaching concepts, might be thought of as a set of "starting-point" disease categories. With experience and embellishment of feature sets, a diagnostician augments this initial set, often creating useful categories that "cross over" the original classic set. Case 2 from the present study is a good example. One might wonder how it is that a number of subjects on this case could generate and consider extensively the hypothesis of partial anomalous pulmonary venous connection, and never once even think of the correct disease, *total* anomalous pulmonary venous connection, a disease that even in its name is so similar. In the classic categorization of diseases, PAPVC, ASD, and ECD, three members of the LCS for this case, all go together in a category of "acyanotic heart diseases" (see Figure 12-10), while the final LCS member, TAPVC, is in a different category, "cyanotic heart diseases." One explanation for these subjects is

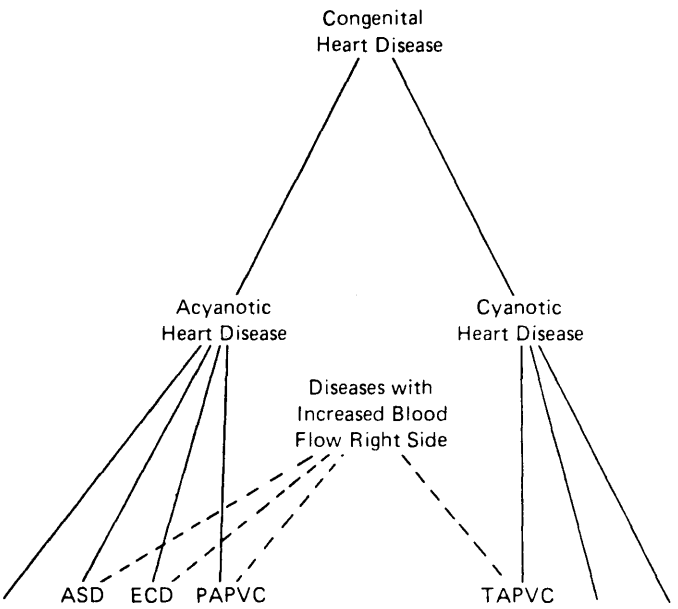


FIGURE 12-10 The classic categorization (solid lines) of the members of the logical competitor set for Case 2 and the expert regrouping (dashed lines) of these diseases.

that they became "stuck in a chunk"; that is, they were in the wrong branch of their classic hierarchy and were not able to benefit from associative triggering or hypothesis activation. The two high-level experts, on the other hand, had created a category for the LCS members that crosses the classic categorization scheme (see Figure 12-10). Creation of this category required them to represent a new disease feature, the feature of "increased blood flow on the right side."

The speculation is that many kinds of logical groupings exist for the expert, tailored to different problem contexts and even different phases of data collection, for example, "the not too sick two-day-old child" in the very early phases of diagnosis. The totality of these groupings for the expert need not be strictly hierarchical; that is, the groupings "cross over" each other in many different ways, forming more a lattice structure than a formal hierarchy (Pople, 1977).

The pervasiveness of groupings in the expert is a logical extension of the general "perceptual chunking hypothesis" of Simon and Chase (1973) and all of its ramifications (Chase and Chi, 1980). The cognitive "chunks" for an environment that people create with experience are those that serve their goals for functioning in that environment [see Egan and Schwartz, (1979) for "electronics trouble shooters"].

12.6.2 Knowledge and Problem Solving

One of the issues we set out to address with this study was the relationship between knowledge and general problem-solving processes. One way to address this issue is from a framework for problem-solving processes set out by Newell (1969). Newell proposed a power-generality dimension for problem-solving procedures. General procedures (weak methods) are those that apply widely, but offer little guarantee of success. Examples are means-ends analyses and "hill climbing." Powerful procedures (strong methods) are those that have well-specified conditions that must be met for their applicability, and hence are tailored to particular closed environments. An example is the formula for solving quadratic equations. Our work and that of others (Elstein et al., 1978) has shown that the *general* problem-solving procedure for diagnosis is one of hypothetico-deduction and that all subjects, regardless of experience, share this general approach. However, the present study has shown that this alone will not get one very far. The general process must be backed up by a rich body of accurate, well-organized medical content.

As problem-solving research has moved from semantically "lean" domains, for example, various toy problems such as the "Tower of Hanoi" and "cryptarithmetic" (Newell and Simon, 1972), to semantically rich domains, such as physics or "engineering thermodynamics" (Bhaskar and Simon, 1977), the role of domain knowledge has become increasingly important as a supplement to general procedures. We speculate that with development of disease knowledge as outlined above, corresponding sets of more powerful procedures, in Newell's sense, are concurrently created. Hence we would propose that as the diagnostician establishes various partitionings of the disease space, for example, the logical competitor sets of various kinds, he or she also establishes associated strong "local" procedures for working within abstracted regions of the space. This would mean, for instance, that the experienced diagnostician would have relatively intact or readily assembled "plans" (Sacerdoti, 1977; vanLehn and Brown, 1979) or "scripts" (Schank and Abelson, 1977) for discriminating hypotheses within conceptual groupings of various kinds and levels of generality.

While related domain knowledge is clearly critical to high-level skill in problem solving in any complex domain and, in particular, in medical diagnosis, this is still not the whole story. Knowledge must be utilized appropriately in the particular contexts where it is needed. What is happening when less experienced subjects fail to consider hypotheses (especially good ones) or evaluate data items poorly? One explanation is that knowledge is stored in memory incorrectly or not stored at all (knowledge "voids"). Another explanation concerns problems of access; subjects simply do not retrieve knowledge they need or retrieve it in some faulty manner.

Postexperimental discussions with the subjects from this study indicated that most subjects, when they failed to generate particular hypotheses or interpreted items poorly (e.g., the click in Case 1), "knew better" in some

sense. Under conditions outside the diagnostic task they could discuss subvalvular aortic stenosis or the import of the click in valvular aortic stenosis, etc. One subject called the experimenter on the day after his session, in which he had erroneously diagnosed Case 2, to tell him that the correct diagnosis had "dawned on him in the shower."

Psychology has long known that the ability to access and use knowledge that one "has" is situationally dependent (Melton, 1963; Tulving and Pearlstone, 1966). For example, knowledge that medical subjects might display on a paper and pencil test is not necessarily what they could display "on-line" in the diagnostic setting. (It was for this reason that the current study, despite its interest in knowledge, was conducted in a diagnostic context rather than in some other manner.) Yet it is this task-accessible knowledge that is crucial to successful performance.

To the extent less experienced diagnosticians have knowledge access problems, several implications for training would seem to follow: First, a disease, other diseases likely to be confused with it in a diagnostic setting, and cues for the grouping should be emphasized together in instruction and, to the extent possible, in the clinical experiences of the diagnostician in training. This encourages the memory unitization of these diseases in categories or other kinds of connected knowledge organizations. Unitization is a hedge against oversight since information in a unit has two modes of "on-line" access, associations from external events and activations directed by the unit itself (Anderson, 1980; Cohen, 1966). Because real clinical experiences are somewhat constrained by the distribution of patients in the training setting, simulated diagnostic encounters (McGuire and Solomon, 1971) could provide a vehicle for augmenting natural experience and for packaging prespecified sets of experiences. Second, tutorial instruction in the diagnostic process itself must attempt to interact with the "on-line" thought processes of the learner as he or she engages in diagnosticlike tasks. This is to help ensure that what is to be taught will be connected both to the situational cues and to the state of active memory likely to exist at some later time when the new material will be needed during a real diagnostic encounter. Expert-based instructional devices (computer-assisted instruction or decision-support systems) that contain expert knowledge and are capable of performing diagnosis in an expertlike manner could provide diagnostic practice exercises in which the device diagnoses a case in parallel with a "student," prompting alternative hypotheses when they are overlooked, correcting erroneous interpretations, and offering instruction when this seems necessary (Brown et al., 1975; Clancey, 1979c; Swanson et al., 1977; Johnson et al., 1979a) (see also Chapter 11). Finally, it would be advantageous if much of the learning of medical content for those in training could be tied as closely as possible to its conditions of ultimate use. "Problem-based learning" approaches to medical education (Barrows and Tamblyn, 1980) seem the prototype of such an endeavor. Under this type of program, much of the basic medical sub-

ject matter (e.g., physiology) that a student learns is organized within representative professional problems, including diagnosis. The problem directs what is to be learned.

12.6.3 Future Directions

Several directions for future research are suggested by the current work. The first of these is the problem of knowledge access and knowledge use. Not much is currently known about the structure of the knowledge base in memory that facilitates its situational use. Yet this is clearly a critical issue in problem solving within semantically rich domains. A second important focus is to investigate the "local" procedures or "scripts" that competent diagnosticians associate with the various partitions of the disease space that they recognize, for example, various disease and problem categories at different levels of generality such as "admixture lesions" or even "the healthy-appearing five-year-old." This appears to be the most promising avenue for studying the *procedures* and strategy of diagnosis that have hitherto been studied only at their *most* general level, that is, at the level of hypothetico-deduction. This will require a better mapping of the types of diagnostic partitions good diagnosticians use—where the current study is only a start. Finally, the current study can be viewed as one step in a cyclical research paradigm that involves experimentation and more formal cognitive simulation. The Minnesota Diagnostic Simulation Model (Swanson, 1978; Swanson et al., 1979) is a model of the expert, and its initial version was built based on studies similar to the current one. As a result of the present study, adjustments and additions to the initial expert simulation model have been made. In addition, the framework now exists for the creation of a more novice simulation. This may enable the study of learning mechanisms (Anderson et al., 1979) responsible for the transition from "noviceness" to expertise. The simulations will also direct a new cycle of more focused experimentation.

It is hoped that the present study provides some guidance for the study of problem solving in semantically rich domains. Such work requires both task-environment and knowledge-base analysis and the creation of problem-solving environments that make the interaction between the problem's information structure and the solver's knowledge structure comprehensible to the observer.

ACKNOWLEDGMENTS

The work on which this paper was based was finished while Feltovich was at the Learning Research and Development Center, University of Pittsburgh, and was supported in part by the Personnel and Training Research

13

Knowledge Organization and Distribution for Medical Diagnosis

Fernando Gomez and B. Chandrasekaran

During the mid-1970s, an AIM research group directed by Professor B. Chandrasekaran was initiated at Ohio State University. Fernando Gomez was a graduate student at the time and was involved in the group's work on MDX, a program for the diagnosis of liver disease. That system provided an experimental environment in which many of the ideas expressed in this chapter were developed.

In this paper, Gomez and Chandrasekaran adopt an analytical view for studying the nature of medical knowledge. Rather than saying "It's all a bunch of random heuristics," they try to formalize the rich structures that make efficient diagnosis possible. They center their representation around concepts, *such as diseases and their causes, in the form of a hierarchical structure similar to a botanical or zoological classification. The key idea is that an expert diagnostician's knowledge is distributed through this hierarchy. Besides being of value for formalizing knowledge in an expert system, this perspective is of value for teaching. Specifically, a student needs to learn this refinement structure for focusing on and further specifying diagnostic hypotheses. The chapter also proposes a useful framework for viewing knowledge interaction in terms of communication via a* blackboard model, *a knowledge representation and control scheme that was first developed for speech understanding (Lesser et al., 1975). The actual system implemented by Chandrasekaran's group is much simpler, however.*

It should be noted that Gomez and Chandrasekaran are trying to capture the compiled *form of human knowledge and are not advocating that we*

 From *IEEE Transactions on Systems, Man and Cybernetics*, Vol. SMC-11, No. 1, pp. 34–42 (1981).

design expert systems in general by intermixing strategic and domain knowledge (cf. NEOMYCIN's separation of disease knowledge from domain-independent meta-rules, Chapter 15). Nor are they claiming that experts do not use general principles for ordering search and selecting alternatives (cf. Swartout's "domain principles," Chapter 16). Rather they are emphasizing that other constructs, in addition to rules, are needed to organize knowledge. Explicating the hierarchical structure of hypotheses and findings implicit in pure rule systems improves system organization for focused reasoning as well as ease of system building [cf. Aikins's "prototype hierarchy" (Aikins, 1980)].

The reader may be interested in pursuing a number of related AI topics, such as studies of epistemology and natural language understanding that are referenced in this chapter.

Concepts lead us to make investigations, are the expressions of our interests, and direct our interests.

Wittgenstein, *Philosophical Investigations*, prop. 560

13.1 Introduction

What are the criteria that should be used to organize the medical knowledge in an automated medical system? We start with the observation that diagnosticians, when they arrive at a diagnosis or diagnoses, have invoked some concepts. These can be diseases, causes of diseases, or other notions that are relevant to the diagnosis. We shall suggest that these concepts form a hierarchical structure similar to that of a botanical or zoological classification. The diagnostician's knowledge is distributed through this hierarchy. The concepts in the hierarchy provide the criteria to organize under them small pieces of knowledge represented in the form of production rules. Thus concepts may be viewed as clusters of production rules. They extend the capabilities of production rules to more complex problem-solving situations. The rules under each concept are further organized into three groups: exclusionary, confirmatory, and recommendation rules.

During the problem-solving process, the concepts can be considered to be *specialists*. They interact and communicate with each other by means of a blackboard, a notion borrowed from Erman and Lesser (1975). In that respect, the ideas presented in this paper can be considered as an extension of the notions of the HEARSAY-II speech understanding system (Carnegie-Mellon University, Computer Science Research Group, 1977) to the medical diagnosis task. Nevertheless, there is an important methodological difference. It is that concepts and not rules provide the principle of knowledge organization.

Section 13.2 contains critical notes on some aspects of knowledge representation. Section 13.3 describes the central features of our ideas on the organization of medical knowledge. Section 13.4 explains the different kinds of rules. Section 13.5 deals with the identity of the notions *concept* and *specialist.* Section 13.6 discusses distributive problem solving. Finally, the paper concludes by indicating some of the similarities and differences between this approach and other medical diagnosis systems.

13.2 On Knowledge Representation

In recent years, there has been much work in knowledge representation in artificial intelligence, but relatively little attention has been paid to how knowledge is used and organized. By *use of knowledge* we mean the invocation and instantiation of the right chunk of knowledge and the determination of the appropriate structure of the knowledge needed for the task being studied. Other authors, especially F. Hayes-Roth (1978), have expressed a similar view.

13.2.1 On the Representation and Use of Knowledge

The assumption that a separation can be established between knowledge representation and its use dates back to the distinction made by McCarthy and Hayes (1969) between epistemologically and heuristically adequate analyses. Underlying this distinction is the belief that the first does not involve the second, and *vice versa.* Most researchers in knowledge representation have, consciously or unconsciously, subscribed to this distinction, which is indirectly related to the Saussorian distinction of *la parole* and *la langue,* better known after Chomsky as the *performance-competence* distinction. The assumption underlying both distinctions is that it is appropriate to study the result of human thought, language, knowledge, etc., by "abstracting out" the homunculus that is using that thought. Both distinctions are influenced by the paradigm that modern logic brought to the study of linguistics and epistemological questions. While logic is no longer a dominant paradigm in AI, much research in knowledge representation nevertheless has concerned itself with the so-called epistemological adequacy of the representation, thus deepening the separation between knowledge representation and its use. In particular, while many of the current techniques of knowledge representation in AI arose as components of localized models

of human cognition, the emphasis has increasingly been on the *formalisms* in the models.

13.2.2 Content and Form in Knowledge Representation

The semantic network (Quillian, 1968) was proposed as a model of semantic memory. But since Quillian's original formulation, the formalistic aspect of it has gained a life of its own—so much so that much of the research in semantic networks scarcely differs from the logic formalism. Recently, some researchers have shown interest in the foundations of semantic networks (Woods, 1975; Brachman, 1979). Important distinctions have been made explicit, but no connection has been established between the proposed improvements to the representational formalisms and the use of the knowledge. It is unclear how the notational inventions will help in the understanding of the task being studied.

Since Minsky's (1975) important paper about frames, little progress has been achieved in extending his ideas, but formalisms (Goldstein and Roberts, 1979; Bobrow and Winograd, 1977) have been built on the outline proposed by Minsky. Minsky revived and began the task of giving computational meaning to an interesting theory of human cognition. The theory says that important aspects of vision, memory, problem solving, and comprehension can be explained as a process of *recognition.* In this process, the input is matched to an internal stereotyped structure called a *frame,* slots in the structure are filled, and others take default values. The notion of a default value was one of Minsky's most insightful ideas. Information not explicitly present in the input could be accounted for by reading the default values.

Frames have proved to be an excellent construct for dealing with extralinguistic knowledge in language. Other authors independently worked out a similar notion called a *script* (Schank and Abelson, 1977). In its representational aspect, frames are an extension of the property list notation. They look very much like a COBOL structure. But just as COBOL programs using structures are not exemplifications of the frame theory, neither are AI programs just because they happen to be written in a frame-type language. Otherwise, one would be confusing the form with the content of the theory. It is precisely the theory that needs to be extended. We know very little about the criteria that govern the recognition of frames (Charniak, 1978), the invocation of the appropriate frame, and the integration of frames in more inclusive structures . The available formalisms inspired by the frame theory, while they differ in the degree of their concern with such issues—FRL (Goldstein and Roberts, 1979) is meant as a programming language, whereas KRL's authors (Bobrow and Winograd, 1977) have shown a great concern with extending and giving depth to the

frame ideas—nevertheless do not provide answers to these questions. [See Lehnert and Wilks (1979) for a sympathetic critique of KRL.]

13.2.3 Production Rules and Organization of Knowledge

In expert knowledge domains, production rules have been extensively used. Despite their apparent simplicity, production rules grasp an important aspect of human cognition. But it seems to us that they have sometimes been used unilaterally to explain cognitive aspects for which other constructs are needed. They were used by Newell and Simon (1972) as techniques to model some aspects of human problem solving. Since this early and seminal work, production rules have been applied to almost every aspect of human cognition. [See Waterman (1978) for an excellent collection of papers about production systems.] Two of the most successful systems, DENDRAL (Buchanan et al., 1969) and MYCIN (Shortliffe, 1976) use production rules as the basic technique to represent knowledge.

Production rules are the right tool to represent some kind of *how-type* knowledge. Winograd (1975) refers to it as *secondary* knowledge. Other authors have used the term *judgmental* (Duda et al., 1978). Some aspects of the knowledge needed to repair a car, to diagnose a disease, etc., are of this type. In the medical diagnosis domain, there are many terms of which a doctor does not need to have a thorough understanding. The knowledge that a diagnostician has that tells him or her that certain lab findings are indications of a certain disease is of that kind. In domains where the required level of comprehension is deeper, for example, natural language understanding, the need for *what-type* knowledge has become apparent. Riesbeck's parser (1978) is a very good example of the integration of the two kinds of knowledge: production rules are used to build and to predict Schank's conceptual dependency structures. But the production rules are embedded in the conceptual structures.

The problem of organization of knowledge is of paramount importance in large knowledge base domains. The problem is not only one of efficiency, but one of focus and control. Things simply do not work if the knowledge is not properly organized [see Lenat and Harris (1978) for a discussion of these problems]. The solution generally offered by the advocates of production rules is a new rule called a *meta-rule* (Davis and Buchanan, 1977). Meta-rules organize the production rules according to some meta-knowledge criterion. While production rules are natural mechanisms to model an important aspect of human cognition, meta-rules seem to be an *ad hoc* solution to the problem. It is doubtful that they have any cognitive counterpart. We think that for an appropriate organization of knowledge, another construct, in addition to the rule, is needed. In the following pages we will show that even in domains in which the knowledge is basically how-type, concepts and not rules should provide the principles of organization.

13.3 The Role of Concepts in Medical Diagnosis

In a recent paper about AI work on natural language, Fodor (1978) has characterized it as suffering from "operationalism, empiricism. . . . " On the other hand, if an empiricist looked at the AI work, of course, beyond Winograd's publications in 1971, he or she would consider it to be "irresponsible talk" about concepts. The existence of concepts and the need to take them seriously are almost granted, in particular by researchers in natural language understanding and in knowledge representation. The representation proposed for concepts has been a what-type of structure called a *frame* in FRL (Goldstein and Roberts, 1979), a *prototype* in KRL (Bobrow and Winograd, 1977), and a *concept* in KLONE (Brachman, 1979). In this paper, we will be speaking of concepts in a different sense, not as what-type structures but as labels that organize how-type knowledge represented in production rules. They can be considered as clusters of production rules.

13.3.1 Concepts and Organization of a Diagnostician's Knowledge

Consider the following production rule: if bilirubin in urine and pruritus, then suggest cholestasis. A diagnostician has thousands of such rules. In our view they are associated with concepts such as "arteriosclerosis," "hepatitis," "cholestasis," etc. These concepts themselves form a hierarchical structure similar to that of the botanical or zoological classification system. The most general concepts are placed at the top of the hierarchy and the most particular at the bottom (see Figure 13-1). Knowledge is distributed through this hierarchy. The structure serves the purpose of differentiating the knowledge, of assimilating new knowledge by inserting it in the appropriate place, or of retrieving the right piece of knowledge as a response to the appropriate query.

For diagnosticians, this hierarchy serves the function of organizing their troubleshooting knowledge. The concrete details for each disease are encoded in the production rules attached to the appropriate concepts. However, it is clear that medical doctors also have additional cognitive structures that organize their knowledge from other views: pathological, physiological, etc. The role of these additional structures during diagnosis then becomes a relevant issue. The cognitive structures corresponding to these other views do not need to be present for purposes of diagnosis, as long as knowledge from these structures relevant to diagnosis is compiled in the diagnostic structure. This can be done by appropriately structuring the relevant concepts and embedding the compiled production rules therein.

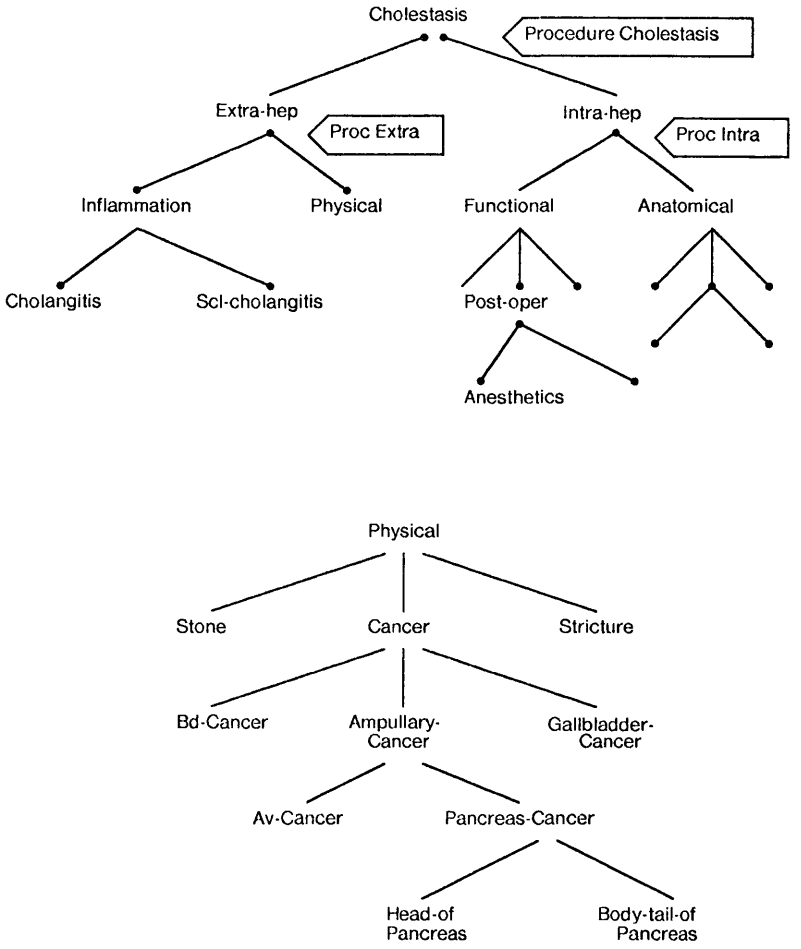


FIGURE 13-1 Conceptual structure of cholestasis.

13.3.2 Commonsense Knowledge

The role of commonsense knowledge structures is of equal interest. A distinction must be made between (a) the commonsense knowledge a physician needs in order to understand the data presented in a medical case and (b) that needed during the process of diagnosis. The patient data are entered in current AI programs for medical diagnosis in the form of a collection of atomistic facts, e.g., "high bilirubin, fever, jaundice." In contrast, consider the following:

> At the age of 19 years, one year prior to his appendectomy, he began to have occipital headaches, usually upon awakening in the morning and occurring once or twice weekly for a 10-year period. These headaches had not been severe enough to interfere with his activities . . . [taken from Harvey and Bordley (1972)].

Here we have a complex temporal interconnection of facts that cannot be decomposed into simple facts. It may be true that in most cases the atomistic collection of data may contain sufficient manifestations to make a correct diagnosis. However, for those cases in which comprehending the complex structure of data is essential to a solution, systems whose data input is atomistic will miss the solution. In order to uncover the needed structure for data input in these cases, it is necessary to make a semantic analysis of the commonsense notions of time and causality in this context. For simple instances of temporal and causal notions, temporal cases could be enough, for instance, structures like "< > while < >," "< > after < >," and "< > causing < >." But the kind of semantic notions and the mechanism needed to integrate these simple structures in more inclusive ones like those needed to understand *the course of the illness* need further study.

Let us consider (b), viz., the use of commonsense knowledge during diagnosis to verify or reject hypotheses. Suppose a doctor has established that a patient has hepatitis and is proceeding to find out the possible causes of the disease. Let a piece of data be "the patient is a farmer." The doctor can bring to mind the knowledge that farmers often drink water from wells and that the patient may have contracted a viral infection from drinking the water. Notice that in this case the piece of world knowledge "farmers usually drink water from wells" was only activated in the context of diagnosing the cause of hepatitis. The datum "the patient is a farmer" would not play any role and thus might have been unnoticed in the context of some other diseases. Medical knowledge has this type of knowledge embedded in it. The right medical context activates this knowledge, which can hence be easily compiled in the form of production rules. In particular, the production rules will be inserted under the concept "virus infection as a cause of hepatitis," explicitly checking whether the patient has been ingesting contaminated water.

The foregoing should not be interpreted as denying that, for a complete model of a physician's reasoning, physiological, anatomical, and commonsense knowledge structures need to be represented in addition to diagnostic knowledge. There is no doubt that a physician uses these other structures and that what-type knowledge must be captured in them. They are needed in order to acquire new pieces of judgmental knowledge, to reconfigure and extend the concepts in the diagnostic structure, and to do productive problem solving involving knowledge in these other domains (which may result in compilable production rules to be added to the diagnostic structure). However, for achieving expert diagnostic performance, we do not believe that these additional structures are needed.

13.3.3 Redundancy and Biasing of Knowledge

The above considerations point to the view that the resulting knowledge structure for the diagnostician must be biased by the function that it is meant to serve. This means that the concepts that make up the structure and their organization are determined by the fact that they are grasping the medical knowledge of a diagnostician and not that of, say, a pathologist. The organization of medical knowledge from a pathologist's point of view will call for a different set of concepts and a different organization in the structure. Similarly, the knowledge encoded under each concept must also be biased toward the diagnostic task. The knowledge diagnosticians have about stone, tumor, etc., is only that necessary to establish them or rule them out in the context of liver diseases. However, the structure does not have to be just a classification of diseases. Other concepts that are not names of diseases may appear in the structure, to the extent that these concepts are needed to properly diagnose a disease. For example, in the structure of Figure 13-1, the concepts "stone," "cancer," and "stricture" are causes of a disease and not themselves diseases.

An organization of knowledge following these principles results in a high level of redundancy. Small pieces of knowledge in the form of production rules will appear grouped under different concepts. Also, the same concept may appear inserted in different places in the cognitive structure of the diagnostician. But the production rules grouped under the concepts will have differences reflecting the differences in the roles of the concept. An example of this occurs in the cholestasis syndrome. Consider the concept "stone" in its role as a cause of cholestasis. There will be production rules to establish or reject stone here, and also to check if a stone is indeed causing obstruction. However, stones may not necessarily cause obstruction directly, but may result in cholangitis. "Stone" would also occur as a concept under "cholangitis." This concept, while sharing some of the same production rules with the other stone concept, nevertheless will also have some rules that are different, because of the different role of this stone concept.

13.4 Kinds of Rules

Three types of rules must be grouped under each concept: confirmatory, exclusionary, and recommendation rules.

13.4.1 Confirmatory Rules

Confirmatory rules look for those manifestations associated with the concept under which they are located. Those manifestations could be sufficient

to establish the concept completely or only enough to postulate it. These rules return a list of the findings on which they establish or postulate the concept.

13.4.2 Exclusionary Rules

The need for exclusionary rules has been recognized. For instance, Pauker et al. (see Chapter 6) use exclusionary rules to reject a disease categorically. In our approach, they are used in a more inclusive sense. They collect all of the negative evidence for a given disease. The evidence does not have to be sufficient to rule out the disease. An obvious reason to have such rules is that physicians need to rule out certain diseases before performing some invasive procedures such as biopsy. A more interesting reason is that doctors frequently use a ruling-out problem-solving strategy. This happens when the data suggest several diseases and there is no conclusive evidence for any of them. Then the strategy consists in ruling out those diseases with the lowest evidence and focusing on the remaining. The use of ruling-out rules is the key methodological principle of differential diagnosis as explained by Harvey and Bordley (1972). The strategy adopted throughout their book is the following: once hypotheses are postulated to explain a given disease, a procedure is invoked that systematically begins to rule out some of them. For instance, in the discussion of a case of splenomegaly (pp. 371–376), they establish up to seven possible major hypotheses: polyarteritis, systemic lupus enterocolitis (SLE), lymphoma, etc. Immediately they say:

> Polyarteritis is rarely associated with very significant splenomegaly, and the arterial lesions should have been seen. Arteritis can occur at all levels and may simulate almost any bowel disease, but at some stage bleeding is usually noted. None of the other clinical manifestations which suggest polyarteritis were noted.

Notice that the reasoning is based on highly categorical production rules. These ruling-out rules are tried even for those hypotheses that will eventually be accepted as explanations of a disease; that is, an attempt may be made to refute a hypothesis even in the presence of positive evidence for it. Nevertheless, it would be incorrect to see this practice as an exemplification of Popper's principle of refutation, viz., hypotheses are not verified but refuted. This is because the clinician not only considers the negative evidence for a given disease, but also the positive evidence. In our opinion, he or she can be viewed as running two procedures. One collects the evidence in favor of a given disease, the other the negative evidence. Then both are weighed, and a decision is made.

13.4.3 Recommendation Rules

Although the knowledge under each concept is mostly the knowledge needed to establish it or rule it out, there are pieces of knowledge under each concept that anticipate its subconcepts. For instance, jaundice, intense and intermittent abdominal pain, and elevated alkaline phosphatase strongly suggest not only a liver disease but also extrahepatic obstruction (one of the liver subconcepts). These pieces of knowledge will be translated during the problem-solving process as "recommendations" of a superconcept to its subconcepts. This knowledge will be represented in production rules that will be applied to the list of positive manifestations found by the confirmatory rules.

13.5 Concepts as Specialists

Given these principles of organization of medical knowledge, the solution of a medical case becomes a problem of taxonomic classification. It is similar to the problem of placing, say, a specimen of maple in a hierarchy of botanical concepts. It consists of identifying its superordinate and subordinate concepts. This is a process of recognition that is intrinsically top-down. Let us consider the use of knowledge in the context of the representational framework we have proposed. The solution consists of taking each concept in the structure as a specialist in that body of knowledge. Each concept interacts with others in the solution of a case by activating simultaneously each subconcept under conditions we explain below.

The decomposition of a body of knowledge into small systems is an old idea in AI. It was one of the central notions in Simon's beautiful book *The Sciences of Artificial Intelligence* (Simon, 1969). Winograd (1972) used the term *specialist* to refer to his semantic program for the noun group. Lenat (1975) used the notion extensively in his notion of "Beings." Rieger and Small (1981) are building a "word expert parser," and Minsky has recently speculated about a "society of minds" (Minsky, 1979).

The idea of viewing the human mind as a society of experts is very attractive. It has its counterpart in the human body system with its multiplicity of functions going on in parallel. The notion of specialist or expert is another metaphor that AI has borrowed from computer science. Any program consisting of a collection of modules or routines can be viewed as a collection of experts. Then the following question arises: what contribution is made by calling them experts? It seems to us that, in order for the notion of the society of experts to be useful, (a) we need criteria for the decomposition of a body of knowledge into small independent units, (b) the decomposition should be such that it will be able to support par-

allelism, and (c) communication and control should resemble those found in human intelligence.

In the case of medical diagnosis, the identification of these components is facilitated by the fact that the medical community is organized as a society of experts. The solution of a medical case requires in many instances the interaction of several specialists. The concepts that make up a diagnostic specialty have already been identified by the medical community. Nevertheless, the right structure and the precise interdependence of concepts for a given disease are by no means clear. One can be easily convinced of this by the fact that often different books about diagnosis do not coincide in the decomposition of the relevant concepts that need to be considered to properly diagnose a disease. The mapping of the medical knowledge as it appears in books into a structure like the one we are proposing is by no means automatic. Epistemic work is needed in order to come up with the right structure and the concepts that form the structure. It is our deep conviction that if automated medical diagnosis succeeds some day, books on medical diagnosis will be written in a very different form from that of the current texts.

13.6 Distributive Problem Solving

A distribution of the diagnostician's knowledge in a hierarchy of concepts, which are considered as independent specialists in that body of knowledge, leads naturally to a distributive problem-solving situation. In order to illustrate this, we recapitulate our basic ideas by considering the medical knowledge of an internist.

Referring to Figure 13-2, we see that the top-level node has no rules in it since it is always established. Its immediate successors are formed by generic diseases such as those related to liver, heart, etc. Under the concept "liver," our internist will have that knowledge needed to determine if a given patient has a liver disease. That specialist will look for those key findings associated with liver diseases, for instance, abdominal pain, jaundice, alcoholism, etc. Also, the specialist will have knowledge to rule out a liver disease and to make some recommendations to its subconcepts. But it will not have the knowledge to discriminate between the different kinds of liver diseases. That knowledge will be located in the two concepts under it, the intrahepatic and extrahepatic specialists.

Pathognomonic knowledge is useful not merely in establishing the concept under which it is located. In medical diagnosis, pathognomonic manifestations are those that indicate the presence of a disease with near certainty. If a concept has pathognomonic knowledge about a successor concept and if the corresponding manifestations are present, control is

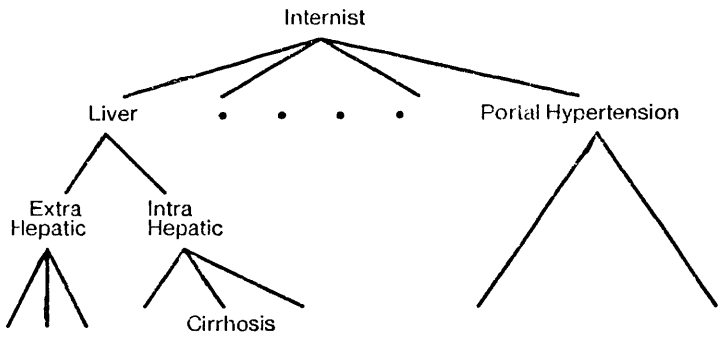


FIGURE 13-2 Top levels of the internist's conceptual structure.

transferred to the successor even if it is located several nodes down the tree.

If we consider the internist's knowledge organization, the nodes in the hierarchy are called *concepts*. Because these knowledge sources interact with others to solve a medical case, they are called *specialists*. Finally, because these concepts are names of diseases that must be verified or rejected, they may be called *hypotheses*. We use these three terms interchangeably in the remainder of the paper. A blackboard will be used to coordinate the work of the specialists in the solution of a case.

13.6.1 The Blackboard

The notion of a blackboard was used by Erman and Lesser (1975) as a way to provide an interface between different knowledge sources. The function of the blackboard in our design is similar to theirs: to provide a way of interaction among the specialists and to hold the current state of the system. The blackboard is divided into the following sections. ACTIVE-HYPOTHESES contains the names of all specialists that are active at a given moment. ESTABLISHED-HYPOTHESES contains the names of all hypotheses that have been established during the solution of a case. A hypothesis is established when the evidence exceeds some threshold. There could be cases in which the evidence in favor of a hypothesis is sufficient to categorically establish it, while in other cases the evidence could be only sufficient to postulate it. REJECTED-HYPOTHESES contains the hypotheses that have been rejected. SUSPENDED-HYPOTHESES contains all hypotheses for which a specialist has not found sufficient evidence to justify pursuing them. This section also includes those hypotheses that were initially postulated but later on abandoned because the evidence did

not exceed the threshold needed to pursue them. Finally, it should be noted that as hypotheses are entered in the various sections of the blackboard, the underlying hierarchical structure among them is preserved.

13.6.2 Activation

We can now consider the activation of the specialists. Once the top-level node is invoked, it activates simultaneously all its immediate successors and enters their names in the ACTIVE-HYPOTHESES section of the blackboard. These act in parallel on the patient's data base. These will look for those manifestations in the patient's data base that are associated with the generic concept they stand for. We can distinguish the following cases:

Case 1. A specialist, say S, can find data to consider that the disease it stands for must be pursued. If so, it will enter in the ESTABLISHED-HYPOTHESES section of the blackboard its name followed by a list of the manifestations on which it based its decision. Then it will activate its immediate successors (if some of the pathognomonic rules have been fired, it could activate some subspecialist down the tree). Upon their activation, S will pass to them the same information it entered in the blackboard plus a list of "recommendations." Finally S will deactivate itself by removing its name from the ACTIVE-HYPOTHESES section of the blackboard. The list of recommendations will contain pieces of advice about such aspects as what kind of rules (disconfirmatory or confirmatory) a given specialist should try first, indications to discourage the subspecialist to do an extensive search, etc. The type of recommendations depends on each disease. They provide another criterion to further organize the production rules under each specialist.

Each specialist, on establishing itself, will add to the list of manifestations, which then will be passed from parent node to child until it reaches a tip node. If the specialist in the tip node succeeds, it will print the list. At that point, the list will contain a classification of the medical case under study. The list could look like:

(Liver (A_1 A_2 ... A_n) Extrahepatic (B_1 B_2 ... B_n) Tumor (C_1 C_2 ... C_n))

where A_i, B_i, and C_i are the manifestations on which each specialist based its decision.

Case 2. A specialist rejects itself. This happens when the exclusionary rules found the presence or absence of data sufficient to rule out the disease. In that case the specialist enters its name in the REJECTED-HY-

POTHESES section of the blackboard followed by a list of the negative evidence and deactivates itself.

Case 3. A specialist suspends itself. It then enters its name in the SUSPENDED-HYPOTHESES section of the blackboard followed by the list of manifestations before suspending itself. The suspension of a specialist can happen because the data it found did not exceed some threshold or because its immediate successors rejected or suspended themselves.

In cases 2 and 3, when the immediate successors of a node have rejected or suspended themselves, a mechanism has to be provided to remove that specialist from the ESTABLISHED-HYPOTHESES section of the blackboard. This can be accomplished by making the last active sibling (if it has suspended or rejected itself) check if any of its other siblings are in the ESTABLISHED-HYPOTHESES section of the blackboard. If none of them is there, it means that all of them have rejected or suspended themselves. In that case, the specialist will move its parent from the ESTABLISHED-HYPOTHESES section of the blackboard to the SUSPENDED-HYPOTHESES section. After that, it will check to see if none of its uncles is in the ESTABLISHED-HYPOTHESES section. If none is there, it will remove its grandparent, and so on.

13.6.3 The OVERVIEW Critic

It is generally accepted that a good practice in the diagnostic process is to explain again all the patient's manifestations from the point of view of the final diagnosis or diagnoses. Harvey and Bordley (1972) considered this to be the final step in the diagnostic process. In our approach, we need to organize a body of knowledge around that methodological idea. This is due to the fact that quite a few suspended hypotheses could result during the diagnostic process. They should be cleared, resulting in a more unified diagnosis. For this purpose we associate with each disease in the top level of the hierarchy an OVERVIEW critic.

OVERVIEW is activated only if the disease with which it is associated is advanced as one of the diagnoses. Basically, what OVERVIEW will do is to check if those manifestations that the specialists entered in the blackboard with each suspended hypothesis appear in the list of manifestations associated with any of the subspecialists of the disease that has been established. If all manifestations associated with a suspended hypothesis can be accounted for by this procedure, OVERVIEW will reject that hypothesis. Otherwise, it will advance that hypothesis as a second or third diagnosis. If the only function of OVERVIEW were this procedure, then it would not have to be associated with any particular disease. We feel, however, that other questions should be formulated by OVERVIEW, such as the relevance of the manifestation to the suspended hypothesis in particular

and to the diagnostic process in general and the chances of the appearance of both the suspended hypothesis and the established one. Further investigation will have to be conducted to determine the nature of these questions concretely. We conjecture that OVERVIEW would have knowledge global to the individual subspecialists into which a disease has been decomposed, as well as knowledge about other diseases in the top level of the conceptual hierarchy.

13.6.4 The Specific Role of the Blackboard

The blackboard can serve many functions in our approach to medical diagnosis. It will be a matter of further study to exploit all of its advantages. We can mention two instances in which its use is necessary. The first one deals with the problem of a disease being secondary to another. For instance, cirrhosis (a liver disease) can cause portal hypertension (which can have many other causes). In the medical jargon, it is said that the clinical manifestations of the latter are *secondary* to the former. However, the manifestations of each disease are different. Following our approach, let portal hypertension be a successor of the top node, internist (see Figure 13-2). Both nodes, viz., cirrhosis and portal hypertension, will be established in parallel in a patient with portal hypertension secondary to cirrhosis. At a given point, the portal hypertension specialist will pass control to subspecialists that will determine the possible causes of the disease. Then one of them is going to contemplate cirrhosis as being the cause. That subspecialist can verify this by looking at the blackboard for cirrhosis. Without this blackboard, the hierarchical call structure would be violated by a call to the cirrhosis specialist, or a redundant and *ad hoc* specialist would need to be created.

The second instance has to do with the fact that the specialists must communicate between each other to reduce the amount of search they must do. Consider specialists associated with different causes of the same syndrome. Although it is possible that a disease can have more than one cause, it is not frequent. Then if a given specialist has already found the cause of a disease, it makes very little sense for its sibling to pursue its search in the presence of very low evidence. As a specific example, consider the situation where extrahepatic cholestasis has been established, and each of its immediate successors, stone and cancer, is investigating itself as its cause (see Figure 13-1). As the stone and tumor experts are working in parallel, suppose the preliminary evidence for stone is low, while the tumor specialist establishes tumor. Now the stone specialist should suspend itself, but only if the information about tumor establishment is made available. This can be made possible by making the specialists (in this case the stone specialist) periodically inspect the ESTABLISHED-HYPOTHESES portion of the blackboard.

13.7 Implementation

In the preceding pages we have described a methodology for knowledge distribution and the associated distributed problem-solving strategies for medical diagnosis. There are two key aspects to the methodology: (1) knowledge is decomposed into a collection of specialists, and (2) these specialists perform problem solving in parallel in certain specified ways, using a blackboard as a record of the global state of problem solving.

A prototype diagnosis system called MDX (Chandrasekaran et al., 1979; Mittal et al., 1979) has been built by our group and has been operational for some time. This implementation has both points of contact with and differences from the methodology described in the preceding pages. The major points of contact are that the current domain of MDX, viz., cholestasis, is organized into a collection of specialists as indicated in Figure 13-1 and that diagnostic knowledge is distributed in this structure following the guidelines spelled out in Section 13.3. The problem-solving strategy is the area of most of the differences between the methodology described in this paper and MDX as implemented. The source of these differences is threefold: (1) the strategy in this paper is of more recent origin and goes beyond the current MDX strategy in power; (2) the methodology emphasizes the parallel invocation of specialists, which is of particular importance in a distributed implementation and of less operational significance in a serial implementation such as MDX; and (3) the domain of implementation is not large enough to need the global state record in the form of a blackboard. The power of a blackboard of the type we have envisaged will be needed as the domain is enlarged. In particular, it will be needed for decisions at the top (internist and one or two levels below) where proper coordination between subspecialists of vastly different scope would be needed.

These differences notwithstanding, MDX is a working implementation of a distributed approach to problem solving. As such, a brief outline of its performance is in order. A more complete discussion of the system and its performance is available in the papers cited earlier.

The top-level specialist in the system is GP (or internist), but all that it can do at this stage of implementation is either to hypothesize cholestasis and transfer control to it or to reject the case. Cholestasis may be hypothesized by a collection of production rules that respond to the relevant lab data and physical signs and symptoms. When cholestasis gets control, its charge is first to establish itself and then to further refine itself to account for all the manifestations. This *establish-refine* strategy is fairly general to the system as it currently exists. The rules used to establish cholestasis are of the confirmatory type mentioned in Section 13.4. The disconfirmatory evidence is not currently used in all the nodes, but where it is used it is in the form of negative weights for the disease for certain combinations of data in an evidence-weighting table.

Once cholestasis, say, is established, a priority scheme is needed to call subspecialists, since MDX is a serial implementation. This priority is provided by a collection of rules that suggest possible specialists on the basis of certain patient data or combination thereof. The criterion for the selection of the rules is that they represent common or easy possibilities. If this criterion is satisfied, the specialists that are called earlier by the priority scheme are more likely to solve most of the cases. Only in "hard" or uncommon cases will the rest of the specialists need to be called.

These specialists are typically called to establish and refine themselves and, when they succeed, to return those abnormal data that they can explain. The specialists that are established and the corresponding data are kept in an ACTIVE list. When the specialists in the top-level ACTIVE list together can explain all the abnormalities in a nonoverlapping way, the case is solved. Note that the specialists lower than cholestasis in the hierarchy may also have their own priority rules to select their subspecialists. The tip nodes, when called, match the data within their scope with confirmatory rules or equivalent tables to establish or reject themselves. This information is passed up to the calling specialist. Each specialist thus organizes, by means of production rules, the priority by which it uses its subspecialists to arrive at an explanation of abnormal data in its scope. When the subspecialists explicitly suggested by the rules fail to explain the case, then an exhaustive interrogation of all subspecialists one level below will be made. Thus the priority rules do not preclude the correct answer from being obtained eventually.

As stated earlier, the current implementation of MDX does not use a blackboard. Consider the case involving cirrhosis and portal hypertension that was discussed in Section 13.6.4. In our current implementation, portal hypertension will neither call up the cirrhosis expert nor look up the blackboard. Instead, it will have a cirrhosis-as-cause-of-hypertension subspecialist, most of whose knowledge would be a replication of the cirrhosis specialist. This is clearly an *ad hoc* solution, but as long as the domain is not very large, it does not produce serious problems.

Another constraint in the MDX implementation has to do with the atomistic nature of the patient manifestations (see Section 13.3). The cholestasis domain has so far not produced sufficiently complex cases for which this data representation presents a serious limitation. However, the future extensions of MDX will increasingly incorporate more sophisticated structured data representations as discussed in Section 13.3.

13.8 Concluding Remarks

The ideas presented in this paper contain points of coincidence with other research in automated medical diagnosis. They coincide with MYCIN (Shortliffe, 1976) in taking production rules as the formalism for repre-

sentation, but in our approach rules are organized under concepts. In INTERNIST (Pople, 1977) the hierarchy of diseases is essential to the problem-solving strategy. In our approach the hierarchy is not only of diseases, but also of causes of them and of any concept relevant to the diagnostic process. The concepts in our hierarchy are *specialists,* aggregates of knowledge about a significant step in the determination of the diagnosis. We coincide with CASNET (see Chapter 20) in the relevance of etiologic reasons in the diagnostic process, but in our approach that is one reason among others. The concepts in our hierarchy are highly compiled. Thus some specialists will have etiologic knowledge, while others will base their reasoning on other types of knowledge depending on the disease. Finally, our approach coincides with PIP (see Chapter 6) in taking each disease as a cluster of knowledge with distinct features. But the structure of diseases is a hierarchy in our approach; in PIP it is not.

An important aspect of our ideas is that medical (or for that matter any) knowledge can be viewed as a collection of essentially decoupled conceptual structures, each with an embedded problem-solving mechanism (reflecting its intended use). In the actual handling of a case, a physician is in the diagnostic mode only part of the time. The incompleteness of the diagnostic structure in a particular physician, as well as other considerations involving therapies, costs and other situational idiosyncrasies, and a perceived need for explanation at different levels will typically cause him or her to switch between different knowledge structures, but a satisfactory accounting of this overall process can be done, in our view, only after the underlying conceptual structures and the problem-solving mechanisms implicit in them are identified. We have advanced in this paper an analysis of one such structure, viz., the diagnostic one.

ACKNOWLEDGMENTS

We thank David C. Brown, James Reggia, Jorgen Hilden, Jack Smith, and Sanjay Mittal for their comments on an earlier draft. Our access to the computing facilities of the Rutgers University Laboratory for Computer Science Research, made possible by a grant (RR-643) from NIH Biotechnology Resources Program, Division of Research Resources, has been essential to our implementation activities. Finally, the National Library of Medicine Biomedical Computing Training Grant to The Ohio State University (LM 07023-02) has helped foster an active interest in these and other fundamental problems in medical knowledge representation.

14

Causal Understanding of Patient Illness in Medical Diagnosis

Ramesh S. Patil, Peter Szolovits, and William B. Schwartz

In most medical AI programs, the use of notions such as causal relationships, temporal patterns, and aggregate disease categories has been limited. Yet studies of clinicians' behavior reveal that a diagnostic or therapeutic program must consider a case at various levels of detail to integrate overall understanding with detailed knowledge.

To explore these issues, Ramesh Patil, Peter Szolovits, and William Schwartz have applied the knowledge-based approach in a detailed study of consultation for electrolyte and acid-base disturbances. The resulting program, Patil's dissertation work, is known as ABEL (for Acid-Base and ELectrolyte program). ABEL and an earlier M.I.T./Tufts program known as the Digitalis Therapy Advisor (Gorry et al., 1978) were important departures from other systems in that they both viewed clinical problem solving as a process of constructing *an explanation of manifestations, what they have called a* patient-specific model. *In ABEL, this description includes data about the patient as well as the program's hypothetical interpretations of these data in a multilevel causal network. Proceeding from the lowest level, the concepts and relations gradually shift in content from pathophysiological to syndromic knowledge. The aggregate level of this description summarizes the patient data, providing a global perspective for efficient exploration of the diagnostic possibilities. The pathophysiological description provides the ability to handle complex clinical situations arising in illnesses*

From the *Proceedings of the Seventh International Joint Conference on Artificial Intelligence,* vol. 2, 1981, pp. 893–899. Used by permission of International Joint Conferences on Artificial Intelligence, Inc.; copies of the *Proceedings* are available from William Kaufmann, Inc., 95 First Street, Los Altos, CA 94022.

with multiple etiologies, to evaluate the physiological validity of diagnostic possibilities being explored, and to organize large amounts of seemingly unrelated facts into coherent causal descriptions.

The approach can be considered to be an outgrowth of PIP (Chapters 6 and 9), but using a complete causal model of disease. While CASNET (Chapter 7) simply propagates weights, ABEL symbolically manipulates—through operations such as aggregation and elaboration—causal concepts on multiple levels of detail. It is hierarchical, the kind of organization promoted in MDX (Chapter 13), but involves a principled abstraction of causes with complex links that can themselves be reasoned about; they are not just pointers connecting diseases.

The ABEL research is evolving into a study of reasoning strategies for using the principled representation of medical knowledge (Patil and Szolovits, 1982). This is clearly the state of the art in medical knowledge representation, with strong implications for producing robust consultation programs. The empirical psychological methodology—studying expert problem solving in detail to derive better representations—has been strongly promoted by the group at M.I.T. and Tufts and is an idea we see in much of the research reported in this volume (Chapters 10, 12, 13, 15, and 16).

14.1 Introduction

We have studied difficulties arising in the operations of the "first generation" of AI programs in medicine and have undertaken the development of knowledge representation structures to support needed improvements. The description of a patient in existing programs such as INTERNIST-I (Pople et al., 1975), PIP (see Chapter 6), and MYCIN (Shortliffe, 1976) starts from a single list of findings about the patient. Using a data base of associations between diseases and findings (or rules establishing those connections), these programs form an interpretation of the patient's condition that is essentially a list of possible diseases, ranked by a calculated estimate of likelihood or degree of belief in each.

Researchers (Patil, 1979; Pople, 1977; Smith, 1978) have recognized the need to use notions such as causal relationships, temporal patterns, and aggregate disease categories in the description of a program's diagnostic understanding, but the mechanisms provided to do this have been too weak. For example, although *causality* appears as a term in descriptions in PIP and INTERNIST-I, in both cases its use is limited to guiding the propagation of likelihood measures. These programs fail to capture the human notion that explanation should rest on a chain of cause-effect deduction. Although the CASNET/Glaucoma (see Chapter 7) program uses a network of causally related states and defines diseases as paths in this

network, its primary reasoning mechanism is nevertheless the local propagation of probability weights.

Similarly, it has been realized that a diagnostic or therapeutic program must consider a case at various levels of detail in order to integrate its overall understanding with its detailed knowledge. This insight also has not prevailed in the actual mechanisms provided in existing programs.

To explore the issues outlined here, we have undertaken a study of the medical problem of providing expert consultation in cases of electrolyte and acid-base disturbances. We have partly completed implementation of a program, ABEL, that is the diagnostic component of our overall effort. In this paper we concentrate on ABEL's mechanism for describing a patient. Called the *patient-specific model* (PSM) (Gorry et al., 1978), this description includes data about the patient as well as the program's hypothetical interpretations of these data in causal hierarchical networks. We describe the representations of medical knowledge and the processing strategies needed to enable ABEL to construct a PSM from the initial data presented to the program about a patient. The same representations and procedures will also be useful to revise the PSM during the process of diagnosis, but we will concentrate here on the logically prior operation of building the PSM.

Our understanding of medical expert reasoning suggests that an expert physician may have an understanding of a difficult case in terms of several levels of detail. At the shallowest that understanding may be in terms of commonly occurring associations of syndromes and diseases, whereas at the deepest it may include a biochemical and pathophysiological interpretation of abnormal findings. For our program to reason at a sophisticated level of competence, it will need to share such a range of representations. The PSM is, therefore, a multilevel causal model, each level of which attempts to give a coherent account of the patient's case. This model also serves as the basis for an English-generation facility that provides explanations of the program's understanding.

The PSM is created by instantiating portions of ABEL's general medical knowledge and filling in details from the specific case being considered. The instantiation of the PSM is very strongly guided by initially given data, because the PSM includes only those disorders and connections that are needed to explain the current case. Instantiation is accomplished by five major operators. *Aggregation* and *elaboration* make connections across the levels of detail in the PSM by filling in the structure above and below, respectively, a selected part of the network. In a domain such as ABEL's, multiple disorders in a single patient and the presence of homeostatic mechanisms require the program to reason about the joint effects of several mechanisms that collectively influence a single quantity or state. *Component decomposition* and *summation* relate disorders at the same level of detail by mutually constraining a total phenomenon and its components; the net change in any quantity must be consistent with the sum of individual

changes in its parts. The final operator, *projection,* forges the causal links within a single level of detail in the search for etiologic explanations. The operators all interact because the complete PSM must be self-consistent both within each level and across all its levels. Therefore, each operation typically requires the invocation of others to complete or verify the creation of related parts of the PSM.

14.2 Hierarchical Representation of Medical Knowledge

Based on our observation that a physician's knowledge is expressed at various levels of detail, we have developed a hierarchical multilevel representation scheme to describe medical knowledge and procedures to instantiate this knowledge to describe a particular patient's illness. The lowest level of description consists of pathophysiological knowledge about diseases, which is successively aggregated into higher-level concepts and relations, gradually shifting the content of the description from physiological to syndromic knowledge. The aggregate syndromic knowledge provides us with a concise global perspective and helps in the efficient exploration of diagnostic possibilities. The physiological knowledge provides us with the capability of handling complex clinical situations arising in patients with multiple disturbances, evaluating the physiological validity of the diagnostic possibilities being explored, organizing a large number of seemingly unrelated facts, and formulating therapy recommendations and prognosis. Finally, since the causal-physiological reasoning tends to be categorical and the syndromic reasoning probabilistic, the hierarchical description allows us to blend together the use of categorical and probabilistic reasoning (see Chapter 9).

14.2.1 Multilevel Description of States

Medical knowledge about different diseases and their pathophysiology is understood in varying degrees of detail. While it may be easier for a program to reason succinctly with medical knowledge artificially represented at a uniform level of detail, we must be able to reason with medical knowledge at different levels of detail to exploit all the medical information available. Although this does not pose any difficulty in medical domains where the pathophysiology of diseases is not well developed, in a domain such as electrolyte and acid-base disturbances where, on the one hand, the pathophysiology of the disturbances is well developed and, on the other, the pathophysiology of many of the diseases leading to these disturbances

is relatively poorly understood, we are constantly faced with this problem.

Second, the information about a patient parallels the physician's medical knowledge about diseases and therefore also comes at different levels of detail. For example, "serum creatinine concentration of 1.5" is at a distinctly different level than "high serum creatinine,"[1] and "lower gastrointestinal loss" is at a different level than "diarrhea." We need some mechanism by which we can interrelate these concepts. Finally, in order to be effective in diagnostic problem solving and communicating with clinicians, we ought to have the ability to portray the diagnostic problem in a small and compact space. Yet to be efficacious, we must maintain the ability to take every possible detail into consideration. We have solved this problem by representing the medical knowledge in five distinct levels of detail from a deep pathophysiological level to a more aggregate level of clinical knowledge about disease associations.

Each level of the description can be viewed as a semantic net describing a network of relations between diseases and findings. Each node represents a normal or abnormal physiological state and each link represents some relation (causal, associational, etc.) between different states. A state (interchangeably used with node) in the system, such as "diarrhea," is represented as a node in the causal network. Each node is associated with a set of attributes describing its temporal characteristics, severity or value, and other relevant attributes. A state is called a *primitive node* if it does not contain internal structure and is called a *composite node* if it can be defined in terms of a causal network of states at the next more detailed level of description. One of the nodes in this causal network is designated as the *focus node,* and the causal network is called the *elaboration structure* of the composite node. The focus node identifies the essential part of the causal structure of the node above it. Indeed, the collection of focus nodes acts to align the causal networks represented by different levels of the PSM. We note that very often a composite node and its focal description at the next level share the same name; this is typical in English, where the level of detail of place names, for example, is often obtained from context and not encoded in the name used. Nodes that do not play a role as the focal definition of any node at a higher level are called *nonaggregable nodes.* They represent a detailed aspect of the causal model that is subsumed under other nodes with different foci at less detailed levels of description.

To illustrate the description of a state at various levels of aggregation, let us consider the electrolyte and acid-base disturbances that occur with diarrhea, which is the excessive loss of lower gastrointestinal fluid (lower GI loss). The composition of the lower gastrointestinal fluid and plasma fluid are as follows:

---

[1]For a muscular patient whose previously known value of creatinine is 1.3 we can assume this to be normal, but for a patient with a previously known value of 1.0 this is definitely high and could imply a loss of about one-third of the kidney function.

	Lower GI fluid	*Plasma fluid*
Na	100–110	138–145 mEq/L
K	30–40	4–5 mEq/L
Cl	60–90	100–110 mEq/L
HCO_3	30–60	24–28 mEq/L

In comparison with plasma fluid, the lower GI fluid is rich in bicarbonate (HCO_3) and potassium (K) and is deficient in sodium (Na) and chloride (Cl). This information is represented in the knowledge base by decomposing lower GI loss into its constituents (and associating appropriate quantitative information with the decomposition). The loss of lower GI fluid would result in the loss of corresponding quantities of its constituents (in proportion to the total quantity of fluid loss) as shown in Figure 14-1.

Therefore, an excessive loss of lower GI fluid without proper replacement of fluid and electrolytes would result in a net reduction in the total quantity of fluid in extracellular compartments (hypovolemia). Because the concentration of K and HCO_3 in lower GI fluid is greater than it is in plasma fluid, there is a corresponding reduction in the concentration of K (hypokalemia) and HCO_3 (hypobicarbonatemia) in the extracellular fluid. Finally, as the concentration of Cl and Na in the lower GI fluid is lower than that in plasma fluid, there is an increase in the concentration of Cl (hyperchloremia) and Na (hypernatremia) in the extracellular fluid. This is represented at the next level of description as shown in Figure 14-2.

Figure 14-3 shows the aggregation of this information along with some additional causes and consequences of lower GI fluid loss at the next more aggregate level of detail.

The lower GI fluid loss at this level is a nonaggregable state and therefore does not have an aggregation at the next level above. Figure 14-4 shows the description of the aggregate effects of diarrhea (one of the causes of lower GI loss).

The summarization of the description of lower GI fluid loss and diarrhea shown in Figure 14-4 is achieved through the use of link aggregation and elaboration, described in the next subsection.

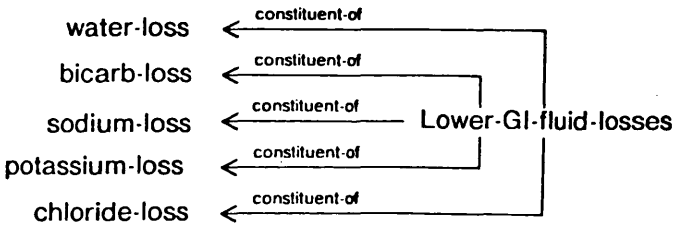


FIGURE 14-1 Effects of lower GI fluid losses on lower GI fluid constituents.

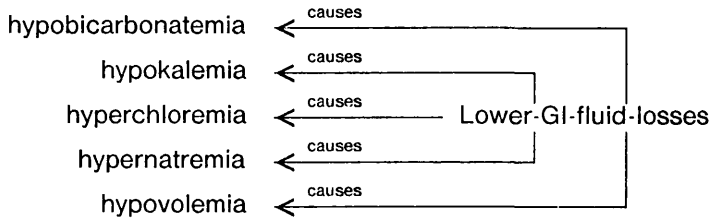


FIGURE 14-2 Effects of lower GI fluid losses at the next level of description.

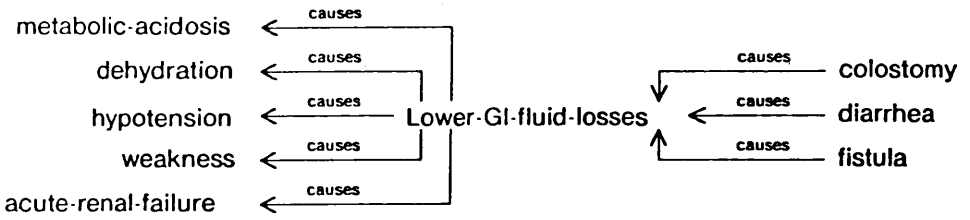


FIGURE 14-3 Aggregation of information in Figure 14-2 with some additional causes and consequences of lower GI fluid loss.

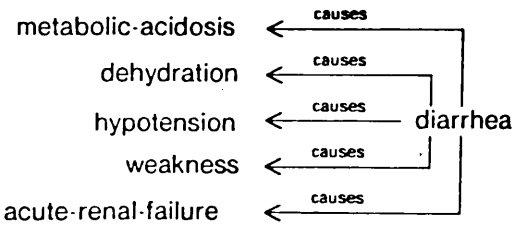


FIGURE 14-4 Summarization of the description of lower GI fluid loss and diarrhea.

14.2.2 Multilevel Description of Causal Links

A causal link specifies the cause-effect relation between the cause (the antecedent) and the effect (the consequent) states. In past programs (e.g., PIP, INTERNIST), causal links were described by specifying the type of causality (may-be-caused-by, complication-of, etc.) and a number or a set of numbers representing in some form the likelihood (conditional proba-

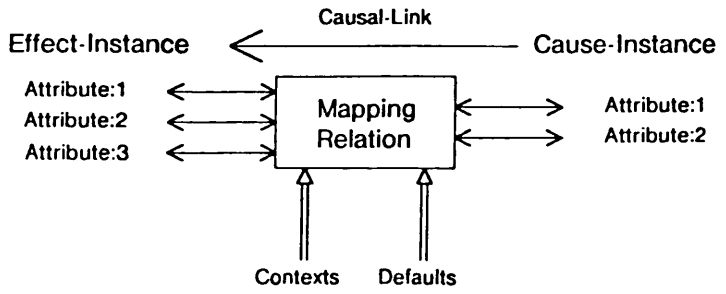


FIGURE 14-5 A causal link in the system.

bility), importance, etc., of observing the effect given the cause or *vice versa.* We now believe that this simple representation of the relation between states is inadequate. The form of presentation of an effect and the conditional probability of observing it depend on various aspects of the cause, such as severity, duration, etc., as well as other factors in the context in which the link is invoked[2] (such as the patient's age, sex, and weight, and the current hypothesis about the patient). Therefore, a causal link in the system (an object denoting the causal relation between a cause-effect pair) specifies a multivariate relation between various aspects of the cause and effect and also specifies the context and assumptions that constrain the causal relation, as shown in Figure 14-5.

One important function of diagnostic reasoning is to relate causally the diseases and symptoms observed in a patient. These causal relations play a central role in identifying clusters that can be meaningfully aggregated in developing coherent diagnoses. The presence or absence of a causal relation between a pair of states can change their diagnostic and prognostic interpretations. Therefore, the system should and does have the capability of hypothesizing the presence or absence of a causal link. This is the reason why links are objects in their own right rather than simple pointers between nodes.

To reason with a causal network representation effectively, a program must make conclusions about a node or link depending only on information that is locally available from the neighborhood of the mechanism in question. If nonlocal effects are to be invoked in causal explanations, they must be explicitly identified (e.g., as part of the context of the causal link), or else they corrupt our ability to reason with any portion of the network. If at some level of detail two distant phenomena interact, we must aggregate the description of the causal network to a level where the two phenomena are adjacent to one another. Further, because the causal relations

[2]For example, a severe diarrhea causes severe hypokalemia, and a mild diarrhea causes mild hypokalemia.

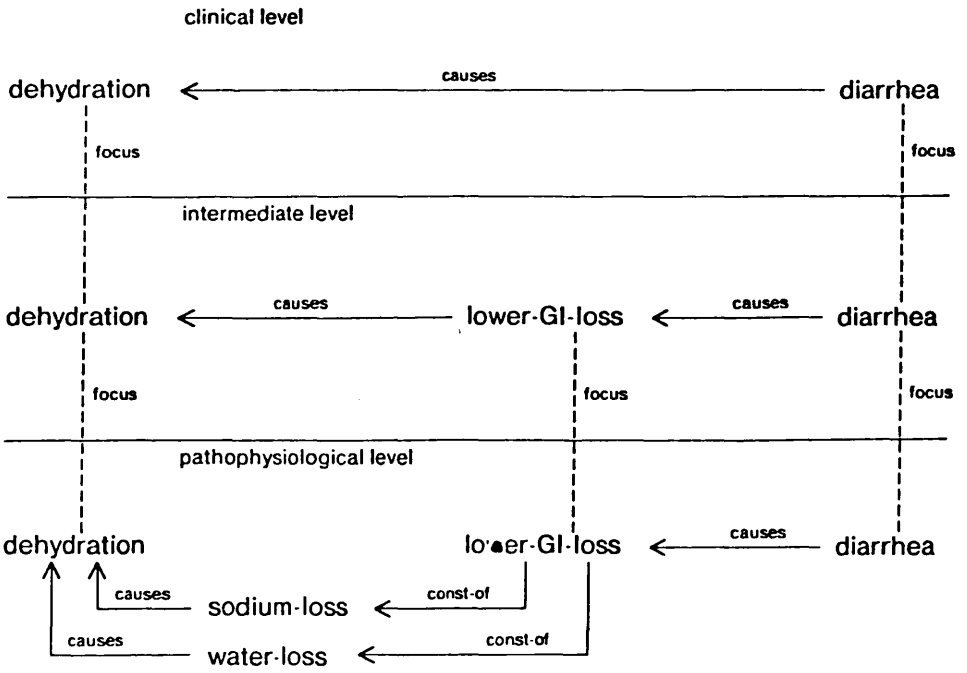


FIGURE 14-6 Causal relation between diarrhea and dehydration.

specified by links are not guaranteed to be true under all circumstances (they represent strong associations, not logical truth), the validity of deductions degrades with every additional intermediate link. That is, a causal pathway containing a large number of links is less likely to be valid than one using only a few links. Therefore, in order to explore a large diagnostic space, we must aggregate the diagnostic space to a level where each link represents an aggregate causal phenomenon covering larger distances and thus minimizing the possibility of error in the deduction. This ability to move from one level of description to another is provided by the multilevel description proposed here.

Links can be categorized, as nodes are, into two types: the *primitive links* and the *composite links*. To illustrate the concept of elaborating causal links to form a causal pathway, let us consider the causal relation between diarrhea and dehydration shown in Figure 14-6. The causal mechanism of diarrheal dehydration can be elaborated as follows: diarrhea causes lower GI fluid loss, which causes dehydration. Expressed at the next level of detail, the lower GI fluid loss can be described as consisting of the loss of water and sodium along with other electrolytes. The water loss in the

presence of the reduced total quantity of extracellular sodium results in lower extracellular volume, which at the higher level of description is described as dehydration.

14.3 Reasoning About Components

One of the important areas of medical diagnosis not adequately addressed by the first generation of AIM programs is the evaluation of the effect of more than one disease present in the patient simultaneously, especially when one of the diseases alters the presentation of the others. This problem does not place serious limitations on programs dealing with single problems such as the therapy of glaucoma or the diagnosis of bacteremia. But, in the case of electrolyte and acid-base disturbances, where a large fraction of cases involve multiple diagnoses, the ability to evaluate the joint influence of multiple diseases and the ability to decompose their influences on observable findings is particularly important.

For example, let us consider a patient with diarrhea and vomiting leading to severe hypokalemia. Let us also suppose that we know about the diarrhea, but we are not aware of the vomiting. The observed hypokalemia is too severe to be properly accounted for by the diarrhea alone. Without the ability to decompose the hypokalemia, we would have to attribute it completely to the diarrhea or completely to something else. In either case[3] we fail because the total state of hypokalemia is inconsistent with any of its possible single causes. Thus any single cause hypothesized by the program (e.g., vomiting) will not be severe enough to account for the observed hypokalemia by itself. As argued above, we need the ability to hypothesize that only a part of the hypokalemia is accounted for by diarrhea. We introduce the notion that any primitive node in the causal hierarchy[4] may have *components,* which are other primitive nodes that together make up the given node.

In our system this is achieved by a pair of operators: *component summation* and its dual, *component decomposition.* Using our example, these operators allow us to attribute only a part of hypokalemia to the diarrhea and to compute that part of hypokalemia that is not caused by diarrhea (called the *unaccounted component* of the hypokalemia). These operations deal not only with the magnitude of some disorder but also with other attributes such as duration. They are implemented by associating with each

[3]All of the previous programs would allow the entire hypokalemia to be accounted for by diarrhea. In particular, PIP, after allowing the hypokalemia to be accounted for by diarrhea, will not allow hypokalemia to lend any support to the hypothesis of vomiting. INTERNIST-1, on the other hand, will allow the entire hypokalemia to lend support to the hypothesis of vomiting as well as allowing it to be explained by diarrhea.

[4]Recall that *primitive* means that it is not the aggregation of a further defined causal structure.

primitive node a multivariate relation that constrains attributes of the node and its constituents. Component summation combines attributes of the components to generate the attributes of the joint node; component decomposition identifies unaccounted components by noting differences between the joint node and its existing components. These operations enrich the PSM by instantiating and unifying component nodes when the case demands them. This occurs whenever multiple causes contribute jointly to a single effect. An important case of this arises whenever feedback is modeled, because in any feedback loop there is at least one node acted on both by an outside factor and by the feedback loop itself.

As the PSM is built, component summation and decomposition operations can cause a node in the program's general knowledge to be instantiated as a node and its several components. If a node is primitive and there are multiple causes, the contribution of each cause is instantiated separately. Then the profile of the combination is computed using component summation. The combined effect is then instantiated and connected to its components by component links.

Because components are defined only for primitive nodes, the instantiation of composite nodes that involve component summation must be in terms of the summation of components in the node's elaboration structure. If the node is composite, we elaborate the constituent nodes around their focal nodes until we reach the primitive nodes associated with them at a level of greater detail. Then we combine these primitive nodes and aggregate their effects back. For example, if we know that a patient has two disturbances, diarrhea and shock, causing metabolic acidosis (Figure 14-7)[5], we evaluate their contribution to metabolic acidosis and then focally elaborate the two components until the metabolic acidosis is described in terms of the quantity of serum bicarbonate lost.[6] We then aggregate the joint effects to derive the actual severity of metabolic acidosis.

As mentioned above, the mechanism of component summation allows us to represent feedback explicitly by representing the primary component of the change (the forward path) and the secondary feedback component (the response of the homeostatic mechanism in defense of the parameter being changed) as components to be summed to yield the whole. Figure 14-8 shows the primary change in serum pH caused by low serum bicarbonate and the response of the respiratory system in defense against the change in serum pH. Read the example as follows: the lowering of the concentration of serum bicarbonate causes a reduction in serum pH, which causes hyperventilation and thus reduces the pCO_2, which in turn causes an increase in the serum pH (negative feedback). This increase is less than the initial reduction, causing a net reduction in serum pH.

---

[5]This is a hypothetical example; in the program this component summation will take place at the pathophysiological level.

[6]The quantities of serum bicarbonate lost may be summed by simply adding the loss due to each cause to evaluate their combined effect.

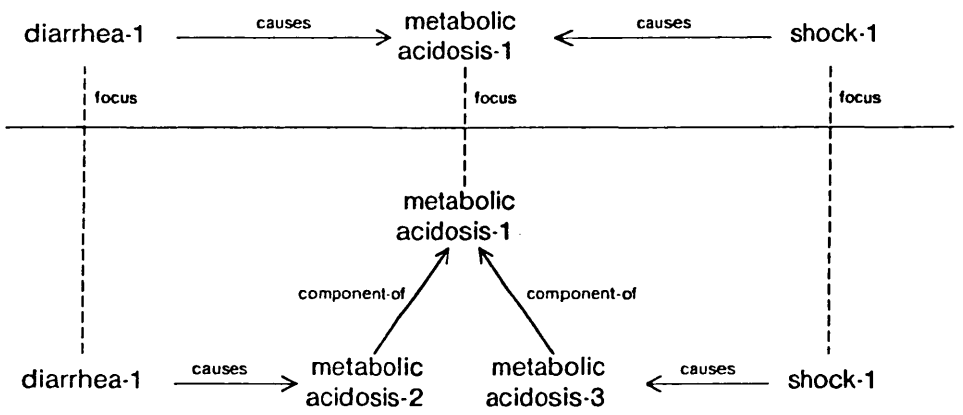


FIGURE 14-7 Metabolic acidosis caused by diarrhea and shock.

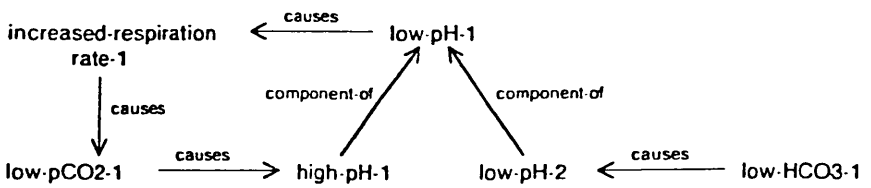


FIGURE 14-8 Primary change in serum pH caused by low serum bicarbonate causes response by respiratory system.

The decomposition of an effect with multiple causes into its causal components also provides us with valuable information in evaluating prognosis and in formulating therapeutic interventions.

14.4 The Patient-Specific Model

Diagnosis is the process of actively seeking information and identifying the disease process(es) causing the patient's illness. In other words, diagnosis involves ascertaining the facts and their implications. The effectiveness of the information-gathering process depends on the analysis of the available facts. From our experience with the existing diagnostic systems (Pople et al., 1975) (see also Chapter 6) we are convinced that a relatively simple

representation of physician's analysis of patient's illness (i.e., a list of disease hypotheses) is incapable of providing the desired level of expertise. The patient description must unify all known facts about the patient, their interpretations, their suspected interrelationships, and disease hypotheses in order to explain these findings. Finally, we observe that at any point in diagnostic reasoning practiced by human experts, there are only a few significantly different explanations for the patient's illness under consideration.

In the program, each such explanation is represented by a patient-specific model (PSM). Note that within each PSM all the diseases, findings, etc., are mutually complementary, while the alternate PSM's are mutually exclusive and competing. In this section we describe procedures for building and extending a patient-specific model based on the known findings and the program's medical knowledge. These operations are *initial formulation* to create an initial patient description from the presenting complaints and laboratory results, *aggregation* to summarize the description at a given level of detail to the next more aggregate level, *elaboration* to elaborate the description at a given level of aggregation to the next more detailed level, and *projection* to hypothesize associated findings and diseases suggested by states in the PSM.

14.4.1 Initial Formulation

From observing the clinical behavior of physicians, we have noticed that when presented with the chief complaints and other voluntarily provided information in a case, the physicians set up a tentative diagnosis. This diagnosis serves as a specific framework that can be used in soliciting information and for organizing the incoming information. Similarly, the program, when provided with the initial findings and a set of serum electrolyte values, constructs a small set of PSM's as its initial possible diagnoses, using the following steps. First, it analyses the electrolytes and formulates all possible single or multiple acid-base disturbances that are consistent with the electrolyte values provided and selects from them a small set that is consistent with the initial findings. Next, it generates a pathophysiological explanation of the electrolytes based on each of the proposed acid-base disturbances. This is performed by elaborating all known clinical information to the pathophysiological level, where its relationships to the laboratory data are determined by projecting the unique causes and definite consequences of every node. Then the program summarizes these pathophysiological descriptions to the clinical level by repeated application of aggregation operations. This process results in the initial description of the patient at every level of detail. It is this description that is later modified by the diagnostic process as new information becomes available. Note that each of the mechanisms, aggregation, elaboration, and projection, are used in the initial formulation of the PSM.

14.4.2 Aggregation

The aggregation process allows us to summarize the description of the patient's illness at any given level to the next more aggregate level. The summarization of a causal network can be achieved by recognizing that a central node and its surrounding causal relationships may be expressed at a more aggregate level by a single node (called *focal aggregation*) and by summarizing a chain of relations between nodes by a single causal relation between the initial cause and the final effect nodes (called *causal aggregation*).

Focal Aggregation

In aggregating a causal network, we must first identify the nodes in the network that form anchor points (i.e., landmarks, points of special significance) around which the causal phenomenon can be summarized. Consider a partially completed PSM in which some nodes at a detailed level of aggregation have been instantiated. Any of these nodes is an anchor point if (1) in the medical knowledge base such a node is the focus of some node at the next more aggregate level in the network and (2) at least one such higher-level node already exists or can be instantiated within the PSM. If it exists and the constraints on the focal link are satisfied, then the focal link connecting the two is instantiated. If it does not exist, then both it and the focal link are instantiated. Finally, if more than one possible description of the node is consistent with the causal structure above, we defer the aggregation process until we can obtain some additional information to resolve this ambiguity.

Causal Aggregation

Once we have determined the focal aggregations for nodes at a given level of aggregation, we need to describe the causal relations among these aggregate nodes. The process of causal aggregation takes a node and its causes and aggregates the relation between them according to one of three rules. First, if the node has no causal predecessors or if none of the causal paths leading into the node (called *predecessor paths*) have a node with a focal aggregation, then the focal aggregation of the node either is an ultimate etiology or is totally unaccounted for and does not need to be causally aggregated. Second, if every predecessor path has a node with a focal aggregation, then the focal aggregation of the node is fully accounted for. The causal aggregation is achieved by instantiating a causal link between the focal aggregation of the node and the first focal aggregation in each path. Finally, if only some of the predecessor paths have nodes with focal aggregations, then the focal aggregation of this node is partially ac-

counted for. The causal aggregation is achieved by decomposing the node into two components: (1) the component due to paths that have focal aggregation (called the *accounted component*), and (2) the component due to paths that do not have focal aggregation (called the *unaccounted component*). Then the focal aggregation of the node is decomposed based on the decomposition at the present level, and the two cases are treated as described above.

14.4.3 Elaboration

Elaboration is the dual of the aggregation operation described above and is used to elaborate the description of a causal network at a given level of aggregation to the next more detailed level. This is achieved by elaborating each link in the causal network by first describing the cause and effect of the link at the next more detailed level (called *focal elaboration*) and then instantiating the causal pathway between these detailed nodes (called *causal elaboration*). If the causal pathway being instantiated interacts with other causal paths in the PSM, the combined effects of the multiple causality are computed using component summation. The combined effects of this summation can then be aggregated to reflect the better understanding of the causal phenomenon at higher levels of aggregation.

Focal Elaboration

Focal elaboration is the inverse of focal aggregation. To focally elaborate a composite node, the program computes the possible profile of the focal concept associated with the given node. If a node at the next lower level of aggregation matches this profile and is consistent with the node above, the program instantiates the focal link connecting the two. If not, it instantiates the focal node and the focal link connecting the two.

Causal Elaboration

Causal elaboration is the dual of causal aggregation. A composite causal link can be elaborated if the cause and the effect nodes of the link have focal elaborations. To elaborate a composite link, the program matches the causal path associated with the link starting at the focal nodes of the cause and the effect of the link with existing paths in the PSM. If some part of this pathway is not present in the PSM, the program recursively calls itself on each link in the pathway (starting from the focus node of the source) that is absent in the PSM. If the link being recursively elaborated is a primitive link and if its effect node is not present in the PSM, the effect node and the link are instantiated. Otherwise, if the effect node is present,

it matches the attributes of the cause and the effect nodes. If they are compatible, it instantiates the link. Otherwise, if the effect node is an observed node,[7] the program decomposes the effect node and instantiates the link connecting the cause and the component of the effect node contributed by it. Otherwise, if the effect node is accounted for by some other cause, it instantiates the combined effect by summing the components of the two causes. Finally, it aggregates the effect node to revise the description at the next more aggregate level.

14.4.4 Projection

The projection operation is used to hypothesize and explain the associated findings and diseases suggested by the states in the PSM. The projection operation is very similar to elaboration. It differs from elaboration in that the causal relation being projected is hypothetical and therefore is not present in the PSM. Furthermore, the projection operation fails if the causal description of the hypothesized link is inconsistent with the description in the PSM at any level of aggregation. As a result, the application of the projection operation cannot result in the decomposition of a fully accounted node, creating an additional unaccounted component and therefore degrading the quality of explanation.

We envision using the projection operation in the diagnostic problem solver for exploring diagnostic possibilities, for evaluating their physiological validity, and in generating expectations about the consequences of hypothesized diagnoses.

14.5 An Example

Let us consider a 40-year-old 70-kg patient who has been suffering from moderately severe diarrhea for the last two days and, as a result, has developed moderately severe metabolic acidosis and hypokalemia. The laboratory analysis of the patient's blood sample (serum analysis) is Na, 140; K, 3.0; Cl, 115; HCO_3, 15; pCO_2, 30; and pH, 7.32.

14.5.1 Initial Formulation

To exercise the program, let us provide it initially with only the laboratory data. Based on these data, the program generates all possible acid-base disturbances that can account for the laboratory data, as follows:

[7]Or if the effect node is a causal predecessor of some observed node that completely accounts for it.

1. metabolic acidosis
2. chronic respiratory alkalosis + acute respiratory acidosis
3. metabolic acidosis + chronic respiratory alkalosis + acute respiratory acidosis
4. metabolic alkalosis + chronic respiratory alkalosis + acute respiratory acidosis

Based on the complexity,[8] likelihood, and severity of each component, the list of possible disturbances is pruned and rank ordered.[9] The rank-ordered list of likely disturbances is

1. metabolic acidosis (severity: 0.4)
2. chronic respiratory alkalosis (severity: 0.68) + acute respiratory acidosis (severity: 0.32)

The program now creates a PSM[10] for each possible acid-base disturbance and asserts in it instantiations of the laboratory data (at the pathophysiological level) and the appropriate acid-base disturbances (at the clinical level). In the rest of the example we will focus on the first acid-base disturbance, metabolic acidosis. The program focally elaborates the metabolic acidosis through the intermediate levels until it reaches the pathophysiological level and thus identifies the amount of HCO_3 loss corresponding to the severity of the metabolic acidosis. Based on this information and the laboratory data, it instantiates the feedback loop corresponding to the acid-base homeostatic mechanism. Next, it projects back[11] each node whose cause can be uniquely determined and projects forward the definite consequences of each node in the PSM. We now have the explanation at the pathophysiological level of the electrolytes consistent with the diagnosis of metabolic acidosis as shown in Figure 14-9.

14.5.2 Aggregation

After the pathophysiological description is completed, this description is aggregated through the intermediate levels to the clinical level of detail. To illustrate this operation, let us consider the low-serum-K-1 node at the

---

[8]Triple disturbances are quite rare and are generally not considered during initial formulation unless there is compelling evidence for their presence.

[9]The rank ordering of the diseases is based on Occam's Razor—simpler hypotheses are preferred.

[10]For ease of explanation, the example described here uses a three-level PSM instead of the five-level PSM used in the program.

[11]Note here that as we are at the pathophysiological level, each link being projected is primitive. Thus projecting back at this level can be restated as instantiating the cause and the link connecting the cause and the effect node.

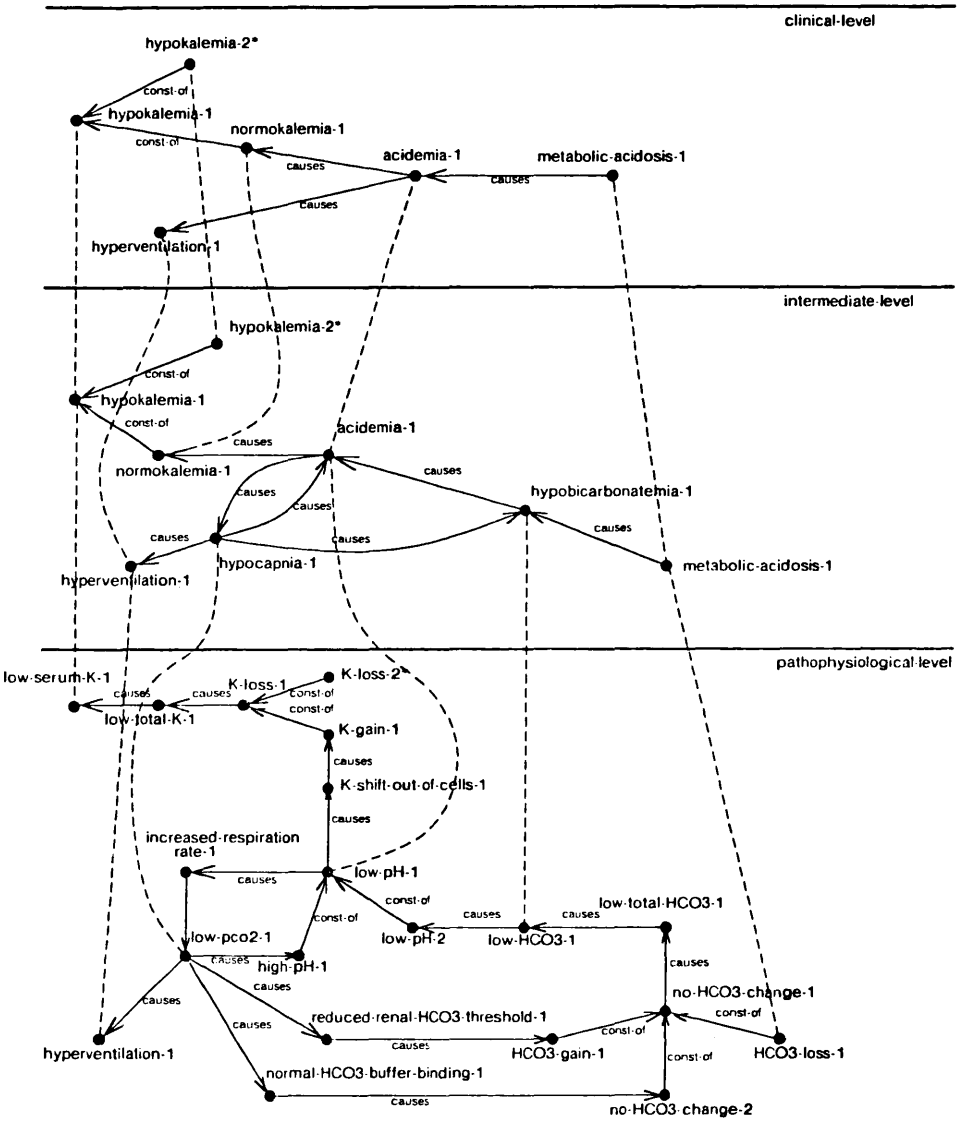


FIGURE 14-9 Initially formulated PSM.

pathophysiological level. Focally aggregating this node, we instantiate hypokalemia-1 as shown in Figure 14-9. To determine the causal aggregation of this node at the next level of detail, we must focally aggregate the first aggregable node in each path leading back, in this case low-pH-1. Focally aggregating low-pH-1, we instantiate acidemia-1. Next, we compute the component of low-serum-K that can be accounted for by low-pH-1 and the

component that remains to be accounted for because of the unaccounted K-loss-2. Then we compute the mapping of these components at the next level of aggregation and instantiate normokalemia-1 (the component accounted for by low-pH-1) and hypokalemia-2 (due to unaccounted K-loss-2). We then connect the normokalemia-1 to acidemia-1 and mark the hypokalemia-2 as unaccounted (indicated in the figure by an asterisk). Next, in order to causally aggregate low-pH-1, we focally aggregate low-pCO_2-1 and low-HCO_3-1 into hypocapnia-1 and hypobicarbonatemia-1, respectively. As each path leading back from low-pH-1 terminates in a node with focal aggregation, the focal aggregation of low-pH-1 (acidemia-1) is a fully accounted node. Therefore, we connect acidemia-1 to hypocapnia-1 and hypobicarbonatemia-1. This process is repeated for each aggregable node at the current level, and then the whole process is repeated at the next level until we reach the clinical level of aggregation.

14.5.3 Projection

To illustrate the projection operation, let us assume that the diagnostic component has hypothesized that the unaccounted component of hypokalemia at the clinical level (hypokalemia-2) is caused by diarrhea and wishes to determine if this is so and how this assumption fits with the current PSM. The result of this operation is shown in Figure 14-10.

To project the link between hypokalemia and diarrhea, the program evaluates the link to determine the attribute profile of the diarrhea consistent with hypokalemia-2, from which it determines the profile of diarrhea at the next more detailed level. It then attempts to match the causal path associated with the link (hypokalemia ← lower-GI-loss ← diarrhea) at the next level. As none of the links in this pathway are present and as this causal pathway is consistent with the description at the next level, the program recursively calls itself on each link in the path. Considering the first link (that is, hypokalemia ← lower-GI-loss), it finds the causal path associated with this link at the next level of detail (low-serum-K ← low-total-K ← K-loss ← lower-GI-loss). Matching this path with the description in the PSM, it finds that all but one link (K-loss ← lower-GI-loss) is already present. Since this link is primitive, the program evaluates the profile of the lower-GI-loss consistent with the unaccounted component of K-loss and instantiates it and the causal link connecting lower-GI-loss-1 to K-loss-2. To reflect this addition at the higher levels of detail, the program aggregates the low-serum-K-1 (the effect node in the path). As the low-serum-K-1 is now a fully accounted node, the component structure associated with its focal aggregation (hypokalemia-1) is deleted, and the causal links associated with the accounted component of hypokalemia-1 and an additional link from lower-GI-loss-1 are connected to it. This process is repeated until we establish the relation between the diarrhea and hypokalemia at the clinical level.

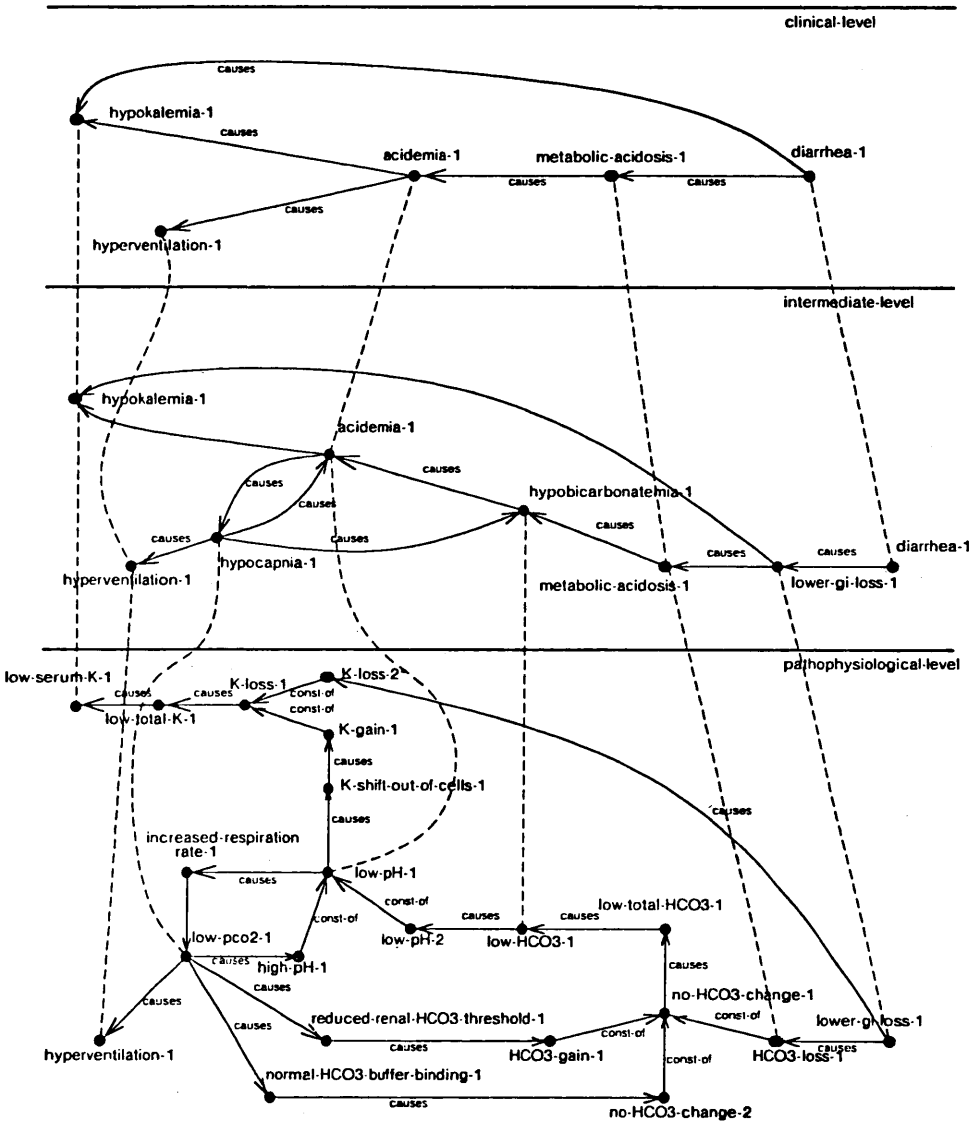


FIGURE 14-10 PSM extended to include diarrhea.

14.5.4 Elaboration

The process of elaboration is similar to that of projection described above and differs from it in two major ways: (1) the causal link and the associated nodes already exist in the PSM at the higher level of aggregation, and (2) we have already determined that the causal link being elaborated is valid.

Therefore, if a causal pathway associated with the link at some level of detail is not consistent with the description in the PSM, the program modifies the PSM appropriately to accommodate the pathway. In the example being described, the second (and more interesting) case does not arise. To demonstrate the elaboration process, let us establish the relation between diarrhea-1 and metabolic-acidosis-1 at the clinical level. The result of elaborating this link is shown in Figure 14-10.

14.6 English Explanation

To illustrate the program's understanding of the patient's illness at various levels of detail, an English generator was implemented to translate the PSM at any given level into its English description.[12] The descriptions are given at three levels of detail in Figure 14-11.

14.7 Conclusion

We have begun a complex and challenging task: to reason about difficult medical problems with a representation that is capable of capturing the subtlety and richness of knowledge and hypotheses used by expert physicians. We have thus far succeeded in creating a representation and a set of structure-building operators that are able to create a patient description based on causal models, multiple levels of detail in description, and the explicit use of components of quantities and states. The various viewpoints on the patient represented by different cuts through this complex description are kept consistent by the operators. We believe that this approach displays a level of understanding not achieved before in medical reasoning programs or other programs that need to describe an organization of hypotheses or mechanisms at different levels of detail.

In continuing to develop our diagnostic and therapeutic programs, we believe that the organizational framework provided by the PSM and its associated operators gives us a suitable machinery for exploring the choice of reasoning strategies and recording our programs' changing conceptions of a case. The rich network of interconnections in the PSM constrains a diagnostic reasoner to generate only a relatively small number of coherent explanations, thereby reducing the space of possibilities to be investigated

[12]The generator makes use of the methodology and some of the code of a generator built by William Swartout as part of an interactive system that explains and justifies portions of expert programs (Swartout, 1981).

Clinical Level

This is a 40-year-old 70.0-kg male patient with moderate diarrhea. His electrolytes are:

Na: 140.0	HCO_3: 15.0	Agap: 13.0
K: 3.0	pCO_2: 30.0	
Cl: 115.0	pH: 7.32	

The diarrhea causes moderate metabolic acidosis, which causes mild acidemia. The acidemia and diarrhea cause mild hypokalemia, and acidemia causes hyperventilation. All findings have been accounted for.

Intermediate Level

This is a 40-year-old 70.0-kg male patient with moderate diarrhea. His electrolytes are: ....

The diarrhea causes moderate lower GI loss, which causes moderate metabolic acidosis. The metabolic acidosis along with moderate hypocapnia causes moderate hypobicarbonatemia. The hypobicarbonatemia along with hypocapnia causes mild acidemia. The acidemia and lower GI loss cause mild hypokalemia, and acidemia causes hypocapnia. The acidemia also causes hyperventilation. All findings have been accounted for.

Pathophysiological Level

This is a 40-year-old 70.0-kg male patient with moderate lower GI loss. His electrolytes are: . . . .

Moderate lower GI loss, reduced renal HCO_3 threshold, and normal HCO_3 buffer binding jointly cause no HCO_3 change. The no HCO_3 change causes low ecf HCO_3, which causes low serum HCO_3. The low serum HCO_3 and low serum pCO_2 jointly cause low serum pH. The low serum pH causes K shift out of cells and causes increased respiration rate. The increased respiration rate causes low serum pCO_2, which causes normal HCO_3 buffer binding. The low serum pCO_2 also causes reduced renal HCO_3 threshold and increased respiration rate causes increased ventilation. The lower GI loss and K shift out of cells jointly cause K loss. The K loss causes low ecf K, which causes low serum K. All findings have been accounted for.

FIGURE 14-11 English explanation at different levels of detail.

in seeking a diagnosis. In particular, enforcing the requirements of causal consistency (at each appropriate level of detail) on any tenable explanation provides a means of pruning the diagnostic space and permits us to try a "hypothesize and debug" reasoning strategy. The multilevel interconnections of the PSM also help us merge decisions and considerations we have described as categorical and probabilistic. Although much work clearly remains before developments such as those described here form the fabric of truly successful medical consulting systems, we have proposed here a useful new representational basis for such work.

ACKNOWLEDGMENTS

This research was supported (in part) by a National Institutes of Health grant (No. 1 PO1 LM 03374-02) from the National Library of Medicine.

15

NEOMYCIN: Reconfiguring a Rule-Based Expert System for Application to Teaching

William J. Clancey and Reed Letsinger

As described in the introduction to GUIDON (Chapter 11), Clancey's work on that system led to an appreciation of the severe limitations of MYCIN's knowledge base if the system were to be used for instructional purposes (Clancey, 1983b). The NEOMYCIN research described in this chapter has been an attempt to rethink the knowledge structure and diagnostic strategy of MYCIN in view of requirements for teaching. This effort has several important products:

- *a better understanding of medical diagnostic strategy and its relation to knowledge structures (such as Feltovich's "logical competitor set," Chapter 12);*
- *a design of a representation framework for separating strategy from domain facts, in which strategy is stated* abstractly *(Clancey, 1983c); and*
- *a body of meta-rules, constituting a* generic procedure *that eases construction of knowledge bases for related problems in other domains (e.g., another diagnostic consultation program).*

The work is also of interest because of its relation to psychological studies (Chapter 12) and explanation methodology (Chapter 16).

From *Proceedings of the Seventh International Joint Conference on Artificial Intelligence*, vol. 2, 829–836 (1981). Used by permission of International Joint Conferences on Artificial Intelligence, Inc.; copies of the *Proceedings* are available from William Kaufmann, Inc., 95 First Street, Los Altos, CA 94022.

NEOMYCIN is a medical consultation system in which MYCIN's knowledge base is reorganized and extended for use in the next version of GUIDON. The new system attempts to capture psychological characteristics of diagnostic reasoning, designed to provide a basis for interpreting student behavior and teaching diagnostic strategy. This psychological orientation provides a constraint for making choices about representation and the reasoning process. In particular, NEOMYCIN captures the forward-directed, "compiled association" mode of reasoning that characterizes expert behavior. Collection and interpretation of data are focused by the "differential" or "working" memory of hypotheses. Moreover, the knowledge base is broadened so that GUIDON can teach a student when to consider a specific infectious disease and what competing hypotheses to consider, essentially the knowledge a human would need in order to use the MYCIN consultation system properly.

In order to articulate this knowledge to a student, it was necessary to greatly revise MYCIN's representation. Kinds of knowledge that were procedurally embedded in MYCIN's rules are stated separately, to make them accessible to the teaching program. The key idea is to represent explicitly and separately a domain-independent diagnostic strategy in the form of meta-rules, knowledge about a disease taxonomy, causal and data/hypothesis rules, and world facts. In essence, the new representation explicitly structures and controls the use of the diagnostic rules, simplifying them by isolating the basic data/hypothesis relations from their application criteria.

A more detailed discussion of methodological issues in the development of NEOMYCIN can be found in Clancey (1984). More recent research, exploiting the features of NEOMYCIN, includes modeling student strategies (London and Clancey, 1982) and stating strategies in explanations (Hasling et al., 1984). With the combination of empirical and knowledge-engineering interests, this research also has implications for incorporating cognitive modeling in new tools for building knowledge bases.

15.1 Introduction

A knowledge base used in a teaching program must explicitly represent what a student might need to be told. Development of intelligent tutoring systems such as SOPHIE (Brown et al., 1975), WHY (Stevens and Collins, 1978), WUMPUS (Goldstein, 1978), and GUIDON (Clancey, 1979a; 1979b) can be viewed, in part, as a problem of knowledge representation. This research has shown the advantages of:

- multiple representations of knowledge (e.g., the simulation model and semantic network in SOPHIE);

- representations that can be both interpreted and used to generate teaching text [e.g., Brown's meteorological automata (Brown et al., 1973) and production rules used in WUMPUS and GUIDON];
- network representations of knowledge that capture "importance" [SCHOLAR (Carbonell, 1970)], "complexity" or "prerequisite" associations [WUMPUS, BIP (Barr et al., 1976)], "analogy" and "generalization" relations (WUMPUS); and
- representations that allow for variants on expert performance (for modeling the student) [WEST (Burton, 1979), BUGGY (Brown and Burton, 1978)].

In the GUIDON program we have been exploring the problem of using MYCIN's rule set as teaching material. MYCIN (Shortliffe, 1976) is a rule-based expert system that provides therapy advice for certain kinds of infectious diseases. It has spawned a class of systems, called EMYCIN systems, that all use the same production rule language and interpreter (van Melle, 1980). GUIDON can operate using the rule set of any EMYCIN system as subject material.

MYCIN's rules were thought to be potentially useful for teaching because formal evaluations indicate that MYCIN captures a high level of expertise (Yu et al., 1979b), and modular design and representational meta-knowledge enable the program to explain its reasoning (Davis, 1976). Ironically, we have found that it is in precisely these two areas—expertise and explanatory capability—so important for a successful teaching program, that MYCIN falls short. To solve these problems, we have implemented a new system we call NEOMYCIN.

15.1.1 The Limitations of MYCIN for Application to Teaching

MYCIN is designed to be used as a consultant; consequently, we encounter difficulties when using it for teaching a student how to be a primary diagnostician. MYCIN's knowledge base is designed to interpret culture results from the blood and the cerebral-spinal fluid (CSF). But the expertise that suggests that such a culture should be taken is not part of the system. It is the user of MYCIN, the person seeking advice, who will think about meningitis in the first place and order the CSF culture and who will consider competing hypotheses (and medical tests) that need to be considered before MYCIN is even brought into the case as a consultant. This knowledge is certainly a critical part of teaching infectious disease diagnosis, but it lies completely outside the scope of the MYCIN knowledge base.

Moreover, protocols of experts who solve the same cases as are presented to MYCIN indicate that the program does not organize or use its knowledge in the same way a human expert does. This result is not sur-

prising, for MYCIN was not designed to simulate the *process* of human reasoning. The rules make use of the same data a physician uses and some of the same intermediate concepts of disease, but MYCIN's weakly focused, exhaustive search is quite dissimilar from how people reason. For GUIDON, our tutorial program, to articulate and recognize the hierarchical organizations of knowledge and search strategies that humans find useful, we need to reorganize MYCIN's rule set and incorporate an explicit model of human diagnostic reasoning, the kind indicated by psychological research in medical problem solving (Miller, 1975; Rubin, 1975; Pauker and Szolovits, 1977; Swanson et al., 1977; Elstein et al., 1978; Kassirer and Gorry, 1978) (see also Chapter 6). In particular, the model must exhibit:

- focused, forward-directed use of data (including *trigger* associations that suggest diagnoses);
- follow-up questions that establish the disease process (part of what a physician calls "forming a picture of the patient"); and
- management of a changing "working" memory of hypotheses under consideration.

In this sense, the development of NEOMYCIN is an attempt to synthesize previous medical psychological research and to analyze its application to the infectious disease problem domain.

15.1.2 Developing a Psychological Model by Modifying EMYCIN

A psychological model of diagnostic reasoning cannot be represented using the EMYCIN representation alone, that is, by simply rewriting MYCIN's rules. For example, the idea of asking a follow-up question is not allowed by MYCIN's rule interpreter. Also, we need to apply rules selectively and nonexhaustively. In general, the rule representation and interpreter must be modified; rules need to be organized so they can be selectively applied in different ways.

Many of the changes to EMYCIN are straightforward. They illustrate how local changes to the "inference engine" of a program can dramatically change how the knowledge base is used in problem solving. For example, a simple change is to provide for data-directed reasoning so new data can cause new subgoals to be set up and pursued. In MYCIN, an *antecedent rule* is tried whenever some piece of information required by the rule's premise becomes known. A NEOMYCIN *trigger rule* is similar, but it allows for new data to be requested in order to apply the rule. For example, one trigger rule is "if the patient has a stiff neck and a headache, then consider meningitis."[1] When a physician hears that the patient has a stiff neck, the

[1]The medical examples in this paper are simplified; we make no claims about completeness or accuracy. They are for purposes of illustration only.

IF: 1) The infection is meningitis,
2) The subtype of meningitis is bacterial,
3) Only circumstantial evidence is available,
4) The patient is at least 17 years old, and
5) The patient is an alcoholic
THEN: There is suggestive evidence that diplococcus-pneumoniae is an organism causing the meningitis

FIGURE 15-1 Typical MYCIN rule.

association to meningitis might come to mind, prompting him or her to determine whether the patient has a headache as well. This behavior is brought about in NEOMYCIN by simply marking trigger rules to distinguish them from ordinary antecedent rules and "throwing a switch" in the rule interpreter so that pursuing new subgoals is enabled for trigger rules.

Besides interpreter changes, different kinds of knowledge had to be separated out of the rules and represented explicitly. Figure 15-1 shows a typical (paraphrased) MYCIN rule, an example of "compiled expertise." We can list some of the individual steps of reasoning and knowledge sources out of which it is composed, unknown to MYCIN, but explicitly represented in NEOMYCIN:

- Analysis of other rules shows that this rule (to determine the organism) is only invoked after it has been established that the patient has an infection. Thus four major subgoals are established in this order: Is there an infection? Is it meningitis? Is it bacterial? Is it *Diplococcus pneumoniae?* Each of these subgoals hypothesizes a more specific cause of disease. Thus, *the ordering of clauses constitutes a top-down refinement strategy.* However, MYCIN does not know about this specialization hierarchy. It does not even know that *Diplococcus pneumoniae* is a bacterium. Perhaps most serious of all for meeting our teaching goals, MYCIN omits intermediate categories such as acute/chronic meningitis and gram-negative meningitis that physicians find helpful. In NEOMYCIN these categories are represented explicitly in an *etiological taxonomy* by allowing parameters to be specializations of one another.
- The clause about the patient's age prevents MYCIN from asking if a child is an alcoholic. MYCIN does not know that the ordering of these clauses is important, or what the relationship is. In NEOMYCIN these world relations are captured by separate *screening rules.*
- When there is laboratory evidence (a culture with visible organisms), this rule does not apply (clause 3). However, a companion rule still allows the circumstantial evidence of alcoholism to be considered, but gives it less weight. This principle of considering circumstantial evidence even when there are hard, physical observations of the cause is not explicitly known to MYCIN. The principle is compiled identically into 40 pairs of rules, rather than being stated as a reasoning rule for combining hard

and soft evidence. NEOMYCIN has rules for reasoning about the evidence it has collected, so connections between data and hypotheses are separate from the contexts in which they will be used.

These forms of knowledge—a (top-down) strategy, an etiological taxonomy, world facts, evidence-weighing rules—form a basis for a psychological model about knowledge organization and access, but they are not sufficient. Consider the above rule again. How does a physician remember to ask about alcoholism? How does he or she remember the connection with *Diplococcus?* Experts use a rich set of organizational aids and mnemonics for accessing their knowledge.

For example, one can think of taking the patient's history as a process of determining the differential of possible causes. Under this *strategy*, the expert follows the principle (rule model) that "compromised host conditions broaden the differential by suggesting special causes." Alcoholism is one of these conditions. So the low-level behavior of asking "Is the patient an alcoholic?" occurs in the context of the general process of diagnosis. In explaining the question to a student, it is important to be able to step back from the immediate concern for supporting a particular disorder and to articulate the general goals and methods of diagnosis itself. At the lowest level, the association to *Diplococcus* might be remembered as a simple causal story: alcoholics breathe in their own secretions, so organisms found in the mouth find their way to the lungs, causing pneumonia.

In summary, NEOMYCIN incorporates these psychological aids for teaching diagnosis:

1. a *representation of diagnostic strategy* that provides a meaningful, useful orientation for collecting data ("attempt to broaden the differential");
2. *structural associations* for indexing evidence to consider (abstractions such as "compromised host conditions" and rule models that use them); and
3. *rule justifications* that relate data/hypothesis associations to underlying causal processes.

15.1.3 The Need for Focusing Strategies

As we mentioned above, we cannot use MYCIN for teaching about meningitis diagnosis because it does not know how patients with meningitis typically appear when the physician first sees them and what competing disorders need to be considered. But if we simply added knowledge about more diseases and when to order laboratory tests we would be in trouble: a top-down diagnostic strategy is inadequate for a broader range of problems. The combinatorics of the search problem for medical diagnosis make it impossible for an expert to consider every infection, to work top-down. Initial information most commonly brings the physician into the *middle* of his or her taxonomic hierarchy (via the "compiled associations" such as the

trigger rule given above). Working from the middle, the physician must first look upward to focus the possibilities ("Is it a traumatic process? cancer?") and then refine downward. The approach used by MYCIN's rules only works because the user of the program is the one who focuses on meningitis. MYCIN can verify that the historical and laboratory evidence is consistent with meningitis, but it does not have the knowledge for considering meningitis in the first place. The program has only two infections to consider and does not know about other causes of the findings reported by the user.

For the program itself to shoulder this focusing burden (so that GUIDON can teach it to a student), we should more properly think of its area of expertise as being related to the observations a user will bring to it rather than the problems it knows how to confirm and refine. Thus MYCIN's area of expertise is "meningitis"; in contrast, NEOMYCIN deals with "abnormal neurological signs" or "headache and fever." In order to give NEOMYCIN the capability to deal with a broader range of problems, to actually have it think of other causes of headache and fever, we did the following:

1. *expanded the etiological knowledge* to include broad categories of other, noninfectious problems, such as "toxic problem," and "neoplastic problem";
2. *incorporated the focusing strategy of "group and differentiate"* so the program could manage this broader range of possibilities; and
3. *added knowledge about disease processes,* knowledge that cuts orthogonally across the etiological taxonomy, so diseases can be compared according to location, extent of the disorder, duration, severity, etc., in order to enhance the program's ability to apply the focusing strategy.

15.2 An Overview of NEOMYCIN

A few words about the character of MYCIN's problem domain are in order. We assume that a diagnosis or problem solution consists of an ordered list of problem causes that have been selected from a fixed, hierarchical space of hypotheses (e.g., "cancer process," "chronic meningitis") or categories of disease and pathophysiological states (e.g., "mass lesion in the brain"). We assume that an *informant* presents a problem to the program, which acts as a *consultant,* the role played by a student using GUIDON. There are two types of data: soft data (circumstantial or historical) and hard data (laboratory or direct measurements). Some of the evidence may be missing, and conclusions will usually be uncertain.

A schematic of the NEOMYCIN system (Figure 15-2) illustrates the various knowledge sources and their relation to the strategic knowledge and differential (the set of diagnoses under consideration). These com-

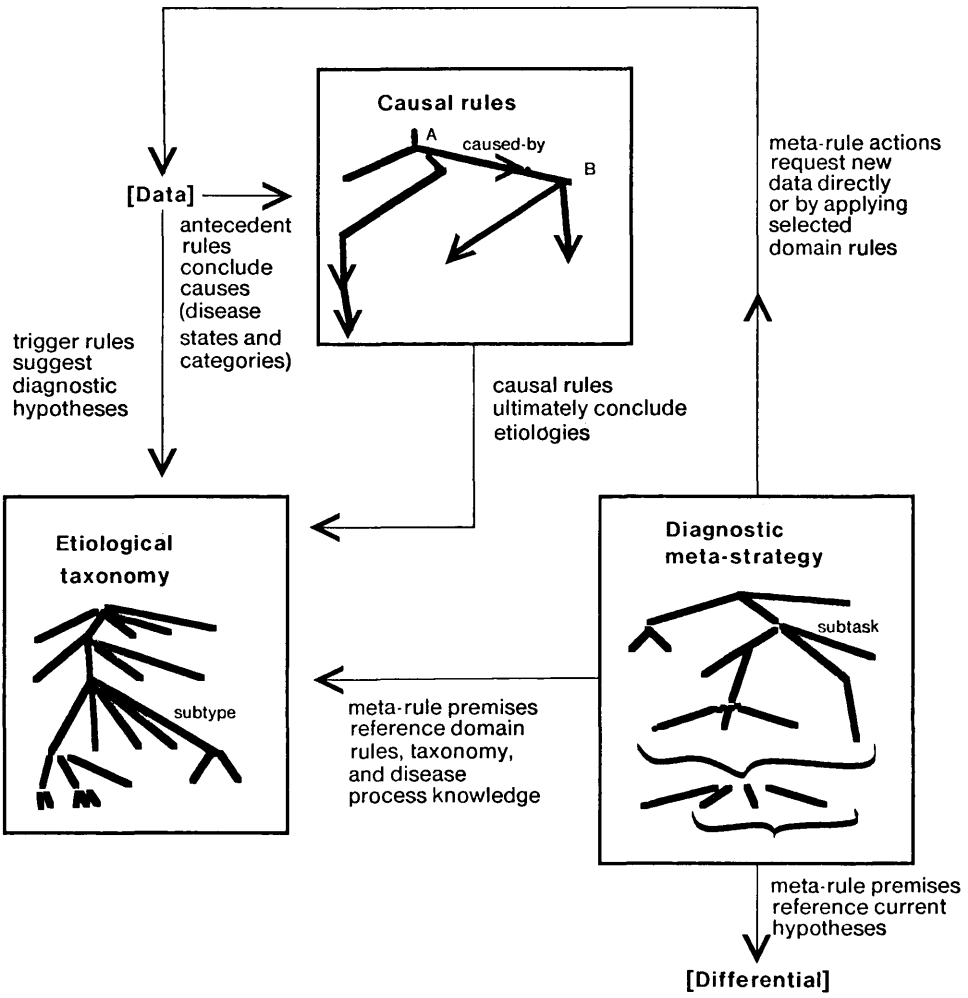


FIGURE 15-2 Components of the NEOMYCIN system.

ponents are shown as icons expanded in subsequent figures. The interpretation of Figure 15-2 follows.

- There are four kinds of domain rules:
  - *Causal rules* form a network of pathophysiological states and disease categories, ultimately linking raw observations (incoming data) to the etiological taxonomy.
  - *Trigger rules* associate data with etiologies, which are placed as hypotheses in the differential (maintained so that general causes are replaced by their more specific descendents).

METARULE397 (for the task group-and-differentiate)

IF: There are two items on the differential that differ in some disease process feature
THEN: Ask a question that differentiates between these two kinds of processes

FIGURE 15-3 A typical strategy rule.

- *Data/hypothesis rules* associate circumstantial and laboratory data with diseases, as do trigger rules, but only those rules focused by the differential are tried when the data are circumstantial (i.e., the associations that "come to mind" are those hypotheses already in the differential, as well as the nodes of the etiological taxonomy that hang below the hypotheses of the differential).
- *Screening rules* (not shown) form a hierarchy of abstractions and restrictions on data (e.g., "if the patient is not immunosuppressed, then he is not an alcoholic"); they are applied by backward chaining, in an attempt to determine a datum without asking the user.

- Other domain knowledge (not shown), orthogonal to the hierarchies of cause, considers diseases as processes having a location, extent, progression of symptoms, etc.
  - One form of disease process knowledge is represented as a framelike description associated with diseases in the etiological taxonomy and is used to differentiate among them.
  - A second form consists of a list of process-oriented, follow-up questions that should be immediately asked when some disease category or pathophysiological state is implicated (e.g., to establish when symptoms occurred and their ordering and change in severity).
- The meta-strategy for doing diagnosis consists of a hierarchy of domain-independent meta-rules. In general, these meta-rules examine the differential and make use of the etiological taxonomy, causal associations, and disease process knowledge to decide what datum to request next. A typical strategy rule is shown in Figure 15-3.

The annotated typescript in the next section shows how these forms of knowledge interact in practice. Subsequent sections provide a few more details about the representation.

15.3 A Sample Case

To illustrate the ideas presented above, a simple case was presented to a simple version of NEOMYCIN (there are generally only one or two rules to establish each hypothesis). Only the collection of circumstantial evidence

is shown in Figure 15-4. Although this trace of reasoning is very detailed, it is included here because it is precisely the *process* by which data, hypotheses, and rules interact that is at issue.

Observe the many different reasons why a question is asked; this is a reflection of the complexity of the diagnostic strategy. NEOMYCIN is hypothesis- and data-directed. In contrast, MYCIN only asks a question to evaluate a clause of a rule for the goal it is pursuing. Its rules are not sorted by conclusion, so its questions appear to skip back and forth randomly among hypotheses. It is not backward chaining *per se* that distinguishes the two systems, for NEOMYCIN essentially backward chains through its strategic rules. It is NEOMYCIN's forward, nonexhaustive reasoning and management of a space of hypotheses that make it reason more like a human does.

7-Dec-80 16:18:25

--------PATIENT-1--------

task MAKE-DIAGNOSIS METARULE384 succeeded.
task IDENTIFY-PROBLEM METARULE385 succeeded.

Initial tasks are unconditional. The first step is to collect identifying data, followed by the reasons for seeking advice.

Please enter information about the patient.

	Name	Age	Sex	Race
1)	** MIKE	4	MALE	CAUCASIAN

2) Please describe the chief complaint:

(Enter keyword or phrase.)

** DIPLOPIA

Diplopia is recognized to be a neurological sign which triggers meningitis, and leads to a follow-up question (associated with NEUROSIGN) . . . .

antecedent RULE376 succeeded.
Conclude: NEUROSIGN is YES (1.0)
antecedent RULE426 succeeded.
Conclude: INFECTIOUS-PROCESS is MENINGITIS (.7)

Differential: Adding MENINGITIS

3) How many days has Mike had abnormal neurologic signs?
** 13

Trying evidence rules: LENSIGN → CHRONIC-MENINGITIS
Trying RULE524; succeeded.
Conclude: MENINGITIS is ACUTE-MENINGITIS (-.3)
Conclude: MENINGITIS is CHRONIC-MENINGITIS (.3)

FIGURE 15-4 Trace of NEOMYCIN's reasoning for a sample case. [For brevity, the trace of rule conclusions is of the form "Conclude: PARAMETER is VALUE (CF)." Parameters are always associated with the context, here PATIENT-1. Strategic tasks are sometimes abbreviated. Figure 15-5 relates meta-rules to tasks.]

Differential:
replace MENINGITIS with CHRONIC-MENINGITIS

Another antecedent rule is associated with diplopia . . . .

antecedent RULE373 succeeded.
Conclude: BRAIN-PRESSURE is YES (.8)

Brain pressure suggests a mass lesion—these antecedent rules all model immediate, forward inferences that we observed in experts . . . .

antecedent RULE375 succeeded.
Conclude: MASS-LESION is YES (.72)

Differential: Adding MASS-LESION

Any other information?
** HEADACHE

Here a trigger antecedent rule requires more information before it can be applied . . . .

4) Does Mike have a stiff neck?
** YES
antecedent RULE424 succeeded.
Conclude: INFECTIOUS-PROCESS is MENINGITIS (.91)

Any other information?
** NO

The program now takes initiative, starting its main history-taking task of "establish the hypothesis space" (hereafter, EHS). The differential has new, unexamined elements, so the "group and differentiate" (G&D) task is invoked first. METARULE400 recognizes that the etiological ancestors of chronic meningitis have not been explored, so an attempt is made to confirm (task PURSUE-HYPOTHESIS = PH) that an infectious process is causing the problem.

Enter EHS loop: focus = NIL
task METARULE427 succeeded.

Review differential: MASS-LESION CHRONIC-MENINGITIS

Enter G&D loop: focus = NIL
task METARULE400 succeeded.

Enter PH loop: focus = INFECTIOUS-PROCESS
task METARULE410 succeeded.

5) Is Mike febrile?
** Y
antecedent RULE423 succeeded.
Conclude:
DISORDER-ETIOLOGY is INFECTIOUS-PROCESS (.7)

There are no further questions to ask; an interrupt returns control to the G&D task. No rules succeed, so control returns to the EHS task. The "explore and refine" (E&R) task silently refines mass lesion, expanding the differential by different etiological categories, and so triggering return to the EHS task.

observed STOP-PURSUING METARULE408

Repeating G&D loop: focus = INFECTIOUS-PROCESS

Repeating EHS loop: focus = INFECTIOUS-PROCESS
task METARULE428 succeeded.

Enter E&R loop: focus = INFECTIOUS-PROCESS
task METARULE429 succeeded.

FIGURE 15-4 continued

observed STOP-EXPLORING METARULE407

Repeating EHS loop: focus = INFECTIOUS-PROCESS
task METARULE427 succeeded.

The first step is again to review the differential, a process observed in experts. Process features of brain abscess and chronic meningitis are compared: they both occur in the central nervous system, are chronic problems, and are infectious, but brain abscess is a localized problem. NEOMYCIN asks a question to discriminate on this basis . . . .

Review differential: BRAIN-ABSCESS HEMATOMA
PUS-IN-BRAIN CHRONIC-MENINGITIS

Enter G&D loop: focus = INFECTIOUS-PROCESS
task METARULE397 succeeded.

6) Does Mike have focal neurological signs?
** NO
Trying evidence rules: FOCALSIGNS → BRAIN-ABSCESS
RULE179 failed due to clause 1

The program has not been supplied with knowledge for confirming other causes of mass lesion (e.g., traumatic hemorrhage, tumor), so it is unable to continue its grouping operation and begins an exploration cycle . . . .

Repeating G&D loop: focus = INFECTIOUS-PROCESS

Repeating EHS loop: focus = INFECTIOUS-PROCESS
task METARULE428 succeeded.

Enter E&R loop: focus = INFECTIOUS-PROCESS
task METARULE402 succeeded.

Enter PH loop: focus = BRAIN-ABSCESS
task METARULE409 succeeded.

Now directly focusing on brain abscess, the program "realizes" that data supplied earlier are relevant (RULE433). Chronic meningitis is then considered by refining it and pursuing specific causes. Pursuing TB, NEOMYCIN follows the strategy of confirming the first ("enabling") step in the disease process: contact with the organism . . . .

Trying evidence rules: MASS-LESION → BRAIN-ABSCESS
Trying RULE433; succeeded.
Conclude: INFECTIOUS-PROCESS is BRAIN-ABSCESS (.216)

Observed STOP-PURSUING METARULE408

Repeating E&R loop: focus = BRAIN-ABSCESS
task METARULE429 succeeded.

Enter PH loop: focus = TB-MENINGITIS
task METARULE411 succeeded.

7) Does Mike have a TB risk factor?
** YES
Trying evidence rules: TBRISK → TB-MENINGITIS
Trying RULE525; succeeded.
observed STOP-PURSUING METARULE408

Focusing strategies dictate that a sibling be considered next. Fungal meningitis is refined, and a child, cryptococcus, pursued . . . .

Repeating E&R loop: focus = TB-MENINGITIS
task METARULE401 succeeded.
Enter PH loop: focus = FUNGAL-MENINGITIS

FIGURE 15-4 continued

Repeating E&R loop: focus = FUNGAL-MENINGITIS
task METARULE399 succeeded.
Enter PH loop: focus = CRYPTOCOCCUS

A cancer patient is at some risk of getting cryptococcal meningitis. Rather than asking directly if the patient has cancer, the program models an expert's efficient casting of a wider net by asking a more general question. Specifically, there are "screening rules," that lead it to determine first if the patient is immunosuppressed (RULE395) and then compromised (RULE343). This is the only form of backward chaining that occurs in NEOMYCIN.[2]

task METARULE431 succeeded.

--[0] Findout: LEUKEMIA
--[1] Findout: IMMUNOSUPPRESSED

Trying RULE343;
8) Is Mike a compromised host (e.g. alcoholic, sickle-cell-disease, immunosuppressed)?
** YES
RULE343 failed due to clause 1

If the patient were not compromised, the program could have concluded that he is not immunosuppressed (RULE343). Now it is unsure and must ask directly. If the patient is not immunosuppressed, the program will know that he does not have leukemia (RULE395). The answer of LEUKEMIA below implies immunosuppressed, so RULE395 fails, and the original goal is determined.

--[1] Finished: IMMUNOSUPPRESSED
9) Is Mike immunosuppressed (e.g. corticosteroid therapy, cytotoxic drug therapy, radiation therapy, leukemia)?
** LEUKEMIA

I will assume that leukemia is one of the diagnoses of Mike
RULE395 failed due to clause 1

--[0] Finished: LEUKEMIA

Trying evidence rules: LEUKEMIA → CRYPTOCOCCUS
Trying RULE056; succeeded.
Conclude: FUNGAL-MENINGITIS is CRYPTOCOCCUS (.3)

Repeating E&R loop: focus = CRYPTOCOCCUS
task METARULE401 succeeded.

Attention turns to a sibling. Again, the "enabling step" is asked about first . . . .

Enter PH loop: focus = COCCIDIOIDES
task METARULE411 succeeded.

10) Has the patient ever been to a cocci-endemic area?
** NO

Trying evidence rules: COCCI-ENDEMIC → COCCIDIOIDES
RULE570 failed due to clause 1
RULE287 failed due to clause 1
observed STOP-PURSUING METARULE408

Repeating E&R loop: focus = COCCIDIOIDES

Repeating EHS loop: focus = COCCIDIOIDES
task METARULE430 succeeded.

[2]*Ed. note:* A later version of NEOMYCIN accomplishes this form of inference by meta-rules.

Having exhausted its limited knowledge, the program finds no other relevant, hypothesis-oriented questions to ask. Several general questions are asked . . . .

11) Is Mike receiving any medications?
** NO

Repeating EHS loop: focus = COCCIDIOIDES
task RULE430 succeeded.

12) Has Mike been recently hospitalized?
** NO

Repeating EHS loop: focus = COCCIDIOIDES

If additional data had been supplied, new hypotheses might have been placed on the differential and strategies for grouping or refining might have been called into play once again. This ends the history-taking process. Next the program would order laboratory tests, process them, and perhaps return to gathering circumstantial evidence.

FIGURE 15-4 continued

15.4 The Diagnostic Meta-Strategy

Formalizing the diagnostic strategy from protocol analysis was the most difficult part of designing NEOMYCIN. Figure 15-5 shows the general outline of the meta-strategy. Each nonterminal node in the tree stands for a task that is achieved by a set of rules. An important aspect of our model of diagnosis is that the process can be taught as a task-posing activity: the problem solver thinks in terms of what he or she is trying to do (e.g., to consider unusual causes and so broaden the differential) in order to bring knowledge sources to mind. Thus the meta-strategy is structured so the tasks make sense as things that experts try to do.

Figure 15-5 shows that the main object of the meta-strategy is to decide what data to collect next (invoke MYCIN's FINDOUT routine), generally by focusing on some hypothesis in the differential. Aside from collecting initial information, the basic idea is that collecting circumstantial evidence is a process of *establishing the hypothesis space.* This process takes the form of considering what could cause the reported data, grouping and refining the differential, and asking general questions.[3] A great deal of what we might call *heuristic confidence* is placed in the general questions, which constitute the outline of the "history-taking process" as it is generally taught to medical students. However, strategies for using causal and disease process knowledge enable the expert to be an efficient problem solver in a combinatorially large space, and these strategies are generally not taught.

[3]*Group and differentiate* is used here in the loose sense of establishing general focus on a process that is consistent with hypotheses suggested independently by the data. *Clustering* (in multiple ways) *and discriminating*, the usual meaning of the term, is one operation for achieving this focus.

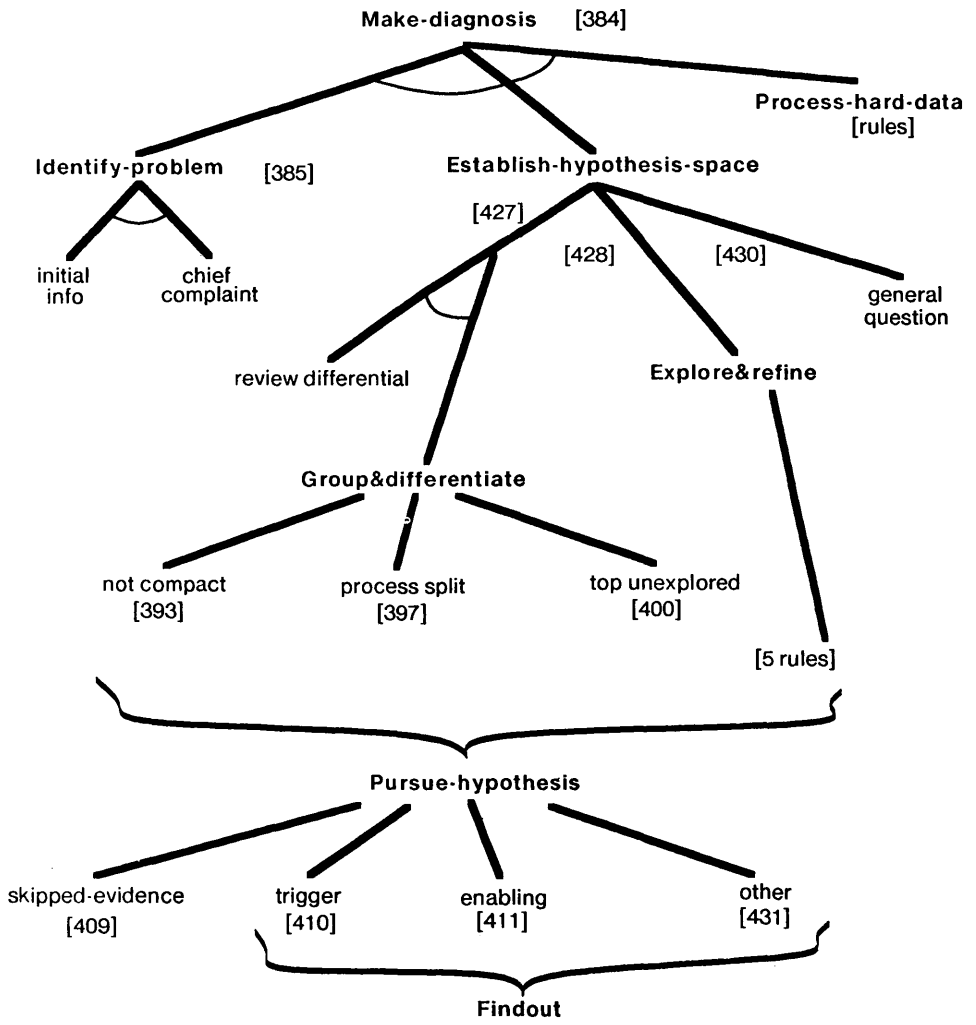


FIGURE 15-5 NEOMYCIN's diagnostic meta-strategy. (Rule numbers in brackets appear in the sample trace.)

The implementation is in terms of hierarchical meta-rules,[4] which as a whole constitute the meta-strategy. Figure 15-6 illustrates how the rules for a given task are treated as a pure production system—they are repeat-

[4]So called because they indirectly control the invocation of the domain-dependent object rules. Davis's conception of meta-rules was that they would directly order object-level rules. However, in our theory of diagnostic strategy, meta-rules reason about the state of the differential and knowledge sources (kinds of evidence) that could change it in desirable ways. Thus, our meta-rules choose *kinds of object rules* (hypothesis-confirming, process-oriented, causal).

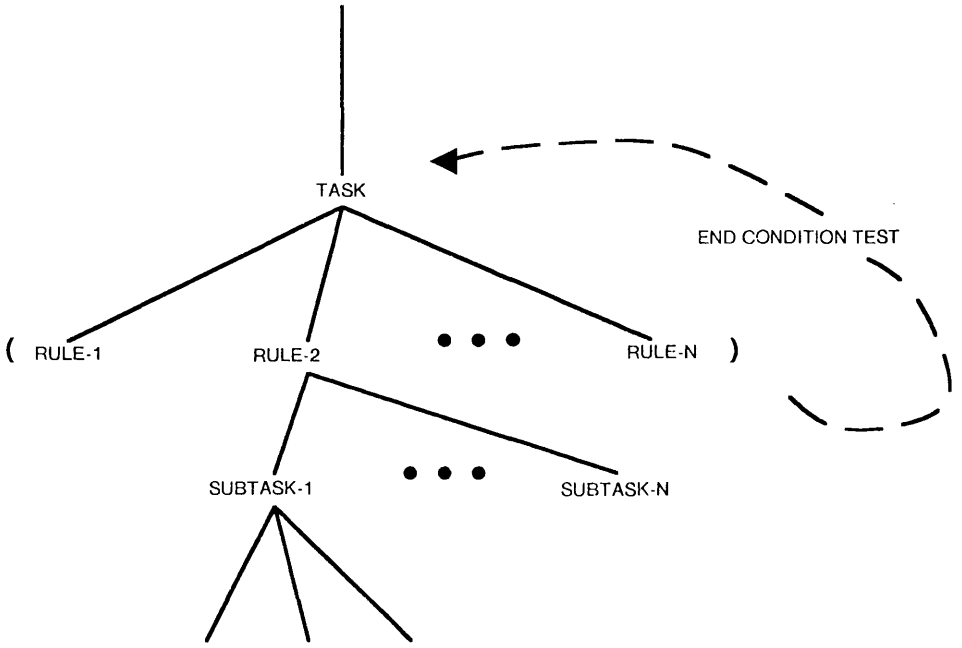


FIGURE 15-6 Rule-based invocation and interruption of strategic tasks.

edly tried in order, returning to the head of the list when one succeeds, stopping when no rule succeeds or an end condition is true.

The end condition is itself determined by rules, and is inherited as we descend into the hierarchy of tasks. The main use for this feature is to allow refocusing when new data change the state of the differential, as well as nonexhaustive consideration of hypotheses.

15.5 Etiological Taxonomy, Causal and Disease Process Knowledge

Some details of the implementation are given in this section. The etiological taxonomy (Figure 15-7) is implemented as EMYCIN parameters in which the values for one parameter (e.g., CHRONIC-MENINGITIS) are themselves parameters (e.g., TB-MENINGITIS and FUNGAL-MENINGITIS). We call these *taxonomic parameters.*

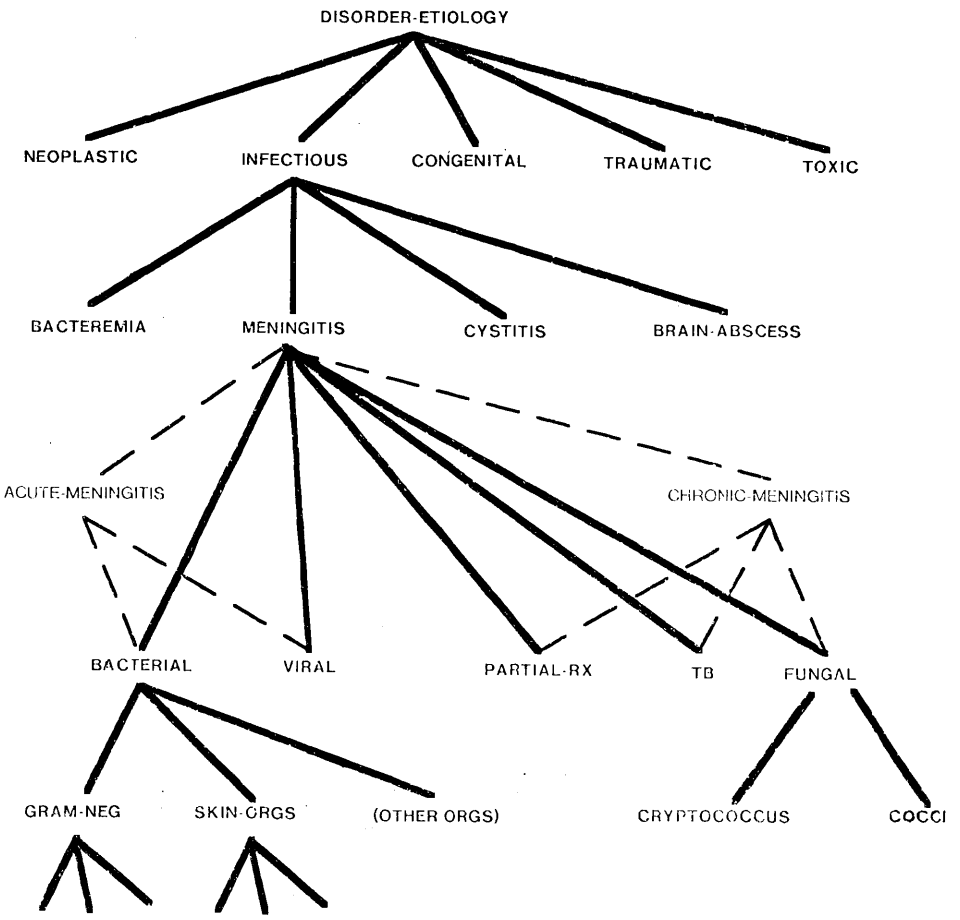


FIGURE 15-7 Portion of etiological taxonomy. (Links represent specialization of cause.)

Causal knowledge (Figure 15-8) is represented as rules modified by a certainty factor, as are all MYCIN rules. A causal rule of the form "if A then B" implies that A is caused by B, the direction of the association that is most generally useful for interpreting data and refining hypotheses. These rules mention *data parameters, taxonomic parameters,* or *state-category parameters.* State-category parameters stand for pathophysiological states or categories of disease (e.g., a mass lesion in the brain). In linking these concepts together, it is important to properly distinguish between causal and subtype links. (While we might say that an unknown mass lesion, a

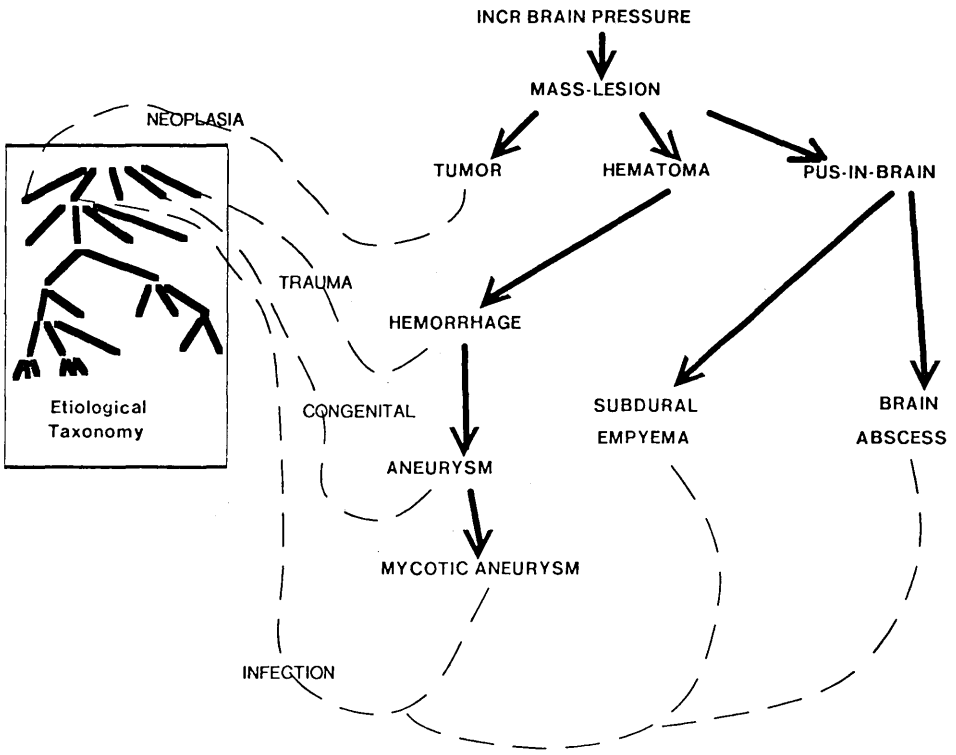


FIGURE 15-8 Portion of causal rule network, showing connection to etiological taxonomy.

space-occupying substance, is caused by a tumor, it is more proper to represent a tumor as a kind of mass lesion.) Causal rules are used by the "explore and refine" task to work backward from state-category hypotheses in the differential to prior causes, and ultimately to diagnostic hypotheses in the etiological taxonomy (as shown in Figure 15-8).

Disease process knowledge is represented as a frame associated with taxonomic parameters. Slots are process descriptors such as EXTENT, LOCATION, and COURSE associated with a literal value and a pointer to the parameter to establish it. For example, associated with BRAIN-ABSCESS is the triple (EXTENT FOCAL FOCALSIGNS), meaning that the extent of the disease is focal and this can be determined by asking about focal signs. Disease process knowledge is orthogonal to the etiological taxonomy, making it useful for grouping and discriminating hypotheses (see sample trace, before question 6).

15.6 Related Research

Besides the intelligent computer-assisted instruction (ICAI) projects cited in the introduction, our work has been motivated by previous research in teaching problem-solving strategies [e.g., Papert (1970); Brown et al. (1977); Wescourt and Hemphill (1978)]. We believe NEOMYCIN is the first attempt to formalize a runnable psychological model of diagnostic strategy that can be presented to a student. As should be obvious from our analysis, a considerable debt is owed to the medical problem-solving literature, cited above.

Both Reggia (1978) and Aikins (1980) modified the MYCIN system to make it more acceptable to physicians, particularly to improve knowledge acquisition. Aikins's use of an etiological taxonomy and trigger rules, derived from Rubin's work, is particularly close to our approach. However, we go a step further by representing strategic knowledge separately in domain-independent form. Our teaching application has also made clear the importance of disease process knowledge for broadening the diagnostic range of a consultation program.

Research in cognitive psychology has been helpful to us, particularly studies at the Learning Research Development Center (Anderson et al., 1981; Chi et al., 1981) (see also Chapter 12) in modeling the differences between experts and novices in geometry and physics problem solving. To some extent, our attempt to "decompile" MYCIN's knowledge is the inverse of Anderson's task of modeling how a novice composes and generalizes knowledge from experience.

15.7 Some Limitations

Pople's experience has been useful to point out limitations in our design. He shows that a simplistic causal network is not adequate when an attempt is made to represent all of general internal medicine (Pople, 1982). For example, when the causal connections between data and the taxonomy are long and complex, it may not be feasible to follow each path (possible cause). His "bridge concepts" [similar to Feltovich's "logical competitor sets" (see Chapter 12)] are attempts to model how an expert jumps over to distal, tentative hypotheses. They essentially provide a quick way to find the intersection of causes for a set of disease symptoms.

Similarly, Rubin's thesis illustrates a number of strategies for combining hypotheses (for example, relating complications and causes) that we have not yet found to be important in MYCIN's domain. To this extent,

our model is not the complete story of human diagnostic reasoning, but it can be built on as we expand our experience into other domains. We do not yet understand how an expert organizes his or her differential; how context is saved and restored from interrupts; how urgency, cost, and human values factor into the diagnostic process; and so on.

15.8 Summary of What We Learned

To teach diagnosis, it is useful to have a psychological model of problem solving. In particular, we need to incorporate into our model the medical knowledge and strategies an expert uses for initial problem formulation. An expert thinks in terms of a hierarchy of causes and the process characteristics of a disease so that he or she can order the data and the search. Moreover, an expert has learned "compiled associations" that allow him or her to efficiently associate hypotheses with data (e.g., trigger rules, Pople's "bridge concepts"), and cast a wide net of questions (e.g., general, screening, and follow-up questions).

Also, we need to represent the various kinds of knowledge explicitly so that they can be accessible for teaching. Our method is to represent strategic knowledge in domain-independent form, wholly separate from the medical knowledge described above. This requires that the medical knowledge be organized so that it can be indexed by the strategies (e.g., as the disease-process frame links abstract features of any disease, such as progression over time, to means for establishing this information in a particular case).

In a sense, we join cognitive psychologists [e.g., Anderson et al. (1981) and Rumelhart and Norman (1980)] in rediscovering the procedural/declarative problem in the context of how knowledge becomes transformed through experience. We recognize that the expert has composed associations, so he or she makes wide, tentative jumps between data and hypotheses. However, we represent these compiled associations declaratively for use in instruction (spelling out the diagnostic procedure in detail), and we record justifications of data-interpretation rules to allow for explanation of reasoning.

15.9 Future Research

Development of NEOMYCIN and GUIDON version 2 will proceed in parallel. Comparisons of NEOMYCIN's performance to MYCIN's will indicate if our more principled representation has changed the performance of the

system. This is a possibility because we have simplified some rules so they represent more closely the associations a human expert normally remembers. Preliminary runs give comparable results, though NEOMYCIN asks fewer questions because of its focused approach. We might also use NEOMYCIN's representation and meta-rules for diagnosis in a nonmedical domain, to test the domain-independence of the model.

GUIDON version 2 will use the NEOMYCIN representation, making it possible to articulate diagnostic strategy. A new phase of development will begin as we try to use the diagnostic strategies (and variants of them) for interpreting student behavior, leading to capabilities to evaluate partial solutions and provide assistance. The first version of GUIDON attempted these things, but was not able to recognize or suggest psychologically valid approaches.

ACKNOWLEDGMENTS

We are especially grateful to Timothy Beckett, M.D., for his patient explanations and willingness to be directed in our discussions. Bruce Buchanan and Bob London have also contributed to the GUIDON project. This research has been supported in part by an ARPA and ONR contract (NO0014-79C-0302). Computational resources were provided by the SUMEX-AIM facility (NIH grant RR00785).

16

Explaining and Justifying Expert Consulting Programs

William R. Swartout

As was mentioned in the introduction to Chapter 14, the ABEL work of Patil, Szolovits, and Schwartz uses a patient-specific model inspired in part by an earlier project from the M.I.T./Tufts group known as the Digitalis Therapy Advisor (Gorry et al., 1978). The Digitalis Therapy Advisor reached an excellent level of performance regarding the appropriate adjustment of digitalis dosing in cardiac patients, and it also provided a rich environment for related work such as the XPLAIN research of William Swartout described in this chapter. Swartout focused on the construction of an explanation capability for the Digitalis Therapy Advisor; the resulting programs have in turn influenced subsequent AI research on explanation.

Traditional methods for generating explanations by a decision-making program have involved displaying "canned" text or converting to English the code of the program (or traces of the execution of that code). While such methods can provide superficially useful explanations of what the program does or did, they generally cannot tell why *what the system is doing is a reasonable thing to be doing. The problem is that the knowledge required to provide these justifications is used (by the programmer) only when the program is being written and does not appear in the code itself.*

Swartout's XPLAIN system, on the other hand, uses an automatic programmer to generate the consulting program by refinement from abstract goals. The automatic programmer uses a domain model, consisting of facts about the application domain, and a set of domain principles that drive the

From the *Proceedings of the Seventh International Joint Conference on Artificial Intelligence,* vol. 2, pp. 815–823, (1981). Used by permission of International Joint Conferences on Artificial Intelligence, Inc.; copies of the *Proceedings* are available from William Kaufmann, Inc., 95 First Street, Los Altos, CA 94022.

refinement process forward. Examining the refinement structure created by the automatic programmer makes possible justifications of the code. This chapter describes XPLAIN and outlines additional advantages this approach has for explanation.

The significance of Swartout's work is not just its use of a system design technique that makes explanation possible. His work reveals how principles *(here, domain strategies by which specific treatment methods are applied) are part of explanation. It is useful to supply not just an "audit trail" of what a problem solver did (on perhaps different levels of detail) but an explanation of* why *the procedure is valid. Swartout's point is that a more powerful knowledge representation is the secret to better explanation, not just better natural language facilities. The same observation holds for tutoring systems (see Chapters 11 and 15).*

16.1 Introduction

To be acceptable, expert programs must be able to explain what they do and justify their actions in terms understandable to the user. Expert programs usually have some heuristic basis. While these heuristics may provide good performance for most cases, there may be unusual cases where they produce erroneous results or where the rationale for using them is faulty. If a user is suspicious of the advice he or she receives, the user should be able to ask for a description of the methods employed and the reasons for employing them. In addition, the scope of expert systems, like that of human experts, is often quite narrow. An explanation facility can help a user discover when a system is being pushed beyond the bounds of its expertise.

In the area of medical consultant programs,[1] the need for explanation is particularly acute. In designing a consultant program, we must consider what sorts of capabilities we are trying to provide for the physician user. If we consider the interaction between a physician and a human consultant, we realize that it is not just a simple one-way exchange where the physician provides data and the consultant provides an answer in the form of a prescription or diagnosis. Rather, there is typically a lively dialogue between the two. The physician may question whether some factor was considered or what effect a particular finding had on the final outcome. Viewed in this light, we realize that a computer program that only collects data and provides a final answer will not be found acceptable by most

[1]Some medical consultant programs include MYCIN, a program that aids physicians with antimicrobial therapy (Shortliffe, 1976); INTERNIST, a program that makes diagnoses in internal medicine (Pople, 1977); and PIP, a program that makes diagnoses primarily in the area of renal disease (see Chapter 6).

physicians. In addition to providing diagnoses or prescriptions, a consultant program must be able to explain what it is doing and justify why it is doing it.

Researchers have recognized this, and many proposals for new expert systems have at least mentioned the need for explanation. Some systems have actually provided an explanatory facility. Yet existing approaches to explanation fail in some important ways. This paper will document these failings and describe an approach toward their solution.

While we have concentrated on the problem of providing explanations to medical personnel, we do not feel that the need for explanation is limited to medicine or that the techniques we have developed for explanation and justification are limited to medical applications. Medical programs provide a good test-bed for the general problem of explaining a consulting program to the audience it is intended to serve.

The next section will describe the Digitalis Therapy Advisor, a program we have chosen as a test-bed for our ideas about explanation, and some of the medical aspects of digitalis therapy. After that, we will describe some of the problems with previous explanation systems and the approach we have taken to overcome those problems.

16.2 Digitalis Therapy and the Digitalis Therapy Advisor

The digitalis glycosides are a group of drugs that were originally derived from the foxglove, a common flowering plant. Their principal effect is to strengthen and stabilize the heartbeat. In current practice, digitalis is prescribed chiefly to patients who show signs of congestive heart failure (CHF) and/or conduction disturbances of the heart. Congestive heart failure refers to the inability of the heart to provide the body with an adequate blood flow. This condition causes fluid to accumulate in the lungs and outer extremities, and it is this aspect that gives rise to the term *congestive*. Digitalis is useful in treating this condition because it increases the contractility of the heart, making it a more effective pump. A conduction disturbance appears as an arrhythmia, which is an unsteady or abnormally paced heartbeat. Digitalis tends to slow the conduction of electrical impulses through the conduction system of the heart, and thus steady certain types of arrhythmias. Due to the positive effect that digitalis has on the heart, it is one of the most commonly used drugs in the United States.

Like many other drugs, digitalis can also be a poison if too much is administered. For a variety of reasons, including a small therapeutic window, subtle signs of toxicity, and high interpatient variability, digitalis is difficult to administer. One complication the physician must deal with is

the possibility that a patient may be more sensitive to the drug (for whatever reason) than the average patient. If a physician knows those factors that make a patient more sensitive, he or she can reduce the likelihood of overdosing (or underdosing) the patient by adjusting the dose depending on whether or not the sensitizing factors are observed.

Over the years, a number of factors that increase the automaticity of the heart[2] have been identified. These include a low level of serum potassium (hypokalemia), a high level of serum calcium (hypercalcemia), damage to the heart muscle (cardiomyopathy), and a recent myocardial infarction, among others. When these exist in conjunction with digitalis administration, the automaticity can be increased substantially. This chapter will describe in detail how those fragments of the Digitalis Therapy Advisor that adjust for the first two sensitivities are justified and explained.

16.2.1 The Digitalis Therapy Advisor Test-Bed

A few years ago, the Digitalis Therapy Advisor was developed at M.I.T. by Pauker, Silverman, and Gorry (Silverman, 1975; Gorry et al., 1978). This program was later revised and given a preliminary explanatory capability (Swartout, 1977). The limitations of these explanations (and of those produced by similar techniques) will be discussed below. This program differed from earlier digitalis advisors (Peck et al., 1973; Jelliffe et al., 1970; Jelliffe et al., 1972; Sheiner et al., 1972) in two important respects. First, when formulating dosage schedules, it anticipated possible toxicity by taking into account the factors that increased digitalis sensitivity and reduced the dose when those factors were present. Second, the program made assessments of the toxic and therapeutic effects that actually occurred in the patient after receiving digitalis to formulate subsequent dosage recommendations. This program worked in an interactive fashion. The program asked the physician for data about the patient and produced recommendations after that data was entered. When the dose of digitalis was being adjusted, the physician was asked to consult with the program again to assess the patient's response. This is the program we used as a test-bed for our work in explanation and justification. In the remainder of the paper, we will refer to this program as the old Digitalis Advisor.

---

[2]In the normal heart, there is a place in the left atrium called the sino-atrial (SA) node, which sets the pace for the heart. Under the right circumstances, other parts of the heart can take over the pace-setting function. Sometimes this can be life-saving, if, for example, the SA node is damaged. But at other times it can be life-threatening, since several pacemakers operating simultaneously tend to increase the likelihood of setting up a dangerous arrhythmia. When we say that digitalis increases the automaticity of the heart, we mean that digitalis increases the tendency of other parts of the heart to take over the pace-setting function from the SA node.

16.3 Kinds of Questions That Arise Concerning the Advisor

In the spring of 1979, we conducted a series of informal trials in an attempt to discover what kinds of questions occurred to medical personnel as they ran the old Digitalis Advisor. In this trial, medical students and fellows were asked to run the program and ask questions (verbally) as they occurred to them. The author attempted to answer these questions. The interactions were tape-recorded and later transcribed.

No formal analysis of the data was attempted, but examination of the transcripts did provide an indication of the types of questions that might arise while running a consulting program. These included:

1. Questions about the methods the program employed:

   User: "How do you calculate your body store goal? That's a little lower than I anticipated."

This sort of question could be answered by the explanation routines of the old Digitalis Advisor. It can also be answered by the system presented in this paper.

2. Justifications of the program's actions:

   User (peruses recommendations): "Why do we want to make a temporary reduction?"

   Author: "We're anticipating surgery coming up and surgery, even noncardiac surgery, can cause increased sensitivity to digitalis, so it wants to temporarily reduce the level of digitalis."

This is exactly the sort of question we are concentrating on in this paper. It cannot be answered by the explanation routines of the old Digitalis Advisor.

3. Questions involving confusion about the meaning of terms:

   User (in response to the question IS THE RENAL FUNCTION STABLE?): "Now this question . . . I'm not really sure . . . 'renal function stable' does it mean stable abnormally or . . . because I mean, the patient's renal function is not normal but it's stable at the present time."

   Author: "That's what it means."

This paper will not address this last type of question.

16.4 Previous Approaches to Explanation

A number of different approaches have been taken to attempt to provide programs with an explanatory capability. The major approaches include (1) using previously prepared text to provide explanations and (2) producing explanations directly from the computer code and traces of its execution.

The simplest way to get a computer to answer questions about what it is doing is to anticipate the questions and store the answers as English text. Only the text that has been stored can be displayed. This is called *canned text,* and explanations produced by displaying canned text are called *canned explanations.* The simplest sorts of canned explanations are error messages. For example, a medical program designed to treat adults might print the following message if someone tried to use it to treat an infant:

THE PATIENT IS TOO YOUNG TO BE TREATED BY THIS PROGRAM.

It is relatively easy to get a small program to provide English explanations of its activity using this canned text approach. After the program is written, canned text is associated with each part of the program explaining what that part of the program is doing. When the user wants to know what is going on, the computer merely displays the text associated with what it is doing at the moment.

There are several problems with the canned text approach to explanation. The fact that the program code and the text strings that explain that code can be changed independently makes it difficult to guarantee consistency between what the program does and what it claims to do. Another problem with the canned text approach is that all questions and answers must be anticipated in advance and the programmer must provide answers for all the questions that the user might ask. For large systems, that is a nearly impossible task. Finally, the system has no conceptual model of what it is saying. That is, to the computer, one text string looks much like any other, regardless of the content of that string. Thus it is difficult to use this approach if we want our system to provide more advanced sorts of explanations, such as suggesting analogies or giving explanations at different levels of abstraction.

Another approach to explanation is to produce explanations directly from the program (Davis, 1976; Shortliffe, 1976; Swartout, 1977; Winograd, 1971). That is, the explanation routines examine the program that is executed. Then by performing relatively simple transformations on the code, these explanation routines can produce explanations of how the system does things. For example, the old Digitalis Advisor could examine the code it used to check for increased digitalis sensitivity caused by increased serum calcium and produce an explanation of how that code worked (as shown in Figure 16-1).

TO CHECK SENSITIVITY DUE TO CALCIUM I DO THE FOLLOWING STEPS:

1. I DO ONE OF THE FOLLOWING:
   1.1 IF EITHER THE LEVEL OF SERUM CALCIUM IS GREATER THAN 10
   OR INTRAVENOUS CALCIUM IS GIVEN THEN I DO THE FOLLOWING SUBSTEPS:
      1.1.1 I SET THE FACTOR OF REDUCTION DUE TO HYPERCALCEMIA TO 0.75.
      1.1.2 I ADD HYPERCALCEMIA TO THE REASONS OF REDUCTION.
   1.2 OTHERWISE, I REMOVE HYPERCALCEMIA FROM THE REASONS OF
   REDUCTION AND SET THE FACTOR OF REDUCTION DUE TO HYPERCALCEMIA TO 1.00.

FIGURE 16-1 Explanation of how the old Digitalis Advisor checked hypercalcemia in general.

The old Digitalis Advisor, like most similar systems, also maintained an execution trace. The trace could be examined by the explanation routines to tell what the system did for a particular patient. Figure 16-2 describes how the system checked for myxedema. The system also had a limited ability to explain why it was asking the user a question. Figure 16-3 shows the system's response when the user wants to know why he or she is being asked about serum calcium.

I CHECKED SENSITIVITY DUE TO THYROID-FUNCTION BY EXECUTING THE FOLLOWING STEPS:

1. I ASKED THE USER THE STATUS OF MYXEDEMA. THE USER RESPONDED THAT THE STATUS OF MYXEDEMA WAS PRESENT.

2. SINCE THE STATUS OF MYXEDEMA WAS PRESENT I DID THE FOLLOWING:

   2.1 I ADDED MYXEDEMA TO THE PRESENT AND CORRECTABLE CONDITIONS. THE PRESENT AND CORRECTABLE CONDITIONS THEN BECAME MYXEDEMA.

   2.2 I REMOVED MYXEDEMA FROM THE DEGRADABLE CONDITIONS. THE DEGRADABLE CONDITIONS THEN BECAME HYPOKALEMIA, HYPOXEMIA, CARDIOMYOPATHIES-MI, AND POTENTIAL POTASSIUM LOSS DUE TO DIURETICS.

   2.3 I SET THE FACTOR OF REDUCTION DUE TO MYXEDEMA TO 0.67. THE FACTOR OF REDUCTION DUE TO MYXEDEMA WAS PREVIOUSLY UNDETERMINED.

   2.4 I ADDED MYXEDEMA TO THE REASONS OF REDUCTION. THE REASONS OF REDUCTION THEN BECAME MYXEDEMA.

FIGURE 16-2 Explaining how thyroid function was checked.

WHAT IS THE LEVEL OF SERUM CALCIUM?

ENTER NUMERICAL VALUE = = = => why? *(The user types "why?" indicating that the user questions why he or she is being asked to supply the value of serum calcium.)*

MY TOP GOAL IS TO BEGIN THERAPY. ONE STEP IN DOING THAT IS TO CHECK SENSITIVITIES. I AM NOW TRYING TO CHECK SENSITIVITY DUE TO CALCIUM.

FIGURE 16-3 A *limited* explanation telling why a question is asked.

Since the explanation routines only perform simple transformations on the program code, the quality of the explanations produced in this manner depends to a great degree on how the system code is written. In particular, the basic structure of the program is not altered significantly, and the names of variables in the explanation are basically the same as those in the program. If the explanations are to be understandable, the expert system must be written so that its structure is easily understood by anyone familiar with its domain of expertise, and the variable and procedure names used in the program must represent concepts that are meaningful to the user.

This method of producing explanations has some advantages. It is relatively simple. If the right way of structuring the problem can be found, it does not impose too great a burden on the programmer; since the explanations reflect the code directly, consistency between explanation and code is assured.

Despite these advantages, there are some serious problems with this technique. It may be difficult or impossible to structure the program so that the user can easily understand it. The fact that every operation performed by the computer must be explicitly spelled out sometimes forces the programmer to program operations that a physician would perform without thinking. That problem is illustrated in Figure 16-2. Steps 2.1, 2.2, and 2.4 are somewhat mystifying. In fact, these steps are needed by the system so that it can record what sensitivities the patient had that made him or her more likely to develop digitalis toxicity. These steps are involved more with record keeping than with medical reasoning, but they must appear in the code so that the computer will remember why it made a reduction. Since they appear in the code, they are described by the explanation routines, although they are more likely to confuse than enlighten a physician user. An additional problem is that it is difficult to get an overview of what is really going on here. While the system is explicit about record keeping, it is not very explicit about the fact that it is going to reduce the dose, though it hints at a reduction by saying that the factor of reduction is being set to 0.67.

An additional problem, and the primary one we will address in this paper, is that while this way of giving explanations can state *what* the system does or did, it has only a limited ability to state *why* the system did what it did (see Figure 16-3). That is, the system cannot give adequate justifications for its actions. In the explanations given above, the system cannot state that it reduces the dose because increased calcium causes increased automaticity. The information needed to justify the program is the information that was used by the programmer to write the program, but it does not have to be incorporated into the program for the program to perform successfully—just as one can successfully bake a cake without knowing why baking powder appears in the recipe. Since it is desirable for expert programs to be able to justify what they do as well as do it successfully, we need to find a way of capturing the knowledge and decisions that went

into writing the program in the first place. The remainder of this chapter will describe recent efforts we have made toward achieving that goal in the context of the Digitalis Therapy Advisor.[3]

16.5 Providing Justifications

We need a way of capturing the knowledge and decisions that went into writing the program. One way to do this is to give the computer enough knowledge so that it can write the program itself and remember what it did. Automatic programming has been researched considerably (Balzer et al., 1977; Barstow, 1977; Green et al., 1979; Long, 1977; Manna and Waldinger, 1977), but using an automatic programmer to help in producing explanations is a new idea. Since we are primarily interested in explanation, we have chosen not to deal with a number of problems that arise in automatic programming, including choosing between different implementations, backup and recovery from dead-end refinements, and optimization.

16.5.1 System Overview

XPLAIN is our framework for creating expert systems. Systems developed within it can be explained and justified. An overview is given in Figure 16-4. The system has five parts: a writer, a domain model, a set of domain principles, an English generator, and a generated refinement structure. The writer is an automatic programmer. It wrote new code that captured the functionality of major portions of the old Digitalis Advisor.[4] The domain model and the domain principles contain knowledge about the domain of expertise. In this case, they contain information about digitalis and digitalis therapy. They provide the writer with the knowledge it needs to write the code for the Digitalis Therapy Advisor. The refinement structure can be thought of as a trace left behind by the writer. It shows how the writer develops the Digitalis Therapy Advisor. When a physician-user runs the Digitalis Therapy Advisor, he or she can ask the system to justify why the program is doing what it is doing. The generator gives the user an answer by examining the refinement structure and the step of the advisor currently being executed. If we wanted to write a new program

[3]Clancey (1979c) notes that even in rule-based systems, knowledge is often too "compiled," resulting in explanation problems very similar to the ones described here.

[4]The code that has been written includes code to anticipate toxicities and to check for and assess various types of toxicities that may occur. As is discussed by Swartout (1981), it should not be too difficult to complete the remainder of the implementation so that the functionality of the old Digitalis Advisor is completely captured.

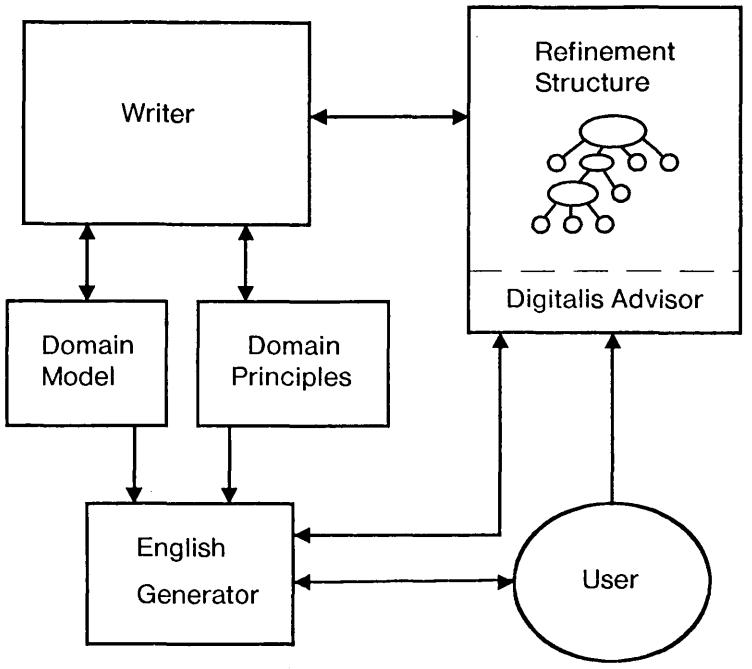


FIGURE 16-4 System overview.

covering a new medical domain, we would have to change the domain model and the domain principles, but we would not have to change the writer or the English generator.[5]

The refinement structure is created by the writer from the top-level goal (in this case, "administer digitalis") as it writes the Digitalis Therapy Advisor. The refinement structure is a tree of goals, each being a refinement of the one above it in the tree (see Figure 16-5). By *refining a goal,* we mean taking a goal and turning it into more specific subgoals. Looking at Figure 16-5, we see that the top of the tree is a very abstract goal, in this case, to administer digitalis. This goal is refined into less abstract steps by the writer. These more specific steps are steps the system executes to administer digitalis. For example, one such step is to anticipate toxicity, that is, to anticipate whether the patient may become toxic due to increased digitalis sensitivity. The writer then refines this more specific goal to a still more specific goal. Eventually, the level of system primitives is reached. System primitives are operations that are built in. Normally they are very basic, simple operations, so the fact that they cannot be explained is usually

[5]Note that the writer writes the program once, and once written, the program is static. It is not written "on the fly" during interaction with the physician user.

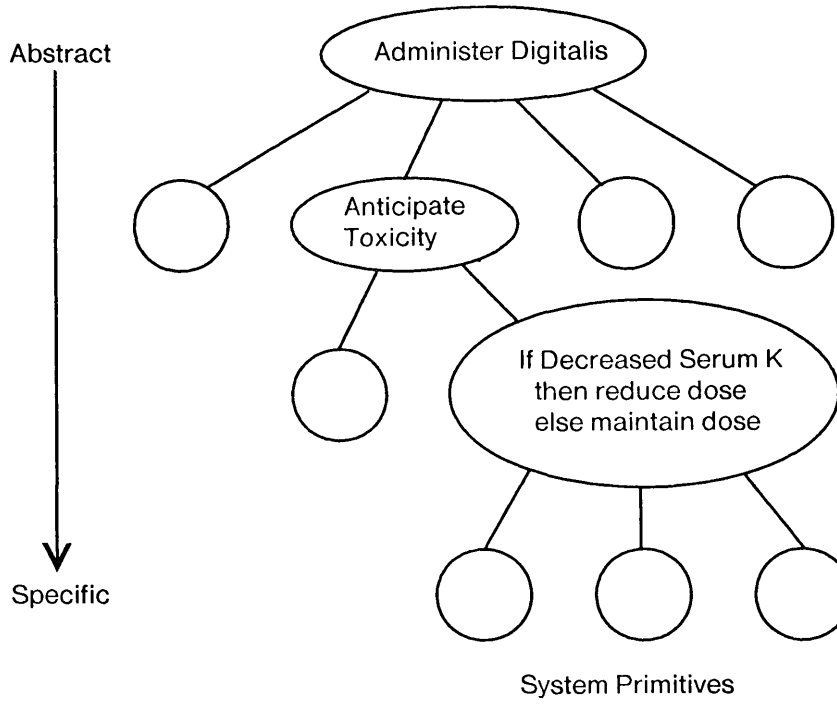


FIGURE 16-5 A sample refinement structure.

not a problem. Typical primitives include those that perform arithmetic operations like PLUS and TIMES and those that set variables to a particular value. The leaves of the refinement structure constitute the basic operations performed by the Digitalis Therapy Advisor, the program that we wanted the automatic programmer to produce.

The domain model is a model of the facts of the domain. In this case, it is a model of the causal relationships in digitalis therapy. A simplified portion of the model is shown in Figure 16-6. In this model, the boxes are states, and the arrows represent causality. This model shows some of the effects of increased digitalis. It also shows that increased serum Ca and decreased serum K can each cause increased automaticity. These facts correspond to the sorts of facts that a medical student learns in class during the first two years of medical school. They are descriptive in the sense that they describe what happens in the domain, but they do not tell how to achieve a goal, such as checking for digitalis sensitivity. The model says that increased digitalis can cause a change to ventricular fibrillation but it does not say what to do about it. Medical students go to medical school for an additional two years, and acquire these procedures by observing more experienced personnel as they practice medicine on the wards. The set of domain principles provides the writer with this sort of problem-solving knowledge.

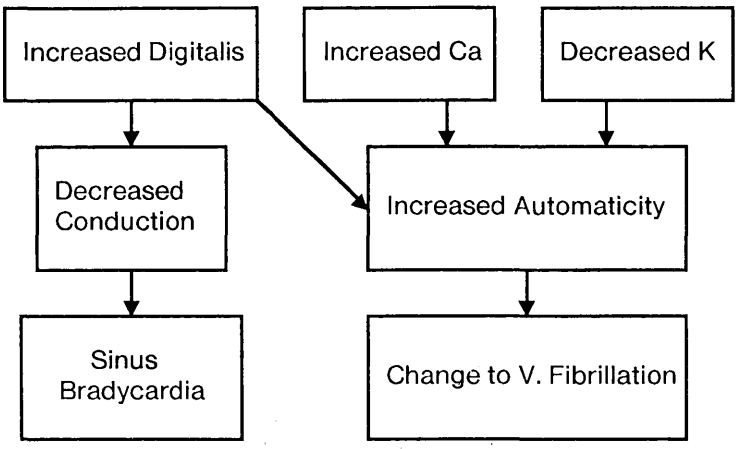


FIGURE 16-6 A simplified portion of the domain model.

Domain principles tell the writer how something (such as prescribing a drug or analyzing symptoms) should be done. They guide it as it refines abstract goals to more specific ones. A (somewhat simplified) domain principle appears in Figure 16-7.[6] This particular principle helps the writer in anticipating digitalis toxicity. It represents the commonsense notion that if one is considering administering a drug and there is some factor that enhances the deleterious effects of that drug, then if that factor is present in the patient, less drug should be given. This principle has three parts: a goal, a domain rationale, and a prototype method.

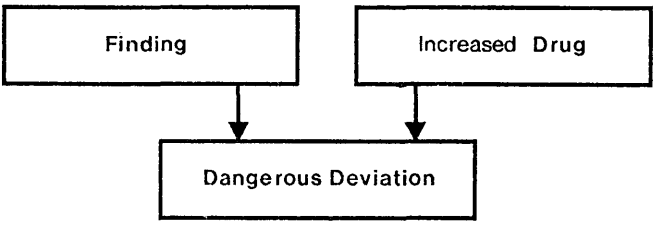
The goal tells the writer what it is that the principle can do. In this case, the principle tells how to anticipate toxicity. The domain rationale is a pattern that is matched against the domain model to provide further information necessary to achieve the goal. In Figure 16-7, arrows represent causality, just as they do in the domain model. Thus the system will look in the domain model to match a **Finding** (e.g., increased Ca) that causes some sort of a **Dangerous Deviation** (e.g., change to ventricular fibrillation) that is also caused by an increased level of the drug. By looking at the domain model, we can see both increased Ca and decreased K will match as findings, since both can cause a change to ventricular fibrillation.

The prototype method is an abstract method that tells the system how to accomplish the goal. The steps of the prototype method are annotated to distinguish implementation details (such as record-keeping) from steps that are significant in medical problem solving. These annotations are used by the explanation routines to filter out implementation details when presenting explanations to medical personnel.

[6]Domain principles are composed of variables and constants. Variables appear in boldface in Figure 16-7. When the writer is matching, a variable in a pattern will match anything that is of the same kind as itself. Thus the variable **Finding** would match increased serum Ca or decreased K, since increased serum Ca and decreased K are both kinds of findings.

Goal: Anticipate Drug Toxicity

Domain Rationale:



Prototype Method:

If the Finding exists

then: reduce the drug dose

else: maintain the drug dose

FIGURE 16-7 An example of a domain principle.

After the domain rationale has been matched against the domain model, the prototype method is instantiated for each match of the domain rationale. When we say that we instantiate the prototype method, we mean that we create a new structure where the variables in the prototype method have been replaced by the things they matched. In this case, two structures would be created. In the first, **Finding** would be replaced by increased serum Ca, and **drug** would be replaced by digitalis. In the second, **Finding** would be replaced by decreased serum K, and **drug** would again be replaced by digitalis. Note that now, with these new structures, we have changed the single abstract problem of how to anticipate toxicity into several more specific ones, such as how to determine whether decreased serum K exists, how to reduce the dose, and how to maintain it.

After instantiation, the more specific goals of the prototype method are placed in the refinement structure as offspring of the goal being resolved. If we look at Figure 16-5, we can see that the instantiated prototype method that checks for decreased serum K has been placed below the anticipate toxicity goal. Once they have been placed in the refinement structure, the newly instantiated goals become goals for the writer to resolve. For example, after the writer applied this domain principle, it would have to find ways of determining whether increased Ca existed in the patient, whether decreased K existed, and ways of reducing and maintaining the dose. The system continues in this fashion, refining goals at the bottom of the structure and growing the tree down until eventually the level of system primitives is reached.

Please enter the value of serum-k: why?

The system is anticipating digitalis toxicity. Decreased serum-k causes increased automaticity, which may cause a change to ventricular fibrillation. Increased digitalis also causes increased automaticity. Thus, if the system observes decreased serum-k, it reduces the dose of digitalis due to decreased serum-k.

Please enter the value of serum-k: 3.7

Please enter the value of serum-ca: why?

(The system produces a shortened explanation, reflecting the fact that it has already explained several of the causal relationships in the previous explanation. Also, since the system remembers that it has already told the user about serum-K, it suggests the analogy between the two here.)

The system is anticipating digitalis toxicity. Increased serum-ca also causes increased automaticity. Thus, (as with decreased serum-k) if the system observes increased serum-ca, it reduces the dose of digitalis due to increased serum-ca.

Please enter the value of serum-ca: 9

FIGURE 16-8 A sample interaction providing justifications.

The system must also take into account interactions between the actions it takes. For example, while the individual instantiations above say that if increased serum Ca exists the dose should be reduced and if decreased serum K exists the dose should be reduced, they do not give any indication of what should happen if *both* increased serum Ca *and* decreased serum K occur. Exactly what should happen depends on the characteristics of the domain. It could be that the occurrence of either sensitivity "covers" for the other, so that only one reduction should be made and the predicate of the If should be made into a disjunction. Or (as is actually the case) it could be that when multiple sensitivities appear, multiple reductions should be made. The way to resolve that is to serialize these two program fragments, connecting the outputs of the first to the inputs of the second. The automatic programmer handles this situation by setting it up as something to be refined. The domain principle used in the refinement of this problem may further constrain the way in which other goals may be refined. The details of this operation will not be presented here. The interested reader should see Swartout (1981).

Once the refinement process is complete, we have a working expert system. A sample interaction with the system is given in Figure 16-8. The first sentence of the explanation was produced by stating the higher goal (that is, anticipate toxicity). Next, the explanation routines located the domain principle that caused the step in question to appear in the program. The domain rationale associated with that principle was then converted to English (with pattern variables replaced by the facts they matched in the domain model). That step produced the next two sentences of the explanation. The last sentence is just the instantiated version of the prototype method of the domain principle. These explanations should be compared

```
(describe-method [(check sensitivities)])

TO CHECK SENSITIVITIES I DO THE FOLLOWING STEPS:

1. I CHECK SENSITIVITY DUE TO CALCIUM.
2. I CHECK SENSITIVITY DUE TO POTASSIUM.
3. I CHECK SENSITIVITY DUE TO CARDIOMYOPATHY-MI.
4. I CHECK SENSITIVITY DUE TO HYPOXEMIA.
5. I CHECK SENSITIVITY DUE TO THYROID-FUNCTION.
6. I CHECK SENSITIVITY DUE TO ADVANCED AGE.
7. I COMPUTE THE FACTOR OF ALTERATION.
```

FIGURE 16-9 An explanation from the old Digitalis Advisor.

with those presented in Figure 16-3 to appreciate the improvement that is possible with this approach. [The generation routines are described in detail in Swartout (1981).]

16.5.2 Explanations of Domain Principles

In the old Digitalis Advisor, when we wanted to give a more abstract view of what was going on, we just described a higher-level procedure (Swartout, 1977). In this regard, we were following the principles of structured programming. While this approach often gave reasonable explanations, there were times when it was considerably less than illuminating. The general method for anticipating digitalis toxicity was called "check sensitivities" in the old Digitalis Advisor. An explanation of it appears in Figure 16-9. While this explanation does tell the user what sensitivities are being checked,[7] it does not say what will be done if sensitivities are discovered, nor does it say why the system considers these particular factors to be sensitivities. Finally, it is much too redundant and verbose. The first objection can be dealt with by removing the calls to lower procedures and substituting the code of those procedures in-line. This results in the somewhat improved explanation produced by XPLAIN when it is asked to describe the method for anticipating digitalis toxicity (see Figure 16-10). However, while this explanation shows what the system does, it does not say why things like increased calcium, cardiomyopathy, and decreased potassium are sensitivities, and if anything, it is even more verbose than the original explanation.

The reason we cannot get the sorts of explanations we want by producing explanations directly from the code is that much of the sort of reasoning we want to explain has been "compiled out." Thus we are forced

[7]The reader may notice that there were more sensitivities checked in the original version of the program than in the current version. We now feel that some of these, such as thyroid function and advanced age, should not be treated as sensitivities *per se* because they tend to have an effect on reducing renal function and hence slowing excretion, rather than on increasing sensitivity to digitalis. The other sensitivities would be easy to add by including the appropriate causal links in the domain model.

(describe-method [((anticipate*o (toxicity*f digitalis))*i 1)])

To anticipate digitalis toxicity:

(1) If the system determines that cardiomyopathy exists, it reduces the dose of digitalis due to cardiomyopathy.

(2) If the system determines that decreased serum-k exists, it reduces the dose of digitalis due to decreased serum-k.

(3) If the system determines that increased serum-ca exists, it reduces the dose of digitalis due to increased serum-ca.

FIGURE 16-10 An explanation from the code for anticipating toxicity.

(describe-proto-method [(anticipate*o (toxicity*f digitalis))])

The system considers those cases where a finding causes a dangerous deviation and increased digitalis also causes the dangerous deviation. If the system determines that the finding exists, it reduces the dose of digitalis due to the finding.

The findings considered are increased calcium and decreased potassium.

FIGURE 16-11 Explanation of a domain principle.

into explaining at a level that is either too abstract or too specific. The intermediate reasoning that we would like to explain was done by a human programmer in the case of the old Digitalis Advisor. However, because the Digitalis Therapy Advisor performance program was produced by an automatic programmer, that reasoning is available in the domain principle. For example, if we were to use the English generator to explain the domain principle that produced the code for anticipating digitalis toxicity rather than the code itself, we would get the explanation that appears in Figure 16-11. Thus the use of an automatic programmer not only allows us to justify the performance program, it also allows us to give better descriptions of methods by making available intermediate levels of abstraction that were not previously available.

16.6 Is Automatic Programming Too Hard?

One possible objection to the whole approach to explanation advocated in this paper is that it is just too hard to get an automatic programmer to write the performance program. Our original plan for producing better explanations was to create structures detailing the development of the performance program, but these structures would be created by hand rather

than automatically, because it was feared that automatic programming was just too hard. However, as the research progressed, it became clear that if we had sufficiently powerful representations available so that it could be said that explanations were being produced from an understanding of the program, then actually writing the program automatically would not be all that much more difficult. This seems to be true in general. It seems that the primary difficulty in both explanation and automatic programming is a knowledge representation problem, and that the kinds of knowledge to be represented in both cases are similar, so that a solution to one makes the other much easier. However, it must be pointed out that the field of automatic programming is still an active research area and a number of difficult problems remain to be solved in addition to the knowledge representation problem, so this conjecture still awaits a final resolution.

16.7 A Summary of Major Points

First, we have argued that to be acceptable, consultant programs must be able to explain what they do and why. Second, we have described the various ways that traditional approaches fail to provide adequate explanations and justifications. Major failings include (1) the inability of such approaches to justify what the system is doing because the knowledge required to produce justifications is not represented within the system, and (2) a lack of distinction between steps required just to get the computer-based implementation to work and those that are motivated by the application domain. Third, we have outlined an approach that captures the knowledge necessary to improve explanations. This involves using an automatic programmer to generate the performance program. As the program is generated, a refinement structure is created that gives the explanation routines access to decisions made during the creation of the program. The improvement in explanatory capabilities that is achieved is due more to the availability of this refinement structure than to the use of more sophisticated English generation functions, since the explanation routines used in this paper do not differ greatly from those used in the old Digitalis Advisor.

ACKNOWLEDGMENTS

This research was supported (in part) by a National Institutes of Health grant (no. 1 P01 LM 03374-01) from the National Library of Medicine. The author wishes to express his thanks to Peter Szolovits for his insightful comments and suggestions during the course of this research.

17

Discovery, Confirmation, and Incorporation of Causal Relationships from a Large Time-Oriented Clinical Data Base: The RX Project

Robert L. Blum

In the mid-1970s Robert Blum, a physician with an interest in medical AI, went to Stanford as a fellow in clinical pharmacology and a doctoral student in computer science. He soon learned about the well-known TOD data base work of James Fries and Gio Wiederhold (the time-oriented data bank that is used as the basis for an international rheumatology network known as ARAMIS—the American Rheumatism Association Medical Information System). Working with Wiederhold, he developed the concept of a computer program to derive new clinical knowledge from such data. His doctoral dissertation, known as RX, used a subset of the ARAMIS data base for this kind of investigation. RX differs from the other systems described in this book because the emphasis is not on consultation but on the use of AI techniques to guide the analysis of collected data. RX is knowledge-based in the sense that it requires not only the observations from a data base, but also underlying knowledge of pathophysiology, causality, and statistics.

As Blum describes in this chapter, the objectives of the RX research are threefold: (1) to automate the processes of hypothesis generation and ex-

From *Computers and Biomedical Research*, 15: 164–187 (1982).

ploratory analysis of data in a large nonrandomized, time-oriented clinical data base, (2) to provide knowledgeable assistance in performing studies on large data bases, and (3) to increase the validity of medical knowledge derived from nonprotocol data (i.e., data that are collected without formal guidelines or an experimental question in mind). In addition to the ARAMIS data and knowledge of pathophysiology and statistics, RX is composed of a discovery module and a study module. The knowledge in the system is organized hierarchically and is used to assist in the discovery and study of new hypotheses. Confirmed results from the data are automatically entered into the knowledge base for future use. Thus the work is related to research in learning, where the goal is to develop programs that can assimilate new knowledge by observing and analyzing past experience.

When RX starts running, it begins the "discovery" process by scanning the ARAMIS data. The discovery module uses lagged, nonparametric correlations to generate hypotheses of clinical interest. These are then studied in detail by the study module, which automatically determines confounding variables and methods for controlling their influence. In determining the confounders of a new hypothesis, the study module uses previously "learned" causal relationships. The study module selects a study design and statistical method based on knowledge of confounders and their distribution in the data base. Most of the RX experiments have used a longitudinal design involving a multiple regression model applied to individual patient records.

The importance of this work lies in its merging of AI, data bases, and statistics and in the thoughtful characterization of causality that Blum has devised. In characterizing the difference between data and knowledge (see Chapter 3), authors have often noted that knowledge is derived from data that are analyzed and validated. In RX we see that this process of data analysis is itself a knowledge-based task. Note, also, that new knowledge, once derived and added to the knowledge base, can then be used to guide further data analyses in the future. The analogy to intellectual growth and learning is clear, but equally evident is the importance of validation before new correlations are accepted as fact. RX continues to be an active area of research for Blum and his colleagues.

17.1 Introduction

Every year, as computers become more powerful and less expensive, increasing amounts of health care data are recorded on them. Motivation for collecting data routinely into ambulatory and hospital medical record systems comes from all quarters. Health practitioners require sets of data for clinical management of individual patients. Hospital administrators require them for billing and resource allocation. Government agencies re-

quire data for assessments of the quality of health care. Third-party insurers require them for reimbursement. Data bases may also be used for performing clinical research, for assessing the efficacy of new diagnostic and therapeutic modalities, and for the performance of postmarketing drug surveillance.

The various uses for data bases may be grouped into two fundamentally distinct categories. The first category pertains to uses that merely require *retrieval of a set of data.* For example, we may wish to know the names of all patients who had a diastolic blood pressure greater than 100 for more than six months and who received no treatment. Uses of medical record systems for patient management, billing, and quality assurance usually fall into this category. The second use of data bases is for *deriving or inferring facts* about the world in general. For example, we might request data from a health insurance data base on occupation and hospital diagnoses to determine whether certain occupations are associated with an increased prevalence of heart disease. Here the predominant interest is in generalizing from the data base and only secondarily in the particular values in the data base. The use of data bases for determining causal effects of drugs, for establishing the usefulness of new tests and therapies, or for determining the natural history of diseases falls into this latter category.

The possibility of deriving medical knowledge from data bases is an important reason for establishing them. Given a collection of large, geographically dispersed medical data bases, it is easy to imagine using them for discovering new causal relationships or for confirming hypotheses of interest.

The RX Project, as this research project is called, is a prototype system for automating the discovery and confirmation of hypotheses from large clinical data bases. The project was designed to emulate the usual method of discovery and confirmation of medical knowledge that characterizes epidemiological and clinical research. The following hypothetical scenario serves to illustrate this method.

17.2 Evolution of Empirical Knowledge

Suppose a medical researcher has noticed an interesting effect in a small group of patients, say unusual longevity. He carefully examines those patients' records looking for possible explanatory factors. He discovers that heavy physical exertion associated with occupation and sports is a possible factor in promoting longevity.

Interested in pursuing the hypothesis that "heavy physical exertion predisposes to long life," the medical researcher consults with a statistician, and together they design a comprehensive study of this hypothesis. First, they analyze the results of the study on their local data base, controlling

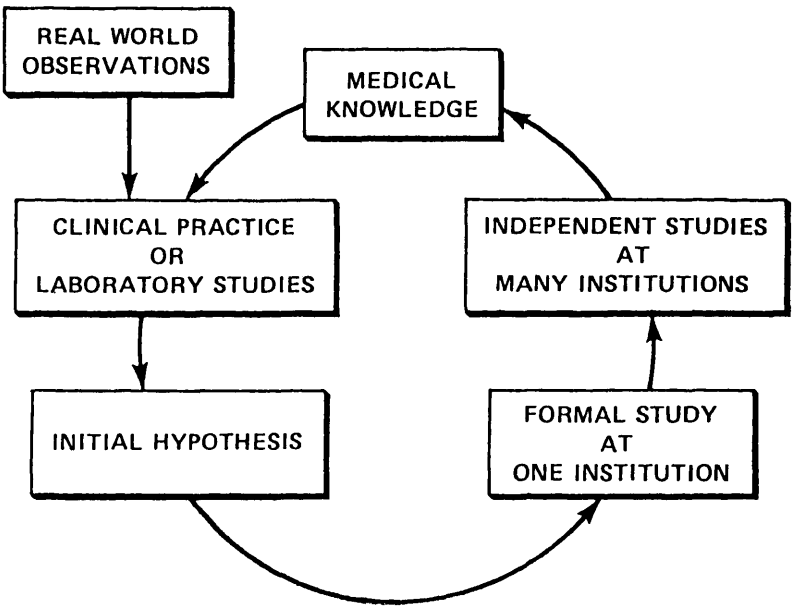


FIGURE 17-1 The evolution of medical knowledge.

for factors known to be associated with longevity. Having confirmed the hypothesis on one data base, they proceed to test the hypothesis on many other data bases, modifying the study design to allow for differences in the type and quantity of data. Having confirmed the hypothesis, they publish the result, and other researchers proceed with further confirmatory studies, attempting to elucidate the mechanism of the "exercise effect." When future researchers study other factors that influence longevity, they control for physical activity.

This cycle in which knowledge gradually evolves from data through a succession of increasingly comprehensive studies is illustrated in Figure 17-1. At each stage of discovery and confirmation existing medical knowledge is used to design and to interpret the studies.

17.3 The RX Project

It is easy to imagine automating at least parts of the above cycle of discovery and confirmation. We obtain our initial hypotheses by selectively combing through a large data base, examining a few patient records guided by prior knowledge. These clues are then studied more comprehensively on the

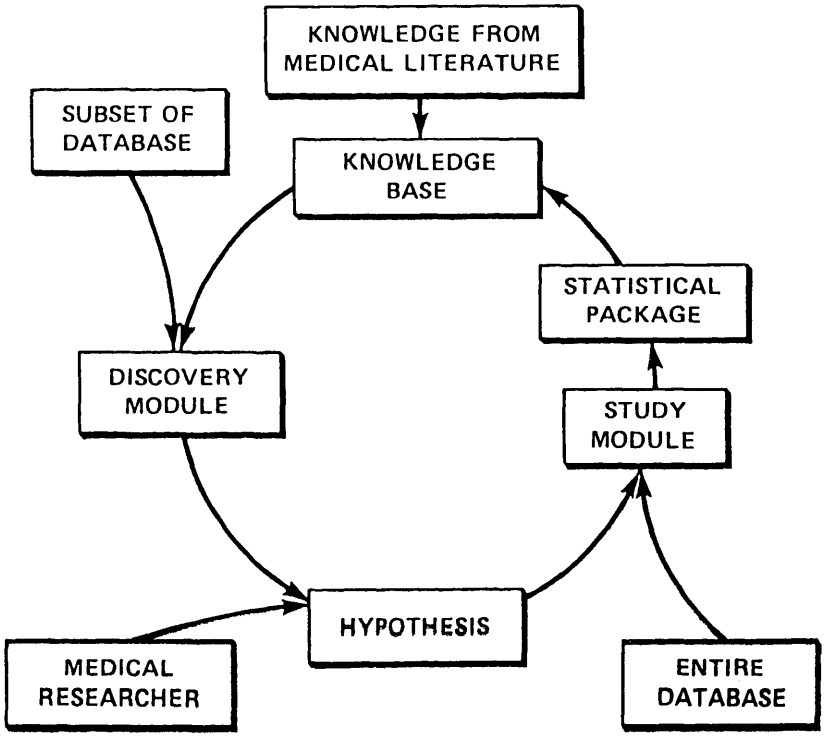


FIGURE 17-2 Discovery and confirmation in RX.

data base as a whole. To design and interpret these studies, medical and statistical knowledge from a computerized knowledge base is used. The final results are incrementally incorporated into the knowledge base, where they can be used in the automated design of future studies.

This describes the RX computer program, a prototype implementation of these ideas. Besides a data base, the RX program consists of four major parts: the discovery module, the study module, a statistical analysis package, and a knowledge base (Figure 17-2).

- The *discovery module* produces hypotheses of the form "*A* causes *B*." The hypotheses denote that in a number of individual patient records "*A* precedes and is correlated with *B*." Information from the knowledge base is used to guide the formation of initial hypotheses.
- The *study module* then designs a comprehensive study of the most promising hypotheses. It takes into account information in the knowledge base in order to control for known factors that may have produced a spurious association between the tentative cause and effect. The study module

uses statistical knowledge in the knowledge base to design an adequate statistical model of the hypothesis.

- The *statistical analysis package* is invoked by the study module to test the statistical model. The analysis package accesses relevant data from patient records, and then applies the statistical model to the data. The results are returned to the study module for interpretation.
- The *knowledge base* is used in all phases of hypothesis generation and testing. If the results of a study are medically and statistically significant, they are tentatively incorporated into the knowledge base where they are used to design further studies. Newly incorporated knowledge is appropriately labeled as to source, validity, evidential basis, and so on. As the knowledge base grows, old information is updated.

Currently, the RX program uses only one data base: a subset of the ARAMIS data base. Also, the extent of medical and statistical knowledge is limited, since the purpose of the research was primarily the development of methodology.

While the program is a prototype, it has been operational since 1979 and has been widely demonstrated. Several interesting medical hypotheses (in varying states of confirmation) have been discovered by the program, including some with little prior supporting evidence.

The objective of this chapter is to present an overview of the RX Project. Details on statistical methods, modeling of causal relationships, and methods of knowledge representation may be found in Blum (1982).

17.4 Time-Oriented Data Bases

The general format of a patient record is illustrated in Table 17-1. Each time a patient is seen in a clinic a number of observations are made. These are recorded with the date of observation in the data base. The recorded

TABLE 17-1 Hypothetical Time-Oriented Record for One Patient

VISIT NUMBER:	1	2	3
DATE:	17 Jan 79	23 Jun 79	1 Jul 79
KNEE PAIN:	severe	mild	mild
FATIGUE:	moderate		moderate
TEMPERATURE:	38.5	37.5	36.9
DIAGNOSIS:	systemic lupus		
WHITE BLOOD COUNT:	3500	4700	4300
CREATININE CLEARANCE:	45		65
BLOOD UREA NITROGEN:	36	33	
PREDNISONE:	30	25	20

characteristics of a patient are known as *primary attributes*, or simply *attributes*. Attributes may be real-valued, rank, categorical, or binary. The term *attribute* includes all recorded signs and symptoms, lab values, diagnoses, therapy, and functional states.

The defining characteristic of a time-oriented data base is that *sequential values for each attribute may be recorded.* Note that different attributes may be recorded on different patients and that the time intervals between values will usually differ. Some attributes may have values that are only sporadically recorded or not at all. In general, the quantity and character of data across patients may vary greatly.

All of the research reported here was done using a subset of the ARAMIS/TOD data base of rheumatology (American Rheumatism Association Medical Information System/Time-Oriented Databank) collected at Stanford University from 1969 to the present (Fries, 1972; Wiederhold et al., 1975). The subset contains the records of 50 patients with severe systemic lupus erythematosus (SLE). The average number of clinic visits for each patient was 50, and the average length of follow-up was five years. Patient records contained 52 attributes.

The size of the data base used in this project, a small sample of the ARAMIS data base, is approximately a half-million characters—much greater than available core storage on our computers after programs have been loaded. Patient records are kept in hash files on disk, where they are stored in compressed and transposed format. Indices for each attribute are maintained specifying numbers of values for each patient. Details of data storage and display methods may be found in Blum (1981).

17.5 Computer Facilities and Languages

Research was performed at two computer facilities at Stanford University: SUMEX-AIM and SCORE. At the time SUMEX-AIM featured a DEC dual processor KI-10 running the TENEX operating system. Currently both SUMEX and SCORE have DEC 20/60's running TOPS-20. The ARAMIS data base *per se* is stored at the Stanford Center for Information Technology on an IBM 370/3033. Data transfer was accomplished by magnetic tape.

All computer programs are written in Interlisp (Teitelman, 1978), a dialect of LISP, a language that is highly suitable for knowledge manipulation. Statistics are performed in IDL (Interactive Data-Analysis Language) (Kaplan et al., 1978), discussed later. The RX source code with knowledge base comprises approximately 200 disk pages of 512 words each.

17.6 The Knowledge Base

While the prospect of using clinical data bases to discover or to confirm medical hypotheses is tantalizing, there are formidable problems in making inferences from nonrandomized, nonprotocol data. These include numerous forms of treatment and surveillance bias, poor adjustment for covariates, inadequate specification of patient subsets, and improper use of statistical analysis (Blum and Wiederhold, 1978; Byar, 1980; Dambrosia and Ellenberg, 1980). The use of nonrandomized data for clinical inference demands more stringent data analysis, study designs of greater sophistication, and more thoughtful interpretation than does the use of data gathered in a randomized trial.

The leitmotif of the RX Project is that derivation of new knowledge from data bases can best be performed by integrating existing knowledge of relevant parts of medicine and statistics into the medical information system. During the evolution of a medical hypothesis, as was illustrated, existing medical knowledge comes into play at every stage.

In the RX computer program the medical knowledge base determines the operation of the discovery module, plays a pivotal role in the creation of subsequent studies in the study module, and, finally, serves as a repository for newly created knowledge. The medical knowledge base grows by automatically incorporating new knowledge into itself. Hence it must be designed in such a way that relationships derived from the data base can be translated into the same machine-readable form as knowledge entered from the medical literature by a researcher. In any case knowledge relevant to a study must be automatically accessible.

The main data structure of RX's knowledge base (KB) is a tree representing a taxonomy of relevant aspects of medicine and statistics. Each object in the tree is represented as a schema containing an arbitrary number of property:value pairs. The RX KB contains approximately 250 schemata pertaining to medicine, 50 pertaining to statistics, and 50 pertaining to the overall system. The medical knowledge in the RX KB covers only a small portion of what is known about systemic lupus erythematosus and some areas of general medicine. The present KB is merely a test vehicle; its size is 50 disk pages or 120,000 bytes.

17.6.1 Medical Knowledge

The medical knowledge base is a subtree of the KB distinct from the statistical knowledge base. Its first-order subtrees are *states* and *actions*, which in turn are broken down into *signs, symptoms, lab findings,* and *diseases* and into *drugs, surgery,* and *physical therapy*. The categories of diseases and other entities follow the conventional nosology based on organ systems and pa-

thology found in any standard textbook of medicine (Isselbacher et al., 1980). I will occasionally refer to each of the objects in the medical KB as a *node* and to the information stored at each node as its *schema.*

The schema for each object is represented as a collection of property:value pairs called a *property list.* In general, the objects in the KB are either primary attributes in the data base or *derived variables,* that is, objects whose values must be derived from primary data. The properties in an object's schema may be grouped into the following categories: *data base schema properties, hierarchical relationship properties, properties describing the definition of an object and its intrinsic properties,* and *properties describing cause/effect relationships to other objects.*

Data Base Schema Properties

Each of the attributes in the clinical data base is represented by a schema in the KB describing its units of measurement, how its values are stored, and so on. This kind of schema is typical of most data bases today (Wiederhold, 1977). As an example, part of the schema for the attribute hemoglobin appears below:

Hemoglobin
---
attribute-type: point-event
value-type: real {i.e., a real-valued number}
range: 0 < value < 25
significance: .1 {i.e., values are rounded to the nearest .1}
units: grams per deciliter

Hierarchical Relationship Properties

Two properties are used to store the position of an object in the medical hierarchy: *specialization* and *generalization,* abbreviated *spec* and *genl* as shown below.

Respiratory Diseases
genl: All Categories of Disease
spec: Pneumonia, Asthma, Emphysema

Pneumonia	Asthma	Emphysema
genl: Respiratory Dis.	*genl*: Respiratory Dis.	*genl*: Respiratory Dis.
spec: Pneumoncoccal Pn.	*spec*: Allergic Asthma	*spec*: CO_2 retention
Klebsiella Pn.	Intrinsic Asthma	

Inheritance mechanisms (Stefik, 1979) are used by the study module as a means for exploiting the knowledge implicit in the hierarchy. For example, in the course of a study, if the expected duration of klebsiella pneumonia was required to construct a statistical model, then a default value might be inherited from the schema for pneumonia.

Properties Pertaining to the Definition and Intrinsic Characteristics of an Object

If an object is a primary data base attribute like hemoglobin, then no definition is required, at least not from a standpoint of deriving values for it. Values for hemoglobin are simply those in the data base.

On the other hand, if the values for an object are derived from primary attributes, the specification of the means for derivation must be recorded in the KB. That is the object's definition. The didactic example below shows a definition for pneumonia.

Pneumonia

definition:		Temperature > 38 degrees C.
	and	WBC > 10,000 cells per mm^3
	and	Chest-XRAY = Lobar Infiltrate

In the RX KB the specification and use of definitions are far more complicated than is suggested by this example. Recall that data base attributes are time-oriented with nonuniform time intervals and frequently missing values. Hence definitions of derived objects must contain time-dependent predicates and mechanisms for handling sporadic values. Definitions can also refer to other derived objects. The temporal characteristics of an object may be specified using other properties in the schema: *expected duration, carry-over, onset delay,* and so on. These parameters are used by the time-dependent predicates when definitions for objects are evaluated.

Properties Specifying Causal Relationships to Other Objects

The final class of properties are those specifying the causal relationships of an object to other objects. In RX all causal relationships are stored using two properties: *effects* and *affected-by.* The *effects* property records a list of those objects directly affected by the object. The *affected-by* property contains a list of objects that directly affect it. Additionally, the detailed characteristics of the causal relationship between a pair of objects is stored on the *affected-by* property. The resulting causal model is a directed cyclic graph; that is, the representation allows for the possibility that A causes B causes A.

Besides the simple fact that A may affect B, each causal relationship is represented by a set of features as follows:

<intensity, frequency, direction, setting, functional form, validity, evidence>

Briefly, these take the following form when both the cause and effect are real-valued:

- *intensity*: the expected change in the effect given a change in the cause, expressed as an unstandardized regression coefficient
- *frequency*: the distribution of the effect across patients, expressed as deciles of the expected effect given a "strong" change in the causal variable
- *direction*: increase or decrease
- *setting*: the clinical circumstances specifically included or excluded from the study, expressed as a Boolean with time-dependent predicates
- *functional form*: the complete statistical model used to study the relationship, expressed in machine-readable form
- *validity*: a 1-to-10 scale distinguishing tentative associations from widely confirmed causal relationships
- *evidence*: a summary of the study performed by the study module, including patient ID's, methods, and intermediate results

The entire causal relationship is machine-readable. This enables it to be used automatically by the study module during subsequent studies. The causal relationships in the KB can also be interactively displayed in a variety of forms. All paths connecting two nodes may be displayed, or only the details of a particular causal relationship: its mathematical form, the evidence supporting it, or its distribution across patients. In the following example the effects of prednisone have been displayed. The verbs and adverbs in the phrases are supplied by a lexicon during machine translation.

```
PREDNISONE, at a level of 30 mgms/day,     {modal effects}
-------------------------------------------
usually increases CHOLESTEROL by 50 to 130 mgms/dl,
usually increases WEIGHT by 3 to 7 kgms,
regularly attenuates NEPHROTIC-SYNDROME by 1. to 2. gms protein/24 hrs,
regularly attenuates GLOMERULONEPHRITIS by 10. to 30. percent activity,
regularly decreases EOSINOPHILS by 2 to 3 percent of WBC,
commonly decreases ANTI-DNA by 50 to 90 percent activity,
occasionally increases GLUCOSE by 20 to 100 mgms/dl.
```

17.7 The Discovery Module

The general methodology used by RX to discover and then to study causal relationships is known as a generate-and-test algorithm. Briefly, the discovery module proposes causal links based on a test for strength of association and time precedence. After a number of tentative links have been added, the study module performs an exhaustive study of them in the same order in which they were added. In the course of this study many tentative links will be removed, and the remaining ones will be labeled with

detailed information on the respective relationships. After a link has been incorporated into the model, it may be used to refine the study of further links.

17.7.1 An Operational Definition of Causality

Underlying the discovery module and the study module is the following operational definition of causality: *A* is said to cause *B* if over repeated observations (1) *A* generally precedes *B*, (2) the intensity of *A* is correlated with the intensity of *B*, and (3) there is no known third variable, *C*, responsible for the correlation.

These properties are the foundation of the RX algorithm. I will refer to these properties as (1) time precedence, (2) covariation or association, and (3) nonspuriousness (Kenny, 1970; Suppes, 1970).

Causality can never be proven using observational data. The persuasiveness of a given demonstration simply depends on the extent to which the three properties have been shown.

17.7.2 Methodology of the Discovery Module

The function of the discovery module is to find candidate causal relationships. To do this, the discovery module exploits only the first two properties of causal relationships: time precedence and covariation.

The discovery module considers all pairs of variables, $\{A, B\}$, where *A* and *B* are either primary attributes in the data base or are derivable from primary attributes. It attempts to determine whether the data suggest that *A* causes *B*, *B* causes *A*, both, or neither. The output of the discovery module is an ordered list of hypotheses. A researcher may designate which potential causes and effects are of interest. For example, certain drugs and diseases might be tagged as being of interest in exploration. The algorithm is intrinsically slow, $O(n^2)$ where *O* is Order and *n* is the number of variables; however, it makes up for this inefficiency by its sensitivity and the speed with which simple correlations can be performed.

A pairwise algorithm was chosen for the discovery module after months of experimentation with multivariate methods. The latter cannot be applied to data of the type recorded in the ARAMIS data base without extensive loss of information. The reason is that values are only sporadically recorded and patients differ widely on covariates. The general philosophy in all RX procedures in either the discovery module or the study module is to analyze data only within *individual patient records*. That is, data in two patient records are never combined before statistical analysis. The computational expense incurred by analyzing individual patient records will decrease markedly when multi-CPU machines become standard.

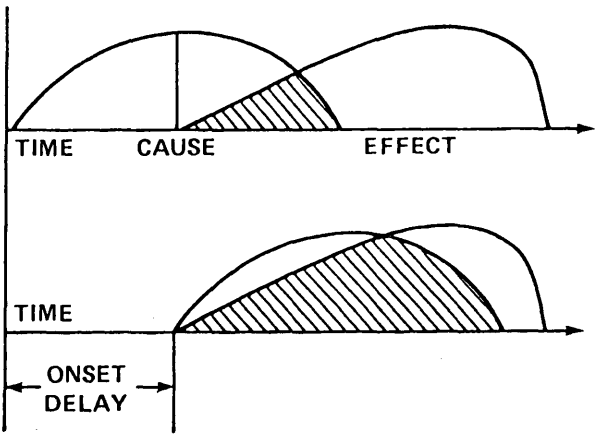


FIGURE 17-3 The principle underlying lagged correlation.

The basic algorithm uses a sliding nonparametric correlation performed on data from an individual patient's record. The principle underlying a lagged correlation is illustrated in Figure 17-3. Given a tentative cause, A, and an effect, B, the basic tool for uncovering a causal relationship is the Spearman correlation coefficient, $r_s(A, B, \tau)$, where τ is the time delay used in computing the correlation.

Selection of Patients for Correlation

In the discovery module only a sample of the patient records are analyzed. The sampling procedure uses a precomputed index called a *records list* associated with every variable in the data base. The records list is a sorted list of the form $((\text{patient}_1, n_1), (\text{patient}_2, n_2), \ldots (\text{patient}_m, n_m))$. The list identifies patients in descending order by their number of recorded values for the variable. That is, patient_1 has n_1 measurements of the variable, and so on.

The sample of records that are analyzed for a given pair of variables, $\{A, B\}$, is the sample $P^*_{\{A, B\}}$, where this is the set with the largest number of pairs of measurements of A and B. Let K denote the number of pairs in the set $P^*_{\{A, B\}}$. In experimental trials of the discovery module, K was set to 10.

The advantage of choosing the sample to be those patients with the most data on A and B is that "one might as well look where the looking is best." If a relationship exists between A and B, then it will be easiest to detect in patients with lots of data on A and B. This heuristic is particularly

valid for medical data when variables are more apt to be recorded when they are abnormal. Therefore, the frequency of observation tends to be correlated with the variance of the variable.

Correlations for the records in $P^*_{\{A, B\}}$ are computed as follows:

for each record in $P^*_{\{A,B\}}$ collect
[for each τ in T^* collect $r_s(A, B, \tau)$]

The *collect operator* denotes assembling a set composed of the value of each iterand. The time delays in T^* over which the correlations are performed are based on information from the knowledge base. That is, the algorithm makes use of prior information on the expected time delays of broad classes of causes and effects.

Combining Correlations Across Patients

That various correlations within and across patient records are based on different numbers of measurements poses a difficulty in combining them. Given equal correlations, we would like to assign more weight to records with more data. Using the p-value of the correlation achieves this and also facilitates combining correlations.

The p-values from the above procedure may be diagrammed as follows:

	τ_1	τ_2	. . .	τ_q
patient_1	$p_{1,1}$	$p_{1,2}$		$p_{1,q}$
patient_2	$p_{2,1}$	$p_{2,2}$		$p_{2,q}$
.	.	.		.
.	.	.		.
.	.	.		.
patient_K	$p_{K,1}$	$p_{K,2}$	. . .	$p_{K,q}$

Here p_{ij} denotes the p-value on the ith patient at the jth time delay. By the method of Fisher, the p-values may be combined to form an overall score s for each time delay τ_j:

$$s(A, B, \tau_j, P^*_{\{A, B\}}) = -2\Sigma\log(p_{i,\tau_j})$$

where the sum is over all patient records in $P^*_{\{A, B\}}$. It can be shown (Mood et al., 1974) that the scores s are distributed as χ^2 on $2p$ degrees of freedom. Since the distribution of the scores is known, their statistical significance may be calculated. Because of autocorrelation, the differences between scores determined at different time lags may not be distributed as χ^2. How-

ever, the significances are not taken literally by the discovery module, but are merely used to rank the hypotheses in terms of promise.

If the difference between the forward and backward sets of scores is large, a strong time precedence of association is implied. Since time precedence is not a sufficient condition for causality, spurious associations may also be reported as significant.

The output of the discovery module is a list of dyadic relations ranked in descending order by strength of unidirectionality of association. The algorithm has proven to be a sensitive, if nonspecific, detector of causal relationships, and is usually capable of accurately discriminating time precedence and determining approximate onset delay.

In the discovery module, only the properties of time precedence and covariation are used in a blind search for clues to causal relationships. Included in its output are many spurious relationships. The objective of the study module is to eliminate those relationships and to carefully examine those that remain in order to detail their characteristics and to store them in the KB.

17.8 The Study Module

The study module is the core of the RX algorithm. It takes as input a causal hypothesis obtained either from the discovery module or interactively from a researcher. It then generates a medically and statistically plausible model of the hypothesis, which it analyzes on appropriate data from the data base.

The study module is patterned after a sequence of steps usually undertaken by designers of large clinical studies. Its design may be considered an exercise in artificial intelligence insofar as it emulates human expertise in this area. There are at least six persons whose knowledge is brought to bear in designing, executing, reporting, and disseminating a large data base study. We may think of the *data base research team* as consisting of a doctor, a statistician, an archivist, a data analyst, a technical writer, and a medical librarian. The study module, in conjunction with the knowledge base (KB), emulates part of their expertise. The steps performed by the study module appear in Table 17-2.

17.8.1 Determination of the Feasibility of a Study

The study module may be operated automatically in batch mode, or it may be run interactively, enabling a researcher to modify the evolving study design. In this presentation we will assume that it is being run interactively.

TABLE 17-2 Steps Performed by the Study Module

1. Parse the hypothesis.
2. Determine the feasibility of the study on the data base.
3. Select confounding variables and causal dominators.
4. Select methods for controlling the causal dominators.
5. Determine proxy variables.
6. Determine eligibility criteria.
7. Create a statistical model.
   a. Select an overall study design.
   b. Select statistical methods.
   c. Format the appropriate data base access functions.
8. Run the study.
   a. Fetch the appropriate data from eligible patient records.
   b. Perform a statistical analysis of each patient's record.
   c. Combine the results across patients.
9. Interpret the results to determine significance.
10. Incorporate the results into the knowledge base.

Throughout this section we will use as an example the hypothesis that the steroid drug prednisone elevates serum cholesterol.

The first general task of the study module, or of the "data base research team," is to determine whether a particular study is feasible given the knowledge and the data available. The first step is the recognition by the program of the terms used in the hypothesis.

Suppose a researcher enters the hypothesis "prednisone elevates cholesterol." A top-down parser is applied to this input string. The pattern that matches is <variable relationship variable> where a variable may be any primary attribute or derived variable in the medical KB. As the parser matches the tokens in the input, it determines their classification in the KB.

```
Prednisone is a known concept.
It is classified as a Steroid which is a Drug which is an Action.

Elevates is a known concept.
It is classified as a Relationship.

Cholesterol is a known concept.
It is classified as a Chemistry which is a Lab-Value which is a State.
```

The classifications are simply determined by following the *generalization* pointers in the knowledge tree. The classification of each variable is not only of interest to the user but facilitates the inheritance mechanisms discussed above. For example, properties of the class steroids may be inherited by the drug prednisone, if they are needed in the course of the study.

To study the relationship between prednisone and cholesterol both variables must have been recorded in some patient records. Hence, the program next examines the intersection of their records lists.

The following list denotes that patient 78 had 32 recorded values for cholesterol, patient 118 had 25 values, and so on.

Cholesterol
---
records: ((P78 32) (P118 25) . . . (P967 1))

17.8.2 Confounding Variables and Causal Dominators

The principal objective of the study module is the demonstration of *nonspuriousness*. In any observational drug study, as in the current one, the possibility must always be addressed that the effect of interest was caused by the disease for which the drug was given rather than by the drug itself. The first step in demonstrating nonspuriousness is identifying the set of possible confounding variables.

A confounding variable is any node, C, that may cause a clinically significant effect on both the causal node, A, and the effect node, B, in our hypothesis. The "clinical significance" of a given change in a variable is determined by a prior partitioning of that variable's range. Every real-valued object in the knowledge base has stored in its schema a *partition list* that divides its range into clinically significant regions.

Let C be the set of known confounders. The determination of C involves tracing the directed graph in the KB starting from A and B.

$$C = \text{Intersection}[\text{Antecedents}(A), \text{Antecedents}(B)]$$

where the list Antecedents(A) is the set of nodes that may produce a clinically significant effect on A. The *antecedents set* of a node is calculated by traversing the causal network in the KB. In the current example, the set C is determined to be {ketoacidosis, hepatitis, glomerulonephritis, nephrotic syndrome}.

Having determined the variables in C, the program displays the causal paths connecting them to A and B. The paths for glomerulonephritis appear below. The intensities of intermediate nodes are calculated using the regression coefficients stored in sequential causal relationships.

Glomerulonephritis {50 percent activity} is treated by Prednisone {30 mgms/day},

Glomerulonephritis can cause Nephrotic Syndrome {4 gms proteinuria/24 hrs} which is treated by Prednisone {20 mgms/day},

Glomerulonephritis can cause Nephrotic Syndrome {4 gms proteinuria/24 hrs} which increases Cholesterol {65 mgms/dl}.

17.8.3 Causal Dominators

To increase statistical power and stability of estimation it is usually desirable to control for as few confounding variables as possible. Since the set C in any real study is apt to be quite large, it is desirable to control for only the essentials. The set of *causal dominators*, C^*, is the smallest subset of C through which all known causal influences on both A and B flow.

The set of causal dominators, C^*, is determined by the following algorithm. The nodes in C are sorted into descending order according to their expected magnitude of effect on the relationship between A and B. More potent confounders appear earlier in the list. To determine C^*, the nodes in the ordered list are checked to determine whether paths to A and B still exist after earlier (more proximal) nodes have been blocked. In the current example, glomerulonephritis is deleted from the confounders since its confounding influence is entirely through nephrotic syndrome.

17.8.4 Controlling Variables Related to the Cause

Suppose prednisone affects cholesterol in some fashion; it is possible that related drugs may also affect cholesterol. We may also want to remove their influence by controlling them. Generally, we would like the program to suggest to us variables related to the cause, since they may also be confounders. These variables may not be in the set C, since causal paths between them and the effect may be unknown.

To select this set of variables related to the causal variable, the program uses the hierarchical structure of the KB. For example, since prednisone is one of the steroids, RX controls for the other steroids [i.e., *siblings* (prednisone) = {dexamethasone ACTH}, the other nodes in the same class, steroids].

17.8.5 Determination of Methods for Controlling Confounding Variables

Three general methods are used by RX to control confounding variables: (1) eliminating entire patient records, (2) eliminating time intervals containing confounding events, and (3) controlling statistically for the presence of the confounder. Eliminating patient records is always the safest and most intellectually reassuring. With statistical control, doubt always remains as to whether the confounder has been entirely eliminated. When eliminating time intervals, there is always the possibility that the confounding influence extends beyond the interval. On the other hand, eliminating patient records is the strategy most wasteful of data. There may be too few records left to analyze, or the generalizability of the result may be diminished.

To determine which method to use for each confounder, some decision criteria must be used. In making this decision and others discussed later, the study module uses decision criteria stored in the KB in the form of *production rules.*

17.8.6 Production Rules

Production rules have been widely used in artificial intelligence research to store domain knowledge (Shortliffe et al., 1975) (see also Chapter 5). A production rule is an IF/THEN rule consisting of a premise and conclusion.

The rule below is stored with other similar rules in the schema for control methods. To choose a control strategy, the rules are exhaustively invoked. Some rules may be used to resolve conflicts, if more than one control method is suggested.

IF the number of patients affected by a variable
is a small percentage of the number of
patients in the study,

AND the variable is present throughout those records,

THEN eliminate those records from the study.

The premise and conclusion of each production rule consists of a few lines of machine-readable code. In some systems (Shortliffe et al., 1975), the code may be mechanically translated into English upon request. To avoid the attendant complexity and to improve the quality of translation, the RX KB simply stores an English translation of each production rule.

In writing programs that use much domain knowledge, it is advantageous to separate the specific knowledge from the general algorithms that use it. Production rules are one method for achieving this modularity. The advantages are that (1) knowledge is more easily examined and updated, (2) dependencies among the knowledge are more easily discovered, and (3) the homogeneous format lends itself to machine translation.

17.8.7 Controlling Confounders

To determine how a particular confounder is to be controlled, the following information is first determined: N, the number of patient records in the study; *%records,* the fraction of records affected by the confounder; and *%visits,* the average fraction of visits affected. Each of these parameters is calculated using the information in the records list for each confounding variable.

If %records or %visits are low, then either records or time intervals may be eliminated. The rules tend to favor the elimination of records if N

is high. Only if N is low and %records or %visits is high is statistical control of the confounder considered.

While the program is running, the user may request a display of the rules that determined the choice of strategy. The user, as always, may override the decision made by the program.

In the prednisone/cholesterol study the program makes the following selections:

Dexamethasone	No control needed, since no values were recorded in the database
ACTH	No control needed
Nephrotic Syndrome	Control statistically using albumin as a proxy
Hepatitis	Eliminate affected time intervals
Ketoacidosis	Eliminate affected time intervals

17.8.8 Choice of Study Design and Statistical Method

Both the study design and the statistical method are selected using decision criteria stored in production rules in the KB. The choice of study design in the present system is simply a choice between a cross-sectional and a longitudinal design. In a cross-sectional design each variable is sampled once in a patient's record; in a longitudinal design variables are repeatedly sampled over time. The longitudinal study design has the advantage of making use of temporal information and multiple observations of variables within individual patient records. A cross-sectional design is only chosen when a longitudinal design is not feasible.

The selection of a particular statistical method uses knowledge encoded in a hierarchically organized, statistical knowledge base. The organization follows the conventional classification as in Armitage (1971) or Brown and Hollander (1977).

On the property list of each node in the tree is an *objectives*, a *prerequisites*, and an *assumptions* property. The objectives property describes the goals of the method. The prerequisites property describes the conditions that must hold for the method to be mechanically applied. The assumptions property describes the assumptions that must hold for the result to be valid.

An example of the schema for multiple regression appears below. The schema stores not only the English text but the equivalent machine-executable code.

```
Multiple-Regression
-------------------
objectives: linear-model
prerequisites:
    one dependent variable
    two or more independent variables
    measurement-level of dependent variable = real valued
```

measurement-level of independent variables = real valued
number of observations > 1 + number of independent variables
assumptions:
independent and identically distributed errors
normally distributed errors
linear and additive effects

To select a statistical method the objectives and prerequisites properties must satisfy the constraints of the study. The tree structure of the KB is used to prune limbs that are not applicable. When there is more than one applicable method, production rules at intermediate nodes arbitrate among methods. The present program does not determine whether the assumptions of a method have been fulfilled; they are merely displayed. However, it does make available tables and plots of residuals, so that the assumptions can be manually checked.

The present version of this *robot statistician* is rudimentary. Each of the nodes in the statistical KB contains about as much knowledge as is shown for multiple regression. No knowledge or methods are present for critically analyzing a fitted model or for revising the model. The current emphasis is simply on selecting a method that may be mechanically applied.

17.8.9 Formatting of Data Base Access Functions

In order to apply the selected analytical methods to the appropriate data, the data must be sampled from patient records at times that reflect the time delays inherent in the underlying processes. These time parameters are obtained by the study module from information in the KB.

For the longitudinal design in the present example the following model is created:

$$\Delta\text{cholesterol} = \beta_0 + \beta_1 \Delta\text{albumin} + \beta_2 \Delta\log(\text{prednisone})$$

where

$$\Delta\text{cholesterol} = \text{cholesterol}(t) - \text{cholesterol}(t_{\text{pchol}})$$

$$\Delta\text{albumin} = \text{albumin}(t - \tau_{\text{NS}}) - \text{albumin}(t_{\text{pchol}} - \tau_{\text{NS}})$$

$$\Delta\log(\text{prednisone}) = \log[\text{prednisone}(t - \tau_{\text{pred}})] - \log[\text{prednisone}(t_{\text{pchol}} - \tau_{\text{pred}})]$$

The time t_{pchol} denotes the time of the preceding measurement of cholesterol (previous to the present one), and τ_{NS} denotes the estimated delay from the start of nephrotic syndrome to the establishment of a steady state for cholesterol. The symbol τ_{pred} is the analogous onset delay for prednisone. No values are sampled during episodes of hepatitis or ketoacidosis. Figure 17-4 illustrates some of the time relationships that might be seen in one patient's record.

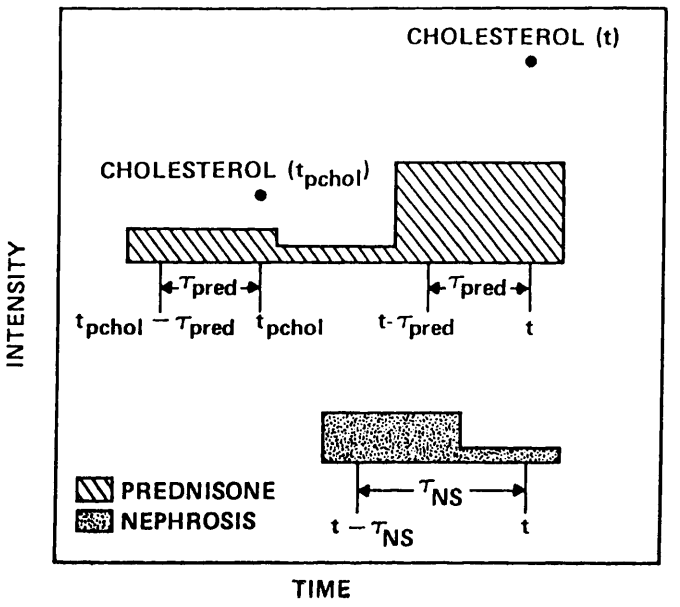


FIGURE 17-4 Time relationships in prednisone/cholesterol study.

Next, the mathematical model must be translated into the appropriate data base access functions. The function *create-access-functions* uses information in the schemata for the variables in the model to format the appropriate access functions. For example, the values for the onset delays and the indicator that there is a need for the log transform are retrieved from the schemata for nephrotic syndrome and prednisone. The estimated time delay for the effect of prednisone on cholesterol is obtained from the discovery module.

17.8.10 Determination of Eligibility Criteria

All patients in a data base may not be eligible for a particular study. Eligibility criteria in the current example are automatically formatted based on the number of relevant observations in a patient's record and the within-patient variance in the causal variable.

The study design cannot be executed on patient records in which there are less than four sets of observations (note that there is 1 degree of freedom for the mean plus 2 degrees of freedom for Δalbumin and for Δprednisone). Furthermore, patient records are excluded in which the coefficient of variation in log(prednisone) is below threshold.

17.8.11 Statistical Analysis: Fitting the Model

Until July 1980, all statistical analyses were performed using SPSS (Nie et al., 1975) as a subroutine; however, this incurred the inefficiency of having to write and read files in formats intended for human usage. Currently, all statistical analysis is performed using IDL (Kaplan et al., 1978). Written in Interlisp, IDL makes available fast numerical computation, matrix manipulation, and a variety of primitive operators for statistical computation.

Most of our studies are sufficiently large that statistical analysis requires use of a separate core image (separate job). The study module writes the study design to disk, then calls IDL. IDL reads the study design, executes it, writes the results to disk, and then calls the study module.

Longitudinal Design Using Weighted Multiple Regressions

The method of analysis that we have most extensively developed combines the results of separate multiple regression analyses performed on individual patients. Recall that individual patient records differ in quantity of data and greatly vary on covariates. By analyzing each patient's record separately, we can determine the distribution of an effect across patients and obtain information as to why some patients exhibit an effect and others do not.

Naturally, we are interested in knowing whether a given causal relationship is statistically significant in the study sample as a whole. The analysis of significance is complicated by the fact that patients have widely varying amounts of data. Intuitively, one would like to weight most heavily those patients in whom a relationship has been most precisely determined, i.e., the patients with the most data; however, these patients may be unrepresentative.

The approach we use is a mixed model. The regression coefficient for each patient is weighted by the inverse of its variance. The mathematical justification for this procedure lies beyond the scope of this paper but may be found in Blum (1982). When there is a large variation in the effect across patients, perfect precision on any one patient is of little advantage, and all patients are weighted nearly equally. When across-patient variation is small, weighting by precision is more appropriate, and the weights diverge.

17.8.12 Interpretation of Results

The final result of the longitudinal design is an estimate of β, the unstandardized regression coefficient of the effect on the cause, and $\text{var}(\beta)$, its variance. The ratio $\beta/[\text{var}(\beta)]^5$ is approximately distributed as a t statistic

TABLE 17-3 Distribution of the Prednisone/ Cholesterol Effect Across Patients (given a baseline value of 230 mg/dl and a change in prednisone from 0 to 30 mg/day)

Range of cholesterol		*Percentage of patients*	*Magnitude of change*
100	150	0	extreme −
150	195	0	strong −
195	210	0	moderate −
210	225	0	weak −
225	230	0	equivocal −
230	235	0	equivocal +
235	250	0	weak +
250	280	10	moderate +
280	360	82	strong +
360	700	8	extreme +

on $n - 1$ degrees of freedom, where n is the number of patients in the study. A two-sided p-value is calculated using the t statistic.

Presently, the interpretation of the results of a study depend only on the magnitude of β and its corresponding p-value. A significant p-value does not necessarily mean the result is medically significant; a p-value can always be made significant if the number of patients is large enough. The program for interpretation uses the following heuristic: if β is large, then for a given p-value, the program assigns a higher validity to the result than it does if β is small.

The clinical significance of β is determined by the magnitude of its expected influence on the effect variable in the study. This is illustrated in Table 17-3, which shows the expected distribution of cholesterol given prednisone at 30 mgms per day.

Recall that the *validity score* is a component of every causal relationship stored in the KB. The validity score is measured on a scale from 1 to 10 summarizing the state of proof of a relationship. The highest score that a study based on a single nonrandomized data base can achieve is 6. Higher scores can only be obtained from replicated studies, the highest scores requiring experimental manipulation and a known mechanism of action. A score of 6 means "strong correlation and time relationship have been demonstrated after known covariates have been controlled in a single data base study."

The discovery module populates the KB with causal links of validity between 1 and 3. The study module overwrites the links that it explores, assigning to those that it confirms scores between 4 and 6.

A statistician or researcher might choose to pursue a given study further, asking "Have the confounding variables in C^* been adequately con-

TABLE 17-4 Effects of Prednisone

	Direction	*Onset delay*	*p-value*
Weight	+	chronic	< .0001
Cholesterol	+	acute	.0001
WBC	+	acute	.0004
% Neutrophils	+	acute	.003
% Lymphs	−	acute	.003
BP-diastolic	+	acute	.004
Glucose	+	acute	.007
Hemoglobin	+	chronic	.009
Wintrobe ESR	−	chronic	.01
Platelets	+	acute	.02
Temperature	−	chronic	.05
anti-DNA	−	chronic	.08
% Eosinophils	−	acute	.15
Urine-RBC's	−	chronic	.17
Creatinine	−	chronic	.19

trolled?" "Are the residuals in each of the regressions independent and identically distributed?" "What accounts for the differences among patients?" A researcher can pursue these questions interactively in RX, incrementally improving the mathematical model (Draper, 1966); however, the automation of this kind of inquiry will require building much greater knowledge into the robot statistician.

17.9 Medical Results

The medical results reported here were generated by running the discovery module and then the study module on a sample data base containing the records of 50 patients with systemic lupus erythematosus (SLE). Many patients had multisystem involvement including glomerulonephritis and nephrotic syndrome.

Table 17-4 shows the effects that were confirmed by the study module for the steroid drug prednisone. The study module automatically incorporated these new links and details of the studies into the knowledge base in the format discussed above.

The effects that were confirmed by the study module for the steroid drug prednisone are shown in Table 17-4. To illustrate the interpretation of Table 17-4, the second row of the table means that prednisone is thought to cause an increase (+) in cholesterol, that the time delay is "acute" (less than one average intervisit interval), and that the effect is highly statistically

significant ($p = .0001$). The study module automatically incorporated these new links and details into the knowledge base in the format discussed above.

Almost all of the acute effects appearing in the table have been extensively confirmed in the medical literature. The effect of prednisone on cholesterol, strongly supported by this study, has only been reported a few times previously. No previous study has recorded the reproducibility of the effect over time or the interpatient variability, as was done here.

The chronic effects of prednisone shown in Table 17-4 are those appearing in a setting of severe SLE. Literature confirmation of these effects has been scant. Because of small numbers of patients, the chronic effects shown here must be further studied. Tables of other empirical results and a discussion of the statistical models used in these studies may be found in Blum (1982).

17.10 Summary

The methods described here emanate from a small set of operational properties of causal relationships. The discovery module uses a nonparametric method for producing a ranked list of causal hypotheses based on strength of time precedence and association. The study module uses a consensual causal model stored in a knowledge base to determine all known confounding variables and to determine appropriate methods of adjusting for them. The statistical model of the tentative causal relationship is then applied to a set of data. If the results indicate that a relationship is significant after controlling for confounding influences, then a new relationship is incorporated into the KB. Subsequent studies may make use of this new link.

All components of the study module can be used in an interactive mode to give a researcher more control in determining the course of the study. For example, the causal model stored in the KB can be queried interactively or changed in the course of a study as new information becomes available. All phases of the statistical analysis can also be interactively modified.

Any methodology that draws causal inferences based on nonrandomized data is subject to an important limitation: *unknown covariates cannot be controlled.* The strength of the knowledge base lies in its comprehensiveness, but even so, it cannot guarantee nonspuriousness. Any single study, particularly one using nonrandomized data, must be viewed skeptically. For this reason, the most conclusive causal relationships that RX discovers are always assigned a modest validity. Only through repeated studies, particularly through experimental manipulation of the causal variable, can a given result become more definitive.

ACKNOWLEDGMENTS

I am grateful to Guy Kraines, Kent Bailey, and Byron William Brown for their assistance with the statistical models; to Gio Wiederhold for project administration and guidance; to Beau Shiel and Ronald Kaplan for their assistance with IDL; and to James Fries, Alison Harlow, and James Standish for assistance in obtaining clinical data.

Funding for this research was provided by the National Center for Health Services Research through grant HS-03650, by the National Library of Medicine through grant LM-03370, and by the Pharmaceutical Manufacturers Association Foundation. Computation facilities were provided by SUMEX-AIM through NIH grant RR-00785 from the Biotechnology Resources Program. Clinical data were obtained from the American Rheumatism Association Medical Information System. The project is continuing under the sponsorship of NCHSR through grant HS-04389.

18

A System for Empirical Experimentation with Expert Knowledge

Peter Politakis and Sholom M. Weiss

When CASNET (Chapter 7) evolved into the general system-building tool known as EXPERT, one of the first applications was a rheumatology consultant program called AI/Rheum (Kingsland and Lindberg, 1983). Developed collaboratively by researchers at Rutgers University and the University of Missouri, AI/Rheum quickly became large and complex, thereby complicating the process of knowledge base maintenance. Peter Politakis, a Rutgers graduate student working with Sholom Weiss and Casimir Kulikowski, accordingly developed a program, named SEEK, that was designed to assist with both expansion and verification of the AI/Rheum knowledge base.

SEEK illustrates how a model of expert reasoning (in this case the rules of rheumatology diagnosis) can be refined with program assistance. The program suggests possible experiments involving generalization or specialization of the preexisting rules in the system. A library of stored patient cases with known conclusions is used as a basis for proposing the experiments. This approach has proven particularly valuable in assisting the expert in a domain like rheumatology where two diagnoses are often difficult to distinguish.

The research on SEEK also has its origins in the knowledge-acquisition tool TEIRESIAS, developed by Davis for MYCIN (Davis, 1979). However, SEEK is able to go a step further by using a somewhat more articulated representation than MYCIN's rules. In AI/Rheum evidence is classified according to major and minor findings, plus required and

From *Proceedings of the Fifteenth Hawaii International Conference on Systems Science*, 2: 649–657 (1982).

excluded findings. Specialization and generalization are accomplished by adding or deleting elements in these lists. The use of symbolic categories of belief (definite, probable, and possible) provides a specifiable means for manipulating the rules.

While based on a simple idea, the SEEK program convincingly demonstrates the value of a richly structured representation and of reasoning from cases as a way of constructing a model. That is, expert knowledge is inseparable from case experience (Schank, 1983), in so far as knowledge explains the cases. The use of a knowledge base to provide an explanatory model has characterized other recent AIM work as well (cf. the diagnostic approach used by Patil, Chapter 14). Another important strength of the SEEK approach is its exhaustive analysis of the entire library of cases, thereby revealing the overall effect of a modification. Experts building the system can accordingly avoid being swayed by one or two cases; they must explain their experiences as a whole.

18.1 Introduction

Over the past decade, much of the research in the development of expert systems has been focused on the acquisition of knowledge in various medical areas: CASNET (Chapter 7)—ophthalmology; INTERNIST (Chapter 8), PIP (Chapter 6)—internal medicine; and MYCIN (Chapter 5)—infectious diseases. A relatively difficult task is to find effective methods for validating a system's knowledge base and evaluating its performance. A step in this direction has been taken in recent work to develop knowledge-engineering tools that would facilitate the building and testing of an expert system. Two examples of generalized knowledge-engineering tools are the EXPERT (Weiss and Kulikowski, 1979) and EMYCIN (van Melle, 1979) systems. These systems provide the builder of an expert system with a prespecified control strategy, a production rule formalism for encoding expert knowledge, explanatory tools for tracing the execution of rules during a consultation session, and a data base system in which cases can be stored for empirical testing. Other work on empirical testing of expert systems has been reported in the development of the PROSPECTOR consultation model for mineral exploration (Gaschnig, 1979). The PROSPECTOR scheme uses sensitivity analysis to determine the effect on the model's conclusions as a result of making changes to certainties in the input data. The empirical testing is based on matching the expert's conclusion to the overall result and also to the intermediate conclusions reached by the model.

As has been demonstrated in the TEIRESIAS system (Davis, 1977), the knowledge-engineering tools that explain a system's decisions are invaluable aids in expert knowledge acquisition and in improving performance.

During a consultation session on a patient case, TEIRESIAS assists the user in composing new rules to correct erroneous conclusions. TEIRESIAS generates its advice about the contents of a new rule by using a *rule model* that summarizes relationships within a subset of the rules in the knowledge base. It does not, however, directly determine the impact of changes to the knowledge base on other cases previously processed by the consultation program.

The approach described in this paper is to integrate performance information into the design of an expert model to automatically provide advice about rule refinement. A system called SEEK has been developed that generates advice in the form of suggestions for possible experiments in generalizing or specializing rules in an expert model. Case experience, in the form of stored cases with known conclusions, is used to interactively guide the expert in refining the rules of a model. In particular, SEEK looks for certain regularities about the performance of the rules in misdiagnosed cases as a basis for suggesting changes to the rules. An expanded description of methods and the uses of SEEK can be found in Politakis (1982).

18.2 The Model

A table of criteria, which is a specialized type of frame or prototype (Aikins, 1979), is prepared for each potential diagnosis. The table consists of two parts:

- major and minor observations that are significant for reaching the diagnosis
- a set of diagnostic rules for reaching the diagnosis

The following example shows observations, grouped under the headings *Major criteria* and *Minor criteria,* for mixed connective tissue disease:

Major criteria	*Minor criteria*
1. Swollen hands	1. Myositis, mild
2. Sclerodactyly	2. Anemia
3. Raynaud's phenomenon or esophageal hypomotility	3. Pericarditis
4. Myositis, severe	4. Arthritis $\leq$ 6 wks
5. CO diff. capacity (normally $<$ 70)	5. Pleuritis
	6. Alopecia

The second part of the table contains the diagnostic rules. In the following example, each column consists of a rule for a specific degree of certainty in the diagnosis:

	Definite	*Probable*	*Possible*
	4 majors	2 majors, 2 minors	3 majors
Requirements	Positive RNP antibody	Positive RNP antibody	No requirement
Exclusions	Positive SM antibody	No exclusion	No exclusion

There are three levels of confidence: definite, probable, and possible. A diagnostic rule is a conjunction of three components, one taken from each row: specific numbers of major or minor observations, requirements, and exclusions. *Requirements* are those combinations of observations that are necessary beyond simple numbers of major and minor findings (although major and minor findings also may be requirements). *Exclusions* are those observations that rule out the diagnosis at the indicated confidence level. The three fixed confidence levels are an important attribute of the model. They substitute for complex scoring functions, which can be a major difficulty in analyzing and explaining model performance (see Chapter 9). It is understood that if a definite diagnosis for a particular disease is made, then even if the rules for the probable or possible diagnosis for the same disease are satisfied, the definite conclusion is appropriate.

As an example, the rule for concluding definite mixed connective tissue disease can be stated as follows: if the patient has 4 or more major observations for mixed connective tissue disease, and RNP antibody is positive, and SM antibody is not positive, then conclude definite mixed connective tissue disease. In most applications, multiple rules are described for each confidence level.

In terms of refinement of a model, the following sections will focus on tools that facilitate identifying two classes of changes that can be made to the rules—generalizations and specializations. *Generalizations* are changes to a rule R that result in a different rule Rg where Rg logically includes R. For example, this can be accomplished by dropping a requirement or decreasing the number of major and minor findings for a rule. *Specializations* are changes to a rule R that result in a different rule Rs where Rs is logically included by R. For example, this can be accomplished by increasing the number of major and minor findings in a rule.

Framelike schemes have been used to represent medical knowledge in the PIP (see Chapter 6) and CENTAUR (Aikins, 1979) systems, which were designed to provide diagnostic consultations in subspecialties of medicine. In addition to representing various clinical states, findings with typical values and frequencies, and related diseases in each disease frame, there were slots containing relatively complex scoring functions that could be specialized for the evaluation of the disease frame. The tabular model is a simple type of frame representation requiring for each diagnostic con-

clusion fixed types (e.g., majors, exclusions) of observations that are relatively easy to understand. Also, scoring follows directly from the three confidence levels of definite, probable, and possible.

18.3 The Rheumatology Application

In collaboration with rheumatologists at the University of Missouri, a consultation model for connective tissue diseases has been realized using the EXPERT system (Weiss and Kulikowski, 1979) for developing consultation models. This subpart of rheumatology is a particularly difficult area for the physician and includes seven diseases: rheumatoid arthritis, systemic lupus erythematosus (SLE), progressive systemic sclerosis, mixed connective tissue disease, polymyositis, primary Raynaud's syndrome, and Sjogren's disease. Some of the difficulties in the differential diagnosis of these diseases may be appreciated by noting that even the experts in this area disagree about some of the diagnoses, that the disease process evolves in atypical ways within patients, and that there is a general lack of pathognomonic criteria to confirm diagnoses objectively (Lindberg et al., 1980).

In terms of building the model in this area, a key aspect throughout its development has been testing the model against a data base of clinical cases that includes the correct diagnosis for each case; a correct diagnosis was decided by an agreement of at least two out of three rheumatologists. After an initial design consisting of 18 observations and 35 rules, the model has undergone many cycles of testing and revision. This incremental process resulted in the expansion of the model to include 150 observations, of which several observations were combined by rules to reach intermediate conclusions, and a total of 147 rules. The model has been critiqued by an external panel of expert rheumatologists, and a review of performance has shown the model to achieve diagnostic accuracy in 94% of 145 clinical cases (Lindberg et al., 1980). Current efforts include expanding the model to cover other rheumatic diseases and to provide advice about treatment management.

18.4 Stages of Model Development

The use of SEEK assumes the specification of a tabular model for each final diagnosis and the entry of cases, including the correct final diagnosis assigned to each case. The stages of model development that will be discussed are listed below.

Stages in the Design of an Expert Model

- Initial design of the model
- Data entry: cases with correct conclusions
- Performance summary of the model
- Analysis of the model
- Generation of model refinement experiments
- Refinement of the model
- Impact of model changes on the data

18.4.1 Initial Design of the Model

A text editor is used to specify an initial design of the model. Any one of three editing modes can be specified by the model designer: table input, table update, or table review and store. For each newly identified final diagnosis, table input mode allows the model designer to list major and minor observations and to specify components of the rules that would conclude the diagnosis. In table update mode, the table for a specified final diagnosis is retrieved, and the model designer can revise the rules or the lists of major and minor observations. When the additions and updates are completed, the table is stored and translated into a format used by SEEK. The translation of the table is to the EXPERT format (Weiss and Kulikowski, 1979) so that a consultation session (to be described in the next section) looks the same as one in EXPERT.

18.4.2 Entry of Data in a Consultation Session

A questionnaire is used to enter the observations, including the correct final diagnosis for a case. Editing facilities are available to review and to change the responses to questions. A case is stored in a data base that is maintained by the system. Figure 18-1 shows the entry of data for a particular case. After all questions have been asked, the system provides a summary of the data in the case. From this, the expert can correct any data entry errors, and, later, the case can be stored in a data base. Cases are usually entered in large groups during a single session. Typically, the tedious cycle that is repeated for each case consists of data entry, fixing errors, and saving the case. However, the expert can request the model's diagnosis for any case and at any time during this stage. An example (continuing with the case entered above) of the interpretative analysis output provided is shown in Figure 18-2. This includes the differential diagnosis (i.e., definite rheumatoid arthritis and possible SLE) followed by detailed lists of findings that provide a more complete picture of the case. These lists are

```
CASE TYPE:
                (1) Case Entry      (2) Visit Entry      (3) Case Review
                (4) Case Deletion   (5) Demo Entry       (6) Program Exit: 1

Enter Name or ID Number: test

Case Type: (1)Real (2)Hypothetical *2
Enter Date of Visit: 6/22/81

Enter Initial Findings (Press RETURN to begin questioning):
*

 1. Extremity Findings:
    1) Arthralgia
    2) Arthritis ≤6 wks. or non-polyarticular
    3) Chronic polyarthritis >6 wks.
    4) Erosive arthritis
    5) Deformity: subluxations or contractures
    6) Swollen hands, observed
    7) Raynaud's phenomenon
    8) Polymyalgia syndrome
    9) Synovial fluid inflammatory
    10) Subcutaneous nodules
  Checklist:
  *1,2,3,4,10
    .
    .
    .
31. Presumptive Diagnosis:
    1) Mixed Connective Tissue Disease
    2) Rheumatoid Arthritis
    3) Systemic Lupus Erythematosus
    4) Progressive Systemic Sclerosis
    5) Polymyositis
    6) Primary Raynaud's
    7) Sjogren's
  Checklist:
  *2
```

FIGURE 18-1 The entry of data for a case.

obtained by matching findings from the case data to prespecified lists that are associated with each final diagnosis in the model; the lists include those findings consistent, not expected, and unknown for the diagnosis.

18.4.3 Model Performance

A typical mode of interaction with SEEK involves iterating through these steps:

- obtain performance of rules on the stored cases
- analyze the rules
- revise the rules

INTERPRETATIVE ANALYSIS
Diagnoses are considered in the categories definite, probable, and possible.

Based on the information provided, the differential diagnosis is

Rheumatoid arthritis (RA)	—Definite
Systemic lupus erythematosus (SLE)	—Possible

Patient findings consistent with RA:
- Chronic polyarthritis >6 wks.
- RA factor (l.f.), titer <1:320
- Subcutaneous nodules
- Erosive arthritis

Patient findings not expected with RA:
- Oral/nasal mucosal ulcers

Patient findings consistent with SLE:
- Platelet count, /cmm: ≤99999
- Oral/nasal mucosal ulcers
- Arthritis ≤6 wks, or non-polyarticular

Patient findings not expected with SLE:
- Erosive arthritis

Unknown findings which would support the diagnosis of SLE:
- LE cells
- DNA antibody (hem.)
- DNA antibody (CIEP)
- DNA (hem.), titer 1:
- FANA
- Sm antibody (imm.)

End of diagnostic consultation: 22-Jun-81.

FIGURE 18-2 The interpretative analysis for the case in Figure 18-1.

In reviewing the performance of a model, the expert's conclusions are matched to the model's conclusions. The expert's conclusion is stored with each case, while the model's conclusion is taken as that conclusion reached with the greatest certainty.

Conditions for Performance Evaluation

The first step is to produce performance results on all stored cases. As mentioned earlier, evaluating performance involves matching the expert's conclusion to the model's conclusion in each case. A practical problem for scoring the results in a particular case occurs when ties in certainty between the expert's conclusion and the model's different conclusion are noted. Whether the model is scored as correct or incorrect for such a case affects the direction of subsequent rule refinements. A decision on how ties should be treated in performance evaluation rests with the problem domain. Whereas ties may be acceptable in particular medical areas for which it is

Current Performance			
			False positives
Mixed connective tissue disease	9/33	(27%)	0
Rheumatoid arthritis	42/42	(100%)	9
Systemic lupus erythematosus	12/18	(67%)	4
Progressive systemic sclerosis	22/23	(96%)	5
Polymyositis	4/5	(80%)	1
Total	89/121	(74%)	

FIGURE 18-3 Summary of the model's performance.

difficult to discriminate between competing diagnoses, they probably would not be acceptable in areas for which the diagnostic choices are well understood and mutually exclusive. Rheumatology is an area that exemplifies the former condition. For instance, particular rheumatic diseases do coexist during the progression of the respective disease processes, and therefore a final diagnosis is difficult to make. In such cases, a tentative diagnosis may be made that does not rule out other related diseases. An interpretation of a model's conclusions could reflect this situation by treating ties in certainty as correct (e.g., ties in certainty at the possible or probable confidence level). There may be exceptions. For example, ties at the definite level and at the null level (i.e., no conclusion was reached by the model) may be considered incorrect for diagnostically related diseases. The point of this discussion is to motivate the need for specifying a condition under which performance evaluation is to be performed. SEEK allows the model designer to specify how ties in confidence are to be treated.

Another condition is to allow the model designer to determine which rules and cases are to be ignored during the evaluation process. This has been found useful when either there are insufficient numbers of cases for a particular final diagnosis or the rules are not deemed to be in a satisfactory state by the model designer. If not turned off, these rules usually interfere in several case diagnoses, and their performance over all cases is therefore quite low. SEEK allows the model designer to specify rules to be turned off for performance evaluation.

Performance Summary of the Model

The results are organized according to final conclusions and show the number of cases in which the model's conclusion matches the expert's conclusion. The column labeled *False positives* shows the number of cases in which the indicated conclusion was reached by the model, but did not match the stored expert's conclusion. In Figure 18-3, the summary of performance for mixed connective tissue disease indicates that 9 cases out of

Rule 72:	2 or more Majors for RA (MJRA) 2 or more Minors for RA (MNRA) No Exclusion for RA (EX102) → Probable Rheumatoid arthritis (RA)
43 Cases:	in which this rule was satisfied.
13 Cases:	in which the greatest certainty in a conclusion was obtained by this rule and it matched the expert's conclusion.
7 Cases:	in which the greatest certainty in a conclusion was obtained by this rule and it did not match the expert's conclusion.

FIGURE 18-4 Summary of a specific rule's performance.

33 were correctly diagnosed. Furthermore, there are no cases that were misdiagnosed by the model as mixed connective tissue disease. The rules that conclude rheumatoid arthritis perform quite well for the stored rheumatoid arthritis cases, but they also appear to be candidates for specialization because of the 9 false positives.

In addition to the results shown in Figure 18-3, performance results about a specific rule can be obtained that show the number of cases in which the rule was satisfied. An example of this is shown in Figure 18-4, and includes the number of cases in which the rule was used successfully (i.e., matching the expert's conclusions stored with the cases) and the number of cases in which the rule was used incorrectly (i.e., not matching the expert's conclusions stored with the cases).

18.4.4 Analysis of the Model

Interactive assistance for rule refinement is provided during the analysis of the model. The model designer has the option of selecting either "single case" or "all cases" as a basis of analysis.

Analysis of the Model in a Single Case

Analysis in a single case proceeds after a case has been chosen from the data base of stored cases. The objective of single case analysis is to provide the model designer with an explanation of the model's results in the case. This is done by first showing the model's confidence in both the expert's conclusion and the model's conclusion. Rules are cited that were used to reach these conclusions. Rules for the expert's conclusion are selected from those rules in the model with the same conclusion as the conclusion stored (by the expert) for a case. If the model's conclusion does not match the expert's conclusion in the case, the system attempts to locate a partially

CASE: 3

Expert conclusion: Progressive systemic sclerosis
Model conclusion: Probable Rheumatoid arthritis

This is the strongest satisfied rule for the expert's conclusion:

Rule 111: 1 or more Majors for PSS (MJPSS) (1 Majors Satisfied)
1 or more Minors for PSS (MNPSS) (3 Minors Satisfied)
→ Possible Progressive systemic sclerosis (PSS)

This is the rule for the model's conclusion:

Rule 72: 2 or more Majors for RA (MJRA) (2 Majors Satisfied)
2 or more Minors for RA (MNRA) (3 Minors Satisfied)
No Exclusion for RA (EX102) (Satisfied)
→ Probable Rheumatoid arthritis (RA)

There exists 1 partially satisfied rule for PSS with weight assignment ≥ that set by RA rule

Rule 112: Requirement 1 for probable PSS (RR105) (Not set)
No Exclusion for probable PSS (ER105) (Satisfied)
→ Probable Progressive systemic sclerosis (PSS)

FIGURE 18-5 Results of a case analysis.

satisfied rule for the expert's conclusion that is the "closest" to being satisfied and would override the model's incorrect conclusion. A procedure for finding the "closest" rule is described later. An example of the results of single case analysis is shown in Figure 18-5. Case 3 is misdiagnosed by the model, which has assigned the certainty value of "possible" to progressive systemic sclerosis. The model's conclusion is rheumatoid arthritis with a certainty value of "probable." Rule 111 and Rule 72 are responsible for reaching these conclusions. Each line printed for a rule contains an internal label for reference purposes, such as MJPSS. In this example, Rule 72 was triggered because two majors and three minors for rheumatoid arthritis are present, and Case 3 did not have the (exclusion) findings that would deny Rule 72. Given this information the model designer can pursue either of two directions to refine the rules: to weaken Rule 72 so that it will not override Rule 111, or to find a stronger rule concluding progressive systemic sclerosis. In response to this latter possibility, SEEK cites Rule 112 as a likely candidate to generalize. A procedure that SEEK uses to identify rules such as Rule 112 is described in the next section.

Besides this information provided in single case analysis, SEEK allows the model designer to interrogate any conclusion in the model, both final and intermediate results. The rules for any conclusion can be cited by specifying a rule number or the internal label tagged to a conclusion (e.g., PSS). In the latter situation, all rules for a conclusion are cited, both totally satisfied and partially satisfied rules in the case. This aids the model designer in reviewing the performance of a subset of the rules on the case data.

Analysis of the Model Based on Case Experience

The first step for the analysis of the model for all cases is to specify a final diagnosis for which rules are to be analyzed. In this manner, the model designer focuses the analysis on the subset of the rules in the model. The analysis is usually done after performance results have been obtained. SEEK assists the model designer in the analysis of a subset of the rules that are relevant to the misdiagnosed cases. An important design consideration for SEEK is to provide the model designer with a flexible means to perform experiments in refining the rules. In this section, advice will be described that helps in determining the specific experiments for rule refinement. Heuristic procedures are needed to select experiments from the many possibilities. For example, SEEK uses a heuristic procedure by tracing rules that conclude the stored expert's conclusion to determine which rules are "closest" to being satisfied. It looks for a partially satisfied rule for which the following conditions hold:

1. the rule concludes at a minimum confidence level that is greater than (or equal to, depending on the treatment of ties) the certainty value for the model's conclusion;
2. the rule contains the maximum number of satisfied components for all rules concluding at that confidence level.

A rule satisfying these conditions is marked for generalization, so that it may be invoked more frequently. The rule used to reach the model's conclusion is marked for specialization, so that it may be invoked less frequently.

In the following example, SEEK analyzes the rules for the specified diagnosis, mixed connective tissue disease, with regard to their use on the stored cases. After analysis, SEEK reports the results by numbering and listing rules that conclude mixed connective tissue disease, for which there exists information to indicate that the rule is a potential candidate for generalization or specialization. Figure 18-6 is a summary of this rule analysis and shows unsatisfied rules in the misdiagnosed cases for mixed connective tissue disease that are candidate rules for generalization. The column labeled *Generalization* contains the number of cases suggesting the generalization of a rule, and the column labeled *Specialization* contains the number of cases suggesting the specialization of a rule.

In Figure 18-6, rules at the possible level of certainty are strong candidates for generalization. Although Rule 56 is not satisfied in eight misdiagnosed cases, if Rule 56 had been satisfied, these eight cases would have been correctly diagnosed. In the eight cases cited for Rule 56, Rule 56 is "closer" to being satisfied than Rule 55 is. A more detailed analysis of each rule, summarizing the satisfied and unsatisfied components of the rule, is normally obtained at this point. Rule 55 can be stated as follows: if the

Mixed Connective Tissue Disease

Rule	*Certainty*	*Generalization*	*Specialization*
54.	Possible	2	0
55.	Possible	7	0
56.	Possible	8	0
57.	Probable	2	0
58.	Probable	2	0

FIGURE 18-6 Summary of rule analysis for the diagnosis of mixed connective tissue disease.

patient has two or more major observations for mixed connective tissue disease and RNP antibody is positive, then conclude possible mixed connective tissue disease. Rule 56 can be stated as follows: if the patient has three or more major observations for mixed connective tissue disease, then conclude possible mixed connective tissue disease. A simple experiment for generalization of Rule 56, which might be tried first because it is the simpler rule, is to decrease the number of major observations required.

The scheme for analysis in all cases focuses on a subset of the rules by gathering empirical information suggesting the generalization and specialization of rules in the set. This can be viewed as a learning system. In Mitchell's version space approach (Mitchell, 1979), two sets of rules are maintained as bounds on the "maximally specialized" rules and the "maximally generalized" rules that are consistent with the training cases presented for a conclusion. A training case is prespecified as either *positive*—a rule must be found to cover the case—or *negative*—no rule should match the case. The scheme seeks to cover all positive cases while allowing no negative cases to match any of the rules. There are no certainty values assigned to the rules in the version space. Our scheme seeks to refine expert-derived rules that have been categorized by confidence levels in the model. Correct classification for all cases is not required. That is, a negative case is allowed to be covered so long as there is a rule for another conclusion that overrides the matched rule(s). A rule is marked for generalization or specialization based on the comparison of the certainty values assigned to the final conclusion expected to that reached by the model. Finally, our scheme is interactive in nature, requiring the involvement of the model designer. It is not intended to be an autonomous learning system.

18.4.5 Generation of Model Refinement Experiments

As was shown in Figure 18-6, SEEK indicated several mixed connective tissue disease rules that are candidates for generalization. In general, there are many possibilities that can be tried for refining the rules in a model. A difficult task is to select a rule or group of rules to work on and then to

24 cases in which the expert's conclusion MCTD does not match the model's conclusion:

1, 4, 11, 12, 14, 15, 42, 47, 49, 57, 60, 67, 71, 75, 78, 80, 84, 93, 99, 100, 104, 105, 107, 130

Proposed Experiments for Mixed Connective Tissue Disease

1. Decrease the number of majors in rule 56.
2. Delete the requirement component in rule 55.
3. Delete the requirement component in rule 54.
4. Decrease the number of minors in rule 57.
5. Delete the requirement component in rule 58.

FIGURE 18-7 List of misdiagnosed cases of mixed connective tissue disease and proposed experiments for improving the rules.

determine plausible refinements beyond classifying a rule as a candidate for generalization or specialization. In this section, an approach to suggest automatically plausible experiments for refining the rules in a model is described.

A heuristic rule-based scheme is used to suggest experiments. The heuristic rules are called EX-rules so as not to confuse them with the expert-modeled rules. The IF part of an EX-rule contains a conjunction of predicate clauses that essentially looks for certain features about the performance of rules in the model, while the THEN part of an EX-rule contains a specific rule refinement experiment. An example of an EX-rule is shown below and is used to suggest the specific generalization experiment to decrease the number of major findings in a rule. Currently, there are eleven EX-rules, which are divided almost equally with respect to the types of experiments (i.e., generalizations or specializations) that may be suggested.

IF: the number of cases suggesting generalization of the rule is greater than the number of cases suggesting specialization of the rule and the most frequent missing component in the rule is the major component,
THEN: decrease the number of major findings in the rule.

Evaluation of an EX-rule begins by instantiating the clauses with the required empirical information about a specific rule in the model. Function calls are used to gather the information. After instantiation, the clauses are evaluated in order beginning with the first clause in the EX-rule. If all clauses are satisfied, then the specific experiment is posted. All EX-rules are evaluated in this manner for a specific rule in the model. The experiments suggested by the EX-rules are narrowed by the expert to those changes consistent with his or her medical knowledge. In Figure 18-7, the experiments for improving the rules used in reaching the diagnosis of mixed connective tissue disease are presented after listing the misdiagnosed cases of mixed connective tissue disease.

The experiments are ordered based on maximum potential performance gain on the cases. Other criteria for ordering can be used such as

```
:why(1)

If rule 56 had been satisfied, 8 currently misdiagnosed MCTD cases would have been diagnosed correctly.
Currently, rule 56 is not used incorrectly in any of the cases. In rule 56 the component missing with the
greatest frequency is Major.

Therefore, we suggest decreasing the number of majors in rule 56.
This would generalize the rule so that it will be easier to satisfy.
```

FIGURE 18-8 Explanation of a proposed experiment.

ease of change (e.g., an experiment that suggests changing the minors in a rule may be preferred over an experiment that suggests changing the majors). An explanation of a particular experiment is provided by a translation of the specific EX-rule used to suggest the experiment into a narrative statement containing the empirical information about the rule. As an example, the support for the first experiment is shown in Figure 18-8. It should be emphasized that a decision as to which experiments, if any, are to be tried is left to the model designer. Even though a particular experiment is supported empirically, the ultimate decision should include justifying an experiment in terms of other knowledge about the domain. For example, is a rule resulting from the first experiment for Rule 56 "medically sound" to make the diagnosis? This can lead to reconsidering the lists of major and minor findings for a particular final diagnosis and to potentially refining these findings.

It should be noted that one is not absolutely certain of a net gain in performance before an experiment is tried. In the case of a generalization experiment, there may be more than one unsatisfied component in a rule marked for generalization; the marking procedure picks the first unsatisfied component in the rule. Facilities for performing experiments and for determining the impact of changes on the cases are described later.

18.4.6 Refinement of the Model

After an experiment to revise the rules has been determined, the model designer can test his or her proposed revision on the cases. This is facilitated by editing capabilities that permit the model designer to interrogate and to modify the rules in the model. The changes are logged separately from the rules in the model so that the original rules can be restored. The editing functions include changing:

- the number of major or minor observations,
- the requirement component,
- the exclusion component, and
- any rule reaching an intermediate result that is used by other rules.

Candidate for Change is MJMCT in rule 56

Rule 56 is:

3 or more Majors for MCTD (MJMCT)
→ Possible Mixed connective tissue disease (MCTD)

Generalization of Rule 56 is:

2 or more Majors for MCTD (MJMCT)
→ Possible Mixed connective tissue disease (MCTD)

FIGURE 18-9 SEEK's description of the proposed rule change.

Continuing with our example, Figure 18-9 shows the response by SEEK for the model designer's suggested change to Rule 56: to change the number of majors required by Rule 56 to be 2 or more majors. The commands that allow the model designer to interrogate and to modify the rules require rule numbers or symbolic labels to reference parts of the model.

18.4.7 Impact of Model Changes on the Data

The results of a specific experiment are obtained by conditionally incorporating the revised rule(s) into the model. The updated model is then executed on the data base of cases. The results are summarized in Figure 18-10 for making the change to Rule 56. In this example, such a modification significantly improves performance. Several misdiagnosed cases of mixed connective tissue disease are now correctly diagnosed by the model.

	Before	*False positives*	*After*	*False positives*
MCTD	9/33 (27%)	0	17/33 (52%)	0
Others	80/88 (91%)	(see below)	80/88 (91%)	(see below)
Total	89/121 (74%)		97/121 (80%)	
	Details of Effect on Other Diseases			
RA	42/42 (100%)	9	42/42 (100%)	8
SLE	12/18 (67%)	4	12/18 (67%)	3
PSS	22/23 (96%)	5	22/23 (96%)	3
PM	4/5 (80%)	1	4/5 (80%)	1

FIGURE 18-10 Results of executing updated model on the data base of cases.

Moreover, there was no adverse side effect of this change on other cases with different stored conclusions. The model designer has the option either to accept or to reject the experiment. If a simple modification does not lead to desirable results, more complicated changes may be tried, such as multiple modifications or dropping a condition in a requirement.

18.5 Discussion

The tabular model appears to be a reasonable framework for encoding expert knowledge in a real and complex application. Excellent performance was achieved for the diagnosis of mixed connective tissue disease (Lindberg et al., 1980). This approach has proven particularly valuable in assisting the expert in domains where two diagnoses are difficult to distinguish. For example, there is a general lack of deterministic clinical criteria to confirm the diagnoses in the connective tissue disease area. The experts obtain by means of empirical testing a measure of the usefulness of the observations expressed in the tabular model. There are limitations to this approach—for some applications it may be difficult to express rules using major and minor observations or using only three levels of confidence. Although this model may not be the most expressive model for capturing expert knowledge, it is a model that is suitable for an empirical analysis leading to experimentation with rule refinement. Samples of cases are not completely representative and cannot begin to match the scope of the expert's knowledge. But as others have found (Gaschnig, 1979), even with small samples of cases, empirical evidence can be of great value in designing and verifying an expert model.

Ideally, a tabular model abstracts the expert's reasoning in diagnostic criteria, while cases cite evidence that is accurately diagnosed by the model. The use of SEEK attempts to achieve this harmony by pointing out potential problems with these dual sources of knowledge. Given the performance of the cases, potential problems with the rules can be identified with the tools described earlier. The summarized performance results are a means for the expert to rethink a tabular model that is performing poorly for a specific diagnosis. The analysis of the tabular rules based on case experience sharply focuses the expert's attention on modifications that potentially result in improved performance and that are medically sound. This can lead to reviewing individual cases for inaccuracies in the data and to reconsidering the importance of specific criteria in the model. It should be emphasized that this process is not intended to "custom-craft" rules solely to the cases, but rather to provide the expert an interactive environment with explicit performance information that needs to be accurately explained. From an artificial intelligence perspective, this may be viewed as a learning process based on experience in developing the model. From the

empirical testing and successive improvements in the performance of the model, the human expert will obtain not only a better formulation of the model but also a better understanding of the explicit diagnostic criteria used in his or her reasoning.

ACKNOWLEDGMENTS

This research was supported in part by grant RR-643 of the National Institutes of Health Biotechnology Resources Program. We gratefully acknowledge the contributions of our collaborators in developing a rheumatology consultation system: G. Sharp, M.D., and D. Lindberg, M.D., and their fellow researchers at the University of Missouri at Columbia. We are also grateful to Casimir Kulikowski for his review of this work.

19

PUFF: An Expert System for Interpretation of Pulmonary Function Data

Janice S. Aikins, John C. Kunz, Edward H. Shortliffe, and Robert J. Fallat

In this and the next chapter we close this volume with discussions of the two AIM systems that had achieved routine use by the end of the first decade of research in the field. It is important to note that neither requires direct interaction with a physician requesting advice. Thus both systems avoid the significant problems of human engineering and user acceptance that define many of the serious research problems that remain unsolved at present [see Teach and Shortliffe (1981) and a further discussion of these points in Chapter 21]. However, each does provide a glimpse of what lies ahead, and their success at difficult tasks is an encouraging indication of the practical impact that we can eventually expect from this kind of work.

Because MYCIN was designed to keep its knowledge base of rules separate from the program that used them to generate advice (Chapter 5), it was recognized that the program itself could be isolated and used in other domains for which additional rule sets were developed. The resulting EMYCIN system (van Melle, 1980) was used to build several other programs during the late 1970s, in both medical and nonmedical domains [e.g., SACON, a program to provide guidance regarding the use of a computer system to aid in aircraft design (Bennett and Englemore, 1979)]. An early system developed using EMYCIN was PUFF, a collaborative effort between computer scientists from Stanford University, researchers from the Institute of Medical Sciences in San Francisco, and physicians from Pacific Medical Center (PMC).

From *Computers in Biomedical Research*, 16: 199–208 (1983).

For several years pulmonary physiologists at PMC had been toying with ideas for the development of a program to interpret pulmonary function test (PFT) results. They had found it difficult to develop a straightforward algorithm for defining the criteria for test interpretation, however, and as a result were continuing to interpret PFT results by hand when the collaboration with Stanford developed. Working in the EMYCIN environment, they were delighted to find that they could more easily distill their criteria for test interpretation by using the production rule formalism. Within a few months a small experimental system was developed and was shown to perform extremely well for analyzing a subset of PFT abnormalities. Thereafter the rule set was expanded, and, when it had stabilized, the clinicians were eager to implement the system for use at PMC. It had been developed at Stanford on the SUMEX-AIM computer, however, and this was an unrealistic vehicle for providing service computing at a hospital in San Francisco. As is described in this chapter, the PUFF rule set was therefore rewritten into a program using the BASIC language and implemented to run on a minicomputer at PMC. It accordingly became a working tool in the pulmonary physiology lab of this large institution. Its performance and the results of a formal evaluation experiment are described here. In addition, Janice Aikins and her coauthors examine some of the elements of the problem that paved the way for its success and also consider the significant limitations of the solution that warrant further study.

19.1 Introduction

Researchers in the field of artificial intelligence are just beginning to produce systems that capture the specialized knowledge of experts and that use this knowledge to perform difficult tasks. Although the technology is still rather new, a small set of programs now exist as "tools" useful for building these so-called expert systems. This paper describes an expert system, called PUFF, that was built using EMYCIN, a generalization of an earlier medical system named MYCIN. The task chosen for PUFF is described briefly, and the rationale for the appropriateness of this choice is presented. PUFF was initially developed on the SUMEX computer, a large research machine at Stanford University, and was later rewritten in a production version to run on the hospital's own minicomputer. We describe here the history of the PUFF project and its current status, including observations about its limitations and successes. We also take a brief look at the knowledge representation and control structure used for the SUMEX version of the system. Finally, the results of a formal evaluation of the production version of PUFF are presented.

19.2 Task

PUFF interprets measurements from respiratory tests administered to patients in the pulmonary (lung) function laboratory at Pacific Medical Center in San Francisco. The laboratory includes equipment designed to measure the volume of the lungs, the ability of the patient to move air into and out of the lungs, and the ability of the lungs to get oxygen into the blood and carbon dioxide out.[1] The pulmonary physiologist interprets these measurements in order to determine the presence and severity of lung disease in the patient. An example of such measurements and an interpretation statement are shown in Figure 19-1. The test measurements listed in the top half of the figure are collected by the laboratory equipment. The pulmonary physiologist then dictates the interpretation statements to be included in a typewritten report. All of the measurements are given as a percentage of the predicted values for a normal patient of the same sex, height, and weight. The interpretation and final diagnosis are a summary of the reasoning about the combinations of measurements obtained in the lung tests.

19.3 Rationale

PUFF's task is to interpret such a set of pulmonary function (PF) test results, and to produce a set of interpretation statements and a diagnosis for the patient. The problem of developing an automated pulmonary function interpretation system was chosen for several reasons:

1. The interpretation of pulmonary function tests is a problem that occurs daily in most hospitals, so a computer program that captures the expertise involved in interpreting these tests, and that can assist in providing interpretations, fills a practical need.
2. The biomedical researchers at Pacific Medical Center (PMC) were interested in the problem and were eager to work with us on developing a solution. It was possible that such a system could enhance the effectiveness of patient care and the pulmonary physician's efficiency. In addition, solution of this relatively simple interpretation problem could identify possibilities for further research into more difficult interpretation tasks.

[1]Measurements include spirometry and, optionally, body plethosmography, single breath CO diffusion capacity, and arterial blood gases. Measurements can be made at rest, following inhalation of a bronchodilator, and during exercise.

PRESBYTERIAN HOSPITAL OF PMC
CLAY AND BUCHANAN, BOX 7999
SAN FRANCISCO, CA. 94120
PULMONARY FUNCTION LAB

WT 40.8 KG, HT 161 CM, AGE 65 SEX F
REFERRAL DX-

			PREDICTED (+/-SD)	OBSER(%PRED)	TEST DATE 5-13-76 POST DILATION OBSER(%PRED)
INSPIR VITAL CAP	(IVC)	L	2.7(0.6)	2.3 (83)	
RESIDUAL VOL	(RV)	L	2.0(0.1)	3.8 (193)	3.1 (154)
FUNC RESID CAP	(FRC)	L	2.9(0.3)	4.6 (158)	3.9 (136)
TOTAL LUNG CAP	(TLC)	L	4.7(0.7)	6.1 (129)	5.5 (116)
RV/TLC		%	42.	62.	55.
FORCED EXPIR VOL	(FEV1)	L	2.3(0.5)	1.5 (66)	1.6 (71)
FORCED VITAL CAP	(FVC)	L	2.7(0.6)	2.3 (85)	2.4 (88)
FEV1/FVC		%	82.	64.	66.
FORCE EXP FLOW	200-1200L/S		3.6(0.8)	1.8	1.9
FORCED EXP FLOW	25-75% L/S		2.6(0.5)	0.7	0.7
FORCED INS FLOW	200-1200L/S		2.5(0.5)	2.5	3.4
AIRWAY RESISTANCE (RAW)	(TLC = 6.1)		2.5	1.5	2.2
DF CAP-HGB = 14.5 (DSBCO)	(TLC = 4.8)		23.	17.4 (72)	

INTERPRETATION: THE VITAL CAPACITY IS LOW, THE RESIDUAL VOLUME IS HIGH AS IS THE TOTAL LUNG CAPACITY, INDICATING AIR TRAPPING AND OVERINFLATION. THIS IS CONSISTENT WITH A MODERATELY SEVERE DEGREE OF AIRWAY OBSTRUCTION AS INDICATED BY THE LOW FEV1, LOW PEAK FLOW RATES AND CURVATURE TO THE FLOW VOLUME LOOP. FOLLOWING ISOPROTERANOL AEROSOL THERE IS VIRTUALLY NO CHANGE.

THE DIFFUSING CAPACITY IS LOW INDICATING LOSS OF ALVEOLAR CAPILLARY SURFACE.

CONCLUSIONS: OVERINFLATION, FIXED AIRWAY OBSTRUCTION AND LOW DIFFUSING CAPACITY WOULD ALL INDICATE MODERATELY SEVERE OBSTRUCTION AIRWAY DISEASE OF THE EMPHYSEMATOUS TYPE. ALTHOUGH THERE IS NO RESPONSE TO BRONCHODILATORS ON THIS ONE OCCASION, MORE PROLONGED USE MAY PROVE TO BE MORE HELPFUL.

PULMONARY FUNCTION DIAGNOSIS: OBSTRUCTIVE AIRWAY DISEASE, MODERATELY SEVERE, EMPHYSEMATOUS TYPE.

FIGURE 19-1 Verbatim copy of pulmonary function report. The data were obtained from equipment and the interpretation dictated by an expert physician.

3. PF data interpretation was a problem that the artificial intelligence researchers were particularly interested in solving in order to demonstrate the generality and power of expert system techniques. Putting a system into clinical use would contribute to the credibility of those techniques, and also would show their promise and limitations in clinical practice. Earlier AI programs had demonstrated competence, but their use had required large amounts of professional time simply for data input. PUFF, however, produced PF data interpretations automatically without the necessity for user interaction. Thus we hoped that PUFF would be used by the clinical staff.

4. PF data interpretation was a problem that was large enough to be interesting (the biomedical researchers did not know how to solve it, and the AI researchers did not know whether their techniques would be appropriate) and small enough that a pilot project of several months' duration could concretely demonstrate the feasibility of a longer development effort. Furthermore, the amount of domain-specific knowledge involved in pulmonary function testing is limited enough to make it feasible to acquire, understand, and represent that knowledge.
5. The domain of pulmonary physiology is a circumscribed field: the data needed to interpret patient status are available from the patient's history and from measurements taken in a single laboratory. Other large bodies of knowledge are not required in order to produce accurate diagnoses of pulmonary disease in the patient.[2]
6. All the data used in the laboratory at PMC were already available in a computer; the computer data were known to be accurate, reliable, and relevant to the interpretation task. The clinical staff in the PF lab were already receptive to the use of computers within their clinical routines.
7. Pulmonary physiologists who interpret test measurements tend to phrase their interpretations similarly from one case to the next. One goal of PUFF was to generate reports from a set of prototypical interpretation statements, thus saving the staff a great deal of tedious work. The staff themselves would not be displaced by this tool because their expertise still would be necessary to verify PUFF's output, to handle unexpectedly complex cases, and to correct interpretations that they felt were inaccurate.

19.4 Project History and Status

This research developed from work done on the MYCIN system (Chapter 5). That program used a knowledge base of production rules (Davis and King, 1977) to perform infectious disease consultations. PUFF was initially built using a generalization of the MYCIN system called EMYCIN (van Melle, 1979). EMYCIN, or "Essential MYCIN," consists of the domain-independent features of MYCIN, principally the rule interpreter, explanation, and knowledge-acquisition modules (Shortliffe et al., 1975). It provides a mechanism for representing domain-specific knowledge in the form of production rules, and for performing consultations in that domain. Just as MYCIN consists of EMYCIN plus a set of facts and rules about diagnosis and therapy of infectious diseases, PUFF consists of the EMYCIN programs plus a pulmonary disease knowledge base.

[2]This was a problem in MYCIN, a related system for determining the diagnosis and therapy for infectious disease cases. The results produced by the system often suffered because it lacked knowledge about related diseases that were also present in the patient.

EMYCIN (and hence the EMYCIN version of PUFF) is written in Interlisp (Teitelman, 1978) and runs on a DEC KI-10 at the Stanford SUMEX-AIM computer facility. In order to run PUFF on a PDP-11 at Pacific Medical Center, a second version of the program was created after the EMYCIN version had been refined. This was done by translating the production rules into procedures and writing them in the BASIC language. Conversion to BASIC was an advantage because the PDP-11 was located on the same site as the laboratory, and its schedule could be easily controlled to support production operation by the system users. However, as a result of the conversion, the production and development versions of PUFF became incompatible, and modifications made to one system were sometimes difficult to make in the other.

The PDP-11 version is now routinely used in the pulmonary function laboratory and provides lung test interpretations for about ten patients daily. Since the system became operational in 1979, it has interpreted the results of over 4000 cases. The BASIC code is currently being converted again so that it will run on a personal computer.

The form of the interpretations generated by PUFF is shown in Figure 19-2. This report is for the same patient as in Figure 19-1, seen several years later. As in the typed report, the pulmonary function test data are set forth, followed by the interpretation statements and a pulmonary function diagnosis. The pulmonary physiologist checks the PUFF report, and, if necessary, the interpretation is edited on-line prior to printing the final report for physician signature and entry into the patient record. Approximately 85% of the reports generated are accepted without modifications. The change made to most others simply adds a statement suggesting that the patient's physician compare the interpretation with tests taken during previous visits. For example, statements such as "These test results are consistent with those of previous visits" or "These test results show considerable improvement over those in the previous visit" might be made. PUFF was not designed to represent knowledge about multiple visits, so this kind of statement must be added by the pulmonary physician.

19.5 Observations

PUFF is a practical assistant to the pulmonary physiologist, and thus is a satisfactory and exciting result of the research done with production rule consultation systems. PUFF's performance is good enough that it is used daily in clinical service, and it has the support of both the hospital staff and its administration. However, improvements could be made in the following areas:[3]

[3]Many of these problems are also present in other rule-based systems; they motivated the development of the experimental CENTAUR system (Aikins, 1980; 1983).

PRESBYTERIAN HOSPITAL OF PMC
CLAY AND BUCHANAN, BOX 7999
SAN FRANCISCO, CA. 94120
PULMONARY FUNCTION LAB

WT 40.8 KG, HT 161 CM, AGE 69 SEX F
REFERRAL DX

**********TEST DATE 5-13-80

		PREDICTED (+/-SD)	OBSER(%PRED)	POST DILATION OBSER(%PRED)
INSPIR VITAL CAP	(IVC) L	2.7	2.3 (86)	2.4 (90)
RESIDUAL VOL	(RV) L	2.0	3.8 (188)	3.0 (148)
TOTAL LUNG CAP	(TLC) L	4.7	6.1 (130)	5.4 (115)
RV/TLC	%	43.	62.	56.
FORCED EXPIR VOL	(FEV1) L	2.2	1.5 (68)	1.6 (73)
FORCED VITAL CAP	(FVC) L	2.7	2.3 (86)	2.4 (90)
FEV1/FVC	%	73.	65.	67.
PEAK EXPIR FLOW	(PEF) L/S	7.1	1.8 (25)	1.9 (26)
FORCED EXP FLOW	25-75%L/S	1.8	0.7 (39)	0.7 (39)
AIRWAY RESIST(RAW)	(TLC = 6.1)	0.0(0.0)	1.5	2.2
DF CAP-HGB = 14.5	(TLC = 4.8)	24.	17.4 (72)	(74%IF TLC = 4.7)

****

INTERPRETATION: ELEVATED LUNG VOLUMES INDICATE OVERINFLATION. IN ADDITION, THE RV/TLC RATIO IS INCREASED, SUGGESTING A MODERATELY SEVERE DEGREE OF AIR TRAPPING. THE FORCED VITAL CAPACITY IS NORMAL. THE FEV1/FVC RATIO AND MID-EXPIRATORY FLOW ARE REDUCED AND THE AIRWAY RESISTANCE IS INCREASED, SUGGESTING MODERATELY SEVERE AIRWAY OBSTRUCTION. FOLLOWING BRONCHODILATION, THE EXPIRED FLOWS SHOW MODERATE IMPROVEMENT. HOWEVER, THE RESISTANCE DID NOT IMPROVE. THE LOW DIFFUSING CAPACITY INDICATES A LOSS OF ALVEOLAR CAPILLARY SURFACE, WHICH IS MILD.

CONCLUSIONS: THE LOW DIFFUSING CAPACITY, IN COMBINATION WITH OBSTRUCTION AND A HIGH TOTAL LUNG CAPACITY IS CONSISTENT WITH A DIAGNOSIS OF EMPHYSEMA. ALTHOUGH BRONCHODILATORS WERE ONLY SLIGHTLY USEFUL IN THIS ONE CASE, PROLONGED USE MAY PROVE TO BE BENEFICIAL TO THE PATIENT.

PULMONARY FUNCTION DIAGNOSIS:
1. MODERATELY SEVERE OBSTRUCTIVE AIRWAYS DISEASE.
   EMPHYSEMATOUS TYPE.

FIGURE 19-2 Pulmonary function report generated by PDP-11 version of PUFF.

- representation of prototypical patterns,
- addition or modification of rules to represent knowledge not previously encoded,
- alteration of the order in which information is requested during the consultation, and
- explanation of system performance.

The first point refers to the fact that many cases can be viewed as relatively simple variations of typical patterns. PUFF does not recognize that a case fits a typical pattern, nor can it recognize that a case differs in some important way from typical patterns. As a result, PUFF's explanations of its diagnoses lack some of the richness of explanation that physicians can use when a case meets, or fails to meet, the expectations of a proto-

typical case. The medical knowledge in PUFF is encoded as *rules*. Rules encode relatively small and independent bodies of domain knowledge. The rule formalism makes modification of the program's knowledge much easier than when that knowledge is embedded in computer code. However, additions or modifications to the rules as referred to in the second point have caused difficulties because changes to one rule sometimes affect the behavior of other rules in unanticipated ways. The last two points apply only to the EMYCIN version of PUFF, which runs interactively in a consultation-style, question-and-answer mode with the user. In that system, questions are sometimes asked in an unusual order, and explanations of both the questions being asked of the user and the final interpretation need to be improved.

Even though PUFF does exhibit certain limitations, the representation of pulmonary knowledge as production rules allows the encoding of interpretive expertise that previously was difficult to define because it is heuristic knowledge of the expert. EMYCIN on the SUMEX computer provided an excellent environment for acquiring, encoding, and debugging this expertise. However, it would have been inefficient and somewhat impractical to use the interactive EMYCIN version of PUFF in a hospital setting. The simplicity of EMYCIN's reasoning process made the translation into BASIC procedures a feasible task, thus allowing the hospital's own computer staff to take over maintenance of the system.

The BASIC version of PUFF runs in batch mode and does not require interaction with a physician. We believe that this system was readily accepted by the pulmonary staff for several reasons. First, the program's interpretations are consistently accurate. Second, explanations of diagnoses are appropriately detailed so that the user has confidence in the accuracy of correct diagnoses and enough information with which to recognize and modify incorrect diagnoses. Third, less physician time is required to produce consistently high-quality reports using the system than is required to analyze and dictate case reports without it. Finally, the program is well integrated into the routine of the laboratory; its use requires very little extra technician effort.

19.6 Overview of EMYCIN-PUFF

19.6.1 Knowledge Representation

The knowledge base of the EMYCIN-PUFF system consists of (a) a set of 64 *production rules* dealing with the interpretation of pulmonary function tests and (b) a set of 59 *clinical parameters*. The production version (BASIC-PUFF) has been extended to include 400 production rules and 75 clinical parameters. The clinical parameters represent pulmonary function test results (e.g., TOTAL LUNG CAPACITY and RESIDUAL VOLUME), pa-

```
RULE011

IF:  1) A: The mmf/mmf-predicted ratio is between 35 and 45, and
        B: The fvc/fvc-predicted ratio is greater than 80, or
     2) A: The mmf/mmf-predicted ratio is between 25 and 35, and
        B: The fvc/fvc-predicted ratio is less than 80
THEN: 1) There is suggestive evidence (.5) that the degree of
        obstructive airways disease as indicated by the MMF
        is moderate, and
      2) It is definite (1.0) that the following is one of the
        findings about the diagnosis of obstructive airways
        disease: Reduced mid-expiratory flow indicates
        moderate airway obstruction.

PREMISE: [$AND ($OR ($AND (BETWEEN* (VAL1 CNTXT MMF) 35 45)
                          (GREATER* (VAL1 CNTXT FVC) 80))
                    ($AND (BETWEEN* (VAL1 CNTXT MMF) 25 35)
                          (LESSP* (VAL1 CNTXT FVC) 80]
ACTION: (DO-ALL (CONCLUDE CNTXT DEG-MMF MODERATE TALLY 500)
                (CONCLUDETEXT CNTXT FINDINGS-OAD
                              (TEXT $MMF/FVC2) TALLY 1000))
```

FIGURE 19-3 A PUFF production rule in English and LISP versions.

tient data (e.g., AGE and REFERRAL DIAGNOSIS), and data that are derived from the rules (e.g., FINDINGS associated with a disease and SUBTYPES associated with the disease). There may be auxiliary information associated with the clinical parameters, such as a list of expected values and an English translation used in communicating with the user.

The production rules operate on associative <attribute object value> triples, where the attributes are the clinical parameters, the object is the patient, and the values are given by the patient data and lung test results. Questions are asked during the consultation in an attempt to fill in values for the parameters.

The production rules consist of one or more premise clauses followed by one or more action clauses. Each premise is a conjunction of predicates operating on associative triples in the knowledge base. A sample PUFF production rule is shown in Figure 19-3.

The rules are coded internally in LISP. The user of the system sees the production rules in their English form, which is shown in the upper part of the figure. The English version is generated automatically from templates, as is described in van Melle (1979).

19.6.2 Control Structure

The EMYCIN-PUFF control structure is primarily a goal-directed backward chaining of production rules. The goal of the system at any time is to determine a value for a given clinical parameter. To conclude a value

for a clinical parameter, the program tries a precomputed list of rules whose actions conclude values for the clinical parameter [refer to van Melle (1979) for details].

If the rules fail to conclude a value for a parameter, a question is then asked of the user in order to obtain that value. An exception to this process occurs for parameters labeled ASKFIRST. These represent information generally known by the user, such as results of pulmonary function tests. For these parameters it is more efficient simply to ask a consultation question than to attempt to infer the information by means of rules.[4]

19.7 Evaluation of the BASIC-PUFF Performance System

The knowledge base from the original performance version of PUFF was tested on 107 cases chosen from files in the pulmonary function laboratory at Pacific Medical Center. Those 107 cases formed a representative sample of the various pulmonary diseases, their degrees, and their subtypes. Modifications were made to the knowledge base, and the cases were tried again. This iteration continued until our collaborating expert was satisfied that the system's interpretations agreed with his own. At this point the system was "frozen," and a new set of 144 cases was selected and interpreted by the system. All 144 cases also were interpreted separately by two pulmonary physiologists (the expert working with us and a physician from a different medical center).

The results of the comparison of interpretations by each diagnostician are presented in the table in Figure 19-4. The table compares close agreement in diagnosing the severity of the disease, where close is defined as differing by at most 1 degree of severity. Thus, for example, diagnoses of mild (degree = 1) and moderate (degree = 2) are considered close, while mild and severe (degree = 3) are not. Further, a diagnosis of normal is not considered to be close to a diagnosis of a mild degree of any disease.

The table shows that the overall rate of agreement between the two physiologists on the diagnoses of disease was 92%. The agreement between PUFF and the physician who served as the expert to develop the PUFF knowledge base (MD-1 in the table) was 96%. Finally, the agreement between PUFF and the physician who had no part in the development of the PUFF knowledge base (MD-2) was 89%. Figure 19-5 shows the distribution of diagnoses by each diagnostician. The number of diagnoses made by each diagnostician does not total 144 because patients were often diagnosed as having more than one disease.

[4]In the BASIC version of PUFF implemented at PMC, all of the test data are known ahead of time so that "asking a question" merely entails retrieving another datum from a stored file.

	PERCENT AGREEMENT		
DIAGNOSIS	MD-1 MD-2	MD-1 PUFF	MD-2 PUFF
NORMAL	92	95	92
OAD	94	99	94
RLD	92	99	85
DD	90	91	85
TOTAL	92	96	89
(S.D.)	(1.63)	(3.83)	(4.69)

Diseases: Normal = Normal Pulmonary Function
OAD = Obstructive Airways Disease
RLD = Restrictive Lung Disease
DD = Diffusion Defect

FIGURE 19-4 Summary of percent agreement in 144 cases.

	DIAGNOSTICIAN		
DIAGNOSIS	MD-1	MD-2	PUFF
NORMAL	31	26	30
OAD	79	85	89
RLD	52	45	55
DD	53	35	52

FIGURE 19-5 Number of diagnoses by each diagnostician for 144 cases.

19.8 Conclusions

The PUFF research has demonstrated that if the task, domain, and researchers are carefully matched, then the application of existing techniques can result in a system that successfully performs a moderately complicated task of medical diagnosis. Success of the program can be measured not only in terms of the system's technical performance, but equally importantly, by the ease and practicality of the system's day-to-day use in the lab for which it was designed. Rule-based representation allowed easy codification and later modification of expertise. The simplicity of the rule interpreter in the Interlisp version facilitated translation into BASIC and implementation on the hospital's own PDP-11 machine. Using EMYCIN allowed the researchers to move quickly from a point where they found it difficult even to describe the diagnostic process to a point where a simple diagnostic model was implemented. Having a diagnostic model allowed them to focus on individual issues in order to improve that model. Although PUFF does not itself represent new artificial intelligence techniques, its success is a testimonial for EMYCIN. In addition, its simplicity has facilitated careful analysis of EMYCIN's rule representation and control structure and has led to other productive research efforts (Aikins, 1980; 1983; Smith and Clayton, 1980).

ACKNOWLEDGMENTS

The PUFF research team consists of an interdisciplinary group of physicians and computer scientists. In addition to the authors, these have included Larry Fagan, Ed Feigenbaum, Penny Nii, Dr. John Osborn, Dr. B. J. Rubin, and Dianne Sierra. We also thank Dr. B. A. Votteri for his help in evaluating PUFF's performance and Doug Aikins for his editorial help with this paper.

The research was funded in part by NIH grants MB-00134 and GM-24669. Computer facilities were provided by the SUMEX-AIM facility at Stanford University under NIH grant RR-00785. Dr. Shortliffe is supported by research career development award LM-00048 from the National Library of Medicine. Dr. Aikins was supported by the Xerox Corporation under the direction of the Xerox Palo Alto Research Center.

20

Developing Microprocessor-Based Expert Models for Instrument Interpretation

Sholom M. Weiss, Casimir A. Kulikowski, and Robert S. Galen

Just as PUFF was built using EMYCIN and then converted to run in a different environment, the last program discussed here was built using the EXPERT system-building tool developed at Rutgers University. In this case, however, Sholom Weiss and Casimir Kulikowski devised a scheme for developing an interpretive system and transferring it to a microprocessor environment. The scheme was successfully implemented and tested by producing a program for interpreting results from a widely used medical laboratory instrument: a scanning densitometer. Specialists in the field of serum protein electrophoresis analysis, including particularly Dr. Robert Galen, provided the knowledge needed to build an interpretive model using EXPERT. By constraining a few of the structures used in the general model, it was possible to develop procedures for automatically translating the model to a specialized application program and then to a microprocessor assembly language program. Thus model development was able to take place on a large machine, using established techniques for capturing and conveniently updating expert knowledge structures, while the final interpretive program was targeted to a microprocessor that was dependent on the application and suitable for installation as an output controller for an electrophoresis device. The experience of Weiss, Kulikowski, and Galen in

From *Proceedings of the Seventh International Joint Conference on Artificial Intelligence*, pp. 853–855 (1981). Used by permission of International Joint Conferences on Artificial Intelligence, Inc.; copies of the *Proceedings* are available from William Kaufmann, Inc., 95 First Street, Los Altos, CA 94022.

carrying out the complete process illustrates many of the requirements involved in taking an expert system from its early development phase to actual implementation and use in a real-world application. The resulting instrument produces interpretations as well as the usual protein electrophoresis curves and component percentages. It is a commercially available product—the first marketed medical device to have used AI techniques in its development.

20.1 Introduction

Most knowledge-based medical consultation systems developed during the 1970s were relatively large-scale experimental prototypes (Chapters 5 through 8). Their advice on diagnostic and treatment problems typically involved approximate reasoning over a space of many interrelated hypotheses, characteristically supported by hundreds of observations linked to them by various types of reasoning rules. By adopting symbolic reasoning methods with more powerful representations than the traditional mathematical decision-making schemes, these knowledge-based systems produced results that were generally easier to analyze, explain, and update than those from more conventional systems. Human-engineering features were often stressed as an important means of enhancing the interaction with the expert systems. Successful clinical experience with many of these systems has been reported in pilot demonstration projects, yet few are in routine clinical use at present.[1] Both technical and social factors contribute to the difficulties of introducing expert systems into the everyday practice of medicine. One often cited technical factor is the slow rate of manual data entry required by most of the larger systems. This problem is minimized for applications where most of the data can be read directly off a clinical instrument and only a few items must be entered manually. The commercial availability and use of automated electrocardiogram interpretation programs (using traditional algorithmic techniques) support this point. Regardless of the methods used in constructing a knowledge base, or its complexity, instrument-derived interpretations are more likely to be accepted because they can be seen as extensions of the instrument. And since many advanced medical instruments are already microprocessor-controlled, it may be possible to add an interpretive module that enhances the performance of such a device at relatively little extra cost.

In this paper we briefly describe how we were able to accelerate the development of interpretive software for a widely used laboratory instrument, the scanning densitometer. We did this by automatically producing

[1]See Chapter 19 for a report on the successful use of PUFF at the Pacific Medical Center in San Francisco.

a computer translation of an expert model for serum protein electrophoresis interpretation, developed on a mainframe computer, into a microprocessor assembly language version. The translation methods have been generalized so that this process can be repeated for EXPERT (Weiss and Kulikowski, 1979) models in any domain, with a few restrictions on the types of knowledge structures used.

By taking this approach, we have demonstrated that knowledge-engineering methods from expert systems can be used to full advantage in producing an effective model, which can then be transferred with ease to a microcomputer.

20.2 Methods

Several general-purpose schemes for building consultation systems have evolved from work on the earlier, more specific domain-dependent systems. Two such schemes that were originally designed for representing medical consultation problems in particular are the EXPERT and EMYCIN (van Melle, 1979) systems. Both provide built-in control mechanisms operating over specific types of production-rule models. The consultation program of EXPERT is primarily event-driven, while that of EMYCIN is predominantly goal-directed.

The EXPERT system has been used in building a number of expert medical consultation models (mainly in ophthalmology, rheumatology, and endocrinology) and pilot prototypes in several nonmedical areas (spectroscopy interpretation, car repair, hazardous spill management, and oil well log interpretation).

The process of model design and transfer that we used in developing the microprocessor-based expert model for serum protein electrophoresis interpretation involved the following tasks:

- specification of the knowledge base using EXPERT,
- empirical testing with several hundred cases,
- refinement of the knowledge base by the expert,
- further cycle of testing with additional cases and review by independent experts,
- test of the final model on the large machine,
- automatic translation of the EXPERT model to a specialized program and a microprocessor assembly language program, and
- interfacing of assembly language model with instrument.

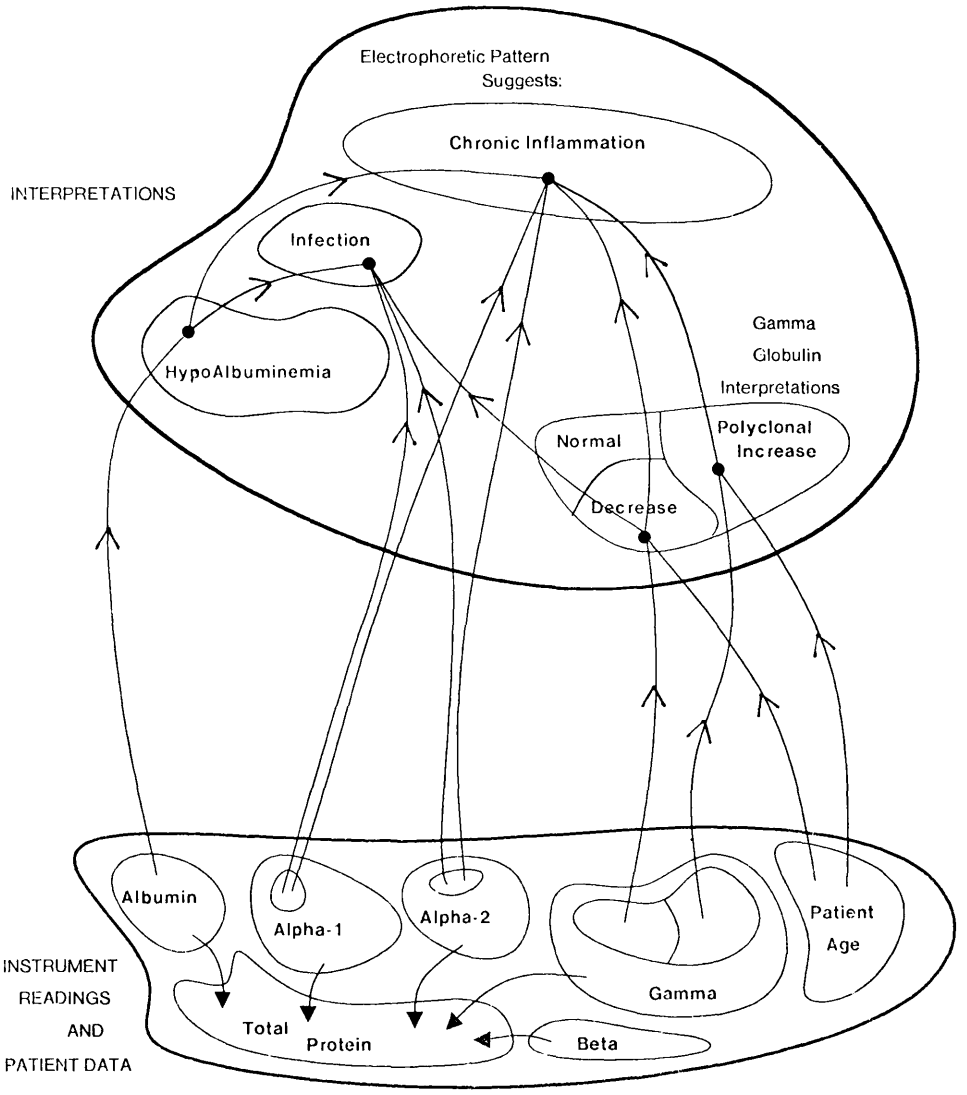


FIGURE 20-1 Sample rules (arrows) linking primary data and interpretative conclusions.

This last step requires detailed knowledge of the instrument. In this application, the manufacturer interfaced the interpretive program to the existing program for printing instrument readings.

Figure 20-1 illustrates the types of conclusions reached by the interpretive system and the type of rules used in reasoning. The most sig-

nificant restriction on the type of production rules used in the model was to limit the use of confidence measures to three values, representing confirmation, denial, and unknown status. In applications of this type, it should be noted that the strategy of questioning is not a significant task because most of the information will be obtained directly from the instrument. In building the EXPERT model, we simulated this situation by entering the values of certain key features of the instrument signal (Figure 20-2) that are currently given as a digital output by the instrument. These features include peaks of the waveform and areas under certain segments of the waveform. A few items (patient identification, age, and some waveform features that are more easily scanned by the technician) are entered manually.

The serum protein electrophoresis model required several stages of refinement over a period of six months, with the aid of one principal expert and comments and suggestions from the independent experts. We began with a relatively small and simple model, having 10 conclusions and a production rule for each. After the first cycle of revision we had about 25 conclusions and 50 rules, which included many for handling interactions among the hypotheses. The current model has 38 conclusions and 82 production rules. Its performance on 256 test cases covering the full spectrum of conclusions is 100% acceptable to our experts. They expect differences of opinion on the amount of detail included in the present set of conclusions, but feel that covering infrequently found problems would detract from a model that is to be disseminated widely. An option for allowing users to add a written record of their own opinions on such unusual cases has been provided in the final microprocessor implementation.

20.3 Conclusions

The completed microprocessor version of the interpretative serum protein electrophoresis model may not look much different than it would if it had been hand-coded directly in the assembly language of the microprocessor or translated from an algorithmic language. There is, nevertheless, a fundamental difference. With our system, we can rapidly produce new versions of the microprocessor program from our high-level EXPERT model in response to any changes suggested by the experts or resulting from future empirical analysis and clinical tests in the field. In contrast, considerable effort would usually be required to recode directly on a microprocessor. Besides, the original expert-derived model is also very different from one produced by more traditional methods. Our conclusions and intermediate hypotheses were developed in such a way that they include not only diagnostic considerations but also prognostic, treatment, and future test selection decisions for motivating their use. The large amount of

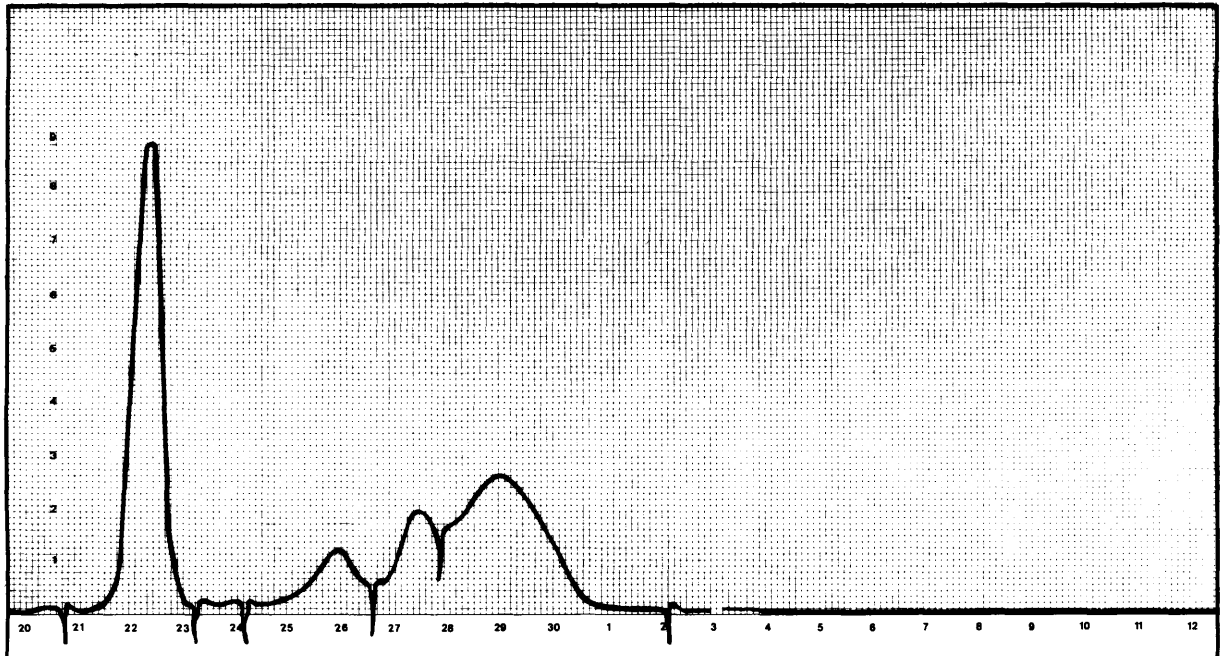


FIGURE 20-2 Interpretative analysis: Electrophoretic pattern suggests chronic inflammation.

experimentation that took place with the model as it went through its cycles of testing and modification could only be carried out on a larger system, with adequate facilities for analyzing many cases and knowledge-engineering tools for changing the model. A recently published version of an interpretative model in this domain, developed with very traditional programming techniques, shows a contrasting sparsity in diagnostic statements (Dito, 1977). In addition, the conclusions of that model appear to be overly specific given the nature of the supporting data. Thus, while programs of this type may be initially simple to implement, they do not incorporate the elements of expert reasoning that are essential to a clinically helpful program.

In conclusion, the work reported here is a novel illustration of the requirements encountered in taking an expert system from an early developmental phase to actual implementation and use in the real world. Such applications can lead to the increasing acceptance of expert systems in medicine and other domains where similar problems can be found.

ACKNOWLEDGMENTS

This work was supported in part by grant RR-643 of the National Institutes of Health, DRR-BRP. Technical assistance was provided by Helena Laboratories, Beaumont, Texas.

21

Anticipating the Second Decade

Edward H. Shortliffe and William J. Clancey

The research efforts described in this book may paradoxically appear to be both hideously complex and yet ridiculously simplistic—complex in the range of concepts they attempt to capture, encode, and use effectively, but simplistic in the important areas of human knowledge and common sense that they ignore (but that we know can be crucial to excellent clinical decision making). Viewed in this light, the research invites the question whether "ultimate" AIM systems, when they are eventually constructed, will be manageable and amenable to ongoing refinement. Or will they become so large and complex that they will totally outgrow the ability of their developers to cope with their knowledge bases and with the need for ongoing verification and updating?

It is certainly true that the research has raised at least as many new questions as it has answered old ones, but such is the nature of scholarly investigation. It is unlikely that we will ever see the day when all questions have been answered and all the problems solved. However, as the field progresses, we believe that useful (albeit limited) tools will increasingly become available, particularly as the hardware revolution (made possible by large-scale integration) provides the AIM field with cost-effective vehicles for moving advice programs from research laboratories to hospitals and private offices. Hardware and software advances are also beginning to offer us new models of system-building environments, ones in which graphical capabilities and interactive tools provide knowledge engineers with effective methods for dealing with systems that are much too large to be managed using traditional hard-copy listings for reference (Tsuji and Shortliffe, 1983).

Many of the ideas presented in this chapter were previously discussed by E. H. Shortliffe (1982a; 1982b).

In this final chapter, we summarize the trends of the past decade while citing the important research problems that remain to be solved in the years ahead. The discussion is motivated by a summary of the design considerations that have been identified by asking physicians what they would demand from a clinical consultation system before they would be willing to use it routinely. We also identify those kinds of medical problems for which practical systems can be built soon, using the kinds of techniques that have been developed during the 1970s. The ultimate systems are still probably many decades away, but existing techniques help define a subset of problems with which we are already prepared to deal.

21.1 What Physicians Want

Researchers in the field of medical decision making must contend with a great deal of ambivalence on the part of the potential physician users of their systems. On the one hand, there is a "show me" attitude expressed by a profession that has heard the potential of clinical computing extolled for more than ten years but has yet to see a widely accepted decision support system. On the other hand, there are indications that the environment is changing, with an increased acknowledgment that clinical decision-making research can validly contribute to medical practice. For example, we have seen significant clinical changes result from theoretical work in clinical decision analysis (e.g., the recent American Cancer Society recommendations regarding mammography and PAP smear screening) and the development of an ambitious, well-received journal in the field (Lusted, 1981). Studies of physician attitudes (Teach and Shortliffe, 1981) have also shown that there is a growing curiosity about computers and a heightened faith in their potential. This phenomenon has been further demonstrated by the emergence of doctors with home computers and customized office systems, and by the success of educational programs designed to introduce physicians to computers for both business and clinical applications.

The study of physicians' attitudes towards clinical consultation systems (Teach and Shortliffe, 1981) showed that a significant segment of the medical community believes that assistance from computer-based consultation systems will ultimately benefit medical practice. Teach and Shortliffe also studied the physicians' demands regarding desirable features for such systems if they are to be useful and clinically accepted. The resulting design considerations highlight performance capabilities that are a challenge to medical computer scientists. Consider, for example, the six design features that physicians rated *most important* for future consultation systems:

1. they should be able to explain their diagnostic and treatment decisions to physician users;

2. they should be portable and flexible so that the M.D. can access them at any time and place;
3. they should display an understanding of their own medical knowledge;
4. they should improve the cost-efficiency of tests and therapies;
5. they should automatically learn new information when interacting with medical experts; and
6. they should display common sense.

No current consultation system meets all these criteria, but the list does help identify both the research challenges that lie ahead and the criteria for assessing new systems that may be introduced. The first, third, fifth, and sixth of these criteria are central issues being addressed by researchers in the AI field and thereby emphasize the importance of AI as an ingredient in the development of clinically acceptable decision aids.

21.2 Two Decades of Research

Medical decision-making research in the 1960s emphasized the use of the computer to deal with probabilistic information, to recognize patterns using numerical techniques, to model physiological processes that were amenable to mathematical simulation, or to encode algorithmic approaches to routine clinical chores. The field was then in its first decade as an identifiable area of research, and the emphasis was on how to get machines to make decisions that were both accurate and reliable. Formal statistical approaches that had been impractical before computers became available were, quite naturally, the first techniques to be tried as physicians and engineers began to appreciate the computer's potential as a clinical tool.

In the 1970s, however, there was a shift in research direction. As was outlined in Chapter 3, investigators increasingly realized that there are several key problems that escape attention if the research focuses solely on the development of techniques for reaching good decisions. These include:

1. the problem of *data acquisition*—how to acquire, encode, and control for variations in the descriptors that define patients and populations;
2. the problems of *knowledge acquisition and representation*—how to acquire and encode the kinds of judgmental perceptions and the commonsense approach that characterize expertise in the clinical decision-making areas being modeled;
3. the problem of *explanation*—how to build decision support programs that not only give advice but are able to defend their decisions in terms physicians can understand; and

4. the *logistics of integration*—how to design and implement computer-based decision aids that fit smoothly into the daily routine of physicians' practices, that acknowledge their hectic schedules, and that seek to demystify and simplify the mechanics of the human-computer interface.

Several early approaches to these problems were developed during the last decade. Large patient data bases have been constructed and used to aid in defining prognoses for new cases (Feinstein et al., 1972; Fries, 1972; Rosati et al., 1975). Investigators who depend on valid statistics to support their decision-making systems have begun to look at geographical variations in populations in order to assess the transferability of programs (de Dombal, 1979). Hospital information systems have become increasingly common and provide promising early models for the way in which relevant data will eventually be routinely acquired (Lindberg, 1977). There has also been complementary work in the development of large computer-based text documents designed to bring up-to-date knowledge of a domain to the practicing physician (Bernstein et al., 1980).

During the same period, AI approaches have become prominent and have suggested several methods for encoding uncertainty, representing expert knowledge, and modeling the reasoning processes of accomplished clinicians. The symbolic reasoning techniques described in this book have suggested ways decision-making programs can explain their reasoning to physicians, thereby allowing the user to decide whether to follow the system's recommendations. Interactive techniques have been developed that also allow experimental systems to interview experts and to acquire new knowledge directly from them (Davis, 1979).

Finally, there have been several notable experiments that have sought new ways to encourage physicians to interact with computer programs. These have included systems using light pens (Watson, 1974) or touch screens (Schultz and Davis, 1979) and decision support programs integrated into large-scale hospital information systems (Pryor et al., 1982). These efforts and others have demonstrated that physicians will learn to use computers and accept their role if the benefits of the technology outweigh the costs of learning how to use the device and integrating it into one's normal routine.

21.3 The Challenges Remaining

A litany of recent accomplishments partly serves to emphasize the significant problems still remaining, however. Many of the experiments we have cited are only first steps toward the development of clinically useful tools. Some of the major barriers are practical ones relating to the logistics of interfacing patient data bases with expert systems, issues of legal liability (Brannigan, 1981), and the problem of training system users and knowl-

edge engineers. At a more basic level, as is true with any emerging science, the development of short-term solutions tends to lead to a new understanding of the nature of the remaining problems and helps define the fundamental research directions for the future. Current results suggest that the following areas are among those requiring attention in the decade ahead:

1. additional *psychological studies,* similar in motivation to some of the pioneering studies of the 1970s (Elstein et al., 1978; Kassirer and Gorry, 1978), that will provide new insights into optimal methods for simulating expert decision-making performance and may suggest novel approaches to the organization of knowledge and its interaction with probabilistic information;
2. improved *techniques for representing and using causal and mechanistic relationships* (because expert decision-making behavior sometimes depends on an ability to reason from "first principles" rather than relying on empirical associations between observations and hypotheses);
3. improved methods for *acquiring expert knowledge, encoding it, and checking it for inconsistencies or incompleteness* (Davis, 1979; Suwa et al., 1982; Politakis and Weiss, 1984), thereby helping avoid the problems of knowledge base construction that have been major impediments to the development of expert systems;
4. enhanced *explanation capabilities,* ideally guided by an improved understanding of how human beings explain things to one another and, in particular, how they adapt their explanations to the knowledge and experience of the individual requesting advice;
5. experimentation with *new machine architectures* (e.g., parallel processing or networking of multiple coordinated processors) that may permit an optimal assignment of languages and interfaces for the individual subtasks required by high-performance decision-making programs;
6. experiments that seek to provide an *optimal melding of symbolic techniques* drawn from artificial intelligence research *and the analytic techniques* of formal statistics, pattern recognition, and decision theory; and
7. research into novel ways that developing *technologies for personal computing and graphics* might heighten both the acceptability and cost-effectiveness of systems to aid physicians with their decision-making tasks.

21.4 Steps in Demonstrating the Effectiveness of a Consultation System

With significant fundamental problems such as those above requiring solutions, can *anything* of practical use for decision support be implemented soon? Can we define clinical problems that are amenable to short-term

solutions and that will allow AIM researchers to undertake validating experiments in active clinical environments rather than in hypothetical experimental settings such as those used for the evaluation of MYCIN (Yu et al., 1979a; 1979b) and INTERNIST-1 (Chapter 8)? We believe that the answer to both of these questions is "yes." Short-term clinical implementation is inherently intertwined with evaluation issues, however, and we have accordingly found it useful to define a series of steps through which an advice system must pass as it moves from a research environment to ongoing clinical use.

Diagnostic programs have tended to be assessed on the basis of their *decision-making accuracy*—the issue that is usually central to the system's design and to the motivation of the system's developers. Yet there are several additional components to the evaluation process when it is performed optimally. In order to control for confounding variables, we have suggested (Shortliffe and Davis, 1975) that system evaluations should be undertaken in a series of steps as follows:

1. *Demonstrate a need for the system.* Are there data indicating that physicians need help with the task for which the consultation system is designed to assist, and if so, is a computer necessary to provide that assistance?
2. *Demonstrate that the system performs at the level of an expert.* Can it be formally shown that the system reaches the same decisions as experts who are presented with the same clinical decision tasks? If there are frequent disagreements, can it be shown that the system is correct at least as often as the experts are? Note that the determination of correctness thereby requires some "gold standard" against which the performance of both experts and the consultation system can be measured.
3. *Demonstrate the system's useability.* Can physicians easily learn to handle the mechanics of interacting with the consultation system? Is the response time adequate? Is the system's performance sufficiently transparent so that the clinician can obtain the information he or she needs in an efficient and straightforward manner?
4. *Demonstrate acceptance of the system by physicians.* Can it be shown that clinicians offered the decision tool will in fact return to use it, even when access to it is entirely optional?
5. *Demonstrate an impact on the management of patients.* If physicians use the system, can it be shown that they follow the advice it offers? If not, has it favorably changed their behavior in some other way?
6. *Demonstrate an impact on the well-being of patients.* If physicians are following the recommendations of the consultation system, can it be shown that patients are benefiting from its use? Are there objective measurements of patient-care quality that can be assessed before and after the decision aid has been introduced?

7. *Demonstrate cost-effectiveness of the tool.* If all the other validation criteria have been satisfied, can it be shown that there is a version of the consultation system that is cost-effective when both costs and benefits are assessed using some generally accepted criterion?

These seven steps for demonstrating the effectiveness of a medical consultation system are idealized and difficult to traverse. We know of no medical decision-making system that has rigorously been shown to meet formal validation criteria at all seven steps of development. In fact, most systems have been assessed only at step 2, and remarkably few have met even the criterion of need specified in step 1.

Some observers of the field may argue that the theoretical issues in the development of high-performance consultation systems are still so great that it is folly to focus attention on steps 3 through 7 at this time. Yet many significant theoretical barriers to the successful implementation of consultation systems do not arise at step 2 and will not be met until the subsequent steps are encountered.

21.5 Characteristics of an Optimal Application Domain

Attitude surveys such as that of Teach and Shortliffe (1981) help delineate some of the issues that must be addressed by system builders if clinically acceptable decision tools are to be developed. However, since most of these issues are best studied and assessed at the later stages of system implementation, scientists who wish to address them in their *current* research must select an appropriate clinical problem area. The following criteria for that selection seem to be particularly pertinent:

1. As indicated above, there must be a *demonstrated need for help* in the domain. A program that deals with an "interesting" problem, but one with which physicians already do rather well, will generate little interest.
2. Equally as important, there must be a *recognized need for help* by the physicians themselves. Data showing poor performance by the overall population of physicians will not necessarily convince individual practitioners that they are among those needing help. Demand will come only from perceived need on the part of the intended users.
3. The domain should ideally provide a *core of formalized and readily available knowledge.* We have learned that knowledge base development can be an arduous and time-consuming aspect of consultation system re-

search. Theoretical issues regarding knowledge completeness, consistency, and acquisition must inevitably be faced when a complex system is built for a domain in which expert knowledge is poorly formalized.

4. The domain must provide a *straightforward mechanism for introducing a computer-based tool into the daily routine* of the physicians who use it. This point has several corollaries. First, use of the computer should ideally replace a task that is already being performed; this helps guarantee that the system will require a minimal additional time commitment. Second, the mechanical interface must be rapid, congenial, and easy to learn to use. And third, the decision tool's design must demonstrate a respect for the physician's hectic schedule.
5. The program should *maintain the physician's role as ultimate decision maker* (e.g., by giving explanations for recommendations and allowing the user to override any advice that is offered).
6. The system developers must be able to identify *highly motivated collaborators* from the domain of expertise.
7. The problem area should allow the initial prototype system to *avoid major theoretical barriers* (e.g., the domain should not require solutions to problems such as the development of approaches to the management of inexact inference, generalized methods for the management of temporal reasoning, encoding of strategic knowledge for domain-specific problem solving, or generation of highly customized explanations that demonstrate "first principle" understanding of the clinical area).

Criteria such as these have guided the development and progress of one of the newer AIM research activities. That project, known as ONCOCIN, is an expert system designed to aid physicians in the management of patients receiving cancer chemotherapy. The program is based on AI representation and control techniques similar to those described in this book (Shortliffe et al., 1981), but much of the effort has focused on getting the program implemented for use by physicians. Experience with the program and its users, both before and after its clinical introduction in May 1981, has recently been described (Bischoff et al., 1983). In order to provide a congenial high-speed interface, the system required a novel system architecture that separated the reasoning and interactive components (Gerring et al., 1982). ONCOCIN has also provided a productive environment for research on methods to ensure knowledge base completeness and consistency (Suwa et al., 1982) and on specialized explanation techniques (Langlotz and Shortliffe, 1983). Because of its initial promising success, plans have been made to convert the program to run on professional workstations and to use them as a vehicle for disseminating the technology to nonacademic settings (Tsuji and Shortliffe, 1983). More detailed discussions of ONCOCIN may be found in a recent book by Buchanan and Shortliffe (1984).

21.6 On Artificial Intelligence and Medical Computer Science

Those who work in the AIM field are uniformly enthused about the field's potential to do social good but are also aware of the common misinterpretation of their goals and of the frequent failure to acknowledge the fundamental research barriers that remain to be conquered. We have already discussed the problems that lie ahead, and we hope that the reader will share the cautious optimism that we feel about the future. Misinterpretations of the goals of AI research, however, at least partly relate to the phrase *artificial intelligence* itself. For example, the eminent essayist Lewis Thomas recently wrote in a "Notes of a Biology Watcher" column in the *New England Journal of Medicine* (Thomas, 1980):

> The most profoundly depressing of all ideas about the future of the human species is the concept of artificial intelligence. The ambition that human beings will ultimately cap their success as evolutionary overachievers by manufacturing computers of such complexity and ingenuity as to be smarter than they are, and that these devices will take over and run the place for human betterment or perhaps, later on, for machine betterment, strikes me as wrong in a deep sense, maybe even evil. Until now, computers have had the look of useful, often indispensable tools . . . [But] this is what the artificial intelligence people are talking about: a mechanical brain with the capacity to look back over the past and make accurate predictions about the future, then to lay out flawless plans for changing that future any way it feels like, and, most appalling of all, capable of feeling like doing one thing or another. Machines like this would be connected to each other in a network all around the earth, doing all the thinking, maybe even worrying nervously. But being right all the time. Leaving us with time for leisure . . .

We are not sure where Thomas obtained his information about the field, but we hope that this volume has demonstrated his misinterpretation of the nature of AI—both regarding the motives of the researchers and regarding expectations of what can and will be accomplished. One is reminded of a recent book by Weizenbaum that questioned not so much what *could* be accomplished by the AI field but what *should* be accomplished (Weizenbaum, 1976).

In response to Thomas's essay, Shortliffe and Buchanan sent a letter to the editor of *New England Journal of Medicine,* a portion of which was published with other letters on the subject (Shortliffe and Buchanan, 1980). We reproduce the entire original letter here:

> Lewis Thomas' polemic against artificial intelligence responds more to the emotional content of the phrase than to the realities of the techniques

and goals associated with this subfield of computer science. It is ironic that his opinion piece should appear at a time when computing techniques drawn from AI are being increasingly applied in clinical domains.

It is commonly accepted that computers can offer the medical professions significant relief from the complexities of routine information handling and data analysis (e.g., office billing systems, CT scanners). Because of the frequently cited explosion of medical knowledge, much research has also focused on computer-based tools to assist physicians with clinical decision making. Medical computing researchers are being drawn to AI largely because they see in the field techniques that will make programs for physicians more congenial, acceptable, and clinically useful. One of the goals of AI is to construct intelligent assistants that reason symbolically using empirical associations, accepted theory, and experts' judgmental knowledge. Although a textbook is a well-accepted tool, it is static and inflexible in the sense that it fails to customize its knowledge to the consideration of specific patients. By reasoning with general knowledge to suggest an individual approach to a patient's management, a program that can function as an intelligent assistant may further enhance the physician's effectiveness.

Thomas would have us believe that AI research purports to create a network of machines "doing all the thinking . . . leaving us with time for leisure." Yet in its medical applications, AI research is seeking ways to overcome the tendency to estrangement between man and machine, a frequent complaint that has tended to limit the utility of clinical computing. AI workers are attempting to provide us with computer-based tools that will make doctors more effective thinkers and clinical decision makers (Shortliffe, 1980). In his fervor for pursuing the philosophical correlates of a phrase like *artificial intelligence*, Thomas loses sight of the fact that "intelligent" knowledge-based machines may continue to serve as the "useful, often indispensable tools" which he admits he has come to appreciate.

The preceding interchange brings us naturally to a further definition of goals that will guide "the second decade" of AIM research that lies ahead. In addition to the research areas previously outlined, it is clear that two issues stand foremost on the medical computing agenda for the 1980s (Shortliffe, 1983): (1) there must be improved education of medical students and practicing physicians regarding computers and decision making, and (2) there must be an enhanced acceptance of *medical computer science* as an intrinsic component of the modern academic medical environment. The financial and academic support necessary for tackling difficult tasks such as those we have outlined will be made available only if there is improved recognition of the field's potential and of the fundamental research questions that exist for the medical computing community.

References

Aikins, J. S. 1979. Prototypes and production rules: An approach to knowledge representation for hypothesis formation. In *Proceedings of the Sixth International Joint Conference on Artificial Intelligence,* pp. 1–3. Stanford, CA: Stanford University, Department of Computer Science.

———. 1980. Prototypes and production rules: a knowledge representation for computer consultations. Ph.D. dissertation, Department of Computer Science, Stanford University. (Report nos. HPP-80-17 and STAN-CS-80-814.)

———. 1983. Prototypical knowledge for expert systems. *Artificial Intelligence* 20: 163–210.

Anderson, J. R. 1976. *Language, Memory and Thought.* New York: Lawrence Erlbaum Associates.

———. 1980. Concepts, propositions, and schemata: What are the cognitive units? In *Cognitive Processes: Nebraska Symposium on Motivation,* eds. H. E. Howe and J. H. Flowers. Lincoln, NE: University of Nebraska Press.

Anderson, J. R., Kline, P. J., and Beasley, C. M. 1979. A general learning theory and its application to schema abstraction. In *The Psychology of Learning and Motivation,* ed. G. H. Bower. New York: Academic Press.

Anderson, J. R., Greeno, J. G., Kline, P. J., and Neves, D. M. 1981. Acquisition of problem-solving skill. In *Cognitive Skills and Their Acquisition,* ed. J. R. Anderson. Hillsdale, NJ: Lawrence Erlbaum Associates.

Anderson, R. H., and Gillogly, J. J. 1977. RAND artificial terminals agent: Design philosophy. Report no. R-1809-ARPA, Rand Corporation, Santa Monica, CA.

Armitage, P. 1971. *Statistical Methods in Medical Research.* Oxford, U.K.: Blackwell Scientific Publications.

Armitage, P., and Gehan, E. A. 1974. Statistical methods for the identification and use of prognostic factors. *International Journal of Cancer* 13: 16–36.

Balzer, R., Goldman, N., and Wile, D. 1977. Informality in program specifications. In *Proceedings of the Fifth International Joint Conference on Artificial Intelligence,* pp. 389–397. Pittsburgh, PA: Carnegie-Mellon University, Department of Computer Science.

Barr, A., and Atkinson, R. C. 1975. Adaptive instructional strategies. Paper presented at the IPN Symposium 7: Formalized Theories of Thinking and Learning and Their Implications for Science Instruction.

Barr, A., Beard, M., and Atkinson, R. C. 1976. The computer as a tutorial laboratory: The Stanford BIP project. *International Journal of Man-Machine Studies* 8: 567–596.

Barr, A., Feigenbaum, E. A., and Cohen, P. (eds.). 1981–1982. *Handbook of Artificial Intelligence,* 3 vols. Los Altos, CA: William Kaufmann.

Barrows, H. S., and Tamblyn, R. M. 1980. *Problem Based Learning: An Approach to Medical Education.* New York: Springer-Verlag.

Barrows, H. S., Feightner, J. W., Neufeld, V. R., and Norman, G. R. 1978. Analysis of the clinical methods of medical students and physicians (final report to Ontario Ministry of Health). Hamilton, Ontario, Canada: McMaster University.

Barstow, D. 1977. A knowledge-based system for automatic program construction. In *Proceedings of the Fifth International Joint Conference on Artificial Intelligence,* pp. 382–388. Pittsburgh, PA: Carnegie-Mellon University, Department of Computer Science.

Beeson, P. W., and McDermott, W. (eds.). 1975. *Cecil-Loeb Textbook of Medicine.* Philadelphia: W. B. Saunders.

Bell, D. A. 1962. *Intelligent Machines: An Introduction to Cybernetics.* London: Pitman.

Benbassat, J., and Schiffmann, A. 1976. An approach to teaching the introduction to clinical medicine. *Annals of Internal Medicine* 84: 477–481.

Bennett, J. S., and Englemore, R. S. 1979. SACON: A knowledge-based consultant for structural analysis. In *Proceedings of the Sixth International Joint Conference on Artificial Intelligence,* pp. 47–49. Stanford, CA: Stanford University, Department of Computer Science.

Bernstein, L. M., Seigel, E. R., and Goldstein, C. M. 1980. The hepatitis knowledge base: A prototype information transfer system. *Annals of Internal Medicine* 93(2): 169–181.

Betaque, N. E., and Gorry, G. A. 1971. Automating judgmental decision making for a serious medical problem. *Management Science* 17: 421–434.

Bhaskar, R., and Simon, H. A. 1977. Problem solving in semantically rich domains: An example from engineering thermodynamics. *Cognitive Science* 1: 193–215.

Bieman, K. 1979. The role of computers in conjunction with analytical instrumentation. *Proceedings of the IEEE* 16: 1287–1299.

Bischoff, M. B., Shortliffe, E. H., Scott, A. C., Carlson, R. W., and Jacobs, C. D. 1983. Integration of a computer-based consultant into the clinical setting. In *Proceedings of the Seventh Annual Symposium on Computer Applications in Medical Care,* pp. 149–152.

Bitzer, M. D., and Bitzer, D. L. 1973. Teaching nursing by computer: An evaluative study. *Computers in Biology and Medicine* 3: 187–204.

Bleich, H. L. 1969. Computer evaluation of acid-base disorders. *Journal of Clinical Investigation* 48: 1689–1696.

———. 1971. The computer as a consultant. *New England Journal of Medicine* 284: 141–147.

———. 1972. Computer-based consultation: Electrolyte and acid-base disorders. *American Journal of Medicine* 53: 285–291.

Blum, R. L. 1981. Displaying clinical data from a time-oriented database. *Computers in Biology and Medicine* 11(4): 197–210.

———. 1982. Discovery and representation of causal relationships from a large time-oriented clinical database: The RX project. In *Lecture Notes in Medical Informatics,* vol. 19, eds. D. A. B. Lindberg and P. L. Reichertz. New York: Springer-Verlag.

Blum, R. L., and Wiederhold, G. 1979. Inferring knowledge from clinical data banks: Utilizing techniques from artificial intelligence. In *Proceedings of the Second Annual Symposium on Computer Applications in Medical Care,* pp. 303–307. Long Beach, CA: IEEE Computer Society.

Bobrow, D. G., and Collins, A. M. (eds.). 1975. *Representation and Understanding: Studies in Cognitive Science.* New York: Academic Press.

Bobrow, D. G., and Norman, D. A. 1975. Some principles of memory schemata. In *Representation and Understanding: Studies in Cognitive Science,* eds. D. G. Bobrow and A. M. Collins, pp. 131–149. New York: Academic Press.

Bobrow, D. G., and Winograd, T. 1977. An overview of KRL, a knowledge representation language. *Cognitive Science* 1: 3–46.

Bobrow, D. G., Kaplan, R. M., Kay, M., Norman, D. A., Thompson, H., and Winograd, T. 1977. GUS, a frame-driven dialog system. *Artificial Intelligence* 8(2): 155–173.

Bonner, R. E., Evangelisti, C. J., Steinbeck, H. D., and Cohen, L. 1964. A diagnostic assistance program. In *Proceedings of the Sixth IBM Medical Symposium,* p. 81.

Brachman, R. J. 1979. What's in a concept: Structural foundations for semantic networks. In *Associative Networks: The Representation and Use of Knowledge by Computers,* ed. E. Findler, pp. 3–50. New York: Academic Press.

Brandt, E. N. 1974. Role of the computer in continuing medical education. *Texas Medicine* 3: 43–48.

Brannigan, V. M., and Dayhoff, R. E. 1981. Liability for personal injuries caused by defective medical computer programs. *American Journal of Law and Medicine* 7: 123–144.

Brodman, K., vanWoerkom, A. J., Erdman, A. J., and Goldstein, L. S. 1959. Interpretations of symptoms with a data processing machine. *AMA Archives of Internal Medicine* 103: 776.

Brooks, R. E., and Heiser, J. F. 1979. Transferability of a rule-based control structure to a new knowledge domain. In *Proceedings of the Third Annual Symposium for Computer Applications in Medical Care,* pp. 56–63. Long Beach, CA: IEEE Computer Society.

Brown, B. W., and Hollander, M. 1977. *Statistics: A Biomedical Introduction.* New York: Wiley.

Brown, J. S., and Burton, R. R. 1978. Diagnostic models for procedural bugs in mathematical skills. *Cognitive Science* 2: 155–192.

Brown, J. S., and Goldstein, I. P. 1977. Computers in a learning society. Testimony for the House Science and Technology Subcommittee on Domestic and International Planning, Analysis and Cooperation.

Brown, J. S., Burton, R. R., and Zdybel, F. 1973. A model-driven question-answering system for mixed-initiative computer-assisted instruction. *Systems, Man and Cybernetics* SMC-3(3): 248–257.

Brown, J. S., Burton, R. R., and Bell, A. G. 1974. SOPHIE: A sophisticated instructional environment for teaching electronic troubleshooting (An example of AI in CAI). Report no. 2790, Bolt Beranek and Newman, Cambridge, MA.

Brown, J. S., Burton, R. R., and Zdybel, F. 1975. Multiple representations of knowledge for tutorial reasoning. In *Representation and Understanding,* eds. D. G. Bobrow and A. M. Collins, pp. 311–349. New York: Academic Press.

Brown, J. S., Rubenstein, R., and Burton, R. 1976. Reactive learning environment for computer-aided electronics instruction. Report no. 3314, Bolt Beranek and Newman, Cambridge, MA.

Brown, J. S., Collins, A. M., and Harris, G. 1977. Artificial intelligence and learning strategies. In *Learning Strategies,* ed. H. O'Neill. New York: Academic Press.

Buchanan, B. G., and Feigenbaum, E. A. 1978. DENDRAL and Meta-DENDRAL: Their applications dimension. *Artificial Intelligence* 11(1): 5–24.

Buchanan, B. G., and Lederberg, J. 1971. The heuristic DENDRAL program for explaining empirical data. In *Proceedings of the International Federation for Information Processing,* pp. 179–188.

Buchanan, B. G., and Shortliffe, E. H. 1984. *Rule-Based Expert Systems: The MYCIN Experiments of the Stanford Heuristic Programming Project.* Reading, MA: Addison-Wesley.

Buchanan, B. G., Sutherland, G., and Feigenbaum, E. A. 1969. Heuristic DENDRAL: A program for generating explanatory hypotheses in organic chemistry. In *Machine Intelligence 4,* eds. B. Meltzer and D. Michie, pp. 209–254. Edinburgh, U.K.: Edinburgh University Press.

Bunge, M. 1963. *Causality—The Place of the Causal Principle in Modern Science.* Cleveland: World Publishing Co.

Burton, R. R. 1976. Semantic grammar: An engineering technique for constructing natural language understanding systems. Report no. 3453, Bolt Beranek and Newman, Cambridge, MA.

———. 1979. An investigation of computer coaching for informal learning activities. *The International Journal of Man-Machine Studies* 11: 5–24.

Burton, R. R., and Brown, J. S. 1976. A tutoring and student modelling paradigm for gaming environments. In *Proceedings of the Symposium on Computer Science and Education,* pp. 236–246.

Byar, D. P. 1980. Why databases should not replace randomized clinical trials. *Biometrics* 36: 337–342.

Carbonell, J. G. 1979. The counterplanning process: Reasoning under adversity. In *Proceedings of the Sixth International Joint Conference on Artificial Intelligence,* pp. 124–130. Stanford, CA: Stanford University, Department of Computer Science.

Carbonell, J. R. 1970. Mixed-initiative man-computer instructional dialogues. Report no. 1971, Bolt Beranek and Newman, Cambridge, MA.

Carey, S. 1973. Cognitive competence. In *The Growth of Competence,* eds. D. Connelly and J. Bruner, New York: Academic Press.

Carnegie-Mellon University, Computer Science Research Group. 1977. Summary of the five-year ARPA effort in speech understanding research. Report, Carnegie-Mellon University.

Carr, B., and Goldstein, I. 1977. Overlays: A theory of modelling for CAI. Report no. 406, Artificial Intelligence Laboratory, Massachusetts Institute of Technology.

Castleman, B. (ed.). 1969. Case records of the Massachusetts General Hospital (Case 30-1969). *New England Journal of Medicine* 281: 206–213.

Catanzarite, V. A., and Greenburg, A. G. 1979. NEUROLOGIST: A computer program for diagnosis in neurology. In *Proceedings of the Third Annual Symposium on Computer Applications in Medical Care,* pp. 64–71. Long Beach, CA: IEEE Computer Society.

Chandrasekaran, B., Gomez, F., Mittal, S., and Smith, J. 1979. An approach to medical diagnosis based on conceptual structures. In *Proceedings of the Sixth International Joint Conference on Artificial Intelligence,* pp. 134–142. Stanford, CA: Stanford University, Department of Computer Science.

Charniak, E. 1978. With a spoon in hand this must be the eating frame. In *Proceedings of the Conference on Theoretical Issues in Natural Language Processing,* pp. 187–193. New York: ACM.

Chase, W. G., and Chi, M. T. H. 1980. Cognitive skill: Implications for spatial skill in large-scale environments. In *Cognition, Social Behavior, and the Environment,* ed. J. Harvey. Potomac, MD: Lawrence Erlbaum Associates.

Chase, W. G., and Simon, H. A. 1973. Perception in chess. *Cognitive Psychology* 4: 55–81.

Chi, M. T. H. 1976. Short-term memory limitations in children: Capacity or processing deficits? *Memory and Cognition* 4: 559–572.

———. 1978. Knowledge structures and memory development. In *Children's Thinking: What Develops?,* ed. R. S. Siegler. Hillsdale, NJ: Lawrence Erlbaum Associates.

Chi, M. T. H., Feltovich, P. J., and Glaser, R. 1981. Categorization and representation of physics problems by experts and novices. *Cognitive Science* 5: 121–152.

Chiese, H. L., Spilich, G. J., and Voss, J. F. 1979. Acquisition of domain-related information in relation to high and low domain knowledge. *Journal of Verbal Learning and Verbal Behavior* 18: 257–274.

Chilanski, R., Jacobsen, B., and Michalski, R. S. 1976. An application of variable-valued logic in inductive learning of plant disease diagnostic rules. In *Proceedings of the Third Illinois Conference on Medical Information Systems.*

Ciesielski, V. (ed.). 1978. *Proceedings of the Fourth Annual AIM Workshop.* Technical report, Rutgers University, New Brunswick, NJ.

Clancey, W. J. 1979a. Tutoring rules for guiding a case method dialogue. *International Journal of Man-Machine Studies* 11: 25–49.

———. 1979b. Transfer of rule-based expertise through a tutorial dialogue. Ph.D. dissertation, Stanford University. Computer Science Department, Report no. STAN-CS-769.

———. 1979c. Dialogue management for rule-based tutorials. In *Proceedings of the Sixth International Joint Conference on Artificial Intelligence,* pp. 155–161. Stanford, CA: Stanford University, Department of Computer Science.

———. 1983a. Communication, simulation, and intelligent agents: Implications of personal intelligent machines for medical education. In *Proceedings of AAMSI Congress 83,* pp. 556–560. Bethesda, MD: American Association for Medical Systems and Informatics.

———. 1983b. The epistemology of a rule-based expert system: A framework for explanation. *Artificial Intelligence* 20(3): 215–251.

———. 1983c. The advantages of abstract control knowledge in expert systems design. In *Proceedings of the Third National Conference on Artificial Intelligence,* pp. 74–78. Menlo Park, CA: SRI International.

———. 1984. Methodology for building an intelligent tutoring system. In *Methods and Tactics in Cognitive Science,* eds. Knitsch, Miller, and Polson. New York: Lawrence Erlbaum Associates.

Cohen, B. H. 1966. Some or none characteristics of coding behavior. *Journal of Verbal Learning and Verbal Behavior* 5: 182–187.

Colby, K. M., Parkison, R. C., and Faught, W. 1974. Pattern-matching rules for the recognition of natural language dialogue expressions. Report no. 234, Artificial Intelligence Laboratory, Stanford University.

Croft, D. J. 1972. Is computerized diagnosis possible? *Computers and Biomedical Research* 5: 351–367.

Crowder, N. A. 1962. Intrinsic and extrinsic programming. In *Proceedings of the Conference on Application of Digital Computers to Automated Instruction,* pp. 55–58. New York: Wiley.

Cumberbatch, J., and Heaps, H. S. 1976. A disease-conscious method for sequential diagnosis by use of disease probabilities without assumption of symptom independence. *International Journal of Biomedical Computing* 7: 61–78.

Dambrosia, J. M., and Ellenberg, J. H. 1980. Statistical considerations for a medical database. *Biometrics* 36: 323–332.

Davis, R. 1976. Applications of meta-level knowledge to the construction, maintenance and use of large knowledge bases. Ph.D. dissertation, Stanford University, Artificial Intelligence Laboratory. (Report nos. HPP-76-7 and AI-283.)

———. 1977. Interactive transfer of expertise I: Acquisition of new inference rules. In *Proceedings of the Fifth International Joint Conference on Artificial Intelligence,* pp. 321–328. Pittsburgh, PA: Carnegie-Mellon University, Department of Computer Science.

———. 1979. Interactive transfer of expertise: Acquisition of new inference rules. *Artificial Intelligence* 12: 121–158.

Davis, R., and Buchanan, B. G. 1977. Meta-level knowledge: Overview and applications. In *Proceedings of the Fifth International Joint Conference on Artificial Intelligence,* pp. 920–927. Pittsburgh, PA: Carnegie-Mellon University, Department of Computer Science.

Davis, R., and King, J. 1977. An overview of production systems. In *Machine Representations of Knowledge,* eds. E. W. Elcock and D. Michie. New York: Wiley.

Davis, R., and Lenat, D. B. 1982. *Knowledge-Based Systems in Artificial Intelligence.* New York: McGraw-Hill.

de Dombal, F. T. 1979. Acute abdominal pain: An O.M.G.E. survey. *Scandinavian Journal of Gastroenterology* 14(56): 30–43.

de Dombal, F. T., and F. Grémy (eds.). 1976. *Decision Making and Medical Care: Can Information Science Help?* Amsterdam: North-Holland.

de Dombal, F. T., Leaper, D. J., Staniland, J. R., McCann, A. P., and Horrocks, J. C. 1972. Computer-aided diagnosis of acute abdominal pain. *British Medical Journal* 2: 9–13.

de Dombal, F. T., Leaper, D. J., Horrocks, J. C., Staniland, J. R., and McCann, A. P. 1974. Human and computer aided diagnosis of abdominal pain: Further report with emphasis on the performance of clinicians. *British Medical Journal* 1: 376–380.

de Groot, A. D. 1965. *Thought and Choice in Chess.* New York: Basic Books.

Diamond, H. S., Weiner, M., and Plotz, C. M. 1974. A computer assisted instructional course in diagnosis and treatment of the rheumatic diseases. *Arthritis and Rheumatism* 17(6): 1049–1055.

Dito, W. 1977. An octal algorithm for pattern coding and computer-assisted interpretive reporting. *American Journal of Clinical Pathology* 68: 575–583.

Draper, N. R. 1966. *Applied Regression Analysis.* New York: Wiley.

Dreyfus, H. L. 1972. *What Computers Can't Do: A Critique of Artificial Reason.* New York: Harper & Row.

Duda, R. O., and Hart, P. E. 1973. *Pattern Classification and Scene Analysis.* New York: Wiley.

Duda, R. O., and Shortliffe, E. H. 1983. Expert systems research. *Science* 220: 261–268.

Duda, R. O., Hart, P. E., and Nilsson, N. J. 1976. Subjective Bayesian

methods for rule-based inference systems. In *Proceedings of the 1976 AFIPS National Computer Conference* 45: 1075–1082.

Duda, R. O., Hart, P. E., Nilsson, N. J., and Sutherland, G. L. 1977. Semantic network representations in rule-based inference systems. Report no. 136, SRI International, Menlo Park, CA.

Duda, R. O., Hart, P. E., Nilsson, N. J., and Sutherland, G. L. 1978. Semantic network representations in rule-based inference systems. In *Pattern-Directed Inference Systems,* eds. D. A. Waterman and F. Hayes-Roth. New York: Academic Press.

Edwards, R. 1968. Conservatism in human information processing. In *Formal Representation of Human Judgment,* ed. B. Kleinmuntz, pp. 17–52. New York: Wiley.

Edwards, W. 1972. N = 1: Diagnosis in unique cases. In *Computer Diagnosis and Diagnostic Methods,* ed. J. A. Jacquez, pp. 139–151. Springfield: Charles C Thomas.

Egan, D. E., and Schwartz, B. J. 1979. Chunking in recall of symbolic drawings. *Memory and Cognition* 7: 149–158.

Elstein, A. S. 1976. Clinical judgment: Psychological research and medical practice. *Science* 194: 696.

Elstein, A. S., and Shulman, L. S. 1971. A method for the study of medical thinking and problem solving. Paper presented at the meeting of the American Educational Research Association, New York.

Elstein, A. S., Loupe, M. J., and Erdman J. G. 1971. An experimental study of medical diagnostic thinking. *Journal of Structural Learning* 2: 45–53.

Elstein, A. S., Shulman, L. S., and Sprafka, S. A. 1978. *Medical Problem Solving: An Analysis of Clinical Reasoning.* Cambridge, MA: Harvard University Press.

Entwisle, G., and Entwisle, D. R. 1963. The use of digital computer as a teaching machine. *Journal of Medical Education* 38: 803–812.

Erman, L. D., and Lesser, V. R. 1975. A multi-level organization for problem solving using many diverse cooperating sources of knowledge. In *Proceedings of the Fourth International Joint Conference on Artificial Intelligence,* pp. 483–490. Cambridge, MA: M.I.T., AI Laboratory.

Fagan, L. M. 1979. Representation of dynamic clinical knowledge: measurement interpretation in the intensive care unit. In *Proceedings of the Sixth International Joint Conference on Artificial Intelligence,* pp. 260–262. Stanford, CA: Stanford University, Department of Computer Science.

———. 1980. VM: Representing time-dependent relations in a medical setting. Ph.D. dissertation, Stanford University (available from University Microfilms, #AAD80-24651).

Fagan, L. M., Kunz, J. C., Feigenbaum, E. A., and Osborn, J. J. 1979. A symbolic processing approach to measurement interpretation in the intensive care unit. In *Proceedings of the Third Annual Symposium on Computer Applications in Medical Care,* pp. 30–33. Long Beach, CA: IEEE Computer Society.

Fahlman, S. E. 1974. A planning system for robot construction tasks. *Artificial Intelligence* 5: 1–50.

Feigenbaum, E. A. 1963. Simulation of verbal learning behavior. In *Computers and Thought,* eds. E. A. Feigenbaum and J. A. Feldman, pp. 297–309. New York: McGraw-Hill.

———. 1977. The art of artificial intelligence: I. Themes and case studies of knowledge engineering. In *Proceedings of the Fifth International Joint Conference on Artificial Intelligence,* pp. 1014–1029. Pittsburgh, PA: Carnegie-Mellon University, Department of Computer Science.

Feigenbaum, E. A., and Feldman, J. (eds.). 1963. *Computers and Thought.* New York: McGraw-Hill.

Feinstein, A. R. 1967. *Clinical Judgment.* Baltimore: Williams and Wilkins.

———. 1970. Quality of data in the medical record. *Computers and Biomedical Research* 3: 426–435.

———. 1977a. Clinical biostatistics XXXVIII. Computer malpractice. *Clinical Pharmacology and Therapeutics* 21(1): 78–88.

———. 1977b. Clinical biostatistics XXXIX. The haze of Bayes, the aerial palaces of decision analysis, and the computerized Ouija board. *Clinical Pharmacology and Therapeutics* 21(4): 482–496.

Feinstein, A. R., Rubinstein, J. F., and Ramshaw, W. A. 1972. Estimating prognosis with the aid of a conversational mode computer program. *Annals of Internal Medicine* 76: 911–921.

Feurzeig, W., Munter, P., Swets, J., and Breen, M. 1964. Computer-aided teaching in medical diagnosis. *Journal of Medical Education* 39: 746–755.

Flehinger, B. J., and Engle, R. L., Jr. 1975. HEME: A self-improving computer program for diagnosis-oriented analysis of hematologic diseases. *IBM Journal of Research and Development* 19: 557–564.

Fodor, J. A. 1978. Tom Swift and his procedural grandmother. *Cognition* 6: 229–247.

Fox, J. 1977. Medical computing and the user. *International Journal of Man-Machine Studies* 9: 669–686.

Freiherr, G. 1979. The seeds of artificial intelligence: SUMEX-AIM. NIH publication 80-2071, Division of Research Resources, Washington, D.C.

Friedman, R. B., and Gustafson, D. H. 1977. Computers in clinical medicine: A critical review. *Computers and Biomedical Research* 8: 199–204.

Friedman, W. F., and Kirkpatrick, S. E. 1977. Congenital aortic stenosis. In *Heart Disorders in Infants, Children, and Adolescents,* eds. A. J. Moss, F. H. Adams, and G. C. Emmanouilides. Baltimore: Williams and Wilkins.

Fries, J. F. 1972. Time-oriented patient records and a computer databank. *Journal of the American Medical Association* 222: 1536–1542.

———. 1976. A data bank for the clinician? (editorial). *New England Journal of Medicine* 294: 1400–1402.

Fukunaga, K. 1972. *Introduction to Statistical Pattern Recognition.* New York: Academic Press.

Gaines, B. R. 1976. Foundations of fuzzy reasoning. *International Journal of Man-Machine Studies* 8: 623–668.

Garland, L. H. 1959. Studies on the accuracy of diagnostic procedures. *American Journal of Roentgenology* 82: 25–38.

Gaschnig, J. 1979. Preliminary performance analysis of the PROSPECTOR consultant system for mineral exploration. In *Proceedings of the Sixth International Joint Conference on Artificial Intelligence,* pp. 308–310. Stanford, CA: Stanford University, Department of Computer Science.

Gerring, P. E., Shortliffe, E. H., and van Melle, W. 1982. The Interviewer/Reasoner model: An approach to improving system responsiveness in interactive AI systems. *AI Magazine* 3(4): 24–27.

Gheorghe, A. V., Bali, H., and Carson, E. 1976. A Markovian decision model for clinical diagnosis and treatment applied to the respiratory system. *IEEE Transactions on Systems, Man and Cybernetics* SMC-6(9): 595.

Gill, P. W., Leaper, D. J., Guillou, P. J., Staniland, J. R., Horrocks, J. C., and de Dombal, F. T. 1973. Observer variation in clinical diagnosis—A computer-aided assessment of its magnitude and importance. *Methods of Information in Medicine* 12: 108–113.

Ginsberg, A. S. 1971. Decision analysis in clinical patient management with an application to the pleural effusion syndrome. Report no. R-751-RC/NLM, Rand Corporation, Santa Monica, CA.

———. 1972. The diagnostic process viewed as a decision problem. In *Computer Diagnosis and Diagnostic Methods,* ed. J. A. Jacquez. Springfield, IL: Charles C Thomas.

Glaser, R., and Pellegrino, J. W. 1980. Improving skills of learning. In *How, and How Much, Can Intelligence Be Increased?,* ed. D. K. Detterman. Norwood, NJ: Ablex.

Glesser, M. A., and Collen, M. F. 1972. Toward automated medical decisions. *Computers and Biomedical Research* 5: 180–189.

Goldstein, I. 1977. The computer as coach: An athletic paradigm for intellectual education. Report no. 389, Artificial Intelligence Laboratory, Massachusetts Institute of Technology.

———. 1978. Developing a computational representation of problem solving skills. Report no. 495, Artificial Intelligence Laboratory, Massachusetts Institute of Technology.

Goldstein, I. P., and Roberts, B. 1979. Using frames in scheduling. In *Artificial Intelligence: An M.I.T. Perspective.* Cambridge, MA: MIT Press.

Goldwyn, R. M., Friedman, H. P., and Siegel, J. H. 1971. Iteration and interaction in computer data bank analysis: A case study in the physiologic classification and assessment of the critically ill. *Computers and Biomedical Research* 4(6): 607–621.

Gorry, G. A. 1967. A system for computer-aided diagnosis. Report no. MAC-44, Project MAC, Massachusetts Institute of Technology.

———. 1968. Strategies for computer-aided diagnosis. *Mathematical Biosciences* 2: 293–318.

———. 1970. Modeling the diagnostic process. *Journal of Medical Education* 45: 293–302.

Gorry, G. A., and Barnett, G. O. 1968a. Experience with a model of sequential diagnosis. *Computers and Biomedical Research* 1: 490–507.

———. 1968b. Sequential diagnosis by computer. *Journal of the American Medical Association* 205: 849–854.

Gorry, G. A., Kassirer, J. P., Essig, A., and Schwartz, W. B. 1973. Decision analysis as the basis for computer-aided management of acute renal failure. *American Journal of Medicine* 55: 473–484.

Gorry, G. A., Silverman, H., and Pauker, S. G. 1978. Capturing clinical expertise: a computer program that considers clinical responses to digitalis. *American Journal of Medicine* 64: 452–460.

Green, C. C. 1969. The application of theorem proving to question-answering systems. Ph.D. dissertation, Department of Electrical Engineering, Stanford University (Stanford AI project memo AI-96).

Green, C. C., Waldingder, R. J., Barstow, D. R., Elschlager, R., Lenat, D. B., McCune, B. P., Shaw, D. E., and Steinberg, L. I. 1974. Progress report on program-understanding systems. Report no. CS-74-444, Computer Science Department, Stanford University.

Green, C. C., Gabriel, R. P., Kant E., Kedzierski, B. I., McCune, B. R., Phillips, J. V., Tappel, S. T., and Westfold, S. J. 1979. Results in knowledge based program synthesis. In *Proceedings of the Sixth Annual International Joint Conference on Artificial Intelligence,* pp. 342–344. Stanford, CA: Stanford University, Department of Computer Science.

Greenes, R. A., Barnett, G. O., Klein, S. W., Robbins, A., and Prior, R. E. 1970. Recording, retrieval, and review of medical data by physician-computer interaction. *New England Journal of Medicine* 282: 307–315.

Greenfield, S., Komaroff, A. L., and Anderson, H. 1976. A headache protocol for nurses: Effectiveness and efficiency. *Archives of Internal Medicine* 136: 1111–1116.

Greeno, J. G. 1979. Trends in the theory of knowledge for problem solving. In *Problem Solving and Education: Issues in Teaching and Research,* eds. D. T. Tuma and F. Reif. Hillsdale, NJ: Lawrence Erlbaum Associates.

Grémy, F. 1976. Decision-making and medical care: Can information science help? In *Proceedings of the IFIP Working Conference on Decision-Making in Medical Care,* eds. F. T. de Dombal and F. Grémy, p. 32. Amsterdam: North-Holland.

Grimm, R. H., Shimoni, K., Harlan, W. R., and Estes, E. H. 1975. Evaluation of patient-care protocol use by various providers. *New England Journal of Medicine* 292: 507–511.

Groner, G. F., Clark, R. L., Berman, R. A., and De Land, E. C. 1971. BIOMOD—An interactive computer graphics system for modeling. In *Proceedings of the Fall Joint Computer Conference,* pp. 369–378.

Groth, T. 1977. Biomedical modelling. In *MEDINFO 77,* pp. 775–784. Amsterdam: North-Holland.

Harless, W. G., Crennon, G. G., Marxer, J. J., Root, J. A., and Miller, G. E. 1971. CASE: A computer-aided simulation of the clinical encounter. *Journal of Medical Education* 46: 443–448.

Hart, P. E. 1975. Progress on a computer-based consultant. Report no. 99, AI Group, SRI International, Menlo Park, CA.

Hart, P. E., and Duda, R. E. 1977. PROSPECTOR—A computer-based consultation system for mineral exploration. Report no. 155, SRI International, Menlo Park, CA.

Harvey, A. M., and Bordley, J., III. 1972. *Differential Diagnosis.* Philadelphia: Saunders.

Hasling, D. W., Clancey, W. J., and Rennels, G. 1984. Strategic explanations for a diagnostic consulting system. *International Journal of Man-Machine Studies* 20(1): 3–19.

Hayes-Roth, F. 1978. The role of partial and best matches in knowledge systems. In *Pattern-Directed Inference Systems,* eds. F. Hayes-Roth and D. A. Waterman. New York: Academic Press.

Heiser, J. F., and Brooks, R. E. 1978. Design considerations for a clinical psychopharmacology advisor. In *Proceedings of the Second Annual Symposium on Computer Applications in Medical Care,* pp. 278–285. Long Beach, CA: IEEE Computer Society.

Hendrix, G. 1975. Expanding the utility of semantic networks through partitioning. In *Proceedings of the Fourth International Joint Conference on Artificial Intelligence,* pp. 115–121. Cambridge, MA: M.I.T., AI Laboratory.

Hess, E. V. 1976. A uniform database for rheumatic diseases. *Arthritis and Rheumatism* 19: 645–648.

Hewitt, C. 1972. Description and theoretical analysis (using schemata) of PLANNER: A language for proving theorems and manipulating models in a robot. Ph.D. dissertation, Massachusetts Institute of Technology.

Hinsley, D. A., Hayes, J. R., and Simon, H. A. 1978. From words to equations: Meaning and representation in algebra word problems. In *Cognitive Processes in Comprehension,* eds. M. A. Just and P. A. Carpenter. Hillsdale, NJ: Lawrence Erlbaum Associates.

Hoffer, E., Barnett, O., Farquar, B. B., and Prather, P. A. 1975. Computer-aided instruction in medicine. *Annual Review of Biophysical Engineering* 4: 103–118.

Horrocks, J. C., and de Dombal, F. T. 1975. Computer-aided diagnosis of dyspepsia. *American Journal of Digestive Diseases* 20: 397–406.

Horrocks, J. C., McCann, A. P., Staniland, J. R., Leaper, D. J., and de Dombal, F. T. 1972. Computer-aided diagnosis: Description of an adaptable system, and operational experience with 2,034 cases. *British Medical Journal* 2: 5–9.

Howard, R. A. 1968. Foundations of decision analysis. (Special issue on

decision analysis.) *IEEE Transactions on Systems Science and Cybernetics* 4(3): 211–219.

Hurst, J. W. 1974. *The Heart, Arteries, and Veins.* New York: McGraw-Hill.

Inglefinger, F. J. 1975. Decision in medicine (editorial). *New England Journal of Medicine* 293: 254–255.

Isselbacher, K. J., Adams, R. D., Braunwald, E., Petersdorf, R. G., and Wilson, J. D. 1980. *Harrison's Principles of Internal Medicine.* New York: McGraw-Hill.

Jacquez, J. A. 1972. *Computer Diagnosis and Diagnostic Methods.* Springfield, IL: Charles C Thomas.

Jelliffe, R. W., and Jelliffe, S. M. 1972. A computer program for estimation of creatinine clearance from unstable serum creatinine levels, age, sex, and weight. *Mathematical Biosciences* 14: 17–24.

Jelliffe, R. W., Buell, J., Kalaba, R., Sridhar, R., and Rockwell, R. 1970. A computer program for digitalis dosage regimens. *Mathematical Biosciences* 9: 179–193.

Jelliffe, R. W., Buell, J., and Kalaba, R. 1972. Reduction of digitalis toxicity by computer-assisted glycoside dosage regimens. *Annals of Internal Medicine* 77: 891–906.

Johnson, D. C., and Barnett, G. O. 1977. MEDINFO—A medical information system. *Computer Programs in Biomedicine* 7: 191–201.

Johnson, P. E., Severance, D. G., and Feltovich, P. J. 1979a. Design of decision support systems in medicine: Rationale and principles from the analysis of physician expertise. In *Proceedings of the Twelfth Hawaii International Conference on Systems Sciences.*

Johnson, P. E., Feltovich, P. J., Moller, J. H., and Swanson, D. B. 1979b. Clinical expertise: Theory and data from the diagnosis of congenital heart disease. Paper presented at the 1979 meeting of the American Educational Research Association, April, San Francisco.

Johnson, P. E., Duran, A. S., Hassebrock, F., Moller, J., Prietula, M., Feltovich, P. J., and Swanson, D. B. 1981. Expertise and error in diagnostic reasoning. *Cognitive Science* 5(3): 235–284.

Kagan, B. M., Fannin, S. L., and Bardie, F. 1973. Spotlight on antimicrobial agents—1973. *Journal of the American Medical Association* 226: 306–310.

Kak, A. C. 1979. Computerized tomography with X-ray, emission and ultrasound sources. *Proceedings of the IEEE* 16: 1245–1271.

Kanal, L. N. 1974. Patterns in pattern recognition: 1968–1974. *IEEE Transactions on Information Theory* IT-20(6).

Kaplan, R. M., Sheil, B. A., and Smith, E. R. 1978. The interactive data-analysis language reference manual. XEROX Palo Alto Research Center, Palo Alto, CA.

Karpinski, R. H. S., and Bleich, H. L. 1971. MISAR: A miniature information storage and retrieval system. *Computers and Biomedical Research* 4: 655–660.

Kassirer, J. P., and Gorry, G. A. 1978. Clinical problem solving: A behavioral analysis. *Annals of Internal Medicine* 89: 245–255.

Kenny, D. 1970. *Correlation and Causality.* Amsterdam: North-Holland.

Kingsland, L. C., and Lindberg, D. 1983. Research methods in AI model building: The history of a project. In *Proceedings of AAMSI Congress 83,* pp. 76–80.

Kirsch, A. D. 1963. A medical training game using a computer as a teaching aid. *Methods of Information in Medicine* 2(4): 138–143.

Kleinmuntz, B., and McLean, R. S. 1968. Diagnostic interviewing by digital computer. *Behavioral Science* 13: 75–80.

Knapp, R. G., Levi, S., Lurie, D., and Westphal, M. 1977. A computer-generated diagnostic decision guide: A comparison of statistical diagnosis and clinical diagnosis. *Computers in Biology and Medicine* 7: 223–230.

Koffman, E. B., and Blount, S. E. 1973. Artificial intelligence and automatic programming in CAI. In *Proceedings of the Third International Joint Conference on Artificial Intelligence,* pp. 86–94. Menlo Park, CA: SRI International.

Komaroff, A. L. 1979. The variability and inaccuracy of medical data. *Proceedings of the IEEE* 16: 1196–1207.

Komaroff, A. L., Black, W. L., and Flatley, M. 1974. Protocols for physician assistants: Management of diabetes and hypertension. *New England Journal of Medicine* 290: 307–312.

Korein, J., Lyman, M., and Tick, J. L. 1971. The computerized medical record. *Bulletin New York Academy of Medicine* 47: 824–826.

Koss, N., and Feinstein, A. R. 1971. Computer-aided prognosis: II. Development of a prognostic algorithm. *Archives of Internal Medicine* 127: 448–459.

Kuipers, B., and Kassirer, J. P. 1983. Causal reasoning in medicine: Analysis of a protocol. *Cognitive Science:* forthcoming.

Kulikowski, C. A. 1970. Pattern recognition approach to medical diagnosis. *IEEE Transactions on Systems Science and Cybernetics* SSC-6: 83–89.

Kulikowski, C., and Weiss, S. 1971. Computer-based models of glaucoma. Report no. 3, Department of Computer Science, Computers in Biomedicine, Rutgers University.

———. 1972. The medical consultant program-glaucoma. Report no. 5, Rutgers University.

———. 1973. An interactive facility for the inferential modeling of disease. In *Proceedings of 1973 Princeton Conference on Information Sciences and Systems,* p. 524.

———. 1982. Representation of expert knowledge for consultation: The CASNET and EXPERT projects. In *Artificial Intelligence in Medicine,* ed. P. Szolovits, pp. 21–56. Boulder, CO: Westview Press.

Kulikowski, C. A., Weiss, S., and Safir, A. 1973. Glaucoma diagnosis and therapy by computer. In *Proceedings of the Annual Meeting of Association for Research in Vision and Opthalmology.*

Kunz, J. C., Fallat, R. J., McClung, D. H., Osborn, J. J., Votteri, B. A., Nii, H. P., Aikins, J. S., Fagan, L. M., and Feigenbaum, E. A. 1978. A

physiological rule-based system for interpreting pulmonary function test results. Report no. HPP-78-19, Heuristic Programming Project, Stanford University.

Langlotz, C. P., and Shortliffe, E. H. 1983. Adapting a consultation system to critique user plans. *International Journal of Man-Machine Studies* 19: 479–496.

Larkin, J. H. 1978. Skilled problem solving in physics: A hierarchical planning model. Unpublished manuscript, University of California, Berkeley.

Leaper, D. J., Horrocks, J. C., Staniland, J. R., and de Dombal, F. T. 1972. Computer-assisted diagnosis of abdominal pain using estimates provided by clinicians. *British Medical Journal* 4: 350–354.

Ledley, R. S., and Lusted, L. B. 1959. Reasoning foundations of medical diagnosis. *Science* 130: 9–21.

Le Faivre, R. A. 1974. *Fuzzy problem solving.* Report no. 37, University of Wisconsin, Madison.

Lehnert, W., and Wilks, Y. 1979. A critical perspective on KRL. *Cognitive Science* 3: 1–28.

Lenat, D. B. 1975. BEINGS: Knowledge as interacting experts. In *Proceedings of the Fourth International Joint Conference on Artificial Intelligence,* pp. 126–133. Cambridge, MA: M.I.T., AI Laboratory.

Lenat, D. B., and Harris, G. 1978. Designing a rule system that searches for scientific discoveries. In *Pattern-Directed Inference Systems,* eds. F. Hayes-Roth and D. A. Waterman. New York: Academic Press.

Lenat, D. B., Hayes-Roth, F., and Klahr, P. 1979. *Cognitive economy.* Report no. N-1185-NSF, Rand Corporation, Santa Monica, CA.

Lesser, V. R., and Erman, L. D. 1979. A retrospective view of the HEARSAY-II architecture. In *Proceedings of the Fifth International Joint Conference on Artificial Intelligence,* pp. 790–800. Pittsburgh, PA: Carnegie-Mellon University, Department of Computer Science.

Lesser, V. R., Fennell, R. D., Erman, L. D., and Reddy, D. R. 1975. Organization of the HEARSAY-II speech understanding system. In *IEEE Transactions on Acoustics, Speech, and Signal Processing* 12: 11–23.

Levi, S., Frant, J. R., Westphal, M. C., and Lurie, D. 1976. Development of a decision guide—Optimal discriminations for meningitis determined by statistical analysis. *Methods of Information in Medicine* 15(2): 87–90.

Lichter, P., and Anderson, D. 1977. *Discussions on Glaucoma.* New York: Grune and Stratton.

Lindberg, D. 1977. *The Growth of Medical Information Systems in the United States.* Lexington, MA: Lexington Books.

Lindberg, D., Sharp, G., Kingsland, L., Weiss, S., Hayes, S., Ueno, H., and Hazelwood, S. 1980. Computer based rheumatology consultant. In *Proceedings of MEDINFO—International Conference on Medical Informatics,* pp. 1311–1315.

Lindsay, R. K., Buchanan, B. G., Feigenbaum, E. A., and Lederberg, J.

1980. *Applications of Artificial Intelligence for Organic Chemistry: The DENDRAL Project.* New York: McGraw-Hill.

Lipkin, M., and Hardy, J. D. 1958. Mechanical correlation of data in differential diagnosis of hematologic diseases. *Journal of the American Medical Association* 166: 113–125.

London, B., and Clancey, W. J. 1982. Plan recognition strategies in student modelling: Prediction and description. In *Proceedings of the Second National Conference on Artificial Intelligence,* pp. 335–338. Los Altos, CA: Kaufmann.

Long, W. J. 1977. A program writer. Report no. TR-187, Laboratory for Computer Science, Massachusetts Institute of Technology.

Lucas, R. V., and Schmidt, R. E. 1977. Anomalous venous connections, pulmonary and systemic. In *Heart Disease in Infants, Children, and Adolescents,* eds. A. J. Moss, F. H. Adams, and G. C. Emmanouilides, pp. 417–470. Baltimore: Williams and Wilkins.

Lusted, L. B. 1968. *Introduction to Medical Decision Making.* Springfield, IL: Charles C Thomas.

———. 1981. A society and a journal. *Medical Decision Making* 1: 7–9.

Mabry, J. C., Thompson, H. K., Hopwood, M. D., and Baker, W. R. 1977. A prototype data management and analysis system (CLINFO): System description and user experience. In *MEDINFO 77,* pp. 71–75. Amsterdam: North-Holland.

Manna, Z. 1969. Correctness of programs. *Journal of Computer and System Sciences.*

Manna, Z., and Waldinger, R. 1977. The automatic synthesis of systems of recursive programs. In *Proceedings of the Fifth International Conference on Artificial Intelligence,* pp. 405–411. Pittsburgh, PA: Carnegie-Mellon University, Department of Computer Science.

MATHLAB Group. 1974. *The MACSYMA Reference Manual.* Cambridge, MA: Massachusetts Institute of Technology.

McCarthy, J. 1968. Programs with common sense. In *Semantic Information Processing,* ed. M. Minsky, pp. 403–418. Cambridge, MA: MIT Press.

McCarthy, J., and Hayes, P. J. 1969. Some philosophical problems from a standpoint of artificial intelligence. In *Machine Intelligence 4,* eds. B. Meltzer and D. Michie. New York: Elsevier.

McDonald, C., Bhargava, B., and Jeris, D. 1975. A clinical information system (CIS) for ambulatory care. In *Proceedings of the 1975 National Computer Conference,* pp. 749–756.

McGuire, C., and Bashook, P. 1978. A conceptual framework for measuring clinical problem solving. Paper presented at the 1978 Annual Meeting of the American Educational Research Association.

McGuire, C. H., and Solomon, L. 1971. *Clinical Simulation.* New York: Appleton-Century-Crofts.

McNeil, B. J., and Adelstein, S. J. 1977. Determining the value of diagnostic and screening tests. *Journal of Nuclear Medicine* 17: 439–448.

McNeil, B. J., Keeler, E., and Adelstein, S. J. 1975. Primer on certain

elements of medical decision making. *New England Journal of Medicine* 293: 211–215.

Melton, A. W. 1963. Implications of short-term memory for a general theory of memory. *Journal of Verbal Learning and Verbal Behavior* 2: 1–21.

Menn, S. J., Barnett, G. O., Schmechel, D., Owens, W. D., and Pontoppidan, H. 1973. A computer program to assist in the care of acute respiratory failure. *Journal of the American Medical Association* 223: 308–312.

Mesel, E., Wirtschafter, D. D., Carpenter, J. T., Durant, J. R., Hencke, C., and Gray, E. A. 1976. Clinical algorithms for cancer chemotherapy—Systems for community-based consultant-extenders and oncology centers. *Methods of Information in Medicine* 15: 168–173.

Meyer, A. V., and Weissman, W. K. 1973. Computer analysis of the clinical neurological exam. *Computers and Biomedical Research* 3: 111–117.

Michie, D. 1974. *On Machine Intelligence.* Edinburgh, U.K.: Edinburgh University Press.

Miller, P. B. 1975. Strategy selection in medical diagnosis, Project MAC. Report no. TR-153, Massachusetts Institute of Technology.

Minsky, M. 1968. *Semantic Information Processing.* Cambridge, MA: MIT Press.

———. 1975. A framework for representing knowledge. In *The Psychology of Computer Vision,* ed. P. H. Winston, pp. 211–277. New York: McGraw-Hill.

———. 1979. The society theory of thinking. In *Artificial Intelligence: An M.I.T. Perspective.* Cambridge, MA: MIT Press.

Mitchell, T. 1979. An analysis of generalization as a search problem. In *Proceedings of the Sixth International Joint Conference on Artificial Intelligence,* pp. 577–582. Stanford, CA: Stanford University, Department of Computer Science.

Mittal, S., Chandrasekaran, B., and Smith, J. 1979. Overview of MDX—A system for medical diagnosis. In *Proceedings of the Third Annual Symposium on Computer Applications in Medical Care,* pp. 34–46. Long Beach, CA: IEEE Computer Society.

Moller, J. H. 1978. *Essentials of Pediatric Cardiology.* Philadelphia: A. F. Davis.

Mood, A. M., Graybill, F. A., and Boes, D. C. 1974. *Introduction to the Theory of Statistics.* New York: McGraw-Hill.

Moss, A. J., Adams, F. H., and Emmanouilides, G. C. 1977. *Heart Disease in Infants, Children, and Adolescents.* Baltimore, MD: Williams and Wilkins.

Myers, J. D., Pople, H. E., and Miller, R. A. 1982. INTERNIST: Can artificial intelligence help? In *Clinical Decisions and Laboratory Use,* eds. Connelly, Benson, Burke, and Fenderson, pp. 251–269. Minneapolis, MN: University of Minnesota Press.

Nash, F. A. 1954. Differential diagnosis: An apparatus to assist the logical faculties. *Lancet* 266: 874.

Newell, A. 1969. Heuristic programming: Ill-structured problems. In *Progress in Operations Research,* ed. J. Aronofsky, pp. 360–414. New York: Wiley.

Newell, A., and Simon, H. A. 1972. *Human Problem Solving.* Englewood Cliffs: Prentice-Hall.

Nie, N. H., Hull, C. H., Jenkins, J. C., Steinbrenner, K., and Bent, D. H. 1975. *SPSS: Statistical Package for the Social Sciences.* New York: McGraw-Hill.

Nii, H. P., and Aiello, N. 1979. AGE (attempt to generalize): A knowledge-based program for building knowledge-based programs. In *Proceedings of the Sixth International Joint Conference on Artificial Intelligence,* pp. 645–655. Stanford, CA: Stanford University, Department of Computer Science.

Nilsson, N. J. (ed.). 1975. Artificial intelligence—Research and applications. AI Group progress report, SRI International, Menlo Park, CA.

———. 1980. *Principles of Artificial Intelligence.* Palo Alto, CA: Tioga Press.

Nordyke, R. A., Kulikowski, C. A., and Kulikowski, C. W. 1971. A comparison of methods for the automated diagnosis of thyroid dysfunction. *Computers and Biomedical Research* 4: 374–389.

Norman, D. A., Rumelhart, D. E., and LNR Research Group. 1975. *Explorations in Cognition.* San Francisco: Freeman.

Norusis, M. J., and Jacquez, J. A. 1975. Diagnosis I. Symptom nonindependence in mathematical models for diagnosis. *Computers and Biomedical Research* 8: 156–172.

Oleson, C. 1977. INTERNIST: A computer-based consultation. In *Computer Networking in the University: Success and Potential.*

Osborn, J. J., Beaumont, J. C., Raison, A., and Abbott, R. P. 1969. Computation for quantitative on-line measurement in an intensive care ward. In *Computers in Biomedical Research,* eds. R. W. Stacey and B. D. Waxman, pp. 207–237. New York: Academic Press.

Paige, J. M., and Simon, H. A. 1966. Cognitive processes in solving algebra word problems. In *Problem Solving,* ed. B. Kleinmutz. New York: Wiley.

Papert, S. 1970. Teaching children programming. In *Proceedings of the IFIP Conference on Computer Education.* Amsterdam: North-Holland.

Patil, R. S. 1979. Design of a program for expert diagnosis of acid base and electrolyte disturbances. Report no. TM-132, Laboratory for Computer Science, Massachusetts Institute of Technology.

Patil, R. S., Szolovits, P., and Schwartz, W. B. 1982. Information aquisition in diagnosis. In *Proceedings of the National Conference on Artificial Intelligence,* pp. 345–348. Pittsburgh, PA: Carnegie-Mellon.

Patrick, E. A. 1977. Pattern recognition in medicine. *IEEE Transactions on Systems, Man and Cybernetics* SMC-6: 4.

Patrick, E. A., Stelmack, F. P., and Shen, L. Y. L. 1974. Review of pattern recognition in medical diagnosis and consulting relative to a new sys-

tem model. *IEEE Transactions on Systems, Man and Cybernetics* SMC-4(1): 1.

Pauker, S. G. 1976. Coronary artery surgery: The use of decision analysis. *Annals of Internal Medicine* 85: 8–18.

———. 1978. The acquisition and use of patient attitudes in clinical decision-making. In *Proceedings of the Fourth Illinois Conference on Medical Information Systems,* p. 130.

Pauker, S. G., and Kassirer, J. P. 1975. Therapeutic decision making: A cost-benefit analysis. *New England Journal of Medicine* 293: 229–234.

Pauker, S. P., and Pauker, S. G. 1977. Prenatal diagnosis: A directive approach to genetic counseling using decision analysis. *Yale Journal of Biology and Medicine* 50: 275–289.

Pauker, S. G., and Szolovits, P. 1977. Analyzing and simulating taking the history of the present illness: Context formation. In *Computational Linguistics in Medicine,* eds. Schneider and Sagvall-Hein, pp. 109–118. Amsterdam: North-Holland.

Pauker S. G., Gorry G. A., Schwartz, W. B., and Kassirer, J. P. 1976. Towards the simulation of clinical cognition. *American Journal of Medicine* 60: 981–996.

Peck, C. C., Sheiner, L. B., Martin, C. M., Combs, D. T., and Melmon, K. L. 1973. Computer-assisted digoxin therapy. *New England Journal of Medicine* 289: 441–446.

Perlman, F., McCue, J. D., and Friedland, G. 1974. Urinary tract infection (UTI)/vaginitis protocol, introduction. Ambulatory Care Project, Lincoln Laboratory, Massachusetts Institute of Technology, and Beth Israel Hospital, Harvard Medical School.

Peters, R. J. 1976. Zero order and non-zero order decision rules in medical diagnosis. Report no. RC 6088, IBM, Thomas J. Watson Research Center.

Piaget, J. 1972. *The Psychology of Intelligence.* Totowa, NJ: Littlefield, Adams.

Pipberger, H. V., McCaughn, D., Littman, D., Pipberger, H. A., Cornfield, J., Dunn, R. A., Batchlor, C. D., and Berson, A. S. 1975. Clinical application of a second generation electrocardiographic computer program. *American Journal of Cardiology* 35: 597–608.

Pliskin, J. S., and Beck, C. H. 1976. Decision analysis in individual clinical decision making: A real-world application in treatment of renal disease. *Methods of Information in Medicine* 15: 43–46.

Politakis, P. 1982. Using empirical analysis to refine expert system knowledge bases. Ph.D. dissertation, Rutgers University.

Pople, H. 1973. On the mechanization of abductive logic. In *Proceedings of the Third International Joint Conference on Artificial Intelligence,* pp. 147–152. Menlo Park, CA: SRI International.

———. 1975. Artificial intelligence approaches to computer-based medical consultation. In *Proceedings of the IEEE, INTERCON* 31(3).

———. 1976. Presentation of the INTERNIST system. In *Proceedings of the AIM Workshop.*

———. 1977. The formation of composite hypotheses in diagnostic problem solving: An exercise in synthetic reasoning. In *Proceedings of the Fifth International Joint Conference on Artificial Intelligence,* pp. 1030–1037. Pittsburgh, PA: Carnegie-Mellon University, Department of Computer Science.

———. 1982. Heuristic methods for imposing structure on ill-structured problems: The structuring of medical diagnosis. In *Artificial Intelligence in Medicine,* ed. P. Szolovits, pp. 119–190. Boulder, CO: Westview Press.

Pople, H., and Werner, G. 1972. An information processing approach to theory formation in biomedical research. In *Proceedings of the AFIPS Spring Joint Computer Conference.*

Pople, H. E., Myers, J. D., and Miller, R. A. 1975. DIALOG: A model of diagnostic logic for internal medicine. In *Proceedings of the Fourth International Joint Conference on Artificial Intelligence,* pp. 848–855. Cambridge, MA: M.I.T., AI Laboratory.

Prutting, J. 1967. Lack of correlation between antemortem and postmortem diagnosis. *New York Journal of Medicine* 67: 2081–2084.

Pryor, T. A., Gardner, R. M., Clayton, P. D., and Warner, H. R. 1982. The HELP system. In *Proceedings of the Sixth Symposium on Computer Applications in Medical Care,* pp. 19–26.

Quillian, M. R. 1968. Semantic memory. In *Semantic Information Processing,* ed. M. Minsky, pp. 216–270. Cambridge, MA: MIT Press.

———. 1969. The teachable language comprehender: A simulation program and theory of language. *Communications of the ACM* 12: 459–476.

Raiffa, H. 1968. *Decision Analysis: Introductory Lectures on Choices Under Uncertainty.* Reading, MA: Addison-Wesley.

Reddy, D. R. 1977. Speech recognition by machine: A review. *Proceedings of IEEE* 64(4): 501.

Reed, S. K. 1978. Category vs. item learning: Implications for categorizational models. *Memory and Cognition* 6: 616–621.

Reggia, J. A. 1978. A production rule system for neurological localization. In *Proceedings of the Second Annual Symposium on Computer Applications in Medical Care,* pp. 254–260. Long Beach, CA: IEEE Computer Society.

Richards, B., and Goh, A. E. S. 1977. Computer assistance in the treatment of patients with acid-base and electrolyte disturbances. In *MEDINFO 77,* pp. 407–410. Amsterdam: North-Holland.

Rieger, C. 1975. The commonsense algorithm as a basis for computer models of human memory, inference, belief and contextual language comprehension. In *Proceedings of a Workshop on Theoretical Issues in Natural Language Processing,* eds. R. Schank and B. L. Nash-Webber.

Rieger, C., and Small, S. 1981. Towards a theory of distributed word expert natural language parsing. *IEEE Transactions on Systems, Man and Cybernetics* SMC-11(1): 43–51.

Riesbeck, C. K. 1978. An expectation-driven production system for natural language understanding. In *Pattern-Directed Inference Systems,* eds. D. A. Waterman and F. Hayes-Roth, pp. 399–414. New York: Academic Press.

Rodnick, J., and Wiederhold, G. 1977. Review of automated ambulatory medical record systems: Charting services that are of essential benefit to the physician. In *MEDINFO 77,* pp. 957–961. Amsterdam: North-Holland.

Rosati, R. A., Wallace, A. G., and Stead, E. A. 1973. The way of the future. *Archives of Internal Medicine* 131: 285–287.

Rosati, R. A., McNeer, J. F., Starmer, C. F., Mittler, B. S., Morris, J. J., and Wallace, G. 1975. A new information system for medical practice. *Archives of Internal Medicine* 135: 1017–1024.

Rosch, E., and Mervis, C. B. 1975. Family resemblances: Studies in the internal structure of categories. *Cognitive Psychology* 7: 573–605.

Rosch, E., Mervis, C. B., Gray, W., Johnson, D., and Boyes-Braem, P. 1976. Basic objects in natural categories. *Cognitive Psychology* 8: 382–439.

Rosenblatt, M. B., Teng, P. K., and Kerpe, S. 1973. Diagnostic accuracy in cancer as determined by post-mortem examination. *Progress in Clinical Cancer* 5: 71–80.

Rubin, A. D. 1975. Hypothesis formation and evaluation in medical diagnosis. Report no. AI-TR-316, Artificial Intelligence Laboratory, Massachusetts Institute of Technology.

Rubin, A. D., and Risley, J. F. 1977. The PROPHET system: An experiment in providing a computer resource to scientists. In *MEDINFO 77,* pp. 77–81. Amsterdam: North-Holland.

Rumelhart, D. E. 1979. Analogical processes and procedural representations. Report no. CHIP-81, University of California, San Diego.

Rumelhart, D. E., and Norman, D. A. 1977. Accretion, tuning, and restructuring: Three models of learning. In *Semantic Factors in Cognition,* eds. R. Klatsky and J. W. Cotton. Hillsdale, NJ: Lawrence Erlbaum Associates.

———. 1980. Analogical processes in learning. Report no. 8005, University of California, San Diego.

Rumelhart, D. E., and Ortony, A. 1977. The representation of knowledge in memory. In *Schooling and the Acquisition of Knowledge,* eds. R. C. Anderson, R. J. Spiro, and W. E. Montague. Hillsdale, NJ: Lawrence Erlbaum Associates.

Sacerdoti, E. D. 1977. *A Structure for Plans and Behavior.* New York: Elsevier.

Safran, C., Tsichlis, P. N., Bluming, A. Z., and Desforges, J. F. 1977. Diagnostic planning using computer-assisted decision making for patients with Hodgkin's disease. *Cancer* 39: 2426–2434.

Schank, R. C., and Abelson, R. P. 1977. *Scripts, Plans, Goals, and Understanding.* Hillsdale, NJ: Lawrence Erlbaum Associates.

Schank, R. C., and Colby, K. M. (eds.). 1973. *Computer Models of Thought and Language.* San Francisco: W. H. Freeman.

Schoolman, H., and Bernstein, L. 1978. Computer use in diagnosis, prognosis, and therapy. *Science* 200: 926–931.

Schultz, J. R., and Davis, L. 1979. The technology of PROMIS. *Proceedings of the IEEE* 16: 1237–1244.

Schwartz, W. B. 1970. Medicine and the computer: The promise and problems of change. *New England Journal of Medicine* 283: 1257–1264.

Schwartz, W. B., Gorry, G. A., Kassirer, J. P., and Essig, A. 1973. Decision analysis and clinical judgment. *American Journal of Medicine* 55: 459–472.

Scott, A. C., Clancey, W. J., Davis, R., and Shortliffe, E. H. 1977. Explanation capabilities of knowledge-based production systems. *American Journal of Computational Linguistics,* microfiche 62.

Shafer, G. 1976. *A Mathematical Theory of Evidence.* Princeton, NJ: Princeton University Press.

Shannon, C. E., and Weaver, W. 1949. *The Mathematical Theory of Communication.* Urbana: University of Illinois Press.

Shavelson, R. J. 1972. Some aspects of the correspondence between content structure and cognitive structure in physics instruction. *Journal of Educational Philosophy* 63: 225–234.

Sheiner, L. B., Rosenberg, B., and Melmon, K. L. 1972. Modeling of individual pharmacokinetics for computer-aided drug dosage. *Computers and Biomedical Research* 5: 441–459.

Sheiner, L. B., Halkin, H., Peck, C., Rosenberg, B., and Melmon, K. L. 1975. Improved computer-assisted digoxin therapy. *Annals of Internal Medicine* 82: 619–627.

Sherman, H., Reiffen, B., and Komaroff, A. L. 1973. Ambulatory care systems. In *Problem-Directed and Medical Information Systems,* ed. M. F. Driggs, pp. 143–171. New York: Intercontinental Medical Book Corporation.

Shortliffe, E. H. 1976. *Computer-Based Medical Consultations: MYCIN.* New York: American Elsevier.

———. 1980. The computer as a clinical consultant. *Archives of Internal Medicine* 140: 313–314.

———. 1982a. Computer-based clinical decision aids: Some practical considerations. In *Proceedings of the AMIA Congress 82,* eds. Lindberg, Collen, and van Brunt, pp. 295–298. New York: Masson Publishing.

———. 1982b. The computer and medical decision making: Good advice is not enough. *IEEE Engineering in Medicine and Biology Magazine* 1(2): 16–18.

———. 1983. The science of biomedical computing. In *Meeting the Challenge: Informatics and Medical Education,* eds. J. C. Pages, A. H. Levy, F. Gremy, and J. Anderson, pp. 1–10. Amsterdam: North-Holland.

Shortliffe, E. H., and Buchanan, B. G. 1975. A model of inexact reasoning in medicine. *Mathematical Biosciences* 23: 351–379.

———. 1980. Artificial intelligence (letter). *New England Journal of Medicine* 302: 1482.

Shortliffe, E. H., and Davis, R. 1975. Some considerations for the implementation of knowledge-based expert systems. *SIGART Newsletter* 55: 9–12.

Shortliffe, E. H., Axline, S. G., Buchanan, B. G., Merigan, T. C., and Cohen, S. N. 1973. An artificial intelligence program to advise physicians regarding antimicrobial therapy. *Computers and Biomedical Research* 6: 544–560.

Shortliffe, E. H., Axline, S. G., Buchanan, B. G., and Cohen, S. N. 1974. Design considerations for a program to provide consultations in clinical therapeutics. In *Proceedings of the Thirteenth San Diego Biomedical Symposium,* pp. 311–319. San Diego, CA: San Diego Biomedical Symposium.

Shortliffe, E. H., Davis, R., Axline, S. G., Buchanan, B. G., Green, C. C., and Cohen, S. N. 1975. Computer-based consultations in clinical therapeutics—Explanation and rule acquisition capabilities of the MYCIN system. *Computers and Biomedical Research* 8: 303–320.

Shortliffe, E. H., Scott, A. C., Bischoff, M., Campbell, A. B., van Melle, W., and Jacobs, C. 1981. ONCOCIN: An expert system for oncology protocol management. In *Proceedings of the Seventh International Joint Conference on Artificial Intelligence,* pp. 876–881. Menlo Park, CA: AAAI.

Siegler, R. S. 1976. Three aspects of cognitive development. *Cognitive Psychology* 8: 481–520.

———. 1978. The origins of scientific reasoning. In *Children's Thinking: What Develops?,* ed. R. S. Siegler. Hillsdale, NJ: Lawrence Erlbaum Associates.

Siklossy, L., and Roach, J. 1973. Proving the impossible is impossible is possible. In *Proceedings of the Third International Joint Conference on Artificial Intelligence,* pp. 383–387. Menlo Park, CA: SRI International.

Silverman, H. 1975. A digitalis therapy advisor. Report no. TR-143, Project MAC, Massachusetts Institute of Technology.

Simon, H. A. 1969. *The Sciences of the Artificial.* Cambridge, MA: MIT Press.

Simon, H. A., and Chase, W. G. 1973. Skill in chess. *American Scientist* 61: 394–403.

Slack, W. V., Hicks, G. P., Reed, C. E., and VanCura, L. J. 1966. A computer-based medical history system. *New England Journal of Medicine* 274: 194.

Slamecka, V., Camp, H. N., Badre, A. N., and Hall, W. D. 1977. MARIS: A knowledge system for internal medicine. *Information Processing and Man* 13: 273–276.

Smith, B. C. 1978. A computational model of anatomy and physiology: A model. Report no. 493, Artificial Intelligence Laboratory, Massachusetts Institute of Technology. (Also published as an MIT-AI tech memo.)

Smith, D. E., and Clayton, J. E. 1980. A frame-based production system architecture. In *Proceedings of the First Annual National Conference on Artificial Intelligence,* pp. 154–156.

Sox, H. C., Sox, C. H., and Tompkins, R. K. 1973. The training of physicians' assistants: The use of a clinical algorithm system. *New England Journal of Medicine* 288: 818–824.

Speicher, C. 1978. Survey of interpretive reporting in clinical pathology. Technical report, Department of Pathology, Ohio State University.

Spilich, G. J., Vesonder, G. T., Chiesi, H. L., and Voss, J. F. 1979. Text processing of domain-related information for individuals with high and low domain knowledge. *Journal of Verbal Learning and Verbal Behavior* 14: 506–522.

Sridharan, N. S. 1978. Guest editorial. *Artificial Intelligence* 11: 1–4.

Sridharan, N. S., and Schmidt, C. 1977. Knowledge-directed inference in BELIEVER. In *Pattern-Directed Inference Systems,* ed. D. A. Waterman and F. Hayes-Roth. New York: Academic Press.

Startsman, T. S., and Robinson, R. E. 1972. The attitudes of medical and paramedical personnel towards computers. *Computers and Biomedical Research* 5: 218–227.

Stead, W. W., Heyman, A., Thompson, H. K., and Hammond, W. E. 1972. Computer-assisted interview of patients with functional headache. *Archives of Internal Medicine* 129: 950.

Stead, W. W., Brame, R. G., Hammond, W. E., Jelovsek, F. R., Estes, E. H., and Parker, R. T. 1977. A computerized obstetric medical record. *Obstetrics and Gynecology* 49: 502–509.

Stedman. 1961. *Stedman's Medical Dictionary,* 20th ed. Baltimore: Williams and Wilkins.

Steele, A. A., Davis, P. J., Hoffer, E. P., and Famiglietti, K. T. 1978. A computer-assisted instruction (CAI) program in diseases of the thyroid gland (THYROID). *Computers and Biomedical Research* 11: 133–146.

Stefik, M. J. 1979. An examination of a frame-structured representation system. In *Proceedings of the Sixth International Joint Conference on Artificial Intelligence,* pp. 845–852. Stanford, CA: Stanford University, Department of Computer Science.

Stevens, A. L., and Collins, A. M. 1978. Multiple conceptual models of a complex system. In *Aptitude, Learning and Instruction: Cognitive Process Analysis,* eds. R. Snow, P. Federico, and W. Montague. Hillsdale, NJ: Lawrence Erlbaum Associates.

Stevens, A. L., Collins, A. M., and Goldin, S. 1978. *Diagnosing students' misconceptions in causal models.* Report no. 3786, Bolt Beranek and Newman, Cambridge, MA.

Strauss, M. B., and Welt, L. G. 1971. *Diseases of the Kidney.* Boston: Little, Brown.

Suppes, P. 1970. *A Probabilistic Theory of Causality.* Amsterdam: North-Holland.

Suppes, P., and Morningstar, M. 1972. *Computer-Assisted Instruction at Stanford, 1966–68: Data, Models, and Evaluation of the Arithmetic Programs.* New York: Academic Press.

Sussman, G. J. 1973. Some aspects of medical diagnosis. Artificial Intelligence Laboratory, Massachusetts Institute of Technology.

Sussman, G. J., and McDermott, D. V. 1972. From PLANNER to CONNIVER—A genetic approach. In *Proceedings of the Fall Joint Computer Conference,* p. 1171.

Sussman, G.J., Winograd, T., and Charniak, E. 1971. MICRO-PLANNER reference manual. Report no. 203A, Artificial Intelligence Laboratory, Massachusetts Institute of Technology.

Suwa, M., Scott, A. C., Shortliffe, E. H. 1982. An approach to verifying completeness and consistency in a rule-based expert system. *AI Magazine* 3(4): 16–21.

Swanson, D. B. 1978. Computer simulation of expert problem solving in medical diagnosis. Ph.D. dissertation, University of Minnesota.

Swanson, D. B., Feltovich, P. J., and Johnson, P. E. 1977. Psychological analysis of physician expertise: Implications for design of decision support systems. In *MEDINFO 77,* pp. 161–164. Amsterdam: North-Holland.

Swanson, D. B., Feltovich, P. J., Johnson, P. E., and Moller, J. H. 1979. A computer simulation study of clinical expertise: Toward a knowledge-based definition of clinical competence. Paper presented at the 1979 annual meeting of the AERA, April, San Francisco.

Swartout, W. R. 1977. A digitalis therapy advisor with explanations. Report no. TR-176, Laboratory for Computer Science, Massachusetts Institute of Technology.

———. 1981. Producing explanations and justifications of expert consulting programs. Ph.D. dissertation, Departments of Electrical Engineering and Computer Science, Massachusetts Institute of Technology. (Report no. LCS-TR-251.)

Swets, J. A., and Feurzeig, W. 1965. Computer-aided instruction. *Science* 150: 572–576.

Szolovits, P. 1976. Remarks on scoring. Unpublished class notes.

———. (ed.). 1982. *Artificial Intelligence in Medicine.* Boulder, CO: Westview Press.

Szolovits, P., and Pauker, S. G. 1976. Research on a medical consultation system for taking the present illness. In *Proceedings of the Third Illinois Conference on Medical Information Systems,* pp. 299–320. Chicago: University of Illinois at Chicago Circle.

———. 1979. Computers and clinical decision-making: Whether, how, and for whom? *Proceedings of the IEEE* 67: 1224–1226.

Taylor, T. R. 1976. Clinical decision analysis. *Methods of Information in Medicine* 15: 216–224.

Teach, R. L., and Shortliffe, E. H. 1981. An analysis of physician attitudes regarding computer-based clinical consultation systems. *Computers and Biomedical Research* 14: 542–558.

Teitelman, W. 1978. *Interlisp Reference Manual.* Palo Alto, CA: Xerox Palo Alto Research Center.

Teitelman, W., and Masinter, L. 1981. The Interlisp programming environment. *Computer* 14: 25–33.

Thomas, L. 1980. On artificial intelligence. *New England Journal of Medicine* 302: 506–508.

Thro, M. P. 1978. Relationships between associative and content structure of physics concepts. *Journal of Educational Psychology* 70: 971–978.

Trigoboff, M., and Kulikowski, C. A. 1977. IRIS: A system for propagation of inferences in a semantic net. In *Proceedings of the Fifth International Joint Conference on Artificial Intelligence,* pp. 274–280. Pittsburgh, PA: Carnegie-Mellon University, Department of Computer Science.

Trzebiakowski, G. L., and Ferguson, I. C. 1973. Computer technology in medical education. *Medical Progress through Technology* 1: 178–186.

Tsuji, S., and Shortliffe, E. H. 1983. Graphical access to the knowledge base of a medical consultation system. In *Proceedings of AAMSI Congress 83,* pp. 551–555. Bethesda, MD: American Association for Medical Systems and Informatics.

Tulving, E., and Pearlstone, Z. 1966. Availability versus accessibility of information in memory for words. *Journal of Verbal Learning and Verbal Behavior* 5: 381–391.

Tversky, A., and Kahneman, D. 1974. Judgment under uncertainty: Heuristics and biases. *Science* 185: 1124–1131.

VanLehn, K., and Brown, J. S. 1979. Planning nets: A representation for formalizing analogies and semantic models of procedural skills. In *Aptitude, Learning, and Instruction: Cognitive Process Analyses,* eds. R. E. Snow, P. A. Federico, and W. E. Montague. Hillsdale, NJ: Lawrence Erlbaum Associates.

van Melle, W. 1974. Would you like advice on another horn? Internal working paper, Computer Science Department, Stanford University.

———. 1979. A domain-independent production-rule system for consultation programs. In *Proceedings of the Sixth International Joint Conference on Artificial Intelligence,* pp. 923–925. Stanford, CA: Stanford University, Department of Computer Science.

———. 1980. A domain-independent system that aids in constructing knowledge-based consultation programs. Ph.D. dissertation, Computer Science Department, Stanford University.

van Melle, W., Scott, A. C., Bennett, J. S., and Peairs, M. 1981. The EMYCIN manual. Report no. HPP-81-16, Heuristic Programming Project, Computer Science Department, Stanford University.

Vickery, D. M. 1974. Computer support of paramedical personnel: The question of quality control. In *MEDINFO 74,* pp. 281–287. Amsterdam: North-Holland.

Vickery, D. M., and Fries, J. F. 1978. *Take Care of Yourself: A Consumer's Guide to Medical Care.* Reading, MA: Addison-Wesley.

Vishnevskiy, A. A., Artobolevskiy, I. I., and Bykovskiy, M. L. 1973. Machine diagnosis and information retrieval in medicine in the USSR. Report no. 73-424, NIH.

Wagner, G., Tautu, P., and Wolber, U. 1978. Problems of medical diagnosis: A bibliography. *Methods of Information in Medicine* 17: 55–74.

Waldinger, R., and Levitt, K. N. 1974. Reasoning about programs. *Artificial Intelligence* 5: 235–316.

Walser, R. L., and McCormick, B. H. 1976. Organization of clinical knowledge in MEDICO. In *Proceedings of the Third Illinois Conference on Medical Information Systems,* p. 159.

Walsh, B. T., Bookhein, W. W., Johnson, R. C., and Tompkins, R. K. 1975. Recognition of streptococcal pharyngitis in adults. *Archives of Internal Medicine* 135: 1493–1497.

Wardle, A., and Wardle, L. 1978. Computer-aided diagnosis: A review of research. *Methods of Information in Medicine* 17: 15–28.

Warner, H. R. 1968. Experiences with computer-based patient monitoring. *Anesthesiology and Analgesia Current Research* 47: 453–461.

———. 1978. Knowledge sectors for logical processing of patient data in the HELP system. In *Proceedings of Second Annual Symposium on Computer Applications in Medical Care,* pp. 401–404. Long Beach, CA: IEEE Computer Society.

Warner, H. R., Olmsted, C. M., and Rutherford, B. D. 1972a. HELP—A program for medical decision-making. *Computers and Biomedical Research* 5: 65–74.

Warner, H. R., Rutherford, B. D., and Houtchens, B. 1972b. A sequential approach to history taking and diagnosis. *Computers and Biomedical Research* 5: 256–262.

Warner, H. R., Morgan, J. D., Pryor, T. A., Clark, S., and Miller, W. 1974. HELP—A self-improving system for medical decision making. In *MEDINFO 74,* pp. 989–1000. Amsterdam: North-Holland.

Warner, H. R., Toronto, A. F., and Veasy, L. G. 1964. Experience with Bayes' theorem for computer diagnosis of congenital heart disease. *Annals of the New York Academy of Science* 115:2.

Wason, P. C., and Johnson-Laird, P. N. 1972. *Psychology of Reasoning: Structure and Content.* Cambridge, MA: Harvard University Press.

Waterman, D. A. 1970. Generalization learning techniques for automating the learning of heuristics. *Artificial Intelligence* 1: 121–170.

———. 1978. Exemplary programming. In *Pattern-Directed Inference Systems,* eds. D. A. Waterman and F. Hayes-Roth, pp. 261–280. New York: Academic Press.

Watson, R. J. 1974. Medical staff response to a medical information system with direct physician-computer interface. In *MEDINFO 74,* pp. 299–302. Amsterdam: North-Holland.

Weber, J. C., and Hageman, W. D. 1972. ATS: A new system for computer-mediated tutorials in medical education. *Journal of Medical Education* 47: 637–644.

Wechsler, H. 1976. A fuzzy approach to medical diagnosis. *International Journal of Biomedical Computing* 7: 191–203.

Weed, L. L. 1968. Medical records that guide and teach. *New England Journal of Medicine* 278: 593–599, 652–657.

———. 1973. Problem-oriented medical records. In *Problem-Directed and Medical Information Systems,* ed. M. F. Driggs. New York: Intercontinental Medical Book Corporation.

Weinberg, A. D. 1973. CAI at the Ohio State University College of Medicine. *Computers in Biology and Medicine* 3: 299–305.

Weiss, S. M. 1974. A system for model-based computer-aided diagnosis and therapy. Ph.D. dissertation, Computers in Biomedicine, Department of Computer Science, Rutgers University.

———. 1976. A system for interactive analysis of a time-sequenced opthalmological database. In *Proceedings of the Third Illinois Conference on Medical Information Systems.* Chicago: University of Illinois at Chicago Circle.

Weiss, S. M., and Kulikowski, C. A. 1979. EXPERT: A system for developing consultation models. In *Proceedings of the Sixth International Joint Conference on Artificial Intelligence,* pp. 942–947. Stanford, CA: Stanford University, Department of Computer Science.

Weiss, S., Kulikowski, C., and Safir, A. 1978. Glaucoma consultation by computer. *Computers in Biology and Medicine* 8(1): 25.

Weizenbaum, J. 1976. *Computer Power and Human Reasoning.* San Francisco: W. H. Freeman.

Wescourt, K. T., and Hemphill, L. 1978. Representing and teaching knowledge for troubleshooting/debugging. Report no. 292, Institute for Mathematical Studies in the Social Sciences, Stanford University.

Wexler, J. D. 1970. Information networks in generative computer-assisted instruction. *IEEE Transactions on Man-Machine Systems* 4: 181–190. (MMS-11.)

Weyl, S., Fries, J., Wiederhold, G., and Germano, F. 1975. A modular self-describing clinical databank system. *Computers and Biomedical Research* 8: 279–293.

Wiederhold, G. 1977. *Database Design.* New York: Wiley.

Wiederhold, G., Fries, J. F., and Weyl, S. 1975. Structured organization of clinical databases. In *Proceedings of the 1975 NCC,* pp. 479–485.

Winograd, T. 1971. A computer program for understanding natural language. Report no. TR-84, Project MAC, Massachusetts Institute of Technology.

———. 1972. Understanding natural language. *Cognitive Psychology* 3: 1–191.

———. 1975. Frame representations and the procedural/declarative con-

troversy. In *Representation and Understanding,* eds. D. G. Bobrow and A. M. Collins. New York: Academic Press.

Winston, P. H. 1972. The M.I.T. robot. In *Machine Intelligence,* eds. B. Meltzer and D. Michie, Edinburgh, U.K.: Edinburgh University Press.

———. 1974. New progress in artificial intelligence. Report no. TR-310, Massachusetts Institute of Technology.

———. 1977. *Artificial Intelligence.* Reading, MA: Addison-Wesley.

Winston, P. H., and Horn, B. 1981. *LISP.* Reading, MA: Addison-Wesley.

Wintrobe, M. M. 1974a. *Clinical Hematology.* Philadelphia: Lea and Febiger.

———. (ed.). 1974b. *Harrison's Principles of Internal Medicine.* New York: McGraw-Hill.

Wirtschafter, D., Carpenter, J. T., and Mesel, E. 1979. A consultant-extender system for breast cancer adjuvant chemotherapy. *Annals of Internal Medicine* 90: 396–401.

Woodbury, M., and Clive, J. 1980. Data-based definitions of disease: Suggestions for a solution of the formal diagnostic problem. In *Proceedings of the Thirteenth Hawaii International Conference on Systems Science,* pp. 590–600.

Woods, W. A. 1975. What's in a link: Foundations for semantic networks. In *Representation and Understanding,* eds. D. G. Bobrow and A. M. Collins, pp. 35–82. New York: Academic Press.

Wortman, P. M. 1972. Medical diagnosis: An information processing approach. *Computers and Biomedical Research* 5: 315–328.

Wortman, P. M., and Greenberg, L. D. 1971. Coding, recoding, and decoding of hierarchical information in long-term memory. *Journal of Verbal Learning and Verbal Behavior* 10: 234–243.

Yerushalmy, J. 1947. Statistical problems in assessing methods of medical diagnosis with special reference to X-ray techniques. *Public Health Reports* 62: 1432.

Young, D. S. 1976. Interpretation of clinical chemical data with the aid of automatic data processing. *Clinical Chemistry* 22(10): 1555.

Yu, V. L., Fagan, L. M., Wraith, S. M., Clancey, W. J., Scott, A. C., Hannigan, J. F., Blum, R. L., Buchanan, B. G., and Cohen, S. N. 1979a. Antimicrobial selection by a computer: A blinded evaluation by infectious disease experts. *Journal of the American Medical Association* 242(12): 1279–1282.

Yu, V. L., Buchanan, B. G., Shortliffe, E. H., Wraith, S. M., Davis, R., Scott, A. C., and Cohen, S. N. 1979b. Evaluating the performance of a computer-based consultant. *Computer Programs in Biomedicine* 9: 95–102.

Zadeh, L. A. 1965. Fuzzy sets. *Information Control* 8: 338–353.

Zobrist, A. L., and Carlson, F. R. 1973. An advice-taking chess computer. *Scientific American* 228: 92–105.

Zoltie, N., Horrocks, J. C., and de Dombal, F. T. 1977. Computer-assisted diagnosis of dyspepsia—Report on transferability of a system, with emphasis on early diagnosis of gastric cancer. *Methods of Information in Medicine* 16: 89–92.

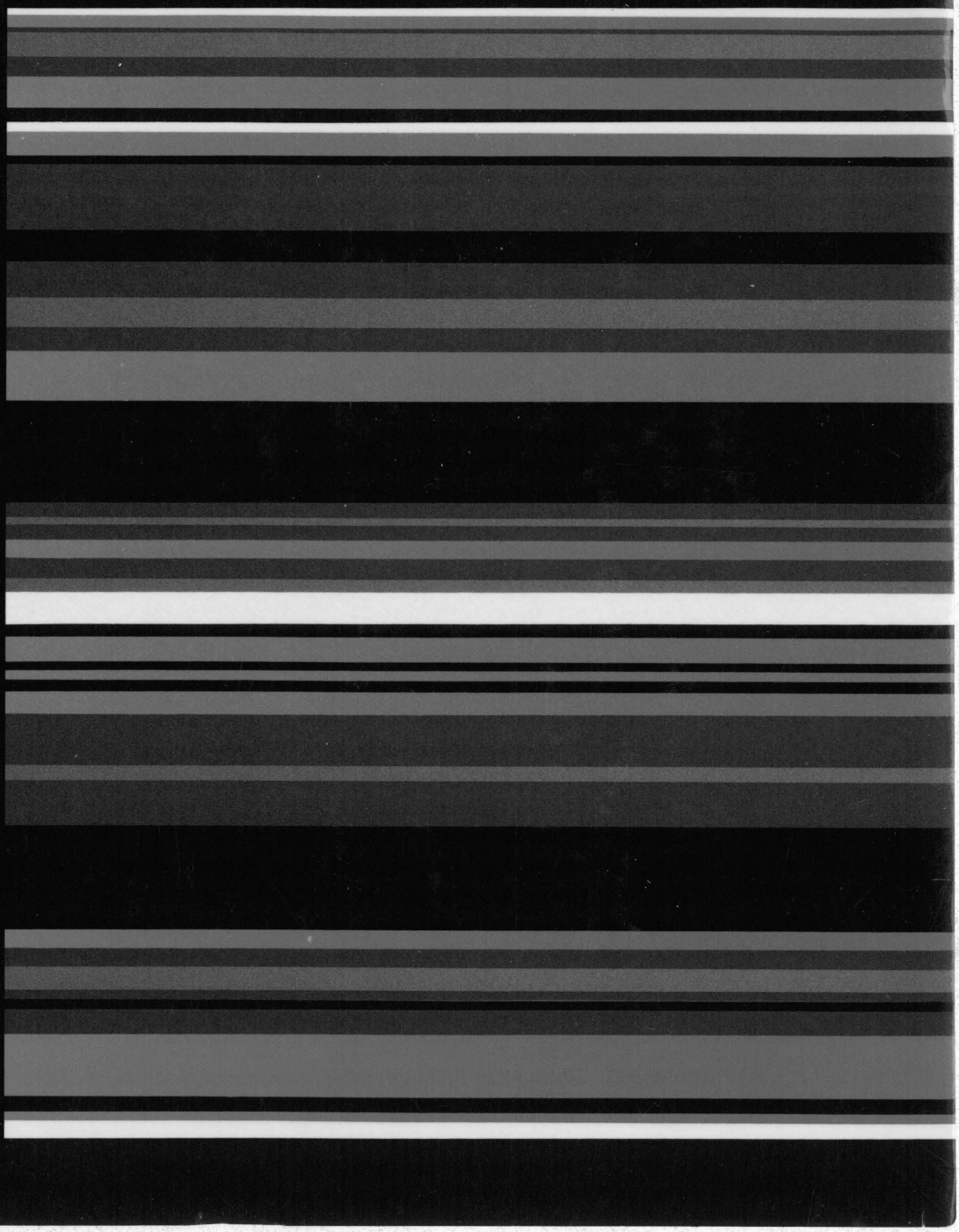
Name Index

Subject Index

READINGS IN MEDICAL ARTIFICIAL INTELLIGENCE

THE FIRST DECADE

William J. Clancey/ Edward H. Shortliffe



READINGS IN MEDICAL ARTIFICIAL INTELLIGENCE
THE FIRST DECADE

William J. Clancey/ Edward H. Shortliffe

Other titles in Artificial Intelligence from Addison-Wesley

Rule-based Expert Systems: The **MYCIN** Experiments of the Stanford Heuristic Programming Project

Bruce G. Buchanan and Edward H. Shortliffe
Stanford University (eds.)
approx. 650 pp., 1984
ISBN 0-201-10172-6

Natural Language Information Processing: A Computer Grammar of English and Its Applications

Naomi Sager, New York University
416 pp., 1981
ISBN 0-201-06769-2

Conceptual Structures: Information Processing in Mind and Machine

John F. Sowa, IBM Systems Research Institute
512 pp., 1983
ISBN 0-201-14472-1

Planning and Understanding: A Computational Approach to Human Reasoning

Robert Wilensky, University of California Berkeley
168 pp., 1983
ISBN 0-201-09590-4

Language as a Cognitive Process — Volume 1: Syntax

Terry Winograd, Stanford University
656 pp., 1983
ISBN 0-201-08571-2

Artificial Intelligence, Second Edition

Patrick Henry Winston, Massachusetts Institute of Technology
400 pp., illus., 1984
ISBN 0-201-08259-4

LISP

Patrick Henry Winston and Berthold K.P. Horn, both of Massachusetts Institute of Technology
430 pp., Paperbound, 1981
ISBN 0-201-08329-9

The Teknowledge Series in Knowledge Engineering

Building Expert Systems

Frederick Hayes-Roth, Teknowledge Inc., Donald A. Waterman, Rand Corporation, Douglas B. Lenat, Stanford University, (eds.)
444 pp., 1983
ISBN 0-201-10686-8

A Guide to Expert Systems

Donald A. Waterman, Rand Corporation,
approx. 200 pp., 1985
ISBN 0-201-08313-2

ADDISON-WESLEY PUBLISHING COMPANY

ISBN 0-201-10854-2