Computer-Based Medical Consultations: MYCIN

Edward Hance Shortliffe
Computer-Based Medical Consultations: MYCIN
Operating and Programming Systems Series
Peter J. Denning, Editor
1. Halstead A Laboratory Manual for Compiler and Operating System Implementation
2. Spirn Program Behavior: Models and Measurement (in prep.)

Programming Languages Series
Thomas E. Cheatham, Editor
1. Heindel and Roberto LANG-PAK—An Interactive Language Design System
2. Wulf et al. The Design of an Optimizing Compiler
3. Maurer The Programmer’s Introduction to SNOBOL
Cleaveland and Uzgalis Grammars for Programming Languages (in prep.)
Hecht Global Code Improvement (in prep.)

Theory of Computation Series
Patrick C. Fischer, Editor
1. Borodin and Munro The Computational Complexity of Algebraic and Numeric Problems

Computer Design and Architecture Series
Edward J. McCluskey, Editor
1. Salisbury Microprogrammable Computer Architectures
3. Wakerly Error-Detecting Codes, Self-Checking Circuits and Applications (in prep.)

Artificial Intelligence Series
Nils J. Nilsson, Editor
1. Sussman A Computer Model of Skill Acquisition
2. Shortliffe Computer-Based Medical Consultations: MYCIN
Computer-Based Medical Consultations: MYCIN

Edward Hance Shortliffe, Ph.D.
Division of Clinical Pharmacology, Stanford University School of Medicine
This book has been adapted in large part from the author's doctoral thesis [Shortliffe, 1974b]. Portions of the work appeared previously in Computers And Biomedical Research [Shortliffe, 1973, 1975b], Mathematical Biosciences [Shortliffe, 1975a], and the Proceedings Of The Thirteenth San Diego Biomedical Symposium [Shortliffe, 1974a].
To Stanford's Medical Scientist Training Program, which is supported by the National Institutes of Health
### MYCIN

**CHAPTER 4. MODEL OF INEXACT REASONING IN MEDICINE**

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1 Introduction</td>
<td>159</td>
</tr>
<tr>
<td>4.2 Problem Formulation</td>
<td>161</td>
</tr>
<tr>
<td>4.3 MYCIN's Rule-Based Approach</td>
<td>164</td>
</tr>
<tr>
<td>4.4 Theoretical Background</td>
<td>167</td>
</tr>
<tr>
<td>4.5 Proposed Model of Evidential Strength</td>
<td>168</td>
</tr>
<tr>
<td>4.6 Model as Approximation Technique</td>
<td>176</td>
</tr>
<tr>
<td>4.7 MYCIN's Use of Model</td>
<td>185</td>
</tr>
<tr>
<td><strong>Appendices</strong></td>
<td></td>
</tr>
<tr>
<td>4.A Paradox of Ravens</td>
<td>188</td>
</tr>
<tr>
<td>4.B Proof of Upper Limit</td>
<td>192</td>
</tr>
</tbody>
</table>

**CHAPTER 5. EXPLANATION SYSTEM**

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1 Introduction</td>
<td>195</td>
</tr>
<tr>
<td>5.2 Using Question-Answering System</td>
<td>197</td>
</tr>
<tr>
<td>5.2.1 Rule-Retrieval Questions</td>
<td>197</td>
</tr>
<tr>
<td>General Questions • Questions Regarding Current Consultation</td>
<td></td>
</tr>
<tr>
<td>5.2.2 Questions Regarding Dynamic Base Data</td>
<td>200</td>
</tr>
<tr>
<td>IQ Questions • EQ Command</td>
<td></td>
</tr>
<tr>
<td>5.2.3 Additional Options</td>
<td>202</td>
</tr>
<tr>
<td>5.3 Future Extensions</td>
<td>203</td>
</tr>
</tbody>
</table>

**CHAPTER 6. FUTURE DIRECTIONS FOR MYCIN**

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.1 Introduction</td>
<td>205</td>
</tr>
<tr>
<td>6.2 Plans for Immediate Future</td>
<td>206</td>
</tr>
<tr>
<td>6.3 Knowledge Acquisition</td>
<td>207</td>
</tr>
<tr>
<td>6.3.1 Current Status of Rule-Acquisition</td>
<td>208</td>
</tr>
<tr>
<td>Subprogram 3 • Interaction of Old and New Rules • Impact of Knowledge Growth on System Performance</td>
<td>215</td>
</tr>
<tr>
<td>6.4 Evaluation of MYCIN</td>
<td>217</td>
</tr>
<tr>
<td>6.4.1 Reliability of MYCIN's Advice</td>
<td>218</td>
</tr>
<tr>
<td>6.4.2 MYCIN's Acceptability to Physicians</td>
<td>219</td>
</tr>
<tr>
<td>6.4.3 MYCIN's Impact on Prescribing Habits</td>
<td>219</td>
</tr>
<tr>
<td>6.4.4 MYCIN's Impact on Patient Care</td>
<td>220</td>
</tr>
<tr>
<td>6.4.5 Speed, Efficiency, and Storage Requirements</td>
<td>221</td>
</tr>
<tr>
<td>6.4.6 Cost of MYCIN's Consultations</td>
<td>223</td>
</tr>
<tr>
<td>6.5 MYCIN and Shared Data Bases</td>
<td>224</td>
</tr>
<tr>
<td>6.6 Prospective Monitoring of Prescribing Habits</td>
<td>226</td>
</tr>
</tbody>
</table>
## Contents

6.7 Educational Applications 230  
6.8 Other Applications of MYCIN Formalism 231  

### CHAPTER 7. CONCLUSION 233  
7.1 Summary 233  
7.1.1 The Clinical Problem 233  
7.1.2 The Solution 235  
7.2 Limitations of MYCIN's Approach 236  
7.3 Contribution to Computer-Based Medical Decision Making 238  
7.4 Contribution to AI 239  

REFERENCES 243  
INDEX 261
Preface

This text is a description of a computer-based system designed to assist physicians with clinical decision-making. This system, termed MYCIN, utilizes computer techniques derived principally from the subfield of computer science known as artificial intelligence (AI). MYCIN's task is to assist with the decisions involved in the selection of appropriate therapy for patients with infections.

MYCIN contains considerable medical expertise and is also a novel application of computing technology. Thus, this text is addressed both to members of the medical community, who may have limited computer science backgrounds, and to computer scientists with limited knowledge of medical computing and clinical medicine. Some sections of the text may be of greater interest to one community than to the other. A guide to the text follows so that you may select those portions most pertinent to your particular interests and background.

Chapter 1, "Introduction" provides an introduction to the fields of medical computing, artificial intelligence, and the clinical problem area for which the MYCIN program is designed. It concludes with an introductory overview of MYCIN and a sample interactive session.

Chapter 2, "Design Considerations for MYCIN," presents the design criteria that were considered during MYCIN's development; acceptability by physicians is emphasized. The chapter ends with a brief discussion of how MYCIN attempts to satisfy the criteria.

Chapter 3, "Consultation System," describes in detail how the MYCIN program makes decisions. The data structures and control structures are discussed in the context of prior work regarding
rule-based problem-solving. Certain subsections of this chapter have been isolated and marked with an asterisk (see below) so that noncomputer scientists can read the more descriptive information without becoming overly immersed in the details of implementation.

Chapter 4, “Model of Inexact Reasoning in Medicine,” covers the general topic of inferential model building and is a somewhat separate topic. It has therefore been written to be self-contained. If your primary interest is in MYCIN’s truth model, you may concentrate on Chapter 4 without needing to refer to other parts of the book for clarification of details.

Chapter 5, “Explanation System,” discusses MYCIN’s ability to answer questions regarding both its knowledge base and the details of a specific consultation. The user interface is described in detail, but implementation information is omitted and may be found elsewhere [Shortliffe, 1974b].

Chapter 6, “Future Directions for MYCIN,” introduces the several plans for evaluation and future extensions of the MYCIN system. These include immediate plans for working on knowledge acquisition procedures, and eventual implementation of the program as one module in a total Hospital Information System.

Finally, Chapter 7, “Conclusion,” summarizes the program’s accomplishments to date and discusses MYCIN’s limitations, plus its contributions to the fields of computer-based medical decision making and artificial intelligence.

Each of the seven chapters is named with an arabic numeral and divided into sections, also specified by arabic numeral designations. Subsections are likewise named. Sections and subsections that are followed by an asterisk (*) denote those parts of the text that contain detailed technical explanations of MYCIN system components; these sections may be skipped. Sections followed by double asterisks (**) are summary sections and may be omitted if you are already well-acquainted with the topic. Reference citations are enclosed in square brackets (e.g., [author, 1976]) and cite the name of the first author plus the year of publication. If an author published more than one referenced article in a single year, a lower-case letter is appended to the date. An alphabetized bibliography appears at the end of the text.

A final point should be made regarding the use of the male pronoun to refer to physicians and patients throughout this work. I have decided to follow convention rather than inject awkwardness in
Preface

an effort to recognize both sexes. It therefore seems wise to stress from the outset that, although such a convention is less than ideal, ‘he’, ‘him’, and ‘his’ are meant to be interpreted without any gender association.

Acknowledgments: This text would still be in its infancy if it were not for the active participation and advice of several individuals. Particularly important to the work was the close association of a group of physicians and computer scientists who have met regularly for several years. At these collaborative sessions Stanley Cohen regularly provided invaluable contributions, and he also was the first person at Stanford to encourage me to extend my time in medical school in order to undertake this research. I am grateful for both his time and his help and guidance. Bruce Buchanan provided computer science expertise, a philosopher’s outlook, and insightful observations without which this project might well have floundered. The analysis in Chapter 4 is largely a product of readings he suggested and mutual discussions of the philosophical issues involved. Stanton Axline also participated in the evolution of MYCIN. He provided the infectious disease expertise needed for the development of the program’s knowledge-base and also suggested several design characteristics that have been implemented to heighten the system’s acceptability to physicians. In late 1973 an additional researcher, Randy Davis, joined the project. Our many conversations have proved invaluable, particularly because he encouraged me to search for ways to generalize MYCIN’s problem-solving approach.

Several other physicians have assisted during the initial development period. The project was conceived during idea sessions at which Thomas Merigan was an active participant, and Gilbert Hunn met regularly with us during the first year. I am also grateful to Michael Podlone and Robert Illa for their assistance testing the program on sample patient cases and for their suggestions regarding its performance. In addition I would like to thank Frank Rhame, Patrick Goodall, Richard Greenman, and Michael Charney for joining with Dr. Merigan to assist with an early evaluation of the program’s performance.

Most of the computer time used for the development of MYCIN was generously contributed by Stanford Research Institute. I am grateful for the advice and assistance of Nils Nilsson, Bertram Ra-
phael, Peter Hart, Richard Waldinger, Dan Lynch, and Richard Fikes, all of whom are associated with SRI. I also want to thank Dr. Nilsson for permission to borrow his thoughts regarding the AI field’s organization as discussed in Section 1.2.

In addition to Dr. Cohen, I would like to thank the other Stanford faculty who have advised me closely and who suggested both academic and research directions appropriate for my interdisciplinary interest: Prof. George Forsythe (Computer Science, deceased), Prof. Byron Brown (Biostatistics), Prof. Patrick Suppes (Philosophy), Dr. Howard Sussman (Pathology), and Prof. Cordell Green (Computer Science).

My thanks also go to the following individuals who have devoted time to useful discussions concerning the research or have assisted with manuscript review: Casimir Kulikowski, Ed Sondik, Ernest Shortliffe, Jonathan King, John Hannigan, Russ Briggs, Ken Colby, James Fries, Mark Perlroth, Roger Schank. The secretarial assistance of Claire Lustig, Sylvia Wyman, and Linda Halloran has also been indispensable throughout MYCIN’s development and I thank them all for the time they have given us.

I want to close with three special acknowledgements: to Bob Greenes for his friendship and for his encouragement regarding a career combining computers and medicine, to G. Octo Barnett for his similar encouragement and for suggesting Stanford as an appropriate medical school for someone desiring to combine the two fields, and to my wife Linda, who somehow managed to adjust to our strange and often incompatible schedules, even when I and my computer terminal tied up the home phone lines for hours at a time.

This work was supported in part by the Medical Scientist Training Program under NIH Grant No. GM-01922. The computer time was provided by Stanford Research Institute under Advanced Research Projects Agency (ARPA) contract DAHCO4-72-C-0008, and by the Stanford University Artificial Intelligence Project. Secretarial and copying services were provided at Stanford University School Of Medicine under BHSRE Grants HS-00739 and HS-01544.

E. H. S.

xvi
Edward Shortliffe’s book represents excellent modern work in artificial intelligence (AI). Leaving aside for the moment its medical focus, the scheme viewed only as a computer system has considerable merit. Dr. Shortliffe broadens the usual definition of an AI program to include the requirement that such a program should justify its decisions: that is, defend its behavior. The MYCIN system can indeed show us the rules and assumptions upon which each decision is based. One advantage of this capability is that the user can evaluate the extent to which the system’s reasoning resembles comparable human decision making, and in many cases the reader of this book will discover how frail a logical basis either machine or human has for many decisions.

In this vein one should also note that MYCIN is distinctly modern AI work in that it relies heavily upon what is called “content specific knowledge”. That is, the program uses a strategy of generalized forms of representation for knowledge and relationships in order to seek and encode advice from experts in its problem domain—in this case infectious disease management. This approach is distinguished from the older and less successful ones which oblige themselves to “reason from first principles”, invoking general concepts and processes rather than rules that have been particularized for the problem at hand. MYCIN then is in that class of AI work that assumes its problem task is difficult and important, and thus utilizes all knowledge helpful in obtaining reliable and credible automated decision making. The application of considerable resources of the system to remaining open-ended is a consequence of this strategy. MYCIN attempts to facilitate “fine tuning” and the acquisition of updated
advice rules in agreement with the opinions of its expert consultants.

Viewed solely from a medical point of view, MYCIN must also be judged a significant accomplishment. The problem domain selected by Shortliffe, namely antibiotic medication for treatment of bacterial infections, is indeed medically exceedingly important. Often this function is performed by a subspecialist consultant in infectious disease. Alternatively the clinical decision to treat with an antibiotic may be based upon the general knowledge of the clinician (internist, family practitioner, surgeon, or pediatrician) in consultation with the laboratorian (pathologist or microbiologist). It is definitely not necessary for MYCIN to command the array of knowledge reflected by these medical specialists. Only a small subset of that knowledge is needed: namely that which relates to the treatment of bacterial infections with antibiotics. As in the case of the human decision maker, MYCIN's knowledge and rules must be up-to-date with respect to microbiological nomenclature and the names and types of pharmacological preparations. As with the human information processors, MYCIN's ultimate decisions can be no more valid than the initial medical observations provided. Basically these are observations of the patient, laboratory isolations of pathogenic microorganisms, and results of antibiotic sensitivity tests. MYCIN, like the physician, will inevitably be led astray by incorrect information about what organism actually was recovered from a patient specimen, or other such errors.

MYCIN shares one more difficulty with its human counterpart: it must deal with the uncertainty that results from missing data. In infectious disease management, some observations simply cannot be recreated. Once antibiotic treatment begins, for example, certain isolations are impossible. Probabilistic operation is natural for a computer program, more so than for most humans. On the other hand, dealing with the conditional statements that relate the probable conclusion to real world events, a task more pertinent to MYCIN's function, is difficult. An example of such a relationship is the appearance of contaminating bacteria in cultures from a patient wound. It requires considerable sophistication for a program to recognize a contaminant. Consider the fact that the hospital laboratorian will ordinarily not feel competent to make the distinction. At least, he or she will feel obliged to report the questionable isolate. That MYCIN can even attempt such things is a tribute to the
Foreword

maturing AI field and serves to set it aside completely from ordinary hospital automation. MYCIN is therefore a significant step forward for AI research, and is directed at the solution of a substantial medical problem.

When one examines in detail the proposal for the scheme to be used in actual hospital and office practices, he or she sooner or later hits upon cost as a potential obstacle. The high cost of operation of artificial intelligence programs written in flexible, high-level languages such as LISP is easily rationalized but not easily avoided. INTERLISP, an interactive version of the same language with additional accommodations, is perhaps 50 times less efficient than FORTRAN, which is perhaps ten times less efficient than an inefficient assembler. Yet LISP permits the construction of lists of relationships describing most naturally the real world information structures necessary for AI programs. LISP also readily permits the alteration or respecification of the lists to reflect a new expert's contribution. Using this capability, MYCIN's formal certainty factors and combining functions permit new knowledge to be incorporated even without explicit commitments as to how or when it will be used. INTERLISP permits easy access to the endlessly evolving structures in order that the computer programs be kept up to date with the changing world. Operational expenses are not related directly to simple machine efficiency. Yet even so, the desirable features of a system such as INTERLISP are expensive.

One could imagine such a system "finished" one day, ready for "hard coding" in an efficient language in an efficient environment. Technically this could easily be accomplished, with a dramatic reduction in operating costs. Based upon past experience with laboratory automation, such a change would be rash because of the quicksilver nature of the medical laboratories. No week will ever pass for a system like MYCIN—or any other medical laboratory computer program—in which changes are not required to up-date the program with changes in practice. The changes need not be conceptual; MYCIN is actually prepared to accept conceptual changes, whereas no ordinary laboratory program can do this. In fact most programs cannot tolerate so much as the change of a field type or an additional digit on a patient number. The majority of the endless changes will be ordinary: new drugs, new names for old drugs, new preparations or formulations, changes in bacteriological nomenclature, new isola-
tion or subculture techniques, new combinations of results to be reported together or separately, or new types of specimens in new preservatives or holding solutions to be treated in new ways by the laboratory workers.

In addition, a number of other computer-based systems are beginning to deal with drug-drug interactions, and some with the ways in which patient status (e.g., renal failure) relates to choice and quantity of the administered drug (including antibiotics). Eventually MYCIN must either talk to such programs, or incorporate their pertinent knowledge. An additional future linkage for the MYCIN system, and an additional advantage in remaining in INTERLISP, is the obvious need to interface with hospital data acquisition systems. MYCIN is undertaking to render consultation; however there is the other world of acquisition, quality control, and reporting of the basic laboratory observations and measurements. Since these other systems are typically rigid, it will be up to the advanced systems like MYCIN to remain adaptable so that interchange of information between the two can take place in the future.

The general argument that highly flexible and accommodating programs are necessary simply to support interaction with medical subject matter experts is a moot point. If there must be a general rule it should be that a system should be capable of taking advice however an expert wants to give it: on-line, off-line, from rules, from tables, from graphs. Many systems, for instance the author mode of computer-aided instruction systems, have been “improved” by compulsory on-line case-building to the point of becoming burdensome. Such approaches fundamentally shift the program updating task from the program to the terminal user. In the case of MYCIN, however, the point is moot because the expert probably needs interaction with the system in order to discover what rule needs challenging and to see the full consequences of a change before it is made. In other words, the user needs the interaction more than the program does. Consequently, in order for the program to continue to grow—at least in wisdom—the relatively expensive high-level interaction may be desirable almost indefinitely.

These reasons suggest that the advantages to MYCIN outweigh the higher costs of operation. Artificial intelligence programs were made possible by flexible programming languages. The lack of flexibility, not the costs, has signaled the eventual death of almost all previous
laboratory systems that used traditional programming techniques. MYCIN should be evaluated—sink or swim—as a relatively costly but exquisite machine tool.

The future importance to medicine of AI programs is twofold. First, it is possible that this approach to knowledge acquisition is easier—at least for the medical people—than the older programming methods that relied upon simple trees or likelihood tables. A MYCIN-like approach could explore in time a more complex problem domain and provide a means serially to examine the state of knowledge in a field not yet wholly rationalized, but such a system could be capable of accepting fairly radical correctives during its subsequent development. MYCIN and its derivatives will want and need changes to assumptions and rules, as long as the changes move it toward more valid decision making. An AI program such as MYCIN can be expected to facilitate the extension of computer-based automation into new medical areas simply because of its ability to evolve a knowledge representation for the field.

Secondly, and perhaps even more exciting is the potential for MYCIN and similar systems to enhance human understanding of the problem domain itself. This advantage is characteristic of good modeling work in general. One expects major gains in perception to emerge from the initial task of formalizing one’s knowledge. The difficulty then (with respect to medicine) lies in the selection of the proper medical problem domain. Minimally, the domain must have some structure, that is, a valid fundamental model of the data generating process. To take an outlandish example, a system such as MYCIN based upon idle ramblings about vapors, phlogistin, and bad air could not make valid decisions, although it might successfully imitate the decisions of physicians of 200 years ago. On the other hand, in the present case we do have the advantage of a microbial model of infectious disease which gives at least the possibility that MYCIN can make a valid decision.

Another requirement for successful AI work in the medical domain is the existence of a reasonable amount of knowledge about the relevant decision making—however organized—in the heads of the expert practitioners. In infectious disease management, there are many explicit rules relating diagnosis to the outcome of drug therapy. In contrast, the action of the individual antibiotics is poorly conceptualized. The testing of bacterial isolates by impregnated discs
for “antibiotic sensitivity” is at best a kind of standardized artifact that results in notoriously poor in vivo versus in vitro correlation. For example, *Salmonella typhosa*, the bacterium that causes typhoid, is by standard laboratory testing “sensitive” to most antibiotics in the test battery, yet experience has shown that successful treatment must be with chloramphenicol (or perhaps ampicillin). Fortunately experts know this and can cite the proper behavioral rules even though they cannot give explanations; unfortunately there are scores of such exceptions, and they are not always known to the non-expert. It is in this type of practical problem in clinical medicine that MYCIN can be useful.

This discrepancy between theory and practice in treatment with antibiotics is related to the well-known scientific quest to deduce pharmacological function from chemical structure. While the chemically complex antibiotic molecules might not be the obvious area in which to anticipate such a discovery, examination of common clinical outcomes and MYCIN relational rules might well contribute to this kind of investigation.

Ultimately this work must be evaluated according to whether it results in drug selections that are associated with improved clinical outcomes, and whether new understanding of biological relationships results from the development of the programs. Indeed there is an excellent chance that MYCIN can improve drug selection for the non-expert. With respect to the possibility of gaining new understanding of infectious disease relationships, MYCIN has a vast advantage over ordinary laboratory automation systems. The traditional systems attempt merely to automate present procedures. MYCIN attempts to formalize the representation of knowledge of infectious disease. Doctor Shortliffe’s book explains quite clearly how he goes about this process. His book is as much a pleasure to read as his system is a pleasure to contemplate.
Chapter 1

Introduction

1.1 Computer Applications in Medicine

1.1.1 PROBLEMS AND PROMISE

In the late 1960’s, David Rutstein wrote a monograph entitled *The Coming Revolution In Medicine* [Rutstein, 1967]. His discussion was based on an analysis of several serious problems for the health professions:

(1) modern medicine’s skyrocketing costs;
(2) the chaos of an information explosion involving both paperwork proliferation and large amounts of new knowledge that no single physician could hope to digest;
(3) a geographic maldistribution of MD’s;
(4) increasing demands on the physician’s time as increasing numbers of individuals began to demand quality medical care.

Rutstein concluded that technology provided a possible partial solution to several of these problems.

In subsequent years technology has indeed increased its influence in the medical sphere, but the problems listed above are still highly visible. Their ultimate solutions will undoubtedly involve a long process, only portions of which can be accomplished by technological innovation alone. Equally important are appropriate supportive legislation, at both state and federal levels, plus a gradual change in the attitudes of health personnel towards their training, their profes-
sional duties, and the technological environment that will increasingly surround them.

The attitudes of health personnel towards computers provide some of the greatest barriers to successful implementation of computer-based systems. A recent study [Startsman, 1972] used an open-ended questionnaire and factor analysis to provide information concerning the optimal interfacing of a computer-based information system with a medical staff. Results indicated that interns, nurses, and ancillary personnel expressed the least willingness to use data processing systems, while medical faculty, pre-clinical medical students, and medical record librarian students were most receptive. Although acknowledging that house staff attitudes may reflect the fast-paced environment in which preoccupation with the immediate physical needs of the patient is the norm, the authors point out that interns and residents comprise precisely the group for which many clinical computing systems should be oriented. Thus, since the study showed that familiarity with computers tends to dispel fears and breed interest, the authors suggest that health personnel should be exposed to data processing techniques during their educational years when they are apt to be most receptive to these kinds of innovation.

The most commonly expressed fears regarding computer applications in medicine involve loss of job (or job stature) due to ‘replacement’ by a computer, and presumed depersonalization of patient care due to machine intervention. In addition, some physicians are concerned about the legal ramifications in the use of, or failure to use, a computer-based facility [Hall, 1972]. Computers appear remarkably cold and sterile, particularly to individuals unfamiliar with their capabilities and limitations. “Scare” articles in professional journals also help reinforce attitudes of distrust [Eisenberg, 1974].

A group at Duke University Medical School has suggested that the key to physician acceptance of computer technology lies in a “practical demonstration that physicians or groups of physicians using [computers] have a clear advantage in practice over physicians who maintain the status quo” [Rosati, 1973]. Applications that can make such a demonstration convincingly, however, are difficult to imagine. Norms of practice already vary considerably, even within close geographic proximity, and mechanisms for measuring one clinician’s “advantage” over another’s have so far tended to emphasize economic considerations (e.g., length-of-stay and utilization review as a primary method for medical audit and quality assessment).
Introduction

The subject of economics also raises important questions regarding the cost of medical computing, another major impediment to acceptance of the technological innovation. Difficulty in quantifying the dollar-value of improved patient care quality has understandably frustrated economists who have tried to apply conventional theory to the unique medical marketplace. As a result, there are now specialists in medical economics who have proposed new conventions and analytical tools for considering questions of cost effectiveness and resource allocation within health care environments [Klarman, 1965]. The basic problem remains unsolved despite these efforts. One of the first questions a hospital administrator asks when a computer system is proposed is: how much will it cost? It is seldom easy to justify such systems as cost effective because the savings are buried in reduced length-of-stay data, in lowered labor or pharmacy charges for the patient, in "improved patient care," or in similar real but imprecise monetary measurements.

Finally, many computer innovations are proposed as time saving techniques for the physician. In an age when a doctor shortage and maldistribution is well recognized [Fein, 1967], such arguments can be highly compelling. By inference, however, any computer program that saves physician time must be doing a task that previously was done by the physician himself. The complex psychological and ethical issues involved here, both for the physician and the patient, will be discussed in greater detail when we describe computer-based clinical decision making in § 1.3.

1.1.2 (***) MEDICAL COMPUTING APPLICATION AREAS

The discussion in § 1.1.1 does not specify which computer applications are relevant to each point because almost all medical computing systems entail similar philosophical, ethical, and economic considerations. In this subsection I briefly describe the major areas of medical computing service and research. The categories are my own, and may therefore be nonexhaustive, but they should serve to give you a general feeling for the ways in which the so-called "computer revolution" is affecting the administration and the practice of medicine. General references on the subject of computer applications in medicine include Lindberg's volume from the University of Missouri [Lindberg, 1968], a comprehensive survey of medical computing in England [Abrams, 1970], a four-volume continuing series that sum-
MYCIN


1.1.2-1 Business Applications

The most widely used and accepted computer-based applications involve hospital accounting systems. Business computing is perhaps the best developed of all computer applications, both because accounting uses have been a major concern of many computer firms since the industry was in its infancy, and because accounting problems are in general well-defined and thereby more straightforward to develop and implement. Automated accounting developed for the business world has required very little adaptation for hospital application. It is hardly surprising, then, that hospital accounting functions have been the first medical functions to be automated. Not only is this priority logical in light of the success and experience that general industry has acquired by using the computer for financial activities, but the application also demonstrates easily recognizable monetary benefits.

The need for computing systems to handle financial data and to print out forms has been heightened in recent years by the explosive rise in hospital rates and the concomitant need for increased and improved communication between the hospitals and third party payers or the government. The private physician has been faced with the same paperwork proliferation on a smaller scale. As a result, several service computing firms offer individual office-based financial packages to practitioners who find it difficult to maintain their patient care schedules, particularly with welfare cases, because processing all the paperwork by hand has become exceedingly tedious and time consuming.

It should be noted that much of the public opinion regarding computers is derived from direct contact at the financial level between the consumer and the computers that send him his bills. Thus a patient who is directed to sit at a console for an automated medical history may well think back to his last erroneous bank statement or computer-generated billing error and rebel at the thought that a similarly error-prone machine is about to take charge of his physical well-being. Physicians asked to read computer-generated summaries
may also question the reliability of the information. Thus improved performance levels for business computer applications, both through increased machine reliability and utilization of well-trained and responsible systems personnel, may be a necessary first step towards improving the public image of computers and thus lowering the barriers of resistance to computing innovation in medicine. This trend is already underway and is being aided by the increasing number of young adults who have grown up in the computer age. The novelty and mysteriousness of computers have made them especially threatening to individuals who remember, for example, the hand-posted billing statements they received in the precomputer era.

A final important point regarding the introduction of financial computing into the doctor’s office is that the related hardware and communications equipment will be increasingly familiar and accessible. The same computer terminal that is purchased for sending daily billing and insurance data from the office to a central financial computing service could presumably be used for connecting with a network of computer-based clinical resources. Thus, little or no additional capital outlay may be necessary for the future physician to interact with computer programs designed to help with the day-to-day practice of medicine. The challenge is, then, to develop good computer-based clinical tools so that the physician will take time to use them regularly (and be willing to pay for the associated computing charges) because they are of genuine help in his practice.

1.1.2-2 Biomedical Engineering

It is convenient to divide medical computing applications into two areas—those identifiable as biomedical engineering tasks and those more appropriately termed information processing or data handling. The primary component of biomedical engineering applications is the analysis of analog signals or the construction of sophisticated technologies for man-machine interaction. This is a vast field that includes such applications as medical computer graphics [Newton, 1973; Cox, 1967; Alderman, 1973], computer assisted pattern recognition from visual signals [Bahr, 1973; Neurath, 1966], computer analysis of real-time data [Computers and Medicine, 1973a; Harrison, 1971; Henry, 1968], and various kinds of patient monitoring.

Patient monitoring includes all applications in which computers
are used to process, or monitor, signals relayed by machines that measure the patient's physiological parameters. By far the largest subfield in this category is the development of programs that analyze electrocardiograms (EKG's) (in recent years, literally hundreds of articles on this subject have been published annually). The vastness of the literature reflects the well-recognized need for computer programs that can assist the physician with EKG analysis; this field is of particular value in medically underserved areas where the expertise of highly trained cardiologists may not be readily available. However, the size of the existing literature also suggests that the ultimate program for this purpose has not yet been created. Indeed, although several programs do very well at EKG analysis [Wartak, 1971; Caceres, 1964; Pryor, 1969; Wolk, 1972], none has yet achieved the accuracy of a good and experienced cardiologist. Similar work has also been done on the even more complex problems of electroencephalogram (EEG) analysis. Results in this field have so far been rather rudimentary and have tended to concentrate on the identification of abnormal spikes in the tracings from the various leads [Walter, 1968; Cox, 1972; Kellaway, 1973].

The phrase "patient monitoring", however, generally implies more than signal sampling and analysis [Warner, 1968; American Medical News, 1970; Felsenthal, 1973]. Also involved is the concept of a warning system, wherein a computer is programmed to sample a patient's physiologic parameters at specified intervals and to warn the nursing or medical staff if an abnormal or dangerous reading is noted. The ethical and legal implications of such systems are only gradually being worked out. Even more revolutionary will be systems in which the computer not only notes the abnormalities but takes corrective action by injecting a drug, altering a pacemaker setting, etc. Although such systems are often discussed, none has yet been implemented for ongoing service.

1.1.2-3 Multi-Phasic Health Testing

As health care critics have increasingly pointed out, the tendency of American medicine is to concentrate on crisis care, largely ignoring the need for improved preventive medicine. The health care and industrial communities have begun to counteract this tendency by screening large populations and identifying individuals with early
Introduction

or latent disease. “Multi-phasic health testing” (MPHT) is the common term for procedures whereby apparently healthy individuals are given a battery of screening tests to determine whether an individual may need further medical attention [Oszustowicz, 1972; Collen, 1964, 1965, 1966, 1969, 1971]. The various MPHT centers use computer technology to varying degrees, in most instances primarily for collecting data and for printing them in an organized fashion that facilitates their review by the staff physicians.

Many MPHT centers also use computers to obtain the patient’s medical history. Automated history-taking has been developed primarily within the last decade [Grossman, 1968; Slack, 1966] and generally involves easy-to-use pushbutton display terminals. The patient sits at the ‘scope for varying lengths of time, usually from 30 to 60 minutes (depending upon the complexity of his complaints), and answers the multiple choice questions by pushing the button beside the correct answer. The programs utilize branching logic so that more specific questions may be asked of patients from whom more detailed information seems relevant.

Such programs have also been used in hospital outpatient clinics. Summaries of the history are legibly printed by the computer for review by the physician when he sees the patient. He may then pursue in detail topics about which the computer has indicated an extensive history may be necessary. Another capability of automated history recording is that of asking the questions in one language and printing the medical history summary for the physician in another. Thus, the computer may serve as a useful translator in cases where, for example, the patient speaks only Spanish or French and the physician only English. Studies to evaluate such systems generally indicate that patients accept the automated history recording more readily than the physician does [Grossman, 1969, 1971]. The summary for the physician is gradually being improved, however, as the designers of these systems gain experience and acquire insight into the reasons for physician resistance.

1.1.2-4 Automated Medical Records

One of the great differences between modern medicine and the clinical practice of a century ago is that the care of a patient is shared, particularly in teaching institutions. Thus the medical record
that once served merely as a worksheet for the individual physician to jot down personal reminders now is an important means of communication for the physicians caring for the patient. Furthermore, the medical record now also serves as an important legal document.

Unfortunately, the medical record has not yet fully evolved to meet the demands of all these requirements. Charts are not usually standardized, are often poorly organized, and tend to be illegible. Redundancy of data is to be expected since health professionals using the medical record tend to duplicate information; they often have neither the time nor the practice to search the chart to see if the data have already been entered.

Recognizing the chaos that arises out of the conventional medical record system, several researchers have suggested new organization techniques and potential mechanisms for automation. Most notable, perhaps, is the Problem Oriented Medical Record (POMR) proposed by Weed [Weed, 1968, 1969a]. He developed the approach at Case Western Reserve, and in recent years has used computer technology to automate the system both there [Weed, 1969b] and at the University of Vermont. The POMR approach has also been advocated as an aid to medical audit [Weed, 1971], although recently questions have been raised regarding its usefulness for this purpose [Fletcher, 1974]. Nonetheless, the system has received wide attention [Bjorn, 1970; Collins, 1973; Esley, 1972; Feinstein, 1973; Goldfinger, 1973; Hurst, 1971a, 1971b, 1972, 1973; Mittler, 1972] and is now used routinely at several hospitals, particularly in the eastern United States. Only Weed’s group has automated the POMR, although similar work has been undertaken at the Massachusetts General Hospital [Greenes, 1969, 1970a] where a computer-based clinical data management system has been utilized in the outpatient hypertension clinic, the coronary care unit, and for systemized input of radiology reports [Pendergrass, 1969; Bauman, 1972]. The important point to note regarding the computer systems of Weed and Greenes is that each is designed for use by the physician himself, both for data input and data retrieval. Thus, in accordance with our comments above, physician acceptance of such systems must remain a primary consideration during program development and implementation.

An alternative to both the traditional source-oriented record and the POMR is the time-oriented databank (TOD) introduced at Stan-
forford Hospital [Fries, 1972]. The TOD System, like the POMR, is primarily a revision in the organization of the hard-copy record. Automation has been introduced only for off-line data entry and analysis. The TOD system emphasizes chronological organization of patient data so that flow-charted trends can be observed over time. Physician interaction with the computer is not yet a part of the TOD approach.

Several other groups have worked with automated records, most of which only peripherally involve the physician. The Kaiser Hospital System is particularly notable in the field [Davis, 1968; Collen, 1964], but other excellent work with both inpatient and outpatient records has also been done elsewhere in the United States [Grossman, 1973; Slack, 1967; Kiely, 1968] and abroad [Buckley, 1973]. Some investigators have looked for ways to automate records without sacrificing the conventional text format [Korein, 1963; Levy, 1964; Bross, 1969], while others have attempted to introduce structure to the records by using checklists or self-encoding forms [Yoder, 1966, 1969; Collen, 1971; Hall, 1967]. Finally, some observers have argued that it is premature to study the structure and optimization of patient data-handling without first assessing and improving the quality of the data themselves [Feinstein, 1970].

1.1.2-5 Laboratory and Pharmacy Systems

Unlike clinical parameters best known to the physician himself, patient data related to lab tests and administered drugs can be acquired from sources other than the doctor. Thus several systems have been developed to aid in the acquisition and control of laboratory and pharmacy data.

Chemistry laboratory systems are perhaps the most common clinical application of computers. Several excellent systems have been designed [Hamilton, 1973a, 1973b; Raymond, 1973; Katona, 1969] to accomplish one or more of the following tasks:

1. accept test orders, in some cases on-line from the wards;
2. generate schedules for the technicians who collect the appropriate samples from the patients;
3. generate worksheets for the technicians running the tests in the laboratory;
MYCIN

(4) provide automatic accessioning for control and identification of samples;
(5) accept test results on-line from various kinds of equipment;
(6) accept other results from terminals in the laboratory;
(7) provide rapid access to test results on any patient;
(8) generate hard-copy records, in a variety of formats, for inclusion in the patient chart or for individual use by physicians.

Other programs suitable for inclusion in the category of laboratory systems are ones for reporting pathology lab diagnoses [Beckett, 1972], for analyzing antimicrobial sensitivity test results [Hulbert, 1973; Groves, 1974] or identification data [Mullin, 1970], for organizing and controlling large collections of laboratory specimens [Bachman, 1973], or for quality control in a microbiology laboratory [Petrali, 1970].

Pharmacy systems generally assist with label printing, inventory control, and maintenance of up-to-date patient drug profiles [Evans, 1971; Almquist, 1972]. One hospital has used such profiles to identify outpatients who are drug abusers [Maronde, 1972]. A novel pharmacy control system has been introduced at Stanford Hospital [Cohen, 1972, 1974] where new drug prescriptions are compared with the patient’s drug profile and warnings for the physician are generated if a potential drug interaction is noted. Finally, the Kaiser Hospital System has reported a computer-based mechanism for monitoring the incidence of adverse drug reactions [Friedman, 1971].

1.1.2-6 Hospital Information Systems

A centralized computer that performs or oversees several of the automated functions described above is called a Hospital Information System (HIS). Since such systems tend to require massive computing facilities, commercial firms are particularly interested in such installations. An HIS usually involves an automated mechanism for patient admission and bed census [Hofmann, 1969] so that a computer-based record for each patient exists from the moment he enters the hospital. The patient record then serves as a focus for information flow. Laboratory and pharmacy data are centrally stored and the system transfers orders directly from the ward, where they are ordered, to the appropriate hospital service. Nursing personnel often use the system to post orders and to indicate when drugs have been
Introduction

administered or other patient care services have been performed. Physicians interact with ward terminals to varying extents, depending both upon the system design and the doctor's willingness to participate. A variety of additional services may also be performed by the central machine. Thus an HIS offers a variety of benefits to the various individuals who may use its data base [Barnett, 1968]:

... To the physician, [HIS is] a system that will provide rapid, accurate, and legible communication of reports, better scheduling procedures and timely and precise implementation of activities ordered for patient care. To the nurse, HIS implies an operation to lighten the clerical load of communication functions, preparing requisitions and transcribing and charting. To the administrator, HIS is a means for using resources more effectively, for gathering the data necessary for appropriate management decisions and for ensuring that information necessary for the patient billing process is readily available and accurate. To the medical research investigator, HIS offers the potential for a data base of patient-care activities that is not only accurate but also organized and easily retrieved and analyzed.

Unfortunately this ideal picture of universal benefit and acceptance of an HIS has yet to be realized. The HIS at El Camino Hospital in Mountain View, California, has served as a model for other institutions considering such ventures. Initiated by Lockheed Aircraft but currently operated by the Technicon Corporation, this large system has surprised observers with its demonstrated cost effectiveness [Batelle Labs, 1973] but has been plagued by low user acceptance, particularly among physicians [Computerworld, 1973; Computers and Medicine, 1973b; Yasaki, 1973]. Suggested reasons for the problems encountered have been numerous. A 1971 article suggested several mechanisms for meeting resistance to hospital automation [Hofmann, 1971], some of which appear to have been overlooked by the El Camino planners. The need for eventual users of the system to participate in the planning process is particularly crucial, as is an effective feedback mechanism so that points of discontent can be overcome before they have a chance to grow. The need for thorough pre-implementation planning of the patient database for an HIS has also been recognized [Sauter, 1973]. Finally there are those who believe that any attempt to introduce a total hospital information system in a single step is doomed to failure from the outset. The alternate approach is to design the various
computer services as modules, perhaps on several small machines, and gradually to integrate them into a total system [Greenes, 1970b; Barnett, 1969; Hofmann, 1968].

1.1.2-7 Decision Support Systems

Computer programs to assist in clinical decision making are the subject of § 1.3. In that section some of the work that preceded the MYCIN system is discussed in detail. Here, we simply note that there are two kinds of clinical decisions that may be involved in such systems—the determination of the patient’s diagnosis or the appropriate way to treat him. In some cases, treatment selection is straightforward once the proper diagnosis has been made. In others, treatment planning may be the most complex step in the decision making process.

1.1.2-8 Computer-Aided Instruction in Medicine

Computer-Aided Instruction (CAI) has become an accepted part of the educational process for many of today’s younger students [Suppes, 1966b, 1969]. As the field has developed, students of the health professions have also begun to benefit from techniques developed by CAI researchers [Stolurow, 1970]. In medical education, a number of successful programs are available nationwide through a network supported by the National Library of Medicine [Wooster, 1973]. Several useful programs most of which avoid problems of natural language understanding, have been developed at Massachusetts General Hospital [Hoffer, 1973]. Ohio State University also has an extensive medical CAI facility [Weinberg, 1973]. Programs that play the role of a patient or otherwise enter into natural language discourse with the student include Cornell’s ATS [Hagemen, 1973; Weber, 1972], and the CASE system at the University of Illinois [Harless, 1973a, 1973b]. A program that simulates the patient-physician encounter, with realistic simulation of the time required for the return of lab results, has also been reported [Friedman, 1973]. Little work has been done to evaluate the cost effectiveness of such systems, but a group at the University of California, San Francisco, has been sufficiently concerned with cost factors that
they have developed a dedicated CAI system for use on inexpensive minicomputers [Kamp, 1973].

1.2 Artificial Intelligence

Although artificial intelligence (AI) has been defined in numerous ways, my preference is to acknowledge the intelligence of any machine that performs a task that a century ago would have been considered a uniquely human intellectual ability. This is a rather broad definition that thus encompasses a much wider group of machines and tasks than is usually ascribed to AI. The appeal of this definition, however, is the avoidance of arguments as to whether or not a specific machine should be classified as a product of the AI field. Furthermore, this definition implies that intelligence is a term that need not apply only to humans. However, one can argue that machine intelligence is not “artificial” at all, but is simply a variety of intelligence where the interplay of emotions, fatigue, and other “uniquely human” characteristics has been eliminated.

The more usual meaning of the term artificial intelligence encompasses a subset of the above definition in which (1) the machine is a digital computer or is controlled by a digital computer, and (2) the task involves symbolic reasoning (“thinking”) rather than arithmetic calculations or information storage and retrieval. AI is therefore generally regarded as a subfield of computer science. The foundations of the field are often attributed to an article written by the late A. M. Turing [Turing, 1950], an English mathematician and logician who proposed an operational test of intelligence, the so-called Turing Indistinguishability Test. In addressing the question “Can machines think?”, he suggests that, for all practical purposes, a machine is intelligent if an individual communicating with the machine (say by means of a teletype) is unable to decide whether he is interacting with a computer or with another human who is also using a teletype.

I shall not attempt to survey the field of artificial intelligence. Several excellent general texts are available that devote considerably more space and energy to such surveys than are available here [Feigenbaum, 1963; Minsky, 1968; Slagle, 1971; Nilsson, 1971]. Critics have also been moved to write entire volumes arguing against the AI field [Dreyfus, 1972]. The reader is therefore encouraged to
consult a recent survey paper [Nilsson, 1974] for a more thorough discussion of AI and for a comprehensive bibliography of the field. An earlier survey of the field also is available [Minsky, 1961]. In the rest of this section I shall follow Nilsson's categories [Nilsson, 1974] for organization of the AI field in an effort to give a brief overview of the kinds of problems with which AI is presently involved.

There are four basic AI methodologies that have been addressed by almost all workers regardless of their specific area of application. In addition, there are approximately eight application areas that encompass most of the work in AI. In the discussion below the eight application areas are listed and briefly described; then the four core topics common to most AI work are introduced.

1.2.1 (**) AREAS OF APPLICATION

1.2.1-1 (**) Game-Playing

Some of the best known work in AI involves the development of computer programs that can play highly complex games [Slagle, 1971]. Programs have been written to play checkers [Samuel, 1959, 1967], chess [Greenblatt, 1967], poker [Waterman, 1970], bridge [Berlekamp, 1963] and several other games that require complex strategies regarding a large number of alternative actions (moves). Such games must be contrasted with a contest such as tic-tac-toe in which the entire range of alternatives can be exhaustively analyzed by a computer and the machine can thereby be programmed never to lose a game.

1.2.1-2 (**) Math, Science, and Engineering Aids

There are few examples of applications in this category (the area into which MYCIN most appropriately falls). Such programs are perhaps best characterized as decision-support systems and, in general, are designed for noncomputer scientists. Some examples of these programs are discussed in § 1.3 and Chapter 3.

1.2.1-3 (**) Automatic Theorem Proving

We are all familiar with high school geometry problems in which the task is to use certain given information in order to prove
Introduction

something else about a geometrical figure. The proving of theorems from known axioms is a general problem area common to various other kinds of deductive logic. Some of the earliest AI programs dealt with this kind of theorem proving, and today the field involves some of the most sophisticated applications that have been developed [Nilsson, 1971; Chang, 1973]. This application area is thus closely related to several others (e.g., robot planning, automatic programming) in which theorem proving techniques are often used as the basic problem-solving methodology.

1.2.1-4 (***) Automatic Programming

Any computer science student who has slaved into the morning hours, trying to find mistakes in one of his programs, can testify to the "intelligence" required in order to write and debug computer programs that perform specified tasks. The idea of a computer that "figures out" how to program itself may seem absurd at first consideration, but considerable progress has been made in this area in recent years [Balzer, 1972]. For example, one approach to the problem is to give the computer some sample program inputs and the corresponding output data. The machine is then asked to create a program that will perform the required transformation.

1.2.1-5 (***) Robots

Science fiction films and modern television notwithstanding, a general purpose robot that walks, talks, and does what you ask it to do has yet to be developed. Work on robotics has involved AI researchers for over a decade, however, and several machines with limited capabilities have been developed [Rosen, 1972; Fikes, 1972; Coles, 1974]. In general this field involves more engineering technology than the other AI application areas because the electrical and mechanical problems in design of the robot itself are substantial. Some projects have limited themselves to computer-controlled arms with associated cameras for scene analysis [Feldman, 1971; Winston, 1972]. These "hand-eye" machines perform tasks in a fixed tabletop environment. Radio-controlled robots on wheels have also been developed [Hart, 1972] and are able to analyze their environment (by means of "on-board" television cameras) and to perform certain
limited tasks. Industry is particularly interested in progress in robotics, as is NASA because of the potential for the use of robots in space exploration. It should be emphasized, however, that the computer program that determines how the robot's task is to be accomplished and then sends appropriate signals to the robot's mechanical devices is an essential part of robot technology and underscores this field's association with the other AI application areas.

1.2.1-6 (**)*Machine Vision

Intimately related to robotics is the development of techniques for analyzing and understanding pictures, usually television pictures [Minsky, 1972; Duda, 1973]. For example, a robot arm that attempts to assemble an engine from parts placed in random locations on a table must be able to locate and recognize the pieces, regardless of their orientation. This problem of scene analysis also involves 3-dimensional perception, edge detection, and disambiguation of lines caused by shadows. Clearly a computer program that makes such judgments on the basis of electrical signals from a television camera is solving a complex intellectual problem.

1.2.1-7 (**)*Natural Language Systems

Computer understanding of natural language [Schank, 1973; Simmons, 1970; Rustin, 1973], either spoken or written, has fascinated computer scientists ever since attempts were first made, in the 1950's, to write programs for translating from one human language to another (e.g., English to Russian). Researchers in this AI application area are closely involved with the field of linguistics, and have been forced to try to understand the nature of language itself. Their problems include analysis of syntax, disambiguation of words with multiple meanings, and analysis of the semantics of language, especially during a lengthy discourse when the over-all context determines the meaning of individual words. Understanding language typed into a machine by teletype has been extended recently to the development of programs that understand spoken words. The latter problem is similar to machine vision in that the program must first analyze electrical signals (in this case, from a microphone rather than a television camera) in order to determine what has been said. Then
an attempt is made to understand the meaning of the words and to have the machine respond appropriately.

1.2.1-8 (***) Information Processing Psychology

Many AI researchers, in accordance with Turing's Indistinguishability Test, are concerned primarily with how well their programs perform the tasks for which they were designed; i.e., they do not necessarily care whether the program solves the problem in the same way that a human does. There are those who believe, however, that by attempting to create programs that solve problems in a manner similar to the workings of the mind, new insights into the psychology of human problem-solving can be discovered. Such work has taken several different forms [Newell, 1970; Schank, 1973; Lindsay, 1972] that interface with all seven of the other AI application areas I have discussed.

1.2.2 (***) AI METHODOLOGIES AND TECHNIQUES

Four core topics in artificial intelligence pervade all eight of the application areas discussed above. Section 1.3 and Chapter 3 describe how the MYCIN system has drawn on work in each of these areas.

1.2.2-1 (***) Modeling and Representation of Knowledge

Writers in the AI field are fond of citing examples of problems that seem exceedingly difficult until a simplified way of expressing the task is discovered. Consider a favorite such example—a 64-square checkerboard, 8 squares on each side, and a box of dominos. Each domino exactly covers two squares. Thus 32 dominos can be used to cover the entire checkerboard. You are asked to arrange 31 dominos on the board so that all squares are covered except the two squares in diagonally opposite corners.

Many people given this task would immediately begin trying to arrange dominos as requested. However, an individual who thinks about the problem in the right way will quickly announce that the task is impossible. The key here is to notice that the diagonally opposite squares on a square checkerboard are always the same color.
Thus, performing the task would require covering 30 squares of one color and 32 squares of the other color. Since every domino must cover one square of each color, dominos arranged on the board must always cover as many squares of one color as the other. Hence the desired final state cannot be achieved (unless some dominos are cut in half).

A variety of modeling and representation schemes has been developed because it has been recognized that the representation of knowledge in the machine may be crucially important to the efficiency with which an AI program is able to perform. These approaches include use of the predicate calculus to represent facts and goals in problem-solving, semantic networks, production systems similar to the grammars that were first proposed by linguists, and procedural representations. The approaches that are most relevant to MYCIN are discussed in Chapter 3.

1.2.2.2 (**) Reasoning, Deduction, and Problem Solving

Since several AI applications involve the writing of programs that solve problems, the development of computer-based problem-solving techniques has been a central concern for many researchers in the field. The most common example used to describe the reasoning tasks involved in the so-called “monkey and bananas” problem. Consider a room containing a monkey, a box, and a bunch of bananas hanging from the ceiling. The distances are such that the monkey is unable to get the bananas unless he is standing on the box. The problem, then, is to write a program that derives a plan so that the monkey can get the bananas. Although the problem may at first seem absurdly simple, it must be remembered that computers have no “common sense” knowledge regarding boxes, monkeys, bananas or distances. The program must therefore be told that boxes may be pushed, that pushing has certain effects on a box and on the individual doing the pushing, that boxes may be climbed on, etc. An intelligent program then deduces, from this basic world knowledge, that the best plan is for the monkey to push the box under the bananas, to climb on the box, and finally to grasp the bananas.

This apparently trivial problem has served as the focus for innovative problem-solving techniques during the past decade. Numerous
Introduction

methodologies and representations of the problem have been suggested. Of course, many problems that are more difficult have been solved, but the puzzle of the “monkey and bananas” remains a convenient common ground for explaining suggested new approaches to computer-based reasoning.

1.2.2-3 (**) Heuristic Search

In many human problem-solving situations there are a large number of possible decisions or actions that may be taken. Imagine, for example, the large number of possible moves at most points during a game of chess or checkers. Since each action may in turn lead to several additional potential actions or responses, the number of possible decisions two or more steps into the future often becomes unmanageable. Humans therefore develop strategies for quickly discounting or eliminating possible actions that they can easily see are less desirable than the two or three best potential decisions. They can thus concentrate on the smaller number of actions, comparing their possible outcomes, and making a reasoned decision on the basis of the most rational alternatives. Programs for solving problems must be given similar strategies so that the machine’s computational power can be efficiently spent concentrating on a small number of possible actions. Despite the computer’s speed and computational powers, many human problems (such as selecting the best move in a game of chess) are so complex that thorough evaluation of each possible move can be shown to require a near-infinite amount of time! Any trick or strategy that can be used by a program in order to limit the number of alternative actions that it must investigate is known as a heuristic. Hence “heuristic search” is the name of the AI problem area in which researchers attempt to identify good strategies that adequately limit the number of alternatives that must be considered (but do not eliminate the alternative that would prove to be the best if all possibilities were thoroughly considered).

1.2.2-4 (**) AI Systems and Languages

A somewhat separate core topic is the development of computing systems and high-level languages for use by AI researchers [Bobrow, 1973]. Since AI applications typically require powerful capabilities
for symbol manipulation, the several common computer languages that emphasize numerical calculations are usually not adequate. Early AI languages emphasized list-processing [Newell, 1957; McCarthy, 1960], but in recent years newer languages have taken on some of the capabilities that originally were left to the applications programmer [Hewitt, 1969; Teitelman, 1974; Rulifson, 1972; Feldman, 1972]. These include search, pattern matching, and backtracking. MYCIN is written in one of these more recent programming languages, a descendant of LISP [McCarthy, 1962] called INTERLISP [Teitelman, 1974].

The brief overview given here has been intended to give you a sense of the kinds of problems and methodologies with which AI is centrally concerned. Perhaps now it is clear why the AI field holds much intuitive appeal for medical researchers who are examining the reasoning processes involved in clinical judgment, medical diagnosis, and the rational selection of appropriate therapy; this point is expanded on in § 1.3. Then, in § 1.4, the medical problem area for which the MYCIN system has been designed is introduced. Finally, § 1.5 introduces the program itself and gives an example of MYCIN’s interactive decision making capabilities.

1.3 Computer-Assisted Medical Decision Making

1.3.1 MAJOR PROBLEM AREA

This section concentrates on an area of medical computing that was mentioned only briefly in § 1.1.2-7. Computer-assisted medical decision making fascinates numerous researchers, partly because analysis of human reasoning is itself challenging, but more importantly because modern medicine has become so complex that no individual can incorporate all medical knowledge into his decision making powers. The field has developed along several dimensions. Therefore it is somewhat difficult to devise an organizational structure for examining the work in this area. Three reasonable dimensions for classifying a computer-based system are:
Introduction

(1) the program’s mode of interaction;
(2) the program’s purpose;
(3) the program’s methodology.

I have chosen to summarize the field in terms of dimension (3), i.e., the various methodologies that have been utilized. The other two dimensions merit brief mention, however.

The decision making program’s mode of interaction, like that of any computer program, is either on-line with the user (usually under some time-sharing monitor) or remote in a batch-processing or other off-line mode. The majority of such programs now operate on-line, interacting either directly with the decision maker or with someone who will transmit the computer’s information to him. There is clearly more opportunity for discourse and explanation in such programs. An interactive system that gives advice in this fashion is often termed a “consultation program.”

The “purpose” of a decision making program would provide a useful basis for classification of the field if there were not so much overlap among the categories. There are at least four kinds of programs along this dimension:

(1) diagnostic programs
(2) prognostic programs
(3) treatment planning programs
(4) educational programs

Programs specifically designed for educational purposes are mentioned in § 1.1.2-8. Any decision program has potential educational side-effects, however, particularly if it is able to explain the basis for its decisions. Similarly, programs for prognosis and treatment planning must in general make a partial diagnosis of the patient’s problem (unless that information is provided by the user at the outset). As is described in § 1.4.1, MYCIN explicitly considers both diagnosis and treatment planning, and also has rules based upon patient prognosis that aid in therapy selection. Furthermore, as is explained in Chapter 2, educational capabilities have been an important design consideration during the current research. The MYCIN system is therefore an example of a system that encompasses all four of the “purpose”
categories I have named. Classification of decision making programs on the basis of these subcategories is hence not particularly useful.

You may well ask why I am so intent upon devising a classification scheme for the programs to be discussed in this section. One answer is that classification leads to structure and, in turn, to understanding. It is therefore the very basis of diagnosis itself [Jelliffe, 1973]. The reasoning processes used by a skilled diagnostician are usually poorly understood, even by the expert (see Chapter 4). Researchers attempting to devise computer-based approaches that parallel human decision making must first therefore assign structure to their problem area in some natural fashion. It is helpful to begin by analyzing the diagnostic process itself [Feinstein, 1967; Card, 1970a; Taylor, 1971] and then to seek a reasonable basis for its automation [Lusted, 1968; Gorry, 1970]. The methodology selected undoubtedly reflects both the specific clinical problem area and the researcher’s own peculiar biases; in fact, the approaches are so numerous that national conferences have been held to communicate the new diagnostic techniques or applications developed [Jacquez, 1972]. Yet, two basic concepts underlying most methodologies are the use of some classification mechanism and, with very few exceptions [Ledley, 1973], the need for numerical techniques.

If the success of medical decision support programs is measured by user acceptance, however, the field has not produced more than a handful of truly useful programs. Croft has examined this field extensively [Croft, 1972] and suggests that attempts to develop new diagnostic models will be largely unsuccessful until three basic problems are solved:

1. lack of standard medical definitions;
2. lack of large, reliable medical data bases;
3. lack of acceptance of computer-aided diagnosis by the medical profession.

Croft explains the significance of the first two obstacles by observing that the more diseases a model is assigned to diagnose, the more difficult is the diagnostic task and, in turn, the less successful a program is apt to be in reaching correct decisions. Despite Croft’s claim that model development should be set aside while his three listed obstacles are first overcome, one may argue that new diagnostic methodologies that pay more attention to the demands of the
user are the only reasonable way to overcome point (3). MYCIN has been designed with this goal in mind (see Chapter 2).

Several attempts have been made to standardize medical definitions. These include the Standard Nomenclature of Diseases and Operations (SNDO), the International Classification of Diseases—Adapted (ICDA), and the Systemized Nomenclature of Pathology (SNOP). Few of these are used extensively in daily medical practice other than for certain reporting purposes. Brunjes has proposed an “anamnestic matrix” concept that would permit computer programs to handle nonstandardized input in a standardized fashion [Brunjes, 1971]. In addition, a British group that evaluated observer variation in history taking and examination found significant degrees of disagreement that were largely reduced when a system of agreed definitions was developed and utilized by the participating physicians [Gill, 1973]. MYCIN has avoided some of these problems by using a large synonym dictionary and by phrasing questions in a manner designed to maximize uniformity of user response (see § 3.3.2-2).

1.3.2 DATA RETRIEVAL AS A DECISION AID

The simplest kind of decision support system merely provides the data for others to make the complicated decisions that depend upon the retrieved information. Such systems generally rely on a computer-based information storage system that accumulates large amounts of data for several patients. Coded information may include physical parameters, diagnosis, treatment plan, and responses to therapy. Physicians may then request information on previous patients who match the current patient on the basis of one or more parameters. Detailed information on how other individuals with similar disease have responded to therapy may help the physician select the best treatment plan for his patient or better evaluate the prognosis for an individual with the particular constellation of symptoms [Feinstein, 1972]. Statistical programs may also provide correlation information that is difficult to deduce merely by looking at retrieved data [Fries, 1972]. A number of medical record systems have been designed with data retrieval requirements as an important consideration [Greenes, 1970a; Shortliffe, 1970; Karpinski, 1971; Feinstein, 1971].
A limited number of medical problem areas are now so well understood that they can be characterized by mathematical formulae. When the computations are complex, physicians are often tempted to take short cuts, making approximations on the assumption that this will compensate for the tendency to forget the formulae or their proper application. Thus computer programs that assist with calculations and their interpretation may be highly useful.

One such clinical problem area is the classification and management of electrolyte and acid-base disorders. The relationship of blood pH to variables such as kidney function and electrolyte levels is well characterized by formulae that utilize the numerical values of blood gas and other laboratory tests. Bleich has written a program that assists the physician with evaluation of such problems [Bleich, 1969, 1971, 1972], and a similar program has been reported by Schwartz [Schwartz, 1970]. These systems were designed primarily to assist physician users. Their developers, therefore, faced many of the same problems of user acceptance and human engineering that have been encountered during the design of MYCIN. Both programs take advantage of time-shared systems with flexible storage mechanisms that permit not only the calculation of patient parameters but also the presentation of useful information regarding the patient's status. Possible etiologies are listed and literature references are given so that the physician may pursue the topic if necessary. A similar program that evaluates the respiratory status of patients in a respiratory care unit, and makes therapeutic recommendations, has also been described [Menn, 1973].

Another problem area in which numerical calculations using well-defined formulae are the primary concern is the customization of drug doses once the agent to be used has been selected. Several examples of programs in this field involve the selection of a digoxin regimen for a patient with heart disease [Sheiner, 1972; Jelliffe, 1972; Peck, 1973]. There is also a program that helps physicians decide on insulin doses for diabetics [Bollinger, 1973]. These systems depend upon a pharmacokinetic model of the body's absorption, metabolism, distribution, and excretion of the drug in question. Inputs to the programs are various clinical parameters for the patient that are then used to calculate the dosage regimen needed to achieve optimal blood levels of the therapeutic agent.
1.3.4 PROBABILISTIC APPROACHES TO DECISION MAKING

Most computer-based decision making tools for medical practitioners are based upon statistical decision theory. The methods used range from simple binary decision trees to conditional probability, discriminant analysis, and clustering techniques.

Explicit decision trees offer advantages in that they clearly represent, when diagrammed, an algorithmic approach to diagnosis. Such diagrams, if memorized or easily accessible, may be useful in visualizing a particular patient’s status and the clinical parameters that should be checked in order to further define his diagnostic (or prognostic) category. The trees are nondynamic, however, and therefore cannot adjust easily to unexpected findings or to unavailable test results. Furthermore, modification of the trees when they are found to be incomplete or inaccurate can be highly complex due to the subtle interrelationships within such reasoning networks. There are several examples of programs that are at least partially dependent upon tree-structured decision pathways [Warner, 1972a; Sletten, 1973; Brodman, 1966; Button, 1973; Koss, 1971; Meyer, 1973].

By far the most commonly used statistical technique employed for computer-based medical decision making is Bayes’ Theorem in its various forms. It is generally utilized as a first-order approximation to conditional probability under the assumption that the patient’s signs and symptoms are jointly independent. In Chapter 4, I discuss the theory in some detail and explain why we chose to reject Bayesian analysis as the basis for MYCIN’s decision model. When comprehensive patient data are available, however, Bayes’ Theorem offers both excellent results and a methodology that lends itself to automation.

In 1964, Warner et al. introduced a computer program that aided in the diagnosis of congenital heart disease [Warner, 1964]. Data had been gathered for several hundred patients with congenital cardiac malformations. As a result, all the conditional probabilities needed for the use of Bayes’ Theorem could be computed. The program accordingly classified new patients with an accuracy similar to that of cardiologists.

Four years later, Gorry and Barnett presented a program that used the same patient data to give results of similar accuracy [Gorry, 1968a]. However, their program used a modification of Bayes’ Theorem (see § 4.2) that permitted diagnoses to be reached in a
sequential fashion. The system was therefore able to suggest the laboratory or physical tests that were most valuable at each step in the decision process. Using a selection function that considered both the current degree of certainty regarding a diagnosis and the cost of additional testing (in terms of money, time delay, and physical pain or inconvenience), the program attempted to minimize the number of tests while maximizing its diagnostic accuracy.

Bayesian programs continue to pervade the literature on computer-based diagnosis. Recent reports from several countries in addition to the United States have presented computer programs using Bayesian analysis both for diagnosis [Gledhill, 1972; Knill-Jones, 1973] and for screening patients who have given automated medical histories [Warner, 1972b]. The technique has been shown to be highly useful in cases where adequate data are available.

Nordyke et al. presented an interesting study using Bayes’ Theorem and two other mathematical techniques for the diagnosis of thyroid disease [Nordyke, 1971]. Having previously reported a pattern recognition approach to the problem [Kulikowski, 1970], the authors compared both Bayes’ Theorem and pattern recognition to a linear discriminant model. “Pattern recognition” is a general term, the interpretation of which depends upon the application area being discussed. In medical diagnosis, the term usually describes a method that “… attempts to extract the most characteristic features of each diagnostic category, rather than trying to discriminate directly between categories. A patient is then classified into the category with which his data shares the most features…” [Nordyke, 1971]. One variation of this technique may be characterized mathematically using a feature extraction procedure that specifies data vectors that may be subjected to cluster analysis. The linear discriminant model, on the other hand, is an attempt to consider the effects of correlation (or second-order interdependence) between characteristics. The discriminant used in the thyroid study is described in detail in the Nordyke paper.

The data used by Nordyke et al. were extracted from the records of 2405 patients who had been seen over a six year period for evaluation of thyroid disease. Their results showed that although the pattern recognition technique performed best in identifying ill patients on the basis of historical data alone, it produced an inordinate number of false positives. Bayes’ Theorem, on the other hand, gave
Introduction

comparatively better diagnostic accuracy as more physical findings and laboratory test results became available. Their report therefore concludes:

Because each of the methods uses the characteristics of a patient differently, some taking advantage of discriminating information at a given stage better than others, it would seem that a combination of these would be best for a sequential diagnostic procedure. However, since the simpler Bayes method provides comparable results at the pre-laboratory stage of diagnosis, it might prove the most effective clinical aid.

Another technique used for sequential decision making is the Shannon entropy formula [Shannon, 1949]:

\[ \text{entropy} = - \sum_i p(X_i) \log p(X_i) \]

Here \( p(X_i) \) is the probability that \( X_i \) is true (e.g., that the patient has disease \( D_i \)). Steps in the sequential process are selected so as to maximize the entropy of the set of possible diagnoses. Several programs have successfully used this selection function [Mullin, 1970; Gleser, 1972], but it should be noted that entropy too is dependent on good probabilistic information.

All the methodologies discussed so far are examples of techniques utilized in the field of decision analysis [Raiffa, 1968]. The last programs for discussion in this subsection are those that encompass several of the techniques—conditional probabilities, decision trees, utility measures, and selection functions for sequential decision making. Ginsberg’s program for diagnosis and management of patients with pleural effusions is an excellent example of this kind of eclectic approach [Ginsberg, 1968, 1970]. In addition, one of the early workers with Bayesian diagnostic programs [Gorry, 1968a, 1968b], has gradually broadened his approach to include several additional facets of decision theory. In joint papers published in the American Journal of Medicine, he and his coworkers presented a comprehensive look at decision theory as applied to medical diagnosis [Schwartz, 1973], and reported a program that uses the techniques to evaluate the etiology of acute renal failure [Gorry, 1973b]. Although neither their techniques nor their results are unique, their presentation is lucid and complete. It has generated positive commentary [Jelliffe, 1973] at a time when, as I have remarked before,
1.3.5 ARTIFICIAL INTELLIGENCE AND MEDICAL DECISIONS

There are relatively few examples of AI programs used for medical decision making. Since 1970, however, a small number of researchers, most of whom have had experience rooted in the traditional approaches described in § 1.3.4, have begun to consider AI techniques. Notable among these is G.A. Gorry, then from M.I.T. He became aware that the purely statistical programs have had three failings that are major impediments to physician acceptance of the systems. First, the programs have no real "understanding" of their problem area. Gorry explains this point as follows [Gorry, 1973a]:

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. From the experts, it is possible to obtain 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.

AI, with its emphasis upon representation of knowledge, offers a natural environment for examining the kind of "concept formation" that Gorry feels is needed.

The second problem is that, even if the traditional programs have been given an understanding of their problem area, they have no mechanism for discussing their knowledge with the user. Physicians are often uninspired by programs that produce a diagnosis and a four-decimal-place probability estimate without being able to answer questions about how the conclusion was reached. Furthermore, physicians attempting to give the programs new information have shared no common language with the computer. Gorry therefore calls for the development of natural language interfaces to permit discourse between physicians and diagnostic programs. Once again AI provides a natural environment for examining this requirement.

The third problem, closely related to the first two, is the need for programs that can explain (i.e., justify) their advice. This capability requires that a program both understand its reasoning processes and be able to generate explanations in a language that is easily under-
Introduction

stood by the physician. Gorry's group has therefore worked on developing knowledge representations and language capabilities that will heighten the acceptability of a system such as their acute renal failure program [Gorry, 1973b]. In Chapter 2, where the design criteria for the MYCIN system are discussed, the similarities between our desiderata and those of Gorry are readily apparent.

The system requirements discussed by Gorry entail more than a natural language "front end" in combination with a statistically-based program. As discussed in Chapter 5, efficient knowledge representation is generally the foundation for man–machine discourse in natural language. Isner's medical knowledge system, for example, has demonstrated the need for an efficient representation scheme, plus a program with problem-solving skills, if a computer system is to communicate with minimally trained users [Isner, 1972]. I do not mean to suggest that statistical theory has no place in AI research. Several AI programs have used traditional numerical techniques [Good, 1970] but have also utilized data structures that facilitate utilization of knowledge in ways that are not possible if system information is stored solely in probability tables. Our own mathematical decision model is introduced in Chapter 3 and discussed in detail in Chapter 4.

Problem-solving techniques from AI also hold a natural appeal for certain researchers in computer-based medical decision making. The various AI methodologies will not be surveyed here because those most pertinent to MYCIN are discussed in Chapter 3. Four medical projects warrant comment in this context, however.

The first is the theory formation system of Pople and Werner [Pople, 1972] that does not attempt diagnosis as such, but does make inferences on the basis of model behavior. The program uses an alternative to deduction and induction—abductive logic [Pople, 1973]. A convincing argument can be made that abduction is the basis for medical diagnosis. Consider, for example, the three statements:

(1) If a person has pneumonia, then he has a fever;
(2) John has pneumonia;
(3) John has a fever.

Deductive logic allows us to derive (3) from (1) and (2); i.e., "since people with pneumonia have fevers, and since John has pneumonia,
John must have a fever.” Induction, on the other hand, uses one or more observations of people for whom (2) and (3) hold in order to infer that (1) is true; i.e., “since I have observed several people with pneumonia, all of whom have fevers, it is perhaps generally true that people with pneumonia have fevers.” Abduction is the remaining combination, namely using (1) and (3) to infer (2); i.e., “since people with pneumonia have fevers, and since John has fever, perhaps it is true that John has pneumonia.” Clearly, the last example parallels a clinical diagnosis on the basis of a patient’s symptomology.

Pople and Werner use the abductive model as the basis of a program for inferring neuroanatomical explanations of the behavior of human neurons in response to central stimulation. The system also includes a simulator that tests hypotheses by modeling them and seeing whether the observed responses are duplicated. The problem, of course, is that the word “perhaps” is not quantified in our explanation of abduction above. It is therefore unclear how to select between two competing hypotheses that are both abductively supported by the same observation(s). In fact, Bayes’ Theorem and the other numerical methods discussed in § 1.3.4 are attempts to solve precisely this problem, although the term “abduction” does not generally appear in the formulation of those techniques.

An Italian group has recently proposed a more quantitative problem-solving approach that uses AI techniques and addresses itself specifically to medical diagnosis [Gini, 1973]. Their central concern, as has been true for several other researchers, is sequential test selection for effective diagnosis, but they propose a model based upon state-transition networks. Having defined operators for transition from one state in the network to another, they present an algorithm for creating a dynamic ordering of the operators on the basis of their “promise.” The algorithm interfaces with a heuristic mechanism for obtaining a diagnosis, i.e., for finding a set of tested symptoms that match a particular disease definition. It is probably wise to reserve judgment about the approach until this model has been automated in a computer program, but it initially appears to offer little advantage over other programs (cf. pattern recognition) that have attempted to define diseases as sets of symptoms.

As I have described (§ 1.2.1-8), there is a large subfield of AI in which investigators are motivated by an interest in psychology. A psychologist from Duke University has reported a fascinating program based upon this approach to medical diagnosis [Wortman,
Introduction

1972]. He views diagnosis as "... a search through a hierarchically organized memory composed of diseases, disease categories, categories of categories, etc. ... along with a parallel hierarchy containing the heuristic decision rules for evaluating these categories." After asking a neurologist to "think aloud" while solving clinical problems, Wortman analyzed the resulting protocols and wrote a program that attempted to mimic the neurologist's approach to cerebellar disease diagnosis. Not only did the program perform as well as the expert in subsequent tests (correctly diagnosing the disease in 19 of 20 sample cases), but it also generated protocols that closely resembled those of the neurologist himself. It is important to note, however, that the program's performance was also based on the expert's subjective probabilities relating cerebellar symptomology to each of the 16 selected diseases that were the subject of the experiment. As a result, Wortman's information processing approach still relies upon the availability of data that reflect the preferences of the expert being modeled. MYCIN also needs such information. AI does not necessarily offer a means for avoiding numerical representation of data relationships, but does suggest new and potentially powerful methods for analyzing the problem domain and selecting relevant knowledge. It will be fascinating to observe Wortman's future work to see if his success continues as the range of possible diagnoses increases and the clinical problem areas are expanded.

Noteworthy work combining AI techniques and mathematical models of disease has been progressing at Rutgers University for the last several years. Like some of the investigators discussed in § 1.3.3, the Rutgers researchers have sought clinical problem areas that could be well-characterized by mathematical models. Envisioning tiered levels of modeling addressed to various degrees of detail, they assert that an appropriate representation scheme provides an important basis for the design of diagnostic strategies [Amarel, 1972]. Their concern reflects a basic agreement with Gorry in his claim that a diagnostic program needs to "understand" the decisions that it reaches [Gorry, 1973a].

The problem area they have selected for testing their approach is the diagnosis and management of glaucoma. This is an ocular disease that may be characterized both by causal relationships over time and mathematical formulae reflecting fluid resistance and flow [Kulikowski, 1971]. They represent disease states in a network based on causal links reflecting various weights (e.g., "always," "almost al-
ways," "sometimes," "never," etc.). The network provides the basis of a consultation program for ophthalmologists who need help in evaluating a patient's status [Kulikowski, 1972a]. Working in close collaboration with an ophthalmologist, the group has also written programs that permit an expert interactively to modify nodes in the causal network or to add new information to the inferential structure [Kulikowski, 1972b]. The result is a dynamic program that has created considerable interest among clinical professionals to whom it has been presented at a national meeting of ophthalmologists [Kulikowski, 1973]. The causal network and mathematical model lend themselves well to the development of novel strategies for test selection during the consultation process [Kulikowski, 1972c]. Furthermore, the group's agreement with Gorry's call for programs that can explain their decisions [Gorry, 1973a] is reflected in the program's ability to present a "parse" of those portions of the network that explain the patient's current clinical state [Kulikowski, 1974]. Although certain of the program's human engineering features currently leave much to be desired (the organization of questions during a consultation and the motivation for individual queries appear somewhat confusing to this observer), the glaucoma system represents a pleasing blend of mathematical and AI techniques that holds great promise for those medical problem areas that can be adapted to this kind of causal modeling.

It is unfortunately the case that most human disease states are not sufficiently well understood to be characterized by well-defined mathematical formulae. Even causal relationships are seldom understood. MYCIN is a program that attempts to use AI techniques to model decision making in ill-defined areas such as these. After all, experts do reach decision when such medical problems arise, and they can usually offer theoretical arguments for making the judgments that they do. Our goal has been to capture such judgmental knowledge and to create a program that uses the information effectively and in a way that is acceptable to the physicians for whom it is designed. These considerations are described in detail in Chapter 2.

1.3.6 PHILosophical Observations

Although medical professionals often demonstrate great resistance to computing innovation, obstacles to acceptance are greatest when
the application demands “hands-on” use at a computer terminal or when the program appears to take over intellectual functions, transcending housekeeping or simple “number crunching” chores. Decision making systems must therefore overcome huge barriers, not only because they usually demand interaction by the professional and are attacking a problem that demands intelligence, but also because the user of the program is in most cases the physician himself. Of all health professionals, the physician is perhaps most pressed for time and most wedded to a self-image that has been ingrained since medical school. Schwartz has discussed this last point [Schwartz, 1970]:

Physicians as a group have traditionally cherished their ability to learn and retain large numbers of facts, to formulate a differential diagnosis and to carry on decision making activities. Introduction of the computer into these processes could well be viewed by the doctor as devaluing his hard-won medical education and as undermining his intellectual contribution to medical care. This loss of self-esteem would, of course, be exacerbated if the patient were to find in the transfer of many intellectual functions from man to machine a basis for viewing the doctor with diminished admiration and respect. Such loss of status could have serious social, economic, and political consequences for a profession that has historically enjoyed eminence in the public mind.

Concern regarding the attitudes of patients is not without foundation. I recently heard a group of individuals agree that, all other things being equal, if they had to choose between a doctor who used computer-based consultation programs and one who did not, they would select the physician who was “intelligent enough” to make decisions for himself.

And what of today’s medical consultants? How will they react if they are made to feel that their professional expertise is no longer in demand because a computer program has intruded into their clinical problem area? The potential economic implications for both the consultant and the practicing physicians are enormous. Not only may the programs infringe directly on the physician’s duties, but, by providing decision support for individuals less highly trained than physicians, may contribute to a reorganization of responsibilities among allied health personnel.

Concerns are also often voiced regarding the effect of such programs on medical education [Schwartz, 1970]. It is not uncommon
to hear the suggestion that such programs will remove the motivation for both doctors and medical students to think or read since they will always know that there is a computer program to help them out if there is something they do not know. Schwartz even suggests that the kind of student attending medical school could change because the primary focus of medical training might become the management of a patient’s emotional needs.

Partially because the public image of computers has grown to encompass visions of massive data banks monitoring the daily lives of the public, physicians often express concern that computers capable of making decisions will be used to monitor their medical practice. In an age when federal legislation is already threatening the sacred privacy of the individual physician entrepreneur, technical innovations that could potentially automate the peer review process are especially threatening (see, for example, the discussion of MYCIN’s possible extension into the monitoring arena, § 6.6).

Finally there are enormous legal questions that remain essentially unanswered at present. Who is culpable if a physician follows a computer’s advice and the patient’s condition worsens, especially under circumstances when a panel of experts agree that an alternate therapy would have been preferable? Must program designers assume legal responsibilities for their system’s mistakes, or does the physician assume ultimate responsibility when he follows a program’s advice?

I have proposed a sufficient number of potentially serious questions that you may have begun to wonder whether research in computer-based medical decision making should be encouraged to continue at all! Let us step back for a moment, however, to ask how many of the itemized concerns are valid and how many are the result, rather, of misunderstanding on the part of physicians and the public or of poor public relations efforts on the part of system designers.

Perhaps the most important point to note initially is that many of the programs have been developed in response to a well-demonstrated need. Despite the availability of expert consultants in university environments, the expertise of specialists is either unavailable or over-taxed in many parts of the country. As a result, local physicians are often forced to make decisions that are less than optimal. Furthermore, even experts may find it difficult adequately
Introduction

to incorporate their experience with several thousand patients into coherent diagnostic strategies. In this sense programs with access to large data bases are potentially useful for physicians at all levels of experience.

Secondly, developers of decision support programs must make it clear, both from their system design and from the tone and content with which they present their work to the medical community, that computer programs for medical decision making are meant to be tools for the physician, not crutches to replace his own clinical reasoning. There is no reason that a computer-based consultation need be any more threatening than a chest xray or a battery of tests from the clinical chemistry laboratory. If a consultation program prods the physician to consider a diagnosis or treatment that might otherwise have slipped his mind, it has done a service both to him and to the patient. Patient education on this point is therefore similarly important. An effort must be made to inform the public that, since certain clinical problems are highly complex, the medical care they receive may be better if their physician seeks the unique capabilities of a computer rather than forging headlong into a diagnostic or therapeutic decision that is based solely upon his current knowledge. After all, few patients object to their physician seeking the advice of a human consultant.

The concern regarding the effect of such programs on medical education may be answered by pointing out that consultation systems, if properly designed, have considerable educational side-effects (see Chapter 2). The physician can therefore become more familiar with the problem area and its important considerations after each consultation session. The result is a growing body of knowledge that may gradually decrease the physician’s need for the program’s advice. A consultation program’s success could in fact be measured in part by the tendency for physicians to become decreasingly reliant upon the system.

What of the specialist’s concern that consultation programs will take over his role? There is some basis for this worry because computer-based consultations are likely to be less expensive than consultations with human experts. However, it is likely that most physicians will prefer the advice of fellow doctors when the experts are readily available. The greatest contribution of computer programs is therefore apt to arise at odd hours when consultants are not
MYCIN

accessible (even the specialists may welcome programs that can assume their roles at 4 a.m.!) or in rural or other nonuniversity environments where the expertise simply does not exist. Furthermore, in an era when the shortage of doctors and their maldistribution are reaching crisis proportions [Fein, 1967; Schwartz, 1970], computer innovation that encourages reallocation of health care responsibilities among medical personnel may perhaps be viewed more as a social boon than an economic threat to physicians.

Even the concerns regarding automated monitoring of physicians' habits may be largely overinflated. In § 6.6 a model is proposed for prospective peer review monitoring that could avoid the threats of retrospective punitive actions on the part of utilization review and medical audit committees. The latter practices are abhorrent to many physicians and partially account for organized medicine's opposition to recent legislation that sets up mandatory peer review mechanisms.

Finally, the questions of legal responsibility are difficult ones to answer since the judicial precedents are not yet well established [Hall, 1972]. However, it seems likely that if the consultation programs are designed to serve as decision tools rather than replacements for the physician's own reasoning processes, the responsibility for accepting or rejecting the computer's advice will probably rest with the physician himself. A more complicated problem arises if a physician diagnoses or treats incorrectly after failing to use a computer program that was readily available to him. Despite the legal questions raised, the potential benefits of decision making programs seem sufficiently large that unanswered judicial concerns should not be allowed to interfere with progress in the field.

1.4 Antimicrobial Selection

1.4.1 (**) NATURE OF THE DECISION PROBLEM

An antimicrobial agent is any drug designed to kill bacteria or to arrest their growth. Thus the selection of antimicrobial therapy refers to the problem of choosing an agent (or combination of agents) for use in treating a patient with a bacterial infection. The terms "antimicrobial" and "antibiotic" are often used interchangeably, although the latter actually refers to any one of a number of drugs that are
isolated as naturally occurring products of bacteria or fungi. Thus, the well-known penicillin mold is the source of an antibiotic, penicillin, that is used as an antimicrobial. Some antibiotics are too toxic for use in treating infectious diseases but are still used in research laboratories (e.g., dactinomycin) or in cancer chemotherapy (e.g., daunomycin). Furthermore, some antimicrobials (such as the sulfonamides) are synthetic drugs and are therefore not antibiotics. There are also semi-synthetic antibiotics (e.g., methicillin) that are produced in chemical laboratories by manipulating a naturally occurring antibiotic molecule. Throughout this text, I shall not rely upon this formal distinction between “antimicrobial” and “antibiotic” but will, rather, use the terms as though they were synonymous. The following list of commonly used antimicrobial agents will introduce you to the names of several of these agents. The list includes many of the generic drugs (i.e., nonbrand names) with which the MYCIN system is familiar:

- ampicillin
- bacitracin
- carbenicillin
- cephalothin
- chloramphenicol
- clindamycin
- colistin
- erythromycin
- ethambutal
- gentamicin
- INH
- kanamycin
- methicillin
- nalidixic-acid
- nitrofurantoin
- PAS
- penicillin
- polymyxin
- rifampin
- streptomycin
- sulfisoxazole
- tetracycline
- vancomycin

This list does not include the several nonbrand name antimicrobials that are chemically related to the generic drugs above but that have some distinctive feature such as a different preferred route of administration.

The name MYCIN is taken from the common suffix shared by several of the antimicrobial agents. It reflects the central concern of the program, namely the selection of an appropriate therapeutic regimen for a patient with a bacterial infection. MYCIN does not yet consider infections caused by viruses or pathogenic fungi, although these other kinds of organisms cause significant diseases that may be difficult to distinguish clinically from disorders with bacterial etiology.

Antimicrobial selection would be a trivial problem if there were a single nontoxic agent effective against all bacteria capable of causing
human disease. However, drugs that are highly useful against certain bacteria are often not the least effective against others. The identity (genus) of the organism causing an infection is therefore an important clue for deciding what drugs are apt to be beneficial for the patient. The following list summarizes the organisms with which MYCIN is familiar. Subtypes are specified only in those cases where the subdivisions have important therapeutic implications:

arizona                      mycobacterium-tb
bacteroides                  neisseria-gonorrhea
borrelia                      neisseria-meningitidis
brucella                      neisseria-species
citrobacter                  pasteurella
clostridium-botulinum        peptococcus
clostridium-species          proteus-mirabilis
clostridium-tetani           proteus-non-mirabilis
corynebacteria-diphtheriae    providence

corynebacteria-species       pseudomonas

diplococcus-pneumoniae       salmonella

e.coli
edwardsiella
enterobacter
fusobacterium
hafnia
hemophilus-influenzae
hemophilus-non-influenzae
herellea
klebsiella
listeria
mima
moraxella
mycobacterium-atypical
mycobacterium-balnei
mycobacterium-leprae
mycobacterium-tb
neisseria-gonorrhea
neisseria-meningitidis
neisseria-species
pasteurella
peptococcus
proteus-mirabilis
proteus-non-mirabilis
providence
pseudomonas
salmonella
serratia
shigella
staphylococcus-coag+
staphylococcus-coag–
streptobacillus
streptococcus-alpha
streptococcus-anaerobic
streptococcus-beta(group-A)
streptococcus-beta(non-group-A)
streptococcus-gamma
streptococcus-group-D
streptococcus-microaerophilic
treponema
vibrio

Selection of therapy is a four-part decision process. First, the physician must decide whether the patient has a significant bacterial infection requiring treatment. If there is significant disease, the organism must be identified or the range of possible identities must be inferred. The third step is to select a set of drugs that may be appropriate. Finally, the most appropriate drug or combination of
drugs must be selected from the list of possibilities. Each step in this decision process is described below.

1.4.1-1 (** Is The Infection Significant?**)

The human body is normally populated by a wide variety of bacteria. Organisms can invariably be cultured from samples taken from a patient’s skin, throat, or stool. These normal flora are not associated with disease in most patients and are, in fact, often important to the body’s homeostatic balance. The isolation of bacteria from a patient is therefore not presumptive evidence of significant infectious disease.

Another complication is the possibility that samples obtained from normally sterile sites (such as the blood, cerebrospinal fluid, or urinary tract) will become contaminated with external organisms either during the collection process itself or in the microbiology laboratory where the cultures are grown. It is therefore often wise to obtain several samples and to see how many contain organisms that may be associated with significant disease.

Because the patient does have a normal bacterial flora and contamination of cultures may occur, determination of the significance of an infection is usually based upon clinical criteria. Does the patient have a fever? Is he coughing up sputum filled with bacteria? Does he have skin or blood findings suggestive of serious infection? Is his chest x-ray normal? Does he have pain or inflammation? These and similar questions allow the physician to judge the seriousness of the patient’s condition and often explain why the possibility of infection was considered in the first place.

1.4.1-2 (**) What is the Organism’s Identity?**

There are a variety of laboratory tests that allow an organism to be identified. The physician obtains a sample from the site of suspected infection (e.g., a blood sample, an aspirate from an abscess, a throat swabbing, or a urine collection) and sends it to the microbiology laboratory for culture. There the technicians first attempt to grow organisms from the sample on an appropriate nutritional medium. Early evidence of growth may allow them to report the morphological and staining characteristics of the organism. However, complete
testing of the organism so that a definite identity is determined usually requires 24–48 hours or more.

The problem with this identification process is that the patient may be sufficiently ill at the time when the culture is first obtained that the physician cannot wait two days before he begins antimicrobial therapy. Early data regarding the organism's staining characteristics, morphology, growth conformation, and ability to grow with or without oxygen may therefore become crucially important for narrowing down the range of possible identities. Furthermore, historical information about the patient and details regarding his clinical status may provide additional useful clues as to the organism's identity.

1.4.1-3 (** What are the Potentially Useful Drugs?

Even once the identity of an organism is known with certainty, its range of antimicrobial sensitivities may be unknown. For example, although a pseudomonas is usually sensitive to gentamicin, an increasing number of gentamicin-resistant pseudomonads are being isolated. For this reason the laboratory will often run in vitro sensitivity tests on an organism they are growing, exposing the bacterium to several commonly used antimicrobial agents. This sensitivity information is reported to the physician so that he will know those drugs that are likely to be effective in vivo.

Sensitivity data do not become available until one or two days after the culture is obtained, however. The physician must therefore often select a drug on the basis of his list of possible identities plus the antimicrobial agents that are statistically likely to be effective against each of the identities. These statistical data are available from many hospital laboratories (e.g., 82% of E.coli isolated at Stanford Hospital are sensitive in vitro to kanamycin) although, in practice, physicians seldom use the probabilistic information except in a rather intuitive sense (e.g., “Most of the E.coli infections I have treated recently have responded to kanamycin”).

1.4.1-4 (** Which Drug is Best for This Patient?

Once a list of drugs that may be useful has been considered, the best regimen is selected on the basis of a variety of factors. These
Introduction

include not only the likelihood that the drug will be effective against the organism, but a number of clinical considerations. For example, it is important to know whether the patient has any drug allergies or whether the drug is contraindicated because of his or her age, sex, or kidney status [Kovnat, 1973]. If the patient has meningitis or brain involvement, does the drug cross the blood-brain barrier? Since some drugs can be given only orally, intravenously (IV), or intramuscularly (IM), the desired route of administration may become an important consideration. The severity of the patient’s disease may also be important, particularly for those drugs for which use is restricted on ecological grounds [Finland, 1970; Rose, 1968] or which are particularly likely to cause toxic complications. Furthermore, as the patient’s clinical status varies over time and more definitive information becomes available from the microbiology laboratory, it may be wise to change the drug of choice or to modify the recommended dosage regimen.

1.4.2 EVIDENCE THAT ASSISTANCE IS NEEDED

The “antimicrobial revolution” began with the introduction of the sulfonamides in the 1930’s and penicillin in 1943. The beneficial effects that these and subsequent drugs have had on mankind cannot be overstated. However, as early as the 1950’s it became clear that antibiotics were often being misused. A study of office practice involving 87 general practitioners [Peterson, 1956] revealed that antibiotics were given indiscriminately to all patients with upper respiratory infections by 67% of the physicians, while only 33% ever tried to separate viral from bacterial etiologies. Despite attempts to educate physicians regarding this kind of inappropriate therapy, similar data are reported even today [Kunin, 1973].

Antibiotic misuse has recently received wide attention [Scheckler, 1970; Roberts, 1972; Kunin, 1973; Simmons, 1974; Carden, 1974]. The studies have shown that very few physicians go through the methodical decision process that I described in § 1.4.1. In the outpatient environment antibiotics are often prescribed without the physician having identified or even cultured the offending organism [Kunin, 1973]. In 1972, the FDA certified enough of the commonly used antibiotics (2,400,00 kg) to treat two illnesses of average duration for every man, woman, and child in the country. Yet it has
been estimated that the average person has an illness requiring antibiotic treatment no more often than once every 5 to 10 years [Kunin, 1973]. Part of the reason for such overprescribing is the patient’s demand for some kind of prescription with every office visit [Muller, 1972]. It is difficult for many physicians to resist such demands, so improved public education is one step toward lessening the problem.

However, antibiotic use is widespread among hospitalized patients as well. Studies have shown that, on any given day, one third of the patients in a general hospital are receiving at least one systemic antimicrobial agent [Roberts, 1972; Scheckler, 1970; Resztak, 1972]. The monetary cost to both patients and hospitals is enormous [Reimann, 1966; Kunin, 1973]. Simmons and Stolley have summarized the issues as follows [Simmons, 1974]:

1. Has the wide use of antibiotics led to the emergence of new resistant bacterial strains?
2. Has the ecology of “natural” or “hospital” bacterial flora been shifted because of antibiotic use?
3. Have nosocomial (i.e., hospital-acquired) infections changed in incidence or severity due to antibiotic use?
4. What are the trends of antibiotic use?
5. Are antibiotics properly used in practice?
   - Is there evidence that prophylactic use of antibiotics is harmful, and how common is it?
   - Are antibiotics often prescribed without prior bacterial culture?
   - When cultures are taken, is the appropriate antibiotic usually prescribed and correctly used?
6. Is the increasingly more frequent use of antibiotics presenting the medical community and the public with a new set of hazards that should be approached by some new administrative or educational measures?

These authors, after stating the issues, proceed to cite evidence that indicates that each of these questions has frightening answers. They show that the effects of antibiotic misuse are so far-reaching that the consequences may often be worse than the disease (real or imagined) being treated!

Our principal concern is with the fifth question, i.e., whether or not physicians are rational in their prescribing habits and, if not, why not? Roberts and Visconti examined these issues in 1,035 patients
Introduction

consecutively admitted to a 500-bed community hospital [Roberts, 1972]. Of 340 patients receiving systemic antimicrobials, only 36% were treated for infection. The rest received either prophylactic therapy (56%) or treatment for symptoms without verified infection (10%). A panel of expert physicians and pharmacists evaluated these therapeutic decisions and only 13% were judged to be rational whereas 66% were assessed as clearly irrational. The remainder were said to be questionable.

Of particular interest were the reasons that therapy was judged to be irrational in those patients for whom some kind of antimicrobial therapy was warranted. This group consisted of 112 patients—50.2% of the 223 patients who were treated irrationally. It is instructive to list the reasons that were cited, along with the corresponding percentages indicating how many of the 112 patients were involved:

<table>
<thead>
<tr>
<th>Reason</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Antimicrobial contraindicated in patient</td>
<td>7.1</td>
</tr>
<tr>
<td>Patient allergic</td>
<td>2.7</td>
</tr>
<tr>
<td>Inappropriate sequence of antimicrobials</td>
<td>26.8</td>
</tr>
<tr>
<td>Inappropriate combination of antimicrobials</td>
<td>24.1</td>
</tr>
<tr>
<td>Inappropriate antimicrobial used to treat condition</td>
<td>62.5</td>
</tr>
<tr>
<td>Inappropriate dose</td>
<td>18.7</td>
</tr>
<tr>
<td>Inappropriate duration of therapy</td>
<td>9.8</td>
</tr>
<tr>
<td>Inappropriate route</td>
<td>3.6</td>
</tr>
<tr>
<td>Culture and sensitivity needed</td>
<td>17.0</td>
</tr>
<tr>
<td>Culture and sensitivity indicate wrong antibiotic being used</td>
<td>16.1</td>
</tr>
</tbody>
</table>

The percentages sum to more than 100% because each therapy may have been judged inappropriate for more than one reason. Thus 62.5% of the 112 patients who required antimicrobial therapy but were treated irrationally were given a drug that was inappropriate for the patient’s clinical condition. This observation reflects the need for improved therapy selection in patients requiring therapy. This is precisely the decision task with which MYCIN is designed to assist.

The hospital at which Roberts and Visconti conducted their study is certainly not the only institution at which physicians tend to prescribe antimicrobials inappropriately. Macaraeg et al. have also reported serious disagreement between some of the practices and opinions of hospital physicians and those of infectious disease experts practicing at the same institution [Macaraeg, 1971]. Recent
review articles [Kunin, 1973; Simmons, 1974] have cited additional studies that have shown similar data.

Now that a need for improved continuing medical education in antimicrobial selection is recognized, there are a variety of valid ways to respond. One is to offer appropriate post-graduate courses for physicians. Another is to introduce surveillance systems for the monitoring and approval of antibiotic prescriptions within hospitals [Edwards, L., 1972; Kunin, 1973]. In addition, physicians should be encouraged to seek consultations from infectious disease experts when they are uncertain how best to proceed with the treatment of a bacterial infection. Finally, an automated consultation system that can substitute for infectious disease experts when they are unavailable or inaccessible could provide a valuable component of the solution to the therapy selection problem. The computer program described in the remainder of this report is an attempt to fill that need.

1.5 MYCIN System

1.5.1 SYSTEM'S ORGANIZATION

MYCIN is an evolving computer program that has been developed to assist physicians who are not experts in the field of the antimicrobials with the decision task discussed in § 1.4.1. Work on the system began early in 1972 when it was recognized that the Stanford community could provide the professional and computing resources necessary for attempting a partial solution to the problem of antibiotic misuse that was discussed in § 1.4.2. The project has involved both physicians, with expertise in the clinical pharmacology of bacterial infections, and computer scientists, with interests in artificial intelligence and medical computing.

The computing techniques used in the development of MYCIN were formulated over several months as the collaborators met in weekly meetings and discussed representative case histories of patients with infections. It was decided to concentrate initially on the process of selecting therapy for patients with bacteremia (i.e., bacteria in the blood). This remains our primary focus to date. As patients with bacteremia were discussed by the clinicians, the project members tried to identify the semi-formal decision criteria that were
being used. It gradually became clear that these criteria, once defined, can be expressed as rules that reflect the knowledge of the experts. Thus, MYCIN was developed as a program that could efficiently utilize such rules in an attempt to model the decision processes of the experts from whom they were obtained.

The discussion in § 1.4.1 pointed out that there are four parts to the process of selecting antimicrobial therapy. MYCIN must accordingly follow each of these steps when giving advice to a physician. To reiterate, decision rules have been sought that allow the program to do the following:

1. decide whether the patient has a significant infection;
2. determine the likely identity of the offending organism;
3. decide what drugs are apt to be effective against this organism;
4. choose the drug that is most appropriate given the patient's clinical condition;

Approximately 200 such decision rules have been identified to date. This corpus of rules is termed the “knowledge-base” of the MYCIN system.

System knowledge must be contrasted with MYCIN’s “data base.” MYCIN uses two kinds of data when it gives advice. Information about the patient under consideration is termed “patient data.” These data are entered by the physician in response to computer-generated questions during the consultation. “Dynamic data,” on the other hand, are the data structures created by MYCIN during the consultation—the deductions it makes and an ongoing record of why these conclusions were reached. This distinction between MYCIN’s knowledge-base and data base should be understood because the terms are used in their respective senses throughout this text.

The program itself consists of three subcomponents, each of which performs a specialized task. Subprogram 1 is the Consultation System, that portion of MYCIN that asks questions, makes conclusions, and gives advice. Subprogram 1 is the subject of Chapter 3.

Subprogram 2 is the Explanation System, the component of MYCIN that answers questions from the user and attempts to explain its advice. The need for such a capability is discussed in Chapter 2, and Chapter 5 explains the implementation details of the explanation capability.
MYCIN

Subprogram 3, the most recent addition to MYCIN, is the Rule-Acquisition System. This module permits experts to teach MYCIN new decision rules or to alter pre-existing rules that are judged to be inadequate or incorrect. Chapter 3 also discusses the need for this kind of capability. Since this subprogram presently exists only in preliminary form, its current capabilities and plans for future extensions are discussed in § 6.3 in the chapter describing future work.

Figure 1-1 is an overview of the three subprograms and the way in which they access MYCIN’s knowledge and data. The heavy arrows indicate the system’s flow of control between the subprograms, while the light arrows represent information flow between program components and MYCIN’s knowledge and data.

Figure 1-1: Diagram demonstrating the flow of control and the flow of information within the MYCIN system. The three subprogram components are enclosed in boxes. Control passes from one subprogram to another as shown by the heavy arrows. Light arrows indicate program access to information used by the system. The program’s knowledge-base is contained in the corpus of rules shown on the right. The way in which the Consultation System uses such rules is described in Chapter 3. [Reproduced from Computers and Biomedical Research [Shortliffe, 1975b] with permission of the publishers.]
Introduction

The physician begins an interactive session by starting the Consultation System (Subprogram 1). When MYCIN asks questions, the physician enters patient data as indicated in Figure 1-1. MYCIN uses its knowledge-base to process this information and to decide what question to ask next. Whenever a conclusion is made, MYCIN saves the information in its dynamic data structure. If the physician wants to interrupt the consultation in order to ask questions, he may enter the Explanation System (Subprogram 2). After the question-answering session, he returns to Subprogram 1 and the consultation proceeds from the point of digression.

When MYCIN is through asking questions, it gives its therapeutic recommendation, and control then automatically passes to Subprogram 2. At this point the physician may ask questions regarding the consultation and how MYCIN reached its decisions. This feature forces MYCIN to justify its conclusions and permits the physician to reject the program’s advice if he feels that some step in the reasoning process has been unsound.

Subprogram 3 is an option available to experts with whom the system is familiar. If an expert (when using Subprogram 2) notes an invalid, incomplete, or missing rule, he may enter the Rule-Acquisition System in order to teach MYCIN the new information. This new knowledge is then incorporated into the corpus of rules so that it will be available to Subprogram 1 during future consultation sessions. As noted above, this feature currently exists only in rudimentary form.

Throughout all three subprograms there are a variety of features designed to heighten MYCIN’s acceptability to physicians. For example, the system is quite tolerant of spelling or typographical errors, and Subprograms 2 and 3 permit the physician to communicate with MYCIN in the language of clinical medicine rather than some specialized computer language. The need for these kinds of human-engineering considerations is discussed in Chapter 2, and the details are described in the chapters that explain each of the subprograms.

1.5.2 SAMPLE CONSULTATION SESSION

This chapter closes with a sample consultation session that illustrates MYCIN’s consultation and explanation capabilities. The following comments will help you follow the sample interactive session:
MYCIN

(1) MYCIN asks questions and makes comments in mixed upper- and lower-case letters;
(2) MYCIN answers questions and displays decision rules using upper-case letters only;
(3) The physician types words in upper-case letters and these phrases always follow MYCIN’s prompt character (**), the double asterisk (note the asterisks distinguish words typed by the physician from upper-case phrases printed by MYCIN);
(4) Explanatory comments in italics have occasionally been inserted in square brackets at the extreme right margin of the page.

The entire sample consultation session, including the question-answering interaction, required approximately 20 minutes at a computer terminal.

*****Welcome from MYCIN*****

Instructions?: (Yes or No)

**YES

I am a computer program named MYCIN that has been designed to advise you regarding an appropriate choice of infectious disease therapy. I understand that you have a patient from whom a possibly positive culture (called CULTURE-1) has been obtained. Please answer the following questions, terminating each response with RETURN.

If you are not certain of your answer, you may modify the response by inserting a certainty factor (a number from 1 to 10) in parentheses after your response. Absolute certainty (10) is assumed for every unmodified answer.

If you wish to change an answer to a previous question, enter CHANGE followed by the number(s) of the question(s) to be altered. Try to avoid going back, however, because the process requires reconsidering the patient from the beginning and it may therefore be slow.

Note that you may also enter UNK (for UNKnown) if you do not know the answer to a question, ? if you would like to see some examples of recognized responses, ?? if you want to see all recognized responses, the word RULE if you would like to see the decision rule which has generated the question being asked, the word WHY if you would like to see a more detailed explanation of the question, or the letters QA if you would like to interrupt the consultation in order to ask questions regarding the current
status of the program's reasoning. If you are ever puzzled about what options are available to you during a consultation, enter the word HELP and a list of options will be listed for you.

Sample Response
Did ORGANISM-1 grow in clumps, chains, or pairs?
**CHAINS (7) PAIRS (3)

SUMMARY OF USER OPTIONS:
UNK - answer not known
? - request for examples of recognized responses
?? - request for list of all recognized responses
RULE - request for current decision rule
QA - request for enter question-answering mode
WHY - request for detailed explanation of the question being asked
CHANGE XXX - go back and re-request answer to question number XXX
STOP - Halts the program without completing the consultation
HELP - Prints this list

[At this point the Consultation System (Subprogram 1) begins.]

7-JUN-74 16:27:27
--------PATIENT-1--------
1) Patient's name: (first-last)
**JEREMIAH SAMPLE
2) Sex:
**MALE
3) Age:
**60
I will refer to the most recent positive culture for which you desire therapeutic advice as:

--------CULTURE-1--------
4) From what site was the specimen for CULTURE-1 taken?
**BLOOD
=BLOOD
5) How many days ago was this blood culture (CULTURE-1) obtained?
**2
I will refer to the first offending organism from this blood culture (CULTURE-1) as:

--------ORGANISM-1--------
6) Enter the identity (genus) of ORGANISM-1:
**UNK
7) Staining characteristics of ORGANISM-1 (gram):
**?
WHAT IS THE STAIN OF ORGANISM-1?
Expected responses are: ACIDFAST GRAMPOS GRAMNEG
Enter HELP for list of user options.
**GRAMNEG
8) Is ORGANISM-1 a rod or coccus (etc.):
**ROD
9) Were any organisms that were significant (but no longer require therapeutic attention) isolated within the last approximately 30 days?
**YES

--------CULTURE-2--------
10) From what site was the specimen for CULTURE-2 taken?
**SPUTUM
11) How many days ago was this upper-resp culture (CULTURE-2) obtained?
**4 1/2 DAYS
PLEASE USE DECIMAL NUMBERS FOR FRACTIONS
4 1/2 DAYS NOT A RECOGNIZED RESPONSE
WHAT IS THE NUMBER OF DAYS SINCE THE SPECIMEN FOR THIS UPPER-RESP CULTURE WAS ACQUIRED?
Expected responses are: NUMBER OF DAYS
Enter HELP for list of user options.
**4.5
I will refer to the prior organism from this upper-resp culture (CULTURE-2) as:

--------ORGANISM-2--------
12) Enter the identity (genus) of ORGANISM-2:
**PSEUDOMONAS
13) Any other significant organisms isolated from this upper-resp culture (CULTURE-2)?
**YES
Introduction

14) Enter the identity (genus) of ORGANISM-3:
**KLEBSIELLA
15) Any other significant organisms isolated from this upper-resp culture (CULTURE-2)?
**YES

16) Enter the identity (genus) of ORGANISM-4:
**ENTEROB
17) Any other significant organisms isolated from this upper-resp culture (CULTURE-2)?
**NO
18) Any other significant earlier cultures from which pathogens were isolated?
**NO
19) What is the suspected non-sterile portal of entry of ORGANISM-1 into this sterile site?
**RULE

RULE121

IF: 1) THE SITE OF THE CULTURE IS ONE OF: THOSE SITES THAT ARE NORMALLY STERILE, AND
2) THE PORTAL OF ENTRY OF THE ORGANISM IS THE SAME AS THE SITE OF A PRIOR ORGANISM WITH THE SAME IDENTITY AS THIS CURRENT ORGANISM

THEN: THERE IS FURTHER SUGGESTIVE EVIDENCE (.3) THAT EACH OF THEM IS A PRIOR ORGANISM WITH THE SAME IDENTITY AS THIS CURRENT ORGANISM

**GU
20) Did ORGANISM-1 grow aerobically?
**YES
21) Did ORGANISM-1 grow anaerobically?
**NO
22) Was any chemotherapy recently begun specifically to combat ORGANISM-1?
**QA
Mycin

[The physician indicates that he wants to enter the Explanation System (Subprogram 2) for a question-answering (QA) session.]

Type HELP for the list of question options.

** IQ WHAT DO YOU THINK THE IDENTITY OF ORGANISM-1 IS AT THIS POINT?**

THE IDENTITY OF ORGANISM-1 IS PSEUDOMONAS (.333) KLEBSIELLA (.333) ENTEROBACTER (.333)

**RETURN**

[Having asked the question of interest, the physician returns to the Consultation System (Subprogram 1).]

22) Was any chemotherapy recently begun specifically to combat ORGANISM-1?

**NO**

23) Is Jeremiah Sample (PATIENT-1) a burn patient?

**RULE**

RULE047

IF: 1) THE SITE OF THE CULTURE IS BLOOD, AND
2) THE IDENTITY OF THE ORGANISM IS NOT KNOWN WITH CERTAINTY, AND
3) THE STAIN OF THE ORGANISM IS GRAMNEG, AND
4) THE MORPHOLOGY OF THE ORGANISM IS ROD, AND
5) THE PATIENT HAS BEEN SERIOUSLY BURNED

THEN: THERE IS WEAKLY SUGGESTIVE EVIDENCE (.4) THAT THE IDENTITY OF THE ORGANISM IS PSEUDOMONAS

**NO**

24) Is the patient’s illness with ORGANISM-1 a hospital-acquired infection?

**YES**

25) Do you consider Jeremiah Sample (PATIENT-1) to be a compromised host? (alcoholic, immunosuppressed, etc.):

**NO**

26) Has Jeremiah Sample (PATIENT-1) had any GU surgery or
**Introduction**

manipulation of the urinary tract? (e.g., indwelling catheter, cystoscopy, etc.)

**WHY**

I AM TRYING TO FIND OUT ABOUT THE IDENTITY OF THE ORGANISM.
I ALREADY KNOW THAT:
THE PORTAL OF ENTRY OF THE ORGANISM IS UPPER-GU
THE MORPHOLOGY OF THE ORGANISM IS ROD
THE STAIN OF THE ORGANISM IS GRAMNEG
THE SITE OF THE CULTURE IS BLOOD

THEREFORE, IF:
1) THE PATIENT HAS NOT HAD A GENITO-URINARY MANIPULATIVE PROCEDURE, AND
2) URINARY-TRACT-INFECTION IS NOT A PROBLEM FOR WHICH THE PATIENT HAS BEEN TREATED

THEN:
THERE IS SUGGESTIVE EVIDENCE (.6) THAT THE IDENTITY OF THE ORGANISM IS E.COLI
(RULE156)

[The WHY option demonstrated here is briefly mentioned in Chapter 2. Its implementation details are not described in this text, however.]

**YES**

27) Does the patient have 'pseudomonas-type' hemorrhagic skin lesions (ecthyma gangrenosum)?
**YES**

28) Was the patient treated for the injection with the enterobacter (ORGANISM-4)?
**NO**

29) Do you feel that the finding of the enterobacter (ORGANISM-4) at this normally non-sterile site is suggestive of disease in this patient?
**NO**

30) Do you believe that the enterobacter (ORGANISM-4) is a contaminant (or normal flora in the case of non-sterile sites)?
**NO**

31) Was the patient treated for the infection with the klebsiella (ORGANISM-3)?
**NO**
32) Do you feel that the finding of the klebsiella (ORGANISM-3) at this normally non-sterile site is suggestive of disease in this patient? **NO
33) Do you believe that the klebsiella (ORGANISM-3) is a contaminant (or normal flora in the case of non-sterile sites)? **UNK
34) Was the patient treated for the infection with the pseudomonas (ORGANISM-2)? **NO
35) Do you feel that the finding of the pseudomonas (ORGANISM-2) at this normally non-sterile site is suggestive of disease in this patient? **YES (8)
36) Has the lab reported the in-vitro antibiotic sensitivities of the pseudomonas (ORGANISM-2)? **NO
37) Has Jeremiah Sample (PATIENT-1) had any operations that you feel may be relevant to the current infection? **YES

-----OPERATION-1------
38) How long ago was OPERATION-1 performed? **11 DAYS
39) Were any antibiotics given during OPERATION-1? **YES

------DRUG-1------
40) What drug? **CEFALOTHIN =CEPHALOTHIN
41) Were any other antibiotics given at the time of OPERATION-1? **NO
42) Any other significant operative procedures? **NO
43) Including this blood culture (CULTURE-1), how many cultures were obtained from this normally sterile site on the same day? **RULE

RULE044

IF: 1) THE SITE OF THE CULTURE IS ONE OF: THOSE SITES THAT ARE NORMALLY STERILE, AND
Introduction

2) THE NUMBER OF CULTURES IN THE SERIES INCLUDING THE CULTURE IS GREATER THAN 1, AND 3) THE NUMBER OF CULTURES CONTAINING THE ORGANISM IS GREATER THAN 1

THEN: THERE IS EVIDENCE (CF= THE QUOTIENT OF THE NUMBER OF CULTURES CONTAINING THE ORGANISM DIVIDED BY THE NUMBER OF CULTURES IN THE SERIES INCLUDING THE CULTURE) THAT THERE IS SIGNIFICANT DISEASE ASSOCIATED WITH THIS OCCURRENCE OF THE ORGANISM

**4

44) How many of the cultures were positive for ORGANISM-1?
**2

[Note that the certainty factor (CF) for RULE044 is calculated from the answers to questions 43 and 44. CF's are explained in Chapter 4.]

45) Any other organisms isolated from this blood culture (CULTURE-1) for which you would like a therapeutic recommendation?
**NO

46) Does Jeremiah Sample (PATIENT-1) show any evidence of infection at a site inaccessible to culture (e.g., brain, bone, liver, etc.)?
**NO

47) Were any additional positive specimens recently drawn from Jeremiah Sample (PATIENT-1) which revealed one or more organisms for which you wish therapeutic advice?
**NO

[MYCIN has now asked all questions necessary in order to decide (a) whether ORGANISM-1 is significant, and (b) the likely identities of ORGANISM-1.]

My therapy recommendation will be based on the following possible identities of the organism(s) that seem to be significant:

<Item 1> The identity of ORGANISM-1 may be PSEUDOMONAS
<Item 2> The identity of ORGANISM-1 may be KLEBSIELLA
<Item 3> The identity of ORGANISM-1 may be ENTEROBACTER
<Item 4> The identity of ORGANISM-1 may be E.COLI
<Item 5> The identity of ORGANISM-1 may be PROTEUS (Subtype unknown)
MYCIN

[MYCIN now begins asking questions to assist with drug selection.]

48) On a scale of 0 to 4, where higher numbers indicate increasing severity, how would you rate Jeremiah Sample (PATIENT-1)'s degree of sickness? **4

49) Does Jeremiah Sample (PATIENT-1) have a clinically significant allergic reaction to any antimicrobial agent? **NO

[Considering GENTAMICIN for use against ORGANISM-1 . . .]

[Rec 1] My preferred therapy recommendation is as follows:
In order to cover for Items <1> <2> <3> <4> <5>:
Give the following in combination:
1 GENTAMICIN
   Dose: 1.7 MG/KG Q8H - IV OR IM
   Comments: MODIFY DOSE IN RENAL FAILURE
2 CARBENICILLIN
   Dose: 25 MG/KG Q2H - IV
   Comments: MODIFY DOSE IN RENAL FAILURE

Do you wish to see the next choice?
**YES
[Considering COLISTIN for use against ORGANISM-1 . . .]
[Rejecting SULFISOXAZOLE for use against ORGANISM-1 . . .]

[Rec 2] Next best therapy recommendation:
In order to cover for Items <1> <2> <3> <4>:
Give: COLISTIN
   Dose: 1.25 MG/KG Q6H - IM
   Comments: MODIFY DOSE IN RENAL FAILURE

Do you wish to see the next choice?
**NO

[The Consultation System (Subprogram 1) is now finished. Control therefore automatically passes to the Explanation System (Subprogram 2).]

Please ask any simple-English questions regarding the above consultation session or the overall knowledge of the consultation system. When discussing the consultation above, please refer to
Introduction

organisms, cultures, etc. by their numeric names (e.g., ORGANISM-1). The names of rules need not be printed in full when using the PR option (‘PR 37’ is okay), but use the whole name in other contexts (e.g., RULE037). Questions may be specified by the numbers that preceded them above. Enter the word STOP when you have no further questions. Certain standard question types may also be preceded by special codes that will speed up the answers to your questions.

Type HELP for the list of question options.

**HELP

** QUESTION-ANSWERING (QA) OPTIONS:
HELP - PRINTS THIS LIST
EQ - REQUESTS AN EXPLANATION OF THE SPECIFIED QUESTION(S) FROM THE CONSULTATION
IQ - PREFIX TO A QUESTION WHICH ASKS ABOUT INFORMATION ACQUIRED BY THE PROGRAM DURING THE CONSULTATION
NO PREFIX - THIS QUESTION QUERIES CONTENTS OF DECISION RULES IN THE SYSTEM
PR - REQUESTS THAT SPECIFIED RULE(S) BE PRINTED
STOP - ESCAPE FROM EXPLANATION SYSTEM
RA - ENTRY TO RULE-ACQUISITION MODULE FOR RECOGNIZED EXPERTS

** IQ WHAT IS THE FINAL DECISION REGARDING THE IDENTITY OF ORGANISM-1?
THE IDENTITY OF ORGANISM-1 IS PSEUDOMONAS (.357) KLEBSIELLA (.177) ENTEROBACTER (.166) E. COLI (.137) PROTEUS (.118) SERRATIA (.042)

** WHAT MADE YOU THINK ORGANISM-1 MIGHT BE A PROTEUS?
I USED RULE084 TO CONCLUDE THAT THE IDENTITY OF ORGANISM-1 IS PROTEUS. THIS GAVE A CUMULATIVE C.F. OF .119. THE LAST QUESTION ASKED BEFORE THE CONCLUSION WAS MADE WAS 24.

** PR 84 163

57
RULE084

IF:  1) THE SITE OF THE CULTURE IS ONE OF: THOSE SITES THAT ARE NORMALLY STERILE, AND  
2) THE LIST OF LIKELY PATHOGENS ASSOCIATED WITH THE PORTAL OF ENTRY OF THE ORGANISM IS KNOWN, AND  
3) THIS ORGANISM AND THE MEMBERS THAT YOU ARE CONSIDERING AGREE WITH RESPECT TO THE FOLLOWING PROPERTIES: GRAM MORPH AIR  
THEN:  THERE IS STRONGLY SUGGESTIVE EVIDENCE (.8) THAT EACH OF THEM IS THE IDENTITY OF THE ORGANISM

RULE163

IF  1) THE MORPHOLOGY OF THE ORGANISM IS ROD, AND  
2) THE STAIN OF THE ORGANISM IS GRAMNEG, AND  
3) THE IDENTITY OF THE ORGANISM IS NOT KNOWN WITH CERTAINTY, AND  
4) THE PATIENT HAS HAD A GENITO-URINARY MANIPULATIVE PROCEDURE  
THEN:  1) THERE IS WEAKLY SUGGESTIVE EVIDENCE (.36) THAT THE IDENTITY OF THE ORGANISM IS PSEUDOMONAS, AND  
2) THERE IS WEAKLY SUGGESTIVE EVIDENCE (.18) THAT THE IDENTITY OF THE ORGANISM IS PROTEUS, AND  
3) THERE IS WEAKLY SUGGESTIVE EVIDENCE (.1) THAT THE IDENTITY OF THE ORGANISM IS SERRATIA, AND  
4) THERE IS WEAKLY SUGGESTIVE EVIDENCE (.18) THAT THE IDENTITY OF THE ORGANISM IS ENTEROBACTER, AND  
5) THERE IS WEAKLY SUGGESTIVE EVIDENCE (.18) THAT THE IDENTITY OF THE ORGANISM IS KLEBSIELLA

** HOW DID YOU DECIDE THAT PSEUDOMONAS WAS THE MOST LIKELY IDENTITY OF ORGANISM-1? 

I USED RULE163 TO CONCLUDE THAT THE IDENTITY OF ORGANISM-1 IS PSEUDOMONAS. THIS GAVE A
Introduction

I USED RULE084 TO CONCLUDE THAT THE IDENTITY OF ORGANISM-1 IS PSEUDOMONAS. THIS GAVE A CUMULATIVE C.F. OF .455. THE LAST QUESTION ASKED BEFORE THE CONCLUSION WAS MADE WAS 24.
I USED RULE040 TO CONCLUDE THAT THE IDENTITY OF ORGANISM-1 IS PSEUDOMONAS. THIS GAVE A CUMULATIVE C.F. OF .833. THE LAST QUESTION ASKED BEFORE THE CONCLUSION WAS MADE WAS 27.

** PR 21

RULE021

IF: THERE IS A PRIOR ORGANISM WHICH MAY BE THE SAME AS THE CURRENT ORGANISM
THEN: IT IS DEFINITE (1) THAT THESE PROPERTIES - IDENTITY - SHOULD BE TRANSFERRED FROM A PRIOR ORGANISM WHICH MAY BE THE SAME AS THE ORGANISM TO THIS CURRENT ORGANISM

** IQ DID YOU DECIDE ORGANISM-2 WAS THE SAME AS ORGANISM-1?

ORGANISM-4 (.19) ORGANISM-3 (.19) ORGANISM-2 (.19) IS A PRIOR ORGANISM WITH THE SAME IDENTITY AS ORGANISM-1

** HOW DID YOU DECIDE THAT ORGANISM-2 WAS THE SAME AS ORGANISM-1?

I USED RULE004 TO CONCLUDE THAT THE PSEUDOMONAS IS A PRIOR ORGANISM WITH THE SAME IDENTITY AS
MYCIN

ORGANISM-1. THIS GAVE A CUMULATIVE C.F. OF .133. THE LAST QUESTION ASKED BEFORE THE CONCLUSION WAS MADE WAS 18.

** PR 45

RULE004

IF: 1) THE IDENTITY OF THE ORGANISM IS NOT KNOWN WITH CERTAINTY, AND 2) THIS CURRENT ORGANISM AND PRIOR ORGANISMS OF THE PATIENT AGREE WITH RESPECT TO THE FOLLOWING PROPERTIES: MORPH GRAM
THEN: THERE IS WEAKLY SUGGESTIVE EVIDENCE (.4) THAT EACH OF THEM IS A PRIOR ORGANISM WITH THE SAME IDENTITY AS THE ORGANISM

RULE005

IF: THIS CURRENT ORGANISM AND A PRIOR ORGANISM WITH THE SAME IDENTITY AS THE ORGANISM AGREE WITH RESPECT TO THE FOLLOWING PROPERTIES: AIR
THEN: THERE IS WEAKLY SUGGESTIVE EVIDENCE (.2) THAT EACH OF THEM IS A PRIOR ORGANISM WITH THE SAME IDENTITY AS THE ORGANISM

** EQ 48

48 QUESTION 48 WAS ASKED IN ORDER TO FIND OUT JEREMIAH SAMPLE'S DEGREE OF SICKNESS (ON A SCALE OF 4) IN AN EFFORT TO EXECUTE RULE068.

** PR 68

RULE068

IF: 1) THE THERAPY UNDER CONSIDERATION IS GENTAMICIN, AND 2) THE IDENTITY OF THE ORGANISM MAY BE PSEUDOMONAS, AND 3) THE PATIENT'S DEGREE OF SICKNESS (ON A SCALE OF 4) IS GREATER THAN 3, AND
Introduction

4) [A - THE PATIENT IS NOT ALLERGIC TO ANTI-BIOTICS, OR B - CARBENICILLIN IS NOT A DRUG TO WHICH THE PATIENT IS ALLERGIC]

THEN: RECORD THE FOLLOWING AS A MORE APPROPRIATE THERAPY THAN GENTAMICIN: THE COMBINATION OF GENTAMICIN WITH CARBENICILLIN

**STOP

50) Do you wish advice on another patient?

**NO

THANK-YOU

*** ***
Chapter 2

Design Considerations for MYCIN†

2.1 Introduction

As discussed in Chapter 1, several computer programs have been written that attempt to model a physician’s decision-making processes. Some of these have stressed the diagnostic process itself [Gorry, 1968a; Warner, 1972b; Wortman, 1972]; others have been designed principally for use as educational tools [Hoffer, 1973; Weinberg, 1973; Harless, 1973]; while still others have emphasized the program’s role in providing medical consultations [Bleich, 1972; Peck, 1973; Kulikowski, 1973]. Actually, each of these applications is inherently interrelated since any program that is aimed at diagnosing disease has potential use for educating and counselling those who lack the expertise or statistical data that have been incorporated into the program. Consultation programs often include diagnosis as a major component although their principal focus involves interactive use by the physician and/or the determination of therapeutic advice.

In general, the educational programs designed for instruction of medical students and other health professionals have met with more long-term success [Wooster, 1973] than has been the case for the diagnostic and consultation programs. The relative success in implementing instructional programs may result because they deal only with hypothetical patients as part of an effort to teach diagnostic and therapeutic concepts, whereas the consultation programs attempt to assist the physician in the management of real patients in

†This chapter is based in large part on a paper presented by the author at the Thirteenth Annual San Diego Biomedical Symposium [Shortliffe, 1974a]. It is reproduced here with permission of the copyright owners.
the clinical setting. A program making decisions that can directly affect patient well-being must fulfill certain responsibilities to the physician if he is to accept the use of the computer and make use of its knowledge. This chapter discusses those clinical responsibilities and specifies the way in which they must be reflected in a system's design; specifically, the ways in which the MYCIN system seeks to satisfy these design considerations are described. Developmental concerns that relate to nonclinical criteria, such as economic, administrative, or legal requirements, are not included in this discussion.

2.2 Design Considerations for Consultation Programs

Physicians will, in general, reject a computer program designed for their use in decision making unless it is accessible, easy to use, forgiving of noncrucial errors from nonexpert typists, reliable, and fast enough to facilitate the physician's task without significantly prolonging the time required to accomplish it. They also require that the program function as a tool to the physician, not as an "all-knowing" machine that analyzes data and then states its inferences as dogma without justifying them.

Those who design computer programs to give advice to physicians should devise solutions to these requirements in an effort to combat the current lack of acceptance of computer-aided diagnosis by the medical profession [Croft, 1972]. The physician is most apt to need advice from such a program when an unusual diagnostic or therapeutic problem has arisen, precisely the circumstances under which the patient is likely to be acutely ill. Time will therefore often be an important consideration in such cases, and the physician may be unwilling to experiment with a program that does not meet the general requirements mentioned above.

With these considerations in mind, we developed the following list of prerequisites for the acceptance of a clinical consultation program. The list is idealistic, and its components are perhaps currently unattainable, but they do serve as useful guides as long-range goals for workers in the field. Each item is discussed in detail below, but a preliminary summary is presented here. In general, a therapeutic or diagnostic consultation program should be:
Design Considerations for MYCIN

(1) useful;
(2) educational when appropriate;
(3) able to explain its advice;
(4) able to understand and respond to simple questions stated in natural language;
(5) able to acquire new knowledge, either through experience or by being told;
(6) easily modified.

These design considerations are related to one another, and the need for each consideration tends to follow from those criteria listed above it. Furthermore, the order of development of capabilities occurs naturally from the bottom to the top of the list; for example, a program may not be able to explain its advice fully until it can respond to simple questions, and a program will not be useful until it can explain its advice. All six considerations, however, are aimed at satisfying those principles that reflect the system’s responsibility to the physician and, through him, to the patient.

2.2.1 PROGRAM SHOULD BE USEFUL

Clearly the ultimate goal of any program is that it be “useful,” and in the case of consultation systems for use by physicians this word has several important implications. Usefulness is measured along three scales:

(1) the need for the assistance that the program provides;
(2) the reliability of the advice;
(3) the mechanics for accessing the machine and retrieving the desired information.

The validity of advice is of crucial importance. The system must give good advice most of the time and must be able to explain itself when it cannot reach a decision. Otherwise, physicians will soon learn that the system is of little practical value and will stop experimenting with it. Evaluative tests should demonstrate that the advice given by the program corresponds to that given by an expert who is provided with the same clinical information, or that the advice is, retrospectively, shown to be valid at least as often as is the advice of
MYCIN

the expert. This requirement means that the program must be given a large amount of knowledge before it is implemented on the hospital wards. In order to insure an accurate data base of clinical knowledge, cooperation and guidance from several experts in the field with which the program is involved is of great importance, and ongoing collaboration with physicians at all levels of system development is even more desirable. Practicing physicians tend to lose interest quickly in an experimental tool that is not clinically useful, even if they are warned that the program is still undergoing developmental work. It is therefore wise to defer implementation until the collaborating experts feel that minimal additional system improvement can be achieved prior to the ultimate test of ongoing clinical use.

The importance of “human-engineering” aspects of program design is often overlooked. Yet ignoring such issues can prevent acceptance of a system which otherwise gives good advice and fulfills the design criteria I have mentioned. In this sense a consultation program is not “useful” unless it is “useable.” Doctors seek mechanisms for saving time without jeopardizing excellence of patient care, so a program that is slow, difficult to access, or frustrating to use will quickly be rejected. Once implemented, the system should be readily available to clinicians who may need its advice on short notice. Care should therefore be taken to provide a sufficient number of terminals so that there need not be lines of physicians waiting for their chance at the program. Furthermore, the user should require minimal training in order to get advice from the system. It is also desirable that system response time be fast and that the time from sign-on to sign-off be kept as short as possible commensurate with the difficulty of the therapeutic or diagnostic problem for which advice is being sought. If the program is not a multiple-choice or light-pen system and therefore requires typing by the physician, the amount of user input should be minimized and misspellings should be tolerated as much as possible. Users without computer experience tend to think that a machine is unintelligent if it cannot realize that “tetracyline” was intended to be “tetracycline,” and a physician will not take kindly to a system that requires that he experiment with two or three spellings until he finds the one with which the program is familiar. Relatively minor issues such as these can make the difference between a successful consultation program, acceptable to clinicians, and one that is not. Minor issues should certainly not be
Design Considerations for MYCIN

ignored until clinical implementation is attempted because the prob­lems can often be solved more easily if they are considered during program development.

2.2.2 PROGRAM SHOULD BE EDUCATIONAL WHEN APPROPRIATE

A physician who seeks advice from a therapeutic consultation program presumably recognizes that he may not have the necessary expertise or data to make the decision on his own. The program will therefore be interacting with an individual who is likely to welcome instructive comments regarding the patient and the way in which the specific therapeutic problem should be approached. However, the physician may not have time for a learning session with the machine. It is therefore not only important that the system be able to explain the knowledge required in order to make an appropriate clinical decision; it should also be sufficiently flexible so that it does not attempt to instruct the user unless requested to do so.

An additional benefit that accompanies the machine’s ability to teach the user about its decision making is the possibility that, when similar clinical circumstances arise in the future, the physician will no longer need to turn to the consultation program. This can help avoid an over-dependence on the machine’s capabilities.

2.2.3 PROGRAM SHOULD BE ABLE TO EXPLAIN ITS ADVICE

In most cases, the educational process I referred to above will be accomplished by having the machine explain the advice it has given. However, explanation serves more than an educational purpose. It also provides the program with a mechanism for justification of decisions; a physician will be more willing to accept a program’s advice if he is able to understand the decision steps that the system has taken. This gives him a basis on which to reject the system’s advice if he finds that the program is not able to justify its decisions sufficiently. It thereby helps the program conform to the physician’s requirement that a consultation system be a tool and not a dogmatic replacement for the doctor’s own decisions. Gorry has also discussed the need for explanation capabilities in diagnostic consultation systems [Gorry, 1973a] and suggests that the lack of such features in

67
Bayesian decision programs [Gorry, 1968a] partially accounts for their limited success when ward implementation has been attempted. Bleich attributes much of the success of his acid-base consultation program [Bleich, 1972] to its ability to discuss both the electrolyte status of the patient and its method for calculating the characteristics of the patient's acid-base disorder.

2.2.4 PROGRAM SHOULD BE ABLE TO UNDERSTAND QUESTIONS

A nonrestrictive mechanism by which the physician can communicate with the program is an important feature of a system designed to explain its decisions and educate the user. This is particularly true if an attempt is made to minimize specialized training for users of the program. Thus the program should be able to understand queries from the physician and it must be able to respond to requests for justification of decisions or machine-generated queries that may be puzzling. Yet few problems have given computer scientists more difficulty than the development of programs that can understand and act upon questions that are presented in natural language. As discussed in § 1.2.1-7, the field of computational linguistics has produced researchers who have approached natural language understanding from several different points of view [Schank, 1972; Winograd, 1972; Woods, 1970], and some investigators have dealt specifically with programs for understanding and answering questions [Simmons, 1970]. These programs have achieved results that are only of limited applicability. It is therefore unlikely that a consultation program developed for use in the clinical setting in the near future will have sophisticated natural language capabilities. Some attempt to solve the problem in a limited sense is appropriate at this time, however, since question-answering is a logical prerequisite for explanatory and educational capabilities.

2.2.5 PROGRAM SHOULD BE ABLE TO ACQUIRE NEW KNOWLEDGE

A program needs to be able to learn new information in any area of medical therapeutics where changes in decision criteria occur with
some regularity. A facility for teaching new knowledge to the system is therefore desirable since expert clinicians are generally the only ones who can determine when the knowledge of the program is outdated or otherwise inadequate. The need for this kind of program reliability was discussed above. There is perhaps no better way to insure the reliability of the program’s knowledge than to permit collaborating experts to experiment with the program during both developmental and implementation stages, to identify weaknesses in the system’s decision criteria, and to make corrections or additions to the program’s knowledge-base. After the program has been implemented in the clinical setting, a knowledge acquisition capability permits the system to continue to improve whenever errors in its decisions are found by an expert familiar with the methods for teaching it the necessary new information.

Realistically, however, few experts in medical therapeutics will have an extensive knowledge of computer programming and the inner workings of the consultation system. It is therefore important to enable the expert to teach the program new decision criteria or information by entering statements in English and letting the program interpret the language and determine how the new data should be incorporated into its knowledge-base. Although the computational problems involved are at least as difficult as those encountered during the question-answering task discussed above, this is a powerful capability that will greatly facilitate growth of the program’s knowledge to a point at which the collaborating experts agree that the time for ward implementation has arrived.

A second kind of self-improvement by the program, and a feature that is more appropriate in some applications (such as therapy advisors) than in others, is the development of mechanisms for monitoring the effects of the system’s advice upon patient welfare and for modifying its decision criteria dynamically in response to such observations. This kind of learning can take place only after implementation in the clinical environment has occurred and only if mechanisms exist for letting the machine know whether the physician has followed its advice and whether the patient has responded as desired to the medication that was administered. The issue should not be ignored during program development, however, because design of data structures and input/output mechanisms may be modified if the future need for such a facility is recognized.
2.2.6 PROGRAM KNOWLEDGE-BASE SHOULD BE EASILY MODIFIED

The need for straightforward system modification follows directly from the desire to permit the program to learn new information and decision criteria directly from the expert. If the teaching process requires intimate knowledge of the system's data base and how it is used, few clinical experts will have the time or inclination to acquire the necessary sophisticated insights into the program. For example, an inference model that depends on a complex decision tree is apt to be difficult to augment without a complete diagram of the tree so that all implications of additions can be observed. A modular system, on the other hand, permits knowledge to be acquired as isolated facts and allows the consultation program itself to decide under what conditions the new information is relevant. This requirement implies a great deal of intelligence in the consultation monitor but avoids the problems that result if the expert is asked to indicate exactly the circumstances under which the information he is offering may be useful.

Modularity of decision criteria also facilitates searches for inconsistencies or contradictions when new information is acquired during the learning process. If all system knowledge is stored in “packets,” comparisons of a new “packet” with those that already exist can be straightforward. Such checks for contradictions are important if the system is to maintain its validity through many teaching sessions, particularly when several experts with different views of the consultation program’s problem area are simultaneously influencing the system’s knowledge.

It is possible that a consultation system can succeed to a certain extent without addressing itself to all of the design criteria just discussed or, on the other hand, that additional criteria need to be added to the list. However, the design considerations outlined in this chapter provide long-range goals that demand attention even for short-range program development since it is likely that the success of consultation programs will be impeded until each of these problems has been solved. MYCIN has been developed with all six design considerations in mind and, although it is not yet implemented for
Design Considerations for MYCIN

ongoing use in the clinical setting, it attempts to solve some of the serious design problems discussed above.

2.3 MYCIN and Acceptability Criteria

Several of MYCIN's interactive capabilities were demonstrated in the sample consideration included in § 1.5.2. In the remainder of this chapter I shall therefore present extracts of an interactive session, rather than an entire consultation, in an effort to point out how MYCIN reflects the six design considerations discussed above. Since the logical order of explanation of the six capabilities is from last to first, MYCIN's approach to each will be discussed in that order here. The programming details, however, will not be presented until Chapters 3, 5, and 6.

2.3.1 MODULARITY TO INSURE STRAIGHTFORWARD MODIFICATION

We accomplished modularity of system knowledge by storing all information in decision rules. These rules are coded in LISP internally, but can be translated into an English language version for communication with the user. For example, a rule that is presented to the physician as:

IF:  1) THE STAIN OF THE ORGANISM IS GRAMPOS, AND
    2) THE MORPHOLOGY OF THE ORGANISM IS COCCUS, AND
    3) THE GROWTH CONFORMATION OF THE ORGANISM IS CLUMPS
THEN: THERE IS SUGGESTIVE EVIDENCE (.7) THAT THE
       IDENTITY OF THE ORGANISM IS STAPHYLOCOCCUS

is actually coded internally as:

PREMISE: ($AND (SAME CNTXT GRAM GRAMPOS)
            (SAME CNTXT MORPH COCCUS)
            (SAME CNTXT CONFORM CLUMPS))
ACTION: (CONCLUDE CNTXT IDENT STAPHYLOCOCCUS TALLY .7)
Since all system knowledge is stored in rules of this sort, and since there is no explicit reference to the way in which one rule relates to the others in the system, the decision rules maintain the modularity that is desirable. The way in which the program monitor (Subprogram 1 shown in Figure 1-1) decides which rules to use during an interactive session is explained in Chapter 3.

2.3.2 ABILITY TO ACQUIRE NEW KNOWLEDGE FROM EXPERTS

When the consultation program learns from the expert, the process is simply one of acquiring a new rule that conveys the information that the expert believes is lacking in the system. MYCIN's first 150 rules were determined over several months of meetings during which the collaborators discussed representative case histories. These rules were coded into LISP by hand and provided the core knowledge-base on which the early version of the consultation program has operated. Many future rules will be acquired interactively from infectious disease experts. Once an expert has determined what information is needed by the program, he indicates to the Rule-Acquisition System (Subprogram 3 shown in Figure 1-1) that he wishes to enter a new rule ("NR"). He is then asked to enter the rule in English, following the format of the other rules in the system. MYCIN attempts to translate the rule into a LISP representation and then uses its LISP-to-English routines to print out a version of the rule as it has understood the meaning. The expert is next asked to approve the new rule or to make modifications as necessary. The following session demonstrates this learning process (user input always follows the double asterisk):

**NR
The new rule will be called RULE200.
If: 1 **THE ORGANISM IS A GRAM NEGATIVE ROD
and 2 **IT IS ANAEROBIC
and 3 **IT WAS ISOLATED FROM THE BLOOD
and 4 **YOU THINK THE PORTAL WAS THE GI TRACT
and 5 **

[user: carriage return with no entry]
Design Considerations for MYCIN

Then: 1 **IT IS PROBABLY A BACTEROIDES**
On a scale of 1 to 10, how much certainty would you affix to this conclusion?
**g**

and 2 **

[User: carriage return with no entry]

This is my understanding of your rule:
RULE200

IF: 1) THE SITE OF THE CULTURE IS BLOOD, AND
2) THE STAIN OF THE ORGANISM IS GRAMNEG, AND
3) THE MORPHOLOGY OF THE ORGANISM IS ROD, AND
4) THE ANAEROBICITY OF THE ORGANISM IS
5) THE PORTAL OF ENTRY OF THE ORGANISM IS GI
THEN: THERE IS STRONGLY SUGGESTIVE EVIDENCE (.9)
THAT THE IDENTITY OF THE ORGANISM IS BACTEROIDES

Okay? (YES or NO)
**YES

A similar procedure is used to alter previously existing rules that the expert finds incomplete or otherwise inadequate. Only after expert clinicians have used these features to teach MYCIN additional knowledge will the validity of the program’s advice reach an adequate level so that implementation in the hospital can be considered.

The mechanism that MYCIN uses for understanding the natural language input of experts avoids syntactic or semantic analysis; it is, rather, a modified key-word approach in which accumulated clues are combined in order to deduce which attributes of organisms, cultures, patients, etc., are being discussed. The technique is further described in Chapter 5.

Once new rules are acquired from the expert, they immediately become available for use by the Consultation System (Subprogram 1). Appropriate checks must be made to insure that the rule does not contradict any previously existing rule, but the expert need not worry about how the rule interrelates with other rules or when the system should use the new knowledge. At present, only a limited
MYCIN

number of rules have been acquired using the scenario demonstrated above (see § 6.3). Future work will concentrate on the problem of letting MYCIN learn rules that discuss attributes with which the system is not already familiar. In such instances, the program's 800-word dictionary is of small value in deducing the meaning of the English-language input and new techniques must be developed so that MYCIN can not only learn new rules of this variety but can also learn about the clinical attributes that are being discussed.

2.3.3 ABILITY TO UNDERSTAND QUESTIONS

MYCIN answers questions about its decisions by retrieving and printing out the relevant rules (Subprogram 2 shown in Figure 1-1). Much of the intelligence of the program thus lies in the portion of the program which finds these "relevant" rules. Questions are answered within 5 to 20 seconds, considerably faster than is the case with more complex natural language question-answering systems in which both syntactic and semantic content must be analyzed. A physician may ask either (a) informational questions (prefaced with the letters "IQ") which query the status of current knowledge about the patient, (b) questions about the deductions of the current consultation, or (c) general questions about any of MYCIN's judgmental rules, whether used in the current consultation or not. The following question–answering examples demonstrate these capabilities:

(a) Informational question:

**IQ WHAT IS THE IDENTITY OF ORGANISM-1?
THE IDENTITY OF ORGANISM-1 IS E.COLI (.43) PROTEUS (.22)
PSEUDOMONAS (.22) KLEBSIELLA (.13)

(b) Questions about the current consultation:

**HOW DID YOU DECIDE THAT ORGANISM-1 MIGHT BE A
**PSEUDOMONAS?
I USED RULE085 TO CONCLUDE THAT THE IDENTITY OF ORGANISM-1 IS PSEUDOMONAS. THIS GAVE A CUMULATIVE CERTAINTY FACTOR OF .6
Design Considerations for MYCIN

(c) General questions:

**PR RULE085
RULE085

IF: 1) THE STAIN OF THE ORGANISM IS GRAMNEG, AND
2) THE MORPHOLOGY OF THE ORGANISM IS ROD, AND
3) THE PATIENT IS A COMPROMISED HOST
THEN: THERE IS SUGGESTIVE EVIDENCE (.6) THAT THE
IDENTITY OF THE ORGANISM IS PSEUDOMONAS

**DO YOU EVER PRESCRIBE CARBENICILLIN FOR PSEUDOMONAS
**INFECTIONS?
Relevant rules: RULE024 RULE068 RULE137 RULE138
Which ones do you wish to see?
** RULE024
RULE024

IF: 1) THE THERAPY UNDER CONSIDERATION IS
GENTAMICIN, AND
2) THE IDENTITY OF THE ORGANISM IS DEFINITELY
PSEUDOMONAS
THEN: RECORD THE FOLLOWING AS A MORE APPROPRIATE
THERAPY THAN GENTAMICIN ALONE: THE
COMBINATION OF GENTAMICIN AND CARBENICILLIN

It is also possible to ask the system (d) to explain questions in
retrospect (e.g., “Why did you ask question 17?”) and (e) to ask for
confirmation of one’s own decision rules (e.g., “Is chloramphenicol
okay for Salmonella infections?”). The implementation details for
these capabilities are the subject of Chapter 5.

2.3.4 ABILITY TO EXPLAIN DECISIONS

It should be clear from the above explanation session that the
ability to answer questions and remember the details of a consulta-
tion provide the mechanism for explaining decisions which may
puzzle the user. Questions of type (b) require that the machine refer
to the specific details of the consultation it has just undertaken in
order to justify its final decisions. This explanation technique allows
MYCIN

the physician to choose those points which he feels need clarification and permits him to avoid questions altogether if he finds the system's advice consistent with his own opinions regarding therapy for the patient.

The question-answering session usually follows a consultation session. However, questions may occur to the physician during the consultation itself and we have therefore implemented two additional explanatory capabilities. One of these allows the user to enter "QA" in response to any question that is asked by MYCIN. This tells the system to digress temporarily from the consultation and to permit the user to ask questions of the type demonstrated in § 2.3.3. In this way, the user can query partial results of the program's reasoning and then return control to MYCIN (by entering the word STOP) so that the consultation may proceed from the point of digression.

The second capability permits the user to demand that MYCIN justify any question that is asked. Whenever a question generated by MYCIN puzzles the physician, he simply enters the word RULE and the program responds by printing out the translation of the decision rule that has generated the current question. After printing out the relevant decision rule, MYCIN repeats its question and the consultation continues unhampered. For example:

17) Are there any factors in this patient which may cause inadequate gastro-intestinal absorption of the penicillin (DRUG-1)?
   **RULE
   RULE049

   IF: 1) THE ROUTE OF ADMINISTRATION OF THE PENICILLIN IS ORAL, AND
       2) THERE IS A GI FACTOR WHICH MAY INTERFERE WITH THE ABSORPTION OF THE PENICILLIN
   THEN: THERE IS SUGGESTIVE EVIDENCE (.6) THAT THE ROUTE OF ADMINISTRATION OF THE PENICILLIN IS NOT ADEQUATE

17) Are there any factors in this patient which may cause inadequate gastro-intestinal absorption of the penicillin (DRUG-1)?
   **NO

A similar capability, the WHY option, provides a more detailed and conversational explanation of the program's reasoning. A discus-
Design Considerations for MYCIN

sion of the complexities of implementation of this and related capabilities, plus a description of their use, may be found elsewhere [Shortliffe, 1975b; Davis, 1976].

2.3.5 EDUCATIONAL CAPABILITIES

As was pointed out in the discussion of the six design considerations, the ability of a consultation program to explain its decisions, and to answer questions about the area of expertise that it is modeling, automatically provides an educational capability. The sample question-answering session and the RULE option demonstrate the variety of ways in which MYCIN educates the user as well as justifies its decisions.

2.3.6 GENERAL USEFULNESS

As has already been stated, the ultimate test of MYCIN’s usefulness and acceptability will come when we finally feel it is ready to be installed in the ward setting. In an effort to prepare for that day, we have tried to develop interactive characteristics that will overcome the standard complaints voiced by physicians who try to use terminal-based systems.

Whenever MYCIN asks a question, it knows the range of possible answers. It therefore compares the physician’s response against the list of recognized responses. If the user’s response is not on the list, it determines whether a simple typographical or spelling error will account for the unrecognized response. If spelling correction is unsuccessful, the system lexicon is checked to see if the user’s answer is a synonym for one of the recognized responses. If this attempt fails, MYCIN prints out a list of recognized responses and asks the question again.

Both spelling-correction and the listing of recognized responses help reduce the level of frustration that can easily alienate novice users of computer systems. Additional features have also been implemented to assist the physician when he is puzzled by a question that MYCIN is asking. If he enters a question-mark ("?") , MYCIN assumes that he would like to see some sample responses. In addition, any question can be answered with the letters UNK (for UNKnown) if the user is uncertain of the answer but wishes MYCIN’s opinion in
spite of the incomplete information. Finally, the RULE and WHY options that have already been mentioned help the user feel comfortable with the system and more inclined to accept MYCIN as the clinical tool it is designed to be.

This chapter has concentrated on explaining why the MYCIN system operates the way that it does. The next three chapters will deal with a description of how these goals have been accomplished. In Chapter 3 the subject is the core consultation program itself (Subprogram 1). Chapter 4 explains the mechanism we have devised for quantification of the program’s decision processes. Then Chapter 5 summarizes MYCIN’s question-answering capabilities (Subprogram 2). The program’s limited ability to learn from experts (Subprogram 3) is included in § 6.3 when I discuss future efforts contemplated for improving MYCIN’s acceptability and for extending its range of uses.
Chapter 3
Consultation System

3.1 Introduction

In this and the succeeding two chapters MYCIN's implementation is presented in considerable detail. My goal is both to explain the data and control structures used by the program and to describe some of the complex and often unexpected problems that arose during system implementation. Less detailed discussions, which provide a general overview of the material in Chapters 3 and 5, may be found elsewhere [Shortliffe, 1973, 1975b]. In Chapter 2 the motivation behind many of MYCIN's capabilities was explained. If you bear those design criteria in mind throughout the remainder of this text, you will see the important role they have played.

This chapter specifically describes the Consultation System (Subprogram 1). As indicated in Figure 1-1, this subprogram uses both system knowledge from the corpus of rules, plus patient data entered by the physician, in order to generate advice for the user. Furthermore, the program maintains a dynamic data base that provides an ongoing record of the current consultation. As a result, this chapter must discuss both the nature of the various data structures and how they are used or maintained by the Consultation System.

Section 3.2 describes the corpus of rules and the associated data structures. It begins by looking at other rule-based systems and proceeds to a formal description of the rules used by MYCIN. Our quantitative truth model is briefly introduced and the mechanism for rule evaluation is explained. This section also describes the clinical parameters with which MYCIN is familiar and which form the basis for the conditional expressions in the PREMISE of a rule.
In § 3.3 MYCIN’s goal-oriented control structure is described; mechanisms for rule invocation and question selection are explained at that time. The section also discusses the creation of the dynamic data base that is the foundation for both the system’s advice and its explanation capabilities as described in Chapter 5.

Section 3.4 is devoted to an explanation of the program’s context tree, i.e., the network of interrelated organisms, drugs, and cultures that characterize the patient and his current clinical condition. The need for such a data structure is clarified and the method for propagation (growth) of the tree is described.

As discussed in § 1.5.1, the final tasks in MYCIN’s clinical problem area are the identification of potentially useful drugs and the selection of the best drug or drugs from that list. MYCIN’s mechanism for making these decisions is discussed in § 3.5.

Section 3.6 discusses MYCIN’s mechanisms for storing patient data and for permitting a user to changes the answer to a question. As will be described, these two capabilities are closely interrelated.

In § 3.7, I briefly mention some contemplated future extensions to the system. The concluding section then summarizes the advantages of the MYCIN approach, making comparisons with previous work in both AI and medical decision making.

3.2 System Knowledge

3.2.1 DECISION RULES

Automated problem-solving systems use criteria for drawing conclusions that often support a direct analogy to the rule-based knowledge representation used by MYCIN. Consider, for example, the conditional probabilities that underlie the Bayesian diagnosis programs discussed in § 1.3.4. Each probability statement provides information that may be expressed in an explicit rule format:

\[ P(h|e) = X \]

means:

IF: E IS KNOWN TO BE TRUE
THEN: CONCLUDE THAT H IS TRUE WITH PROBABILITY X
Consultation System

The advantages of an explicit rule format are discussed in Section 4.3.

It is important to note, however, that the concept of rule-based knowledge is not unique, even for medical decision making programs. As will be explained, MYCIN's innovation rests with its novel application of representation techniques and goal-oriented control structures that have been developed by AI researchers. The contributions of the program to AI and medical decision making are summarized in Chapter 7.

3.2.1-1 Previous Rule-Based Systems

The need for representation of knowledge in IF-THEN format so pervades problem-solving in AI that many AI programs can be interpreted as rule-based systems once we recognize that all deductive or inferential statements are, in effect, decision rules. In fact, several of the new AI languages have provided data structures and control structures based on rules (theorems) [Bobrow, 1973]. For example, PLANNER [Hewitt, 1969, 1971, 1972] provides a formalism for the statement of theorems such as:

(CONSEQUENT
 (PART $?X $?Z)
 (GOAL (PART $?X $?Y))
 (GOAL (PART $?Y $?Z)))

This theorem simply states, in rule form, that:

IF: YOU CAN FIND AN X THAT IS PART OF A Y, AND
YOU CAN FIND A Z SUCH THAT THE Y IS PART OF THE Z
THEN: YOU CAN CONCLUDE THAT THE X IS PART OF THE Z

Although there are several examples of AI programs that use some variety of rule-based knowledge, only four representative cases will be introduced here. The control structures used for processing the "rules" in these systems are not discussed until § 3.3.1.

The first example is the theorem-proving question–answering program named QA3 [Green, 1969]. As was pointed out in the example from PLANNER above, a theorem may be considered a rule. Green
MYCIN

states his rules in the predicate calculus. For example:

[1] (FA (X) (IF (IN X FIREMEN) (OWNS X RED-SUSPENDERS)))
[2] (FA (X) (IF (IN X FIRECHIEF) (IN X FIREMEN)))

are universally quantified expressions of the following rules:

[1] IF: X IS A FIREMAN
    THEN: X OWNS RED SUSPENDERS

[2] IF: X IS A FIRECHIEF
    THEN: X IS A FIREMAN

Green's program uses such "rules" to answer questions regarding system knowledge. The questions themselves may be stated as rules:

[3] Question: (FA (X) (IF
    (IN X FIRECHIEF) (OWNS X RED-SUSPENDERS)))

that is,

[3] Is the following rule valid?
    IF: X IS A FIRECHIEF
    THEN: X OWNS RED SUSPENDERS

QA3 uses [1] and [2], plus the "resolution principle" for theorem proving [Robinson, 1965], to show that [3] is a valid rule and thereby to answer the question affirmatively. Resolution is mentioned again during the discussion of control structures in § 3.3.1.

The second example of a rule-based system is the program designed by Colby et al. for modeling human belief structures [Colby, 1969]. They acquired statements of belief from a human subject and coded them as either facts or rules of inference. Facts had associated numerical weights representing their degree of credibility to the human subject, but the rules reflected simple implication without any weighting of the strength of the relationship. For example:

(F 80 SELF NOTLIKE (CHILD1 HAS AGGRESSIVENESS))

is their system's representation for the fact (F) that the subject (SELF) found it strongly credible (80) that she did not like the
aggressiveness of one of her children (CHILD1). A sample rule from their data base is:

\[(R \text{ THEPARENT SLAP HISCHILD IMPLIES THEPARENT DISTRESS HISCHILD})\]

“Implies” in their rules does not necessarily correspond to logical implication. Instead it may represent relationships that are logical, causal, temporal, or conceptual. Furthermore, the rules are similar to those of MYCIN in that they represent judgments of a human subject (cf. expert) rather than natural laws.

The main task for Colby and his coworkers involved estimating the credibility of a given proposition describing some actual or hypothetical situation. They tested their model by writing a program that used the belief structures obtained from their human subject in order to assess the credibility of a new hypothesis not already in the data base. They then compared the judgment of the program with the credibility estimate of the subject herself. System rules and facts were linked in a graph structure that was searched by a variety of algorithms in an attempt to assess the credibility of a new proposition. Unfortunately, the human subject left the study before a formal evaluation of the program’s credibility estimates could be undertaken.

In the late 1960’s, Waterman developed a rule-based system for playing poker [Waterman, 1970]. He selected this game because, unlike chess or other games commonly modeled by computer programs, poker is characterized by imperfect knowledge regarding the opponent’s position. Close attention was paid to the optimal representation of heuristics needed by a poker playing machine. He decided that a good representation should:

1. permit separation of the heuristics from the main body of the program;
2. provide identification of individual heuristics and an indication of how they are interrelated;
3. be compatible with generalization schemes.

Clearly these desiderata correspond closely to the criterion of knowledge modularity I discussed in Chapter 2. Waterman’s concern with these factors stemmed from his desire to create a program that would
not only play poker but also learn new heuristics that could be incorporated in a straightforward fashion and would permit improvement of the system’s game over time.

Waterman pictured poker as a succession of states, with each play causing a transition from one state to another. The situation at any given time could therefore be characterized by a state vector, and game heuristics would involve decisions based upon the current status of the state vector. Thus heuristics could be represented as production rules or so-called situation-action (SA) rules, i.e., if S is true, then take action A. I shall not present Waterman’s formal representation here since that would necessitate a description of his rather complex state vector, but the following excerpt from his paper [Waterman, 1970] should give an adequate description of the kind of heuristic rules that he was able to code:

If your hand is excellent then bet low if the opponent tends to be a conservative player and has just bet low. Bet high if the opponent is not conservative, is not easily bluffed, and has just made a sizable bet. Call if the pot is extremely large, and the opponent has just made a sizable bet.

The program could be taught such heuristics explicitly and was also able to generalize new rules from its experience when playing the game. The result was a system that eventually played an admirable game of poker.

The last rule-based system for discussion in this section is one of the foremost examples of AI techniques effectively applied to a real-world problem domain. HEURISTIC DENDRAL is a large set of programs designed to aid in the identification of chemical structures from mass spectral data [Feigenbaum, 1968; Buchanan, 1969]. The input to the system is the data derived for an unknown organic molecule that has been subjected to mass spectral analysis. HEURISTIC DENDRAL uses this input, plus a complex theory of mass spectroscopy embodied in SA rules, to predict spectra for each of the structural hypotheses. The program has a heuristic hypothesis generator that first compiles a set of all reasonable structures on the basis of primary spectral observations. It then uses SA rules acquired from experts in mass spectroscopy to predict spectra for each of the structural hypotheses. A final evaluation stage selects the one or more hypotheses for which the predicted
Consultation System

mass spectrum most closely resembles the spectrum that was empirically observed.

Acquiring the mass spectral rules from experts in organic chemistry, who may have limited knowledge of computers or of the DENDRAL program, has proven to be a task of considerable difficulty [Buchanan, 1970]. One is immediately reminded of the challenge in getting well-formed decision rules for MYCIN by discussing patients with infectious disease experts. An example of one of DENDRAL’s SA rules is the following:

\[
\begin{align*}
1 & 2 & 3 & 4 & 5 & 6 & 7 \\
\text{SA Rule: } & (C,H)-\text{CH2-NH-CH2-CH2-CH2-(C,H)} & \rightarrow & \text{Breakbond (4 5)}
\end{align*}
\]

This rule states that a seven membered chain with the characteristics shown in the Situation part of the rule is apt to undergo a bond break between atoms 4 and 5 when subjected to mass spectral bombardment. It is therefore useful in predicting the spectrum of a molecule that satisfies the situation part of the rule (since peaks in a mass spectrum correspond to molecular fragments of a specific identifiable mass).

The SA rules used by HEURISTIC DENDRAL have many similarities to those used in Waterman’s program [Waterman, 1970]. Just as Waterman chose a production rule system in part so that new heuristics could be learned and integrated with ease, DENDRAL has broadened its scope to consider mechanisms for inferring new SA rules. This adjunct to HEURISTIC DENDRAL is known as META-DENDRAL [Buchanan, 1971, 1972]. The idea is to analyze the spectra of known molecules in an effort to infer the theoretical basis for the data that are observed. Because system knowledge is maintained in modular SA rules and is not embedded within the programs themselves, this kind of system enhancement is greatly facilitated. The result is a program that often performs at the level of a post-doctoral chemist and is able to analyze and draw inferences on such complex cyclic structures as estrogenic steroids [Buchanan, 1973].

The decision criteria stored in MYCIN’s rules are in many ways similar to the “rules” or “theorems” that form the knowledge base of the programs I have discussed. All the systems keep their rules separate from their programs so that the functions are domain independent and attempts at generalization are facilitated. As dis-
3.3.1, the rules are actually used in a variety of fashions. Regardless of control structures, however, the advantages of identifiable packets of knowledge should now be clear. A final point to note is that, unlike the rules in the other systems described, MYCIN's decision criteria contain explicit weighting factors that reflect the strength of the indicated inference.

### 3.2.1-2 Representation of Rules

The 200 rules currently in the MYCIN system consist of a PREMISE, an ACTION, and sometimes an ELSE clause. Every rule has a name of the form "RULE###" where "###" is a three digit number. When discussing rules in their most general form, it will often be useful to adopt a shortened form of notation. I shall use upper-case letters for conditions and conclusions, inserting a right arrow to indicate implication. Thus

\[ A \land B \rightarrow C \]

signifies the rule for which the PREMISE is the conjunction of conditions A and B and the ACTION is C.

The details of rules and how they are used are discussed throughout the remainder of this chapter. I therefore offer a formal definition of rules that will serve in part as a guide for what is to follow. The rules are stored as LISP data structures in accordance with the following Backus Normal Form (BNF) description:

\[
\begin{align*}
<\text{rule}> & : = <\text{premise}> <\text{action}> | <\text{premise}> <\text{action}> <\text{else}> \\
<\text{premise}> & : = ($\text{AND} <\text{condition}> \ldots <\text{condition}> ) \\
<\text{condition}> & : = ( <\text{func1}> <\text{context}> <\text{parameter}> ) | \\
& \quad ( <\text{func2}> <\text{context}> <\text{parameter}> <\text{value}> ) | \\
& \quad ( <\text{special-func} > <\text{arguments} > ) | \\
& \quad ($\text{OR} <\text{condition}> \ldots <\text{condition}> ) \\
<\text{action}> & : = <\text{concpart}> \\
<\text{else}> & : = <\text{concpart}> \\
<\text{concpart}> & : = <\text{conclusion}> | <\text{actfunc}> | \\
& \quad (\text{DO-ALL} <\text{conclusion}> \ldots <\text{conclusion}> ) | \\
& \quad (\text{DO-ALL} <\text{actfunc}> \ldots <\text{actfunc} > ) \\
<\text{context}> & : = \text{See } \S 3.2.2 \\
<\text{parameter}> & : = \text{See } \S 3.2.3 \\
<\text{value}> & : = \text{See } \S 3.2.3
\end{align*}
\]
Thus the PREMISE of a rule consists of a conjunction of conditions, each of which must hold for the indicated ACTION to be taken. Negations of conditions are handled by the individual predicates ($<\text{func1}>$ and $<\text{func2}>$) and therefore do not require a $\text{NOT}$ function to complement the Boolean functions $\text{AND}$ and $\text{OR}$. If the PREMISE of a rule is known to be false, the conclusion or action indicated by the ELSE clause is taken. If the truth of the PREMISE cannot be ascertained, or the PREMISE is false but no ELSE condition exists, the rule is simply ignored.

The PREMISE of a rule is always a conjunction of one or more conditions. Disjunctions of conditions may be represented as multiple rules with identical ACTION clauses. A condition, however, may itself be a disjunction of conditions. These conventions are somewhat arbitrary but do provide sufficient flexibility so that any Boolean expression may be represented by one or more rules. As is discussed in § 3.3, multiple rules are effectively $\text{OR}$'ed together by MYCIN's control structure.

For example, 2-leveled Boolean nestings of conditions are acceptable as follows:

Legal:

1. $A \& B \& C \rightarrow D$
2. $A \& (B \text{ OR } C) \rightarrow D$
3. $(A \text{ or } B \text{ or } C) \& (D \text{ or } E) \rightarrow F$

Illegal:

4. $A \text{ or } B \text{ or } C \rightarrow D$
5. $A \& (B \text{ or } (C \& D)) \rightarrow E$

Rule [4] is correctly represented by the following three rules:

6. $A \rightarrow D$
7. $B \rightarrow D$
8. $C \rightarrow D$
whereas [5] must be written as:

[9] \( A \& C \& D \rightarrow E \)
[10] \( A \& B \rightarrow E \)

Unlike rules that involve strict implication, the strength of an inference in MYCIN’s rules may be modified by a certainty factor (CF). A CF is a number from -1 to +1, the nature of which is described in § 3.2.4 and in Chapter 4. The notation for indicating the strength of an implication will be as follows:

\( A \& B \& C \sim a \rightarrow D \)

Here the rule states that the conjunction of conditions A, B, and C implies D with certainty factor a.

The following three examples are rules from MYCIN that have been translated into English from their internal LISP representation (§ 3.2.7). They represent the range of rule types available to the system. The details of their internal representation will be explained as I proceed.

RULE037

IF: 1) THE IDENTITY OF THE ORGANISM IS NOT KNOWN WITH CERTAINTY, AND
2) THE STAIN OF THE ORGANISM IS GRAMNEG, AND
3) THE MORPHOLOGY OF THE ORGANISM IS ROD, AND
4) THE AEROBICITY OF THE ORGANISM IS AEROBIC
THEN: THERE IS STRONGLY SUGGESTIVE EVIDENCE (.8) THAT THE CLASS OF THE ORGANISM IS ENTEROBACTERIACEAE

RULE145

IF: 1) THE THERAPY UNDER CONSIDERATION IS ONE OF: CEPHALOTHIN CLINDAMYCIN ERYTHROMYCIN LINCOMYCIN VANCOMYCIN, AND
2) MENINGITIS IS AN INFECTIOUS DISEASE DIAGNOSIS FOR THE PATIENT
THEN: IT IS DEFINITE (1) THE THERAPY UNDER CONSIDERATION IS NOT A POTENTIAL THERAPY FOR USE AGAINST THE ORGANISM
RULE060

IF: THE IDENTITY OF THE ORGANISM IS BACTEROIDES
THEN: I RECOMMEND THERAPY CHOSEN FROM AMONG THE FOLLOWING DRUGS:

1 - CLINDAMYCIN (.99)
2 - CHLORAMPHENICOL (.99)
3 - ERYTHROMYCIN (.57)
4 - TETRACYCLINE (.28)
5 - CARBENICILLIN (.27)

Before I can explain how rules such as these are invoked and evaluated, it is necessary further to describe MYCIN’s internal organization. I shall therefore temporarily digress in order to lay some groundwork for the description of the evaluation functions in § 3.2.5.

3.2.2 CATEGORIZATION OF RULES BY CONTEXT

3.2.2-1 Context Tree

Although it is common to describe diagnosis as inference based on attributes of the patient, MYCIN’s decisions must necessarily involve not only the patient but also the cultures that have been grown, organisms isolated, and drugs that have been administered. Each of these is termed a “context” of the program’s reasoning (see <context> in the BNF description of rules, § 3.2.1-2). (This use of the word “context” should not be confused with its meaning in high level languages that permit temporary saving of all information regarding a program’s current status—a common mechanism for backtracking and parallel processing implementations). MYCIN currently knows about ten different context-types:

CURCULS - a current culture from which organisms were isolated
CURDRUGS - an antimicrobial agent currently being administered to a patient
CURORGS - an organism isolated from a current culture
OPDRGS - an antimicrobial agent administered to the patient during a recent operative procedure
OPERS - an operative procedure which the patient has undergone
PERSON - the patient himself
POSSTHER - a therapy being considered for recommendation
PRIORCULS - a culture obtained in the past
PRIORDRGS - an antimicrobial agent administered to the patient previously
PRIORORGS - an organism isolated from a prior culture

Except for PERSON, each of these context-types may be instantiated more than once during any given run of the consultation program. Some may not be created at all if they do not apply to the given patient. However, each time a context-type is instantiated it is given a unique name. For example, CULTURE-1 is the first CURCUL and ORGANISM-1 is the first CURORG. Subsequent CURCULS or PRIORCULS are called CULTURE-2, CULTURE-3, etc.

The context-types instantiated during a run of the consultation program are arranged hierarchically in a data structure termed the “context tree.” One such tree is shown in Figure 3-1. The context-type for each instantiated context is shown in parentheses beside its name. Thus, to clarify terminology, we note that a node in the context tree is called a context and is created as an instantiation of a

---

**SAMPLE CONTEXT TREE**

![Context Tree Diagram]

Figure 3-1: Context tree for a sample patient with two recent positive cultures, an older one, and a recent significant operative procedure. Nodes in the tree are termed “contexts.”
Consultation System

category-type. This sample context tree corresponds to a patient from whom two current cultures and one prior culture were obtained. One organism was isolated from each of the current cultures, but the patient is being treated (with two drugs) for only one of the current organisms. Furthermore, two organisms were grown from the prior culture but therapy was instituted to combat only one of these. Finally, the patient has had a recent operative procedure during which he was treated with an antimicrobial agent.

The context tree is useful not only because it gives structure to the clinical problem (Figure 3-1 already tells us a good deal about PATIENT-1), but also because we often need to be able to relate one context to another. For example, in considering the significance of ORGANISM-2, MYCIN may well want to be able to reference the site of the culture from which ORGANISM-2 was obtained. Since the patient has had three different cultures, we need an explicit mechanism for recognizing that ORGANISM-2 came from CULTURE-2, not CULTURE-1 or CULTURE-3. The technique for dynamic propagation (i.e., growth) of the context tree during a consultation is described in § 3.4).

3.2.2-2 Interrelationship of Rules and Context Tree

The 200 rules currently used by MYCIN are not explicitly linked in a decision tree or reasoning network. This feature is in keeping with our desire to keep the system knowledge modular and manipulable. However, rules are subject to categorization in accordance with the context-types for which they are most appropriately invoked. For example, some rules deal with organisms, some with cultures, and still others deal solely with the patient himself. MYCIN’s current rule categories are as follows (context-types to which they may be applied are enclosed in parentheses):

- CULRULES - rules that may be applied to any culture (CURCULS or PRIORCULS)
- CURCULRULES - rules that may be applied only to current cultures (CURCULS)
- CURORGRULES - rules that may be applied only to current organisms (CURORGs)
MYCIN

DRGRULES - rules that may be applied to any antimicrobial agent that has been administered to combat a specific organism (CURDRUGS PRIORDRGS)

OPRULES - rules that may be applied to operative procedures (OPERS)

ORDERRULES - rules that are used to order the list of possible therapeutic recommendations (POSSOTHER)

ORGRULES - rules that may be applied to any organism (CURORGS or PRIORORGS)

PATRULES - rules that may be applied to the patient (PERSON)

PDRGRULES - rules that may be applied only to drugs given to combat prior organisms (PRIORDRUGS)

PRCULRULES - rules that may be applied only to prior cultures (PRIORCULS)

PRORGRULES - rules that may be applied only to organism isolated from prior cultures (PRIORORGS)

THERULES - rules that store information regarding drugs of choice (§ 3.5).

Every rule in the MYCIN system belongs to one, and only one, of these categories. Furthermore, selecting the proper category for a newly acquired rule does not present a problem. In fact, as is discussed in § 6.3, category selection can be automated to a large extent.

Consider now a rule such as:

RULE124

IF: 1) THE SITE OF THE CULTURE IS THROAT, AND
2) THE IDENTITY OF THE ORGANISM IS STREPTOCOCCUS
THEN: THERE IS STRONGLY SUGGESTIVE EVIDENCE (.8) THAT THE SUBTYPE OF THE ORGANISM IS NOT GROUP-D

This is one of MYCIN's ORGRULES and may thus be applied either to a CURORGS context or a PRIORORGS context. Referring back to Figure 3-1, suppose RULE124 above were applied to ORGANISM-2. The first condition in the PREMISE refers to the site of the culture from which ORGANISM-2 was isolated (i.e., CULTURE-2) and not to the organism itself (i.e., organisms do not have SITES, but cultures do). The context tree is therefore impor-
Consultation System

tant, as I mentioned above, for determining the proper context when a rule refers to an attribute of a node in the tree other than the context to which the rule is being explicitly applied. Note that this means that a single rule may refer to nodes at several levels in the context tree. The rule is categorized simply on the basis of the lowest context-type (in the tree) that it may reference. Thus RULE124 is an ORGRULE rather than a CULRULE.

3.2.3 CLINICAL PARAMETERS

This subsection describes the data types indicated by <parameter> and <value> in the BNF description of rules (§ 3.2.1-2). Although I have previously asserted that all MYCIN’s knowledge is stored in its corpus of rules, the clinical parameters and their associated properties comprise an important class of second level knowledge. I shall first explain the kind of parameters used by the system, and will then describe their representation.

A clinical parameter is a characteristic of one of the contexts in the context tree, i.e., the name of the patient, the site of a culture, the morphology of an organism, the dose of a drug, etc. All such attributes will be termed “clinical parameters.” A patient’s status would be completely specified by a context tree in which values were known for all the clinical parameters characterizing each node in the tree (assuming the parameters known to MYCIN encompass all those that are clinically relevant—a dubious assumption at present). In general this is more information than is needed, however, so one of MYCIN’s tasks is to identify those clinical parameters that need to be considered for the patient about whom advice is being sought. This task is similar to the problem of sequential test selection that was relevant to many of the programs discussed in § 1.3.

The concept of an attribute-object-value triple is common to much of the AI field. This associative relationship is a basic data type for the SAIL language [Feldman, 1972] and is the foundation for the property-list formalism in LISP [McCarthy, 1962]. Relational predicates in the predicate calculus also represent associative triples. The point is that many facts may be expressed as triples that state that some object has an attribute with some specified value. Stated in the order <attribute object value>, examples include:
Note that the last two examples are different from the others since they represent a rather different kind of relationship. In fact, several authors would classify the first six as “relations” and the last two as “predicates,” using the simpler notation:

\[ \text{MAN(BOB)} \]
\[ \text{–WOMAN(BOB)} \]

Regardless of whether it is written as \text{MAN(BOB)} or \text{(MAN BOB TRUE)}, this binary predicate statement has rather different characteristics from the relations that form natural triples. This distinction will become more clear later (see “yes–no” parameters below).

MYCIN stores inferences and data using the attribute-object-value concept I have just described. The object is always some context in the context tree, and the attribute is a clinical parameter appropriate for that context. Information stored using this mechanism may be retrieved and updated in accordance with a variety of conventions described throughout this chapter.

### 3.2.3-1 Three Kinds of Clinical Parameters

There are three fundamentally different kinds of clinical parameters. The simplest variety are the ones we call “single-valued” parameters. These are attributes such as the name of the patient or the identity of the organism. In general, they have a large number of possible values that are mutually exclusive. As a result, only one can be the true value, although several may seem likely at any point during the consultation.

“Multi-valued” parameters also generally have a large number of possible values. The difference is that the possible values need not be
Consultation System

mutually exclusive. Thus, such attributes as a patient’s drug allergies or a locus of infection may have multiple values, each of which is known to be correct.

The third kind of clinical parameter corresponds to the binary predicate discussed above. These are attributes that are either true or false for the given context. For example, the significance of an organism is either true or false (yes or no), as is the parameter indicating whether the dose of a drug is adequate. Attributes of this variety are called “yes–no” parameters. They are, in effect, a special kind of “single-valued” parameter for which there are only two possible values.

3.2.3-2 Classification and Representation of Parameters

The clinical parameters known to MYCIN are categorized in accordance with the context to which they apply. These categories include:

- **PROP-CUL**: those clinical parameters that are attributes of cultures (e.g., site of the culture, method of collection)
- **PROP-DRG**: those clinical parameters that are attributes of administered drugs (e.g., name of the drug, duration of administration)
- **PROP-OP**: those clinical parameters that are attributes of operative procedures (e.g., the cavity, if any, opened during the procedure)
- **PROP-ORG**: those clinical parameters that are attributes of organisms (e.g., identity, gram stain, morphology)
- **PROP-PT**: those clinical parameters that are attributes of the patient (e.g., name, sex, age, allergies, diagnoses)
- **PROP-THER**: those clinical parameters that are attributes of therapies being considered for recommendation (e.g., recommended dosage, prescribing name)

These categories encompass all clinical parameters used by the system. Note that any of the nodes (contexts) in the context tree for the patient may be fully characterized by the values of the set of clinical parameters in one of these categories.

Each of the 65 clinical parameters currently known to MYCIN has an associated set of properties that is used during consideration of
Figure 3-2: Examples of the three types of clinical parameters. As shown, each clinical parameter is characterized by a set of “properties” described in the text.

the parameter for a given context. Figure 3-2 presents three clinical parameters that together demonstrate several of these properties:

**EXPECT**
- this property indicates the range of expected values that the parameter may have.
  - if = (YN) then the parameter is a “yes–no” parameter
  - if = (NUMB) then the expected value of the parameter is a number
if = \{\text{\textsc{oneof} <list>\}}\) then the value of the parameter must be a member of <list>.

if = (\text{\textsc{any}}) then there is no restriction on the range of values that the parameter may have.

**PROMPT**
- this property is a sentence used by MYCIN when it requests the value of the clinical parameter from the user; if there is an asterisk in the phrase (see Figure 3-2), it is replaced by the name of the context about which the question is being asked; this property is used only for “yes-no” or “single-valued” parameters.

**PROMPT1**
- this property is similar to PROMPT except it is used if the clinical parameter is a “multi-valued” parameter; in these cases MYCIN only asks the question about a single one of the possible parameter values; the value of interest is substituted for (VALUE) in the question.

**LABDATA**
- this property is a flag that is either T or NIL; if T it indicates that the clinical parameter is a piece of primitive data, the value of which may be known with certainty to the user (see § 3.3.2-1).

**LOOKAHEAD**
- this property is a list of all rules in the system that reference the clinical parameter in their PREMISE.

**UPDATED-BY**
- this property is a list of all rules in the system in which the ACTION or ELSE clause permits a conclusion to be made regarding the value of the clinical parameter.

**CONTAINED-IN**
- this property is a list of all rules in the system in which the ACTION or ELSE clause references the clinical parameter but does not cause its value to be updated.

**TRANS**
- this property is used for translating the clinical parameter into its English representation (see § 3.2.7); the context of the parameter is substituted for the asterisk during translation.

**DEFAULT**
- this property is used only with clinical parameters for which \text{\textsc{expect}} = (\text{\textsc{numb}}); it gives the expected units for numerical answers (e.g., days, years, grams, etc.).

**CONDITION**
- this property, when utilized, is an executable LISP expression that is evaluated before MYCIN requests the value of the parameter; if the CONDITION is true, the question is not asked (e.g., “Don’t ask for an organism’s subtype if its genus is not known by the user”).

The uses of these properties will be discussed throughout the remainder of this chapter and in Chapter 5. However, a few additional points are relevant here. First, it should be noted that the order of
rules on the properties LOOKAHEAD, UPDATED-BY, and CONTAINED-IN is arbitrary and does not affect the program’s advice. Second, TRANS is the only property that must exist for every clinical parameter. Thus, for example, if there is no PROMPT or PROMPT1 stored for a parameter, the system assumes that it simply cannot ask the user for the value of the parameter. Finally, note in Figure 3-2 the difference in the TRANS property for “yes–no” and non-“yes–no” parameters. In general, a parameter and its value may be translated as:

THE <attribute> OF <object> IS <value>

However, for a “yes–no” parameter such a FEBRILE, it is clearly necessary to translate the parameter in a fashion other than:

THE FEBRILE OF PATIENT-1 IS YES

Our solution has been to suppress the YES altogether and simply to say:

PATIENT-1 IS FEBRILE

3.2.4 CERTAINTY FACTORS

Chapter 4 presents a detailed description of certainty factors and their theoretical foundation. This section therefore provides only a brief overview of the subject. A familiarity with the characteristics of certainty factors (CF’s) is necessary, however, for the discussion of MYCIN during the remainder of this chapter.

The value of every clinical parameter is stored by MYCIN along with an associated certainty factor that reflects the system’s “belief” that the value is correct. This formalism is necessary because, unlike domains in which objects either have or do not have some attribute, in medical diagnosis and treatment there is often uncertainty regarding attributes such as the significance of the disease, the efficacy of a treatment, or the diagnosis itself. As discussed in § 1.3, most medical decision making programs use probability to reflect the uncertainties. CF’s are an alternative to conditional probability that offer several advantages in MYCIN’s domain (as described in Chapter 4).

A certainty factor is a number between −1 and +1 that reflects the
degree of belief in a hypothesis. Positive CF's indicate there is evidence that the hypothesis is valid. The larger the CF, the greater the belief in the hypothesis. When CF=1, the hypothesis is known to be correct. On the other hand, negative CF's indicate that the weight of evidence suggests that the hypothesis is false. The smaller the CF, the greater the belief that the hypothesis is invalid. CF=-1 means that the hypothesis has been effectively disproven. When CF=0, there is either no evidence regarding the hypothesis, or the supporting evidence is equally balanced by evidence suggesting that the hypothesis is not true.

MYCIN's hypotheses are statements regarding values of clinical parameters for the various nodes in the context tree. For example, sample hypotheses are:

\[ h_1 = \text{The identity of ORGANISM-1 is streptococcus} \]
\[ h_2 = \text{PATIENT-1 is febrile} \]
\[ h_3 = \text{The name of PATIENT-1 is John Jones} \]

We use the notation CF\[h,E\] = X to represent the certainty factor for the hypothesis h based upon evidence E. Thus if CF\[h_1,E\] = .8, CF\[h_2,E\] = -.3, and CF\[h_3,E\] = +1, the three sample hypotheses above may be qualified as follows:

\[ \text{CF\[h_1,E\] = .8} \quad : \quad \text{There is strongly suggestive evidence (.8) that the identity of ORGANISM-1 is streptococcus} \]
\[ \text{CF\[h_2,E\] = -.3} \quad : \quad \text{There is weakly suggestive evidence (.3) that PATIENT-1 is not febrile} \]
\[ \text{CF\[h_3,E\] = +1} \quad : \quad \text{It is definite (1) that the name of PATIENT-1 is John Jones} \]

Certainty factors are used in two ways. First, as noted, the value of every clinical parameter is stored with its associated certainty factor. In this case the evidence E stands for all information currently available to MYCIN. Thus, if the program needs the identity of ORGANISM-1, it may look in its dynamic data base and find:

\[ \text{IDENT of ORGANISM-1} = ((\text{STREPTOCOCCUS .8})) \]

The second use of CF's is in the statement of decision rules themselves. In this case the evidence E corresponds to the conditions in the PREMISE of the rule. Thus
MYCIN

A & B & C → D

is a representation of the statement CF[D, (A&B&C)] = x. For example, consider the following rule:

IF: 1) THE STAIN OF THE ORGANISM IS GRAMPOS, AND
    2) THE MORPHOLOGY OF THE ORGANISM IS COCCUS, AND
    3) THE GROWTH CONFORMATION OF THE ORGANISM IS CHAINS
THEN: THERE IS SUGGESTIVE EVIDENCE (.7) THAT THE IDENTITY OF THE ORGANISM IS STREPTOCOCCUS

This rule may also be represented as CF[h1,e] = .7 where h1 is the hypothesis that the organism (context of the rule) is a streptococcus and e is the evidence that it is a gram positive coccus growing in chains.

Since diagnosis is, in effect, the problem of selecting a disease from a list of competing hypotheses, it should be clear that MYCIN may simultaneously be considering several hypotheses regarding the value of a clinical parameter. These hypotheses are stored together, along with their CF's, for each node in the context tree. We use the notation Val[C,P] to signify the set of all hypotheses regarding the value of the clinical parameter P for the context C. Thus if MYCIN has reason to believe that ORGANISM-1 may be either a streptococcus or staphylococcus, although pneumococcus has been ruled out, its dynamic data base might well show:

Val[ORGANISM-1,IDENT] = ((STREPTOCOCCUS .6)
                           (STAPHYLOCOCCUS .4)
                           (DIPLOCOCUS-PNEUMONIAE −1))

Note that Chapter 5 shows that the sum of the CF's for supported hypotheses regarding a “single-valued” parameter (i.e., those parameters for which the hypotheses are mutually exclusive) should not exceed 1. “Multi-valued” parameters, on the other hand, may have several hypotheses that are all known to be true. For example:

Val[PATIENT-1,ALLERGY] = ((PENICILLIN 1) (AMPICILLIN 1)
                           (CARBENICILLIN 1) (METHICILLIN 1))

As soon as a hypothesis regarding a “single-valued” parameter is
Consultation System

proved to be true, all competing hypotheses are effectively dis­proved:

\[ \text{Val[ORGANISM-1,IDENT] = ((STREPTOCOCCUS 1) (STAPHYLOCOCCUS-1) (DIPLOCOCCUS-PNEUMONIAE -1))} \]

In Chapter 4 it is demonstrated that \( \text{CF[h,E]} = -\text{CF[not.h,E]} \). This observation has important implications for the way MYCIN handles the binary-valued attributes we call “yes–no” parameters. Since “yes” is “not.no,” it is not necessary to consider “yes” and “no” as competing hypotheses for the value of a “yes–no” parameter (as we do for “single-valued” parameters). Instead we can always express “no” as “yes” with a reversal in the sign of the CF. This means that \( \text{Val[C,P]} \) is always equal to the single value “yes,” along with its associated CF, when P is a “yes–no” parameter.

In § 3.3.3-2, I discuss MYCIN’s mechanism for adding to the list of hypotheses in \( \text{Val[C,P]} \) as new rules are invoked and executed. The following points should be emphasized here, however:

1. the strength of the conclusion associated with the execution of a rule reflects not only the CF assigned to the rule, but also the program’s degree of belief regarding the validity of the PREMISE;
2. the support of several rules favoring a single hypothesis may be assimilated incrementally on the list \( \text{Val[C,P]} \) by using special combining functions described in § 4.6.

3.2.5 FUNCTIONS FOR EVALUATION OF PREMISE CONDITIONS

This section describes the evaluation of the individual conditions (see <condition>, § 3.2.1-2) in the PREMISE of rules. Conditions in general evaluate to “true” or “false” (T or NIL). Thus, they may at first glance be considered simple predicates on the values of clinical parameters. However, since there may be several competing hypotheses on the list \( \text{Val[C,P]} \), each associated with its own degree of belief as reflected by the CF, conditional statements regarding the value of parameters can be quite complex. All predicates are implemented as LISP functions. The functions that undertake the required analysis are of three varieties, specified by the designations <func1>,

101
MYCIN

<func2>, and <special-func> in the BNF rule description (§ 3.2.2-1). This section explains the <func1> and <func2> predicates. The <special-func> category is deferred until § 3.2.6-2, however, so that I may first introduce our specialized knowledge structures (§ 3.2.6-1).

There are four predicates in the category <func1>. These functions do not form conditionals on specific values of a clinical parameter, but are concerned with the more general status of knowledge regarding the attributes in question. For example, KNOWN[ORGANISM-1, IDENT] is an invocation of the <func1> predicate KNOWN; it would return true if the identity of ORGANISM-1 were known, regardless of the value of the clinical parameter IDENT. KNOWN and the other <func1> predicates may be formally defined as follows:

**Predicates of the class <func1>:**

Let V=Val[C,P] be the set of all hypotheses regarding the value of the clinical parameter P for the context C

Let Mv=Max[V] be the most strongly supported hypothesis in V (i.e., the hypothesis with the largest CF)

Let CFmv=CF[Mv,E] be the certainty factor of Mv given E, where E is the total available evidence

Then, if P is either a “single-valued” or “multi-valued” parameter, the four predicates (functions) may be specified as follows:

<table>
<thead>
<tr>
<th>FUNCTION</th>
<th>IF:</th>
<th>THEN:</th>
<th>ELSE:</th>
</tr>
</thead>
<tbody>
<tr>
<td>KNOWN[C,P]</td>
<td>CFmv&gt;.2</td>
<td>T</td>
<td>NIL</td>
</tr>
<tr>
<td>NOTKNOWN[C,P]</td>
<td>CFmv&lt;.2</td>
<td>T</td>
<td>NIL</td>
</tr>
<tr>
<td>DEFINITE[C,P]</td>
<td>CFmv=1</td>
<td>T</td>
<td>NIL</td>
</tr>
<tr>
<td>NOTDEFINITE[C,P]</td>
<td>CFmv&lt;1</td>
<td>T</td>
<td>NIL</td>
</tr>
</tbody>
</table>

In words, these definitions reflect MYCIN’s convention that the value of a parameter is KNOWN if the CF of the most highly supported hypothesis exceeds .2. The .2 threshold was selected empirically. The implication is that a positive CF less than .2 reflects so little evidence supporting the hypothesis that there is virtually no reasonable hypothesis currently known. The interrelationships among these functions are diagrammed on a CF number line in...
Figure 3-3: Diagram indicating the range of CF values over which the \(<\text{func1}>\) predicates hold true when applied to multivalued or single-valued (i.e., non-"yes-no") clinical parameters. Vertical lines and zeros distinguish closed and nonclosed certainty factor ranges, respectively.

Figure 3-3. Regions specified are the range of values for \(\text{CF}_{\text{mv}}\) over which the function returns T.

As was pointed out in the previous section, however, "yes-no" parameters are special cases because we know \(\text{CF}[\text{YES,E}] = -\text{CF}[\text{NO,E}]\). Since the values of "yes-no" parameters are always stored in terms of YES, MYCIN must recognize that a YES with \(\text{CF}=-.9\) is equivalent to a NO with \(\text{CF}=.9\). The definitions of our four \(<\text{func1}>\) predicates above do not reflect this distinction. Therefore, when \(P\) is a "yes-no" parameter, the four functions are specified as follows:

\[
\begin{array}{llll}
\text{FUNCTION:} & \text{IF:} & \text{THEN:} & \text{ELSE:} \\
\text{KNOWN}[C,P] & |\text{CF}_{\text{mv}}> .2 & T & \text{NIL} \\
\text{NOT KNOWN}[C,P] & |\text{CF}_{\text{mv}}\leq .2 & T & \text{NIL} \\
\text{DEFINITE}[C,P] & |\text{CF}_{\text{mv}}=1 & T & \text{NIL} \\
\text{NOT DEFINITE}[C,P] & |\text{CF}_{\text{mv}}<1 & T & \text{NIL} \\
\end{array}
\]

Figure 3-4 shows the relationship among these functions for "yes-no" parameters.

There are nine predicates in the category \(<\text{func2}>\). Unlike the \(<\text{func1}>\) predicates, these functions control conditional statements regarding specific values of the clinical parameter in question. For example, \(\text{SAME}[\text{ORGANISM-1}, \text{IDENT}, \text{E.COLI}]\) is an invocation
Figure 3-4: Diagram indicating the range of CF values over which the <func1> predicates hold true when applied to “yes-no” clinical parameters.

of the <func2> predicate SAME; it would return true if the identity of ORGANISM-1 were known to be E.coli. SAME and the other <func2> predicates may be formally defined as follows:

Predicates of the class <func2>:

Let \( V = \text{Val}[C,P] \) be the set of all hypotheses regarding the value of the clinical parameter \( P \) for the context \( C \).

Let \( I = \text{Intersection}[V,LST] \) be the set of all hypotheses in \( V \) which also occur in the set \( LST \); \( LST \) contains the possible values of \( P \) for comparison by the predicate-function; it usually contains only a single element; if no element in \( LST \) is also in \( V \), \( I \) is simply the empty set.

Let \( Mi = \text{Max}[I] \) be the most strongly confirmed hypothesis in \( I \); thus \( Mi \) is NIL if \( I \) is the empty set;

Let \( CFmi = CF[Mi,E] \) be the certainty factor of \( Mi \) given \( E \), where \( CFmi = 0 \) if \( Mi \) is NIL

Then the <func2> predicates are defined as follows:

<table>
<thead>
<tr>
<th>FUNCTION:</th>
<th>IF:</th>
<th>THEN:</th>
<th>ELSE:</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAME[C,P,LST]</td>
<td>CFmi&gt;.2</td>
<td>CFmi</td>
<td>NIL</td>
</tr>
<tr>
<td>THOUGHTNOT[C,P,LST]</td>
<td>CFmi&lt;-0.2</td>
<td>-CFmi</td>
<td>NIL</td>
</tr>
<tr>
<td>NOTSAME[C,P,LST]</td>
<td>CFmi=.2</td>
<td>T</td>
<td>NIL</td>
</tr>
<tr>
<td>MIGHTBE[C,P,LST]</td>
<td>CFmi&gt;.2</td>
<td>T</td>
<td>NIL</td>
</tr>
<tr>
<td>VNOTKNOWN[C,P,LST]</td>
<td></td>
<td>T</td>
<td>NIL</td>
</tr>
<tr>
<td>DEFIS[C,P,LST]</td>
<td>CFmi=+1</td>
<td>T</td>
<td>NIL</td>
</tr>
</tbody>
</table>
The names of the functions have been selected to reflect their semantics. Figure 3-5 shows a graphic representation of each function and also explicitly states the interrelationships among them.

Note that SAME and THOUGHTNOT are different from all the other functions that I have discussed in that they return a number (CF) rather than T if the defining condition holds. This feature permits MYCIN to record the degree to which PREMISE conditions are satisfied. In order to explain this point, I must discuss the $\text{AND}$ function that oversees the evaluation of the PREMISE of a rule. The reader will recall the BNF description from § 3.2.1-2:

\[
\begin{align*}
\text{DEFNOT}[C,P,LST] & \quad \text{CFmi} = -1 & \quad T & \quad \text{NIL} \\
\text{NOTDEFIS}[C,P,LST] & \quad 0.2 < \text{CFmi} < 1 & \quad T & \quad \text{NIL} \\
\text{NOTDEFNOT}[C,P,LST] & \quad -1 < \text{CFmi} < -0.2 & \quad T & \quad \text{NIL}
\end{align*}
\]

$$\text{SAME} \text{ or NOTSAME} = \text{THOUGHTNOT} \text{ or MIGHTBE} = \text{T}$$

$$\text{NOTSAME} = \text{VNOTKNOWN} \text{ or THOUGHTNOT}$$

$$\text{THOUGHTNOT} = \text{NOTDEFNOT} \text{ or DEFNOT}$$

$$\text{MIGHTBE} = \text{VNOTKNOWN} \text{ or SAME}$$

$$\text{SAME} = \text{NOTDEFIS} \text{ or DEFIS}$$

Figure 3-5: Diagram indicating the range of CF values over which the $\text{func2}$ predicates hold true. The logical relationships of these predicates are also summarized beneath the diagram.
$\text{AND}$ is similar to the standard LISP "AND" function in that it evaluates its conditional arguments one at a time, returning false (NIL) as soon as a condition is found to be false, and otherwise returning true (T). The difference is that $\text{AND}$ expects some of its conditions to return numerical values rather than simply T or NIL. If an argument condition returns NIL (or a number equal to .2 or less) it is considered false and $\text{AND}$ stops considering subsequent arguments. On the other hand, nonnumeric values of conditions are interpreted as indicating truth with $\text{CF}=1$. Thus each true condition either returns a number or a non-NIL value that is interpreted as 1. $\text{AND}$ then maintains a record of the lowest value returned by any of its arguments. This number, termed $\text{TALLY}$, is a certainty tally that indicates MYCIN’s degree of belief in the PREMISE (see Combining Function 2 in § 4.6). Thus $.2<TALLY\leq 1$, where $TALLY=1$ indicates that MYCIN believes the PREMISE to be true with certainty.

Most of the predicates that evaluate conditions in the PREMISE of a rule return either T or NIL as we have shown. Consider, however, the semantics of the most commonly used function, SAME, and its analogous function, THOUGHTNOT. Suppose MYCIN knows:

\[
\text{Val} \left[ \text{ORGANISM-2, IDENT} \right] = ((\text{STREPTOCOCCUS} .7) (\text{STAPHYLOCOCCUS} .3))
\]

Then it seems clear that $\text{SAME}[\text{ORGANISM-1, IDENT, STREPTOCOCCUS}]$ is in some sense "more true" than $\text{SAME}[\text{ORGANISM-1, IDENT, STAPHYLOCOCCUS}]$, even though both hypotheses exceed the threshold $\text{CF}=.2$. If $\text{SAME}$ merely returned T, this distinction would be lost. Thus, for this example:

\[
\text{SAME}[\text{ORGANISM-1, IDENT, STREPTOCOCCUS}] = .7 \\
\text{SAME}[\text{ORGANISM-1, IDENT, STAPHYLOCOCCUS}] = .3
\]

whereas

\[
\text{KNOWN}[\text{ORGANISM-1, IDENT}] = T
\]

and

\[
\text{NOTDEFIS}[\text{ORGANISM-1, IDENT, STREPTOCOCCUS}] = T
\]
Consultation System

A similar argument explains why THOUGHTNOT returns a CF rather than T. It is unclear whether any of the other <func2> predicates should return a CF rather than T; my present conviction is that the semantics of those functions do not require relative weightings in the way that SAME and THOUGHTNOT do.

Let me give a brief example, then, of the way in which the PREMISE of a rule is evaluated by $AND. Consider the following ORGRULE:

IF: 1) THE STAIN OF THE ORGANISM IS GRAMNEG, AND
    2) THE MORPHOLOGY OF THE ORGANISM IS ROD, AND
    3) THE AEROBICITY OF THE ORGANISM IS AEROBIC
THEN: THERE IS STRONGLY SUGGESTIVE EVIDENCE (.8) THAT THE CLASS OF THE ORGANISM IS ENTEROBACTERIACEAE

which is internally coded in LISP as:

PREMISE: ($AND (SAME CNTXT GRAM GRAMNEG)
          (SAME CNTXT MORPH ROD)
          (SAME CNTXT AIR AEROBIC))
ACTION: (CONCLUDE CNTXT CLASS
          ENTEROBACTERIACEAE TALLY .8)

Suppose this rule has been invoked for consideration of ORGANISM-1; i.e., the context of the rule (CNTXT) is the node in the context tree termed ORGANISM-1. Now suppose that MYCIN has the following information in its data base (how it gets there is the subject of § 3.3.3):

Val[ORGANISM-1,GRAM] = ((GRAMNEG 1.0))
Val[ORGANISM-1,MORPH] = ((ROD .8) (COCCUS .2))
Val[ORGANISM-1,AIR] = ((AEROBIC .6) (FACUL .4))

$AND begins by evaluating SAME[ORGANISM-1,GRAM, GRAMNEG]. The function returns CF=1.0, so TALLY is set to 1.0 (see definition of TALLY in the description of $AND above). Next $AND evaluates the second PREMISE condition, SAME [ORGANISM-1,MORPH,ROD], which returns 0.8. Since the first two conditions both were found to hold, $AND evaluates SAME [ORGANISM-1,AIR,AEROBIC] which returns 0.6. Thus, TALLY is set to 0.6 and $AND returns T. Since the PREMISE is true, MYCIN may now draw the conclusion indicated in the ACTION portion of
MYCIN

the rule. Note, however, that CONCLUDE has as arguments both .8 (i.e., the CF for the rule as provided by the expert) and TALLY (i.e., the certainty tally for the PREMISE). CONCLUDE and the other functions that control inferences are described in § 3.3.3-2.

3.2.6 (*) STATIC KNOWLEDGE STRUCTURES

Although all MYCIN's inferential knowledge is stored in rules, there are various kinds of static definitional information that are stored differently even though they are accessible from rules.

3.2.6-1 (*) Tabular and List-based Knowledge

There are three categories of knowledge structures that could be discussed in this section. However, one of them, MYCIN's 800-word dictionary, is used principally for natural language understanding. Its details are described elsewhere [Shortliffe, 1974b]. The other two data structures are simple linear lists and knowledge tables.

Simple lists: Simple lists provide a mechanism for simplifying references to variables and optimizing knowledge storage by avoiding unnecessary duplication. Two examples should be sufficient to explain this point.

In § 3.2.3-2, I showed that the EXPECT property for the clinical parameter IDENT is:

\[(\text{ONEOF (ORGANISMS)})\]

ORGANISMS is the name of a linear list containing the names of all bacteria known to MYCIN (see § 1.5.1). There is also a clinical parameter named COVERFOR for which the EXPECT property is:

\[(\text{ONEOF ENTEROBACTERIACEAE (ORGANISMS) G+COCCI G–COCCI})\]

Thus, by storing the organisms separately on a list named ORGANISMS, we avoid having to duplicate the list of names in the EXPECT property of both IDENT and COVERFOR. Furthermore, using the variable name rather than internal pointers to the list structure facilitates references to the list of organisms whenever it is needed.
Consultation System

A second example involves the several rules in the system that make conclusions based on whether an organism was isolated from a site that is normally sterile or nonsterile. **STERILESITES** is the name of a simple list containing the names of all normally sterile sites known to the system. There is a similar list named **NONSTERILESITES**. Thus many rules can have the condition (SAME CNTXT SITE STERILESITES) and the sites need not be listed explicitly in each rule.

**Knowledge tables:** In conjunction with the special functions discussed in the next subsection, MYCIN’s knowledge tables permit a single rule to accomplish a task that would otherwise require several rules. A knowledge table contains a comprehensive record of certain clinical parameters plus the values they take on under various circumstances. For example, one of MYCIN’s knowledge tables itemizes the gramstain, morphology, and aerobicity for every bacterial genus known to the system. Consider, then, the task of inferring an organism’s gram stain, morphology, and aerobicity if its identity is known with certainty. Without the knowledge table, MYCIN would require several rules of the form:

IF: THE IDENTITY OF THE ORGANISM IS DEFINITELY W
THEN: 1) IT IS DEFINITE (1) THAT THE GRAMSTAIN OF THE ORGANISM IS X, AND
2) IT IS DEFINITE (1) THAT THE MORPHOLOGY OF THE ORGANISM IS Y, AND
3) IT IS DEFINITE (1) THAT THE AEROBICITY OF THE ORGANISM IS Z

Instead MYCIN contains a single rule of the following form:

**RULE030**

IF: THE IDENTITY OF THE ORGANISM IS KNOWN WITH CERTAINTY
THEN: IT IS DEFINITE (1) THAT THESE PARAMETERS - GRAM MORPH AIR - SHOULD BE TRANSFERRED FROM THE IDENTITY OF THE ORGANISM TO THIS ORGANISM

Thus if ORGANISM-1 is known to be a streptococcus, MYCIN can use RULE030 to access the knowledge table to look up the organism’s gramstain, morphology, and aerobicity.
3.2.6-2 Specialized Functions

The efficient use of knowledge tables requires the existence of four specialized functions (the category <special-func> from § 3.2.1-2). As explained below, each function attempts to add members to a list named GRIDVAL and returns T if at least one element has been found to be placed in GRIDVAL.

Functions of the class <special-func>:

Let V = Val[C, P] be the set of all hypotheses regarding the value of the clinical parameter P for the context C.
Let CLST be a list of objects which may be characterized by clinical parameters.
Let PLST be a list of clinical parameters.
Then:

\[
\begin{align*}
\text{FUNCTION} & \quad \text{Value of GRIDVAL} \\
\text{SAME2}[C, CLST, PLST] & \quad \{ X | X \in CLST \land (\text{for all } P \in PLST) \text{SAME}[C, P, Val[X, P]] \} \\
\text{NOTSAME2}[C, CLST, PLST] & \quad \{ X | X \in CLST \land (\text{for at least one } P \in PLST) \text{NOTSAME}[C, P, Val[X, P]] \} \\
\text{SAME3}[C, P, CLST, P^*] & \quad \{ X | X \in CLST \land \text{SAME}[C, P, Val[X, P^*]] \} \\
\text{NOTSAME3}[C, P, CLST, P^*] & \quad \{ X | X \in CLST \land \text{NOTSAME}[C, P, Val[X, P^*]] \} \\
\text{GRID}[<object>, <attribute>] & \quad \{ X | X \text{ is a value of the } <attribute> \text{ of } <object> \} 
\end{align*}
\]

GRID is merely a function for looking up information in the specialized knowledge table.

The use of these functions is best explained by example. Consider the following verbalization of a rule given us by one of our collaborating experts:

If you know the portal of entry of the current organism and also know the pathogenic bacteria normally associated with that site, you have evidence that the current organism is one of those pathogens so long as there is no disagreement on the basis of gramstain, morphology, or aerobicity.

This horrendous sounding rule is coded quite easily using SAME2[C, CLST, PLST], where C is the current organism, CLST is the list of pathogenic bacteria normally associated with the portal of
Consultation System

entry of C, and PLST is the set of properties (GRAM MORPH AIR). GRID is used to set up CLST. The LISP version of the rule is:

RULE084

PREMISE:  ($AND (GRID (VAL CNTXT PORTAL) PATH-FLORA) (SAME 2 CNTXT GRIDVAL (QUOTE (GRAM MORPH AIR)))

ACTION: (CONCLIST CNTXT IDENT GRIDVAL .8)

Note that GRID sets up the initial value of GRIDVAL for use by SAME2, which then redefines GRIDVAL for use in the ACTION clause. This rule is translated (in somewhat stilted English) as follows:

RULE084

IF: 1) THE LIST OF LIKELY PATHOGENS ASSOCIATED WITH THE PORTAL OF ENTRY OF THE ORGANISM IS KNOWN, AND
2) THIS CURRENT ORGANISM AND THE MEMBERS YOU ARE CONSIDERING AGREE WITH RESPECT TO THE FOLLOWING PROPERTIES: GRAM MORPH AIR
THEN: THERE IS STRONGLY SUGGESTIVE EVIDENCE (.8) THAT EACH OF THEM IS THE IDENTITY OF THIS CURRENT ORGANISM

SAME2 and NOTSAME2 can also be used for comparing the values of the same clinical parameters for two or more different contexts in the context tree. For example:

SAME2[ORGANISM-1 (ORGANISM-2 ORGANISM-3) (GRAM MORPH)]

On the other hand, SAME3 and NOTSAME3 are useful for comparing different parameters of two or more contexts. Suppose you need a predicate that returns T if the site of a prior organism (ORGANISM-2) is the same as the portal of entry of the current organism (ORGANISM-1). This is accomplished by:

SAME3[ORGANISM-1 PORTAL (ORGANISM-2) SITE]
3.2.7 (*) TRANSLATION OF RULES INTO ENGLISH

Rules are translated into a subset of English using a set of recursive functions that piece together bits of text. I shall demonstrate the process using the PREMISE condition (GRID (VAL CNTXT PORTAL) PATH-FLORA) that is taken from RULE084 as discussed in § 3.2.6-2.

The reader will recall that every clinical parameter has a property named TRANS that is used for translation (§ 3.2.4-2). In addition, every function, simple list, or knowledge table that is used by MYCIN’s rules also has a TRANS property. For our example the following TRANS properties are relevant:

- GRID: (THE (2) ASSOCIATED WITH (1) IS KNOWN)
- VAL: (((2 1)))
- PORTAL: (THE PORTAL OF ENTRY OF *)
- PATH-FLORA: (LIST OF LIKELY PATHOGENS)

The numbers in the translations of functions indicate where the translation of the corresponding argument should be inserted. Thus the translation of GRID’s second argument is inserted for the “(2)” in GRID’s TRANS property. The extra parentheses in the TRANS for VAL indicate that the translation of VAL’s first argument should be substituted for the asterisk in the translation of VAL’s second argument. Since PORTAL is a PROP-ORG, CNTXT translates as THE ORGANISM and the translation of (VAL CNTXT PORTAL) becomes:

THE PORTAL OF ENTRY OF THE ORGANISM

Substituting VAL’s translation for the (1) in GRID’s TRANS, and PATH-FLORA’s translation for the (2), the final translation of the conditional clause becomes:

THE LIST OF LIKELY PATHOGENS ASSOCIATED WITH THE PORTAL OF ENTRY OF THE ORGANISM IS KNOWN

Similarly,

(GRID (VAL CNTXT CLASS) CLASSMEMBERS)
Consultation System

translates as:

THE LIST OF MEMBERS ASSOCIATED WITH THE CLASS OF THE ORGANISM IS KNOWN

All other portions of rules use essentially this same procedure for translation. An additional complexity arises, however, if it is necessary to negate the verbs in ACTION or ELSE clauses when the associated CF is negative. The translator program must therefore recognize verbs and know how to negate them when evidence in a PREMISE supports the negation of the hypothesis that is referenced in the ACTION of the rule.

3.3 Use of Rules to Give Advice

The discussion in § 3.2 was limited to the various data structures used to represent MYCIN’s knowledge. The present section proceeds to an explanation of how MYCIN uses that knowledge in order to give advice.

The discussion begins with a summary of previous goal-oriented or rule-based problem-solving systems. I then describe MYCIN’s control structure for selecting rules and deciding when to ask questions of the user. Subsequent sections explain the mechanisms for creation of the program’s record of the consultation. They also describe a variety of nontrivial complexities that arose during implementation of the system’s control structure.

3.3.1 PREVIOUS GOAL-ORIENTED PROBLEM-SOLVING SYSTEMS

Early AI research on machine reasoning concentrated on programs that could solve simple puzzles. From this work a number of problem-solving techniques were developed, many of which continue to pervade AI investigation. These have been summarized as follows [Nilsson, 1974]:

(1) heuristic search
(2) problem spaces and states
(3) operators for state transformations  
(4) goal and subgoal states  
(5) means-ends analysis  
(6) reasoning backwards

I will not attempt to discuss all of these here, but will concentrate instead on the techniques used by the four “rule-based” systems that were selected for discussion in § 3.2.1-1 and on the various methodologies for goal-oriented problem-solving.

Although MYCIN shares its rule-based knowledge representation with several other AI programs, none of the systems described in § 3.2.1-1 uses its rules in the way that MYCIN does. Waterman’s system, for example, makes decisions by comparing the current state vector with the “situation” portion of the SA rules [Waterman, 1970]. The rules are maintained in an ordered list and the matching-search begins with the first rule in the list. Searching stops as soon as a match is found; thus the first matched rule defines the program’s “move” in the poker game. Subsequent rules in the list that might also match the current state vector are ignored. As a result, the order of rules in the rule-list is of crucial importance. In general, the most specific rules are placed early in the list so that they effectively filter out state vectors that are well-characterized and for which well-defined heuristics exist.

Although system knowledge is kept modular by the SA rule approach, the rules are implicitly interrelated by their ordering in the list. Furthermore, in HEURISTIC DENDRAL [Buchanan, 1969], the interrelationships may be explicit in that the action portion of one rule may include a pointer to one or more other rules. As a result, integration of new rules and modifications to old knowledge may be complicated. Waterman’s program attempts to learn new heuristics for incorporation into the ordered list of rules, and META-DENDRAL [Buchanan, 1972], also tackles the problem of generalization (theory formation). Both programs must therefore select the appropriate location or mechanism for incorporating a new rule and, in some cases, must modify other rules so that the new SA heuristic will be invoked under appropriate circumstances.

Colby’s system [Colby, 1969] interrelates its rules in a directed graph [Tesler, 1968]. In judging the credibility of a proposition P, the program looks for relevant beliefs in the graph structure. A
directly relevant belief is one that can be derived from P in a single step. These beliefs then serve as the “heads” of paths in the graph to be searched. Therefore, Colby’s system clearly depends upon explicit interrelationships of both inferential rules and “facts” (see § 3.2.1-1). Furthermore, the program uses the rules primarily as a kind of pattern matching mechanism during the evaluation of the proposition in question. Despite its use of rules, the program is not really a problem-solving system and its similarity to MYCIN is therefore largely superficial.

Green’s QA3, on the other hand, is a problem-solving system with a theoretical foundation firmly linked to the puzzle-solving programs that I mentioned above [Nilsson, 1974]. As explained in § 3.2.1-1, QA3’s task is to use axioms and theorems (expressed in the first-order predicate calculus) to answer questions [Green, 1969]. Questions are themselves expressed as theorems (rules) and the program attempts to derive the theorem from its knowledge-base. The steps in the proof are remembered and then form the basis of the answer to the question. Thus the question (expressed as a theorem) is a “goal-statement” and the program must have mechanisms for selecting relevant pieces of knowledge that can be combined to accomplish the goal.

QA3’s technique for combining knowledge is a modified form of the resolution principle [Robinson, 1965]. The principle explains how to derive a new logical statement, when possible, from a specified pair of clauses. However, a variety of additional strategies is needed for deciding which pieces of knowledge to attempt to resolve. Green’s technique is to try to show that the negation of the question is inconsistent with the rest of the system’s knowledge. Aided by heuristic search strategies including the set-of-support [Wos, 1965], unit preference [Wos, 1964] and subsumption [Robinson, 1965], QA3 works backwards from the negation of the question, attempting to derive a contradiction. Thus, this theorem-proving approach may be considered goal-oriented in that it works backwards from its goal rather than resolving knowledge clauses at random in hopes of eventually deriving the answer to the question under consideration.

Another intuitively pleasing technique that has found application within the realm of problem-solving [Fikes, 1971; Newell, 1961] is known as means–ends analysis. Often explained in terms of state transition, the technique is based upon the recognition of differences
between the current state of the system and the desired state (goal). As a result, useful intermediate states (subgoals) can be defined so that the problem may be reduced to a number of subproblems, each much easier than the total task. Plans for accomplishing each subgoal may then be combined to create a total strategy for achieving the goal.

It is not always natural to express knowledge in terms of operators for state transition, however. As early as 1957, a system was introduced to solve logical problems by working backwards from the goal without means–ends analysis [Newell, 1957]. More recent systems have also utilized the goal-oriented approach [Hewitt, 1969; Rulifson, 1972]. In fact, the consequent theorems of PLANNER [Hewitt, 1972] (implemented in MICRO-PLANNER; see also § 3.2.1-1), provide a control mechanism for knowledge use which seems strikingly similar to those that should ideally be used for medical decision making. I will attempt to justify this claim after a brief description of PLANNER's deductive mechanisms. The examples used here are taken from a recent discussion of AI languages [Bobrow, 1973].

PLANNER's data types include assertions, goals, and theorems. Consider, for example, a program that knew the following facts:

\[
(PART \text{ ARM } \text{ PERSON}) \\
(PART \text{ HAND } \text{ ARM}) \\
(PART \text{ FINGER } \text{ HAND})
\]

where these stand for attribute-object-value triples such as those I discussed in § 3.2.3-1. Suppose the program were now asked to decide whether a finger is part of a person, i.e.:

\[
(\text{GOAL } (PART \text{ FINGER } \text{ PERSON}))
\]

The PLANNER "GOAL" formalism first looks to see if the fact appears in the program's knowledge-base. Since it does not, it looks instead for a "consequent theorem" with a pattern that matches the GOAL statement (PART FINGER PERSON). Variable positions in patterns are characters preceded by ' $? '. Thus the following consequent theorem matches the GOAL:

\[
\text{CONSEQUENT} \\
(PART $?X $?Z) \quad \leftarrow \text{(pattern)} \\
(\text{GOAL } (PART $?X $?Y)) \\
(\text{GOAL } (PART $?Y $?Z))
\]
When instantiated for the current GOAL, the theorem becomes:

\[
\text{(CONSEQUENT} \\
\quad \text{(PART FINGER PERSON)} \\
\quad \text{(GOAL (PART FINGER $?Y))} \\
\quad \text{(GOAL (PART $?Y PERSON))})
\]

or, in words, to show that a finger is part of a person, find something ($?Y$) of which a finger is a part and which itself is a part of a person. Thus the program has two new instantiated GOAL statements:

\[
\text{(GOAL (PART FINGER $?Y))} \\
\text{(GOAL (PART $?Y PERSON))}
\]
The first GOAL statement immediately finds from its knowledge base that (PART FINGER $?Y) holds for $?Y = HAND. Thus the second GOAL becomes (GOAL (PART HAND PERSON)); this can, in turn, be derived by recursive use of the consequent theorem given above. Figure 3-6 diagrams the reasoning network that develops below the initial GOAL. Note that the terminal nodes in this little tree correspond to facts already in the data base. As is later shown, MYCIN's decision process may also be diagrammed as a reasoning network with a goal at the top and known data as terminal nodes.

Another PLANNER construct is the "antecedent theorem." Whenever anything is asserted in a PLANNER program, i.e., added to the data base, the system compares the new knowledge with the pattern portion of all antecedent theorems in the system. Continuing the example from above, consider the following theorem:

\[
\text{(ANTECEDENT}
\begin{align*}
\text{(PART $?X $?Y)} & \leftarrow \text{(pattern)} \\
\text{(GOAL (PART $?Y $?Z))} \\
\text{(ASSERT (PART $?X $?Z))}
\end{align*}
\]

Suppose the assertion (PART FINGER HAND) were now added to a program that already knew (PART HAND ARM) and (PART ARM PERSON). The new assertion would match the pattern in the theorem above (note I have not yet mentioned any GOALs) and would therefore invoke the following instantiation:

\[
\text{(ANTECEDENT}
\begin{align*}
\text{(PART FINGER HAND)} \\
\text{(GOAL (PART HAND $?Z))} \\
\text{(ASSERT (PART FINGER $?Z))}
\end{align*}
\]

which says, in words, that since a finger is part of a hand, if you can find something ($?Z$) which a hand is part of then you can assert that a finger is part of it too. (GOAL (PART HAND $?Z)) is in this case easily proven from the data base by setting $?Z$ to ARM. Thus the antecedent theorem succeeds and asserts (PART FINGER ARM). However, this new assertion also matches the pattern portion of the antecedent theorem, so the theorem is once again invoked. This time the observation (PART ARM PERSON) leads to the conclusion (PART FINGER PERSON).
A potential problem with antecedent theorems, as should be clear from this example, is that they have a capability to clutter up the system's knowledge-base with facts (assertions) that will never be used in achieving goals. When used judiciously they are powerful mechanisms for simplifying future goals that are likely to need the generated assertions, but the consequent theorems suggest a sense of purpose which is highly appealing for problem-solving applications.

The distinction between consequent and antecedent theorems provides a useful basis for considering some of the different approaches to the medical diagnosis problem. Antecedent theorems may in one sense be compared with a comprehensive process for medical data collection. Clinical screening exams, of course, have their place (§ 1.2.2-3), but medical education tends to stress the rational selection of tests based upon indications in the patient. The alternate approach is to order every test imaginable (including a lengthy history and physical exam) and then to sift through the data in hopes of recognizing unusual patterns or clusters of symptoms that may lead to a diagnosis. The second alternative is not only expensive and time-consuming, but it also requires remarkably little analytical skill on the part of the clinician. The approach does occur, however, particularly among medical students before their clinical skills are well-developed.

The selection of tests on the basis of specific indications, on the other hand, indicates an organized approach to problem-solving that parallels that found in consequent theorems. The good clinician tends to work backwards from his goal (i.e., to diagnose and treat his patient appropriately), making hypotheses and selecting tests in accordance with his desire to minimize unnecessary time-delays or monetary expenditures. This comparison to PLANNER-type consequent theorems may at first seem rather vague, but I shall show in subsequent sections that MYCIN indeed does reason backwards, avoiding the "shotgun approach" of a diagnostic system based solely upon mechanisms analogous to antecedent theorems.

3.3.2 MYCIN'S CONTROL STRUCTURE

MYCIN's rules are directly analogous to the PLANNER consequent theorems discussed in § 3.3.1. They permit a reasoning chain (see Figure 3-6) to grow dynamically on the basis of the user's
answers to questions regarding the patient. In this subsection, I describe that reasoning network, explaining how it grows and how MYCIN manages to ask questions only when there is a reason for doing so.

3.3.2-1 Consequent Rules and Recursion

As discussed in § 1.4.1, MYCIN's task involves a four-stage decision problem:

1. Decide which organisms, if any, are causing significant disease;
2. Determine the likely identity of the significant organisms;
3. Decide which drugs are potentially useful;
4. Select the best drug or drugs.

Steps 1 and 2 are closely interrelated since determination of an organism's significance may well depend upon its presumed identity. Furthermore, MYCIN must consider the possibility that the patient has an infection with an organism not specifically mentioned by the user (e.g., an occult abscess suggested by historical information or subtle physical findings). Finally, if MYCIN decides that there is no significant infection requiring antimicrobial therapy, it should skip steps 3 and 4, advising the user that no treatment is thought to be necessary. MYCIN's task area therefore can be defined by the following rule:

RULE092

IF: 1) THERE IS AN ORGANISM WHICH REQUIRES THERAPY, AND
2) CONSIDERATION HAS BEEN GIVEN TO THE POSSIBLE EXISTENCE OF ADDITIONAL ORGANISMS REQUIRING THERAPY, EVEN THOUGH THEY HAVE NOT ACTUALLY BEEN RECOVERED FROM ANY CURRENT CULTURES

THEN: DO THE FOLLOWING:
1) COMPILE THE LIST OF POSSIBLE THERAPIES WHICH, BASED UPON SENSITIVITY DATA, MAY BE EFFECTIVE AGAINST THE ORGANISMS REQUIRING TREATMENT, AND
Consultation System

2) DETERMINE THE BEST THERAPY 
RECOMMENDATIONS FROM THE COMPILED LIST 
OTHERWISE: INDICATE THAT THE PATIENT DOES NOT REQUIRE 
THERAPY

This rule is one of MYCIN’s PATRULES (i.e., its context is the patient; see § 3.2.2-2) and is known as the “goal rule” for the system. A consultation session with MYCIN results from a simple two-step procedure (Subprogram 1 shown in Figure 1-1):

(1) Create the patient context as the top node in the context tree 
(see § 3.4 for an explanation of how nodes are added to the tree) 
(2) Attempt to apply the goal-rule to the newly created patient context 

After the second step, the consultation is over and Subprogram 1 relinquishes control to the Explanation System (Subprogram 2 shown in Figure 1-1). My purpose here, then, is to explain how the simple attempt to apply the goal rule to the patient causes a lengthy consultation with an individualized reasoning chain.

When MYCIN first tries to evaluate the PREMISE of the goal rule, the first condition requires that it know whether there is an organism that requires therapy. MYCIN then reasons backwards in a manner that may be informally paraphrased as follows:

How do I decide whether there is an organism requiring therapy? Well, RULE090 tells me that organisms associated with significant disease require therapy. But I don’t even have any organisms in the context tree yet, so I’d better ask first if there are any organisms and if there are I’ll try to apply RULE090 to each of them. However, the PREMISE of RULE090 requires that I know whether the organism is significant. I have a bunch or rules for making this decision (RULE038 RULE042 RULE044 RULE108 RULE122). For example, RULE038 tells me that if the organism came from a sterile site it is probably significant. Unfortunately I don’t have any rules for inferring the site of a culture, however, so I guess I’ll have to ask the user for this information when I need it . . .

This goal-oriented approach to rule invocation and question selection is automated via two interrelated procedures, a MONITOR that
THE MONITOR FOR RULES

START

CONSIDER THE FIRST CONDITION IN THE PREMISE OF THE RULE

HAS ALL NECESSARY INFORMATION BEEN GATHERED TO DECIDE IF THE CONDITION IS TRUE?

GATHER THE NECESSARY INFORMATION USING THE FINDOUT MECHANISM

IS THE CONDITION TRUE?

no (or unknown)

REJECT THE RULE

EXIT

yes

ADD THE CONCLUSION OF THE RULE TO THE ONGOING RECORD OF THE CURRENT CONSULTATION

EXIT

no

ARE THERE MORE CONDITIONS TO CHECK?

yes

CONSIDER THE NEXT CONDITION IN THE PREMISE

Figure 3-7: Flow chart describing the rule MONITOR that analyzes a rule and decides whether it applies in the clinical situation under consideration. Each condition in the PREMISE of the rule references some clinical parameter, and all such conditions must be true for the rule to be accepted. [Reproduced from Computers and Biomedical Research [Shortliffe, 1975b] with permission of the publishers.]
Consultation System

analyzes rules and a FINDOUT mechanism that searches for data needed by the MONITOR.

The MONITOR analyzes the PREMISE of a rule, condition by condition, as shown in Figure 3-7. (As discussed in § 3.2.5, the MONITOR uses the $\text{AND}$ function to oversee the PREMISE evaluation.) When the value of the clinical parameter referenced in a condition is not yet known to MYCIN, the FINDOUT mechanism is

THE FINDOUT MECHANISM

Figure 3-8: Flow chart describing the strategy for determining which questions to ask the physician. The derivation of values of parameters may require recursive calls to the MONITOR, thus dynamically creating a reasoning chain specific to the patient under consideration. [Reproduced from Computers and Biomedical Research [Shortliffe, 1975b] with permission of the publishers.]
invoked in an attempt to obtain the missing information. FINDOUT then either derives the necessary information (from other rules) or asks the user for the data.

FINDOUT has a dual strategy depending upon the kind of information required by the MONITOR. This distinction is demonstrated in Figure 3-8. In general, a piece of data is immediately requested from the user (an ASK1 question) if it is considered in some sense "primitive," as are, for example, most laboratory data. Thus, if the physician knows the identity of an organism (e.g., from a lab report), we would prefer that the system request that information directly rather than try to deduce it via decision rules. However, if the user does not know the identity of the organism, MYCIN uses its knowledge base in an effort to deduce the range of likely organisms.

"Nonlaboratory data" are those kinds of information that require inference even by the clinician: e.g., whether an organism is a contaminant or a previously administered drug was effective. FINDOUT always attempts to deduce such information first, asking the physician only when MYCIN's knowledge-base of rules is inadequate for making the inference from the information at hand (an ASK2 question).

In § 3.2.3-2 I described the representation of clinical parameters and their associated properties. The need for two of these properties, LABDATA and UPDATED-BY, should now be clear. The LABDATA flag for a parameter allows FINDOUT to decide which branch to take through its decision process (Figure 3-8). Thus, IDENT is marked as being LABDATA in Figure 3-2.

Recall that the UPDATED-BY property is a list of all rules in the system that permit an inference to be made regarding the value of the indicated parameter. Thus, UPDATED-BY is precisely the list I have called Y in Figure 3-8. Every time a new rule is added to MYCIN's knowledge-base, the name of the rule is added to the UPDATED-BY property of the clinical parameter referenced in its ACTION or ELSE clause. Thus, the new rule immediately becomes available to FINDOUT at times when it may be useful. It is not necessary explicitly to specify its interrelationships with other rules in the system.

Note that FINDOUT is accessed from the MONITOR, but the MONITOR may also be accessed from FINDOUT. This recursion allows self-propagation of a reasoning network appropriate for the
Consultation System

patient under consideration and selects only the necessary questions and rules. The first rule passed to the MONITOR is always the goal rule. Since the first condition in the PREMISE of this rule references a clinical parameter of the patient named TREATFOR, and since the value of TREATFOR is, of course, unknown before any data have been gathered, the MONITOR asks FINDOUT to trace the value of TREATFOR. This clinical parameter is not a LABDATA so FINDOUT takes the left-hand pathway in Figure 3-8 and sets Y to the UPDATED-BY property of TREATFOR, the two-element list (RULE090 RULE149). The MONITOR is then called again with RULE090 as the rule for consideration, and FINDOUT is utilized to trace the values of clinical parameters referenced in the PREMISE of RULE090. Note that this process parallels the verbal description of MYCIN's reasoning that was given above. (The reference to tree propagation, however, will not be explained until § 3.4.)

It is important to recognize that FINDOUT does not check to see whether the PREMISE condition is true. Instead the FINDOUT mechanism traces the clinical parameter exhaustively and returns its value to the MONITOR where the conditional expression may then be evaluated. (The process is slightly different for "multivalued" parameters; see § 3.3.2-2.) Hence FINDOUT is called at most one time for a clinical parameter (in a given context, see § 3.4). When FINDOUT returns a value to the MONITOR it marks the clinical parameter as having been traced. Thus, when the MONITOR reaches the question "HAS ALL NECESSARY INFORMATION BEEN GATHERED TO DECIDE IF THE CONDITION IS TRUE?" (Figure 3-7), the parameter is immediately passed to FINDOUT unless it has been previously marked as traced.

Figure 3-9 is a portion of MYCIN's initial reasoning chain. A comparison with Figure 3-6 will reemphasize the similarities between MYCIN's control structure and the goal-oriented consequent theorems used by PLANNER. In Figure 3-9 the clinical parameters being traced are underlined. Thus REGIMEN is the top goal of the system (i.e., it is the clinical parameter in the ACTION clause of the goal rule). Below each parameter are the rules (from the UPDATED-BY property) which may be used for inferring the parameter's value. Clinical parameters referenced in the PREMISE of these rules are then listed at the next level in the reasoning network. Rules with multiple PREMISE conditions have their links numbered in accor-
Figure 3-9: An example of the kind of reasoning network generated by the MONITOR and FINDOUT mechanisms. Names of clinical parameters are underlined. When rules have multiple conditions in their PREMISE, a number has been included to specify the position of the associated clinical parameter within the PREMISE conditions.
Consultation System

dance with the order in which the parameters are traced (by FINDOUT). ASK1 indicates that a parameter is LABDATA so its value is automatically asked of the user when it is needed. ASK2 refers to parameters that are not LABDATA but for which no inference rules currently exist, e.g., whether the dose of a drug is adequate. One of the goals in the future development of MYCIN’s knowledge base is to acquire enough rules allowing the values of non-LABDATA parameters to be inferred so that ASK2 questions need no longer occur.

Note that the reasoning network in Figure 3-9 is drawn to reflect maximum size. In reality many portions of such a network need not be considered. For example, RULE042 (one of the UPDATED-BY rules under SIGNIFICANCE) is rejected if the SITE condition is found to be false by the MONITOR. When that happens, neither COLLECT nor SIGNUM need to be traced by FINDOUT and those portions of the reasoning network are not created. Thus, the order of conditions within a PREMISE is highly important. In general, conditions referencing the parameters that are most common (i.e., which appear in the PREMISE of the most rules) are put first in the PREMISE of new rules to act as an effective screening mechanism.

A final comment is necessary regarding the box labelled “REJECT THE RULE” in Figure 3-7. This step in the MONITOR actually must check to see if the rule has an ELSE clause. If so, and if the PREMISE is known to be false, the conclusion indicated by the ELSE clause is drawn. If there is no ELSE clause, or if the truth status of the PREMISE is uncertain (e.g., the user has entered UNKNOWN when asked the value of one of the relevant parameters, see § 3.3.2-2), the rule is simply ignored.

3.3.2-2 Asking Questions of User

As was emphasized in Chapter 2, the conventions for communication between a program and the physician are a primary factor determining the system’s acceptability. We have therefore designed a number of features intended to simplify the interactive process that occurs when FINDOUT reaches one of the boxes entitled “ASK THE USER FOR THE VALUE OF THE PARAMETER” (Figure 3-8).

When MYCIN requests the value of a “single-valued” or “yes-no” parameter, it uses the PROMPT property as described in § 3.2.3-2.
The user's response is then compared with the EXPECT property of the parameter. If his answer is one of the expected responses, the program simply continues through the reasoning network. Otherwise, MYCIN checks the system dictionary to see if the user's response is a synonym for one of the recognized answers. If this attempt also fails, MYCIN uses INTERLISP spelling-correction routines [Teitelman, 1974] to see if a simple spelling or typographical error will account for the unrecognized response. If so, the program makes the correction, prints its assumption, and proceeds as though the user had made no error. If none of these mechanisms succeeds, MYCIN tells the user that his response is not recognized, displays a list of sample responses, and asks the question again. Examples of these features are included in the sample consultation session at the end of Chapter 1.

"Multivalued" parameters are handled somewhat differently. FINDOUT recursively traces such parameters in the normal fashion, but when forced to ask a question of the user it customizes its question to the condition being evaluated in the MONITOR. Suppose, for example, the MONITOR were evaluating the condition (SAME CNTXT INFECT MENINGITIS), i.e., "Meningitis is an infectious disease diagnosis for the patient." If FINDOUT were to ask the question using the regular PROMPT strategy, it would request:

"What is the infectious disease diagnosis for PATIENT-1?"

The problem is that the patient may have several diagnoses, each of which can be expressed in a variety of ways. If the physician were to respond:

"A meningeal inflammation that is probably of infectious origin"

MYCIN would be forced to try to recognize that this answer implies meningitis. Our solution has been to customize questions for "multivalued" parameters to reflect the value being checked in the current PREMISE condition. The PROMPT1 property is used, and questions always expect a yes–or–no response:

"Is there evidence that the patient has a meningitis?"
## Consultation System

The advantages of this approach are the resulting ability to avoid natural language processing during the consultation itself, and the posing of questions that are specific to the patient under consideration.

In addition to the automatic spelling-correction capability described above, the user is given a number of options that may be utilized whenever MYCIN asks him a question:

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>UNKNOWN</strong></td>
<td>(may be abbreviated U or UNK) used to indicate that the physician does not know the answer to the question, usually because the data are unavailable.</td>
</tr>
<tr>
<td>?</td>
<td>used to request a list of sample recognized responses.</td>
</tr>
<tr>
<td>??</td>
<td>used to request a list of all recognized responses.</td>
</tr>
<tr>
<td>RULE</td>
<td>used to request that MYCIN display the translation of the current decision rule. FINDOUT simply translates the rule being considered by the MONITOR. This feature provides a simple capability for explaining why the program is asking the question. However, it cannot explain motivation beyond the current decision rule.</td>
</tr>
<tr>
<td>QA</td>
<td>used to digress temporarily in order to use the Explanation System (Subprogram 2). The features of this system are explained in Chapter 5.</td>
</tr>
<tr>
<td>WHY</td>
<td>used to request a detailed explanation of the question being asked. This feature is much more conversational than the RULE option above and permits investigation of the current state of the entire reasoning chain. This explanation capability is described elsewhere [Shortliffe, 1975b; Davis, 1976].</td>
</tr>
<tr>
<td>CHANGE XXX</td>
<td>used to change the answer to a previous question. Whenever MYCIN asks a question it prints a number in front of the prompt. Thus CHANGE 4 means “Go back and let me reanswer question #4”. The complexities involved in this process are discussed in § 3.6.1.</td>
</tr>
<tr>
<td>STOP</td>
<td>halts the program without completing the consultation.</td>
</tr>
<tr>
<td>HELP</td>
<td>prints this list.</td>
</tr>
</tbody>
</table>

### 3.3.3 CREATION OF DYNAMIC DATA BASE

Figure 1-1 showed that the Consultation System maintains an ongoing record of the consultation. These dynamic data include
MYCIN

information entered by the user, inferences drawn using decision rules, and record-keeping data structures that facilitate question answering by the Explanation System (see Chapter 5).

3.3.3-1 Data Acquired from User

Except for questions related to propagation of the context tree, all queries from MYCIN to the physician request the values of specific clinical parameters for specific nodes in the context tree. The FINDOUT mechanism screens the user’s response, as described in § 3.3.2-2, stores it in MYCIN’s dynamic data base, and returns the value to the MONITOR for evaluation of the conditional statement that generated the question in the first place (§ 3.3.2-1). The physician’s response is stored, of course, so that future rules containing conditions referencing the same clinical parameter will not force the question to be asked a second time.

As we noted in § 3.2.4, however, the values of clinical parameters are always stored along with their associated certainty factors. A physician’s response must therefore have a CF associated with it. MYCIN’s convention is to assume CF=1 for the response unless the physician explicitly states otherwise. Thus the following exchange:

7) Staining characteristics of ORGANISM-1 (gram):
   **GRAMNEG

results in:

Val[ORGANISM-1,GRAM] = ((GRAMNEG 1.0))

If, on the other hand, the user thinks he knows the answer to a question but wants to indicate his uncertainty, he may enter a certainty factor in parentheses after his response. MYCIN expects the number to be an integer between -10 and +10; the program divides the number by 10 to obtain a CF. Using integers simplifies the user’s response and also discourages comparisons between the number and a probability measure. Thus the following exchange:

8) Enter the identity (genus) of ORGANISM-1:
   **ENTEROCOCCUS (8)
Consultation System

results in

\[
\text{Val[ORGANISM-1,IDENT]} = ((\text{STREPTOCOCCUS-GROUP-D .8}))
\]

This example also shows how the dictionary is used to put synonyms into standardized form for the patient’s data base (i.e., enterococcus is effectively another name for a group-D streptococcus).

A variant of this last example is the user’s option to enter multiple responses to a question so long as each is modified by a CF. For example:

13) Did ORGANISM-2 grow in clumps, chains, or pairs?

**CLUMPS (6) CHAINS (3) PAIRS (-8)

results in

\[
\text{Val[ORGANISM-2,CONFORM]} = ((\text{CLUMPS .6}) (\text{CHAINS .3}) (\text{PAIRS -.8}))
\]

The CF’s associated with the parameter values are then used for evaluation of PREMISE conditions as described in § 3.2.5. Note that the user’s freedom to modify his answers increases the flexibility of MYCIN’s reasoning. Without the CF option, the user might well have responded UNKNOWN to question 13 above. The demonstrated answer, although uncertain, gives MYCIN much more information than would have been provided by an UNKNOWN.

3.3.3-2 Data Inferred by System

This subsection explains the <conclusion> item from the BNF rule description (§ 3.2.1-2), i.e., the functions that are used in ACTION or ELSE clauses when a PREMISE has shown that an indicated conclusion may be drawn. There are only three such functions, two of which (CONCLIST and TRANSLIST) reference knowledge tables (§ 3.2.6) but are otherwise dependent upon the third, a function called CONCLUDE. CONCLUDE takes five arguments:

- **CNTXT** - the node in the context tree about which the conclusion is being made
- **PARAM** - the clinical parameter whose value is being added to the dynamic data base
MYCIN

VALUE - the inferred value of the clinical parameter
TALLY - the certainty tally for the PREMISE of the rule (see § 3.2.4)
CF - the certainty factor for the rule as judged by the expert from whom the rule was obtained

The translation of CONCLUDE depends upon the size of CF:

- $\text{CF} \geq .8$ - "There is strongly suggestive evidence that . . ."
- $.4 \leq \text{CF} < .8$ - "There is suggestive evidence that . . ."
- $\text{CF} < .4$ - "There is weakly suggestive evidence that . . ."
- Computed $\text{CF}$ - "There is evidence that . . ."

Thus the following conclusion:

(CONCLUDE CNTXT IDENT STREPTOCOCCUS TALLY .7)

translates as:

THERE IS SUGGESTIVE EVIDENCE (.7) THAT THE IDENTITY OF THE ORGANISM IS STREPTOCOCCUS

If, for example, the rule with this ACTION clause were successfully applied to ORGANISM-1, an organism for which no previous inferences had been made regarding identity, the result would be:

$\text{Val}[\text{ORGANISM-1,IDENT}] = (\text{STREPTOCOCCUS X})$

where X is the product of .7 and TALLY (see Combining Function 4, § 4.6). Thus the strength of the conclusion reflects both the CF for the rule and the extent to which the PREMISE of the rule is believed to be true for ORGANISM-1.

Suppose a second rule were now found which contained a PREMISE true for ORGANISM-1 and which added additional evidence to the assertion that the organism is a streptococcus. This new evidence somehow has to be combined with the CF (=X) that is already stored for the hypothesis that ORGANISM-1 is a streptococcus. If Y is the CF calculated for the second rule (i.e., the product of the TALLY for that rule and the CF assigned to the rule by the expert), the CF for the hypothesis is updated to Z so that:

132
Consultation System

\[ \text{Val[ORGANISM-1,IDENT] = ((STREPTOCOCCUS Z))} \]

where Combining Function 1 gives \( Z = X + Y(1-X) \). This function is justified and discussed in detail in § 4.6.

Similarly, additional rules leading to alternate hypotheses regarding the identity of ORGANISM-1 may be successfully invoked. The new hypotheses, along with their associated CF’s, are simply appended to the list of hypotheses in \( \text{Val[ORGANISM-1,IDENT]} \). Note, of course, that the CF’s of some hypotheses may be negative, indicating there is evidence suggesting that the hypothesis is not true. When there is both positive and negative evidence for a hypothesis, Combining Function 1 must be used in a modified form. See Chapter 4 for these details, especially § 4.7 where MYCIN’s use of the CF model is discussed with an example.

A final point to note is that values of parameters are stored identically regardless of whether the information has been inferred or acquired from the user (§ 3.3.3-1). The source of a piece of information is maintained in a separate record (§ 3.3.3-3). It is therefore easy to incorporate new rules that infer values of parameters for which ASK2 questions to the user once were necessary.

3.3.3-3 Creating an Ongoing Consultation Record

In addition to information provided or inferred regarding nodes in the context tree, MYCIN’s dynamic data base contains a record of the consultation session. This record provides the basis for answering questions about the consultation (Chapter 5).

There are two general types of records kept. One is information about how values of clinical parameters were obtained. If the value was inferred using rules, a record of those inferences is stored with the rules themselves. Thus whenever an ACTION or ELSE clause is executed, MYCIN keeps a record of the details.

The second record provides a mechanism for explaining why questions were asked. MYCIN maintains a list of questions, their identifying number, the clinical parameter and context involved, plus the rule that led to generation of the question. The program then uses this list in responding to the EQ option (see Chapter 5) during interactive sessions between the physician and Subprogram 2.
3.3.4 (*) SELF-REFERENCING RULES

As new rules were acquired from the collaborating experts, it became apparent that MYCIN would need a small number of rules that departed from the strict modularity to which we had otherwise been able to adhere. For example, one expert indicated that he would tend to ask about the typical pseudomonas-type skin lesions only if he already had reason to believe that the organism was a pseudomonas. If the lesions were then said to be evident, however, his belief that the organism was a pseudomonas would be increased even more. A rule reflecting this fact must somehow imply an orderedness of rule invocation, i.e., “Don’t try this rule until you have already traced the identity of the organism by using other rules in the system”. Our solution has been to reference the clinical parameter early in the PREMISE of the rule as well as in the ACTION. For example:

RULE040

IF: 1) THE SITE OF THE CULTURE IS BLOOD, AND
     2) THE IDENTITY OF THE ORGANISM MAY BE PSEUDOMONAS, AND
     3) THE PATIENT HAS ECTHYMA GANGRENO SUM SKIN LESIONS
THEN: THERE IS STRONGLY SUGGESTIVE EVIDENCE (.8) THAT THE IDENTITY OF THE ORGANISM IS PSEUDOMONAS

Note that RULE040 is thus a member of both the LOOKAHEAD property and the UPDATED-BY property for the clinical parameter IDENT. Rules with the same parameter in both PREMISE and ACTION are termed “self-referencing” rules. The ordered invocation of such rules is accomplished by a generalized procedure described below.

As discussed in § 3.3.2-1, a rule such as RULE040 is originally invoked because MYCIN is trying to infer the identity of an organism, i.e., FINDOUT is asked to trace the parameter IDENT and recursively sends the UPDATED-BY list for that parameter to the MONITOR. When the MONITOR reaches RULE040, however, the second PREMISE condition references the same clinical parameter
Consultation System

currently being traced by FINDOUT. If the MONITOR merely passed IDENT to FINDOUT again (as called for by the simplified flow chart in Figure 3-7), FINDOUT would begin tracing IDENT for a second time, RULE040 would be passed to the MONITOR yet again, and an infinite loop would occur.

The solution to this problem is to let FINDOUT screen the list I call Y in Figure 3-8, i.e., the UPDATED-BY property for the parameter it is about to trace. Y is partitioned by FINDOUT into regular rules and self-referencing rules (where the latter category is defined as those rules that also occur on the LOOKAHEAD list for the clinical parameter). FINDOUT passes the first group of rules to the MONITOR in the normal fashion. After all these rules have been tried, FINDOUT marks the parameter as having been traced and then passes the self-referencing rules to the MONITOR. In this way, when the MONITOR considers the second condition in the PREMISE of RULE040, the conditional is evaluated without a call to FINDOUT because the parameter has already been marked as traced. Thus, the truth of the PREMISE of a self-referencing rule is determined on the basis of the set of non-self-referencing rules that were first evaluated. If one of the regular rules permitted MYCIN to conclude that an organism might be a pseudomonas, RULE040 might well succeed when passed to the MONITOR. Clearly this mechanism for handling self-referencing rules satisfies the intention of an expert when he gives us decision criteria in self-referencing form.

It should be noted that this approach minimizes the potential for self-referencing rules to destroy certainty factor commutativity. By holding these rules to the last we insure that the certainty tally for their PREMISE (see TALLY, § 3.2.5) is the same regardless of the order in which the non-self-referencing rules were executed. If there is more than one self-referencing rule that is successfully executed for a given context and parameter, however, the order of their invocation may affect the final CF. The approach we have currently implemented thus seeks merely to minimize the potential inconsistent effects of self-referencing rules.

3.3.5 (*) PREVENTING REASONING LOOPS

Self-referencing rules are actually a special case of a more general problem. Reasoning loops involving multiple rules cannot be handled
by the mechanism described in § 3.3.4. The difference is that self-referencing rules are intentional parts of MYCIN’s knowledge base whereas reasoning loops are artifacts that must somehow be avoided.

For the following discussion, I introduce the following notation:

\[ q \ x : : > y \]

means that decision rule “q” uses clinical parameter X to reach a conclusion regarding the value of clinical parameter Y. Thus, a self-referencing rule may be represented by:

\[ a \ E : : > E \]

where E is the clinical parameter that is referenced in both the PREMISE and the ACTION of the rule. Consider now the following set of rules:

\[
\begin{align*}
[1] \ & A : : > B \\
[2] \ & B : : > C \\
[3] \ & C : : > D \\
[4] \ & D : : > A \\
\end{align*}
\]

Statement [1], for example, says that under certain unspecified conditions, the value of A can be used to infer the value of B. Now suppose that the MONITOR asks FINDOUT to trace the clinical parameter D. Then MYCIN’s recursive mechanism would create the following reasoning chain:


The difference between this looped reasoning chain and a self-referencing rule is that rule [4] was provided as a mechanism for deducing the value of A, not for reinforcing the system’s belief in the value of D. In cases where the value of A is of primary interest, the use of rule [4] would be appropriate. MYCIN solves this problem by keeping track of all parameters currently being traced by the FINDOUT mechanism. The MONITOR then simply ignores a rule if one of the parameters checked in its PREMISE is already being
Consultation System

The result, with the value of D as the goal, is a three-membered reasoning chain in the case above:

\[ A : \Rightarrow B : \Rightarrow C : \Rightarrow D \]

Rule [4] is rejected because parameter D is already being traced elsewhere in the current reasoning chain. If the value of A were the main goal, however, the chain would be:

\[ B : \Rightarrow C : \Rightarrow D : \Rightarrow A \]

Note that this simple mechanism allows us to have potential reasoning loops in the knowledge-base but to select only the relevant nonlooping portions for consideration of a given patient.

A similar problem can occur when a rule permits two conclusions to be made, each about a different clinical parameter. MYCIN prevents loops in such circumstances by refusing to permit the same rule to occur twice in the current reasoning chain.

3.4 Propagation of Context Tree

The mechanism by which the context tree is customized for a given patient has not yet been discussed. As described in § 3.3.2-1, the consultation program begins simply by creating the patient context and then attempting to execute the goal rule. All additional nodes in the context tree are thus added automatically during the unwinding of MYCIN’s reasoning regarding the PREMISE of the goal rule. This section first explains the data structures used for creating new nodes; then mechanisms for deciding when new nodes should be added are discussed.

3.4.1 DATA STRUCTURES USED FOR SPROUTING BRANCHES

Section 3.2.2-1 was devoted to an explanation of the context tree. At that time, I described the different kinds of contexts and explained that each node in the tree is an instantiation of the appropri-
ate context-type. Each context-type is characterized by the following properties:

- **PROMPT1**: a sentence used to ask the user whether the first node of this type should be added to the context tree; expects a "yes-no" answer.
- **PROMPT2**: a sentence used to ask the user whether subsequent nodes of this type should be added to the context tree.
- **PROMPT3**: replaces PROMPT1 when it is used; this is a message to be printed out if MYCIN assumes that there is at least one node of this type in the tree.
- **PROPTYPE**: indicates the category of clinical parameters (see § 3.2.3-2) that may be used to characterize a context of this type.
- **SUBJECT**: indicates the categories of rules that may be applied to a context of this type.
- **SYN**: indicates a conversational synonym for referring to a context of this type; MYCIN uses SYN when filling in the asterisk of PROMPT properties for clinical parameters.
- **TRANS**: used for English translations of rules referencing this type of context.
- **TYPE**: indicates what kind of internal name to give a context of this type.
- **MAINPROPS**: lists the clinical parameters, if any, that are to be automatically traced (by FINDOUT) whenever a context of this type is created.
- **ASSOCWITH**: gives the context-type of nodes in the tree immediately above contexts of this type.

Two sample context-types are shown in Figure 3-10. The following observations may help clarify the information given in that figure:

1. **PRIORCULS**: Whenever a prior culture is created, it is given the name CULTURE-# (see TYPE), where # is the next unassigned culture number. The values of SITE and WHENCUL are immediately traced using the FINDOUT mechanism (see MAINPROPS). The culture node is put in the context tree below a node of type PERSON (see ASSOCWITH) and the new context may be characterized by clinical parameters of the type PROP-CUL (see PROPTYPE). The prior culture may be the context for either PRCULRULES or CULRULES (see SUBJECT) and is translated, in questions to the user, as “this <site> culture” (see SYN) where “<site>” is replaced by the site of the culture if it is known. The use of PROMPT1
and PROMPT2 is demonstrated in the sample consultation at the end of
Chapter 1.

(2) CURORG: Since there is a PROMPT3 rather than a PROMPT1, MYCIN
prints out the PROMPT3 message and assumes (without asking) that there
is at least one CURORG for each CURCUL (see ASSOCWITH); the other
CURORG properties correspond to those described above for
PRIORCULS.

Whenever MYCIN creates a new context using these models, it
prints out the name of the new node in the tree, e.g.:
Thus the user is familiar with MYCIN's internal names for the cultures, organisms, and drugs under discussion. The node names may then be used in MYCIN's questions at times when there may be ambiguity regarding which node is the current context, e.g.:

Is the patient's illness with the staphylococcus (ORGANISM-2) a hospital-acquired infection?

It should also be noted that when PROMPT1 or PROMPT2 is used to ask the physician a question, he need not be aware that the situation is different from that occurring when FINDOUT asks questions. All the user options described in § 3.3.2-2 operate in the normal fashion.

Finally, the MAINPROPS property requires brief explanation. The claim was previously made that clinical parameters are traced and their values requested by FINDOUT only when they are needed for evaluation of a rule that has been invoked. Yet, we must now acknowledge that certain LABDATA parameters are automatically traced whenever a node for the context tree is created. The reason for this departure is our attempt to keep the program acceptable to physicians. Since the order of rules on UPDATED-BY lists is arbitrary, the order in which questions are asked is somewhat arbitrary as well. We have found that physicians are annoyed if the "basic" questions are not asked first, as soon as the context is created. The MAINPROPS convention forces certain standard questions early in the characterization of a node in the context tree. Parameters not on the MAINPROPS list are then traced in an arbitrary order that depends upon the order in which rules are invoked.

The MAINPROPS convention may be compared to the antecedent theorems of PLANNER that were discussed in § 3.3.1. Although I argued then against a system based solely upon antecedent theorems, I did acknowledge that they were powerful for certain purposes when they did not clutter memory with unnecessary information. Since the parameters on MAINPROPS lists are important pieces of information that would uniformly be traced by FINDOUT anyway, the convention we have implemented forces a standardized ordering of the "basic" questions without generating useless information.
3.4.2 EXPLICIT MECHANISMS FOR BRANCHING

There are two situations under which MYCIN attempts to add new nodes to the context tree. The simpler case occurs when rules explicitly reference contexts that have not yet been created. Suppose, for example, MYCIN is trying to determine the identity of a current organism and therefore invokes the following CURORGRULE:

RULE004

IF: 1) THE IDENTITY OF THE ORGANISM IS NOT KNOWN WITH CERTAINTY, AND
   2) THIS CURRENT ORGANISM AND PRIOR ORGANISMS OF THE PATIENT AGREE WITH RESPECT TO THE FOLLOWING PROPERTIES: GRAM MORPH

THEN: THERE IS WEAKLY SUGGESTIVE EVIDENCE THAT EACH OF THEM IS A PRIOR ORGANISM WITH THE SAME IDENTITY AS THIS CURRENT ORGANISM

The second condition in the PREMISE of this rule references other nodes in the tree, namely nodes of the type PRIORORG. If no such nodes exist, the MONITOR asks FINDOUT to trace PRIORORG in the normal fashion. The difference is that PRIORORG is not a clinical parameter but a context-type. FINDOUT therefore uses PROMPT1 of PRIORORG to ask the user if there is at least one organism. If so, an instantiation of PRIORORG is added to the context tree and its MAINPROPS are traced. PROMPT2 is then used to see if there are any additional prior organisms and the procedure continues until the user indicates there are no more PRIORORGs that merit discussion. Finally, FINDOUT returns the list of prior organisms to the MONITOR so that the second condition in RULE004 can be evaluated.

3.4.3 IMPLICIT MECHANISMS FOR BRANCHING

There are two kinds of implicit branching mechanisms. One of these is closely associated with the example of the previous section. As shown in Figure 3-1, a prior organism is associated with a prior culture. But the explicit reference to prior organisms in RULE004
made no mention of prior cultures. Thus, if FINDOUT tries to create a PRIORORGs in response to an explicit reference but finds there are no PRIORCULS, the program knows there is an implied need to ask the user about prior cultures before asking about prior organisms. Since PRIORCULS are associated with the patient himself, and since the patient node already exists in the context tree, only one level of implicit branching is required in the evaluation of RULE004.

The other kind of implicit branching occurs when the MONITOR attempts to evaluate a rule for which no appropriate context exists. For example, the first rule invoked in an effort to execute the goal rule is a CURORGRULE (see RULE090, Figure 3-9). Since no current organism has been created at the time the MONITOR is passed this CURORGRULE, MYCIN automatically attempts to create the appropriate nodes and then to apply the invoked rule to each.

3.5 Selection of Therapy

The discussion in § 3.3 and 3.4 concentrated on the PREMISE of MYCIN's principal goal rule (RULE092, § 3.3.2-1). This section explains what happens when the PREMISE is found to be true and the two-step ACTION clause is executed.

Unlike other rules in the system, the goal rule does not lead to a conclusion (§ 3.3.3-2) but instead instigates actions. The functions in the ACTION of the goal rule thus correspond to the <actfunc> class that was introduced in the BNF description of § 3.2.1-2. The first of these functions causes a list of potential therapies to be created. The second allows the best drug or drugs to be selected from the list of possibilities.

3.5.1 CREATION OF POTENTIAL THERAPY LIST

There is a class of decision rules, the THERULES (§ 3.2.2-2), that are never invoked by MYCIN's regular control structure because they do not occur on the UPDATED-BY list of any clinical parameters. These rules contain sensitivity information for the various organisms known to the system. For example:

RULE088

IF: THE IDENTITY OF THE ORGANISM IS PSEUDOMONAS

142
Consultation System

THEN: I RECOMMEND THERAPY CHOSEN FROM AMONG THE FOLLOWING DRUGS:
1 - COLISTIN (.98)
2 - POLYMYXIN (.96)
3 - GENTAMICIN (.96)
4 - CARBENICILLIN (.65)
5 - SULFISOXAZOLE (.64)

The numbers associated with each drug are the probabilities that a pseudomonas isolated at Stanford Hospital will be sensitive (in vitro) to the indicated drug. The sensitivity data were acquired from Stanford's microbiology laboratory (and could easily be adjusted to reflect changing resistance patterns at Stanford or the data for some other hospital desiring a version of MYCIN with local sensitivity information). Rules such as the one shown here provide the basis for creating a list of potential therapies. There is one such rule for every kind of organism known to the system.

MYCIN selects drugs only on the basis of the identity of offending organisms. Thus, the program's first task is to decide, for each current organism deemed to be significant, which hypotheses regarding the organism’s identity (IDENT) are sufficiently likely so that they must be considered in choosing therapy. MYCIN uses the CF's of the various hypotheses in order to select the most likely identities (see § 4.7). Each identity is then given an “item number” (see below) and the process is repeated for each significant current organism. The “Set of Indications” for therapy is then printed out, e.g.:

My therapy recommendation will be based on the following possible identities of the organism(s) that seem to be significant:
<Item 1> The identity of ORGANISM-1 may be
      STREPTOCOCCUS-GROUP-D
<Item 2> The identity of ORGANISM-1 may be
      STREPTOCOCCUS-ALPHA
<Item 3> The identity of ORGANISM-2 is PSEUDOMONAS

Each item in this list of therapy indications corresponds to one of the THERULES. For example, Item 3 corresponds to RULE088 above. Thus, MYCIN retrieves the list of potential therapies for each indication from the associated THERULE. The default (in vitro) statistical data are also retrieved. MYCIN then replaces the default sensitivity data with real data about those of the patient's organisms,
if any, for which actual sensitivity information is available from the laboratory. Furthermore, if MYCIN has inferred sensitivity information from the *in vivo* performance of a drug that has already been administered to the patient, this information also replaces the default sensitivity data. Thus, the “compiled list of potential therapies” is actually several lists, one for each item in the Set of Indications. Each list contains the names of drugs and, in addition, the associated number representing MYCIN’s judgment regarding the organism’s sensitivity to each of the drugs.

3.5.2 SELECTING PREFERRED DRUG FROM LIST

When MYCIN recommends therapy it tries to suggest a drug for each of the items in the Set of Indications. Thus, the problem reduces to one of selecting the best drug from the therapy list associated with each item. Clearly the probability that an organism will be sensitive to a drug is an important factor in this selection process. However, there are several other considerations. MYCIN’s strategy is to select the best drug on the basis of sensitivity information but then to consider contraindications for that drug. Only if a drug survives this second screening step is it actually recommended. Furthermore, MYCIN also looks for ways to minimize the number of drugs recommended and thus seeks therapies that cover for more than one of the items in the Set of Indications. The selection/screening process is described in the following two subsections.

3.5.2-1 Choosing Apparent First Choice Drug

The procedure used for selecting the apparent first choice drug is a complex algorithm that is somewhat arbitrary and is currently under revision. In this section, I shall therefore describe the procedure in somewhat general terms since the actual LISP functions and data structures are not particularly enlightening.

There are three initial considerations used in selecting the best therapy for a given item:

(1) the probability that the organism is sensitive to the drug;
(2) whether the drug is already being administered;
Consultation System

(3) the relative efficacy of drugs that are otherwise equally supported by the criteria in (1) and (2).

As is the case with human consultants, MYCIN does not insist on a change in therapy if the physician has already begun a drug that may work, even if that drug would not otherwise be MYCIN's first choice. Drugs with sensitivity numbers within .05 of one another are considered to be almost identical on the basis of criterion (1). Thus RULE088 above, for example, indicates no clear preference among colistin, polymyxin, and gentamicin for pseudomonas infections (if default sensitivity information from the rule is used). However, our collaborating experts have ranked the relative efficacy of antimicrobials on a scale from 1 to 10. The number reflects such factors as whether the drug is bacteriostatic or bacteriocidal, or its tendency to cause allergic sensitization. Since gentamicin has a higher relative efficacy than either colistin or polymyxin, it is the first drug considered for pseudomonas infections (unless known sensitivity information or previous drug experience indicates that an alternate choice is preferable).

Once MYCIN has selected the apparent best drug for each item in the Set of Indications, it checks to see if one of the drugs is also useful for one or more of the other indications. For example, if the first choice drug for item 1 is the second choice drug for item 2, and if the second choice drug for item 2 is almost as strongly supported as the first choice drug, item 1's first choice drug also becomes item 2's first choice drug. This strategy permits MYCIN to attempt to minimize the number of drugs to be recommended.

A similar strategy is used to avoid giving two drugs of the same drug class. For example, MYCIN knows that if the first choice for one item is penicillin and the first choice for another is ampicillin, then the ampicillin may be given for both indications.

In the ideal case, MYCIN will find a single drug that effectively covers for all the items in the set of indications. But even if each item remains associated with a different drug, a screening stage to look for contraindications is required. This rule-based process is described in the next subsection. It should be stressed, however, that the manipulation of drug lists described above is algorithmic, i.e., it is coded in LISP functions that are called from the ACTION clause of the goal rule. There is considerable "knowledge" in this process. Since rule-
MYCIN

based knowledge provides the foundation of MYCIN's ability to explain its decisions, it would be desirable eventually to remove this therapy selection method from functions and place it in decision rules. I will return to this point in § 3.7.

3.5.2-2 Rule-based Screening for Contraindications

Unlike the complex list manipulations described in the previous subsection, criteria for ruling out drugs under consideration may be effectively placed in rules. The rules in MYCIN for this purpose are termed ORDERRULES. The advantages to placing this knowledge in rules are the ones I discussed in Chapter 2, i.e., modularity, ease of modification, and facilitation of explanation and other question-answering. A sample rule of this type is:

RULE055

IF: 1) THE THERAPY UNDER CONSIDERATION IS TETRACYCLINE, AND 2) THE AGE (IN YEARS) OF THE PATIENT IS LESS THAN 13
THEN: THERE IS STRONGLY SUGGESTIVE EVIDENCE (.8) THAT TETRACYCLINE IS NOT AN APPROPRIATE THERAPY FOR USE AGAINST THE ORGANISM

In order to use MONITOR and FINDOUT with such rules, we must construct appropriate nodes in the context tree and must be able to characterize them with clinical parameters. The context-type used for this purpose is termed POSSTHER (§ 3.2.2-1) and the parameters are classified as PROP-THER (§ 3.2.3-2). Thus, when MYCIN has selected the apparent best drugs for the items in the Set of Indications, it creates a context corresponding to each of these drugs. POSSTHER contexts occur below CURORGS in the context tree. FINDOUT is then called to trace the relevant clinical parameter that collects contraindication information (i.e., this becomes a new goal statement) and the normal recursive mechanism through the MONITOR insures that the proper ORDERRULES are invoked.

ORDERRULES allow a great deal of drug-specific knowledge to be stored. For example, RULE055 above insures that tetracycline is ruled out in youngsters who still have developing bone and teeth.
Consultation System

Similar rules tell MYCIN never to given streptomycin or carbenicillin alone, not to give sulfonamides except in urinary tract infections, and not to give cephalothin, clindamycin, lincomycin, vancomycin, cefazolin, or erythromycin if the patient has meningitis. Other ORDERRULES allow MYCIN to consider the patient’s drug allergies, dosage modifications, or ecological considerations (e.g., save gentamicin for pseudomonas, serratia, and hafnia unless the patient is so sick that you cannot risk using a different aminoglycoside while awaiting lab sensitivity data). Finally, there are rules that suggest appropriate combination therapies (e.g., add carbenicillin to gentamicin for known pseudomonas infections). In considering such rules MYCIN often is forced to ask questions that never arose during the initial portion of the consultation. Thus the physician is asked additional questions during the period after MYCIN has displayed the items in the Set of Indications but before any therapy is actually recommended.

After the presumed first-choice drugs have been exposed to the ORDERRULE screening process, MYCIN checks to see whether any of the drugs is now contraindicated. If so, the process described in §3.5.2-1 is repeated. New first-choice drugs are then subjected to the ORDERRULES as I have described above. The process continues until all the first-choice drugs are found to have been instantiated already as POSTHERS. These then become the system’s recommendations. Note that this strategy may result in the recommendation of drugs that are only mildly contraindicated so long as they are otherwise strongly favored. The therapy recommendation itself takes the following form:

My preferred therapy recommendation is as follows:
In order to cover for Items <1> <2> <3>:
Give the following in combination:
1. PENICILLIN
   Dose: 285,000 UNITS/KG/DAY - IV
2. GENTAMICIN
   Dose: 1.7 MG/KG Q8H - IV OR IM
   Comments: MODIFY DOSE IN RENAL FAILURE

The user may also ask for second, third, and subsequent therapy recommendations until MYCIN is able to suggest no reasonable alternatives. The mechanism for these iterations is merely a repeat of
MYCIN

the processes described above but with recommended drugs removed from consideration.

3.6 Mechanisms for Storage of Patient Data

3.6.1 CHANGING ANSWERS TO QUESTIONS

If a physician decides he wants to change his response to a question that he has already answered, MYCIN must do more than merely redisplay the prompt, accept the user's new answer, and make the appropriate change to the value of the clinical parameter in question. The question was originally asked because the PREMISE of a decision rule referenced the clinical parameter. Thus, his original response affected the evaluation of at least one rule, and subsequent pathways in the reasoning network may have been affected as well. It is therefore necessary for MYCIN somehow to return to the state it was in at the time the question was originally asked. Its subsequent actions can then be determined by the corrected user response.

Reversing all decisions made since a question was asked is a complex problem, however. The most difficult task is to determine what portions of a parameter's cumulative CF preceded or followed the question requiring alteration. In fact, the extra data structures needed to permit this kind of backing up are so large and complicated, and would be used so seldom, that it seems preferable simply to restart the consultation from the beginning when the user wants to change one of his answers.

Restarting is of course also less than optimal, particularly if it requires that the physician reenter the answers to questions that were correct the first time around. Our desire to make the program acceptable to physicians required that we devise some mechanism for changing answers, but restarting from scratch also had obvious drawbacks regarding user acceptance of the system. We therefore needed a mechanism for restarting MYCIN's reasoning process but avoiding questions that had already been answered correctly. When FINDOUT asks questions it therefore uses the following algorithm:

[1] before asking the question, check to see if the answer is already stored (in the Patient Data Table, see [3]); if the answer is there, use that value rather than asking the user; otherwise go to [2].

148
Consultation System

[2] ask the question using PROMPT or PROMPT1 as usual.
[3] store the user’s response in the Patient Data Table under the appropriate clinical parameter and context.

The Patient Data Table, then, is a growing record of the user’s responses to questions from MYCIN (see Patient Data, Figure 1-1). It is entirely separate from the dynamic data record (§ 3.3.3-1) that is explicitly associated with the nodes in the context tree. Note that the Patient Data Table contains only the text responses of the user; there is no CF information (unless included in the user’s response), nor are there data derived from MYCIN’s rule-based inferences.

The Patient Data Table and the FINDOUT algorithm above make the task of changing answers much simpler. The technique MYCIN uses is the following:

[a] Whenever the user wants to change the answer to a previous question, he enters CHANGE <numbers>, where <numbers> is a list of the questions whose answers need correction (see § 3.3.2-2);
[b] MYCIN looks up the indicated question numbers in its question record (see § 3.3.3-3);
[c] The user’s responses to the indicated questions are removed from the current Patient Data Table;
[d] MYCIN reinitializes the system, erasing the entire context tree, including all associated parameters; however, it leaves the Patient Data Table intact except for the responses deleted in [c];
[e] MYCIN restarts the consultation from the beginning.

This simple mechanism results in a restarting of the Consultation System (Subprogram 1) but does not require that the user enter correct answers a second time. Since the Patient Data Table is saved, step [1] of the FINDOUT algorithm above will find all the user’s responses until the first question requiring alteration is reached. Thus, the first question asked the user after he gives the CHANGE command is, in fact, the earliest of the questions he wants to change. There may be a substantial pause after the CHANGE command while MYCIN reasons through the network to the first question requiring alteration, but a pause is to be preferred over a mechanism requiring reentry of all question answers. The implemented technique is entirely general because answers to questions regarding tree propagation (§ 3.4.1) are also stored in the Patient Data Table.
3.6.2 REMEMBERING PATIENTS FOR FUTURE REFERENCE

When a consultation is complete, the Patient Data Table contains all responses necessary for generating a complete consultation for that patient. It is therefore straightforward to store the Patient Data Table (on disk or tape) so that it may be reloaded in the future. FINDOUT will automatically read responses from the Table, rather than ask the user, so a consultation may be run several times on the basis of only a single interactive session.

There are two reasons for storing Patient Data Tables for future reference. One is their usefulness in evaluating changes to MYCIN’s knowledge base. The other is the resulting ability to re-evaluate patients once new clinical information becomes available.

3.6.2-1 Evaluating New Rules

New rules may have a large effect on the way a given patient case is handled by MYCIN. For example, a single rule may reference a clinical parameter not previously sought or may lead to an entirely new chain in the reasoning network. It is therefore useful to reload Patient Data Tables and run a new version of MYCIN on old patient cases. A few new questions may be asked (because their responses are not stored in the Patient Data Table). Conclusions regarding organism identities may then be observed, as may the program’s therapeutic recommendations. Any changes from the decisions reached during the original run (i.e., when the Patient Data Table was created) must be explained. When a new version of MYCIN evaluates several old Patient Data Tables in this manner, aberrant side effects of new rules may be found. Thus stored patient cases provide a useful mechanism for screening new rules before they become an integral part of MYCIN’s knowledge base.

3.6.2-2 Re-evaluating Patient Cases

The second use for stored Patient Data Tables is the re-evaluation of a patient once additional laboratory or clinical information becomes available. If a user answers several questions with UNKNOWN during the initial consultation session, MYCIN’s advice will, of course, be based on less than complete information. After storing the
Consultation System

Patient Data Table, however, the physician may return for another consultation in a day or so once he has more specific information. MYCIN can use the previous Patient Data Table for responses to questions whose answers are still up-to-date. The user therefore needs to answer only those questions that reference new information. A mechanism for the physician to indicate directly what new data are available has not yet been automated, however.

A related capability to be implemented before MYCIN becomes available in the clinical setting is a SAVE command. If a physician must leave the computer terminal midway through a consultation, this option will save the current Patient Data Table on the disk. When he returns to complete the consultation he will reload the patient record and the session will continue from the point at which he entered the SAVE command.

It should be understood that saving the current Patient Data Table is not the same as saving the current state of MYCIN's reasoning. Thus, as we have stated above, changes to MYCIN's rule corpus may result in different advice from an identical Patient Data Table. Finally, I wish to emphasize our awareness that disk storage of patient information immediately raises questions of data confidentiality. We will attempt to insure appropriate data protection when MYCIN is available in the clinical setting.

3.7 Future Extensions

In this section I summarize some current ideas for improvement of the consultation program described in this chapter. Each of the topics mentioned is the subject of current efforts by one or more of the researchers currently associated with the MYCIN project.

3.7.1 DYNAMIC ORDERING OF RULES

The order in which rules are invoked by the MONITOR is currently controlled solely by their order on the UPDATED-BY property of the clinical parameter being traced. (An exception to this point is the self-referencing rules, see § 3.3.4.) The order of rules on the UPDATED-BY property is also arbitrary, tending to reflect nothing more than the order in which rules were acquired. Since FINDOUT
MYCIN sends all rules on such lists to the MONITOR, and since our certainty factor combining function is commutative (§ 4.6), the order of rules is unimportant.

Some rules are much more useful than others in tracing the value of a clinical parameter. For example, a rule with a six-condition PREMISE that infers the value of a parameter with a low CF requires a great deal of work (as many as six calls to FINDOUT) with very little gain. On the other hand, a rule with a large CF and only one or two PREMISE conditions may easily provide strong evidence regarding the value of the parameter in question. It may therefore be wise for FINDOUT to order the rules in the UPDATED-BY list on the basis of both information content (CF) and the work necessary to evaluate the PREMISE. Then if the first few rules are successfully executed by the MONITOR, the CF associated with one of the values of the clinical parameter may be so large that invocation of subsequent rules will require more computational effort than they are worth. If FINDOUT therefore ignores such rules (i.e., does not bother to pass them to the MONITOR), considerable time savings may result. Furthermore, entire reasoning chains will in some cases be avoided and the number of questions asked the user could accordingly be decreased.

3.7.2 DYNAMIC ORDERING OF CONDITIONS WITHIN RULES

The MONITOR diagram in Figure 3-7 reveals that conditions are evaluated strictly in the order that they occur within the PREMISE of the rule. In fact, I have stressed that the order of conditions is therefore important and that the most commonly referenced clinical parameters should be placed earliest in the PREMISE.

Suppose, however, that in a given consultation the clinical parameter referenced in the fourth condition of a rule has already been traced by FINDOUT because it was referenced in some other rule that the MONITOR has already evaluated. As currently designed, MYCIN checks the first three conditions first, even if the fourth condition is already known to be false. Since the first three conditions may well require calls to FINDOUT, the rule may generate unnecessary questions and expand useless reasoning chains.

The solution to this problem would be to redesign the MONITOR so that it reorders the PREMISE conditions, first evaluating those
Consultation System

that reference clinical parameters which have already been traced by FINDOUT. In this way a rule will not cause new questions nor additions to the reasoning network if any of its conditions are known to be false at the outset.

3.7.3 PRE-SCREENING OF RULES

An alternate approach to the problem described in the preceding section would be for FINDOUT to judge the implications of every parameter it traces. Once the value has been determined by the normal mechanism, FINDOUT could use the LOOKAHEAD list for the clinical parameter in order to identify all rules referencing the parameter in their PREMISE conditions. FINDOUT could then evaluate the relevant conditions and mark the rule as failing if the condition turns out to be false. Then, whenever the MONITOR begins to evaluate rules that are invoked by the normal recursive mechanism, it will check to see if the rule has previously been marked as false by FINDOUT. If so, the rule could be quickly ruled out without needing to consider the problem of re-ordering the PREMISE conditions.

At first glance, the dynamic re-ordering of PREMISE conditions appears to be a better solution than the one I have just described. The problem with rule pre-screening is that it requires consideration of all rules on the parameter's LOOKAHEAD list, some of which may never actually be invoked during the consultation. Thus the disadvantages are similar to those that can accompany the PLANNER antecedent theorems that were previously described (§ 3.3.1).

3.7.4 PLACING ALL KNOWLEDGE IN RULES

Although most of MYCIN's knowledge is placed in decision rules, I have pointed out several examples of knowledge that is not rule-based. The simple lists and knowledge tables of § 3.2.6 may perhaps be justified on the basis of efficiency arguments, especially since those knowledge structures may be directly accessed and utilized by rules.

However, the algorithmic mechanisms for therapy selection that were described in § 3.5 are somewhat more bothersome. Although we have managed to put many drug-related, decision criteria in the
ORDERRULES, the mechanisms for creating the potential therapy lists and for choosing the apparent first choice drug are programmed explicitly in a series of relatively complex LISP functions. Since MYCIN's ability to explain itself is based upon rule-retrieval (Chapter 5), the system cannot give good descriptions of these drug selection procedures. It is therefore desirable to place more of the drug selection knowledge in rules.

Such efforts should provide a useful basis for evaluating the power of our rule-based formalism. If the goal-oriented control structure we have developed is truly general, one would hope that algorithmic approaches to the construction and ordering of lists could also be placed in decision rule format. We therefore intend to experiment with ways of incorporating the remainder of MYCIN's knowledge into decision rules that are invoked by the standard MONITOR/FINDOUT process.

3.7.5 NEED FOR CONTEXT GRAPH

The context tree used by MYCIN is the source of one of the system's primary problems in attempting to simulate the consultation process. As was pointed out in § 3.2.2-1, every node in the context tree leads to the uppermost patient node by a single pathway. In reality, however, drugs, patients, organisms, and cultures are not interrelated in this highly structured fashion. For example, drugs are often given to cover for more than one organism. The context tree does not permit a single CURDRUG or PRIORDRUG to be associated with more than a single organism. What we need, therefore, is a network of contexts in the form of a graph rather than a pure tree. The current reasons why MYCIN needs a tree structured context network are explained in § 3.2.2. We have come to recognize that a context graph capability is an important extension of the current system, however, and this will be the subject of future design modifications. When implemented, for example, it will permit a physician to discuss a prior drug only once even though it may have been given to cover for several prior organisms.

3.8 Advantages of MYCIN Approach

There are four principal advantages of the MYCIN approach that have contributed to the system's current level of success. Each of
these distinguishes MYCIN from the medical decision making programs described in § 1.3. They also reflect MYCIN’s debt to previous work in the AI field.

3.8.1 MODULARITY OF KNOWLEDGE

As discussed in Chapter 2, one of the major design considerations during the development of MYCIN has been the isolation of pieces of knowledge as discrete facts. MYCIN’s decision rules achieve this goal. Since each rule represents a discrete packet of knowledge, the integration of new information into the system is simplified. Furthermore, the rules can serve as the basis for MYCIN’s explanation and question-answering capabilities (Chapter 5).

Modularity of knowledge is seldom found in diagnostic programs. Some statisticians would argue, in fact, that the interrelationships of observations are so complex that a formal Bayesian approach is the only reasonable way to guarantee good predictions. As I argue in Chapter 4, however, the statistician’s stance is greatly weakened when the knowledge is primarily judgmental and it defies statistical formulation. By accepting the inexact nature of many medical decisions, and by acknowledging that the quantification scheme accompanying our rules is only an approximation technique, we are left free to isolate our knowledge statements and to reap the associated benefits provided by that representation scheme. In fact, almost all of those capabilities that make MYCIN truly innovative may be directly attributed to the program’s rule-based representation of knowledge.

3.8.2 DYNAMIC REASONING CHAIN

It is reasonable to ask why MYCIN does not create an explicit decision tree from its rules, code them for maximal efficiency, and then rely upon conventional techniques for decision analysis based upon progression through a branching tree. It must be remembered, however, that the reasoning network for MYCIN is goal-oriented (Figure 3-9). Conventional decision trees start at the top node and follow a path through the tree based on decisions reached at each subsequent node. When a terminal node in the tree is finally reached, that is the diagnosis. MYCIN’s terminal nodes, on the other hand, correspond to starting points in the accumulation of data (i.e., ASK1
MYCIN

or ASK2 nodes, Figure 3-9). MYCIN’s task is to determine which of these terminal nodes to use in an effort to reach the top of the tree. Thus, the form of MYCIN’s reasoning network is distinctive from a conventional decision tree in that the top node represents the goal for MYCIN rather than the starting point.

Although MYCIN’s rules do not naturally form a conventional decision tree, it is possible that a researcher with experience constructing decision trees could, in time, convert MYCIN’s knowledge base into a traditional tree-shaped format. This has not seemed to be a particularly natural approach, however. There are three principal factors that would complicate any such attempt:

1. Although decision theory has provided mechanisms for incorporating probabilistic knowledge into decision trees, there is no obvious mechanism for combining MYCIN’s certainty factors with a branching network;
2. MYCIN’s current control structure depends upon a dynamic set of contexts and the ability to use rules more than once; this suggests that a decision tree using MYCIN’s knowledge would need to have mechanisms for reusing certain portions, perhaps by defining decision tree “macros”;
3. MYCIN’s reasoning network is actually not tree-shaped; this complexity was not shown in the sample network of Figure 3-9, but since MYCIN’s rules often form reasoning loops (§ 3.3.5) and since a single observation often affects several of the ascending branches in the network, a graph structure would actually provide a more accurate representation of MYCIN’s reasoning network.

It has also been suggested that, even if we do not convert MYCIN’s reasoning network to a conventional decision tree, we could at least explicitly “compile” it. It should be noted, however, that since MYCIN works backwards from the goal-rule, there is no disadvantage to creating a dynamic reasoning chain as it proceeds. The total network that could be created from MYCIN’s rules is so vast that it appears preferable simply to create the portion of the network that is appropriate for the patient under consideration. An explicit network would not avoid the need for MYCIN to work backwards from the topmost goal node. Furthermore, it would introduce the obvious disadvantage that newly acquired rules could not be automatically incorporated into MYCIN’s reasoning as they are by the current dynamic control structure.
3.8.3 DOMAIN-INDEPENDENT CONTROL STRUCTURE

Except for the functions described in § 3.5, most of MYCIN’s functions are domain-independent. In particular, the entire MONITOR/FINDOUT mechanism contains no explicit knowledge of the problem domain for which it has been designed. It is therefore tempting to consider writing new rules for additional medical (or nonmedical) problem areas and to see whether the MYCIN formalism will allow valid consultations in those areas as well. Of course, new clinical parameters and their associated properties would also have to be created, but the resulting knowledge structures are designed to be capable of forming the basis both for consultation sessions using Subprogram 1 and for question-answering sessions using Subprogram 2 (Chapter 5).

Use of the MYCIN approach for another problem area has not yet been attempted, however, and it would therefore be premature to claim that MYCIN’s approach can indeed be generalized for other domains. One reason that we have not attempted to apply the approach elsewhere is our recognition, based on experience to date, that the formulation of new decision rules is no straightforward matter, at least for medical applications. Physicians have not in general structured their own decision processes, and a clinical expert who consistently makes excellent recommendations may have great difficulty describing the steps in reasoning that he uses to make his decisions. Thus, although we are hopeful that the MYCIN formalism can be adapted to another problem area with minimal modification, such efforts would be distracting at a time when our principal concern is the expansion of MYCIN’s clinical expertise regarding antimicrobial therapy.

3.8.4 REASONING WITH JUDGMENTAL KNOWLEDGE

The primary advantage of the MYCIN approach, however, is its ability to model medical reasoning that is based upon neither diagnostic algorithms, physiological models, nor statistical analysis. In fact, MYCIN’s principal contribution to the field of computer-based medical decision making may well be its reasoning model that uses the informal judgmental knowledge of physician experts. Other programs have attempted to use “estimates” provided by expert physi-
MYCIN

cians [Leaper, 1972] but have been limited by efforts to couch these estimates in probabilistic terms. MYCIN not only provides an intuitively pleasing mechanism for recording (decision rules) and interpreting (certainty factors) these numbers, but it provides a flexible control structure and interactive capabilities that encourage the physician to accept the program as the useful and cooperative clinical tool that it is designed to be.
Chapter 4

Model of Inexact Reasoning in Medicine†

4.1 Introduction

Efforts to develop techniques for modeling clinical decision making have had a dual motivation. Not only has their potential clinical significance been apparent, but the design of such programs has required an analytical approach to medical reasoning that has in turn led to a distillation of decision criteria that in some cases had never been explicitly stated before. It is a fascinating and educational process for experts to reflect on the reasoning steps that they have always used when providing clinical consultations.

As discussed in § 1.3, several programs have successfully modeled the diagnostic process [Gorry, 1968a, 1973; Warner, 1964]. Each of these examples has relied upon statistical decision theory as reflected in the use of Bayes’ Theorem for manipulation of conditional probabilities. Use of the theorem, however, requires either large amounts of valid background data or numerous approximations and assumptions. The success of Gorry and Barnett’s early work [Gorry, 1968a], and a similar study by Warner et al. using the same data [Warner, 1964], depended to a large extent upon the availability of good data regarding several individuals with congenital heart disease. Gorry et al. [Gorry, 1973b] have had similar access to data relating the symptoms and signs of acute renal failure to the various potential etiologies.

†Much of the material in this chapter has appeared in an article in *Mathematical Biosciences* [Shortliffe, 1975a]. That paper was co-authored with Dr. Bruce Buchanan who contributed substantially to the development of the model.
MYCIN

Although conditional probability provides useful results in areas of medical decision making such as those I have mentioned, vast portions of medical experience suffer from so little data and so much imperfect knowledge that a rigorous probabilistic analysis, the ideal standard by which to judge the rationality of a physician’s decisions, is not possible. It is nevertheless instructive to examine models for the less formal aspects of decision making. Physicians seem to use an ill-defined mechanism for reaching decisions despite a lack of formal knowledge regarding the interrelationships of all the variables that they are considering. This mechanism is often adequate, in well-trained or experienced individuals, to lead to sound conclusions on the basis of a limited set of observations. (Intuition may also lead to unsound conclusions, as noted by Schwartz et al. [Schwartz, 1973].)

These intuitive and inexact aspects of medical reasoning are reflected in an argument expounded by Helmer and Rescher [Helmer, 1960]. They assert that the traditional concept of “exact” versus “inexact” science, with the social sciences accounting for the second class, has relied upon a false distinction usually reflecting the presence or absence of mathematical notation. They point out that only a small portion of natural science can be termed exact—areas such as pure mathematics and subfields of physics in which some of the exactness “has even been put to the ultimate test of formal axiomatization.” In several areas of applied natural science, on the other hand, decisions, predictions, and explanations are only made after exact procedures are mingled with unformalized expertise. Society’s general awareness regarding these observations is reflected in the common references to the “artistic” components in the “science of medicine.”

This chapter examines the nature of such nonprobabilistic and unformalized reasoning processes, considers their relationship to formal probability theory, and proposes a model whereby such incomplete “artistic” knowledge might be quantified. We have developed this model of inexact reasoning in response to MYCIN’s needs; i.e., the goal has been to permit the opinion of experts to become more generally available to nonexperts. The model is, in effect, an approximation to conditional probability. Although conceived with MYCIN’s problem area in mind, it is potentially applicable to any domain in which real world knowledge must be com-
Model of Inexact Reasoning in Medicine

combined with expertise before an informed opinion can be obtained to explain observations or to suggest a course of action.

The presentation begins with a brief discussion of Bayes' Theorem as it has been utilized by other workers in this field. The theorem serves as a focus for discussion of the clinical problems that we would like to solve by using computer models. The potential applicability of the proposed decision model is then introduced in light of MYCIN's rule-based design. Once the problem has been defined in this fashion, the criteria and numerical characteristics of our quantification scheme are proposed. The chapter concludes with a discussion of how the model is being used by MYCIN when it offers opinions to physicians regarding antimicrobial therapy selection.

4.2 Problem Formulation

The medical diagnostic problem can be viewed as the assignment of probabilities to specific diagnoses after analyzing all relevant data. If the sum of the relevant data (or evidence) is represented by $E$, and $D_i$ is the $i$th diagnosis (or "disease") under consideration, then $P(D_i | E)$ is the conditional probability that the patient has disease $i$ in light of the evidence $E$. Diagnostic programs have traditionally sought to find a set of evidence that allows $P(D_i | E)$ to exceed some threshold, say .95, for one of the possible diagnoses. Under these circumstances the second ranked diagnosis is sufficiently less likely (<.05) that the user is content to accept disease $i$ as the diagnosis requiring therapeutic attention. (Several programs have also included utility considerations in their analyses. For example, an unlikely but lethal disease that responds well to treatment may merit therapeutic attention because $P(D_i | E)$ is nonzero, even though very small.)

Bayes' Theorem is useful in these applications because it allows $P(D_i | E)$ to be calculated from the component conditional probabilities:

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

In this representation of the theorem, $D_i$ is one of $n$ disjoint diag-
noses. $P(D_i)$ is simply the *a priori* probability that the patient has disease $i$ before any evidence has been gathered. $P(E|D_i)$ is the probability that a patient will have the complex of symptoms and signs represented by $E$, given that he has disease $D_i$.

I have so far ignored the complex problem of identifying the "relevant" data that should be gathered in order to diagnose the patient's disease. Evidence is actually acquired piece-by-piece, the necessary additional data being identified on the basis of the likely diagnosis at any given time. Diagnostic programs that mimic the process of analyzing evidence incrementally often use a modified version of Bayes' Theorem that is appropriate for sequential diagnosis [Gorry, 1968a]:

Let $E_1$ be the set of all observations to date, and $S_1$ be some new piece of data. Furthermore, let $E$ be the new set of observations once $S_1$ has been added to $E_1$. Then

$$P(D_j|E) = \frac{P(S_1|D_j&E_1)P(D_j|E_1)}{\sum_{j=1}^{n} P(S_1|D_j&E_1)P(D_j|E_1)}$$

The successful programs that use Bayes' Theorem in this form required huge amounts of statistical data, not merely $P(D_j|S_k)$ for each of the pieces of data $S_k$ in $E$, but also the interrelationships of the $S_k$ within each disease $D_j$. For example, although $S_1$ and $S_2$ are independent over all diseases, it may be true that $S_1$ and $S_2$ are closely linked for patients with disease $D_j$. Thus relationships must be known within each $D_j$; overall relationships are not sufficient. The congenital heart disease programs [Gorry, 1968a; Warner, 1964] were able to acquire all the necessary conditional probabilities from a survey of several hundred patients with confirmed diagnoses and thus had nonjudgmental data on which to base their Bayesian analyses.

Edwards has summarized the kinds of problems that can arise when an attempt is made to gather the kinds of data needed for rigorous analysis [W. Edwards, 1972]:

... My friends who are expert about medical records tell me that to attempt to dig out from even the most sophisticated hospital's records the frequency of association between any particular symptom and any particular diagnosis is next
Model of Inexact Reasoning in Medicine

to impossible—and when I raise the question of complexes of symptoms, they stop speaking to me. For another thing, doctors keep telling me that diseases change, that this year’s flu is different from last year’s flu, so that symptom-disease records extending far back in time are of very limited usefulness. Moreover, the observation of symptoms is well-supplied with error, and the diagnosis of diseases is even more so; both kinds of errors will ordinarily be frozen permanently into symptom-disease statistics. Finally, even if diseases didn’t change, doctors would. The usefulness of disease categories is so much a function of available treatments that these categories themselves change as treatments change—a fact hard to incorporate into symptom-disease statistics.

All these arguments against symptom-disease statistics are perhaps somewhat overstated. Where such statistics can be obtained and believed, obviously they should be used. But I argue that usually they cannot be obtained, and even in those instances where they have been obtained, they may not deserve belief.

An alternative to exhaustive data collection is to use the knowledge that an expert has about the disease—partly based upon experience and partly on general principles—to reason about diagnoses. In the case of this judgmental knowledge acquired from experts, the conditional probabilities and their complex interrelationships cannot be acquired in an exhaustive manner. Opinions can be sought and attempts made to quantify them, but the extent to which the resulting numbers can be manipulated as probabilities is not clear. We shall explain this last point more fully as we proceed. First, let us examine some of the reasons that it might be desirable to construct a model that allows us to avoid the inherent problems of explicitly relating the conditional probabilities to one another.

As was pointed out in § 3.2, a conditional probability statement is, in effect, a statement of a decision criterion or rule. For example, the expression $P(D_i | S_k) = X$ can be read as a statement that there is a $100X\%$ chance that a patient observed to have symptom $S_k$ has disease $D_i$. Stated in rule form:

\[
\begin{align*}
\text{IF:} & \quad \text{THE PATIENT HAS SIGN OR SYMPTOM } S_k \\
\text{THEN:} & \quad \text{CONCLUDE THAT HE HAS DISEASE } D_i \text{ WITH PROBABILITY } X
\end{align*}
\]

I shall often refer to statements of conditional probability as decision rules or decision criteria in the diagnostic context. The value of $X$ for such rules may not be obvious (e.g., “$y$ strongly suggests that $z$ is true” is difficult to quantify), but an expert may be able to offer an
MYCIN

estimate of this number based upon clinical experience and general knowledge, even when such numbers are not readily available otherwise.

A large set of such rules obtained from textbooks and experts would clearly contain a large amount of medical knowledge. It is conceivable that a computer program could be designed to consider all such general rules and to generate a final probability of each $D_i$ based upon data regarding a specific patient. Bayes' Theorem would only be appropriate for such a program, however, if values for $P(S_1 \mid D_i)$ and $P(S_1 \mid D_i \& S_2)$ could be obtained. As has been noted, these requirements become unworkable, even if the subjective probabilities of experts are used, in cases where a large number of diagnoses (hypotheses) must be considered. The first would require acquiring the inverse of every rule, and the second requires obtaining explicit statements regarding the interrelationships of all rules in the system.

In short, we would like to devise an approximate method that allows us to compute a value for $P(D_i \mid E)$ solely in terms of $P(D_i \mid S_k)$, where $E$ is the composite of all the observed $S_k$ (see § 4.5 and 4.6). Such a technique will not be exact, but since the conditional probabilities reflect judgmental (and thus highly subjective) knowledge, a rigorous application of Bayes' Theorem will not necessarily produce accurate cumulative probabilities either. Instead we look for ways to handle decision rules as discrete packets of knowledge and for a quantification scheme that permits accumulation of evidence in a manner that adequately reflects the reasoning process of an expert using the same or similar rules.

4.3 Mycin's Rule-Based Approach

As has been discussed, MYCIN's principle task is to determine the likely identity of pathogens in patients with infections and to assist in the selection of a therapeutic regimen appropriate for opposing the organisms under consideration. In Chapter 3, we explained how MYCIN models the consultation process, utilizing judgmental knowledge acquired from experts in conjunction with certain statistical data that are available from the clinical microbiology laboratory and from patient records. MYCIN's decision rules are similar in form to those just introduced in § 4.2.
Model of Inexact Reasoning in Medicine

It is useful to consider the advantages provided by a rule-based system for computer use of judgmental knowledge. It should be emphasized that we see these advantages as being sufficiently strong in certain environments that we have devised an alternative and approximate approach that parallels the results available from using Bayes' Theorem. I do not argue against the use of Bayes' theory in those medical environments in which sufficient data are available to permit adequate use of the theorem.

The advantages of rule-based systems for diagnostic consultations include:

(1) the use of general knowledge (from textbooks or experts) for consideration of a specific patient; even well-indexed books may be difficult for a nonexpert to use when considering a patient whose problem is not quite the same as those of patients discussed in the text;

(2) the use of judgmental knowledge for consideration of very small classes of patients with rare diseases about which good statistical data are not available;

(3) ease of modification; since the rules are not explicitly related to one another and there need be no prestructured decision tree for such a system, rule modifications and the addition of new rules need not require complex considerations regarding interactions with the remainder of the system's knowledge;

(4) facilitated search for potential inconsistencies and contradictions in the knowledge base; criteria stored explicitly in packets such as rules can be searched and compared without major difficulty;

(5) straightforward mechanisms for explaining decisions to a user by identifying and communicating the relevant rules;

(6) an augmented instructional capability; a system user may be educated regarding system knowledge in a selective fashion, i.e., only those portions of the decision process that puzzle him need be examined.

One of MYCIN's rules, which I shall use for illustrative purposes throughout this chapter, is the following:

IF: 1) THE STAIN OF THE ORGANISM IS GRAM POSITIVE, AND

2) THE MORPHOLOGY OF THE ORGANISM IS COCCUS, AND

3) THE GROWTH CONFORMATION OF THE ORGANISM IS CHAINS

THEN: THERE IS SUGGESTIVE EVIDENCE (.7) THAT THE IDENTITY OF THE ORGANISM IS STREPTOCOCCUS
This rule was acquired from an expert in infectious disease therapy and reflects his belief that gram positive cocci growing in chains are apt to be streptococci. When asked to weight his belief in this conclusion, he indicated a 70% belief that the conclusion was valid. In the English language version of the rules, the program uses phrases such as “suggestive evidence” as in the above example. However, the numbers following these terms, indicating degrees of certainty, are all that is used in the model. The English phrases are not given by the expert and then quantified; they are, in effect, “canned-phrases” used only for translating rules into English representations. The prompt used for acquiring the certainty measure from the expert is: “On a scale of 1 to 10, how much certainty do you affix to this conclusion?”

Translating to the notation of conditional probability, the rule above at first seems to say \( P(H_1 | S_1 & S_2 & S_3) = .7 \) where \( H_1 \) is the hypothesis that the organism is a streptococcus, \( S_1 \) the observation that the organism is gram positive, \( S_2 \) that it is a coccus, and \( S_3 \) that it grows in chains. Questioning of the expert gradually reveals, however, that despite the apparent similarity to a statement regarding a conditional probability, the number .7 differs significantly from a probability. The expert may well agree that \( P(H_1 | S_1 & S_2 & S_3) = .7 \), but he becomes uneasy when he attempts to follow the logical conclusion that therefore \( P(\neg H_1 | S_1 & S_2 & S_3) = .3 \). The three observations are evidence (to degree .7) in favor of the conclusion that the organism is a streptococcus and should not be construed as evidence (to degree .3) against streptococcus. I shall refer to this problem as Paradox 1 and return to it later in the exposition after the interpretation of the .7 in the rule above has been introduced.

It may at first seem tempting to conclude that the expert is irrational if he is unwilling to follow the implications of his probabilistic statements to their logical conclusions. Another interpretation, however, is that the numbers he has given should not be construed as probabilities at all, that they are judgmental measures that reflect a level of belief. The nature of such numbers, and the very existence of such concepts, have interested philosophers of science for the last half century. Some of these philosophical issues are briefly discussed in § 4.4. I then proceed to a detailed presentation of the proposed quantitative model. In the last section of this
Model of Inexact Reasoning in Medicine

chapter, I shall show how the model has been implemented for ongoing use by the MYCIN program.

4.4 Theoretical Background

Although probability is a familiar concept defined axiomatically in any introductory statistics book [Parzen, 1960], the P-function has been subjected to a variety of interpretations [Swinburne, 1973; Harré, 1970; Ramsey, 1931; Savage, 1954; deFinetti, 1972; Keynes, 1921; Carnap, 1950]. I shall not describe all of these because, as has been observed, imperfect knowledge and the dependence of decisions on individual judgments make the P-function no longer seem entirely appropriate for modeling many of the decision processes in medical diagnosis.

Carnap [Carnap, 1950] and Hempel [Hempel, 1945] discuss an interpretation of probability known as confirmation. Carnap distinguishes confirmation from the traditional P-function, defining the former as the degree to which an hypothesis is supported by an evidence statement. Thus, it should be noted that the term confirmation does not indicate that an hypothesis is proven but rather that an observation lends credence to it. The measure of support is commonly represented by the notation \( C[h, e] \), i.e., the degree of confirmation of the hypothesis \( h \) based upon the observation \( e \).

Quantifying confirmation and then manipulating the numbers as though they are probabilities quickly leads to apparent inconsistencies or paradoxes [Carnap, 1950; Hempel, 1945; Barker, 1957; Salmon, 1973, 1966]. Carl Hempel [Hempel, 1945] presented his famous Paradox of the Ravens early in his discussion of the logic of confirmation. Let \( h_1 \) be the statement that “All ravens are black” and \( h_2 \) the statement that “All nonblack things are nonravens.” Clearly \( h_1 \) is logically equivalent to \( h_2 \). If one were to draw an analogy with conditional probability, it might at first seem valid, therefore, to assert that \( C[h_1, e] = C[h_2, e] \) for all \( e \). However, it appears counter-intuitive to state that the observation of a green vase supports \( h_1 \) even though the observation does seem to support \( h_2 \). \( C[h, e] \) is therefore different from \( P(h|e) \) for it seems somehow wrong that the observation of a vase could logically support an
assertion about ravens. A re-examination of this paradox in light of our proposed quantification scheme is included as an appendix to this chapter (Appendix 4.A).

In their analyses of confirmation, several authors [Harre, 1970; Carnap, 1950; Hempel, 1945; Barker, 1957; Salmon, 1973, 1966] note that $C[h,e]$ does not equal $1-C[\neg h,e]$, an observation reminiscent of our Paradox 1 from § 4.3. Furthermore, they recognize the need for an independently introduced disconfirmation function because, as Harre puts it [Harre, 1970], "to confirm something to ever so slight a degree is not to disconfirm it at all, since the favourable evidence for some hypothesis gives no support whatever to the contrary supposition in many cases."

The inadequacies of probability in the analysis of real-world problems have led to a variety of alternate approaches. These include the theory of "fuzzy sets" [Zadeh, 1965; Goguen, 1968], the theory of "choice" [Tversky, 1972; Luce, 1965], and the logic of "surprise" [Shackle, 1952, 1955]. However, the theory of confirmation seems to parallel more closely the kind of decision task involved in medical diagnosis. We have therefore sought to develop a quantification scheme that reflects the observations of philosophers who have dealt with the logic of confirmation. However, the scheme I propose meets desiderata derived intuitively from the problem at hand rather than from a formal list of acceptability criteria. Such criteria are proposed by several authors such as Carnap [Carnap, 1950], Swinburne [Swinburne, 1970], Salmon [Salmon, 1966], and Törnebohm, [Törnebohm, 1966]. Although our model was not developed with any such list of criteria as guidance, I shall show (§ 4.5 and 4.6) that the technique we propose satisfies Törnebohm's criteria in light of the approximation mechanisms that we have introduced for the combination of incrementally acquired evidence.

4.5 Proposed Model of Evidential Strength

This section introduces our quantification scheme for modeling inexact medical reasoning. It begins by defining the notation that we use and by describing the terminology. A formal definition of the quantification function will then be presented. The remainder of the section discusses the characteristics of the defined functions. It closes
with consideration of the model when it is compared to Törnebohm’s criteria for acceptability of a quantification technique regarding evidential strength [Törnebohm, 1966].

Although the proposed model has several similarities to a confirmation function such as those mentioned above, I shall introduce new terms for the measurement of evidential strength. This convention will allow me to clarify from the outset that I seek only to devise a system that captures enough of the flavor of confirmation theory that it can be used for accomplishing our computer-based task. We have chosen “Belief” and “Disbelief” as our units of measurement, but these terms should not be confused with their formalisms from epistemology. The need for two measures was introduced above in our discussion of a disconfirmation measure as an adjunct to a measure for degree of confirmation. The notation will be as follows:

(1) \[ MB[h,e] = X \] means “The measure of increased Belief in the hypothesis \( h \), based on the evidence \( e \), is \( X \)”

(2) \[ MD[h,e] = Y \] means “The measure of increased Disbelief in the hypothesis \( h \), based on the evidence \( e \), is \( Y \)”

The evidence \( e \) need not be an observed event, but may be a hypothesis (itself subject to confirmation). Thus, I may write \( MB[h_1,h_2] \) to indicate the measure of increased Belief in the hypothesis \( h_1 \) given that the hypothesis \( h_2 \) is true. Similarly \( MD[h_1,h_2] \) is the measure of increased Disbelief in hypothesis \( h_1 \) if hypothesis \( h_2 \) is true.

To illustrate in the context of the sample rule from MYCIN, consider \( e = \) “The organism is a gram positive coccus growing in chains” and \( h = \) “The organism is a streptococcus.” Then, \( MB[h,e] = .7 \) according to the sample rule given us by the expert. The relationship of the number .7 to probability will be explained as I proceed. For now let me simply state that the number .7 reflects the extent to which the expert’s Belief that \( h \) is true is increased by the knowledge that \( e \) is true. On the other hand, \( MD[h,e] = 0 \) for this example, i.e., the expert has no reason to increase his Disbelief in \( h \) on the basis of \( e \).

In accordance with subjective probability theory, it may be argued that the expert’s personal probability \( P(h) \) reflects his Belief in \( h \) at any given time. Thus \( 1-P(h) \) can be viewed as an
estimate of the expert’s Disbelief regarding the truth of \( h \). If \( P(h|e) \) is greater than \( P(h) \), the observation of \( e \) increases the expert’s Belief in \( h \) while decreasing his Disbelief regarding the truth of \( h \). In fact, the proportionate decrease in Disbelief is given by the ratio:

\[
\frac{P(h|e) - P(h)}{1 - P(h)}
\]

This ratio is called the measure of increased Belief in \( h \) resulting from the observation of \( e \), i.e., \( MB[h,e] \).

Suppose, on the other hand, that \( P(h|e) \) were less than \( P(h) \). Then the observation of \( e \) would decrease the expert’s Belief in \( h \) while increasing his Disbelief regarding the truth of \( h \). The proportionate decrease in Belief is in this case given by the ratio:

\[
\frac{P(h) - P(h|e)}{P(h)}
\]

We call this ratio the measure of increased Disbelief in \( h \) resulting from the observation of \( e \), i.e., \( MD[h,e] \). Törnebohm suggests a similar measure of evidential strength [Törnebohm, 1966], but uses \( C(H) \) instead of \( P(H) \), where \( C(H) \) is the amount of information contained in \( H \).

To summarize these results in words, we consider the measure of increased Belief, \( MB[h,e] \), to be the proportionate decrease in Disbelief regarding the hypothesis \( h \) that results from the observation \( e \). Similarly, the measure of increased Disbelief, \( MD[h,e] \), is the proportionate decrease in Belief regarding the hypothesis \( h \) that results from the observation \( e \), where Belief is estimated by \( P(h) \) at any given time and Disbelief is estimated by \( 1-P(h) \). These definitions correspond closely to the intuitive concepts of confirmation and disconfirmation that we have discussed above. Note that since one piece of evidence cannot both favor and disfavor a single hypothesis, when \( MB[h,e]>0 \), \( MD[h,e]=0 \) and when \( MD[h,e]>0 \), \( MB[h,e]=0 \). Furthermore, when \( P(h|e)=P(h) \) the evidence is independent of the hypothesis (neither confirms nor disconfirms) and \( MB[h,e]=MD[h,e]=0 \).

The above definitions may now be specified formally in terms of conditional and \( a \) priori probabilities:
Model of Inexact Reasoning in Medicine

\[
\begin{align*}
\text{MB}[h,e] &= \begin{cases} 
1 & \text{if } P(h) = 1, \\
\frac{\max[P(h|e), P(h)] - P(h)}{\max[1,0] - P(h)} & \text{otherwise,}
\end{cases} \\
\text{MD}[h,e] &= \begin{cases} 
1 & \text{if } P(h) = 0, \\
\frac{\min[P(h|e), P(h)] - P(h)}{\min[1,0] - P(h)} & \text{otherwise.}
\end{cases}
\end{align*}
\]

Note that here \(P(h)\) is used to denote \textit{a priori} probabilities. More correctly they might be written as \(P(h|0)\), i.e., the probability of \(h\) on no evidence. Examination of these expressions will reveal that they are identical to the definitions introduced above. The formal definition is introduced, however, to demonstrate the symmetry between the two measures. In addition, we define a third measure, termed a certainty factor (CF) that combines the MB and MD in accordance with the following definition:

\[
\text{CF}[h,e] = \text{MB}[h,e] - \text{MD}[h,e]
\]

The certainty factor thus is an artifact for combining degrees of Belief and Disbelief into a single number. Such a number is needed in order to facilitate comparisons of the evidential strength of competing hypotheses. The use of this composite number will be described below in greater detail. The following observations help to clarify the characteristics of the three measures that I have defined (MB, MD, CF):

**Characteristics of Belief Measures**

(1) Range of degrees:
(a) \(0 \leq \text{MB}[h,e] \leq 1\).
(b) \(0 \leq \text{MD}[h,e] \leq 1\).
(c) \(-1 \leq \text{CF}[h,e] \leq +1\).

(2) Evidential strength and mutually exclusive hypotheses†:
If \(h\) is shown to be certain \([P(h|e)=1]\):

†There is a special case of characteristic (2) that should be mentioned. This is the case of logical truth or falsity where \(P(h|e)=1\) or \(P(h|e)=0\), regardless of \(e\). Popper has also suggested a quantification scheme for confirmation [Popper, 1959] in which he uses \(-1 < C[h,e] < +1\), defining his limits as:
MYCIN

(a) $MB[h,e] = \frac{1 - P(h)}{1 - P(h)} = 1$.
(b) $MD[h,e] = 0$.
(c) $CF[h,e] = 1$.

If the negation of $h$ is shown to be certain [$P(\neg h|e) = 1$]:

(a) $MB[h,e] = 0$.
(b) $MD[h,e] = \frac{0 - P(h)}{0 - P(h)} = 1$.
(c) $CF[h,e] = -1$.

Note that this gives $MB[\neg h,e] = 1$ if and only if $MD[h,e] = 1$ in accordance with the definitions of $MB$ and $MD$ above. Furthermore, the number 1 represents absolute Belief (or Disbelief) for $MB$ (or $MD$). Thus if $MB[h_1,e] = 1$ and $h_1$ and $h_2$ are mutually exclusive, $MD[h_2,e] = 1$.

(3) Lack of evidence:

(a) $MB[h,e] = 0$ if $h$ is not confirmed by $e$ (i.e., $e$ and $h$ are independent or $e$ disconfirms $h$).
(b) $MD[h,e] = 0$ if $h$ is not disconfirmed by $e$ (i.e., $e$ and $h$ are independent or $e$ confirms $h$).
(c) $CF[h,e] = 0$ if $e$ neither confirms nor disconfirms $h$ (i.e., $e$ and $h$ are independent).

We are now in a position to examine Paradox 1 (§ 4.3), the expert’s concern that although evidence may support a hypothesis with degree $X$, it does not support the negation of the hypothesis with degree $1 - X$. In terms of our proposed model, this reduces to the assertion that, when $e$ confirms $h$:

$$CF[h,e] + CF[\neg h,e] \neq 1.$$ 


This proposal led one observer [Harre, 1970] to assert that Popper’s numbering scheme “obliges one to identify the truth of a self-contradiction with the falsity of a disconfirmed general hypothesis and the truth of a tautology with the confirmation of a confirmed existential hypothesis, both of which are not only question begging but absurd.” As I shall demonstrate in § 4.6, we avoid Popper’s problem by introducing mechanisms for approaching certainty asymptotically as items of confirmatory evidence are discovered.
Model of Inexact Reasoning in Medicine

This intuitive impression is verified by the following analysis:

\[
\text{CF}[\neg h,e] = \text{MB}[\neg h,e] - \text{MD}[\neg h,e]
\]

\[
= 0 - \frac{P(h|e) - P(\neg h)}{-P(\neg h)}
\]

\[
= \frac{[1 - P(h|e)] - [1 - P(h)]}{1 - P(h)} = \frac{P(h) - P(h|e)}{1 - P(h)},
\]

\[
\text{CF}[h,e] = \text{MB}[h,e] - \text{MD}[h,e]
\]

\[
= \frac{P(h|e) - P(h)}{1 - P(h)} - 0.
\]

Thus,

\[
\text{CF}[h,e] + \text{CF}[\neg h,e] = \frac{P(h|e) - P(h)}{1 - P(h)} + \frac{P(h) - P(h|e)}{1 - P(h)}
\]

\[
= 0.
\]

Clearly this result occurs because (for any \( h \) and any \( e \) \( \text{MB}[h,e] = \text{MD}[\neg h,e] \)). This conclusion is intuitively appealing since it states that evidence that supports a hypothesis disfavors the negation of the hypothesis to an equal extent.

We noted earlier that experts are often willing to state degrees of belief in terms of conditional probabilities but they refuse to follow the assertions to their logical conclusions (e.g., Paradox 1 above). It is perhaps revealing to note, therefore, that when the \textit{a priori} belief in a hypothesis is small (i.e., \( P(h) \) is close to zero), the CF of a hypothesis confirmed by evidence is approximately equal to its conditional probability on that evidence:

\[
\text{CF}[h,e] = \text{MB}[h,e] - \text{MD}[h,e] = \frac{P(h|e) - P(h)}{1 - P(h)} - 0 \approx P(h|e),
\]

whereas, as shown above, \( \text{CF}[\neg h,e] \approx -P(h|e) \) in this case. This observation suggests that confirmation, to the extent that it is adequately represented by CF's, is close to conditional probability (in certain cases) although it still defies analysis as a probability measure.

We believe, then, that the proposed model is a plausible representation of the numbers an expert gives when asked to quantify the
MYCIN

strength of his judgmental rules. He gives a positive number (CF>0) if the hypothesis is confirmed by observed evidence, suggests a negative number (CF<0) if the evidence lends credence to the negation of the hypothesis, and says there is no evidence at all (CF=0) if the observation is independent of the hypothesis under consideration. The CF combines knowledge of both $P(h)$ and $P(h|e)$. Since the expert often has trouble stating $P(h)$ and $P(h|e)$ in quantitative terms, there is reason to believe that a CF that weights both the numbers into a single measure is actually a more natural intuitive concept (e.g., "I don't know what the probability is that all ravens are black, but I do know that every time you show me an additional black raven my belief is increased by $X$ that all ravens are black."

If we therefore accept CF's rather than probabilities from experts, it is natural to ask under what conditions the physician's behavior based upon CF's is irrational. We know from probability theory, for example, that if there are $n$ mutually exclusive hypotheses $h_i$, at least one of which must be true, then $\Sigma^n P(h_i|e)=1$ for all $e$. In the case of certainty factors, we can also show that there are limits on the sums of CF's of mutually exclusive hypotheses. Judgmental rules acquired from experts must respect these limits or else the rules will reflect irrational quantitative assignments. (Note we assert that behavior is irrational if actions taken or decisions made contradict the result that would be obtained under a probabilistic analysis of the behavior.)

Sums of CF's of mutually exclusive hypotheses have two limits—a lower limit for disconfirmed hypotheses and an upper limit for confirmed hypotheses. The lower limit is the obvious value that results because $CF[h,e] \geq -1$ and because more than one hypothesis may have $CF=-1$. Note first that a single piece of evidence may absolutely disconfirm several of the competing hypotheses. For example, if there are $n$ colors in the universe and $C_i$ is the $i$th color, then $ARC_i$ may be used as an informal notation to denote the hypothesis that all ravens have color $C_i$. If we add the hypothesis $ARC_0$ that some ravens have different colors from others, we know $\Sigma^n P(ARC_i)=1$. Consider now the observation $e$ that there is a raven of color $C_n$. This single observation allows us to conclude that $CF[ARC_i,e]=-1$ for $1 \leq i \leq n-1$. Thus, since these $n-1$ hypotheses are absolutely disconfirmed by the observation $e$, $\Sigma^{n-1} CF[ARC_i,e]=-\sum_{i=1}^{n-1} 1 = -(n-1)$. This analysis leads to the general statement that, if $k$
mutually exclusive hypotheses \( h_i \) are disconfirmed by an observation \( e \):

\[
\sum_{i=1}^{k} CF[h_i,e] > -k \quad \text{(for } h_i \text{ disconfirmed by } e).}
\]

In the colored raven example, the observation of a raven with Color \( C_n \) still left two hypotheses in contention, namely \( ARC_n \) and \( ARC_0 \). What, then, are \( CF[ARC_n,e] \), \( CF[ARC_0,e] \), and the sum of \( CF[ARC_n,e] \) and \( CF[ARC_0,e] \)? The values of \( CF[ARC_n,e] \) and \( CF[ARC_0,e] \) are intimately related with the Paradox of the Ravens as discussed in Appendix 4.A. The limit on their sum, however, is important here as we attempt to characterize the rational use of CF’s. In fact, it can be shown that, if \( k \) mutually exclusive hypotheses \( h_i \) are confirmed by an observation \( e \), the sum of their CF’s does not have an upper limit of \( k \) but rather:

\[
\sum_{i=1}^{k} CF[h_i,e] < 1 \quad \text{(for } h_i \text{ confirmed by } e).}
\]

In fact, \( \sum_{i=1}^{k} CF[h_i,e] \) is equal to 1 if and only if \( k=1 \) and \( e \) implies \( h_1 \) with certainty, but the sum can get arbitrarily close to 1 for small \( k \) and large \( n \). The analyses that lead to these conclusions are included as Appendix 4.B.

The last result allows us critically to analyze new decision rules given by experts. Suppose for example, we are given the following rules: \( CF[h_1,e]=.7 \) and \( CF[h_2,e]=.4 \) where \( h_1 \) is “The organism is a streptococcus”, \( h_2 \) is “The organism is a staphylococcus”, and \( e \) is “The organism is a gram positive coccus growing in chains.” Since \( h_1 \) and \( h_2 \) are mutually exclusive, the observation that \( \sum_1^2 CF[h_i,e] > 1 \) tells us that the suggested certainty factors are inappropriate. The expert must either adjust the weightings or we must normalize them so that their sum does not exceed 1. In other words, because behavior based on these rules would be irrational, we must change the rules.

In concluding this section, I shall briefly examine Törnebohm’s criteria for acceptability of a theory of confirmation [Törnebohm, 1966]. He states that:

It would be desirable to have a measure of evidential strength or degree of confirmation \( Dc \) satisfying the following conditions:
Mycin

Dc1. If $E$ L-implies $H$, then $Dc(H|E) = \max$.
Dc2. If $E$ L-implies not $H$, the $Dc(H|E) = \min$.
Dc3. $Dc(HE|E) = Dc(H|E)$
Dc4. If $E$ and $H$ are independent of each other, then $Dc(H|E) = 0$.

Unfortunately it does not seem possible to construct a reasonable measure satisfying all these conditions . . .

Note that CF[$H,E$] satisfies Dc1, Dc2, and Dc4 for max=1 and min=-1. However, it can be shown† that CF[$HE, E$] = CF[$H, E$] if and only if $P(E|H) = 1$. Thus, despite its intuitive appeal, the CF we have defined fails to satisfy all four acceptability criteria suggested by Törnebohm. I shall point out later, however, that the conventions we have adopted for combining CF's allow us to satisfy Dc3.

4.6 Model as Approximation Technique

Certainty factors provide a useful way to think about confirmation and the quantification of degrees of belief. However, I have not yet described how the CF model can be usefully applied to the medical diagnosis problem. The remainder of this chapter will explain conventions that we have introduced in order to utilize the certainty factor model. Our starting assumption is that the numbers given us by experts who are asked to quantify their degree of Belief in decision criteria are adequate representations of the numbers that

†I shall demonstrate the result for $E$ confirming $H$. The proof for $E$ disconfirming $H$ is similar.

\[
\text{CF}[HE,E] = \text{MB}[HE,E] - \text{MD}[HE,E] \\
= \text{MB}[HE,E] - 0 \\
= \frac{P(HE|E) - P(HE)}{1 - P(HE)} = \frac{P(HE) - P(HE)}{1 - P(HE)}
\]

But

\[
= \text{MB}[H,E] - 0 \\
= \frac{P(H|E) - P(H)}{1 - P(H)}
\]

Thus CF[$HE, E$] = CF[$H, E$] if and only if:

\[
P(H) = P(HE) = P(E|\bar{H}) P(H)
\]

i.e., $P(E|\bar{H}) = 1$

176
Model of Inexact Reasoning in Medicine

would be calculated in accordance with the definitions of MB and MD if the requisite probabilities were known.

In § 4.2, when discussing Bayes' Theorem, I explained that I would like to devise a method that allows us to approximate the value for $P(D_i | E)$ solely from the $P(D_i | S_k)$, where $D_i$ is the $i$th possible diagnosis, $S_k$ is the $k$th clinical observation, and $E$ is the composite of all the observed $S_k$. I have explained why probabilities are inadequate representations of the decision rules with which we wish to deal. Thus our goal should be rephrased in terms of certainty factors as follows:

Suppose that MB$[D_i, S_k]$ is known for each $S_k$, MD$[D_i, S_k]$ is known for each $S_k$, and $E$ represents the conjunction of all the $S_k$. Then our goal is to calculate CF$[D_i, E]$ from the MB's and MD's known for the individual $S_k$'s.

Suppose that $E = S_1 \& S_2$, and that $E$ confirms $D_i$. Then:

$$CF[D_i, E] = MB[D_i, E] - 0 = \frac{P(D_i | E) - P(D_i)}{1 - P(D_i)} = \frac{P(D_i | S_1 \& S_2) - P(D_i)}{1 - P(D_i)}.$$  

Clearly there is no exact representation of CF$[D_i, S_1 \& S_2]$ purely in terms of CF$[D_i, S_1]$ and CF$[D_i, S_2]$. As was true for the discussion of Bayes' Theorem in § 4.2, the relationship of $S_1$ to $S_2$, within $D_i$ and all other diagnoses, needs to be known in order to calculate $P(D_i | S_1 \& S_2)$. Furthermore, the CF scheme adds one complexity not present with Bayes' Theorem because we are forced to keep MB's and MD's isolated from one another.‡ I shall therefore introduce an approximation technique for handling the net evidential strength of incrementally acquired observations. The combining convention must satisfy the following criteria (where $E_+$ represents all confirming evidence acquired to date, and $E_-$ represents all disconfirming evidence acquired to date):

‡Suppose $S_1$ confirms $D_i$ (MB>0) but $S_2$ disconfirms $D_i$ (MD>0). Then consider CF$[D_i, S_1 \& S_2]$. In this case, CF$[D_i, S_2 \& S_2]$ must reflect both the disconfirming nature of $S_2$ and the confirming nature of $S_1$. Although these measures are reflected in the component CF's (it is intuitive in this case, for example, that CF$[D_i, S_2] ≤ CF[D_i, S_2 \& S_2] ≤ CF[D_i, S_1]$), we shall demonstrate that it is important to handle component MB's and MD's separately in order to preserve commutativity (see item (3) of Defining Criteria).
MYCIN

Defining Criteria

(1) Limits:
   (a) $MB[h,E_+]$ increases towards 1 as confirming evidence is found, equalling 1 only if a piece of evidence logically implies $h$ with absolute certainty.
   (b) $MD[h,E_-]$ increases toward 1 as disconfirming evidence is found, equalling 1 only if a piece of evidence logically implies not $h$ with certainty.
   (c) $CF[h,E_-] \leq CH[h,E_+ & E_-] \leq CF[h,E_+]$.
      These criteria reflect our desire to have the measure of Belief approach certainty asymptotically as partially confirming evidence is acquired, and to have the measure of Disbelief approach certainty asymptotically as partially disconfirming evidence is acquired.

(2) Absolute confirmation or disconfirmation:
   (a) If $MB[h,E_+]=1$, then $MD[h,E_-]=0$ regardless of the disconfirming evidence in $E_-`; i.e., $CF[h,E_+] = 1$.
   (b) If $MD[h,E_-]=1$, then $MB[h,E_+]=0$ regardless of the confirming evidence in $E_+`; i.e., $CF[h,E_-] = -1$.
   (c) The case where $MB[h,E_+]=MD[h,E_-]=1$ is contradictory and hence the CF is undefined.

(3) Commutativity:
   If $S_1 & S_2$ indicates an ordered observation of evidence, first $S_1$ and then $S_2$:
   (a) $MB[h,S_1 & S_2] = MB[h,S_2 & S_1]$.
   (b) $MD[h,S_1 & S_2] = MD[h,S_2 & S_1]$.
   (c) $CF[h,S_1 & S_2] = CF[h,S_2 & S_1]$.
      The order in which pieces of evidence are discovered should not affect the level of Belief or Disbelief in a hypothesis. This criterion assures that the order of discovery will not matter.

(4) Missing information:
   If $S_3$ denotes a piece of potential evidence, the truth or falsity of which is unknown:
   (a) $MB[h,S_1 & S_3] = MB[h,S_1]$.
   (b) $MD[h,S_1 & S_3] = MD[h,S_1]$.
   (c) $CF[h,S_1 & S_3] = CF[h,S_1]$.
      The decision model should function by simply disregarding rules of the form $CF[h,S_2]=X$ if the truth or falsity of $S_2$ cannot be determined.

There are a number of observations to be made on the basis of these criteria. For example, items (1) and (2) indicate that the MB of a hypothesis never decreases unless its MD goes to 1. Similarly the
Model of Inexact Reasoning in Medicine

MD never decreases unless the MB goes to 1. In § 4.5, where it was always true that MB=0 or MD=0, it was always the case that either CF=MB=0 or CF=0-MD. As evidence is acquired sequentially, however, both the MB and MD may become nonzero. Thus CF=MB-MD is an important indicator of the net Belief in a hypothesis in light of current evidence. Furthermore, a certainty factor of zero may indicate either absence of both confirming and disconfirming evidence (as discussed in § 4.5), or the observation of pieces of evidence that are equally confirming and disconfirming. In effect CF[h,e]=0 is the "don’t know more than I did before" value (i.e., equally confirmed and disconfirmed). Negative CF’s indicate that there is more reason to disbelieve the hypothesis than to believe it. Positive CF’s indicate that the hypothesis is more strongly confirmed than disconfirmed.

It is important also to note that, if $E=E_\&E_\&$, then CF[h,E] represents the certainty factor for a complex new rule that could be given us by an expert. CF[h,E], however, would be a highly specific rule customized for the few patients satisfying all the conditions specified in $E_\&$ and $E_\&$. Since the expert gives us only the component rules, we seek to devise a mechanism whereby a calculated cumulative CF[h,E], based upon MB[h,E_&] and MD[h,E_&], gives a number close to the CF[h,E] that would be calculated if all the necessary conditional probabilities were known.

With these comments in mind, I therefore present the following four combining functions, the first of which satisfies the criteria that I have outlined. The other three functions are necessary conventions for implementation of the model.

### Combining Functions

(1) Incrementally acquired evidence:†

(a) \[ MB[h,S_1\&S_2] = \begin{cases} 
0 & \text{if } MD[h,S_1\&S_2] = 1, \\
MB[h,S_1] + MB[h,S_2](1 - MB[h,S_1]) & \text{otherwise.} 
\end{cases} \]

(b) \[ MD[h,S_1\&S_2] = \begin{cases} 
0 & \text{if } MB[h,S_1\&S_2] = 1, \\
MD[h,S_1] + MD[h,S_2](1 - MD[h,S_1]) & \text{otherwise.} 
\end{cases} \]

†It has been pointed out that the first of these functions is equivalent to:

\[ MB[h,S_2] = \frac{MB[h,S_1\&S_2] - MB[h,S_1]}{1 - MB[h,S_1]} \]

Thus this combining function parallels our original definition of an MB, but with MB's
(2) Conjunctions of hypotheses:
   (a) $\text{MB}[h_1 & h_2, E] = \min(\text{MB}[h_1, E], \text{MB}[h_2, E])$.
   (b) $\text{MD}[h_1 & h_2, E] = \max(\text{MD}[h_1, E], \text{MD}[h_2, E])$.

(3) Disjunctions of hypotheses:
   (a) $\text{MB}[h_1 \lor h_2, E] = \max(\text{MB}[h_1, E], \text{MB}[h_2, E])$.
   (b) $\text{MD}[h_1 \lor h_2, E] = \min(\text{MD}[h_1, E], \text{MD}[h_2, E])$.

(4) Strength of evidence:
   If the truth of falsity of a piece of evidence $S_1$ is not known with certainty, but a CF (based upon prior evidence $E$) is known reflecting the degree of Belief in $S_1$, then if $\text{MB}'[h, S_1]$ and $\text{MD}'[h, S_1]$ are the degrees of Belief and Disbelief in $h$ when $S_1$ is known to be true with certainty (i.e., these are the decision rules acquired from the expert) then the actual degrees of Belief and Disbelief are given by:
   (a) $\text{MB}[h, S_1] = \text{MB}'[h, S_1] \cdot \max(0, \text{CF}[S_1, E])$.
   (b) $\text{MD}[h, S_1] = \text{MD}'[h, S_1] \cdot \max(0, \text{CF}[S_1, E])$.

This criterion relates to our statement early in § 4.5 that evidence in favor of a hypothesis may itself be an hypothesis subject to confirmation. Suppose, for instance, you are in a darkened room when testing the generalization that all ravens are black. Then the observation of a raven that you think is black, but that may be navy blue or purple, is less strong evidence in favor of the hypothesis that all ravens are black than if the sampled raven were known with certainty to be black. Here the hypothesis being tested is “All ravens are black” and the evidence is itself an hypothesis, namely the uncertain observation that “This raven is black.”

Function (1) simply states that, since an MB (or MD) represents a proportionate decrease in Disbelief (or Belief), the MB (or MD) of a newly acquired piece of evidence should be applied proportionately to the Disbelief (or Belief) still remaining. Function (2)(a) indicates that the measure of Belief in the conjunction of two hypotheses is only as good as the Belief in the hypothesis that is believed less strongly, whereas Function (2)(b) indicates that the measure of Disbelief in such a conjunction is as strong as the Disbelief in the most strongly disconfirmed. Function (3) yields complementary results for disjunctions of hypotheses. The corresponding CF’s are substituted for the probability measures that we lack. Note also that this formula bears the same relationship to our MB definition as the sequential diagnosis form of Bayes’ Theorem does to the simple Bayes formula (§ 4.2).
Model of Inexact Reasoning in Medicine

merely calculated using the definition $\text{CF} = \text{MB} - \text{MD}$. The reader is left to satisfy himself that Function (1) satisfies the Defining Criteria. (Note that $\text{MB}[h, S_2] = \text{MD}[h, S_2] = 0$ when examining Criterion (4).) Functions (2) and (3) are needed in the use of Function (4). Consider, for example, a rule such as:

$$\text{CF}'[h, S_1 \& S_2 \& (S_3 \lor S_4)] = X;$$

i.e., in our format, a rule such as:


Then, by Function (4):

$$\text{CF}[h, S_1 \& S_2 \& (S_3 \lor S_4)] = X \cdot \max(0, \text{CF}[S_1 \& S_2 \& (S_3 \lor S_4), E])$$

$$= X \cdot \max(0, \text{MB}[S_1 \& S_2 \& (S_3 \lor S_4), E]$$

$$- \text{MD}[S_1 \& S_2 \& (S_3 \lor S_4), E]).$$

Thus, we use Functions (2) and (3) to calculate:

$$\text{MB}[S_1 \& S_2 \& (S_3 \lor S_4), E] = \min(\text{MB}[S_1, E], \text{MB}[S_2, E], \text{MB}[S_3 \lor S_4, E])$$

$$= \max(\text{MB}[S_2, E], \text{MB}[S_3 \lor S_4, E]).$$

$\text{MD}[S_1 \& S_2 \& (S_3 \lor S_4), E]$ is calculated similarly. It is also worth noting that Function (2) gives, for $H$ confirmed by $E$:

$$\text{CF}[HE, E] = \text{MB}[HE, E] - \text{MD}[HE, E]$$

$$= \min(\text{MB}[H, E], \text{MB}[E, E]) - \max(\text{MD}[H, E], \text{MD}[E, E])$$

$$= \min(\text{MB}[H, E], 1) - \max(\text{MD}[H, E], 0)$$

$$= \text{MB}[H, E] - \text{MD}[H, E]$$

$$= \text{CF}[H, E]$$

181
Thus the use of an approximation via Function (2) allows us to satisfy Dc3 of Törnebohm’s criteria (see end of § 4.5) and hence to satisfy all his requirements for a quantitative approach to confirmation.

An analysis of Function (1) in light of the probabilistic definitions of MB and MD does not prove to be particularly enlightening. The assumptions implicit in this function include more than an acceptance of the independence of $S_1$ and $S_2$. The function was conceived purely on intuitive grounds in that it satisfied the four Defining Criteria I have listed. However, some obvious problems are present. For example, the function always causes the MB or MD to increase, regardless of the relationship between new and prior evidence. Yet Salmon has discussed an example from subparticle physics [Salmon, 1973] in which either of two observations taken alone confirm a given hypothesis, but their conjunction disproves the hypothesis absolutely! Our model assumes the absence of such aberrant situations in the field of application for which it is designed. The problem of formulating a more general quantitative system for measuring confirmation is well recognized and referred to by Harré [Harré, 1970]: “The syntax of confirmation has nothing to do with the logic of probability in the numerical sense, and it seems very doubtful if any single, general notion of confirmation can be found which can be used in all or even most scientific contexts.” Although we have suggested that perhaps there is a numerical relationship between confirmation and probability, we agree that the challenge for a confirmation quantification scheme is to demonstrate its usefulness within a given context, preferably without sacrificing human intuition regarding what the quantitative nature of confirmation should be.

Our challenge with Function (1), then, is to demonstrate that it is a close enough approximation for our purposes. We have attempted to do so in two ways. First we have implemented the function as part of the MYCIN system (§ 4.7) and have demonstrated that the technique models the conclusions of the expert from whom the rules were acquired. Second, we have written a program that allows us to compare CF’s computed both from simulated real data and by using Function (1). Our notation for the following discussion will be as follows:
Model of Inexact Reasoning in Medicine

\[ CF^*[h,E] \] = the computed CF using the definition of CF from § 4.5 (i.e., "perfect knowledge" since \( P(h|E) \) and \( P(h) \) are known)

\[ CF[h,E] \] = the computed CF using Function (1) and the known MB's and MD's for each \( S_k \) where \( E \) is the composite of the \( S_k \)'s (i.e., \( P(h|E) \) not known but \( P(h|S_k) \) and \( P(h) \) known for calculation of \( MB[h,S_k] \) and \( MD[h,S_k] \))

The program was run on sample data simulating several hundred "patients." Clearly the question to be asked was whether \( CF[h,E] \) is a good approximation of \( CF^*[h,E] \). Figure 4-1 shows a graph summarizing our results. For the vast majority of cases, the approximate.

![Figure 4-1: Chart demonstrating the degree of agreement between CF and CF* for a sample data base. CF is an approximation to CF*. The terms are defined in the text.](image-url)
tion does not produce a $\text{CF}[h, E]$ radically different from the true $\text{CF}^*[h, E]$. In general, the discrepancy is greatest when Function (1) has been applied several times (i.e., several pieces of evidence have been combined). This result is in keeping with Zadeh's observation from fuzzy logic that "the more steps there are in the proof, the fuzzier the result" [Zadeh, 1974]. The most aberrant points, however, are those that represent cases in which pieces of evidence were strongly interrelated for the hypothesis under consideration (termed "conditional nonindependence"). This result is expected because it reflects precisely the issue that makes it difficult to use Bayes' Theorem for our purposes.

Thus I should make it clear that we have not avoided many of the problems inherent with the use of Bayes' Theorem in its exact form. We have introduced a new quantification scheme which, although it makes many assumptions similar to those made by subjective Bayesian analysis, permits us to utilize criteria as rules and to manipulate them to the advantages described in § 4.3. In particular, the quantification scheme also allows us to consider confirmation separately from probability and thus to overcome some of the inherent problems that accompany an attempt to put judgmental knowledge into a probabilistic format. Just as Bayesians who use their theory wisely must insist that events be chosen so that they are independent (unless the requisite conditional probabilities are known), we must insist that dependent pieces of evidence be grouped into single rather than multiple rules. As Edwards has pointed out [W. Edwards, 1972], a similar strategy must be used by Bayesians who are unable to acquire all the necessary data:

... [An approximation] technique is the one now most commonly used. It is simply to combine conditionally non-independent symptoms into one grand symptom, and obtain [quantitative] estimates for that larger more complex symptom.

The system therefore becomes unworkable for applications in which large numbers of observations must be grouped in the PREMISE of a single rule in order to insure independence of the decision criteria. In addition, we must recognize logical subsumption when examining or acquiring rules and thus avoid counting evidence more than once. For example, if $S_1$ implies $S_2$, then $\text{CF}[h, S_1 \& S_2] = \text{CF}[h, S_1]$ regard-
Model of Inexact Reasoning in Medicine

less of the value of $CF[h,S_2]$. Function (1) does not "know" this. Rules must therefore be acquired and utilized with care (see § 6.3).

The justification for our approach therefore rests not with a claim of improving on Bayes’ Theorem but rather with the development of a mechanism whereby judgmental knowledge can be efficiently represented and utilized for the modeling of medical decision making, especially in contexts where (a) statistical data are lacking, (b) inverse probabilities are not known, and (c) conditional independence can be assumed in most cases.

4.7 Mycin’s Use of Model

Formal quantification of the probabilities associated with medical decision making can become so frustrating that some investigators have looked for ways to dispense with probabilistic information altogether [Ledley, 1973]. Diagnosis is not a deterministic process, however, and we believe that it should be possible to develop a quantification technique that approximates probability and Bayesian analysis and that is appropriate for use in those cases where formal analysis is difficult to achieve. The certainty factor model that we have introduced is such a scheme. It has been implemented as a central component of the MYCIN system. The program uses certainty factors to accumulate evidence and to decide upon likely identities for organisms causing disease in patients with bacterial infections. A therapeutic regimen is then determined—one that is appropriate to cover for the organisms requiring therapy.

All of the program’s knowledge is stored in decision rules such as those described in § 4.2 and 4.3. Each rule has an associated certainty factor that reflects the measure of increased Belief or Disbelief of the expert who suggested the rule. The capturing of such quantitative medical intuitions has been the subject of recent investigations by others [Card, 1970b] but, as we have noted, our approach has been simply to ask the expert to rate the strength of the inference on a scale from 1 to 10 (see § 4.3).

MYCIN remembers the alternate hypotheses that are confirmed or disconfirmed by the rules for inferring an organism’s identity. With each hypothesis is stored its MB and MD, both of which are initially zero. When a rule for inferring identity is found to be true for the patient under consideration, the ACTION portion of the rule allows
either the MB or the MD of the relevant hypothesis to be updated using the first Combining Function (§4.6). When all applicable rules have been executed, the final CF may be calculated, for each hypothesis, using the definition $\text{CF} = \text{MB} - \text{MD}$. These alternate hypotheses may then be compared on the basis of their cumulative certainty factors. Hypotheses that are most highly confirmed thus become the basis of the program’s therapeutic recommendation.

Suppose, for example, that the hypothesis $H_1$ that the organism is a streptococcus has been confirmed by a single rule with a CF=.3. Then, if $E$ represents all evidence to date, $\text{MB}[H_1,E] = .3$ and $\text{MD}[H_1,E] = 0$. If a new rule is now encountered which has CF=.2 in support of $H_1$, and if $E$ is updated to include the evidence in the PREMISE of the rule, we now have $\text{MB}[H_1,E] = .44$ and $\text{MD}[H_1,E] = 0$. Suppose a final rule is encountered for which CF=-.1. Then if $E$ is once again updated to include all current evidence, we use Function (1) to obtain $\text{MB}[H_1,E] = .44$ and $\text{MD}[H_1,E] = .1$. If no further system knowledge allows conclusions to be made regarding the possibility that the organism is a streptococcus, we calculate a final result that $\text{CF}[H_1,E] = .44 - .1 = .34$. This number becomes the basis for comparison between $H_1$ and all the other possible hypotheses regarding the identity of the organism.

It should be emphasized that this same mechanism is used for evaluating all knowledge about the patient, not just the identity of pathogens. When the user answers a system-generated question, the associated certainty factor is assumed to be +1 unless he explicitly modifies his response with a CF (multiplied by 10) enclosed in parentheses. Thus, for example, the following interaction might occur (MYCIN’s prompt is in lower-case letters):

```
14) Did the organism grow in clumps, chains, or pairs?
**CHAINS (6) PAIRS (3) CLUMPS (-8)
```

This capability allows the system automatically to incorporate the user’s uncertainties into its decision processes. A rule that referenced the growth conformation of the organism would in this case find:

- $\text{MB}[\text{chains}, E] = 0.6$, $\text{MD}[\text{chains}, E] = 0$
- $\text{MB}[\text{pairs}, E] = 0.3$, $\text{MD}[\text{pairs}, E] = 0$
- $\text{MB}[\text{clumps}, E] = 0$, $\text{MD}[\text{clumps}, E] = 0.8$. 186
Consider, then, the sample rule we introduced in § 4.2:

\[ \text{CF}[\text{H}_1, \text{S}_1 \& \text{S}_2 \& \text{S}_3] = 0.7, \]

where \( \text{H}_1 \) is the hypothesis that the organism is a streptococcus, \( \text{S}_1 \) is the observation that the organism is gram positive, \( \text{S}_2 \) that it is a coccus, and \( \text{S}_3 \) that it grows in chains. Suppose gram stain and morphology were known to the user with certainty so that MYCIN has recorded:

\[ \text{CF}[\text{S}_1, \text{E}] = 1, \quad \text{CF}[\text{S}_2, \text{E}] = 1. \]

In the case above, however, MYCIN would find that:

\[ \text{CF}[\text{S}_3, \text{E}] = 0.6 - 0 = 0.6. \]

Thus, it is no longer appropriate to use the rule in question with its full confirmatory strength of .7. That \( \text{CF} \) was assigned by the expert on the assumption that all three conditions in the \textit{PREMISE} would be true with certainty. The modified \( \text{CF} \) is calculated using the fourth Combining Function (§4.6):

\[ \text{CF}[\text{H}_1, \text{S}_1 \& \text{S}_2 \& \text{S}_3] = \text{MB}[\text{H}_1, \text{S}_1 \& \text{S}_2 \& \text{S}_3] - \text{MD}[\text{H}_1, \text{S}_1 \& \text{S}_2 \& \text{S}_3] \\
= 0.7 \cdot \max(0, \text{CF}[\text{S}_1 \& \text{S}_2 \& \text{S}_3, \text{E}]) - 0. \]

Calculating \( \text{CF}[\text{S}_1 \& \text{S}_2 \& \text{S}_3, \text{E}] \) using the second Combining Function, this gives:

\[ \text{CF}[\text{H}_1, \text{S}_1 \& \text{S}_2 \& \text{S}_3] = 0.7 \cdot 0.6 - 0 \]
\[ = 0.42 - 0, \]

i.e.,

\[ \text{MB}[\text{H}_1, \text{S}_1 \& \text{S}_2 \& \text{S}_3] = 0.42 \]

and

\[ \text{MD}[\text{H}_1, \text{S}_1 \& \text{S}_2 \& \text{S}_3] = 0. \]

Thus, the strength of the rule is reduced to reflect the uncertainty regarding \( \text{S}_3 \). Function (1) is now used to combine .42 (i.e.,
MYCIN

MB[\(H_1, S_1 \& S_2 \& S_3\)] with the previous MB for the hypothesis that the organism is a streptococcus.

I have shown that the numbers thus calculated are approximations at best. Hence, it does not seem justifiable simply to accept as correct the hypothesis with the highest CF after all relevant rules have been tried. Therapy is therefore chosen to cover for all identities of organisms that account for a sufficiently high proportion of the possible hypotheses on the basis of their CF’s. This is accomplished by ordering them from highest to lowest and selecting all those on the list until the sum of their CF’s exceeds \(z\) (where \(z\) is equal to .9 times the sum of the CF’s for all confirmed hypotheses).

Finally, it should be noted that our definition of CF’s allows us to validate those of our rules for which frequency data become available. This will become increasingly important as the program becomes a working tool in the clinical setting where it can actually be used to gather the statistical data needed for its own validation. In the meantime, validation will necessarily involve the comments of recognized infectious disease experts who will be asked to evaluate the program’s decisions and advice. Early experience with a limited set of rules has provided suggestive evidence that MYCIN will someday give advice similar to that suggested by infectious disease experts [Shortliffe, 1974b]. We are therefore gaining confidence that the certainty factor approach will continue to prove itself as the number of decision rules increases and we acquire rules from additional infectious disease experts.

Appendices

APPENDIX 4.A PARADOX OF RAVENS

In order to examine the Paradox of the Ravens (§ 4.4), I introduce the following informal notation:

- \(iRB\) - the hypothesis that exactly \(i\) ravens are black
- \(ARB\) - the hypothesis that all ravens are black (i.e., \(yRB\), where \(y\) = the number of ravens)
- \(inBnR\) - the hypothesis that exactly \(i\) nonblack objects are nonravens
- \(AnBnR\) - the hypothesis that all nonblack objects are nonravens (i.e., \(znBnR\) where \(z\) = the number of nonblack objects)
Model of Inexact Reasoning in Medicine

BR - the observation of a raven that is found to be black
nBnR - the observation of a nonblack object that is found to be a nonraven

The paradox, then, is based on the observation that it is counterintuitive to assert that CF[ARB,nBnR] = CF[AnBnR,nBnR]. Yet our definition of a CF quickly leads to the conclusion that the equality does hold since ARB is logically equivalent to AnBnR and thus P(ARB|nBnR) = P(AnBnR|nBnR). It may therefore be tempting to assert that the certainty factor model of confirmation has failed to provide insight into the paradox.

However, as Suppes has pointed out [Suppes, 1966a], the reason the paradox occurs is because we are convinced that “we are right in our intuitive assumption that we should look at randomly selected ravens and not randomly selected nonblack things in testing the generalization that all ravens are black.” Expressed in terms of certainty factors, our intuition is that CF[ARB,BR] >> CF[ARB,nBnR] and, in fact, that CF[ARB,nBnR] = 0. Thus we prefer to sample ravens rather than nonblack objects in testing the hypothesis ARB, i.e., we feel that a black raven is significantly greater evidence in favor of the hypothesis than is a green vase.

Let us use our definition of CF, then, to calculate both CF[ARB,BR] and CF[ARB,nBnR]. We define:

\[ y = \text{the number of ravens in the universe} \]
\[ z = \text{the number of nonblack objects in the universe} \]

We then make the following two assumptions:

(1) \( z \geq y \)

This assumption, although clearly true for the example at hand, may seem bothersome as a requirement for the analysis. However, it can be shown that, in fact, the paradox is reversed for \( z < y \). Consider, for example, a universe of 100 ravens and 5 nonblack objects that may or may not be ravens. In this case observation of a green vase is clearly better evidence in favor of the hypothesis that all ravens (in this limited universe) are black than is the observation of a black raven.

Suppes uses another example to make this point [Suppes, 1966a]. Suppose we want to test the generalization that all voters in a specific district are literate. We can either sample voters and see whether they are literate or else sample illiterate individuals and check to be sure they are nonvoters. The preferable strategy seems intuitively to depend upon
whether there are more voters than illiterate individuals, i.e., on the relationship between \( z \) and \( y \) from our example.

(2) We initially have no knowledge regarding either colors of ravens nor distributions of colors in the universe. This assumption allows us to state that, before observing any ravens, we believe all the hypotheses \( iRB \) to be equally likely. This amounts to the assumption of a uniform distribution of the \( P(iRB) \) before sampling begins. The analysis proceeds more easily with this assumption, but it should be clear that another prior distribution will not alter the qualitative nature of our final result. Thus:

\[
P(iRB) = \frac{1}{y+1} \quad \text{for } 0 \leq i \leq y
\]

which leads to the conclusion that \( P(ARB) = P(yRB) = \frac{1}{y+1} \).

Using assumptions (1) and (2) we can also show that:

\[
P(inBnR) = \begin{cases} 
0 & \text{for } 0 \leq i < z - y \\
\frac{1}{y+1} & \text{for } z - y \leq i \leq z
\end{cases}
\]

The proof is left for you to complete (note that there can be no fewer than \( z - y \) nonravens among the \( z \) nonblack objects). It leads to the conclusion that \( P(AnBnR) = P(znBnR) = \frac{1}{y+1} \). This is an important result since \( ARB \) and \( AnBnR \) are logically equivalent and we therefore must require that \( P(ARB) = P(AnBnR) \).

From our definitions of certainty factors, we now note that:

\[
CF[ARB,BR] = MB[ARB,BR] - MD[ARB,BR] = MB[ARB,BR] - 0 = \frac{P(ARB|BR) - P(ARB)}{1 - P(ARB)} = \frac{P(ARB|BR) - \frac{1}{y+1}}{1 - \frac{1}{y+1}}
\]

and:

\[
CF[ARB,nBnR] = MB[ARB,nBnR] - MD[ARB,nBnR] = MB[ARB,nBnR] - 0 = \frac{P(ARB|nBnR) - P(ARB)}{1 - P(ARB)} = \frac{P(AnBnR|nBnR) - P(AnBnR)}{1 - P(AnBnR)} = \frac{P(AnBnR|nBnR) - \frac{1}{y+1}}{1 - \frac{1}{y+1}}
\]

190
Thus we can calculate $CF[ARB, BR]$ if we can derive $P(ARB|BR)$ and can calculate $CF[ARB, nBnR]$ if we can derive $P(AnBnR|nBnR)$. Both of the requisite conditional probabilities can be found using Bayes’ Theorem:

\[
P(ARB|BR) = \frac{P(BR|ARB)P(ARB)}{\sum_i P(BR|iRB)P(iRB)} = \frac{y}{\sum_i \frac{1}{i(y+1)}} = \frac{2/(y+1)}{\sum_i i \frac{1}{y(y+1)}}
\]

\[
P(AnBnR|nBnR) = \frac{P(nBnR|AnBnR)P(AnBnR)}{\sum_i P(nBnR|iBnR)P(iBnR)} = \frac{\frac{2z}{(y+1)(2z-y)}}{\frac{2}{y+1} - \frac{z}{2z-y}}
\]

Note that $P(ARB|BR) = P(ARB|nBnR)$ if $z=y$!

Thus:

\[
CF[ARB, BR] = \frac{\frac{2}{y+1} - \frac{1}{y+1}}{1 - \frac{1}{y+1}} = \frac{1}{y}
\]

and:

\[
CF[ARB, nBnR] = \frac{\frac{2z}{(y+1)(2z-y)} - \frac{1}{y+1}}{1 - \frac{1}{y+1}} = \frac{1}{2z-y}
\]

Note that $CF[ARB, BR] \geq CF[ARB, nBnR]$ and that the equality only holds when $z=y$. Thus, if there are fewer ravens than nonblack objects, observing a black raven confirms the hypothesis ARB more strongly than a green vase confirms that all ravens are black.
MYCIN

But we wished to show that our intuition is correct in suggesting that \( CF[ARB, BR] \gg CF[ARB, nBnR] \) and that \( CF[ARB, nBnR] = 0 \). As mentioned in the discussion of assumption (1) above, our intuition is tainted by our knowledge of real work. For instance, we may be willing to accept estimates of \( y \) and \( z \) such that \( y = 10^7 \) and \( z = 10^{15} \). Actually \( z \) is undoubtedly larger, but these numbers will suffice for current purposes. Then:

\[
CF[ARB, BR] = \frac{1}{10^7} = .0000001 \\
CF[ARB, nBnR] = \frac{1}{2 \cdot 10^{15} - 10^{17}} \approx \frac{1}{2 \cdot 10^{15}} \\
\approx .0000000000000005
\]

Clearly \( CF[ARB, nBnR] \) is essentially zero, and \( CF[ARB, BR] \) is significantly greater than \( CF[ARB, nBnR] \). Note, however, that these results are obtained only because we are willing to accept the original estimates for \( x \) and \( y \).

APPENDIX 4.B PROOF OF UPPER LIMIT

I include here a proof of the assertion that the sum of the CF's of confirmed but mutually exclusive hypotheses cannot exceed 1. Since \( MD[h, e] = 0 \) for a hypothesis that is confirmed by \( e \), \( CF[h, e] = MB[h, e] \) when \( e \) confirms \( h \). Suppose there are \( n \) mutually exclusive hypotheses \( h_i \) confirmed by evidence \( e \). Then we wish to identify the upper limit on \( \Sigma_i^n CF[h_i, e] \), i.e., on \( \Sigma_i^n MB[h_i, e] \). To simplify the manipulation of symbols, let:

\[
a_i = P(h_i | e) \quad \text{such that} \quad \sum_{i=1}^n a_i < 1, \\
b_i = P(h_i) \quad \text{such that} \quad \sum_{i=1}^n b_i < 1 \quad \text{and} \quad 0 < b_i < 1 \quad \text{for all} \quad i.
\]

Then:

\[
a_i > b_i \quad \text{for all} \quad i \quad \text{since the} \quad h_i \quad \text{are confirmed by} \quad e
\]

We wish to find the upper limit, if any, on:

\[
\Sigma MB[h_i, e] = \sum_{i=1}^n \frac{a_i - b_i}{1 - b_i}
\]

192
Model of Inexact Reasoning in Medicine

Proof: We first note that, for \( n = 1 \):

\[
\sum_{i=1}^{n} \frac{a_i - b_i}{1 - b_i} = \frac{a_i - b_i}{1 - b_i} < 1 \quad \text{since} \quad a_i < 1.
\]

For \( n > 1 \), however:

\[
\sum_{i=1}^{n} \frac{a_i - b_i}{1 - b_i} < \sum_{i=1}^{n} \frac{a_i}{(1 - b_i) \prod_{j \neq i} (1 - b_j)} \quad \text{(since} \quad \prod_{i=1}^{n} (1 - b_j) < 1 \text{)}
\]

\[
< \frac{\sum_{i=1}^{n} (a_i - b_i)}{\prod_{i=1}^{n} (1 - b_i)} = \frac{\sum_{i=1}^{n} a_i - \sum_{i=1}^{n} b_i}{\prod_{i=1}^{n} (1 - b_i)}.
\]

But:

\[
\prod_{i=1}^{n} (1 - b_j) = 1 - \sum_{i=1}^{n} b_i + \sum_{i=1}^{n} \sum_{j \neq i} b_i b_j - \sum_{i=1}^{n} \sum_{j \neq i} \sum_{k \neq j \neq i} b_i b_j b_k + \cdots
\]

\[
= 1 - \sum_{i=1}^{n} b_i + \sum_{i=1}^{n} \sum_{j \neq i} b_i b_j \left( 1 - \sum_{k \neq j \neq i} b_k \right)
\]

\[
+ \sum_{i=1}^{n} \sum_{j \neq i} \sum_{k \neq j \neq i} \sum_{l \neq k \neq j \neq i} b_i b_j b_k b_l \left( 1 - \sum_{m \neq l \neq k \neq j \neq i} b_m \right) + \cdots.
\]

And since \( \Sigma b_i < 1 \), \( 1 - \Sigma b_i > 0 \) in all terms above. Thus \( \prod_{i=1}^{n} (1 - b_j) > 1 - \Sigma^{n} b_i \). Therefore:

\[
\sum_{i=1}^{n} \frac{a_i - b_i}{1 - b_i} < \frac{\sum_{i=1}^{n} a_i - \sum_{i=1}^{n} b_i}{\prod_{i=1}^{n} (1 - b_i)}
\]

\[
< \frac{\sum_{i=1}^{n} a_i - \sum_{i=1}^{n} b_i}{1 - \sum_{i=1}^{n} b_i} < \frac{1 - \sum_{i=1}^{n} b_i}{1 - \sum_{i=1}^{n} b_i} \quad \text{(since} \quad \sum_{i=1}^{n} a_i < 1 \text{)}
\]

\[
< 1.
\]

Thus we have demonstrated that 1 is the upper limit for the sum of the CF's of confirmed mutually exclusive hypotheses.
The rather weak inequality we have shown is better understood, however, if we examine a special case. Suppose there are $m$ mutually exclusive hypotheses such that $\sum_{i=1}^m P(h_i)=1$. We assume that each is initially equally likely, i.e., $P(h_i)=1/m$. Suppose now that first $n$ of the $m$ hypotheses are confirmed by the evidence $e$. Then:

$$\sum_{i=1}^n CF[h_i,e] = \sum_{i=1}^n MB[h_i,e] - \sum_{i=1}^n MD[h_i,e]$$

$$= \sum_{i=1}^n \frac{P(h_i|e) - P(h_i)}{1 - P(h_i)} - 0 = \sum_{i=1}^n \frac{P(h_i|e) - 1/m}{1 - 1/m}$$

$$= \sum_{i=1}^n \frac{mP(h_i|e) - 1}{m-1} = \frac{1}{m-1} \left[ m \sum_{i=1}^n P(h_i|e) - n \right]$$

$$= \frac{m \sum_{i=1}^n P(h_i|e) - n}{m-1} < 1.$$  

This interesting result shows that the sum is equal to 1 only if $h_1$ is taken to be certain on the basis of $e$ and when $n=1$. If only two hypotheses remain possible after $e$ has been observed and all the others have been ruled out with certainty, $\sum_{i=1}^n P(h_i|e)=1$ but $\sum_{i=1}^n CF[h_i,e]=(m-2)/(m-1)$ and is therefore less than 1.
Chapter 5
Explanation System

5.1 Introduction

As was emphasized in Chapter 2, one of the primary requirements for user acceptance of a consultation program is an ability to explain decisions. Rule-based knowledge has greatly simplified the implementation of such a capability in the MYCIN system. The portion of the system used for explanation is termed Subprogram 2 (shown in Figure 1-1). It is automatically invoked at the end of each consultation session, and may also be accessed optionally during the consultation itself (see the QA option, § 3.3.2-2).

Since MYCIN explains decisions only in response to queries from the user, the explanation system is also a question-answering (QA) system. Subprogram 2 is therefore often called the QA-module, a term that reflects MYCIN’s debt to other AI programs for answering questions [Simmons, 1970; Fox, 1970].

The ability to answer questions obviously requires that the queries be understood. Since we have been anxious to minimize special training needed for use of the MYCIN system, we have been eager to let the physician ask questions using simple English. As discussed in § 1.3.1-7, however, writing programs to understand natural language is complex because of the myriad ways that individuals may choose to express themselves. Although several powerful techniques have been developed [Winograd, 1972; Woods, 1970; Schank, 1972], they all suffer from being either somewhat slow computationally or difficult to generalize in domains other than those for which they were designed. Since physicians will quickly reject a system that takes 2 or 3 minutes to answer a question, we sought an approach that would
emphasize speed of response rather than human-style discourse. Yet we did want to make the system powerful enough to answer most questions that a physician might want to ask. Since the goals of rapid response and powerful capabilities tend to work at cross purposes, we have been forced to try to strike a balance between the two. The approach described in this chapter is thus neither as fast as desirable (it requires 5 to 20 seconds to answer a question) nor as powerful (it has no sense of discourse, anaphora, or complex syntax). However, its performance is usually adequate, and an experienced user who becomes aware of its limitations is able to retrieve most of the information he desires. Furthermore, it should be emphasized that the consultation itself, which is after all the primary focus of the MYCIN system, requires no natural language processing. Use of the QA module is optional, and a physician who is in a hurry therefore need not take the time to seek explanations if he is satisfied with the advice the program has given.

As demonstrated in the sample consultation at the end of Chapter 1, the explanation system offers several options to the user:

**QUESTION-ANSWERING (QA) OPTIONS**

- **HELP** - prints this list
- **EQ** - requests an explanation of the specified question(s) from the consultation
- **IQ** - prefix to a question which asks about information acquired by the program during the consultation
- **NO PREFIX** - this question queries contents of decision rules in the system
- **PR** - requests that specified rule be printed
- **STOP** - escape from explanation system
- **RA** - entry to rule-acquisition module for recognized experts

In this chapter I describe each of these options. Only the **IQ** and **NO PREFIX** options require natural language processing.

Section 5.2 describes how each option is used, giving examples of each. However, the implementation details are rapidly changing and have been described fully elsewhere [Shortliffe, 1974b]. They will therefore not be included in this report. The chapter concludes with a brief discussion, in § 5.3, of the Explanation System’s limitations.
and of how we intend to improve the program's capabilities in the future.

5.2 Using Question-Answering System

Unlike the Consultation System (Subprogram 1) in which MYCIN takes the initiative, asking questions and waiting for the physician to respond, the Explanation System expects the user to guide the interaction. This approach allows the system to instruct the physician or explain its advice only with regard to specific topics that may be puzzling to the user. Thus MYCIN prints its prompt characters (the double asterisk—"**"), waits for a question, performs the requested procedure, redisplays the prompt characters, and then waits for the next user input. This process continues until the user enters the word STOP.

In this section, I describe the capabilities of the Explanation System, i.e., the various QA-options listed in §5.1. MYCIN checks every input sentence to see if it begins with one of the special prefixes (HELP, EQ, IQ, PR, STOP, or RA). If not, it assumes that the user has asked a rule-retrieval question. Examples of rule-retrieval questions are discussed in §5.2.1. The EQ and IQ options are explained in §5.2.2.

Option prefixes serve one of two purposes. Most allow MYCIN to perform certain repetitive tasks without invoking time-consuming natural language routines (e.g., HELP, EQ, PR, STOP, and RA). The IQ option, on the other hand, is required at present so that MYCIN can distinguish between the two principal kinds of questions that do involve English language understanding. Without the IQ prefix MYCIN would need to deduce the distinction on semantic or syntactic grounds, a complex problem that we have temporarily avoided by using the prefix mechanism. The distinction between IQ and rule-retrieval questions will be clarified as we proceed.

5.2.1 RULE-RETRIEVAL QUESTIONS

Since most of MYCIN's knowledge is contained in its corpus of 200 rules, many questions can be effectively answered by retrieving and displaying the relevant rule(s). Thus the challenge for MYCIN's
QA routines is to “understand” a question well enough to decide which rules should be retrieved.

5.2.1.-1 General Questions

MYCIN can answer rule-retrieval questions that are either specific to a given consultation (§ 5.2.1-2) or general in nature. General questions reference the corpus of rules without considering the status of the dynamic data base (Figure 1-1), i.e., they ask questions about MYCIN’s knowledge rather than about how that knowledge has been applied to the patient under consideration. The following examples demonstrate how MYCIN uses both rule-retrieval and its LISP-to-English translation capability in order to answer general questions from the user:

**WHAT DO YOU PRESCRIBE FOR ACTINOMYCETES INFECTIONS?**

**RULE002**

IF: THE IDENTITY OF THE ORGANISM IS ACTINOMYCETES
THEN: I RECOMMEND THERAPY CHOSEN FROM AMONG THE FOLLOWING DRUGS:
1 - PENICILLIN (.99)
2 - TETRACYCLINE (.99)
3 - ERYTHROMYCIN (.99)

**UNDER WHAT CONDITIONS IS AN ORGANISM APT TO BE A CONTAMINANT?**

Relevant rules: Several rules including:
RULE042 RULE106 RULE109 RULE123 RULE155
Which do you wish to see?
**RULE155**

IF: 1) THE SITE OF THE CULTURE IS ONE OF THOSE SITES THAT ARE NORMALLY STERILE, AND
   2) THE STAIN OF THE ORGANISM IS GRAMNEG
THEN: THERE IS STRONGLY SUGGESTIVE EVIDENCE (.8) THAT THE ORGANISM IS NOT A CONTAMINANT

**WHY DO YOU ASK THE AGE OF THE PATIENT?**
Explanation System

Relevant rules: Several rules including
RULE118 RULE135 RULE137 RULE138 RULE150
Which do you wish to see?
**150

RULE150

IF: 1) THE THERAPY UNDER CONSIDERATION IS SULFISOXAZOLE, AND
2) THE AGE (IN YEARS) OF THE PATIENT IS LESS THAN .58, AND
3) THE PATIENT IS JAUNDICED
THEN: IT IS DEFINITE (1) THAT SULFISOXAZOLE IS NOT A POTENTIAL THERAPY FOR USE AGAINST THE ORGANISM

**WHAT DOES IT MEAN IF PENICILLINASE IS ADDED TO THE CULTURE BOTTLE?

RULE039

IF: 1) THE SITE OF THE CULTURE IS BLOOD, AND
2) THE STAIN OF THE ORGANISM IS GRAMPOS, AND
3) PENICILLINASE WAS ADDED TO THE CULTURE
THEN: THERE IS WEAKLY SUGGESTIVE EVIDENCE (.4) THAT THE ORGANISM IS A CONTAMINANT

**WHAT IS RULE030?

RULE030

IF: THE IDENTITY OF THE ORGANISM IS KNOWN WITH CERTAINTY
THEN: IT IS DEFINITE (1) THAT THESE PROPERTIES - GRAM MORPH AIR - SHOULD BE TRANSFERRED FROM THE IDENTITY OF THE ORGANISM TO THIS ORGANISM

**IS THERE A RULE THAT DISCUSSES STREPTOCOCCAL MORPHOLOGY?

No. I have no such rules at present.

**DO YOU EVER CONCLUDE THAT GRAM POSITIVE COCCI ARE CONTAMINANTS?

THE FOLLOWING RULE MAY BE RELEVANT: RULE039
5.2.1-2 Questions Regarding Current Consultation

As discussed in § 3.3.3-3, MYCIN maintains a record of how it makes decisions during every consultation. This record is then used by the QA system in order to answer rule-retrieval questions that are specific to the current consultation. MYCIN answers these questions differently from the way it responds to general rule-retrieval questions. Instead of displaying all the relevant rules, the program mentions what rules were used, when they were invoked during the consultation (i.e., after which question number), and what certainty factor (CF) was associated with the specified conclusion. The following examples from the sample consultation in Chapter 1 demonstrate this capability:

**WHAT MADE YOU THINK ORGANISM-1 MIGHT BE A PROTEUS?


I USED RULE084 TO CONCLUDE THAT THE IDENTITY OF ORGANISM-1 IS PROTEUS. THIS GAVE A CUMULATIVE C.F. OF .119. THE LAST QUESTION ASKED BEFORE THE CONCLUSION WAS MADE WAS 24.

**HOW DID YOU DECIDE THAT ORGANISM-2 WAS THE SAME AS ORGANISM-1?


I USED RULE004 TO CONCLUDE THAT THE PSEUDOMONAS (ORGANISM-2) IS A PRIOR ORGANISM WITH THE SAME IDENTITY AS ORGANISM-1. THIS GAVE A CUMULATIVE C.F. OF .133. THE LAST QUESTION ASKED BEFORE THE CONCLUSION WAS MADE WAS 18.

5.2.2 QUESTIONS REGARDING DYNAMIC DATA BASE

The questions described in § 5.2.1-2 involve both rule-retrieval and access to the consultation record which is part of the dynamic data
Explanation System

base (shown in Figure 1-1). There are two additional kinds of question, however, which reference dynamic data but do not access the program's knowledge base of rules. These are described in the following two subsections.

5.2.2-1 IQ Questions

IQ questions require natural language processing in much the same way that rule-retrieval questions do. The IQ prefix is simply a flag for MYCIN so that it knows that the following question does not require rule-retrieval. IQ stands for "Informational Question." Unfortunately, it is often unclear to the novice user when this prefix is needed, so we are currently looking for ways to dispense with it (§ 5.3).

An IQ question merely requests the current value of some clinical parameter from the dynamic data base (see § 3.3.3). It does not ask how the value was acquired (i.e., whether supplied by the physician or inferred using rules). Thus "How did you decide ORGANISM-1 is a pseudomonas?" is a rule-retrieval question, but "Is ORGANISM-1 a pseudomonas?" requires the IQ prefix. The following examples from the sample consultation in Chapter 1 demonstrate the IQ capability:

**IQ WHAT IS THE FINAL DECISION REGARDING THE IDENTITY OF
**ORGANISM-1?
THE IDENTITY OF ORGANISM-1 IS PSEUDOMONAS (.357)
KLEBSIELLA (.177) ENTEROBACTER (.166) E.COLI (.137)
PROTEUS (.118) SERRATIA (.042)

**IQ DID YOU DECIDE ORGANISM-2 WAS THE SAME AS
**ORGANISM-1?
ORGANISM-4 (.19) ORGANISM-3 (.19) ORGANISM-2 (.19) IS A
PRIOR ORGANISM WITH THE SAME IDENTITY AS
ORGANISM-1

5.2.2-2 EQ Command

During a consultation the user may request an explanation of any question that he is asked (see the RULE and WHY options, § 3.3.2-2). We also wanted to permit the physician to request such explanations after the consultation is complete. Therefore MYCIN maintains a record of every question asked. (This same record is used...
for changing the answers to questions, as described in § 3.6.1.) The EQ option allows the physician to ask MYCIN for explanations of consultation questions (EQ = Explain Question). MYCIN answers by specifying the clinical parameter and context that were being considered when the question was asked. It also gives the rule that caused the question to be generated. Thus:

**EQ 48
48 QUESTION 48 WAS ASKED IN ORDER TO FIND OUT THE PATIENT'S DEGREE OF SICKNESS (ON A SCALE OF 4) IN AN EFFORT TO EXECUTE RULE068.

The EQ command accepts a list of question numbers as arguments and explains each in the manner demonstrated. The user may then display any rules with which he is not familiar by using the PR command (§ 5.2.3). Note that the EQ command requires no language processing. If anything following the command is not a legal question number, it is simply ignored.

5.2.3 ADDITIONAL OPTIONS

In addition to the options already described, the user of the explanation system may give the HELP, STOP, RA, and PR commands. The first three take no arguments. HELP simply displays the list of user options and thus parallels the HELP command available during the consultation itself (§ 3.3.2-2). STOP provides a mechanism for escaping from Subprogram 2 once the user is through asking questions. RA is available only to experts who are known to the system. It permits the user to enter the Rule-Acquisition System (Subprogram 3 shown in Figure 1-1) which is described in Chapter 6.

The PR command provides a quick way to ask the rule-retrieval question “What is RULE030?” (see § 5.2.1-1). It accepts one or more numbers as arguments and assumes that they correspond to the numbers of rules that the user wishes to see. Thus “PR 30” causes RULE030 to be printed. Several examples of the PR option are included in the sample consultation at the end of Chapter 1. Readers who are interested in the details of how the options described in this section have been implemented should consult Chapter 6 in [Shortliffe, 1974b].
Explain System

5.3 Future Extensions

Improvements to MYCIN's language and explanation capabilities must necessarily bear in mind the important balance between comprehension and speed of execution. By customizing MYCIN's capabilities to the unique characteristics of its rule-based knowledge, we have managed to devise a surprisingly powerful although simplistic approach to question-answering. MYCIN does not "understand" questions in the sophisticated ways that characterize the most powerful and general of today's natural language systems. Yet it still manages to answer many questions adequately without a large expenditure of computer time during the analysis of each question. Since the language capabilities of MYCIN have been developed in response to a clear need for an explanation system (Chapter 2), rather than because of an inherent interest in the theory of language or computational linguistics, we are content at present to build upon the simple characteristics and limited power of MYCIN's current approach.

We are less than pleased, however, with those aspects of the current approach that will clearly interfere with the program's acceptability to physicians. Although doctors can learn to phrase their questions simply and to expect rules in response, limits on the kind of questions that can be asked or answered commonly lead to user frustration. We have therefore identified the following short-term goals for improvement of the Explanation System's language capabilities:

1. Development of a mechanism for permitting the physician to ignore the distinction between IQ and rule-retrieval questions; the IQ prefix should be unnecessary and MYCIN should itself deduce when a question is merely asking for the value of a parameter rather than for rule-retrieval.
2. Development of a mechanism for answering questions regarding those parts of MYCIN's knowledge that are not rule-based (see §3.2.6-1); the current approach does not permit QA access to simple lists or knowledge tables.
3. Development of methods, as discussed in §3.7, for moving algorithmic knowledge from functions to rules so that questions regarding therapy selection may be answered using standard rule-retrieval techniques.

Finally, work is currently underway to improve MYCIN's explanation capabilities during the consultation itself. The RULE command
we described in § 3.3.2-2 is less than satisfactory as an explanation or educational mechanism because it does not explain why the current rule has been invoked by MYCIN's goal-oriented control structure. A series of commands to allow the user to manipulate the entire reasoning chain is currently under development and should greatly enhance MYCIN's ability adequately to explain its questions and reasoning processes [Shortliffe, 1975b; Davis, 1976].
Chapter 6

Future Directions for MYCIN

6.1 Introduction

There are several questions regarding MYCIN’s performance that are currently unanswered. Most of these involve issues that cannot be adequately analyzed until the program has been introduced for ongoing use in the clinical setting. This chapter introduces our plans for clinical implementation and evaluation of MYCIN. It also discusses some immediate and long range goals for expansion of MYCIN’s capabilities.

Section 6.2 discusses the current status of the MYCIN project, the short range goals, and the way in which the research group is currently organized. Section 6.3 then briefly describes one of MYCIN’s goals that has already received considerable attention, namely the problem of rule-acquisition (Subprogram 3 shown in Figure 1-1). I first explain the current operation of the Rule-Acquisition System and then proceed to a discussion of what additional capabilities will be needed. I also discuss the way in which MYCIN can automatically identify and correct inconsistencies or contradictions as new knowledge is added to the corpus of system rules, and conclude with an assessment of how a growing rule corpus will affect system performance.

Section 6.4 discusses evaluation questions that must be answered both before and after MYCIN is implemented in the ward setting. The remainder of the chapter then deals with issues that are not immediate concerns but which reveal the potential for eventual wide influence of a program like MYCIN. Section 6.5 deals with how MYCIN could efficiently be implemented as a module in a total
MYCIN

Hospital Information System (HIS) or in any environment where computer-based patient data could be shared. Section 6.6 takes the HIS example one step further, pointing out ways in which MYCIN could be instituted as a nonpunitive peer review mechanism for prospective monitoring of physician prescribing habits. Section 6.7 then discusses the potential educational applications of MYCIN, and I conclude with brief mention of other task domains in which the MYCIN formalisms can perhaps be applied.

6.2 Plans for Immediate Future

The work described in this text has involved the combined efforts of several collaborating physicians and computer scientists. After a two year growing period, during which the program gradually took shape, MYCIN began to interest other individuals who were able to devote time to the project. Research funding also became available and, as a result, MYCIN currently involves the full time efforts of at least eight individuals. This infusion of people with diverse interests, but united by a common fascination with applications of AI in medicine, has enabled MYCIN to begin to expand in a number of new directions. In this section, I shall describe some of these projects.

The primary concern at present is to introduce MYCIN in the clinical setting at Stanford Hospital. This initially involves developing the program’s knowledge base for bacteremia until we are convinced that MYCIN does indeed give expert advice for patients with that subset of bacterial infections. Clinical fellows in infectious diseases and clinical pharmacology are currently analyzing MYCIN’s rules and exercising the program with actual patient cases in an effort to identify additional rules, both for bacteremia and other infectious disease problems, that will help to improve the program’s performance.

Once the knowledge-base is deemed adequate, interactive terminals will be placed on appropriate wards at Stanford Hospital and the affiliated Veterans Administration Hospital in Palo Alto. Since users often need to refer back to parts of a consultation, quiet but fast hard-copy terminals will probably be utilized. After physicians have been educated regarding MYCIN’s availability and how it is used, a
Future Directions for MYCIN

formal evaluation of the program’s clinical impact and acceptability will be undertaken. Current prescribing habits will be monitored prior to introduction of the program so that valid control data will be available.

Chapters 3 and 5 both closed with discussions of some of the recognized improvements needed for Subprograms 1 and 2. Work on some of these problems is already underway. In particular, one project member is studying the problem of transferring function-based knowledge about drug selection to rules. A second investigator is examining the current design of the Explanation System to see whether the IQ prefix can be dropped from informational questions (§ 5.2.2-1) without introducing so much syntactic or semantic processing that the QA-module becomes unworkably slow.

Finally, one project member is examining several issues related to computer programs that “understand” their own operation. MYCIN provides an interesting practical environment for this kind of theoretical study because its goal-oriented control structure and formalized rules provide generalized data structures that do let the program analyze itself. The WHY option to which we have alluded (§ 3.3.2-2) is the first result of this work, but attention is also being paid to the semantics of certainty factors, rule-acquisition, and problems resulting from the interaction of new rules with a large corpus of pre-existing rules.

6.3 Knowledge Acquisition

Although we have already spent much time studying mechanisms for acquisition of new rules and have also undertaken some preliminary programming, so many of the problems in this domain remain unsolved that we have postponed discussing the current status until this chapter. Rule-acquisition is accomplished via Subprogram 3 (shown in Figure 1-1). As indicated in the figure, this subprogram may be entered from Subprogram 2 if the user is an infectious disease expert who is recognized by the system (see the RA option, § 5.2.3). The expert enters a new rule in English, it is translated into LISP, and the rule is then added to the knowledge base so that it will be available for future consultations.

It might seem reasonable to call rule-acquisition either teaching
(by the expert) or learning (by the machine). Both terms are potentially misleading, however, because “teaching” may lead to confusion with Computer-Aided Instruction (CAI) and “learning” has a rather special meaning in the AI field. When a program “learns,” the term usually means that experience has allowed an intelligent program to infer a truth or strategy and to incorporate the fact or heuristic into its knowledge-base. For example, I used the word when describing Waterman’s poker program [Waterman, 1970] and its ability to “learn” heuristics. The classic example of a learning program is Samuel’s checker-playing system that modifies its evaluation function in response to experience playing the game and has thereby improved so that it regularly beats its creator [Samuel, 1959, 1967]. Winston described a program that “learns” how to identify geometric objects from examples and counterexamples [Winston, 1970].

As currently envisioned, Subprogram 3 differs from these examples of “learning” programs in that it waits to be told what it needs to know. Thus the expert must deduce exactly what information is missing from the system or what previous rule is incorrect. Although the Explanation System simplifies this task, the expert is the primary problem solver for improving MYCIN’s knowledge base. Possible mechanisms for changing this emphasis are discussed in § 6.5.

6.3.1 CURRENT STATUS OF RULE-ACQUISITION

The current version of Subprogram 3, although it is limited in usefulness, does serve to demonstrate both the generality of MYCIN’s natural language capabilities and a potential methodology for powerful interactive knowledge acquisition. Limited effort has been spent on this capability to date, and the speed with which a mechanism for learning simple rules was developed suggests that more concentrated efforts in this area may well prove fruitful in a relatively short period of time.

6.3.1-1 Subprogram 3

Subprogram 3 allows an expert either to enter a new decision rule or to change a pre-existing rule that is in some way inadequate. Both tasks require similar computer processing, so I shall first discuss
Future Directions for MYCIN

acquisition of new rules and then explain the necessary modifications for altering old rules.

Subprogram 3 acquires new rules using the following ten-step procedure:

(1) Tells expert name of rule he is creating;
(2) Acquires PREMISE conditions one-by-one, translating each from English into the corresponding LISP representation;
(3) Acquires ACTION clauses one-by-one, translating each into its LISP representation and requesting an associated certainty factor (CF) when necessary;
(4) Displays English translation of rule using standard LISP-to-English routines (§ 3.2.7);
(5) Asks user to approve translated version; if rule is not correct, allows him to make changes and then goes back to Step (4);
(6) Searches for contradictions, inconsistencies, or subsumptions involving the new rule and other rules that are already part of the knowledge base; interacts with user as necessary in order to clarify any problems that are noted;
(7) Asks for assistance classifying rule, if necessary (§ 3.2.2-2);
(8) Adds rule to LOOKAHEAD list for all clinical parameters referenced in PREMISE (§ 3.2.3-2);
(9) Adds rule to either CONTAINED-IN list or UPDATED-BY list for all clinical parameters referenced in ACTION (§ 3.2.3-2);
(10) Tells expert that new rule is part of MYCIN's knowledge-base.

Note that Step (9) insures that the rule will automatically be invoked when appropriate because FINDOUT will find the rule on the UPDATED-BY list of the parameter for which the new decision criterion is useful (§ 3.3.2-1). Note also that the current procedure ignores ELSE clauses since these are seldom needed; the ability to add an ELSE clause to a rule may eventually be added.

Section 6.3.1-2 discusses our initial plans regarding the implementation of Step (6). The impact of a new rule on the pre-existing corpus is an important theoretical problem, particularly in light of the complexities introduced by our certainty factor scheme. In Section 6.3.1-3, I close with a discussion of the way a growing rule corpus may affect the speed and efficiency of consultation sessions.

This subsection will first conclude, however, with a sample interactive session illustrating the current capabilities of the Rule-
Acquisition System. As usual, words entered by the physician follow the double asterisk:

**NR
The new rule will be called RULE200

IF: 1 **ORGANISM IS A GRAM NEGATIVE ROD
    and 2 **IT IS ANAEROBIC
    and 3 **IT ENTERED THE BLOOD FROM THE GI TRACT
    and 4 **
THEN: 1 **IT IS PROBABLE A BACTEROIDES
    On a scale of 1 to 10, how much certainty would you affix to this conclusion
    **9
    and 2 **

This is my understanding of your rule:
RULE200

IF: 1) THE SITE OF THE CULTURE IS BLOOD, AND
    2) THE SITE OF THE CULTURE IS GI, AND
    3) THE STAIN OF THE ORGANISM IS GRAMNEG, AND
    4) THE MORPHOLOGY OF THE ORGANISM IS ROD, AND
    5) THE AEROBICITY OF THE ORGANISM IS ANAEROBIC
THEN: THERE IS STRONGLY SUGGESTIVE EVIDENCE (.9) THAT THE IDENTITY OF THE ORGANISM IS BACTEROIDES

Okay? (YES or NO)
**NO

In this example MYCIN has incorrectly interpreted the expert’s statement regarding the gastrointestinal (GI) tract. The physician has therefore indicated that the program’s understanding of the rule is not yet correct. The session continues:

Change PREMISE? (YES or NO)
**YES
Want to see the current version? (YES or NO)
**NO
Change (C), Delete (D), or Add (A) a component:
**C
Which Component? (Enter number)
**2

210
Enter the new component:
New: **PORTAL OF ENTRY IS THE GI TRACT**
More changes to this PREMISE? (YES or NO) **NO**
Change ACTION? (YES or NO) **NO**

This is my understanding of your rule:

**RULE200**

IF: 1) THE SITE OF THE CULTURE IS BLOOD, AND
   2) THE STAIN OF THE ORGANISM IS GRAMNEG, AND
   3) THE MORPHOLOGY OF THE ORGANISM IS ROD, AND
   4) THE AEROBICITY OF THE ORGANISM IS ANAEROBIC, AND
   5) THE PORTAL OF ENTRY OF THE ORGANISM IS GI
THEN: THERE IS STRONGLY SUGGESTIVE EVIDENCE (.9) THAT THE IDENTITY OF THE ORGANISM IS BACTEROIDES

Okay? (YES or NO) **YES**

This time the rule correctly reflects the intention of the expert. Note that the PREMISE conditions have been reordered for efficiency; clauses referencing parameters which are most likely to have been referenced before (and whose values are thus apt to be known) are placed near the beginning of the PREMISE. The algorithm used for reordering is described elsewhere [Shortliffe, 1974b].

Steps (1) through (5) of the acquisition procedure have been completed in the sample session above. Of the remaining steps, only Step (7) currently requires further interaction with the user. MYCIN can easily infer that the new rule is some kind of organism rule, but it is not obvious whether it should be classified as an ORGRULE, a CURORGRULE, or a PRORGRULE (§3.2.2.2). Therefore, MYCIN concludes with the following question:

This rule may best be described as a rule which:
1 - Applies to all organisms
2 - Applies to prior organisms only
3 - Applies to current organisms only
**1
Thank you for your assistance.
RULE200 is now part of the Consultation System.

RULE200 is thus classified as an ORGRULE and the rule-acquisition procedure is complete. The rule has been added to the LOOKAHEAD list for SITE, GRAM, MORPH, AIR, and PORTAL and to the UPDATED-BY list for IDENT. Thus the rule will be invoked whenever MYCIN is trying to infer the identity of an organism. The internal representation of the rule created by the above interaction is as follows:

RULE200

PREMISE: ($AND (SAME CNTXT SITE BLOOD)  
(SAME CNTXT GRAM GRAMNEG)  
(SAME CNTXT MORPH ROD)  
(SAME CNTXT AIR ANAEROBIC)  
(SAME CNTXT PORTAL GI))

ACTION: (CONCLUDE CNTXT IDENT BACTEROIDES TALLY .9)

MYCIN's mechanism for changing rules parallels the above procedure, starting at the point where the expert was asked if he wanted to change the PREMISE of RULE200. Thus when the physician indicates that he wants to change a rule, he is asked for the name of the rule requiring alteration and is then permitted to modify only that portion of the rule which is faulty. It is not necessary to delete the erroneous rule and to re-enter it from the beginning as though it were new.

Although we are eager to permit experts to teach the system new rules, there are potential dangers in letting anyone have uncontrolled access to MYCIN's knowledge base. This observation is particularly worrisome while the Step (6) consistency check is in rather rudimentary form. We therefore do not yet automatically store new rules as part of the permanent Consultation System. Instead they are stored temporarily in a file assigned specifically to the expert from whom the rules were acquired. Whenever that expert uses the system he may load his personal rules and they are temporarily added to MYCIN's knowledge base. MYCIN project members have an opportunity to examine both the new rules and the English text from
which they were derived, however, before the new knowledge is transferred from the expert's personal file to the permanent Consultation System.

6.3.1-2 (*) Interaction of New and Old Rules

Step (6) of the rule-acquisition procedure (§ 6.3.1-1) requires a screening process to see if the new rule improperly interacts with other rules in the knowledge-base. Although we have given considerable thought to this problem, Subprogram 3 does not yet undertake this consistency check. Programs to accomplish the necessary screening will be written in the near future, however, and I present here some preliminary observations.

Subsumption: I mentioned the problem of subsumption several times in Chapter 4. Of all the aberrant interactions of new rules with the pre-existing corpus, subsumption is perhaps the easiest to handle in an automated fashion. Suppose, for example, there were already a rule in the corpus as follows (see § 3.2.4 for an explanation of the notation):

\[ a \quad A \land B \land C \rightarrow \neg D \]

If an expert now entered the following new rule, a problem of subsumption would arise:

\[ b \quad A \land B \land C \land E \land F \rightarrow \neg D \]

Clearly any time rule [b] is satisfied, rule [a] will also be satisfied since the PREMISE of [b] subsumes the PREMISE of [a]. Yet rule [a] adds nothing to [b] and it would be improper to use both rules in the same context. On the other hand, eliminating [a] is not an adequate solution because [a] may apply in contexts where [b] does not and in those cases the knowledge inherent in [a] is needed.

The solution to the problem is to modify [a] so that it is no longer subsumed by [b] but so that it still will hold for all contexts that fail for [b] but would succeed for [a]. Namely, we propose replacing [a] with a new rule [c]:

\[ c \quad A \land B \land C \land (\neg E \lor \neg F) \rightarrow \neg D \]
Now any context that would have satisfied [a] will succeed either for [b] or [c] but not for both. Negation here implies a predicate's complement over the certainty factor range. Thus not.SAME is NOTSAME, not THOUGHTNOT (§ 3.2.5). The transition from [a] to [c] may be accomplished automatically except for the possible change in CF (from $x$ to $z$). We therefore propose displaying [c] for the expert and asking for the CF he would assign.

Finding old rules such as [a] that are subsumed by new rules such as [b] does not require a search through the entire rule corpus. MYCIN merely uses the LOOKAHEAD and UPDATED-BY lists for the clinical parameters in [b] to find rules that use all or some of the same parameters to deduce values of the same parameter. These rules may then be checked for subsumption.

**Single-Rule Contradictions:** It is also easy to find single rule contradictions using LOOKAHEAD and UPDATED-BY lists. However, the discrepancies cannot be handled in an automated fashion and the inconsistencies must be "discussed" with the user. Two rules contradict each other if they use the same conditions to reach the same conclusion but with different certainty factors. Clearly the extreme case occurs when one CF is positive and the other is negative; in such instances the experts disagree not only on the degree of evidence but also on the direction of evidence! Although such contradictions have not yet arisen during the development of MYCIN, Subprogram 3 must be prepared to identify and handle such problems if they do occur. Hopefully the expert will usually suggest a compromise CF that is also acceptable to the expert from whom the old rule was acquired. Expert clinicians often disagree on clinical questions, however, and we must be willing to accept this fact during the design of MYCIN's knowledge acquisition capabilities. If no acceptable compromise can be found, it may be necessary to store both of the rules and later to ask the user whether he wishes advice based upon the rules acquired from Dr. X or those from Dr. Y. This solution does not seem unreasonable since physicians commonly do have to choose among consultants.

**Multiple-Rule Contradictions:** The most complex interactions between a new rule and the pre-existing corpus occur when the new rule is inconsistent not with a single old rule but with a reasoning chain of old rules. Not only are such inconsistencies difficult to find, but it is also difficult to judge the severity of contradictions because of the interaction of reasoning chains with the CF's of the compo-
Future Directions for MYCIN

ent rules. In fact, unless the new rule has CF=1 or the reasoning chain is comprised only of rules with CF=1 (a situation for which the descriptive term “unity path” has been coined), it may perhaps be argued that no true contradiction exists. We are currently examining the nature of such inconsistencies in order to decide both how to find them using an automated mechanism and also under what conditions they may be ignored.

6.3.1-4 Impact of Knowledge Growth on System Performance

A question we are often asked is whether rule-acquisition will lead to an exponential growth problem. If each new rule permitted an entire new pathway to sprout in the reasoning network below it, we would have to expect exponential growth of search time as the number of rules increased. Indeed, if each new rule referenced several clinical parameters with which the system was not already familiar, and if each of these attributes in turn required a series of rules for use in inferring its value, both the size of the network and the time required for a consultation would grow unmanageably large. Our experience has so far indicated, however, that most new rules reference only the 65 clinical parameters with which the system is already familiar. Since each of these attributes is traced by the FINDOUT mechanism at most once during a consultation session, a new rule referencing parameters already traced for other reasons will generate no additional search time (except for that required to evaluate the single rule itself). Thus, growth in the size of the reasoning network and in search time is at most linear for a new rule that references only clinical parameters that are already recognized and traced by MYCIN. Furthermore, the new rule will have no effect whatsoever on search time in consultations where it is not invoked by the dynamic FINDOUT mechanism. Since we expect that the number of clinical parameters will not increase in proportion to the number of rules, we do not anticipate exponential growth problems.

6.3.2 FUTURE EXTENSIONS

The current rule-acquisition mechanism is limited in scope and applicability for a variety of reasons. Although the current approach may perhaps be adjusted so that it will accept all well-formed rules referencing clinical parameters known to the system, its dependence
upon MYCIN's dictionary (see Chapter 8 in [Shortliffe, 1974b]) results in an inability to learn rules that relate to unfamiliar concepts. Suppose, for example, that an expert wishes to enter a rule in which a condition involves a clinical parameter that has not been used by any existing rule. The system will miss the relevant vocabulary clues since there will be no dictionary entries related to that parameter.

Furthermore, suppose the new parameter refers to a context-type that has never before been created. For example, the length of time a Foley catheter has been in place might be a new parameter (TIME-IN-PLACE) for a new kind of context (CATHETER or PROCEDURE). Learning about new kinds of nodes for the context tree is a problem for which the current approach is clearly ill-equipped.

Finally, the most serious problem arises if the expert references a predicate (function) which has not previously been programmed. An attempt to automate this kind of rule-acquisition will perhaps be dependent upon the AI field known as automatic programming.

Future extensions to Subprogram 3 will therefore attempt to handle unrecognized parameters, contexts, and functions. We have not yet defined how best to approach these problems. We are currently relying on a variety of specialized "service" functions for defining new parameters explicitly and for easily entering and editing LISP versions of rules when Subprogram 3 has failed to acquire their English versions correctly.

In closing this section, I should also mention an entirely different approach to knowledge acquisition that was recently suggested. MYCIN could ask an expert how to treat a specific patient and, if the advice differed from that which the program would have recommended, could seek explanations from the expert. For example:

MYCIN: What would you prescribe?
EXPERT: I would give X.
M: Oh? Why would you give X?
E: It is the first choice drug for Y.
M: True. But how did you know the organism was Y?
E: Because its PARAM was Z.
M: I see. But how did you know its PARAM was Z?
The natural language problems inherent in this approach appear to be somewhat formidable, but the idea is rather enticing and may appeal to researchers with an interest in the theoretical problems underlying this kind of AI task.

6.4 Evaluation of MYCIN

In Chapter 2, I pointed out that the primary design consideration for MYCIN was that it be useful. The other five acceptability criteria discussed in that chapter (i.e., an educational capability, explanation, natural language understanding, knowledge acquisition, and modularity of knowledge) were justified in terms of their ability to enhance the system’s usefulness. It was also explained that a consultation program’s usefulness can be measured along three scales:

1. the need for the assistance which the program provides;
2. the reliability of the advice;
3. useability; i.e., the mechanics for accessing the machine and retrieving the desired information.

Evaluating the MYCIN system thus requires an assessment of the program’s performance along all three of these dimensions.

Section 1.4.2 addressed itself to the first of these three usefulness scales. There is ample evidence that antimicrobial agents are misused and that physicians would benefit from a mechanism that could improve the basis for antimicrobial therapy selection. An implied second component to this question is whether MYCIN is actually able to encourage more rational antimicrobial prescribing habits. Clearly this question cannot be answered until the program has been implemented for ongoing use in the clinical setting.

The reliability of MYCIN’s advice, on the other hand, can be assessed even before the program becomes generally available. In fact, as has previously been stated, we do not plan to implement MYCIN on the hospital wards until we are convinced that the program does give reliable advice for patients with bacteremia. We have therefore devised an experimental method for judging the system’s validity and have undertaken preliminary studies using this technique. Section 6.4.1 describes the first such study.

The third usefulness scale (i.e., the system’s useability) has been
considered throughout MYCIN’s development. The success of attempts to make the program easy-to-use cannot be rigorously evaluated, however, until MYCIN is generally available. At that time, we will be able to talk to physicians who have interacted with the system and to compile data indicating whether they consult MYCIN regularly or lose interest after one or two encounters.

Evaluation of MYCIN will therefore be a continuing process occurring in stages. The first phase involves validation of the program’s advice and will thus predate implementation. Subsequent stages will assess acceptability, clinical impact, and other questions that can be adequately answered only after MYCIN is generally available. In § 6.4.2 through § 6.4.6, I discuss some of these questions and our plans for analyzing them.

6.4.1 RELIABILITY OF MYCIN’S ADVICE

It is our belief that MYCIN should not be made available clinically until its advice has been shown to be reliable; a program that demonstrates its usefulness from the outset is more apt to gain the acceptance of the physicians for whom it is designed. Evaluation of MYCIN’s reliability thus requires a pre-implementation testing procedure with which we are currently involved. A semiformal study was undertaken in mid-1974 [Shortliffe, 1974b] when the program appeared to be performing well on the basis of rules provided by the clinical collaborators. Five infectious disease experts (not associated with the project) were asked to evaluate the program’s performance with fifteen cases of bacteremia selected without bias from current inpatients. The program’s principal problem was found to be a lack of sufficient rules to allow it to consider adequately the severity of a patient’s illness. Nevertheless, the experts approved MYCIN’s therapy recommendation in 72% of the 75 evaluations (5 experts reviewing 15 patients each). Differences of opinion regarding the best therapy as judged by the experts themselves were considerable however, and this lack of unanimity introduces some challenging problems to an evaluation study of this kind.

Since the experts have helped to identify errors and gaps in the program’s knowledge-base, additional rules are currently being added. This process of formal evaluation followed by new rule addition and re-evaluation will continue until the independent experts agree with MYCIN’s reasoning as much as they agree with each
Future Directions for MYCIN

other’s. We expect that the program will be ready for introduction on the wards once the experts approve its performance in at least 90% of cases.

6.4.2 MYCIN’S ACCEPTABILITY TO PHYSICIANS

Unless MYCIN is accepted by the physicians who must use it, its ability to give valid advice will be of little value. Chapter 2 emphasized those features of the system that were designed to heighten its acceptability. Once the system is generally available, however, new requirements may become evident. We must therefore implement ongoing mechanisms for identifying those aspects of the program that interfere with the willingness of physicians to use it.

One approach will be to keep a record of physicians who have tried the system and to interview them in order to assess their reactions. It is inevitable that MYCIN’s interactive capabilities will have to be constantly modified and improved as feedback from physician users is obtained. Another feedback mechanism will be to permit the physician to type in comments at any time during a consultation. Such remarks can be stored in the computer and regularly reviewed by MYCIN project members.

A second tactic is to identify those physicians who have never tried the system and to find out why. If they are simply unaware of the program’s existence, that failing can be easily rectified by an appropriate publicity campaign. If their failure to consult MYCIN results from a basic aversion to interacting with a computer, on the other hand, or if they have heard negative comments about the program from their colleagues, it is important to determine whether changes in the system or its mode of interaction will help to make it more attractive. Although physicians have been involved in the design of MYCIN from the outset, it is unlikely that all the concerns of potential users will have been taken into account. We must therefore be prepared to modify the program, or even radically to overhaul it, in an effort to maximize MYCIN’s use by those physicians who may need it when they prescribe for an infectious disease problem.

6.4.3 MYCIN’S IMPACT ON PRESCRIBING HABITS

A second important set of questions to be answered once the system is implemented involves its effect on physician prescribing
habits. This can be adequately assessed only if control data regarding current prescribing practices are obtained before MYCIN becomes available. It will then be possible to judge whether antimicrobials are used more appropriately after MYCIN has begun to exert its influence.

It will also be important to assess whether physicians who use MYCIN actually follow its advice. When they do not, we should find out why since that may help with the specification of missing decision rules. If, on the other hand, they reject MYCIN's advice and prescribe less appropriately, an attempt must be made to understand why MYCIN failed to influence them. For example, there may be problems with the Explanation System that prevent it from convincing the user that the program's reasoning is valid.

The educational impact of MYCIN can also be judged by monitoring prescribing habits before and after the system is available. It is possible that MYCIN will result in a new awareness of antibiotic prescribing habits throughout the hospital staff so that even physicians who have never used the program will prescribe more appropriately. Furthermore, clinicians who use the program extensively at first may grow to depend upon it less as they become more familiar with the important therapeutic considerations.

6.4.4 MYCIN'S IMPACT ON PATIENT CARE

Influencing physician prescribing habits is not a sufficient goal for MYCIN unless it also has demonstrably beneficial effects upon patient care. It will therefore be necessary to develop mechanisms for measuring MYCIN's effect on the quality of care for patients with bacterial infections.

A number of approaches are possible. One is merely to monitor the response of a patient's disease when he is treated with the regimen suggested by MYCIN. Not only may such monitoring provide evidence that MYCIN is suggesting appropriate therapy but, in cases where the patient does not respond as desired, it may also help identify inadequacies in the decision rules that have been given to MYCIN by experts.

Monitoring individual patients provides information that is more anecdotal than statistically significant, however. It may therefore be wise to gather data reflecting trends in length-of-stay for hospitalized
patients, incidence of adverse reactions to antimicrobial agents, or pharmacy costs to the patient. All these parameters may reflect beneficial effects of MYCIN that can be verified statistically.

6.4.5 (*) SPEED, EFFICIENCY, AND STORAGE REQUIREMENTS

Descriptions of MYCIN often lead to questions regarding the potential difficulty in implementing a completed system without the program proving too large and slow. The final answers concerning these issues will not be available until we get a better feel for how many new rules and system changes will be necessary before MYCIN can become an effective and acceptable clinical tool. We have devoted considerable thought and discussion, however, to the running time and storage requirements of a high performance consultation program such as the one we hope MYCIN will eventually become. Although economic considerations may eventually require that the program be translated for use on a small computer (see § 6.4.6), we are convinced that response time or computer storage limitations are unlikely to present difficulties in implementing a completed version of MYCIN under the present TENEX operating system [Myer, 1971]. Some of the considerations involved in this conclusion are:

Space: The TENEX system that we currently use allocates up to 256 thousand virtual words of memory (512 pages) to each user. Of the 490 pages that we currently use, approximately 320 pages are used by the INTERLISP system, which includes such features as the spelling corrector, CLISP (Conversational LISP), and the LISP compiler. Of the remaining 170 pages, approximately 100 pages (50 K) are for the compiled MYCIN program. The other 70 pages contain MYCIN’s rules, clinical parameters, knowledge tables, and working space. The current program appears to operate adequately within these space limitations. As noted in § 1.6.2, that lengthy sample consultation required approximately 20 minutes at a computer terminal, including the time devoted to the optional use of the Explanation System. Moreover, the following options are available to accommodate future growth of the system:

(1) Smaller LISP: Many INTERLISP features are useful for developing a new program but are not essential for running a performance system. For
example, the LISP compiler, LISP editor, and CLISP are all unnecessary for MYCIN’s purposes. In response to the demand of many INTERLISP users that the language dispense with certain features in return for increased memory availability, the language will soon have an “overlay” feature that will permit INTERLISP users to customize versions of LISP in accordance with their individual requirements. When implemented, the “overlay” capability will permit us to create a much smaller version of LISP containing only those features needed by MYCIN.

(2) Modular Programs: The three major components of the MYCIN system (Subprograms 1, 2, and 3; Figure 1-1) are currently loaded into core for every run of the program. However, this is not necessary. For a consultation session only Subprogram 1 needs to be used. At the end of an advice-giving session (or in response to the QA command, § 3.3.2-2), the Explanation System can be added to the Consultation System. The Rule-Acquisition System will not be used at all during standard consultations. Since Subprogram 3 depends upon the expert being able to run Subprogram 1 and 2 as well, however, space considerations may be most important during rule-acquisition sessions. The “overlay” feature mentioned above should alleviate some of these space problems by permitting the three subprograms to be loaded when needed and then deleted programmatically.

(3) The Rule Corpus: By far the fastest growing part of the system is the rule corpus. Although the rest of MYCIN is continually being modified, its size has not increased substantially for several months. Relative to the rest of the program, MYCIN’s 200 current rules take up only a small amount of space (16 pages = 8 K). Thus, we believe that the system can easily accommodate the many additional rules which we recognize will be needed.

(4) Recoding For Efficiency: In the initial stages of this work, less attention was paid to space considerations than to major design considerations. As we proceed further with development of the program, we expect to be able to recode parts to enable them to make more efficient use of working space and to take up less space themselves.

Running Time: Because MYCIN requires substantial interaction at the terminal, it is, to a large extent, input–output bound. However, at times the system becomes compute bound, such as when it must chain through a large number of rules that do not generate questions, or when it is garbage collecting the working space. Except for a few lapses during these compute bound activities, the program’s running time is currently acceptable. We are therefore developing ways to
future optimize our rule searching strategy (§ 3.7) and to reuse active core locations so that fewer garbage collections will occur.

The number of users in a time sharing environment is also a major consideration. To alleviate this potential problem once MYCIN is implemented on the wards, it is possible to arrange for changes to the scheduling algorithm during periods of peak use, and we can at least alert the physician to a potential slow-down when the number of other users is large. It is also worth noting that the times when consultants in infectious disease therapy are least apt to be available (i.e., late at night and on weekends) are precisely those periods when time-sharing systems are most apt to have a low number of users. Thus, the system becomes a particularly viable alternative to the human consultant when he is unavailable.

Since the efficiency of MYCIN is another important consideration, we have accomplished a substantial improvement in execution time by compiling our code for service use of the program. The INTERLISP block compiler may appropriately be used for portions of the code and will give us extra efficiency not attainable by compiling each function individually.

We believe that the present organization of the knowledge base makes for efficient processing of the set of rules. When the number of rules increases substantially, we expect that the present organization will continue to cope successfully for three reasons. First, the rules are divided by context-type so that potentially useful rules are eliminated from consideration if their classification is inappropriate for the context being examined. Second, the rules are linked together in such a way that determining the truth or falsity of the PREMISE of one rule does not require a search of all other rules. Finally, since we have devised a strategy for recognizing those branches of the reasoning network that have already been searched, new rules that reference clinical parameters with which the system is already familiar will not result in exponential growth of the search space.

6.4.6 COST OF MYCIN’S CONSULTATIONS

An important topic that has previously been ignored in this volume is the cost of a system like MYCIN. The present system was developed on a large computer (Digital Equipment Corporation
Mycin

PDP-10) that is seldom found in hospitals. Furthermore, the operating system and the INTERLISP language [Teitelman, 1974] are designed primarily for AI applications and are therefore mostly found in university or government research environments. Before MYCIN can become generally available outside the university environment, therefore, it will probably need to be rewritten for a computing system that is more accessible to those hospitals most in need of the program’s services. As a result, any attempt to evaluate the cost of a consultation with MYCIN would be premature at present. Research and development expenses naturally bear little resemblance to the costs that will be incurred once MYCIN is an ongoing service system on an in-hospital computer. INTERLISP has been a powerful development tool, but it is slow and demands more computing power than most hospitals can afford.

6.5 MYCIN and Shared Data Bases

Section 1.2.2-6 described Hospital Information Systems (HIS) and their potential for assisting with information handling chores in the clinical environment. Regardless of whether such systems are implemented as a single large installation, or as a set of integrated but independently developed submodules, they are characterized by large amounts of diverse patient data that can be shared among the system components.

Let us consider what MYCIN’s role might be in an HIS that contains up-to-date patient information in the following categories:

(1) chemistry laboratory data (including hematology)
(2) pharmacy data
(3) microbiology laboratory data
(4) clinical data traditionally found in the patient chart

It should be clear that most of the clinical parameters used by MYCIN may be classified in one of these categories. Thus, if MYCIN were a component in a comprehensive HIS and could reference the patient’s information from the above four data bases, several of the questions currently asked of the physician would no longer be necessary. For example, information regarding current and prior
cultures would be available from data base (3) and the patient's recent drug history could be found in data base (2). In fact, any piece of information currently classified as LABDATA (§ 3.2.3-2) would presumably be available from one of the four data bases. The user would therefore be asked to interact with MYCIN only for consideration of those non-LABDATA parameters for which the rule corpus was unable to infer values (Figure 3-8). This corresponds to the observation that ASK1 questions would no longer be necessary and that only ASK2 questions would need to be displayed for the physician (see Figure 3-9).

As was pointed out in § 3.3.2-1, however, one of the goals in the future development of MYCIN's knowledge-base is to acquire enough rules allowing the values of non-LABDATA parameters to be inferred so that ASK2 questions need no longer occur. One of the impediments to this goal has been the tendency for such rules to generate large numbers of highly specific questions that make MYCIN appear to be groping for ideas and that are thus annoying for the user. Consider, for example, the non-LABDATA parameter COMPROMISED that is a "yes-no" parameter indicating whether the patient is a compromised host. There are currently no rules for inferring the value of this parameter, so an ASK2 question is generated whenever FINDOUT tries to find its value (Figure 3-8). If MYCIN were to make the conclusion on its own, rather than to leave the decision up to the judgment of the user, the program would require a series of rules itemizing disease categories that suggest that a patient's immune response system is not functioning normally. Such rules would in turn generate a series of apparently groping questions such as "Does the patient have leukemia?," "Is the patient an alcoholic?," etc. If a series of questions regarding diagnoses could be answered via queries sent to other HIS data bases, however, the more basic rules regarding compromised-host status could be added to MYCIN's knowledge base without generating annoying questions for the physician.

The discussion of the preceding paragraphs indicates the way in which access to shared clinical data bases could reduce the number of questions asked of the physician by MYCIN. Since much of MYCIN's current time requirement is bound by the terminal-based interaction with the physician, an efficient linkage between MYCIN and other data bases might well decrease the time from sign-on until MYCIN's
recommendation becomes available. In the extreme case, one can imagine a user simply giving MYCIN the name of his patient and answering no additional questions. MYCIN would evaluate the patient on the basis of primitive data (LABDATA) obtainable directly from the clinical laboratory, microbiology, pharmacy, and medical record data bases. After a variable length of time (depending upon the complexity of the patient's infectious disease problem), a therapeutic recommendation would be printed by MYCIN and the physician would be able to use the Explanation System (Chapter 5) to query the program regarding the reasoning behind the suggested regimen.

A formally constituted HIS is not a prerequisite for the shared data base application of MYCIN just described. All that is really necessary is the up-to-date data bases plus communication links between the computers in which the information is stored. Stanford Hospital already has all four of the required data bases: pharmacy [Cohen, 1974], microbiology [Petralli, 1970], clinical chemistry [Sussman, unpublished], and medical records [Fries, 1972]. Unfortunately, all four systems were developed independently and currently operate on separate computers. Since all the programs would benefit from access to one another's patient data, however, communication links between the machines are being contemplated. As soon as these are available, we hope to connect MYCIN to the network and to develop the mechanisms for direct access to patient data in accordance with the model that I described above.

If the four clinical data bases are effectively linked, as is planned, another potential addition to MYCIN would be an ability to monitor a patient's response to the recommended therapy. In this way, it could perhaps acquire statistics that would enable it to alter its drug selection strategy or first-choice drugs. If this capability were implemented, it would resemble the kind of machine "learning" discussed at the beginning of § 6.3.

6.6 Prospective Monitoring of Prescribing Habits

Of all the issues currently involving American organized medicine, there is perhaps none more emotion-laden than the question of peer review. Known euphemistically as patient care appraisal,
quality-of-care assessment, or quality assurance, peer review has entered the political arena since a Social Security amendment was signed into law in 1972. Known as Public Law 92-603, the legislation requires that Professional Standards Review Organizations (PSRO) be set up to monitor medical practice, to identify problems, and to take steps to correct them. PSRO's are to be instituted locally in all parts of the country, and physician organizations initially have priority in establishing them.

Although physicians had begun to participate in peer review activities prior to passage of the new legislation, until recently emphasis has been on assessing those parameters of practice that are most easily measured. Thus utilization review committees and tissue review boards have traditionally taken on the primary peer review responsibilities. PL 92-603 has sparked new interest in peer review issues, however, both with regard to how review should be undertaken and whether the government should be able to interfere in an area that had previously been the concern solely of the medical practitioners themselves. Organized medicine has many reservations regarding PSRO [Watts, 1973], and the strengths and weaknesses of the legislation have been much analyzed [Welch, 1973].

As mentioned in § 1.4.6, it is my conviction that the primary reasons for physician opposition to peer review legislation result from the fact that medicine is one of the few professions in which individuals have traditionally been free from close observation and criticism. Legislation to promote government influence on medical care delivery, whether it be Medicare or PSRO, is thus met with widespread opposition and, in some cases, fear [Gottesman, 1972]. What is particularly worrisome to physicians is the potential for being punished when they make decisions that are judged by others to be mistakes.

Regardless of whether PSRO deserves opposition, the bill has been signed into law and is not apt to be repealed. It is therefore appropriate to look for ways to insure that the new peer review mechanisms will both accomplish the goals of the legislation and will be at least mildly acceptable to physicians. I therefore cite the following proposed criteria for acceptability of the developing peer review mechanisms:

(1) They should be able to judge questions of medical care, not merely parameters such as length-of-stay data;
(2) They should emphasize educational benefits rather than punitive actions when errors are noted;
(3) They should ideally inform the physician of a possible error before it is too late to rectify matters;
(4) They should encourage feedback from physicians regarding strengths and weaknesses of the approach.

The importance of the second point cannot be overstated. There has already been experience to indicate that patient care monitoring can be made acceptable to physicians if they are not led to believe that they will be punished when errors are observed [Alper, 1974].

With criteria such as those above in mind, authors have begun to suggest ways to choose peer review methods [Brook, 1973]. For several years there have been efforts to assess quality of care by reviewing patient charts [Fessel, 1972]. Medical audit of this variety is difficult, however, because the task is arduous, it requires a time-commitment from the reviewing physicians, and the criteria for judging care are, in general, ill-defined. One innovation has been the institution of departmental medical audit workshops at which physicians attempt to delineate what should be the criteria for quality care at their hospital. These criteria can then be used for assessment of care when medical records are reviewed.

The above discussion has been an attempt to lay the groundwork for justifying the claim that MYCIN provides a useful model for a peer review mechanism satisfying the cited acceptability criteria. I shall explain this model by describing an existing system and discussing how MYCIN could be adapted in a similar way.

The MEDIPHOR System [Cohen, 1974] was briefly mentioned in § 1.2.2-5. This is a large computer program developed at Stanford Medical School for the prospective control and study of drug interactions in hospitalized patients. Using a comprehensive and documented data base of drug interaction information, the system generates warnings to pharmacists, nursing personnel, and physicians when potentially interacting drug combinations have been prescribed. Drug profiles for patients are available to the system because it also serves as a label printing machine in the hospital pharmacy. Whenever a label is printed, the computer records the information in the patient’s drug profile. Thus, whenever a new drug is prescribed, the machine can use its drug interaction data base to search for interactions between the new prescription and drugs the patient is
Future Directions for MYCIN

already receiving. If a potential interaction is found, a warning is printed in the pharmacy and sent to the ward along with the drug. There the physician and nursing staff may consider the interaction information before the interacting drug is administered. If the physician decides to give the drug, he at least knows about the potential for adverse effects and is therefore careful to monitor the appropriate clinical parameters of the patient.

The MEDIPHOR system offers many of the advantages of the peer review acceptability criteria I described. Clearly it addresses itself to an important clinical practice question that is difficult to assess even by chart review. Furthermore, it points out potential problems before they occur and thus reveals its educational emphasis. Physicians are more apt to be defensive about their decisions if possible errors are not pointed out until two or three months after the incident. By that time, notification appears to be a scolding since it is too late for corrective action to be taken. Finally, a system like MEDIPHOR can also be used to accumulate the data necessary for judging trends in the quality of care, at least for the topic of drug interactions.

Suppose, now, that the various computer-based data banks at Stanford Hospital were joined by communication links as discussed in § 6.5. In that section I explained how MYCIN could provide consultations without asking questions of the physician so long as all pertinent data were available in one of the Stanford data bases. Under those circumstances, the physician seeking advice is needed only to initiate the consultation. Consider, then, the potential for initiating the consultation program not in response to a request from a physician seeking advice but instead whenever an antimicrobial agent is prescribed in the hospital pharmacy. The MEDIPHOR system could notify MYCIN regarding the patient, drug, and dose. MYCIN could then use its knowledge base to decide how it would treat the patient and whether the drug actually prescribed is appropriate. If a prescription were clearly inappropriate, MYCIN could send the relevant information back to MEDIPHOR and a warning could in turn be generated in the pharmacy. This warning would then be returned to the ward with the prescribed drug where the physician could consider MYCIN’s recommendations before deciding whether to administer the drug he had originally prescribed. The physician would, in effect, receive a consultation from MYCIN when he needed it rather than when he asked for it.
This approach to peer review provides an exciting potential for impacting the antimicrobial prescribing habits of physicians, and for monitoring other clinical practice questions as MYCIN-like knowledge bases are developed for additional problem areas. In this sense peer review may be considered as “covert consultation” in much the same sense that human consultations may be looked upon as “overt peer review.” This model for prospective monitoring of prescribing habits is particularly appealing because it satisfies our proposed acceptability criteria for a peer review mechanism.

This section concludes with an example of a situation in which the monitoring model we have described would have been highly useful. During early development of the MYCIN system, we reviewed several patient charts in an effort to identify decision rules needed by the program. In one such chart we found that a patient had been treated with streptomycin as a single agent to combat an organism which was known to be resistant to streptomycin \textit{in vitro}. Furthermore, the patient who was given the drug (which is toxic to the kidney) had chemistry laboratory values for BUN and creatinine indicating that he was in renal failure. In short, the streptomycin therapy was highly inappropriate. If MYCIN had been monitoring antimicrobial prescriptions in the hospital pharmacy, it would have automatically evaluated the streptomycin prescription. The lab values for BUN and creatinine would have been available from the clinical laboratory data base, and the microbiology data base would have revealed the organism’s resistance to the drug. MYCIN would therefore have concluded that the streptomycin was inappropriate and a warning would have been generated. It is possible, in turn, that the warning would have had a beneficial educational impact on the physician who made the improper therapeutic decision. As was discussed in §1.5.2, there is much evidence that this kind of inappropriate prescribing of an antibiotic is not an isolated incident, although the above example is, perhaps, somewhat extreme.

6.7 Educational Applications

As I have emphasized throughout this report, an ability to instruct the user was an important consideration during the design of MYCIN. We believe it is possible to learn a great deal simply by
Future Directions for MYCIN

asking MYCIN for consultative advice and taking advantage of the program's explanation capabilities. It is quite likely, in fact, that medical students in their clinical years will comprise a large percentage of MYCIN's regular users once it is available on the wards.

It would be possible, however, to adapt MYCIN so that its emphasis became primarily educational rather than consultative. This could be accomplished in a number of ways. In one scenario, MYCIN would present a sample patient to a student. The program would then judge the student's ability to ask important questions and to reach valid conclusions regarding both the identity of the organism(s) and the most appropriate therapeutic regimen. By comparing the student's questions and decisions to its own, MYCIN could infer inadequacies in the user's knowledge and enter into a tutorial discourse customized for the student. A similar instructional session might be generated even for actual patient cases provided by the student. Although there is great potential for this kind of educational use of MYCIN's knowledge base, we have no plans to pursue this application in the near future.

6.7 Other Applications of MYCIN Formalism

In § 3.8.3 I noted that one of the principal advantages of the MYCIN approach is its domain independent control structure. Attempts have also been made to preserve generality in Subprograms 2 and 3. We have not yet tested this claim with a second data base, however. As explained in Chapter 3, acquiring rules and defining parameters are such complex and time-consuming tasks that we have so far been unable to experiment with alternate clinical problem areas.

Our current plan is gradually to broaden MYCIN's knowledge base into other infectious disease topics (i.e., in addition to bacteremia). We feel it is important, however, eventually to test the approach in medical decision areas that have nothing to do with antimicrobial therapy. Not only will this assist in determining the generality of the MYCIN formalism, but it will also help us define which clinical problems are best suited for a rule-based system rather than for Bayesian or model-based approaches. As I have stated before, MYCIN's formalism seems to be most appropriate for applications in
MYCIN

which informal judgmental knowledge is the basis for decisions. If good statistical information is available or a problem is suited to physiological modeling, an alternate approach may be preferable. Until MYCIN is tested in new arenas, we will be unable to reach justifiable decisions regarding these issues.

It is also interesting to ask whether MYCIN's approach can be usefully applied to nonmedical problems. Although we have no current intention to investigate such questions ourselves, other AI researchers have begun to indicate an interest in pursuing this rule-based approach for nonmedical applications. Of particular relevance, of course, are those problems that can benefit from a technique for coding the heuristics of an individual.
Chapter 7

Conclusion

7.1 Summary

MYCIN is a large computer system developed (to the extent described in this text) over a two year period. The program's knowledge-base, and many aspects of system design, were contributed by collaborating physicians and computer scientists who met regularly throughout the two years. The project has recently expanded to include additional physicians and computer scientists who are contributing full-time efforts to the future expansion of MYCIN’s capabilities.

This chapter summarizes the material that has been presented in this text. In this section, I reiterate the clinical problem for which MYCIN is designed to offer advice. I then briefly review how the program attempts to solve the problem. Section 7.2 then summarizes the limitations of the MYCIN approach. Section 7.3 discusses MYCIN’s contribution to computer-based medical decision making, and I conclude in § 7.4 with consideration of the program’s contribution to the field of AI.

7.1.1 THE CLINICAL PROBLEM

The principal goal of the MYCIN project has been to devise a computer-based system for assisting and educating physicians who need advice about appropriate antimicrobial therapy. The basis of rational infectious disease therapy is identification of the offending micro-organisms. Accurate identification is important because drugs that are highly effective against certain bacteria are often useless
against others. The patient's clinical status and history, including such information as previous infections and treatment, provide valuable data to assist the physician with the identification task. However, bacteriological cultures that use specimens taken from the site of the patient's infection usually provide the most definitive identifying information.

Initial culture reports from a microbiology laboratory may become available within 12 hours from the time a clinical specimen is obtained from the patient. The information in these early reports often serves to classify the organism in general terms but does not permit precise identification. It may be clinically unwise to postpone therapy until identification of the infecting organism can be made with certainty, however—a process that usually requires 24 to 48 hours or longer. Thus, it is often necessary for the physician to estimate the range of possible organisms and to start appropriate treatment even before the laboratory is able to identify the offending organism and its antibiotic sensitivities.

As discussed in § 1.5.2, there is ample evidence that physicians often do not choose antimicrobial therapy wisely. Studies discussed in that chapter have shown that physicians may reach therapeutic decisions that differ significantly from those that would have been suggested by infectious disease experts. It is not uncommon for physicians to treat patients for whom experts believe no antimicrobial therapy is indicated. Furthermore, nonexperts sometimes choose a drug regimen designed to cover for all possibilities, prescribing either several drugs or one of the so-called “broad-spectrum” antibiotics, even though appropriate utilization of clinical clues might have led to a more rational (and often less toxic) therapy. Since professional resources are often overburdened in today's hospitals, a computer-based system that could serve effectively in a consultation role to the nonexpert—and gain his respect—would be highly useful. MYCIN has been designed to provide readily accessible advice and instruction that will help bridge this gap between practicing physicians and experts in infectious disease therapy.

MYCIN has also been developed with an awareness of the current lack of acceptance of computer-assisted decision making by the medical profession. We have attempted to analyze the reasons for the common opposition to such programs and to endow MYCIN with
Conclusion

characteristics that will make it more acceptable. These points are discussed in detail in § 1.4.6 and in Chapter 2.

7.1.2 THE SOLUTION

The MYCIN system is offered as a solution to the two problems described in the previous section; i.e., it attempts to give good advice regarding antimicrobial selection and it attempts to do so in a way which will make the system acceptable to physicians. In order to solve both these problems, MYCIN has been designed with three principal capabilities in mind:

(1) an ability to give good advice;
(2) an ability to explain the basis for its advice;
(3) an ability to acquire new knowledge easily so that its advice can improve over time.

Thus, MYCIN consists of three subprograms, each of which addresses itself to one of these three goals.

Subprogram 1 is a Consultation System. This component uses information about a patient, plus MYCIN’s knowledge of bacterial infections, in order to decide (a) whether the patient needs to be treated, (b) the likely identity of offending organisms, (c) the possible drugs for use against these organisms, and (d) the best drug or drugs for the particular patient in light of his current clinical condition. Information about the patient is entered by the physician in response to questions asked by MYCIN. Each question asks for the value of some clinical parameter used by the program when it makes decisions. If all such values were known, the patient’s clinical status would be fully characterized. MYCIN’s task, however, is efficiently to select those of the clinical parameters (currently 65 in number) that are needed for adequate consideration of a given patient. The program’s current knowledge is stored in 200 isolated decision rules, each of which is invoked only if the program has reason to believe it may be useful. This efficient use of system knowledge is accomplished by a goal-oriented control structure which dynamically creates a reasoning network appropriate for the clinical problem under consideration. The details of Subprogram 1 are the subject of Chapter 3.
Subprogram 2 is an Explanation System. This component attempts to answer questions from the user both during and after a consultation session. Furthermore, it attempts to do so in terms that will convince the physician that it reaches decisions in much the same way that he does. The user may ask MYCIN to explain the reason for a question during the consultation or may demand explanations of decisions that the program has reached. In an effort to make such explanations easy to obtain, even by a novice user, Subprogram 2 has been given a limited ability to understand simple English. In addition, its responses to questions are expressed in English and require no knowledge of MYCIN’s internal representation or control structure in order to be understood. The details of Subprogram 2 are described in Chapter 5.

Subprogram 3 is a Rule-Acquisition System designed for use by experts in infectious disease therapy. The capabilities of this system component are currently incomplete, but it is possible for an expert to teach MYCIN certain simple rules which are then incorporated into the system’s knowledge base for use in future consultations. An expert is encouraged to use Subprograms 1 and 2 in an effort to identify problems with MYCIN’s knowledge of infectious disease therapy. Subprogram 3 then permits him to enter new rules or to modify old ones which he has found to be inadequate. The rule-acquisition procedure, like Subprogram 2, attempts to understand knowledge statements expressed in English so that the expert need not learn a computer language nor details of MYCIN’s implementation. Subprogram 3 is the subject of § 6.3.

7.2 Limitations of MYCIN’s Approach

One of MYCIN’s principal limitations is shared by all clinical consultation programs: it requires that the physician take the initiative in asking for an interactive session. Despite attention that may be paid to “human engineering” issues during the development of such programs, physicians seldom choose to use computers for tasks they feel they can do themselves [Croft, 1972; Startsman, 1972]. Evidence that the medical profession may be failing to perform a job optimally, such as I presented in § 1.5.2, may not be sufficiently compelling for the individual practitioner to seek out a program’s
advice. Programs which can monitor physician prescribing habits automatically and generate warnings when appropriate [Cohen, 1974] may thus be more likely to influence physicians and the general quality of patient care.

The MYCIN approach is also limited by its complex control structure and representation scheme. These are probably unnecessary for medical inference tasks in which uncertainty and incomplete knowledge are uncommon occurrences. Until the MYCIN formalism has been tested on other medical decision problems, however, it will be difficult to predict what proportion might be handled more directly by other techniques. Furthermore, the efficiency advantages of approaches relying more heavily on statistical theory must be weighed against the importance of the natural mechanism for explanation and knowledge modification that may be achieved through MYCIN's flexible representation scheme.

Another limitation results because MYCIN currently requires more computing power and memory (a Digital Equipment Corporation PDP-10 with 256K of core) than is reasonable to expect to find in most hospital computing environments. Although the PDP-10 has been a powerful research machine for development of the MYCIN system, we are hopeful that the program may eventually be translated and simplified for smaller machines that may more reasonably be purchased or shared by hospitals. At present the system functions solely in a research environment, and long-range cost estimates are therefore not feasible.

MYCIN's explanation capabilities are limited somewhat by the current state of the art in computer-based processing of natural language. Although the program can understand simple, straightforward questions (Chapter 5), automated language understanding has not progressed enough to allow any program to handle unrestricted discourse.

Finally, it should be noted that MYCIN's reasoning mechanism must necessarily become more complex as the number of rules (i.e., amount of knowledge) increases. The present program exhaustively searches all rules that seem to be useful and does not distinguish those that are more useful than others. We have identified several heuristics for limiting the search space for a consultation (§ 3.7), and these must be implemented as MYCIN's knowledge grows or else the problem of excessive use of computing time will become overriding.
MYCIN

7.3 Contribution to Computer-Based Medical Decision Making

MYCIN has several novel attributes that distinguish it from other programs for medical decision making. Foremost among these is its ability to reason with informal judgmental knowledge acquired from experts. Although the system makes no attempt explicitly to model the psychological processes of a clinical decision maker, its modular decision rules and the certainty factor quantification scheme permit a physician's intuitions to be coded without major difficulty. Thus MYCIN's decisions need not depend upon the diagnostic algorithms, physiologic models, nor the statistical analyses that pervade much of the field (§ 1.3). The MYCIN formalism is therefore potentially applicable to decision making in the large number of clinical problem areas for which pathophysiology is poorly understood and statistical data are incomplete or nonexistent.

It should be noted that the MYCIN approach does not rule out applications for which reliable data become available. The formal certainty factor definitions and combining functions permit probabilistic information and judgmental knowledge to be used in unison. Furthermore, extensions to MYCIN may permit causal links to be coded in rule form so that the present control structure need not be modified. Although MYCIN may not provide the "best" solution for decision making in every clinical problem area, it may well serve as a useful adjunct to alternative techniques in most medical decision making applications.

Another important contribution of MYCIN's approach is its ability to reach decisions based upon whatever information is available at the time of the consultation. As is true for human consultants, MYCIN gives more reliable advice as more comprehensive information becomes available. Explicit decision trees or decisions based upon clinical algorithms tend to require pieces of information in a fixed order; if a datum is unavailable, the physician must wait for the appropriate test result before completing the consultation session. Of course, there are times when so little information is available that MYCIN cannot reach a reasonable decision. In general, however, MYCIN makes the best decision it can on the basis of current data and the user is encouraged to return for more definitive advice as further information becomes available. In a problem area
such as the treatment of infectious disease, interim decisions while awaiting further data are often in the best interests of the acutely ill patient.

Avoiding explicit decision trees has provided other advantages besides an ability to operate solely on the basis of current information. Most important among these is MYCIN's ability to incorporate new knowledge without explicitly being told how or when it will be useful. The program's control structure for dynamic reasoning (Chapter 3) automatically utilizes any rule-based knowledge that appears to be relevant. Storing knowledge in rules has also facilitated an ability to explain why questions are asked and to justify the basis for the program's therapeutic recommendations.

Finally, MYCIN has been designed to be more than merely an interesting theoretical approach to medical decision making in this therapeutic problem area. From the outset we have stressed the goal of eventually implementing the program for ongoing use by physicians. We have sought to understand why such programs have met resistance in the past, and we have in turn implemented a number of features, including a comprehensive explanation capability, designed to heighten MYCIN's acceptability to physicians. Although the program is not yet sufficiently knowledgeable for ongoing clinical use (§ 6.4.1), physicians who have used the system have uniformly indicated that they believe the program can become sufficiently reliable and will hence be used by the clinicians for whom it has been designed.

7.4 Contribution to AI

MYCIN's mechanisms for representing and utilizing judgmental knowledge also heighten its interest for computer scientists working in AI. Unlike formal problem-solving systems based on axiomatic knowledge, MYCIN suggests an approach for modeling the kinds of inexact reasoning that typify many real-world problems. AI researchers have recognized the need for some way to combine the attributes of decision theory with those of machine problem-solving [Feldman, 1974], and MYCIN provides what is perhaps the first general approach to this problem. Certainty factors are potentially applicable to a number of AI application areas. For example, conversations with
AI researchers have revealed that tasks such as identifying objects in machine vision or phonemes in speech understanding are typified by the kind of indecision that CF's are designed to handle.

Although neither MYCIN's goal-oriented control structure nor its dependence upon rule-based knowledge is unique (see Chapter 3), no other AI system has used its knowledge in quite the same way. As I have emphasized, MYCIN's formalism is domain independent and thus may prove useful for AI researchers who wish to automate other tasks that are dependent upon the heuristics of individuals. Furthermore, the use of rules with $CF=1$, or with a certainty factor derived from reliable statistical data, provides a mechanism for coding theorems, real-world data, and definitional information. This formal knowledge may then be used simultaneously with the informal knowledge that is representative of the intuitive inexact reasoning that typifies much of human problem-solving.

MYCIN has also been developed with more attention to human engineering than is typical of much of the AI field. The goal has been to develop mechanisms for interacting with medical professionals who are not only unfamiliar with AI but have often never used computers before. MYCIN's rules have therefore served as a highly useful representation scheme since they can be individually retrieved in order to explain why questions have been asked or to justify aspects of the program's advice. As AI applications for use by scientists and other individuals become more common, MYCIN may well suggest some useful guidelines for interactions with novice computer users.

Another lesson to be learned from MYCIN is that a single programmer, working full-time for two years with a powerful interactive language such as INTERLISP, can create an AI program that serves a useful purpose. Observers often bemoan the current state of the art in AI, asserting that it will be years before machines can perform problem-solving tasks at a level approximating that of humans. MYCIN has shown, however, that if researchers are willing to accept the current limitations of the AI field, and to select real-world goals that are compatible with those limitations, a useful system can be developed using techniques for representation and control that would not have been available if it were not for prior work in AI. MYCIN's question-answering (QA) skills are an example of this point. The techniques used for natural language understanding are
dependent upon several simplifying assumptions that ignore syntax, semantics, and the psychology of language. The last two of these are perhaps the principal barriers to further AI progress in the field of linguistics (Chapter 5). The limited QA capabilities that result, however, are in general satisfactory for the application area in which they are to be used. Although it would clearly be preferable if the program could participate in free form discourse, MYCIN has shown that a useful interim solution can be developed once the current limitations of the field have been accepted.

Finally, MYCIN has contributed to the AI field by providing evidence that suggests that current AI techniques may be adequate for assisting professionals with an important real-world problem. There has been a tendency for theoretical AI work to concentrate on tasks that are often described as “toy problems.” Although such problems are generally nontrivial and AI researchers can themselves appreciate the challenges involved, the relative paucity of AI programs that deal with real-world tasks has not always benefitted the image of the field. Although MYCIN’s effectiveness as a clinical tool has not yet been fully demonstrated, the preliminary evaluations mentioned in § 6.4 make us optimistic about its future. We are therefore pleased to be able to offer MYCIN as an example of a way in which current AI technology can potentially contribute to the betterment of public health through improved care for patients with infections.
References


MYCIN


References

chemistry.” In Machine Intelligence 5 (B. Meltzer and D. Michie, eds.),

programming study of theory formation in science.” Proc. 2nd Interna­
tional Joint Conference on AI, Imperial College, London (1971).

theory formation: data interpretation and rule formation.” In Machine
Intelligence 7 (B. Meltzer and D. Michie, eds.), Edinburgh University
Press (1972).

molecules: rule formation on non-homogeneous classes of objects.” AI
Memo 215, Stanford Artificial Intelligence Laboratory, Stanford Uni­

computer-based medical record system.” Meth. Inform. Med. 12, 137–
142 (1973).


19 (April 1974).


Carnap, R. (1962), “The aim of inductive logic.” In Logic, Methodology, and
Philosophy of Science (E. Nagel, P. Suppes, and A. Tarski, eds.).

Chang, C.L., and Lee, R.C. (1973), Symbolic Logic and Mechanical Theorem

based system for prospective detection and prevention of drug interac­

interactions.” Proc. MEDINFO IFIP Conference, Stockholm, Sweden
(August 1974).

algorithm on the data base of a human belief structure.” AI Memo 94,
MYCIN

Stanford Artificial Intelligence Project, Stanford University, Stanford,

Intelligence 2, 1–25 (1971).

indistinguishability tests for the validation of a computer simulation of

for the recognition of natural language dialogue expressions.” AI Memo
234, Stanford Artificial Intelligence Project, Stanford University, Stan­

Coles, L.S. (1974), “Categorical bibliography of literature in the field of robot-
ics.” Tech Report #88, Stanford Research Institute, AI Center, Menlo

Collen, M.F. et al. (1964), “Automated multiphasic screening and diagnosis.”

screening.” In Computers In Biomedical Research (R.W. Stacy and B.D.

Collen, M.F. (1966), “Periodic health examination using an automated multitest


processes—the problem oriented medical record.” The New Physician,
626–630 (October 1973).

Computers and Medicine (1973a), The EMI Scanner, Vol. II, No. 5 (September/
October 1973).

Computers and Medicine (1973b), MIS-1 After A Full Year’s Operation: Pros

Computerworld (1973), “MD’s at odds over hospital MIS.” (June 6 1973).

grammed console: an aid to radiation treatment planning.” Proc. 8th
IBM Medical Symposium, 179–186 (1967).

Cox, J. et al. (1972), “Digital analysis of the electroencephalogram, the blood
pressure wave, and the electrocardiogram.” Proc. of the IEEE, Vol. 60,
#10, 1137 (1972).


References

Davis, R. (1976), Ph.D. Dissertation, Computer Science Department, Stanford University, Stanford, Calif. (in preparation).


References


medical history data using a real-time computer system.” Proc. of the Annual Conf. on Engineering and Biology 10, 39.3 (November 1968).
References


Mycin


References


MYCIN


References


Resztak, K.E. and Williams, R.B. (1972), “A review of antibiotic therapy in


References


MYCIN


References


## Index

<table>
<thead>
<tr>
<th>Term</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abductive logic</td>
<td>29, 30</td>
</tr>
<tr>
<td>Accounting</td>
<td>4</td>
</tr>
<tr>
<td>Analog signals</td>
<td>5</td>
</tr>
<tr>
<td>$\text{AND}$</td>
<td>86, 87, 106ff, 123</td>
</tr>
<tr>
<td>Antibiotics</td>
<td>36–44, 217, 219–221, 229–235</td>
</tr>
<tr>
<td>misuse of</td>
<td>41–44</td>
</tr>
<tr>
<td>Antimicrobials</td>
<td>See Antibiotics</td>
</tr>
<tr>
<td>Attitudes towards computers</td>
<td>1–3, 22, 64, 234, 236</td>
</tr>
<tr>
<td>Attribute-object-value triple</td>
<td>93, 94</td>
</tr>
<tr>
<td>Automated programming</td>
<td>15, 216</td>
</tr>
<tr>
<td>Backus Normal Form (BNF)</td>
<td>86</td>
</tr>
<tr>
<td>Barnett, G. O.</td>
<td>11, 25, 159</td>
</tr>
<tr>
<td>Bayes' Theorem</td>
<td>25, 30, 68, 80, 155, 159ff</td>
</tr>
<tr>
<td>threshold</td>
<td>102</td>
</tr>
<tr>
<td>Certainty factors</td>
<td>88, 98–101, 130–133, 152ff, 171ff, 207ff, 238–241</td>
</tr>
<tr>
<td>characteristics</td>
<td>171, 172</td>
</tr>
<tr>
<td>combining functions</td>
<td>101, 132, 179, 180, 186, 187</td>
</tr>
<tr>
<td>defining criteria</td>
<td>178, 181, 182</td>
</tr>
<tr>
<td>effect on rule translation</td>
<td>113, 132</td>
</tr>
<tr>
<td>MB</td>
<td>169ff</td>
</tr>
<tr>
<td>MD</td>
<td>169ff</td>
</tr>
<tr>
<td>Chemistry laboratory systems</td>
<td>9, 10, 224</td>
</tr>
<tr>
<td>Choice</td>
<td>168</td>
</tr>
<tr>
<td>Clinical decision making</td>
<td>12, 20–36, 159, 238–239</td>
</tr>
<tr>
<td>Clinical parameter(s)</td>
<td>93ff, 216</td>
</tr>
<tr>
<td>Cohen, S. N.</td>
<td>10, 226, 228, 237</td>
</tr>
<tr>
<td>Colby, K. M.</td>
<td>82, 114</td>
</tr>
<tr>
<td>Common-sense knowledge</td>
<td>18</td>
</tr>
<tr>
<td>Computational linguistics</td>
<td>See Natural language processing.</td>
</tr>
<tr>
<td>Computer-aided instruction</td>
<td>(CAI), 12, 13, 63, 67, 77, 208, 230</td>
</tr>
<tr>
<td>Computers, attitudes towards</td>
<td>1–3, 22, 64, 234, 236</td>
</tr>
<tr>
<td>Concept formation</td>
<td>28, 114</td>
</tr>
<tr>
<td>Confidentiality</td>
<td>151</td>
</tr>
<tr>
<td>Confirmation</td>
<td>167ff</td>
</tr>
<tr>
<td>Consultation programs</td>
<td>21, 63ff</td>
</tr>
<tr>
<td>CAI</td>
<td>See Computer-aided instruction (CAI)</td>
</tr>
<tr>
<td>Carnap, R.</td>
<td>167, 168</td>
</tr>
<tr>
<td>Causal networks</td>
<td>31, 32</td>
</tr>
</tbody>
</table>

(See also MYCIN, Subprogram 1)
Context(s), 89ff, 137ff, 154, 216, 223
Cost of computing, 3, 223
Croft, D. J., 22, 64, 236

Decision theory, 25–28, 239
Decision trees, 70, 156, 239
DENDRAL, 84ff, 114ff
Diagnosis, 23ff
Disbelief, 169ff
Disconfirmation. See Confirmation.
Drug interactions, 10, 228
Drug reactions, adverse, 10

Education
  medical, 33, 35
  public, 42
Educational programs. See Computer-aided instruction.
Edwards, W., 162, 184
EKG/EEG analysis, 6
Evaluation, 65 (See also MYCIN, evaluation)
Evidential strength. See Confirmation.
Explanation capabilities, 28, 67, 75, 81, 146 (See also MYCIN, Subprogram 2)

Feigenbaum, E. A., 84
FINDOUT, 123ff, 149ff, 157, 209, 215, 225
Fries, J., 9, 23, 226
Functions. See Predicates.
Fuzzy sets, 168, 184

Games, 14, 83ff, 114, 208
Gini, G., 30
Ginsberg, A. S., 27
Graphics, 5
Green, C. C., 81, 115
Greenes, R. A., 8, 23

Harré, R., 167, 168, 172, 182
Heuristics, 19, 83, 115, 152, 208, 232, 237, 240
Helmer, O., 160
Hempel, C. G., 167, 168
Hewitt, C., 81, 116

HIS. See Hospital Information System (HIS).
History taking, 7, 26
Hospital administration, 3, 11
Hospital Information Systems (HIS), 10–12, 224
Human engineering, 65, 66, 77, 127ff, 140, 186, 219, 225, 236, 240
Human problem-solving, 17, 240

Interactive capabilities. See Human engineering.
INTERLISP, 20, 221–224, 240

Knowledge-acquisition. See MYCIN, Subprogram 3.
Knowledge tables, 109
Kulikowski, C. A., 26, 27, 31, 32, 63

Learning, machine, 69, 208
Linguistics. See Natural language.

Lists, simple, 108

Machine learning. See Learning, machine.
Means–ends analysis, 115
Medical audit. See Peer review.
Medical computing, legal aspects, 2, 34, 36
Medical economics, 3
Medical education, 33, 35
Medical records, 7–9
  problem-oriented, 8
Medical students, 2, 119, 231
MEDIPHOR System, 10, 226, 228, 237
Microbiology lab systems, 10, 224
Modeling, 31, 32
Modularity, knowledge, 70, 71, 91, 146, 155, 238–241

MONITOR, 121ff, 151ff, 157
Monkey-and-bananas problem, 18, 19
Multi-phasic health testing (MPHT), 7

MYCIN.

CF's. See Certainty factors.
clinical parameter(s), 93ff, 216
context(s), 89ff, 137ff, 154, 216, 223
control structure, 119–127
data base, 45–47
design criteria, 64–71, 79, 203, 217, 235, 239
dictionary, 74, 108, 131, 216
domain independence, 157, 240
dynamic data, 45–47, 133, 149, 198, 200–202
evaluation, 150, 217–224, 241
goal rule, 121
growth, 215, 221–223, 237
knowledge base. See MYCIN, rules.

name, 37

rules, 45–47, 72, 86ff, 161, 164–167, 184, 207–217, 222, 223, 238
acquisition. See MYCIN, Sub-program 3.
retrieval, 197–200
self-referencing, 134
translation to English, 112, 113
sample consultation, 47–61
selection of therapy, 142–148, 233–236

Subprogram 1 (Consultation System) 45–47, 73, 79–158, 197, 207, 222, 235

Subprogram 2 (Explanation System), 45–47, 74, 121, 195–204, 207, 220–222, 226, 236, 240

Subprogram 3 (Rule-Acquisition System) 45–47, 69, 72, 202, 207–217, 222, 236

P-function. See Probability.
Paradox of ravens, 167, 175, 180, 188–192
Pathology laboratory systems, 10
Patient Data Table, 148ff
Patient monitoring, 5
Peer review, 34, 36, 226–230
Petralli, J., 10, 226
Pharmacokinetics, 24
Pharmacy systems, 10, 224, 228
Philosophy of science, 166–168
PLANNER, 81, 116ff, 140, 153
POMR. See Medical records, problem-oriented.
Pople, H., 29, 30
Popper, K. R., 171, 172
Predicate calculus, 82, 93
Predicates, 102, 103, 110, 216
Preventive medicine, 6
Probability, 25–28, 98, 159ff
Problem-solving, human, 17, 240
machine, 18, 19, 239
Professional standards review organizations. See PSRO.
Prognosis, 23, 25
PSRO, 227, 228
Psychology, 17, 30, 31
Public education, 42

QA3, 81, 115
QA-module. See MYCIN, Sub-program 2.
Quality assessment. See Peer review.
Question-answering. See Explanation capabilities.

Reasoning, goal-oriented, 121, 155
Reasoning loops, 135ff
Reasoning network, 124ff, 155, 205
Reliability of programs, 65
Representation of knowledge, 18, 29
Rescher, N., 160
Resolution principle, 82, 115
Robotics, 15, 16
Rule-acquisition. See MYCIN, Sub-program 3.
MYCIN

Rules. See MYCIN, rules.
Rutstein, D. D., 1

SA rules. See Situation-action rules.
SAIL, 93
Salmon, W. C., 167, 168, 182
Samuel, A. L., 14, 208
Scene analysis, 16, 240
Schwartz, W. B., 24, 27, 33, 160
Shannon entropy formula, 27
Situation-action rules, 84, 114ff
Speech understanding, 16, 17, 240
Stanford University Hospital, 8–10, 143, 206, 226, 228
State-transition networks, 30
Suppes, P., 12, 189
Surprise, logic of, 168

Tables, knowledge, 109
TALLY, 106ff, 135
TENEX, 221, 224
Theorem proving, 14, 15
Theorem(s) antecedent, 118ff, 153

consequent, 116ff.
Theory formation. See Concept formation.
Time-oriented databank, 8
Tissue review. See Peer review.
Törnebohm, H., 168, 169, 171, 175, 176, 182
Treatment planning, 23, 24, 40, 41
Turing, A. M., 13
Turing Indistinguishability Test, 13, 17

Unity path, 215
Utilization review. See Peer review.

Vision, machine, 16, 240

Warner, H. R., 6, 25, 26, 63, 159, 162
Waterman, D. A., 83ff, 114, 208
Weed, L. L., 8
Werner, G., 29
Winston, P. H., 208
Wortman, P. M., 30, 31, 63