

Report 79-28
Stanford -- KSL

Steps Toward Mechanizing Discovery.
Bruce G. Buchanan,
Oct 1979

[Scientific DataLink](#)

card 1 of 1

STANFORD HEURISTIC PROGRAMMING PROJECT

OCTOBER 1979

MEMO HPP-79-28

STEPS TOWARD MECHANIZING DISCOVERY

BY

BRUCE G. BUCHANAN

COMPUTER SCIENCE DEPARTMENT

STANFORD UNIVERSITY

STANFORD, CALIFORNIA 94305

*PRESENTED AT THE PITTSBURGH UNIVERSITY CONFERENCE ON THE LOGIC OF
DIAGNOSIS, OCTOBER, 1973.

Steps Toward Mechanizing Discovery

Bruce G. Buchanan
Computer Science Department
Stanford University

presented at the
Pittsburgh University Conference on the Logic of Diagnosis
October, 1978

1 Introduction

This is the 30th anniversary of the publication of Prof. Hempel's landmark in the philosophy of science, "Studies in the Logic of Explanation". Because this paper is about computer programs that generate explanations, my debt to Prof. Hempel will be obvious.¹ However, insofar as I wish to use the term 'discovery' to cover the activity of finding explanations, I know that Prof. Hempel will not entirely agree with these ideas about mechanizing the activity.

The purpose of this paper is to elaborate a very simple idea: that discovery in science and medicine can be profitably viewed as systematic exclusion of hypotheses. That is, hypotheses that explain empirical data can be found systematically by methods that can be implemented in computer programs. The conditions under which this view makes sense are an important part of the elaboration. Two necessary conditions are that the space of relevant hypotheses is definable and that there exist criteria of rejection and acceptability. Because the space of hypotheses is immense for most interesting problems, it is also desirable that there exist criteria for guiding a systematic search. This idea was voiced recently by Linus Pauling, in an interview about his creative work:

So what I'm saying is, it's important to have a big background of knowledge. Also to do a lot of thinking. Probably part of the secret of being successful in a field involving discovery is the sort of judgment that keeps you from working in the wrong direction. a student once asked me, 'Dr. Pauling, how do you go about having good ideas?' and I answered, 'You have a lot of ideas and you throw away the bad ones.' [13]

¹ In addition, I owe a considerable debt to colleagues and friends at Stanford who helped shape these ideas, Profs. E.A. Feigenbaum and J. Lederberg in particular.

Kenneth Schaffner has argued[18] that scientific discovery and justification have essentially the same logic, in the sense that the forms of inference and the scientific norms are the same in both cases. His carefully reasoned study of a major discovery in molecular biology leads him to the conclusion that:

Essentially the same forms of inference and general types of considerations or factors can be used in searching for a new hypothesis, i.e., in generating it, as can be used to assess and criticize or defend it after the hypothesis is found.

Consistently with Schaffner's conclusion, this paper argues that many of the same criteria used to judge a hypothesis can be used as constraints on the hypothesis generator. Moreover, the decision to use information during generation rather than testing is based on extra-logical considerations such as cost, convenience and efficiency. The feasibility of implementing these ideas in a computer program is demonstrated in the second half of this paper by describing existing programs that formulate hypotheses in organic chemistry and molecular genetics.

1.1 Computer Programs as Models

We turn to computer programs as illustrations because their methods are necessarily explicit (although not always clear). We are not proposing that the traditional dream of an infallible method for investigating nature scientifically will be realized through computer programming. It is unlikely that a machine will ever be programmed to do induction in the same straight-forward way and with the same guarantees that it can be programmed to add a column of figures, for instance. However, a program can reason symbolically using inexact methods in problem areas for which infallible methods are missing. There are numerous examples in which numerical algorithms fail to provide satisfactory explanations in science and medicine. For solutions to these kinds of problems, we turn to a branch of computer science known as heuristic programming.

Heuristic programs differ from other computer programs in the extent of the guarantees they impart to their conclusions. A heuristic program, as distinguished from an algorithm, has been characterized as "a process which may solve a given problem, but offers no guarantees of doing so." [15] Polya describes heuristics in a similar way:

Heuristic reasoning is reasoning not regarded as final and strict, but as provisional and plausible only, whose purpose is to discover the solution to the present problem. [16]

2 Heuristic Search as a Method of Discovery

2.1 Searching for a Hypothesis

In general, the problem for a discovery method is twofold: (i) to choose a language, L, in which to express hypotheses, explaining data in a scientific domain, and (ii) to choose a satisfactory sentence of L which explains the data. In paradigm revision, to use Kuhn's terms, the first half of the problem is crucial, for the choice of the language establishes boundaries on the factual content of the paradigm. Choosing to speak of light as travelling (Toulmin's example, [22]) determines in large measure the kinds of questions we ask about light and the kinds of answers the paradigm will furnish.

On the other hand, choosing a language is not part of the problem of finding an explanation in normal science or clinical medicine, for the language in which hypotheses are expressed is just the language of the current paradigm. That is, once a paradigm is established, scientists routinely describe their work within that language; only when description and explanation within the language of normal science fails does a scientist again face the problem of choosing a new language in which to express hypotheses. At this point we will not discuss the problems of choosing a language, although I will return to it at the end.

The second half of the problem -- choosing one of the sentences in L to serve as a hypothesis -- is a problem which a logic of discovery might help solve. The problem addressed here is to find efficient methods for picking out sentences of L which are most likely to succeed as hypotheses in a given case. a grossly inefficient method is to generate the sentences of L, say in lexicographic order, and test each one. It is clear that scientists do not resort to an enumeration and one-by-one trial of sentences of L, for we would expect no progress in science with such inefficiency. For similar reasons, exhaustive enumeration of hypotheses by a computer is out of the question for all but trivial problems.

As an alternative we could consider discovery to be merely successful guessing, as is often suggested, and program a machine to generate hypotheses randomly -- perhaps restrained within the correct subject area. It could test each random hypothesis against the criteria of success and stop when a hypothesis met those criteria. Although some inquiring minds may work in this random manner, it hardly recommends itself as a rational method. In addition, neither random search nor exhaustive enumeration carries any sense of terminating when there is no sentence in L that meets the criteria.

2.2 Heuristic Search

The method of systematic exploration sketched above is very like the old method of induction by elimination. Solutions to problems can

be found and proved correct, in this view, by enumerating possible solutions and refuting all but one. Obviously the method is used frequently in contemporary science and medicine, and is as powerful as the generator of possibilities. However according to Laudan, the method of proof by eliminative induction, advanced by Bacon and Hooke, was dropped after Condillac, Newton and LeSage argued successfully that it is impossible to enumerate exhaustively all the hypotheses that could conceivably explain a set of events [12]. The force of the refutation lies in the open-endedness of the language of science: within a fixed language the method reduces to modus tollens.

The computational method known as heuristic search is in some sense a revival of those old ideas on induction by elimination, but with machine methods of generation and search substituted for exhaustive enumeration. Instead of enumerating all sentences in the language of science and trying each one in turn, a computer program can use heuristics enabling it to discard large classes of hypotheses and search only a small number of remaining possibilities. Heuristic search is the best known and most widely used method underneath symbolic reasoning programs. It has been applied, with some variations, in problem areas ranging from chess to chemistry and from mathematics to molecular genetics. The method depends essentially on heuristics, or rules of thumb, that guide the search for hypotheses and set the criteria for plausibility of hypotheses.

The key ideas behind heuristic search are:²

- a) implicitly define the complete space of hypotheses within language L;
- b) define a stepwise generator of progressively more precise hypotheses within this space;
- c) during generation, eliminate incorrect or implausible classes of hypotheses;
- d) after generation, test the generated candidates in order to eliminate some and rank the rest.

There are many ill-defined terms in this characterization. Instead of discussing alternative definitions, however, I will illustrate the use of heuristic search in three computer programs that formulate explanatory hypotheses in science.

2.3 Heuristic DENDRAL

Two sets of computer programs developed at Stanford University use heuristic search for hypothesis formation in organic chemistry and thus constitute an empirical demonstration of the feasibility of the

² For more information on heuristic search and other symbolic computation techniques, (See [23].)

ideas advanced here. These programs are called Heuristic DENDRAL and Meta-DENDRAL. They are closely related but work at two different levels of scientific activity. Another program that reasons about molecular genetics will be discussed later.

Both programs are organized around the heuristic search method in which relevant hypothesis spaces are searched systematically under heuristic constraints. The fundamental assumption is that discovery can be viewed as systematic exclusion of hypotheses from a mechanical hypothesis generator. In all interesting problems the hypothesis space is so large that the heuristics must be powerful enough to exclude large classes of hypotheses before they are generated and the generator must be flexible enough to exclude classes of hypotheses under many different descriptions.

This methodology requires (1) a constructive definition of possible hypotheses that can be used as a generator of items in the hypothesis space (steps a and b above) and (2) criteria for selecting and rejecting classes of hypotheses (steps c and d above). The discussions of the two programs focus on these two requirements.

All of these programs are predicated on the assumption that finding scientific hypotheses can be treated in much the same way as other cognitive problems and does not require unique solution methods. Problem solving in science, including hypothesis formation, has many of the same information processing characteristics as problem solving in other areas.[17] The Heuristic DENDRAL program [3]³ is designed to aid organic chemists in determining the molecular structure of organic compounds from empirical data. The observed data for an unknown compound are explained in the sense that once the correct molecular structure is hypothesized the data can be predicted from the deductive application of a theory (of the analysis technique) to the hypothesized chemical structure.

Inferring structural information from empirical data generally requires a thorough understanding of the instrument or technique that produced the data.⁴ For example, the reasons for noisy, ambiguous, or missing data, as well as the origins of data points, must be factored into the interpretation. The value of an analytical procedure in chemistry lies in its ability to provide information about the composition and structure of the unknown compound. For example, a chemist wants to know about the presence or absence of oxygen in the unknown, the number of oxygen atoms doubly bonded to carbon atoms, the sizes of rings, and so on, in order to determine the topological structure of the molecule.

Structure Elucidation with Constraints from Mass Spectrometry

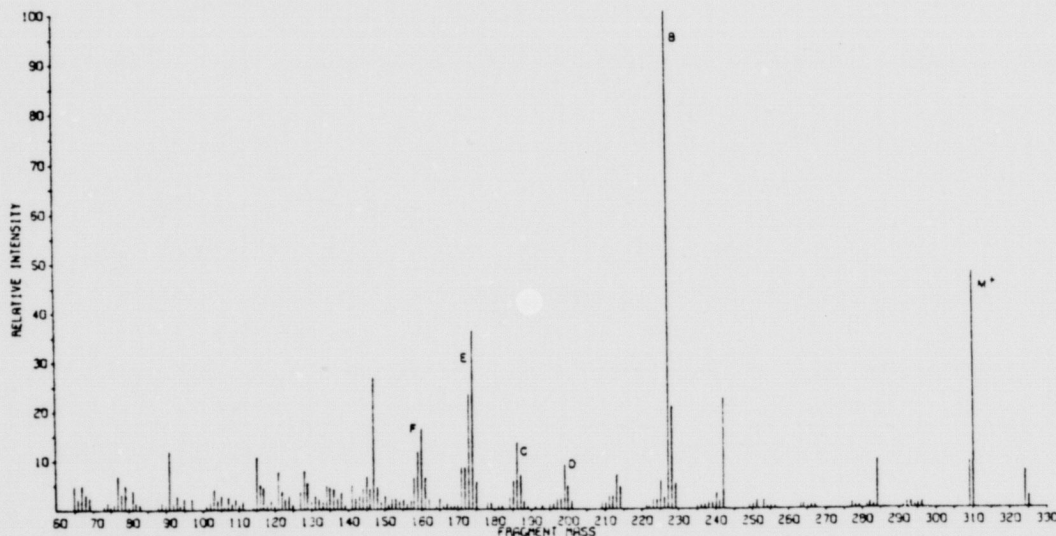
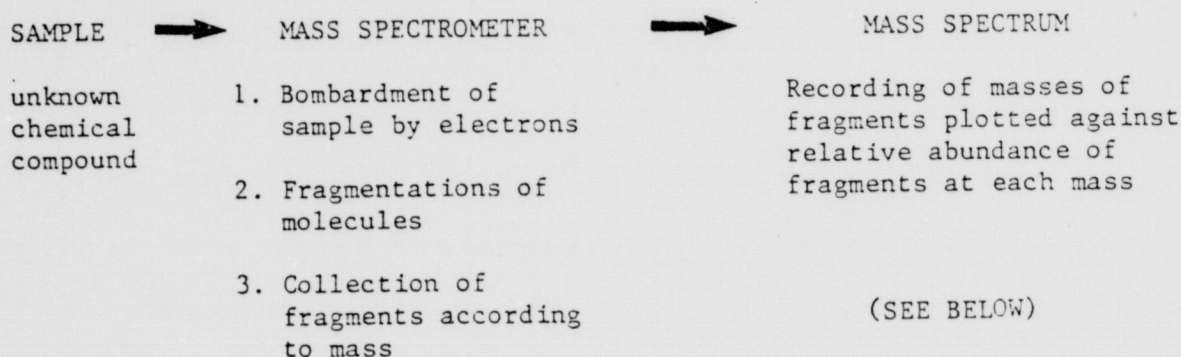
³ Design issues for this program - and similar programs - are discussed in [9].

⁴ It is common practice to consult libraries of data associated with known compounds in order to find the best library entry that explains one's present set of data. However, the DENDRAL programs are meant to discover explanations even when the data have not been previously recorded in a library.

Parts of the Heuristic DENDRAL program have been highly tuned to work with experimental data from an analytical instrument known as a mass spectrometer. (See [6] for technical details.) Mass spectrometry is a new and still developing analytic technique. It is not ordinarily the only analytic technique used by chemists, but is one of a broad array, including nuclear magnetic resonance (NMR), infrared (IR), ultraviolet (UV), and "wet chemistry" analysis. It is particularly useful when the quantity of the sample to be identified is very small, for mass spectrometry requires only micrograms of sample.

A mass spectrometer bombards the chemical sample with electrons, causing fragmentations and rearrangements of the molecules. Charged fragments are collected by mass. The data from the instrument, recorded in a histogram known as a mass spectrum, show the masses of charged fragments plotted against the relative abundance of the fragments at a mass. Although the mass spectrum for each molecule may be nearly unique, it is still a difficult task to infer the molecular structure from the 100-300 data points in the mass spectrum because not only does a spectrum contain "noise peaks" and overlapping peaks originating from many parts of the molecule, but the theory of mass spectrometry is not complete.

Figure 1. Data to be Explained



This is mostly a task within routine science, in Kuhn's term again; no new terms or relations need to be postulated to account for the data. Occasionally new techniques are invented which allow identification of more classes of compounds. But these new techniques are quite readily assimilated by the routine scientists making the identifications.

Generating Hypotheses

The DENDRAL generator of molecular structures (named CONGEN for Constrained Generator) is the heart of the whole program. The problem description it starts with is a list of chemical atoms (including the number of atoms of each type) together with constraints on the ways groups of atoms can and cannot be associated. The language in which hypotheses are expressed, and generated, is the so-called "ball and stick" language of chemical atoms and bonds. Substructural units which the data indicate are necessary parts of the explanation are grouped together as "superatoms". Other constraints for good and bad arrangements of atoms (and superatoms) are specified on lists known as GOODLIST and BADLIST, respectively.⁵

CONGEN produces a complete and non-redundant list of molecular structures containing exactly the specified atoms and satisfying the specified constraints. Of the enormously large numbers of chemical structures with a specified atomic composition, the generator avoids constructing any that fail to contain the superatoms and prunes whole classes of structures (before instantiation) with respect to GOODLIST, BADLIST and the other constraints.

Although the generator is mathematically complex, the underlying principle is to break the large generation problem into discrete steps and build progressively larger hypotheses at each step. Thus at various stages of the generation process different heuristics can prune whole classes of partially specified hypotheses.

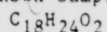
⁵ Note that the source of these constraints has deliberately been left unspecified. In other publications [19] we describe a problem solving program whose task it is to infer constraints from mass spectrometry data. However, CONGEN does not know where the constraints come from, and most of the time chemists are much better at making these inferences from their data than the program is anyway.

CONGEN produces a complete and non-redundant list of molecular structures containing exactly the specified atoms and satisfying the specified constraints. Of the enormously large numbers of chemical structures with a specified atomic composition, the generator avoids constructing any that fail to contain the superatoms and prunes whole classes of structures (before instantiation) with respect to GOODLIST, BADLIST and the other constraints.

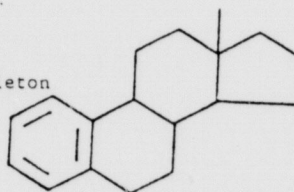
Although the generator is mathematically complex, the underlying principle is to break the large generation problem into discrete steps and build progressively larger hypotheses at each step. Thus at various stages of the generation process different heuristics can prune whole classes of partially specified hypotheses.

Figure 2. Simplified Structure Elucidation Example

1. chemical composition of unknown sample:



2. Specify constraints: (a) estrogenic steroid skeleton



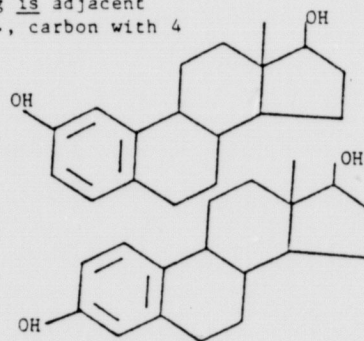
- (b) 2 hydroxyl groups [-OH]
 (c) one hydroxyl on aromatic ring
 (d) one hydroxyl on five-membered ring

3. Generate structural hypotheses with additional constraints that
 (3a) the hydroxyl in the aromatic ring is not adjacent to a ring juncture
 (3b) the hydroxyl in the 5-membered ring is adjacent to the quaternary carbon atom (i.e., carbon with 4 non-hydrogen neighbors)

4. Generated candidates =

H1: 2-hydroxy-estradiol

H2: 3-hydroxy-estradiol



5. Rank hypotheses using the heuristic that the preferred position for hydroxyls in estrogenic steroids is that shown in H2.

most plausible hypothesis = H2

left unspecified. In other publications [19] we describe a problem solving program whose task it is to infer constraints from mass spectrometry data. However, CONGEN does not know where the constraints come from, and most of the time chemists are much better at making these inferences from their data than the program is anyway.

The unconstrained algorithm has been proved to produce all possible chemical graphs with a specified composition, without duplication.[1] Since there are hundreds of possibilities for six-atom structures, thousands for 7-8 atoms, millions and tens of millions for 15-20 atoms, the generator cannot profitably examine every possible explanation. The need for constraints is obvious, as is the need for a machine to carry out the generation and systematic exclusion of hypotheses.

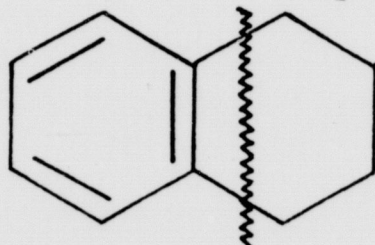
CONGEN has been used as an aid to discovery by many chemists around the world. It is not without limitations, which are largely due to restrictions imposed by the ball and stick language.⁶ However, the constraints are well enough understood to allow chemists to use this tool effectively [3].

2.4 Meta-DENDRAL

The Meta-DENDRAL program[4] is designed to aid chemists find and explain regularities in a collection of data. The hypotheses it formulates are general rules that explain the observed regularities. This activity is close to what most of us would call the "essence" of science, for it involves classifying phenomena, noticing regularities and discovering universal generalizations about them. Although the most revolutionary discoveries involve postulating new theoretical entities (that is, developing a new theory) finding general rules is also a creative activity within an existing theory. It is at this level of scientific activity that the Meta-DENDRAL program operates. It does not postulate new terms but tries to find new regularities and explain them with rules written in the pre-defined vocabulary.

Figure 3. Example of a General Rule
Explaining Part of the Collection of Data

R1: If there is an aromatic ring joined to another ring, break the second ring at the ring junctures.



⁶ For example, the program knows nothing of polymers, it knows nothing of bond angles or lengths, and it is limited in practice to structures containing at most 50-100 chemical atoms within 10-12 building block superatoms.

The "observations" the program starts with are empirical data collected on several known chemical compounds. The data have been limited to data from one commonly used analytical instrument, a mass spectrometer (described above). The regularities that the program can find are common modes of fragmentation and rearrangement of the known structures in the mass spectrometer. And the explanations of those regularities that it can discover are general rules causally relating "essential" features of the molecules to their fragmentation behavior in the instrument (see Fig. 4).

Figure 4: Explaining Part of the Collection of Data
Example of a General Rule R1: If there is an aromatic ring joined to another ring, break the second ring at the ring junctures.

The fixed vocabulary of the theory is known to the program, as are criteria for allowable extensions to the theory. A rule describes a general configuration of atoms around bonds that make those bonds less stable under high energy electron bombardment of the molecules in the mass spectrometer. The program emulates many of the reasoning aspects of manual approaches to rule discovery. It reasons symbolically, using a modest amount of chemical knowledge. It decides which data points are important and looks for fragmentation processes that will explain them. Then, as a chemist does, the program tests and modifies the rules.

Finding Regularities

The program first looks for regularities in the mass spectrometric behavior of the given molecules.⁷ Each data point is associated with the possible fragmentation processes that could have produced that datum within the mass spectrometer. Then the fragmentation processes for which there is substantial evidence are postulated as empirical regularities in the observed behavior of the given molecules. This part of the program is called INTSUM, for interpretation and summary of the data.

The INTSUM task is very much what Whewell called "colligation of facts" in which observations are gathered and interpreted [7]. Whewell noticed that interpretation of facts depends on an observation language whose legitimacy is supported by a theory, that fact gathering and theory building go hand in hand.

For each molecule in a given set, INTSUM first produces the plausible bond cleavage processes which might occur, i.e., breaks and combinations of breaks, with and without transfer of hydrogens and other neutral species. These processes are associated with specific bonds in a portion of molecular structure, or skeleton, that is chosen

⁷ The regularities in the data points themselves can be found by statistical pattern recognition programs and used to help interpret new data [24]. However, this is not the approach used here since we are looking for relations that provide some deeper insights into the theory of mass spectrometry.

because it is common to the molecules in a given set. Then INTSUM examines the spectra of the molecules looking for evidence (spectral peaks) for each process. INTSUM gives explanations of spectral peaks for each molecule and then produces a summary showing the total evidence associated with each possible process.

Generating Explanatory Rules

The program next looks for rules to explain the regularities by searching a space of possible rules in much the same way that Heuristic DENDRAL searches its space of possible molecular structures to find an explanation of observed data. Whole classes of rules are avoided if all of their instances would fail to meet syntactic and evidential criteria of rulehood, thus constraining the search to a small fraction of the total space.

The form of any rule is a conditional sentence relating the subgraph environments of chemical bonds with fragmentations involving these bonds. For example, one well known rule (which the program also rediscovered) is that the presence of a nitrogen atom in the environment of a carbon atom causes the C-N bond to cleave in the mass spectrometer, regardless of the other attachments of the atoms. The rule in this example was found by the program after noticing that in a dozen nitrogen-containing molecules there was substantial evidence associated with the cleavage of every C-N bond. Although many of the data points could be explained in other ways this rule was preferred for reasons of simplicity and total evidential support. Occam's Razor argues for dropping the redundant explanatory rules.

We use heuristic search to examine possible generalizations of the environments, where each generalization can be interpreted as a possible rule when coupled with information about the bond cleavages and transfers of hydrogens or other neutral species. Conceptually, the generator begins with the most general subgraph description R^*R (where R is any unspecified atom and where the asterisk is used to indicate the bond cleaved with the charged fragment written to the left of the asterisk). Then it generates refined descriptions by successively specifying one additional feature, in all possible ways. Currently we describe bond environments in terms of a topological, or connectivity, model of structure within the ball and stick language. We specify atom type, degree of substitution, number of hydrogen neighbors or number of multiple bonds at any atom place. Other features of atoms can also be used if they are computable from the connectivity graph model of a molecule, e.g., ring size or chain length. But this is the current vocabulary in terms of which rules are written. The most useful rules lie somewhere between the overly-general environment R^*R and the overly-specific complete bond environment descriptions for a whole molecule.

The program continues to make rules more specific until it finds a daughter rule that is (a) specific enough to focus on one type of process, (and to avoid many potential counterexamples) and (b) general enough to account for more than a few special cases.

Modifying Rules

The last phase of the program (RULEMOD) evaluates the plausible rules generated by RULEGEN and modifies them by making them more general or more specific. Its task is to analyze the validity of predictions made by the rules on the original set of molecules, modify the subgraph descriptions of the rules to improve the accuracy of their predictions, merge similar rules, and finally select a subset of the modified rules. RULEMOD will typically turn out a set of 8 to 12 rules covering substantially the same data points as an original set of approximately 25 to 100 rules, but with fewer incorrect predictions. The modification step is necessary after generation because, for efficiency reasons, not all the criteria of plausibility were used as constraints during generation. The local evaluation in RULEGEN has not discovered that different RULEGEN pathways may yield rules which are different but explain many of the same data points. Thus there is often a high degree of overlap in those rules. Rules may have many counterexamples because this expensive test is also not made during generation. The scoring function used to discard implausible rules and rank the plausible ones captures the following intuitions:

a) the score should reflect the strength of evidence, i.e., it should be proportional to average peak intensity;

b) data points (peaks) that are uniquely explained by a rule should count more than peaks that can be explained by two or more rules;

c) negative evidence (peaks predicted and not found) should count heavily against a rule.

d) Since the number of molecules in the set remains the same during rule formation and we insure that every rule applies to a minimum number of molecules (in our case half the molecules), the score for purposes of selection need not explicitly factor in the sample size. When we want to compare sets of rules formed from different sets of molecules, however, it will be necessary to weight the scores of rules by the number of molecules considered.

We have shown that the Meta-DENDRAL program is capable of rationalizing the mass spectral fragmentations of sets of molecules in terms of substructural features of the molecules. On known test cases, aliphatic amines and estrogenic steroids, the Meta-DENDRAL program rediscovered the well-characterized fragmentation processes reported in the literature. On the three classes of ketoandrostanes for which no general class rules have been reported, the mono-, di-, and triketoandrostanes, the program found general rules describing the mass spectrometric behavior of those classes.[5] This is the first instance we know of in which general symbolic laws have been discovered by a computer program and published in a scientific journal.

Programs with knowledge of the scientific domain can provide "smart" assistance to working scientists, as shown by the reasoned suggestions this program makes about extensions to mass spectrometry theory. We are aware that the program is not discovering a new

framework for mass spectrometry theory; to the contrary, it comes close to capturing in a computer program all we could discern by observing human problem-solving behavior. It is intended to relieve chemists of the need to exercise their personal heuristics over and over again, and thus we believe it can aid chemists in suggesting more novel extensions to existing theory.

2.5 MOLGEN

The MOLGEN computer program⁸ is a third program we are building at Stanford. It is still under development and is considerably newer than DENDRAL and Meta-DENDRAL, so it will be described only briefly. The intent of MOLGEN is to put as much knowledge of molecular biology into a computer program as we need in order to provide intelligent assistance to working scientists. We are focusing on the knowledge used by faculty and staff in the Department of Genetics at Stanford in designing and debugging experiments with DNA. The experiments are of two major classes: structural analysis of DNA segments, and synthesis of new DNA segments with specified features.

The techniques for manipulating DNA segments must be described to the program - there are enzymes that cut and join as well as chemical and physical techniques. The knowledge base is stored as a hierarchy of concepts with many features associated with each concept. For example, under the concept CUTTING TECHNIQUES will be found NUCLEASE (i.e., an enzyme that breaks chemical bonds in the DNA backbone), and under that will be found ENDONUCLEASE and EXONUCLEASE. One of the site-specific ENDONUCLEASES is Eco-R1, a particular enzyme isolated from E.coli that cleaves DNA in specific places. Associated with the name of the particular enzyme will be a definition of its recognition site and a specification of experimental conditions under which the enzyme will work, such as the optimal pH range.

In designing experiments, the program is strongly driven by the goal as specified by the geneticist. The key to its operation is systematic generation of plausible experimental plans. It starts from an abstract description of the goal and successively refines the description in all plausible ways. For example, a synthesis experiment often involves implanting one DNA segment inside a larger DNA molecule. One of the abstract concepts involved in implantation is cutting; then one way of cutting is by using a nuclease, and so on until one of the particular steps is described as mixing the DNA with the cutting enzyme ECO-R1 under specified conditions.

The program may be able to propose novel experiments for the following three reasons:

- 1) The knowledge base may eventually contain specific knowledge about different techniques that are not known by the same human scientist.

⁸ MOLGEN has been implemented largely by M. Stefik and P. Friedland. See [14] for details.

2) The combinatorics of applying one technique after another get to be more than humans cope with.

3) The program will be free to explore alternatives to the published methods that human scientists come to rely on heavily. The stereotypical experimental steps are handy mnemonic devices for humans, but they may also keep a person from seeing something novel. While MOLGEN's proposals themselves will be built from known techniques, its sense of "combinatorial play" will be unrestrained by stereotypes.

MOLGEN uses a large knowledge base of facts, associations and heuristics to define the space of plausible sequences of experimental steps. Although the total space is immense, the reasoning program is guided by knowledge of plausible abstract plans and plausible refinements of those. The final hypothesis, then, is a detailed description of an experiment that achieves the goal, within a specified language of laboratory objects and procedures. The program is beginning to be used by molecular biologists, but without conclusive results at this time.

2.6 Limitations of The Heuristic Search Method

The major limitation of the heuristic search method in any domain is the necessity of finding (or inventing) a generator of possible solutions. In the case of molecular structures, finding the generating algorithm took many years. Lederberg's notational algorithm for unringed graph structures (described in [2]) was mapped into a generating algorithm with little difficulty, but the symmetries of cyclic graphs complicated the generation problem immensely. Not until considerable mathematical expertise had been focused on the problem was a generator invented that carried guarantees of complete and non-redundant generation. In the rule formation domain that means that we needed to invent a program that generates possible rules. That, in turn, required a strict definition of the allowable forms of the rules and a definition of the allowable primitive terms that add content to the form. The representation we have found for expressing rules is fixed for any one run, but can, at least, be modified or extended manually between runs.

A second major limitation on heuristic search is the necessity of finding heuristics, or rules of thumb, that guide the generator and constrain it from producing all syntactically allowable hypotheses. For rule generation it was necessary to find heuristics that steer the generator toward the small number of interesting rules and away from the very large number of uninteresting rules. The problem is that it is difficult to find these guiding principles. In addition, putting confidence in the heuristics requires an act of faith. Once that step is made, however, there is often the temptation to put too much faith in the heuristics, and forget that the solutions were found in the context of a large number of assumptions. For example, one might tend

to forget the criteria for data filtering, or the restrictions on how complex the hypotheses were allowed to become, or the criteria for excluding implausible hypotheses, or the limitations of the conceptual framework.

On the other hand, the strengths of the method should be equally obvious:

1) We can guarantee that all implausible hypotheses have been excluded and the remaining hypotheses are all and only the plausible ones. (The guarantee holds only within the conceptual frame defined for the program but is good for the life of that framework.)

2) We can encode a scientist's incomplete and uncertain knowledge of the domain to help define the criteria of plausibility. The programs can be given these items in many levels of certitude - from those most central to the paradigm to those that are little more than ephemeral intuitions.

3 Summary and Conclusions

The traditional problem of finding an effective method for formulating true hypotheses that best explain phenomena has been transformed into finding heuristic methods that generate plausible explanations. The problem of giving rules for producing true scientific statements has been replaced by the problem of finding efficient heuristic rules for culling the reasonable candidates for an explanation from an appropriate set of possible candidates.

In the most creative heights of science, hypothesis formation is farthest from the "reach of method" as Whewell says. But within the comfort of an established scientific theory, paradigm, or conceptual scheme, hypothesis formation usually does not involve the introduction of new concepts. The concepts are given and the task of a logic of suggestion is to show how hypotheses should be formulated in terms of these concepts. Depending on the purposes at hand, and in part upon the science, the hypothesis may either explain a puzzling phenomenon (or set of phenomena) or describe objects and events within the scope of the science.

The problems with formulating this kind of logic of discovery are both difficult and numerous. Before any methods, heuristic or otherwise, can be given for "discovering" explanations or regularities, precise criteria of success for the logic must be formulated. When such criteria are clarified and refined for the specific science considered, then the methods could be said to succeed when they produce hypotheses which meet the criteria. The methods themselves will also be difficult to formulate in specific instances because of the difficulties in understanding the problem, representing

the space of possible solutions, dividing the task into subproblems, and planning a solution, to mention the outstanding ones.

To a modest degree, the Heuristic DENDRAL, Meta-DENDRAL and MOLGEN programs capture many of the notions of a logic of discovery. They are more systematic and less random than we have come to expect of creative guessers in science, but their methods are also more teachable and their results more reproducible.

References

1. H. Brown and L. Masinter, "An Algorithm for the Construction of the Graphs of Organic Molecules," Discrete Mathematics, 8, 227, (1974).
2. E.G. Buchanan, A.M. Duffield, and A.V. Robertson. "An Application of Artificial Intelligence to the Interpretation of Mass Spectra," in G.W.A. Milne (ed.), Mass Spectrometry Techniques and Applications, New York: Wiley, 1971.
3. B.G. Buchanan and E.A. Feigenbaum, "DENDRAL and Meta-DENDRAL: Their Applications Dimension," Artificial Intelligence, 11, 5, (1978).
4. B.G. Buchanan and Tom Mitchell, "Model-Directed Learning of Production Rules," in D.A. Waterman and F. Hayes-Roth (eds.), Pattern-Directed Inference Systems, New York: Academic Press, 1978.
5. B.G. Buchanan, D.H. Smith, W.C. White, R.J. Gritter, E.A. Feigenbaum, J. Lederberg, and Carl Djerassi, "Application of Artificial Intelligence for Chemical Inference XXII. Automatic Rule Formation in Mass Spectrometry by Means of the Meta-DENDRAL Program," Journal of the American Chemical Society, 98, 6168, (1976).
6. H. Budzikiewicz, C. Djerassi, and D. H. Williams, Mass Spectrometry of Organic Compounds, San Francisco: Holden-Day, Inc., 1967.
7. Robert E. Butts, "Whewell's Logic of Induction", in Giere & Westfall (eds.), Foundations of Scientific Method: The Nineteenth Century. Bloomington: Indiana Univ. Press, 1973.
8. R. E. Carhart, D. H. Smith, H. Brown and C. Djerassi, J. Amer. Chem. Soc., 97, 5755 (1975).
9. C.W. Churchman and B.G. Buchanan, "On the Design of Inductive Systems: Some Philosophical Problems," British Journal for the Philosophy of Science, 20, 311, (1969).

10. E.A. Feigenbaum, and J. Feldman (eds.). Computers and Thought. New York: McGraw-Hill, 1963.
11. R.N. Giere, and R.S. Westfall, (eds.), Foundations of Scientific Method: The Nineteenth Century. Bloomington: Indiana University Press, 1973.
12. Laurens Laudan,, "Peirce and the Trivialization of the Self-Correcting Thesis", in Giere & Westfall (eds.).
13. G. Leeson, "A Chat with Linus Pauling," California Living (section of the San Francisco Sunday Examiner and Chronicle), July 17, 1977.
14. N. Martin, P. Friedland, J. King, and M.J. Stefik, "Knowledge Base Management for Experiment Planning," Proceedings of the Fifth IJCAI, MIT, August 1977, p.882.
15. Newell, A. "GPS, a Program that Simulates Human Thought." in Feigenbaum & Feldman (eds).
16. Polya, G. How to Solve It. Princeton, New Jersey: Princeton University Press, 1948.
17. Simon, Herbert A. "Does Scientific Discovery Have a Logic?", Philosophy of Science 40 471-480, (1973).
18. K. Schaffner, "Logic of Discovery and Justification in Regulatory Genetics," Stud. Hist. Phil. Sci. 4 349-385, (1974).
19. D. H. Smith, B. G. Buchanan, R. S. Engelmore, a M. Duffield, A. Yeo, E. A. Feigenbaum, J. Lederberg, C. Djerassi, J. Amer. Chem. Soc., 94 , 5962 (1972);
20. D. H. Smith, B. G. Buchanan, R. S. Engelmore, H. Adlercreutz, C. Djerassi, J. Amer. Chem. Soc. 95, 6078 (1973).
21. D.H.Smith, B.G.Buchanan, W.C.White, E.A.Feigenbaum, C.Djerassi, J.Lederberg, Tetrahedron, 38, 3117 (1973).
22. S.E. Toulmin, Foresight and Understanding. Bloomington: Indiana University Press, 1961.

23. P.H. Winston, Artificial Intelligence, Reading, Mass.: Addison-Wesley Publishing Company, 1977.
24. G. S. Zander and P. C. Jurs, Anal. Chem., 47, 1562 (1975).

Copyright © 1985 by KSL and
Comtex Scientific Corporation

FILMED FROM BEST AVAILABLE COPY