BULLETIN (New Series) OF THE
AMERICAN MATHEMATICAL SOCIETY
Volume 11, Number 2, October 1984

COMPUTER MODELING OF SCIENTIFIC AND MATHEMATICAL DISCOVERY PROCESSES[1]

BY HERBERT A. SIMON[2]

Just forty years ago, in Chicago, John von Neumann delivered the eighteenth Josiah Willard Gibbs Lecture. His topic was the ergodic theorem and statistical mechanics. Within a month of that occasion, he and his colleague, Oscar Morgenstern, published their seminal work, *The theory of games and economic behavior*, applying discrete mathematics to problems of bargaining and competition in economic and social affairs.

Neither lecture nor book gave any hint of von Neumann's new preoccupation with the electronic digital computer, which had begun in the summer of that same year, 1944. We were then standing on the very brink of the computer era, and within a decade some of us found, in the new computer programming languages, a novel mathematical formalism that seemed ideally suited to building and testing theories of human decision making and problem solving. Some of us became so intrigued with the power and possibilities of these new languages that we largely adopted them in place of the older formalisms of applied mathematics—derived from analysis, discrete mathematics, topology, and logic—as our principal tools of theory formulation, especially in cognitive psychology.

Programming languages as formalisms. In this paper I should like to show how the new programming languages can be used to express theories of human problem solving; and I shall take as my domain of examples, theories about problem solving that require applying mathematics to empirical phenomena. Hence, the paper will have two intertwined, and perhaps incestuous, themes: the first concerns the processes for applying mathematics; the second concerns computer programming languages as mathematical formalisms for expressing theories of such processes. So we shall deal with mathematics applied to the theory of how mathematics is done.

1980 *Mathematics Subject Classification*. Primary 00A25.

[1] The Josiah Willard Gibbs Lecture presented at the 90th Annual Meeting of the American Mathematical Society in Louisville, January 25, 1984 under the title *Computer programs that model the process of scientific and mathematical discovery*; received by the editors April 23, 1984.

[2] The publications on which a large part of this Gibbs Lecture are based are the product of the BACON project, in which Patrick W. Langley, Gary L. Bradshaw, Jan Zytkow and I have been engaged over the past five years. I want to acknowledge the partnership of these colleagues, while absolving them of responsibility for the particular form that this exposition of our work takes.

0273-0979/84 $1.00 + $.25 per page

By "processes for applying mathematics" I mean the psychological processes used in finding and proving theorems, in discovering mathematical formulas, and in manipulating mathematical expressions. The processes of mathematics are usually thought of as "deductive". A mathematical proof is certainly a deductive object: each step in the proof is derived from the previous steps and axioms by application of a (usually small) set of rules of inference. If a proof is written out in detail, its validity can be checked rigorously, step by step, by application of a mechanical algorithm. *Finding* a proof, on the other hand, is an inductive process, a process of heuristic search through a (usually immense) space of possible paths. Finding a proof, at least in its more impressive manifestations, is usually thought to call for "creative" processes, that are only with difficulty (or not at all) reducible to a mechanical algorithm. Finding a mathematical formula to fit data also is an inductive process that requires heuristic search, and might likewise appear not to be reducible to an algorithm.

The first aim of the paper is to discuss the progress that has been made in the past forty years toward understanding this process of heuristic search and the nature of the steps that are often called "intuitive" or "creative". The progress has depended heavily on our gradually growing ability to write computer programs that capture the heuristic search process, and thus provide theories of that process in the new mathematical formalism of programming languages.

The second aim of the paper is to illustrate the formalism itself, and to show, thereby, one important and very distinctive way in which computer programs can be employed as tools of applied mathematics. Gibbs reminded his Yale faculty colleagues (and us) that "mathematics *is* a language". And I wish to add to his reminder that "a computer programming language *is* a mathematical language".

Formally, a computer program is a set of difference equations. It defines the state and output of the computer at time T as a function of its state and input at time $T - 1$. The time increment is the basic instruction cycle time of the system. As a set of difference equations, a program is a familiar mathematical object. In other respects, it exhibits a number of novelties. The most important of these novelties is that the arguments in computer instructions need not be real or complex numbers, but may be symbols of a wide variety of types, including words and expressions of natural language and symbol structures that represent geometrical configurations. As we have known since the work of Post, Church, and Turing (Kleene, 1952, pp. 298–301), a computer is a quite general symbol manipulating system, and we are very little constrained in the interpretations we may place on the symbols or the ways in which we may operate upon them.

One price we pay for this power of representation is that we are seldom able to integrate computer programs in closed form in order to derive general theorems about their behavior. Instead, our main route to understanding their implications is actually to compute that behavior for a number of specific cases. Hence, if we wish to see how well a program describes human behavior in some task domain, we give to the computer tasks from that domain and

compare its trace with the sequences of behaviors of human subjects confronted with the same tasks. This technique of simulation is the analogue, in the nonnumerical arena, of numerical analysis with difference equations of more traditional kinds.

Production systems. The kinds of programs with which we will be concerned here are called *production systems* (Barr & Feigenbaum, 1981, pp. 190–199). In a production system all of the instructions have identical form, which may be represented thus:

$$C \rightarrow A.$$

The expression, C, to the left of the arrow, is called the *condition side* of the production; the expression, A, to the right, is called the *action side*. The condition side consists of a set of tests; the action side, of a sequence of symbol-manipulating actions. Whenever the conditions are satisfied by the current state of the system, the actions are executed.

One additional specification must be added to make such a system operative: a so-called "conflict resolution" rule that determines priority of execution among productions when the conditions of two or more are satisfied simultaneously. The simplest conflict resolution rule, which will suffice for our purposes here, is to scan the list of productions in top-down order, and to execute the actions of the first one whose conditions are found to be satisfied. It has been shown that production systems can have all the generality and power of a Turing machine.

	If "$X = N$" → Halt and Check
$ax + b = cx + d$	If Nx on right → Subtract(Nx)
$(a - c)x + b = d$	If N on left → Subtract(N)
$(a - c)x = (d - b)$	If Nx on left ($N \neq 1$) → Divide(N)
$x = (d - b)/(a - c)$	

FIGURE 1. *A Production System for Solving Equations*

The right side of Figure 1 shows a simple system of four productions that is capable of solving (symbolically or numerically) a broad class of linear algebraic equations in one variable. In this production system, the symbol N stands for any constant in the equations (e.g., a, b, c, d); x stands for itself. The first production may be read, "If the equation is in the form, an "x" followed by an " = " followed by any N, then halt and test if N can be substituted for x in the original equation." The second production may be read, "If there is a term of the form "Nx" on the right-hand side of the equation, subtract that term from both sides and simplify." (Separate productions could take care of combining similar terms, but I have chosen to embed simplification in the other productions.) The remaining productions have readings analogous to that of the second.

The reader can easily satisfy himself that, given the equation shown in the first line on the left side of Figure 1, the production system will take precisely the three steps that follow in order to solve the equation, and then will halt and

check its answer. We may postulate that when elementary or high school children learn to solve simple equations, they are acquiring productions somewhat like those described here. It is perhaps reassuring that only four productions need to be mastered, although more would have to be added to accommodate a wider range of equations (for example, equations with fractional coefficients in some of the terms).

The phrase "acquiring productions" should not be confused with "memorizing productions". Memorizing the production rules would not help a student one whit to solve equations. What has to be acquired is the ability to notice and recognize the critical features of the equations that are mentioned in the condition parts of the productions, and to associate with these recognized features the relevant actions to be taken. The perceptual component of this skill—the ability to recognize "instantly" when a particular action is appropriate—is not much, if at all, emphasized in algebra textbooks that we have examined. The books are very good in explaining what actions are legitimate—that is, which ones preserve the value of the unknown. They tend not to explain how the student can tell *when* a particular action should be taken. The condition sides of the productions used in a proof correspond to what we usually call the motivations for the proof steps.

These comments are by way of an aside. It is not without interest that production systems that simulate school-level mathematical skills have potential implications for pedagogy. Understanding the production systems that underlie the processes used by mathematicians (or budding mathematicians) enables us to ask with some precision what methods might be effective in helping learners to acquire these productions (Larkin et al., 1980).

Let me now turn to the main topic, which is to look at systems that perform tasks much more complex than solving simple linear equations. In particular, we will consider the task that frequently confronts a user of mathematics in the sciences: finding a mathematical law that fits some given set of numerical data. I should warn the reader that the mathematics involved in my examples is elementary, even trivial. Only one of them invokes the calculus. But this is simply a reflection of the fact that much of the mathematics used in empirical science, especially the empirical science of the 18th and 19th centuries with which we shall be concerned here, is quite simple. The complexity lies in finding how to apply the mathematics to the empirical data.

We can think of such a task as an exercise in curve fitting, or we can think of it as a creative task of discovering new laws to describe and explain empirical phenomena. The difference between these two views of the process does, indeed, lie in the eye of the beholder. I will make reference to a number of historical examples of scientific discovery by induction from data in order to persuade you that the task, however interpreted, is not a trivial one, and that it plays a significance role in the progress of science.

The Process of Discovery

My discussion of the process of discovery will draw very heavily upon research I have been doing in collaboration with Patrick Langley, Gary Bradshaw, and Jan Zytkow. This work is largely embodied in a computer

program, BACON, whose first versions were produced by Langley, and which has since been considerably elaborated by the joint efforts of our research group (Bradshaw, Langley & Simon, 1980, 1983; Langley, 1979; Langley, Bradshaw & Simon, 1983; Langley, Zytkow, Simon & Bradshaw, 1983). The BACON program has capabilities for detecting lawfulness in data and extracting that lawfulness in the form of equations that fit the data. While BACON has capabilities for ignoring moderate amounts of noise in data, we will not consider here the problem of approximation, but will assume that the data are exact.

Fitting mathematical laws to data. The problem of fitting laws to data has an entirely definite criterion for solution. The problem is solved as soon as a function is found that fits the data. There is, of course, no guarantee that the function is unique. For any finite set of data points, there exist an infinite set of functions that can fit them. If any one of these is discovered, BACON's search process halts. Nor is there any guarantee of the inductive validity of the function that is found. If new observations are added to the data set, the function originally discovered may or may not fit them. Both the ambiguity of the solution and the lack of a guarantee of inductive validity are important properties of BACON, which we believe are found also in the real world of science. I will have more to say about both of these issues a little later.

In the BACON system, the problem of finding a function to fit a set of data is approached by the method of selective search. Candidate functions are generated, and then tested to see if they fit. But because the spaces of functions to be searched are enormous, the search cannot be a matter of mere trial and error, but must be highly selective.

One principle of selectivity embodied in the generator of candidates is that it tries "simple" functions before it tries "complex" functions. The notion of simplicity here is pragmatic, yet not wholly arbitrary. Unless the generator is to be given the full list of candidate functions in extension, it must manufacture them, combinatorially and recursively, from some small set of primitive elements. It will therefore look at the primitives and their immediate combinations first, and then at more elaborate combinations, continually creating new functions from those already in the pool. We can simply use the order of generation as our simplicity measure, in which case the principle of "simplest first" becomes tautological. Or we might define complexity in terms of numbers of parameters, in which case there would usually be a high correlation between this measure and order of generation.

To say that the generator should be selective means that it should employ heuristic principles in generating new functions that will produce plausible or likely candidates early in the search. To do this it must, of course, make use of information extracted from the data themselves, so that the order of generation will depend upon the data to be fitted. The nature of these heuristic principles will be considered below in some detail.

Not only does the process described here not guarantee a unique solution, it also does not guarantee that any solution whatsoever will be found. The search may terminate without one. But that is reasonable, for there are many bodies

of data, at any moment in the history of science, that have not until that moment found a description or explanation. A system that always found answers to the questions posed to it could hardly be regarded as a realistic theory of discovery in the world of science.

An example: extrapolation of letter sequences. Before turning to actual examples drawn from astronomy, physics, and chemistry, let me illustrate the nature of law-finding by heuristic search in a simpler situation. The so-called series completion task is a common component of standard intelligence tests. The test items are sequences of letters or digits, for example: ABMCDMEFM.

The task is to extrapolate the sequence. A mathematician will immediately object that *any* sequence of letters is as good an extrapolation as any other. Or if pattern is wanted, the entire sequence, ABMCDMEFM, can simply be repeated indefinitely. Both observations are correct—but will not earn the observer a passing grade on the intelligence test. The answer that is "obviously" expected, and the only one that will be graded correct, is GHM.... The test taker is expected to detect a simple pattern in the sequence presented, and to use that simple pattern to make the extrapolation. The pattern is based, in turn, on the fact that the relations of *same* and *successor* may hold between various pairs of symbols. Thus, B is the successor to A and C to B in the Roman alphabet. Moreover, the letter M is repeated in every third position. Thus, if we denote the successor relation by N (for "next"), the equality relation by S (for "same"), and positions in the sequence by subscripts indicating cycle and position within cycle, we might represent the pattern by three equations:

$$x_{i3} = S(x_{13}), \quad x_{i2} = N(x_{i1}), \quad x_{(i+1)1} = N(x_{i2}),$$

with initial conditions x_{13} = M, x_{11} = A.

The set of relations among symbols that we are prepared to recognize defines a space of possible patterns. However, these are not generated randomly. The first step in the generation of a pattern is to detect pairs of symbols between which the relations of same and successor hold. The next is to detect the periodicity of these relations. With that information in hand for the initial segment of the sequence, a candidate pattern is constructed and tested against the remainder of the sequence. Some trial and error may be necessary, particularly when redundant and accidental relations are present (consider the sequence, KLMMNMOPM...), but in general only a few candidates need be generated before one is found that fits the finite test sequence.

As we have already seen, the pattern that is found need not be unique, and there is no guarantee that when additional symbols from the "true" sequence (i.e., the sequence in the test constructor's mind) are presented, they will fit the pattern that has been induced. By the method of generating the pattern, however, the one that is discovered will generally be among the simplest available, that is, among those that can be defined with the fewest symbols in the pattern description language.

Laboratory study of human subjects solving series completion problems provides solid evidence that the heuristic search procedure sketched above

describes accurately the way in which people approach these problems (Simon & Kotovsky, 1963; Kotovsky & Simon, 1973; reprinted as Chapters 5.1 and 5.2 in Simon, 1979). They notice the *S* and *N* relations between symbols, detect the periodicity of the sequence, describe the pattern, and extrapolate. They do not address the question of uniqueness, and they exhibit high agreement in their descriptions of the pattern. The presence in the sequence of "spurious" relations makes the task more difficult. (E.g., LMMMNM is far more difficult than ABMBCM.)

Data-Driven Discovery by BACON

The BACON program (named after Sir Francis Bacon) was initially designed to show how the discovery of scientific laws could be driven by data without guidance from existing theory. The BACON program is written as a production system, but I will provide only an English-language description of it here.

Of course in much scientific activity, the search for new laws is guided by theoretical conceptions that determine or suggest what data are relevant and even what forms the laws might take. However, in many important cases, especially during the early stages in the study of some phenomena and before any theory has emerged, the data themselves provide the only information that is available about the likely directions of search. I should like to describe three examples of such situations and show how BACON handles them.

Kepler's Third Law. Kepler's discovery of the three laws of planetary motion that bear his name provides one of the most important and striking examples in the history of science of data-driven discovery. His Third Law states the relation between the distances of the planets from the Sun (D) and their respective periods of revolution about it (P):

$$P = KD^{3/2}.$$

No theory was available to explain this regularity until Newton, two-thirds of a century after Kepler, proposed the inverse-square law of gravitational attraction. Only the data themselves provided any clues as to the form of the regularity. How could it be found?

There are several potential routes to the discovery. Nowadays, we might be motivated to graph the data points and, noticing the relation to be curvilinear, regraph them on log-log paper. Then, the linearity of the relation would be obvious, and even the approximation of the slope to 3/2 might be noticed.

BACON follows a different route to the same result. Its first heuristic is to search for correlations between the observables, and here it finds that as P increases, D increases monotonically. On the basis of this clue, it tests whether their ratio, $P/D = V_1$, is a constant. Of course it is not, but BACON retains V_1 as a new variable.

Now BACON notices that V_1 varies with D, and similarly computes their ratio:

$$V_1/D = P/D^2 = V_2.$$

Again, V_2 is not a constant, but varies inversely with V_1. Multiplying these two quantities together, BACON finds:

$$V_1V_2 = P^2/D^3 = K, \quad \text{a constant.}$$

Thus BACON, seeking an invariant function of the observables, has found Kepler's Third Law with little extraneous search and with the help of two simple heuristics:

1. If two quantities covary (countervary), test their ratio (product) for invariance.

2. Retain the ratios (products) so obtained and, treating them as new variables, continue to apply the same process to new pairs of variables.

> Of course the space of functions of the observable variables that these heuristics will induce BACON to search is severely limited. But any listing of important laws of eighteenth and nineteenth century physics and chemistry will show that a large proportion of them fall in this space. The addition to BACON of the ability to generate exponentials, logarithms, and trigonometric functions of variables is easily accomplished and would greatly enlarge the space. However, it might also greatly extend the search process unless additional heuristics were available to guide it and make the generation of functions more selective.

Perhaps the conservative conclusion to be drawn from the example of Kepler's Third Law is that relatively simple curve-fitting methods may suffice to discover significant scientific laws in data, even without guidance from theory, and without extensive search.

Conservation of momentum. In its search for Kepler's Third Law, BACON introduced as new variables several functions of the original observables. In that case, none of the functions had an interesting physical interpretation. We will now consider a case where BACON, enroute to discovering a law, is motivated to introduce a new function that turns out to correspond to an important physical concept, inertial mass.

We suppose a spring is attached at its ends to two bodies A and B. We stretch the spring and release it, accelerating the two bodies; and we measure the initial accelerations. We repeat the experiment with the spring stretched to different lengths. Given the data on the accelerations, a_{Ai} and a_{Bi}, of the two bodies, BACON will almost immediately discover that the ratio of their absolute values, K_{AB}, is a constant.

Suppose the experiment is now repeated with new pairs of bodies, selected from the set $\{A, B, C, D, \ldots\}$, generating new constants K_{AB}, K_{AC}, K_{BC}, and so on. BACON will now apply its third heuristic:

3. When a set of invariants is found, each involving the same relation between a pair of objects from a set of objects, attribute to each object

an *intrinsic property*, and try to express the invariant relation as a function of the values of these intrinsic properties.

In the case before us, BACON will attribute to each object O a property m_O, and, letting $m_A = 1$, will set $m_B = K_{AB}$, $m_C = K_{AC}$, and so on. Now, on testing the value of K_{BC}, BACON will find that $a_B/a_C = K_{BC} = m_C/m_B$, that is, that transitivity holds for the ratios of the accelerations of different pairs of bodies. Moreover, $m_B a_B + m_C a_C = 0$. In this way, BACON introduces the intrinsic property that we know as inertial mass and rediscovers the law of conservation of momentum.[3]

BACON has rediscovered and introduced other important physical properties in a similar way: for example, the coefficient of refraction from data on the paths of light rays passing from one medium into another, voltage from data on currents in electrical circuits, and specific heat from data derived from calorimetric experiments.

Integer ratios. An example from late eighteenth century chemistry will illustrate another of BACON's heuristics: its search for integer ratios among quantities (Langley, Bradshaw & Simon, 1983). If BACON is given data on the ratios of the weights of oxygen to the weights of nitrogen in some of the oxides of nitrogen (which include N_2O, NO, N_2O_3, NO_2, N_2O_4, and N_2O_5), it will seek to express all of these ratios as integer multiples of a greatest common denominator. In the example of the oxides of nitrogen, the integer multiples for the compounds listed above are 1, 2, 3, 4, 4, and 5, respectively. In this way BACON rediscovers Dalton's law of simple multiple proportions.

The same heuristic finds Prout's hypothesis: that the atomic weights of all of the elements are integer multiples of the weight of hydrogen. (In this case, we must be careful not to give BACON data on elements that, like chlorine, violate the hypothesis. These anomalies were only explained with the discovery of isotopes.) If BACON is given the volumes of inputs and outputs of gaseous reactions, it rediscovers Gay-Lussac's law of combining volumes and attributes to the substances involved an intrinsic property that we would interpret as their molecular weights. It also distinguishes between molecular and atomic weights—a distinction that was not fully clarified in chemistry until the 1860s.

The results we have just described derive from the addition to BACON of a single additional heuristic:

4. Look for integer ratios between the pairs of values of newly defined intrinsic variables.

Planck's law of black-body radiation. I will conclude this section with a brief account of Planck's discovery of the law of back-body radiation. That discovery would not be made by BACON in its present form, but it can be shown clearly what additions would be needed to give BACON the capability of making it. The case is of particular interest because of the fundamental importance of the law to the development of quantum theory, and because enough is known of its history to demonstrate conclusively that its discovery

[3] Actually, BACON introduces $1/m$ rather than m.

by Planck was wholly data-driven, and that its rationalization in terms of physical mechanisms followed afterwards (Pais, 1982; Kuhn, 1978).

In the autumn of the year 1900, it was widely believed that the distribution of intensities of black-body radiation as a function of wavelength and temperature was best described by Wien's law:

$$K_\nu = k_1 e^{-x}.$$

That law had also been obtained a few years earlier by a curve-fitting exercise, but early in 1900 Planck had constructed a derivation of it in terms of classical physical mechanisms drawn from thermodynamics and electromagnetics. On a Sunday afternoon in October, however, a colleague called on Planck to report that recent experiments had shown conclusively that Wien's law held only for large values of x, and that for values close to zero, K_ν was definitely linear in $1/x$. Before he retired that night, Planck had discovered the law that now bears his name. The route to the discovery almost certainly took the following path:

The problem was to find an interpolating function that, in the limit as x became large, would approach Wien's law, and in the limit as x became small, would become proportional to $1/x$. An obvious way to examine these limits was to expand e^x into a Taylor's series: $e^x = 1 + x + \cdots$. But by subtracting 1 from both sides, we get a function that asymptotically varies as x; while the left side, $e^x - 1$, approaches e^x as x grows without limit. Writing Wien's law in the form k_1/e^x, it becomes clear immediately that the desired function is $K_\nu = k_1/(e^x - 1)$. This new expression indeed fit the data excellently over the entire range of x.

Having discovered a formula that fit the data, Planck spent the next two months constructing a physical model of black-body radiation from which he could deduce the desired result. He was successful at this, but only at the cost of making some very unorthodox assumptions about the underlying probabilities—assumptions that implied the quantization of the phenomena.

We need not debate which was more important: the discovery of the law, or its rationalization in terms of physical principles. (The law is as acceptable today as when it was discovered; however, today we would describe the physical phenomena somewhat differently than Planck did.) Nor need we debate whether the steps I have described as "obvious" and "clear" were actually so; for on this last point I have some casual empirical evidence.

Over lunch, I have presented to two colleagues, on separate occasions, the problem of finding an interpolating function having the properties described above. That is, I asked them to find a function that would go to e^x as x increased without limit, and to x as x approached zero. I made no reference to black-body radiation, but simply described the problem as one that had arisen in my own work. The colleagues to whom I presented it are both distinguished for research that involves the use of mathematics to model physical phenomena. Each gave me the "correct" answer in less than two minutes, and each used Taylor's expansion of e^x as his route to the solution. Neither was reminded of the black-body radiation law by my description of the problem, and they were appropriately surprised when I explained my deception. Both were, of course, familiar with Planck's law, but not with Wien's.

The heuristic underlying this particular act of data-driven discovery is clear.

5. If functions are known that fit empirical data over two different parts of the total range of the data, find a function that approaches the original functions asymptotically in the appropriate regions.

Final comments on data-driven discovery. We have now seen four examples of data-driven discovery, three of them accomplished by BACON. BACON's heuristics make it possible to fit functions to data using only a small amount of highly selective search. The four BACON search heuristics that were described, and the additional heuristic used to find Planck's law, are all completely general in the sense that they make no reference to the physical phenomena from which the data derive. In this sense, the process can be described as pure curve fitting. What is remarkable about it is that discoveries of important physical theories can be made in this way.

THEORY - DRIVEN DISCOVERY

When something is already known about the phenomena from which data are derived, then the existing theory can be a source of additional selective heuristics to guide search for a new law to fit the data. We may call search "theory driven" when it employs such heuristics. In this section I will present two examples of theory-driven discovery of scientific laws, the first illustrating the use of conservation principles as heuristics, the second, an atomic hypothesis that implies the conservation of atoms.

Black's law of heat. Joseph Black was the first to state the law for the equilibrium temperature of a mixture of two quantities of (possibly different) substances at different initial temperatures (Bradshaw, Langley & Simon, 1983). Writing T_1 and T_2 for the initial temperatures, T_F for the equilibrium temperature of the mixture, and C_1 and C_2 for the heat capacities of the two substances (the products of their masses by their specific heats), Black's law may be written

$$T_F = (C_1T_1 + C_2T_2)/(C_1 + C_2).$$

Now from data on the initial and final temperatures and the masses of the quantities of several substances, BACON can induce this law, employing in its search a combinatorial experimental design that varies one of the independent variables at a time. In the course of deriving the law, BACON will invent the concept of specific heat and will assign specific heats to the substances used in the experiments. While the discovery is straightforward, for the law has a relatively simple form, a good deal of data has to be processed along the way.

If some conservation assumptions are added to BACON's heuristics, then the search becomes much more direct and rapid. Specifically, suppose we assume that both total heat ($H = CT$) and heat capacity C are extensive and additive quantities. Then

$$H_F = C_FT_F = (C_1 + C_2)T_F,$$

and also

$$H_F = H_1 + H_2 = C_1T_1 + C_2T_2,$$

so that, equating the right-hand expression of the first line with the right-hand expression of the second, we obtain Black's law immediately.

In this case the theoretical assumptions of conservation of heat and of heat capacity are so powerful that they allow Black's law to be deduced without any reference whatsoever to empirical data. The data are now needed only to check the empirical validity of the resulting law.

The conservation principle, though genuinely a theoretical assumption, is as general as the heuristics used by BACON in data-driven discovery, for it can be applied as a hypothesis to any of the variables that appear in a problem. In a somewhat similar way, BACON can use symmetry assumptions to cut down its search for laws that involve several variables of the same kind. For example, in deriving Black's law without conservation assumptions, BACON could reduce its search substantially by assuming that the law it is seeking must be symmetrical in T_1 and T_2, and in C_1 and C_2.

On the basis of our explorations of Black's law, we may add to our heuristics:

6. Hypothesize that extensive quantities are conserved.
7. Hypothesize that similar variables will enter into laws symmetrically.

Discovery of molecular structure. An important task of chemistry is to assign molecular formulas to compounds and elements. Dalton applied, for this purpose, his law of simple proportions, combined with the hypothesis that if two or more molecular formulas were consistent with the data, the simplest should be used. Application of this rule produced some correct and many erroneous molecular formulas, and this domain remained in considerable confusion until the 1860s. The situation was gradually clarified by consistent application of the atomic hypothesis combined with Gay-Lussac's (and Avogadro's) hypothesis that equal volumes of gasses under standard conditions contain equal numbers of molecules.

The particular form of the atomic hypothesis that we need for this purpose involves the assumptions that atoms combine in "packets" (molecules), and that total numbers of atoms of each kind are conserved in chemical reactions. These assumptions together with Gay-Lussac's hypothesis impose strong constraints upon molecular structure. Consider, for example, the reaction for forming ammonia. One volume of nitrogen and three of hydrogen combine to form two volumes of ammonia vapor. We start by postulating the simplest possible molecular structures for nitrogen and hydrogen: one atom per molecule each, and similarly, the simplest structure, NH, for ammonia. The reaction would then be:

$$\text{N H H H} \rightarrow \text{NH NH}.$$

But this equation does not balance, for there is only one N on the left side. Therefore, we change our assumption about the structure of the nitrogen molecule:

$$\text{NN H H H} \rightarrow \text{NH NH}.$$

Now the nitrogen balances, but the hydrogen does not. We therefore add another hydrogen atom to each ammonia molecule, but now there is a

deficiency of hydrogen on the left side. After several more steps of this kind, gradually enlarging the molecules, we finally arrive at:

$$\text{NN HH HH HH} \rightarrow \text{NHHH NHHH}.$$

What we have done, of course, is to solve some simple Diophantine equations by a crude, but not ineffective, iterative method. We could just as well have worked with numbers as with the diagrams. The heuristic we have used in the search is the conservation heuristic applied to numbers of atoms.

Comments: theory-driven search. The two examples of theory-driven search illustrate especially the power of conservation laws in facilitating discovery. In fact, as the case of Black's law illustrates, if the theoretical assumptions are sufficiently strong, the form of the function that will fit the data can be deduced directly without any need for induction from data. In these cases the form of the data is predicted and subsequent observations can be used to test the derived law and, thereby, the assumptions that led to it.

Histories of science tend to place great emphasis—perhaps too much emphasis—upon the competition among theories, without equal attention to the origins of those theories. The development of science is seen as a struggle among theories: the phlogisten versus the oxygen theory of combustion, wave versus particle theories of light, classical mechanics versus relativity, classical physics versus quantum mechanics. Data are then seen largely as the products of "critical" experiments and, hence, as the adjudicators of controversy.

A more balanced account of history would focus more clearly on how theories emerge, and not just on how, once they have appeared, they are tested. Such an account would show that both existing theories and new empirical observations play essential roles in the development of new theory.

Discovery of Mathematical Concepts

While the BACON program was developed especially to explore processes of discovery in the empirical sciences, there have also been some explorations of processes of discovery in mathematics itself. Among these are the AM and Eurisko programs developed by Douglas Lenat (Lenat, 1977, 1983). The goal embodied in these programs is to generate interesting new concepts and conjectures about those concepts. The programs accomplish this with the use of heuristic search not unlike that employed by BACON.

The AM program is provided with four kinds of inputs. First, it is supplied with a few primitive concepts in some domain. For example, it may be given knowledge about sets, subsets, union and intersection of sets, and so on. Second, it is given the goal of creating new concepts in that domain, and conjectures relating to those concepts. Third, it is given some criteria for evaluating concepts as more or less interesting. Concepts may be adjudged interesting, for example, to the extent that they are related to other interesting concepts, to the extent that examples can be found of them (but not too easily), to the extent that they constitute limiting cases of more general concepts, and so on. Fourth, AM is given some heuristics to aid it in searching for concepts. For example, it may generalize or specialize concepts it has already obtained or been given, or may try to find examples of concepts.

The criteria and heuristics provided to AM are relatively many, numbering in the hundreds. But, like BACON's heuristics, they are quite general and do not make specific reference to particular task domains. The memory structures, too, of AM are homogeneous. Various properties may be predicated of concepts and relations stated between concepts. In addition, programs, written in the LISP programming language, may be associated with concepts, enabling the system to create examples or to test whether a particular object is or is not an example of the concept.

When AM was tested, using concepts drawn from set theory as its initial stock of information, it was able to build upon these a substantial collection of new concepts and conjectures. In about two hours of processing time on a large computer (PDP-10), it reinvented the integers and the operations of addition, subtraction, multiplication, and division upon them, the prime numbers, and numbers with maximal numbers of prime factors (a concept that had earlier been studied by Ramanujan). As each of these was discovered, it was judged interesting and provided a basis for the next steps of discovery. The two principal conjectures at which AM arrived were Goldbach's conjecture and the fundamental theorem of arithmetic. AM has no capabilities for proving theorems, so could not test these conjectures.

EURISKO extends even further AM's principle of homogeneity of memory structure and program. While AM's heuristics are immune from alteration by the program itself, so that new heuristics cannot be learned, EURISKO is designed so that new heuristics can be introduced into the system in exactly the same way as new concepts of any other kind. Lenat's publications on these two programs provide a more complete picture of their structure and performance. From the brief description given here, it can be seen that they illustrate mechanisms of discovery in mathematical domains just as BACON illustrates mechanisms of discovery in the domains of natural law.

Conclusion

It has sometimes been thought that while deductive processes could be carried out by mechanism, inductive processes were beyond the reach of mechanistic algorithms. For many years philosphy of science has been extremely skeptical of the possibly of creating a theory of scientific or mathematical discovery. The testing of theories could be mechanized, but not their discovery.

That skepticism reflects a romantic rather than a scientific view of the nature of human thinking. Scientists and mathematicians, especially good scientists and mathematicians, are not engaged in pure trial-and-error search, whether exhaustive or random. They have reasons, embedded in heuristics (hence not always conscious), for searching along particular paths rather than others. Through understanding these reasons, these heuristics, we gain insight into the discovery process, and through that insight we come to see why some methods of discovery are more effective than others.

It has been the purpose of this paper to describe the progress that has been made toward understanding the processes of discovery, and toward building computer programs capable of carrying out those processes and actually

making substantial discoveries (or rediscoveries). In it I have proposed a mathematical formalism (symbolic difference equations in the form of computer programs) that can be used to model the human thought processes used in problem solving and discovery.

I have illustrated the application of this formalism to the processes of discovery in the natural sciences; and I have shown how these hypothesized processes have been tested against our knowledge of a number of historically important scientific discoveries.

Through the kind of research described here, we are learning a great deal about the processes of thinking—particularly, but not exclusively, in the realm of applying mathematics in modeling, and thereby understanding physical situations. Other research, which I do not have space to report here, is addressed to understanding (with the help of the same computer modeling techniques) the processes that students use to formulate and solve problems in secondary school and college mathematics, physics, and chemistry.

Research on discovery processes is an important and exciting activity in its own right. It addresses one of the fundamental questions that has always fascinated mankind: how can a mechanism like the brain perform the functions of mind? But our growing understanding of the processes of discovery, and of other thinking processes, also holds promise of important social applications. For a deeper understanding of the human mind holds great promise for improving our pedagogical techniques in all domains of science and in pure and applied mathematics. But that is a topic that would take us beyond the goals of this paper.

REFERENCES

A. Barr and E. A. Feigenbaum (Editors), *Handbook of artificial intelligence*, Vol. 1, Kaufmann, Los Altos, Calif., 1981.

G. Bradshaw, P. Langley and H. A Simon, BACON.4. *The discovery of intrinsic properties*, Proc. Third National Conf. Canad. Soc. Computational Studies of Intelligence, 1980, pp. 19–25.

_____, *Studying scientific discovery by computer simulation*, Science **222** (1983), 971–975.

S. C. Kleene, *Introduction to metamathematics*, Van Nostrand, Princeton, N. J., 1952.

K. Kotovsky and H. A. Simon, *Empirical tests of a theory of human acquisition of concepts for sequential patterns*, Cognitive Psychology **4** (1973), 399–424.

T. S. Kuhn, *Black-body theory and the quantum discontinuity, 1894–1912,* Oxford Univ. Press, N. Y., 1978.

P. W. Langley, *Rediscovering physics with BACON. 3*, Proc. Sixth Internat. Joint Conf. Artificial Intelligence, Vol. 1, Tokyo, 1979, pp. 505–507.

P. W. Langley, G. L. Bradshaw and H. A. Simon, *Rediscovering chemistry with the BACON system*, Machine Learning, An Artificial Intelligence Approach (R. S. Michalski, J. G. Carbonell and T. M. Mitchell, eds.), Tioga, Palo Alto, Calif., 1983.

P. W. Langley et al., *Mechanisms for qualitative and quantitative discovery*, Proc. Internat. Machine Learning Workshop (Urbana), Univ. of Illinois, 1983, pp. 121–132.

J. Larkin et al., *Expert and novice performance in solving physics problems*, Science **208** (1980), 1335–1342.

D. B. Lenat, *Automated theory formation in mathematics*, Proc. Fifth Internat. Joint Conf. Artificial Intelligence, 1977, pp. 833–842.

_____, *EURISKO: A program that learns new heuristics and domain concepts*, Search and Heuristics (J. Pearl, ed.), North-Holland, 1983.

A. Pais, *'Subtle is the Lord...' The science and life of Albert Einstein*, Oxford Univ. Press, N. Y., 1982.

H. A. Simon, *Models of thought*, Yale Univ. Press, New Haven, Conn., 1979.

H. A. Simon and K. Kotovsky, *Human acquisition of concepts for sequential patterns*, Psychological Rev. **70** (1963), 534–536.

H. A. Simon, P. Langley and G. Bradshaw, *Scientific discovery as problem solving*, Synthese **47** (1981), 1–27.

DEPARTMENT OF PSYCHOLOGY, CARNEGIE - MELLON UNIVERSITY, PITTSBURGH, PENNSYLVANIA 15213