THE UBIQUITY OF DISCOVERY[1]

1977 Computers and Thought Lecture

Douglas B. Lenat

Computer Science Department
Carnegie-Mellon University
Pittsburgh, Pa. 15213

Abstract

As scientists interested in studying the phenomenon of "intelligence", we first choose a *view* of Man, develop a *theory* of how intelligent behavior is managed, and construct some *models* which can test out and refine that, theory The view we choose is that Man is a *symbolic information processor* The theory is that sophisticated cognitive tasks can be cast as searches or explorations, and that each human possesses (and efficiently accesses) a large body of infoimal uiles of thumb *{heinistics)* which constrain his search The source of what we colloquially call "intelligence" *is* seen to be very efficient searching of an *a priori* immense space Some computational models which incorporate this theory arc described. Among them is *AM,* a computer program which develops new mathematical concepts and conjectures' involving them, A M is guided in this exploration by a collection of 250 more or less general heuristic rules. The operational nature of such models allows experiments to be performed upon them, experiments which help us test and develop hypotheses about intelligence. One interesting result has been the *ubiquity* of this kind of heuristic guidance: intelligence permeates everyday problem solving and invention, as well as the kind of problem solving and invention that scientists and artists perform.

1. INTRODUCTION

Much of the behavior which we regard as "intelligent" involves some sort of *discovery* process.[2] This is obvious for some of the the most creative and intellectually difficult human activities (identifying an unknown chemical compound, composing a new sonnet, deriving a new c.osmological model, conjecturing a new theorem, solving the NY Times crossword puzzle,...). We'll see that it's no less true for our everyday activities (cutting cheese, finding our way about Boston, solving the Pittsburgh Press crossword puzzle,...).

1•1• Model-Building in Scienee

We in the field of Artificial Intelligence (AI) want to understand how its's possible to do such things, to understand the mechanisms of intelligence. To go about it scientifically, we must first propose some hypotheses, use the results of experiments to modify and develop them into a theory, and then embody that theory in several concrete, testable models. This is the paradigm of Science; it has great power, as we all know.

In very "hard" sciences, objective data are available about isolated and relatively simple phenomena. This enables the construction of small yet quite rigorous, predictive models, using the language of mathematics (e.g., Maxwell's equations for electromagrietisrn). But in the so-called "soft" fields, the phenomena cannot be measured precisely, or are not so reproducible, or (as in

[1] This work was supported in part by the Defense Advanced Research Projects Agency (R4620-73-O0074) and monitored by the Air Force Office of Scientific Research.

[2] Much of the rest of "intelligence" involves algorithmic solving of well-structured problems. That topic will not be emphasized in this paper. We take the position that "algorithms known and used by experts" is just a proper subset of "knowledge experts use to reduce search".

the case of human intelligence) cannot be easily isolated for study. The resulting models are usually only *descriptive,* and may often be presented in everyday prose (e.g., psychological theories of personality).

Scientific models serve two functions: (i) to *unify* large masses of empirical data; and (ii) to *predict* new effects, which subsequently can be tested for by conducting experiments. The physicist's model (his set of equations) is better than the psychologist's model (his prose description) because he is able to draw on the power of established mathematics[3] to make his model *quantitatively predictive.*

What kind of model can we build for the phenomena of discovery and creativity? The more formal, mathematical models have the greatest potential predictive power, but humans seem just too complicated, sophisticated, and unpredictable for their behavior to be captured by a few equations. Planets and atoms behave in much more regular a fashion than do people.

Is that the end, then? Are we forced to build purely descriptive models of creativity? Are we limited to sterile prose discussions about the mysteries of incubation and illumination (e.g., as in [Poincare' 1929] and [Polya 1954])? Can we draw only metaphorical pictures (e.g., as in the 'hooked atoms' image that Einstein reports by introspection [Hadamard 1945])? The answer, until quite recently, was unfortunately "Yes".

1.2. Choosing an Approach in Science

F.ach science is differentiated from the others not merely by the set of *phenomena* it claims as its object of study, but also by the *approach* it takes (the science's *view* of those phenomena; its *paradigm* [Kuhn 1970], if you will).

So even though we've decided to study the phenomena of human intelligence (creativity, problem solving, etc.), we must still choose a *view* of Man.

If we view Man as a *Gestalt* actor whose internal thought processes can't be investigated, then we are called "classical psychologists" and we study his external behavior. If we view Man as a brain, as a piece of hardware built out of neuions, then we're called "biologists" and we study rieuruphysiological responses (eg., by implanting electrodes) If we view Man as a machine, then we're called "cyberneticists", and we investigate mathematical piopertics of feedback networks of simple components. If we view Man as a collection of atomic particles, then were called foolish : this is too fine a "granularity" with which to investigate intelligence.

Another view arose about twenty years ago, from three separate sources: engineering (Broadbent), psycholinguistics (Chomsky), and computer science (Newell and Simon). Man can be viewed as a *symbolic information processor.* If we adopt that view of Man, and are interested in the mechanisms of *human memory,* then we're called "cognitive psychologists". If we adopt that view and are interested in the mechanisms of *human thinking,* then we're "information processing psychologists". Finally, if we view Man as information processor only to learn more about problem solving and creativity, then we're working in the field of "Artificial Intelligence"

1.3. The Foundation for Artificial Intelligence

Suppose we view Man as information processor [Newell and Simon 1976], How can we construct some models of intelligence which are predictive rather than just descriptive? We might build *operational* models, which can *exhibit* whatever behavior our theory called for. To do this, we need to use a general purpose symbol manipulator, an automatic way to carry out each bit of information processing.

[3] The power to solve analysis problems in closed form (e.g., to solve a differential equation simply by repeatedly manipulating it according to Known transformations), or the power to make approximations when necessary, or the power to somehow "run" the model (to "grind out" solutions to his equations).

[4] Attempts to formalize "soft" phenomena do go on continually, but the interpretation of the resultant rigorous mathematics is often a topic of heated debate (a current example is Catastrophe theory [Kolata 1977]).

5 Unlike the Brownian motion of atoms in a perfect gas, the fundamental information processes of intelligence are not random.

No one view is "righl" or "wrong"; each is adopted because from it we can build a model, which in turn has some practical consequences and uses. When I'm ill, I want to go to a doctor who practices medicine based on Man as an animal, not Man as an economic agent. It is not productive to argue whether or not any specific view of Man is "correct", is "immoral", etc. At the present limited state of our understanding, any one view is bound to be simplistic and incomplete. On the other hand, we never capriciously adopt a view with impunity. It is the whole man who lives and reacts, even though we can only view him, first this way, then that.

A general purpose information processing system must provide (i) a way to specify what processing' gets clone when, *(ii)* a memory in which to store symbol structures, and *(iii)* an "engine" which can actually cause such processing to occur The science of Artificial Intelligence ("AI") sprang into being soon after the invention of one such computational engine, the general purpose electronic digital computer. In fact, AI is sometimes called "Machine Intelligence".

Computational models of some task can be used directly to carry out (a simulation of) the modelled activity (composing sonnets, doing astrophysics research, devising new ways to cut cheese, defining and investigating new concepts in mathematics, navigating this city, etc.). If our computer program does perform the desired task adequately, then we accept it as verification that our theory adecjuately *explains* one way in which intelligent performance at that task might be achieved.

1.4. The Paradigm of AI Research

We in A I have evolved the following paradigm:

1) Choose some human cognitive activity (like playing chess, proving theorems, understanding spoken English),
ii) Develop hypotheses and eventually a theory about what kinds of information processing could be taking place to produce such ability,
iii) Incorporate that theory into a computer program, which serves as the model.[8] That computer program is made to carry out the original activity, and the researcher can observe how well it does,
iv) By experimenting with his program, he attempts to find out where the apparent "intelligence" is really coming from

Over the last twenty years, we've hypothesized and tested scores of models, for several different sophisticated tasks. There have been many successes, and many failures - and we've learned much from thern. Some of these experiments will be described later (Section 3); for now, I just want to present a single, very central result.

It turns out that we can model a surprising variety of cognitive activities (problem solving, invention, recognition) as a *search* or exploration, in which the performer is guided by a large collection of informal rules of thumb, which we shall call "heuristics". But what's really exciting is that not only can this single theory (intelligence as heuistic rule guided search) explain the behavior of the brainstorming math researcher and the wandering Boston visitor, but when we go off and build up such models in detail, we find that they can all contain more or less the *same informal rules.*

Before trying to justify this result, let's notice two rather surprising consequences of it:

(i) Every day, each of us is forced to - and does successfully - carry out a great deal of "creative research" just to deal effectively with our complicated world;
(ii) There's a large component of non-formal, "plain common sense" knowledge that is necessary to do creative new work in lhe sciences or the arts.

Now that you see where I want to take you, let's see how to get there. I'll have to spell out this "heuristic rule based search" theory of intelligence, convince you that it makes sense and that it can account for the way in which people perform such disparate tasks as solving the 8 puzzle and performing scientific research. Also, I must demonstrate that there is a large core of heuristics which is common to all such activities.

2. HEURISTIC RULE GUIDED SEARCH

2.1. The Theory

There is a theory of intelligence lurking here, upon which some models - some computer programs - have been constructed. The theory goes something like this:

1. Human cognitive tasks can be cast as *searches,* as explorations wandering toward some goal which is well- or ill defined.
2. We are guided in these searches by a large collection of informal rules of thumb: heuristics.
3. We access potentially relevant heuristics in each situation, and either (a) select and then follow a single relevant heuristic, or (b) quickly "stitch together" some of the relevant ones, and then follow the "combined" advice.

Symbol processing is not the same as mere data processing. A "universal Turing machine", e.g., is just a data processor, because the marks on its tape aren't symbols, they don't represent anything but themselves. It was some years after its discovery that the digital computer was perceived as a general symbol manipulator.

[8] AI deviates from cognitive psychology at this stage. Psychologists would run experiments on people, to see if they really do fit the theory. We in AI are more concerned with whether the programs containing the hypothesized mechanisms are capable of any sort of "intelligent" behavior — even if differs somewhat from human performance at that task.

That's it.[9] It sounds plausible; in fact, it sounds trivial. Yet the models which incorporate this theory are capable of simulating sophisticated behaviors at many tasks which one would suppose require intelligence: organic chemistry problem solving, organic chemistry research, chess playing, discovery of new math concepts and conjectures

2.1.1 Intelligence and Information

The theory contains two important implicit assumptions, which might be worth stating explicitly:

> Man is viewed as a processor of symbolic information.
> Man exhibits "intelligence" by his performance at various cognitive tasks.

Let's take a moment to try to justify these remarks, that intelligence has something to do with information processing.

Twenty-seven years ago, Alan Turing [1950] rejected as meaningless the question *"Can machines' think?"*. He replaced it with a game, called the Imitation Game (now commonly referred to as the Turing Test). One version of that game would go as follows: A human interrogator is placed in an isolated room A teletype exists in the room, and by using it he can communicate with a computer and with another human, both located in the next room. The interrogator asks them each some questions, and then must guess which is the human and which is the machine. If we can program the machine in such a way that it fools the interrogator into making the wrong identification at least 507. of the time, then we shall say that the machine (as programmed) is "intelligent". Many of you are familiar with this game. Now let me introduce you to a slightly different one:

Thirteen years ago, Keith Gunderson [1964] rejected as meaningless the question *"Can rocks thfnk?"*. He replaced it with a game, an Imitation Game. A human interrogator is placed in an isolated room A small hole exists near the bottom of the door, through which the interrogator can shove most of his foot. On the other side of the door are located a human and a rock. Sometimes the human will stomp on the interrogator's foot, and sometimes the rock will be dropped on it. The interrogator must guess which one is the human. If we can shape a rock in such a way that it fools the interrogator at least 507. of the time, then we shall say that the rock is "intelligent".

Why does the dialogue test sound so much less silly, so much more indicative of intelligence, than the stomping-test? Because unrestricted dialogue is open-ended; to do well at it requires a massive wealth of knowledge, experiences, cognitive abilities, emotions, and common sense. Unrestricted foot stomping requires none of these. In short, the first test permits genuine interrogation of information and information processing capabilities, while the second test doesn't. If you agree that the first test is genuine and the second one bogus, it must be because intelligence has *something* to do with sophisticated processing of massive quantities of infot mation.

Thus it seems that the informal ion processing view of Man is an especially good one from which to *study* intelligence. Let's elaborate on it a bit more, and then go on and see what happens when one tries to build models based on it

2.2, Some Examples

Let's take a look at some heuristic searches that people perform, and in the process hopefully convey their universality, their ubiquity, and their power. We'll then be *in* a position to examine some specific AI models (computer programs) that they have built; this will be done in Section 3.

2.2.1 Everyday Problem Solving

Suppose we decide to plan a nip from CMU to MIT. How can we find a good route to take? We will probably find a detailed road map and begin *searching.* We have some powerful rules of thumb which make our search very short, usually. We look for some main highways that will take us most of the way, and then do some "fixing up" around the termini (from CMU to the first highway, from the end of that stretch to the beginning of the next one,.., to MIT).

This is the heuristic of planning in an abstraction space: we take the original detailed map (our "search space") and simply ignore all but the biggest roads marked on it. Needless to say, this makes the map much simpler. We also assume that whenever two big roads go nearby each other, they do in fact connect, and that big roads

[9] Notice the conspicuous absence of the word "representation" anywhere in the theory. To.design and construct a model for this theory would entail grappling with representational issues, much as any running instance of an abstract algorithm must exist on some particular machine. But the validity and power of the theory are independent of representation, just as the validity and complexity of an algorithm are independent of which machine it's implemented on.

which go near to cities do in fart pass through them. Next, we solve the problem in this very small space (called the "abstraction space"), Fimally, we use that solution as the skeleton of a real solution We may have a few more searches to perfonu (e.g., how do we *realty* get. from the Boston exit to the MIT campus itself) - but notice that all these additional searches will be small, localized fixups to the skeleton solution Using the heuristic method of *planning* has reduced our search dramatically.

This brief example has hopefully demonstrated how we can explain eveiyday problem solving behavior in terms ot the "heuristic rule seairh" theory of intelligence. We've seen a typical problem, and explained how to model it, how one could imagine even a computer being able to solve it: Cast it as a quest for a solution m some huge "search space", with the searcher being heavily constrained by knowledge embedded in general common sense rules of thumb (hem islics). In the next subsection, we'll show that the same kind of analysis can explain episodes of brainstoiming (in particular, the inventing some new kitchen gadget).

*2.2.2. E*veryday Invention

Suppose we're confronted with the following problem: we're fond of eating cheese, but every time we cut it with a knife, it crumbles il we try to cut it very thin. We could continue just cutting cheese the old way, but let's assume we try to design an improved tool We are now facing a search in an enormous space of possibilities. P>ut we have many informal rules of thumb which may help us

1. Sometimes, there will be a good way (perhaps a recent invention) to do somothing, more general than what is strictly asked for.
2. Consider what variables affect the success/failure of the current (inadequate) technique. Look for motivation at the extreme cases of the various known relationships involving those variables.
3. Look carefully at what is truly wanted; maybe the problem can be completely bypassed; at least, perhaps it's over-specified.

One of them (*2) says to look for motivation at the *extreme* cases of various known relationships. There seems to be some relationship between the thickness of the knife and the thinness of the slices we can cut. So we might consider as an extreme the thinnest knife possible, just a knife *edge,* and *voila'.* we've invented the wire cheese cutter. Or we might look at the extreme case of the relation that says that most cheese can be cut thinner if it's softer. We may ask ourselves if we can get the cheese very soft; an extreme case of *that* would be to get the cheese *directly under the knife-edge* completely molten; and *voila',* we've invented the "hot-knife"; probably General Electric will come out with one soon.

Notice that the same heuristic leads to several solutions. This was but a tiny example of heuristic guidance in action. In reality, we possess many hundreds of heuristics. Incorporating them into a computer model which could then tirelessly apply them seems like a very promising direction to follow We'll follow it in Section 3. Bear in mind, however, that inventions are rarely made with little search. In the cheese cutting case, there might be many ways of applying each heuristic, only a few of which are even remotely viable.

It's worth noting that both of the other heuristics might be used, too, for this situation. The first one could have us look at recent inventions which "cut": a laser for cheese slicing? The third heuristic might have us build a better mousetrap: go into business selling already sliced cheese; or try to devise a "cheese press" that takes the crumbs and squeezes them together into slices.

*2.2.2.1 Jndging 'Intnrcstinfliioss**

Let's put our theory to a most severe test, by asking whether it can account for our everyday judgments about what is and isn't "interesting". This would seem to be the realm of intuition, emotion, taste, and aesthetics — the antithesis of logical empiricism

If in fact there is a large "hormonal component" to such judgments, then it is not surprising if they lie outside the power of any theory founded on a view of Man which ignores that aspect of him. So even a partial success at explaining "taste" in terms of heuristic rules would be a major triumph for the theory, showing it had some relevance to these phenomena

Such a partial explanation is easy to find. Texts on the arts abound with "rules" of composition. They analyze works in terms of symmetry, coincidence, recency, unconventionaliiy, harmony, balance, utility, associations, etc. A typical rule might say

IF two apparently disparate, par is of the work arc suddenly recognized as being very closely related,
THEN that increases the interetigncM; of both parts, and of the whole work as well.

This explains the eternal popularity of the "recurrent theme" in all genres of literature and cinema. It explains the impact of having a single melody recur frequently but with slight differences each time (the concept of "musical variations") Dickens' success is due in no small measure to his mastery of this heuristic (i.e.,

"coincidence"): the reader is always astonished and pleased when two separate diameters are discovered to be one and the same person. The above heuristic also explains why (aside from cost) the hot-kiufe might be better received than the laser-knife (the consumer alrearly possesses hot combs, electiic blankets, electric knives,... so the hot knife would be instantly accommodated, with even a modicom of pleasure at the new bit of "closure" that had occurred in his world).

2.2.3 Scientific Problem Solving

Suppose we're designing a molecular genetics experiment, which will ultimately result in a certain biochemical *end* product. We must design a very long chain of steps (perhaps reaching into the thousands), each of the form "raise the temperature to x", "add the following" nucleotide...", etc

The search for a successful solution (a path, a sequence of steps leading to the final product) is made much simpler when we're able to apply some heuristics. It's a common practice, for example, to ignore certain kinds of minor steps, and to concentrate on finding a relatively small sequence of important intermediate products. That is, we should first find a solution in a heavily restricted (hence *small)* space. Afterwards, we can "fix up" the connections between these steps (e.g., in case one of them must occur at 12° and the next must occur at *25°)* by adding many minor steps. We don't begin to execute the first step until this detailed sequence of steps is fully specified. We solved the problem by applying a general heuristic: planning. The basic technique of planning can be expiessed as follows.

IF you arc lated with a search through a large space of possible solutions;, and some aspects of each "intermediate stale" along the way to a solution are mote important than others.
THEN try to ignore some of the delailed aspects of the problem, thereby simplifying il. Solve the simpler problem, and try to extend that solution into a solution of the original hard problem.

In the molecular genetics case, we chose to ignore such factors as the temperature at which a step must be carried out. The heuristic used is just the one we used in the previous subsection to find a route from CMU to MIT. There, we chose to ignore all minor roads on the map. This repetition is not accidental; it illustrates the commonality between everyday problem solving and scientific problem solving. The next section will show the similarity between everyday invention and scientific invention. And that's what we set out to show!

2.2.4 Scientific Invent inn

Suppose we're confronted with the following problem; we're fond of factoring numbers (e.g., 12=4x3), but often there are many little sets of factors that they crumble into (e.g., 12 crumbles into the following four sets: {1,12), {1,2.6), {1,3,4}. {1,2,2,3}). We may be content with this situation, or we may try to improve it in some way, or we may wish merely to find out more about factoring.

The latter two options both involve searches in enormous spaces (looking for a new kind of factoring; looking for new facts about factoring). But we have many rules of thumb to help us, including the three rules which were displayed in Section 2 22.

One of them («2) says to look for motivation at the *extreme* cases of various known relationships. The relation in this case is "Divisors of". It maps a number into a set of numbers (e.g., Divisors of(I2) {1.2.3.4.6,12}). An extreme case would be when it mapped a number into an extreme kind of set - say a singleton, doubleton, or empty set. In other words, consider the-set of numbers with no divisors, with one divisor, and with two d.ivi.sois. V*oila,* we've invented prime numbers. Notice that we used the very same heuiistic (#2 in Section 222) that Jed earlier to the invention of the wire cheese cutter and the hotknife.

2.2 4.I Judging' Initiestingness'

Even allowing that the "heuristic rule guided search" theory could account for the *discovery* of prime numbers, could it also explain how a researcher might have noticed that they were valuable, interesting, worth naming, and worth remembering?

Let's look at how the following three heuristic rules could lead to the conclusion that "Primes" is interesting, soon after it had first been defined.

IK specializations of conccpl 0 have just been created, and the current task is to find examples of eaeh of them,
THKN one rnetliod is to look over I lie known examples of C; they may he examples of some of the new specialized eoneepls as well.

IK all examples of a concept C, turn out to be examples of another concept D as well, and C was not previously known to he a specialization of D,
THKN conjecture that C is a specialization of D, and raise the "interestiriftiicss" value o| hnl.lt concepts.

IK all examples of a concept turn out to he in the domain of a rarely-applicable function K,

TH KN il's worth compuing all their K-values (their images uinlrr llto (million I"), ami Minlying lltal roller! ion of V values as a Kcparnlr concept.

Suppose we've just defined the set of numbers which have no divisors, the set of numbers which have only one divisor, only two divisors, only three-divisors. The lust of the three heuristics tolls us a quick way to find examples of those new special kinds of numbers: look at the known examples of numbeis, say the integers from I to 1000, and dump each one into whichever new specialized set(s) it belongs. If we do this, we get the following; empirical results:

Numbers with no divisors:	(none found)
Numbers with I divisor:	1
Numbers with 2 divisors	2, 3, 5, 7, 11, 13, 17, 19, ...
Numbers with 3 divisors:	4, 9, 25, -19, 121, 169, 289, ...

We can then apply the second heuristic: to each set of numbeis, to see if it's interesting. In this way, we'll notice that the last set, numbeis with three divisors, are all perfect squares. Heuristic #3 directs us to take then square roots. Lo and behold tliey'ie piecisely the thud set of numbers (i.e., numbers with two divisors; i.e. primes) Heuristic #2 notices this, and drastically increases the intereslin^ness values .of both Nos-with-3-divisors and Primes.

So we were able to fit this kind of judgmental evaluation of "intercstingness" into the same theory. Notice that the heuristics above are actually quite general: they (or at least analogues of them) can be used to evaluate works of art and new mechanical gadgets, as well as to evaluate new mathematical definitions. We saw the analogue of heuristic #2 used before, in Section 2 2 2 1, to help judge the intetestingness of A Tale of Two Cities and of a new kind of cheese cutter. Other analogous heuristics, which favor various kinds of "closure" are used in all sciences (e.g., consider the popularity of "charm", even before the confirming discovery of the D-meson [Richter 1977]). Partial de-mystification of such phenomena as illumination, incubation, aesthetic taste, etc., is one valuable result of this kind of AI research.

3. MODELS

Let me now describe a few landmark computer programs that have been written, models based on the theory of intelligence as heuristic rule guided search.

10 In the terminology of that section, the two "disparate parts" were "numbers-with-2-divisors" and "numbers-with-3-divisors".

These programs form but a thin sliver of the work that has gone on under the label of "Artificial Intelligence". 1 don't want to distract from the themes of this talk by cataloging (lie various efforts; a glimpse at the table of contents of this conference's proceedings will give you a pictuie of then scope.

3..L LT and GPS:General Problem Solving

Probably one of the earliest AI piograms written was the Logic Theorist ("IT", to its friends) [Newell, Shaw, and Simon 1957] It was repeatedly given symbolic logic theorems from Pihuipia Mathematica, and its task was to find a formal proof of each theorem LT had some given axioms and rules of infeienre. To search for a proof, in a completely exhaustive, systematic manner, would be quite an undertaking! But LT had a few heuristics which constrained its search:

> "...selecth/e principles that enable solutions to be found after examining only a relatively tiny subset of the set of possibilities. One such principle — illustrated by the methods in the Logic Theorist - is to Renerate only elements of the set that are already guaranteed to possess at least some of the properties that define a solution. Another principle -- illustrated by the matching process in LT -- is to make use of information obtained sequentially in the course of generating possible solutions in order to guide the continuing search. A third principle -- illustrated by the use of similarity tests in LT -- is to abstract from the detail of the problem expressions and to work in terms of the simpler abstractions."
>
> - [Newell and Simon 1972]. p. 137.

After they learned the power of heuristics from LT, the same group of researchers then worked on a program called GPS, for General Problem Solver. Its aim was to embed the above heuristics in a domain independent form, one not tied to solving any particular problem, the way that LT was tied to propositional calculus It was hoped that any problem could be representee] in the GPS formalism, and hence solved by GPS. GPS contained a few new heuristics:

1. Means ends analysis: Taking differences (between the-current state and the desired goal), locating operators relevant to reducing those differences,

AI includes such disparate pursuits as machine perception (speech understanding, image parsing,...), natural language understanding, novel ways to represent Knowledge, novel ways to control the attention of the computer, and ways to transfer some of what's been learned into improved teaching methods for people.

and applying those operators. Note that this heuristic guides *forward* search toward a goal.

2. Setting up subgonls, especially subgoals of the form "GPS wants to apply operator X, so X's preconditions must becomr satisfied (True)". This heuristic is equivalent to the idea of *problem ieduction,* to divide and conquer strategies, etc.
3. Work on the most difficult of the subproblerns first. Try to reduce the most important difference first.
4. Planning in an abstraction space. Systematically ignore some kind of details in the original problem. This results in a much smaller search space, m which a solution can be more readily found. This solution is then used as a *guide* to finding a solution in the original, huge search space.

I el's say a few more words about that last heuristic, planning, and how it was used when GPS was set to the very same task which LT had al tempted: propositional calculus theorem proving. What GPS did was to take all the axioms, rules of inl'eience, and the desired theorem, and then simply remove all the logical connectives from them. Thus the *modus ponens* rule of inference would be transformed from *"from 'P' and 'P implies 0', conclude* '()'" into this more abstract form: *"from 'P' and 'POJ, conclude* 'Q" This would sometimes produce fallacious proofs, or proofs with missing steps, etc., but by and large it was quite cost effective, allowing the rapid solution of many otherwise intractable pioblerns. After the planning process was over, there would still have to be some "fixing up" of the details between steps; this was also what we observed for route planning and for molecular genetics experiment planning,

A decade or two of research has shown that, yes, a large array of problems can be cast in terms that GPS can deal with (states, operators, difference tables), but no, GPS' general heuristics just aren't powerful enough to solve problems as difficult as can be tackled by humans.

3-2. DENDRAL: Scientific Problem Solving

The next giant step along the line of development we're following was taken by Ed Feigenbaum and Joshua Lederberg [Lederberg 1964], soon to be joined by Bruce tUichanan [Buchanan et al 1969]. They'* recognized what was lacking: expertise. Humans had to train in specialized fields for quite a while before tbey were able to solve any hard problems in them. This phenomenon ought not just be a reflection of human brain frailties, it might indicate a necessary requirement for intelligent problem solving in a complex knowledge rich domain.

in 1965 they conceived a new computer program, and in the process a whole new appmach for AI: they were willing to commit then program to working on a very specific kind of problem - in their case, the enumeration of atom bond graphs of organic molecules {based on analysis of mass spectrograph data and nuclear mass resonance data). They built their program, DENDRAL, around a body of heuristics -- not just a few, but a few *tens* of heuristic rules. Some of them were as general as the rules that LT had, but most of them were domain-specific informal rules of rhumb which they extracted from chemists (including some of the top experts in the field). These new heuristics were task-dependent: they won't work equally well for identifying images as for identifying unknown compounds; they were specific to the field of mass spectroscopy. Even more important than their specificity is then power: they constrain the searching tremendously.'*

The success of this research confirmed some of the conclusions of Newell, Shaw, and Simon: the tradeoff between generality and powei, the importance of heuristics for guidance Feigenbaum, Lederberg, and Piiichanan were willing to tap some of the "power" that humans need for.specific technical tasks.

3.3. AM: Scientific .Invention

We now jump to quite recent research, by the author. The next step in the progression we're chronicling involved collecting *hnuheds* of heuristics, used in some *very* difficult task, on the frontiers of what humans can perform. In this case, the task chosen was that of discovering interesting new concepts in elementary mathematics, that of scientific theory formation, of open-ended research This is scientific *invention,* as compared to Dendral, which performs scientific *problem solving.*

The size of the space to be searched is greatly reduced, hence searching will take much less time. Thus, heuristics (which serve to constrain) are equivalent to power (more efficient searching).

This is not the "inductive vs. deductive" issue: both Dendral and author's program, called AM, performed quite inductive tasks. The difference was that Dendral was given specific problems to solve (specific mass spectra to identify), whereas AM's activities were open-ended research, with no particular goal specified.

/* At about the same time, Joel Moses independently arrived at the same conclusions. He developed an expert program for the task of symbolic integration [Moses 1967].

AM's heuristics guided it to make promising new definitions, explore those new concepts, and judge the interestingness of its discoveries AM noticed connections between concepts, and was thus a theorem *proposer,* but it had no theorem *proving* abilities whatsoever. A *different* set of heuristics would be required if AM were supposed to prove any of the conjectures it proposes Making the necessary definitions to notice the fundamental theorem of arithmetic (numbers factor *uniquely* into primes) is quite a bit different (more ill-de.fmed, more sophisticated) than proving it once you've conjectmed it.

3.3.1 Math Discovery as Heuristic Rulc-Cuided Search

The task which AM performs is the discovery of new mathematics concepts and relationships between them. The simple paradigm it follows for this task is the one specified by our theory (in Section 21):

1. The activity of open-ended math research is viewed as a search, an exploration in a space of partially-developed concepts. The goal of this search is ill-defined; it is to maximize the interestingness value of what's being worked on at the moment.

2. AM is guided in this process by a collection of a few hundred heuristic rules. They are relatively general rules of thumb which guide it to define and study the most plausible thing next.

3. In each situation, AM accesses potentially relevant heuristic rules, finds which of them are truly relevant, and then follows them.

For example, AM possessed a rule of the form *"If f is an interesting relation*, *Then look at its inverse".* This rule fired (was relevant and was actually followed) after AM had studied "multiplication" for a while. The rhs (right hand side, THEN- part) of the rule directed AM to define and study the relation "divisors-of (e.g., divisors of(12) - {1,2,3,16,12}). Another heuristic rule which later fired said "*// / is a relation from A into B, then it's worth examining those members of A which map into extremal members of B*'\ This is a specialized version of our old friend, rule *2 from Section 2.2.2. In this case, f was matched to "divisors-of", A was "numbers", B was "sets of numbers", and an extremal member of B might be, e.g., a very *small* set of numbers. Thus this heuristic rule caused AM to define the set of numbers with no divisors, the set of numbers with only 1 divisor, with only 2 divisors, etc. One of these sets (the last one mentioned) turned out subsequently to be quite important; these numbers are of course the primes (as we saw in Section 224). The above heuristic also directed AM to study numbers with very *many* divisors; such "highly composite" numbers were also found to be interesting (by AM, by the author, and by professional mathematicians) Heuristics like those in Section 2.2.4.1 quickly led AM to boost the "interestingness" rating of the Primes concept.

3.3.2 Representation of Mathematical Knowledge

What exactly does it mean for AM to "have the notion of" a concept? It means that AM *represents* that concept somehow, that there is a data structure of some kind which is meant to correspond to, and contain information about, that concept. While the entire issue of representation of knowledge lias been de-emphasized in this paper, it is helpful to glance at how one typical concept looked after AM had defined and explored it:

```
NAMK:  Prime Number*, Primes, Numl»ers-u'itli-2- Divisors
DKFINITIONS:
    ORIGIN:  NiiMil>ei-of-«liv.sor*~of(x) - ?.
    PRKI>.-GAI,0|II,1IS: PrimeU) * (V./.)(/|x -> z=l XOH /,«x)
    ITKRATIVK: (for x>I): Kor i from 2 to x-1, -(.|x)
KX AMPITS:  2, 3, .'>, 7, 11, 13, 17
    BOUNDARY:  2, 3
    IUHINDAKY-TAIITKKS:  0, 1
    KA1UIKKS:  12
CKNKRAUZATIONS: Niunhers, Numbers with an even
     no. of divisors, Niimlx is with a prime no. of divisors
SPECIALIZATIONS: Prime, pairs, Prime uniqiielv-addablrs
C.ON.JKC'.S:  Unique faelorr/.at ion, Coldbaeh's conjecture
ANALOGIES:  Maximally-divisible number* are
          converse extremes of Divisors-of
INTEREST:  ConjecV tying Primes lo Times, lo Divisors-of,
          and to other closely related operations
WORTH:  800
```

The representation of a concept is as a collection of *facets,* each of which can have some associated value. For example, the value of the Worth facet of the Primes concept is 800. Another sample concept, "Sets", is presented earlier in these proceedings, in Section 2.1 of [Tcnat 1977],

3.3.3 Flow of Control

AM is initially given a collection of 115 core concepts, with only a few facets (i.e., slots) filled in for each. AM repeatedly chooses some facet of some concept, and tries to fill in some entries for that particular slot. Thus a "job" for AM is simply to engage in a mini-research project, to commit a simple act of discovery. Its overall task — to discover interesting concepts and conjectures — is accomplished as a composition of hundreds of these repeated attempts at little discoveries. To decide which small job to work on next, AM maintains an *agenda* of jobs, a global queue ordered by priority. A typical job

is *"Fill4n examples of Primes".* The agenda may contain hundreds of such entries. AM repeatedly selects the top job from the agenda and tries to carry it out. This is the whole control structure! Of course, we must still explain how AM creates plausible new jobs to place on the agenda, how AM decides which job will be the best one to execute next, and how it carries out a job. .

A heuristic rule is *relevant* to a job if and only if executing that rule brings AM closer to satisfying that job. Potential relevance is determined *a priori* by where the rule is stored. A rule tacked onto the Domain/range facet of the Compose concept would be presumed potentially relevant to the job *"Fill in the Domain of Sort olnseit".* The left hand side (IF- part) of each potentially relevant rule is evaluated to determine whether the rule is truly relevant.

Once a job is chosen from the agenda, AM gathers together all the potentially relevant heuristic, rules - the ones which might accomplish that job. The truly relevant ones are executed (followed), and then AM picks a new job. While a rule is executing, three kinds of actions or effects can occur:

(i) Facets of some concepts can get filled in (e.g., examples of primes may actually be found and tacked onto the "Example*" facet of the "Primes" concept).
(ii) New concepts may be created (e.g., the concept "primes which arc uniquely representable as the sum of two other primes" may be somehow be deemed worth studying).
(iii) New jobs may be added to the agenda (e.g., the current activity may suggest that the following-job is worth considering: "Generalize the concept of prime numbers").

The concept of an agenda is certainly not new: schedulers have been around for a long time. But one important feature of AM's agenda scheme *is* a new idea: attaching - and using -- a list of quasi-symbolic reasons to each job which explain why the job is worth considering, why it's plausible It is the responsibility of the heuristic rules to include reasons for any jobs they propose.

AM uses each job's list of reasons in three ways:

I. When a job already on the agenda is re-suggested, the supporting reasons are examined: If the job is being proposed for a *new* reason, then its priority (and hence its position on the agenda) will be raised; but if the job is being proposed for an already-recorded reason, then it's priority rating won't change.
2 Once a job has been selected, the quality of the reasons is used to decide how much time and space the job will be permitted to absorb, before AM quits and moves on to a new job.
3. To explain to the human observer precisely why the chosen (current) job is a plausible thing for AM to concentrate upon

Each of AM's *250* heuristic rules is attached to the most, general (o» abstract) concept C for which it is deemed appropriate. The relevance of heuristic rules is assumed to be inherited by all C's specialisations. For example, a heuristic method which is capable of inverting any relation will be attached to the concept "Relation"; but it is certainly also capable of inverting any permutation. If there are no known methods specific to the latter job, then AM will follow the Generalisation links upward from Permutation to rejection to Function to Relation..., seeking methods for inversion. Of course the more general concepts' methods tend to be weaker than those of the specific concepts

In other words, the aggregate of the Generalization/Specialisation relationships among the *concepts* induces a similar graph structure upon the set of *heuristic rules.* This "inheritability property" permits potentially relevant rules to be located efficiently.

3.3.4 Behavior of this Rule System

AM began its investigations with scanty knowledge of a hundred elementary concepts of finite set theory. Most of the obvious set-theoretic concepts and relationships were quickly found (e.g., de Morgan's laws; singletons), but no sophisticated set theory was ever done (e.g., cliagnnalizatinn). Rather, AM discovered natural numbers and went off exploring elementary number theory. Arithmetic operations were soon found (as analogs to set-theoretic operations), and AM defined such concepts as prime pairs, Diophantine equations, the unique factorization of numbers into primes, and Goldbach's conjecture. Many concepts which we know to be crucial were never[15] uncovered, however: remainder, gcd, greatcr-than, infinity, proof, etc.

[15] AM did not run forever (despite what anybody at SUMEX tells you). All the discoveries mentioned were made in a run lasting one rpu hour (Intorlisp+100k, SUMEX PDP-10 KI). Two hundred jobs in toto were selected from the agenda and executed. On the average, a job was granted 30 cpu seconds, but actually used only 18 seconds. For a typical job, about 35 rules were located as potentially relevant, and about a dozen actually fired. AM began with 115 concepts and ended up with three times that many. Half of the synthesized concepts were technically termed "losers" (both by the author and by AM).

Although AM fared well according to several different measures of performance (see Section 7.1 in [Lenat 1976]), it had some difficulties. As AM ran longer and longer, the concepts it denned were furthei and further from the primitives it began with; while the general set-theoretic heuristics were technically valid for dealing with primes and arithmetic, they weie simply too general, too weak to guide effectively. The key deficiency was the lack of adequate *mcla* rules [Davis 1977]: heuristics which cause the creation and modification of new heuristics. We are attempting to remedy this in EUR IS KG (see Section 3.4).

AM did demonstrate that scientific theory formation (the defining and exploring of new concepts and relationships) *could* be mechanized, could be modelled as heuristic rule guided search, using a few hundred heuristics for guidance. This is a significant verification of the theory of intelligence presented in Section 2.1.

3.-4. Other Heuristic Rule Guided...ExportJProfirams

There are several programs like AM in existence, knowledge based expert programs which perform under the guidance of a large collection of heuristic rules.

> The MYCIN program [Aikms 1977] [Shortliffe 1974] contains a couple hundred judgmental rules which were extracted from physicians, and it uses them to make diagnoses of various blood and meningitis infections

> TF.IRES1AS [Davis 1977] uses "meta-lcvcf knowledge rules to aid a human expert in transferring his knowledge to a program. Its initial task has been to assist physicians in adding new rules to MYCIN.

> MOLGEN [Martin et .al 1977] attacks the molecular genetics experiment-planning problem discussed earlier (in Section 22.3)

> META-DENDRAL [Buchanan and Mitchell 1977] is a theory formation program which looks over mass spectra and then correct identifications, and then abstracts that data into new fragmentation rules, new pieces of mass spectroscopy theory. These rules are then usable by the Dendral program, as if they had been extracted from a human expert.

> The PECOS program [Barstow 1977] contains rules about computer programming, and is a key component in an automatic programming system.

> UT-ITP [Bledsoe and Tyson 1975] is a natural deduction system, guided by a collection of judgmental rules useful when constructing formal proofs.

> The PROSPECTOR program [Duda et al 1977] performs geological analysis of aerial photographs, aiding a human expert in the evaluation of the mineral potential of exploration sites. Tts rules were gleaned from geologists, much as MYCIN'S were from physicians.

> The M-Method program [Zaripov 1975] for improvising variations on a given melody is based around a body of musicology rules.

> PUFF is a medical expert program, much like MYCIN in design, whose field of expertise is pulmonary disorders. "Everything PUFF knows about pulmonary function diagnosis is contained in (currently) 55 rules of the IF..THEN... form." [Feigenbaum 1977]

> The EURISKO program [Lenat et al 1977] is perhaps the most ambitious effort yet along this line, attempting to discover new heuristic rules in the domains it investigates. It is "ambitious" because even professional scientists are very poor at formulating - or even recognizing - new heuristics.[16] EURISKO's method for discovering and developing heuristics is simply to *not distinguish* between concepts and heuristics; i.e., each heuristic is represented internally as a full-fledged concept. So, e.g., any heuristic which can advise when it's time to generalize or forget any concept, can also automatically tell when it's time to generalize or forget any heuristic. Any method for creating a new concept out of old ones can be used to create new heuristics out of old ones. Evaluating the new heuristics is done just like evaluating any new concepts: by observing them in action, by gathering empirical data about them.

Often, discovering a single powerful heuristic can trigger a scientific revolution [Kuhn 1970] (e.g., Einstein's discovery of the heuristic "counterintuitive mathematical systems might have physical reality" led to a relatively important new paradigm in physics not too long ago. "Counterintuitive mathematical systems may be consistent and interesting" led to a parallel revolution in geometry fifty years earlier.