On the Thresholds of Knowledge.

Conference Paper · January 1987

DOI: 10.1109/AIIA.1988.13308 · Source: DBLP

Douglas B. Lenat
Cycorp, Inc.

ON THE THRESHOLDS OF KNOWLEDGE

Douglas B. Lenat
MCC
3500 W. Balcones Center
Austin, TX 78759

Edward A. Feigenbaum
Computer Science Department
Stanford University
Stanford, CA 94305

Abstract

We articulate the three major fmdings of AI to date: (1) The Knowledge Principle: if a program is to perform a complex task well, it must know a great deal about the world in which it operates. (2) A plausible extension of that principle, called the Breadth Hypothesis: there are two additional abilities necessary for intelligent behavior in unexpected situations: falling back on increasingly general knowledge, and analogizing to specific but far-flung knowledge. (3) AI as Empirical Inquiry: we must test our ideas experimentally, on large problems. Each of these three hypotheses proposes a particular threshold to cross, which leads to a qualitative change in emergent intelligence. Together, they determine a direction for future AI research.

1. The Knowledge Principle

For over three decades, our field has pursued the dream of the computer that competently performs various difficult cognitive tasks. The evidence suggests the need for the computer to have and use domain-specific knowledge.

Intelligence is the power to rapidly find an adequate solution in what appears *a priori* (to observers) to be an immense search space. So, in those same terms, we can summarize the empirical evidence: "Knowledge is Power" or, more cynically "Intelligence is in the eye of the (uninformed) beholder." The *knowledge as power* hypothesis has received so much confirmation that we now assert it as:

> The Knowledge Principle (KP): A system exhibits intelligent understanding and action at a high level of competence primarily because of the *specific* knowledge that it can bring to bear: the concepts, facts, representations, methods, models, metaphors, and heuristics about its domain of endeavor.

The word *specific* in the KP is important. Knowledge is often considered Compiled Search; despite that, the KP claims that only a small portion of the knowledge can be generalized so it applies across domains, without sacrificing most of its power. Why? Many searches are costly, and it's *not* costly to preserve the knowledge for future use. We all know about electricity, but few of us have flown kites in thunderstorms. In other words, *generality is not enough;* if you stop after acquiring only the general methods, your search for solutions to problems will not be constrained adequately.

There is a continuum between the power of already knowing and the power of being able to search for the solution; in between lie, e.g., generalizing and analogizing and plain old observing (for instance, noticing that your opponent is Castling.) Even in the case of having to search for a solution, the *method* to carry out the search may be something that you already know, or partial-match to get, or search for in some other way. This recursion bottoms out in things (facts, methods, etc.) that are already known. Though the knowledge/search tradeoff is often used to argue for the primacy of search, we see here that it equally well argues for the primacy of knowledge.

Before you can apply search *or* knowledge to solve some problem, though, you need to already know enough to at least state the problem in a well-formed fashion:

> The Well-Formedness Threshold: For each task, there is some minimum knowledge needed for one to even formulate it.

A more positive way to view this Threshold is that a large part of solving a problem is accomplished by having a good representation; that determines what distinctions to make explicitly and which ones are irrelevant.

Beyond this bare minimum, today's expert systems (ES) also include enough knowledge to reach the level of a typical practitioner performing the task [Feigenbaum 77]. Up to that "competence" level, the knowledge-search tradeoff is strongly tipped in favor of knowledge:

> The Competence Threshold: Difficult tasks succumb nonlinearly to knowledge. There is an ever greater "payoff" to adding each piece of knowledge, up to some level of competence (e.g., where an NP complete problem becomes Polynomial). Beyond that, additional knowledge is useful but not frequently needed (e.g., handling rare cases.)

Crossing the Competence Threshold, one enters the realm of experts. There, the knowledge-search tradeoff is fairly evenly balanced; otherwise, the general practitioners would have all acquired such knowledge themselves. Most current ES, in what we would still call the "first era" of expert systems, incorporate an amount of knowledge greater than that minimal level of competence for a task, yet less than all the existing expert knowledge about that task:

> The Total Expert Threshold: Eventually, almost all of the rare cases are handled as well. Continuing to add knowledge beyond this expert level is even *less* useful (per piece of knowledge added).

Human experts in a field are distributed between the Competent and Total Expert levels (see Figure 1). This does not mean that other knowledge is useless, just that it is not already understood to be relevant; e.g., even very far-flung knowledge might be useful to analogize to.

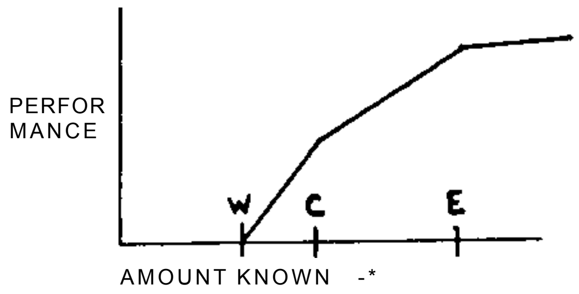

Figure 1. The level of performance of a program for some task, as a function of the amount of knowledge it embodies. The thresholds are (W)ell-formed, (C)ompetent, and Total (E)xpert. Beyond that last one lies "unrelated" knowledge.

The above arguments describe how the KP *might* work; but why *does* it work so frequently? Many useful real-world tasks are sufficiently narrow that the Competence Threshold can be crossed with only 50 -1000 if/then rules, and an equal number of additional rules takes one much of the way toward the Total Expert Threshold. Moreover, current experts don't already have all those rules explicitly codified; standard software design methodology can't build a program "in one pass" to perform the task. However, as the developing ES makes mistakes, the experts can correct them, and those corrections incrementally accrete the bulk of the hitherto unexplicated rules. In this manner, the system incrementally approaches competence and even expertise.

The newly added rules need to seamlessly interface to the existing ones; to put this the other way, you can never be sure in advance how the knowledge already in the system is going to be used, or added to, in the future. Thus:

> Explicit Knowledge Principle: Much of the knowledge in an intelligent system needs to be represented explicitly (although compiled forms of it may also be present).

When knowledge - including procedural knowledge - is represented as explicit objects, mcta-rules can apply to it, e.g., helping to acquire, check, or debug other rules. Such *knowledge objects* can be more easily analogized to, and enable generalizations to be structurally induced from them.

What about the control structure of an intelligent system? Even granted that lots of knowledge is necessary, might we not need sophisticated as-yet-unknown reasoning methods?

> Knowledge is All There Is Hypothesis: No sophisticated, as-yet-unknown control structure is required for intelligent behavior; control strategies are knowledge, and a standard evaluator can apply them.

On the one hand, we already understand deduction, induction, analogy, specialization, generalization, etc., etc., well enough to have Knowledge be our bottleneck, not control strategies. On the other hand, all such strategies and methods are themselves just pieces of knowledge. The control structure of the intelligent system can be *opportunistic:* select one strategy, apply it for a while, monitor progress, and perhaps decide to switch to another strategy (when some other piece of knowledge suggests it do so.)

2. Evidence for the Knowledge Principle

Fifty years ago, before the modern era of computation began, Turing's theorems and abstract machines gave a hint of the fundamental idea that the computer could be used to model the symbol-manipulating processes that make up that most human of all behaviors: thinking.

Thirty years ago, following the 1956 Dartmouth Summer Conference on AI, the work began in earnest. The founding principle of the AI research paradigm is really an article of faith, first concretized by Newell and Simon:

> The Physical Symbol System Hypothesis: The digital computer has sufficient means for intelligent action; to wit: representing real-world objects, actions, and relationships internally as interconnected structures of symbols, and applying symbol manipulation procedures to those structures.

The early dreaming included intelligent behavior at very high levels of competence. Turing speculated on wide-ranging conversations between people and machines, and also on expert level chess playing programs. Newell and Simon also wrote about champion chess programs, and began working with Cliff Shaw toward that end. Gelemter, Moses, Samuel, and many others shared the dream.

At Stanford, Lederberg and Feigenbaum chose to pursue the AI dream by focusing on scientific reasoning tasks. With Buchanan and Djerassi, they built Dendral, a program that solved structure elucidation problems at a high level of competence. Many years of experimenting with Dendral led to some hypotheses about what its source of power might be, how it was able to solve chemical structure problems from spectral data. Namely, the program worked because it had enough knowledge of basic and spectral chemistry.

Figure 2, below, shows that as each additional source of chemical knowledge was added, the Dendral program proposed fewer and fewer candidates (topologically plausible structures) to consider. See [Buchanan *et al].* When rules of thumb for interpreting NMR data were also added to the program, many problems -- such as the one illustrated -- resulted in only a single candidate isomer being proposed as worth considering! Threatened by an *a priori* huge search space, Dendral managed to convert it into a tiny search space. That is, Dendral exhibited intelligence.

Information Source	# of structures generated
Topology (limits of 3D Space)	42,867,912
Chemical Topology (valences)	14,715,814
Mass Spectrograph (heuristics)	1,284,792
Chemistry (first principles)	1,074,648
NMR (interpretation rules)	1

Figure 2: Dendral working on a typical problem: Finding all atom-bond graphs that could have the formula $C_{20}H_{43}N$.

When searching a space of size 1, it is not crucial in what order you expand the candidate nodes. If you want to speed up a blind search by a factor of 43 million, one could perhaps parallelize the problem and (say, by 1995) employ a 43-mega-processor; but even back in 1965 one could, alternatively, talk with the human experts who routinely solve such problems, and then encode the knowledge they bring to bear to avoid searching.

Obvious? Perhaps, in retrospect. But at the time, the prevailing view in AI (e.g, the Advice Taker) ascribed power to the reasoning processes, to the inference engine and not to the knowledge base. The *knowledge as power* hypothesis stood as a *contra-hypothesis.* It stood awaiting further empirical testing to either confirm it or falsify it.

The 1970*5 were the time to start gathering evidence for or against the Knowledge Principle. Medical and scientific problem solving provided the springboard. Shortliffe's Mycin program formed the prototype for a large suite of expert-level advisory systems which we now label "expert systems." [Feigcnbaum] Its reasoning system was simple (exhaustive backward chaining) and *ad hoc* in parts. DEC has been using and extending McDermott's RI program since 1981; its control structure is also simple: exhaustive forward chaining. These ES could interact with professionals in the jargon of the specialty; could explain their line of reasoning by displaying annotated traces of rule-firings; had subsystems (resp., Teresias and Salt) which aided the acquisition of additional knowledge by guiding the expert to find and fix defects in the knowledge (rule) base.

In the past decade, thousands of expert systems have mushroomed in engineering, manufacturing, geology, molecular biology, financial services, machinery diagnosis and repair, signal processing and in many other fields.

Very little ties these areas together, other than that in each one, some technical problem-solving is going on, guided by heuristics: experiential, qualitative rules of thumb - rules of good guessing. Their reasoning components are weak; in their knowledge bases lies their power. In the details of their design, development, and performing lies the evidence for the various adjunct propositions from Sec. 1.

In the 80's, many other areas of AI research began making the shift over to the knowledge-based point of view. It is now common to hear that a program for understanding natural language must have extensive knowledge of its domain of discourse. Or: a vision program must have an understanding of the "world" it is intended to analyze scenes from. Or even: a machine learning program must start with a significant body of knowledge which it will expand, rather than trying to learn from scratch.

3, The Breadth Hypothesis

A limitation of first-era expert systems is their brittleness. They operate on a high plateau of knowledge and competence until they reach the extremity of their knowledge; then they fall off precipitously to levels of ultimate incompetence. People suffer the same difficulty, too, but their plateau is much broader and their fall is more graceful. Part of what cushions the fall are layer upon layer of weaker, more general models that underlie their specific knowledge.

For example, if an engineer is diagnosing a faulty circuit s/he's unfamiliar with, s/he can bring to bear general electronics knowledge, circuit analysis techniques, experiences with the other products manufactured by the same company, handbook data for the individual components, common sense familiarity with water circuits (looking for leaks, or breaks), electrical devices (turn it off and on a few times), or mechanical devices in general (shake it or smack it a few times.) S/he mightanalogize to the last few times their car's engine failed, or even to something more distant (e.g., a failed love or a broken arm).

Are we, of all people, advocating the use of general problem solving methods and a breadth of knowledge? Yes! That does not contradict the KP, since most of the power still derives from a large body of specific task-related expertise. But a significant component of intelligence is still due to:

> The Breadth Hypothesis (BH): Intelligent performance often requires the problem solver to fall back on increasingly general knowledge, and/or to analogize to specific knowledge from far-flung domains.

Domain-specific knowledge represents the distillation of experience in a field, nuggets of compiled hindsight. In a situation similar to the one in which they crystallized, they can powerfully guide search. But when confronted by a *novel* situation, we turn to Generalizing and Analogizing.

Generalization often involves accessing a body of general knowledge, one that's enormous, largely present in each person, yet rarely passed on explicitly from one person to another. It is *consensus reality:* "water flows downhill", "living things get diseases", "doing work requires energy", "people live for a single, contiguous, finite interval of time". Lacking these simple common sense concepts, ES' mistakes often appear ridiculous in human terms: a skin disease diagnosis program might decide that a ten year old car with reddish spots on its body had measles.

Analogy involves *partial-matching* from your current situation to another (often simpler) one. Why does it work? There is much common causality in the world; that leads to similar events A and B; people (with our limited perception) then notice a little bit of that shared structure; finally, since we *know* that human perception is *often* limited, people come to rely on the following rule of thumb:

> Analogical Method: If A and B appear to have some unexplained similarities, Then it's worth your time to hunt for additional shared properties.

This rule is general but inefficient. There are many more specialized ones for successful analogizing in various task domains, in various user-modes (e.g., by someone in a hurry, or a child), among analogues with various epistemological statuses, depending on how much data there is about A and B, and so on. These are some of the n dimensions of Analogy-space; we can conceive having a special body of knowledge - an ES - in each cell of that n-dimensional matrix, to handle just that sort of analogical reasoning.

Why focus on Causality? If cause(A) and cause(B) have no specific common generalization, then similarities between A and B are more likely to be superficial coincidences, useful perhaps as a literary device but not as a heuristic one.

The above is really just a rationalization of how analogy *might* work. The reason this *frequently* succeeds has to do with three properties that happen to hold in the real world:

(1) The distribution of causes wrt effects. If there were a vast number of distinguishable kinds of causes, or if there were only a couple, then Analogy would be less useful.

(2) The moderately high frequency with which we must cope with novel situations, and the moderate degree of novelty they present. Lower frequency, or much higher or lower novelty, would decrease the usefulness of Analogy.

(3) The obvious metric for locating relevant knowledge -- namely, "closeness of subject matter" - is at best an imperfect predictor. Far-flung knowledge *can* be useful.

Analogizing broadens the relevance of the entire knowledge base. It can be used to construct interesting and novel interpretations of situations and data; to retrieve knowledge that has not been stored the way that is now needed; to guess values for attributes; to suggest methods that just might work; and as a device to help students learn and remember. Today, we suffer with laborious manual knowledge entry in building £S, carefully codifying knowledge and placing it in a data structure. Analogizing may be used in the future not only as an inference method inside a program, but also as an aid to adding new knowledge to it

Successful analogizing often involves components of both vertical (simplifying) and horizontal (cross-field) transformation. For instance, consider reifying a country as if it were an individual person: "Russia is angry". That accomplishes two things: it simplifies dealing with the other country, and it also enables our vast array of firsthand experiences (and lessons learned) about inter-personal relations to be applied to international relations.

4. Evidence for the Breadth Hypothesis

If we had as much hard evidence about the BH as we do for the KP, we would be calling it the Breadth *Principle.* Still, the evidence is there, if we look closely at the limits of what AI programs can do today. For brevity, we will focus on Natural Language Understanding (NL) and Machine Learning (ML), but similar results are appearing in most other areas of AI as well. As Mark Stefik recently remarked to us, *"Much current research in AI is stalled. Progress will be held back until a sufficient corpus of knowledge is available on which to base experiments."*

4.1 The limits of Natural Language Understanding

To understand sentences in a natural language, one must be able to disambiguate which meaning of a word is intended, what the referent of a pronoun probably is, what each ellipsis means,... These are knowledge-intensive skills.

1.1 saw the Statue of Liberty flying over New York.
2. The box is in the pen. The ink is in the pen.
3. Mary saw a dog in the window. She wanted it
4. Napolean died on St. Helena. Wellington was saddened.

Figure 3. Sentences presume world knowledge furiously.

Consider the first sentence in Fig. 3. Who's flying, you or the statue? Clearly we aren't getting any clues from English to do that disambiguation; we must know about people, statues, passenger air travel, the size of cargo that is shipped by air, the size of the Statue of Liberty, the ease or difficulty of seeing objects from a distance,... On line 2, one "pen" is a corral, the other is a writing implement. On line 3, does "it" refer to the dog or the window? What if we'd said "She *smashed* it"? A program which *understood* line 4 should be able to answer 'Did Wellington hear of Napolean's death?' and "Did Wellington outlive Napolean?"

For any particular sample text, an NL program can incorporate the necessary body of twentieth century Americana, of common sense facts and scripts, may be required for semantic disambiguation, question answering, anaphoric reference, and so on. But then one turns the page; the new text requires more Semantics to be added.

In a sense, the NL researchers *have* cracked the language understanding problem. But to produce a general Turing-testable system, they would have to provide more and more semantic information, and the program's semantic component would more and more resemble the immense KB mandated by the Breadth Hypothesis.

Have we overstated the argument? Hardly; if anything we have drastically *understated* it! Look at almost any newspaper story, e.g., and attend to how often a word or concept is used in a clearly metaphorical, non-literal sense. Once every few minutes, you might guess? No! Reality is full of surprises. The surprise here is that almost every sentence is packed with metaphors and analogies [Lakoff]. An unbiased sample: here is the first article we saw today (April 7,1987), the lead story in the Wall Street Journal:

'Texaco lost a major ruling in its legal battle with Pennzoil. The Supreme Court dismantled Texaco[1] s protection against having to post a crippling $12 billion appeals bond, pushing Texaco to the brink of a Chapter 11 filing."

Lost? Major? Battle? Dismantled? Posting? Crippling? Pushing? Brink? The example drives home the point that, far from overinflating the need for real world knowledge in language understanding, the usual arguments about disambiguation barely scratch the surface. (Drive? Home? The point? Far? Overinflating? Scratch? Surface? oh no, I can't stop!!!) These layers of analogy and metaphor eventually "bottom out" at physical - somatic - primitives: up, down, forward, back, pain, cold, inside, see, sleep, taste, growth, containment, movement, birth, death, strain, etc.

NL researchers - and dictionaries - usually get around analogic usage by allowing several meanings to a word. Definition #1 for "war" is the literal one, and the other definitions are various common metaphorical uses of "war."

There are many hundreds of thousands - perhaps a few million - things we authors can assume you readers know about the world: the number of tires an auto has; who Reagan is; what happens if you fall asleep when driving - what we called consensus reality. To use language effectively, we select the best consensus image to quickly evoke in your mind the complex thought we want to convey. If our program doesn't already know most of those million shared concepts (experiences, objects, processes, patterns,...), it will be awkward for us to communicate with it in NL.

It is common for NL researchers to acknowledge the need for a large semantic component nowadays; Schank and others were saying similar things a decade ago! But the first serious efforts have only recently begun to try to actually build one (CYC [Lenat 86] and the Japanese Electronic Dictionary Research (EDR) project), so we will have to wait several years until the evidence is in.

4.2. The limits of Machine Learning (Induction)

We will pick on AM and Eurisko because they exemplify the extreme knowledge-rich end of the current ML spectrum. Many experiments in machine learning were performed on them. We had many surprises along the way, and we gained an intuitive feel for how and why heuristics work, for the nature of their power and their brittleness. [Lcnat &Brown] presents many of those surprises.

Despite their relative knowledge-richness, the ultimate limitations of these programs derive from their small size. Not their small number of methods, which were probably adequate, but the small initial knowledge base they had to draw upon. One can analogize to a campfire fire that dies out because it was too small, and too well isolated from nearby trees, to start a major blaze. As Porter recently remarked to us: *Nothing new is learned except with respect to what's already known.* Minsky cites a variant of this relationship in his afterword to [Vinge]: *The more you know, the more (and faster) you can learn.*

Learning can be considered a task. Like other tasks, it is subject to the Knowledge Principle. The inverse of this enabling relationship is a disabling one, and that's what ultimately doomed AM and Eurisko:

> Knowledge Facilitates Learning (Catch 22):
> If you don't know very much to begin with,
> don't expect to learn a lot quickly.

This is the standard criticism of pure Baconian induction. "To get ahead, get a theory." Without one, you'll be lost. It will be difficult (or time-consuming) to determine whether or not each new generalization is going to be useful. This theme is filtering into ML in the form of explanation-based learning and goal-based learning.

Don't human beings violate this Catch, starting from nothing? Maybe, but it's not clear what we start with. Evolution has produced not merely physically sophisticated structures, but also brains whose architecture is well suited to learning many of the simple facts that are worth learning about the world. Other senses, e.g., vision, are carefully tuned as well, to supply the brain with data that is already filtered for meaning: edges, shapes, motion, etc. The exploration of those issues is beyond the scope of this paper, and probably beyond the scope of twentieth century science, but neonatal brains are far from *tabula rasae.*

Besides starting from well-prepared brain structures, humans also have to spend a lot of time learning. It is unclear what processes go on during infancy. Once the child begins to communicate by speaking, *then* we are into the symbolic sort of learning that AI has traditionally focused on.

5. The Empirical Inquiry Hypothesis

We scientists have a view of ourselves as terribly creative, but compared to Nature we suffer from a poverty of the imagination; it is thus much easier for us to uncover than to invent Premature mathematization keeps Nature's surprises hidden. E.g., contrast the astonishing early empirical studies by Piaget (Stages of Development) with his subsequent five decades of barren attempts to mathematize them. This attitude leads to our central methodological hypothesis, our paradigm for AI research:

> Empirical Inquiry Hypothesis (EH): Intelligence is still so poorly understood that Nature still holds most of the important surprises in store for us. So the most profitable way to investigate AI is to embody our hypotheses in programs, and gather data by running the programs. The surprises usually suggest revisions that start the cycle over again. Progress depends on these experiments being able to falsify our hypotheses; i.e., these programs must be capable of behavior not expected by the experimenter.

Early AI programs often surprised their builders: Newell and Simon's LT program and Gelernter's geometry program, circa 1960. Then fascination with axiomatizing and proving set in, and surprises from "the real world" became rare. The inverse to the Empirical Inquiry Hypothesis is cruel:

> If one builds urograms which cannot possibly surprise him/her, then one is using the computer either (a) as an engineering workhorse, or (b) as a fancy sort of word processor (to help articulate one's hypothesis), or, at worst, (c) as a (self-) deceptive device masquerading as an experiment.

Most expert systems work falls into the former category; DART 's use of MRS exemplifies the middle [Genesereth]; PUP5 (by the young Lenat) and HACKER (by the young Sussman) exemplify the latter category.

PUP5 could not avoid synthesizing the one program it was built to synthesize; it "succeeded" but taught us nothing about intelligence. The AM program was the direct result of Lenat's violent recoil away from that methodology. There was no particular target behavior that AM was designed with; rather, it was an experiment: What would happen if a moderate sized body of a few hundred math heuristics were applied to a starting set of 100 simple math concepts. AM provided hundreds of surprises, including many experiments that led to the construction of Eurisko. Eurisko ran for several thousand cpu hours, in half a dozen varied domains. Once again, the ultimate limitation was not what we expected (cpu time), or hoped for (the need to learn new representations), but rather something at once surprising and daunting: the need to have a massive fraction of consensus reality already in the machine. Progress along our path was due to running large experiments:

> Difficult Problems Hypothesis: There are too many ways to solve simple problems. Raising the level and breadth of competence we demand of a system makes it easier to test and raise its intelligence.

Cognitive Psychology, e.g., traditionally sidesteps hard-to-quantify phenomena such as scientific creativity or reading and comprehending a good book, in favor of very simple tasks such as remembering nonsense syllables, tf a "messy" task *is* studied, then usually either (1) it is abstracted and simplified beyond recognition (e.g., BACON), or (2) the psychologist focuses on (and varies) one specific variable, so "respectable" statistical tests for significance can be run.

Much of the confusion in our field may be due to our casual mixing together of two quite different things: AI *goals* and AI *strategies* for achieving those goals. The confusion arises because many entries appear on both lists. But almost any strategy can apply toward any goal. E.g.,
(1) An *expert system strategy* for a *language understanding goal* might be to build a rule based system containing rules like "If a person gets excited, they break more grammatical rules than usual." By contrast,
(2) A *language understanding strategy* for an *expert system goal* might be an English front end that helps an expert enter and edit rules.

All scientific disciplines adopt a paradigm: a list of the problems that are acceptable and worthwhile to tackle, a list of the methods that can and should be tried, and the standards by which the results are judged. Adopting a paradigm is done for reasons of cognitive economy, but each paradigm is one narrow view. Adding to the confusion, some paradigms in AI have grown up both around the various goals *and* around the various strategies!

Finer distinctions can be drawn, involving the *tactical* choices to be made, but this turns out to be misleading. How? Tactics that appear to be superficially different may share a common source of power [Lenat 84]: E.g., predicate calculus and frames both rely on a judicious dividing up of the world. And some tactics which appear superficially similar to each other may draw on very different sources of power (e.g., if/then rules representing logical assertions, versus if/then rules of good guessing.)

The KP and BH and EH are all *strategic* statements. Each could be prefaced by the phrase *"Whichever of the ultimate goals for AI you are pursuing,..."* The strategic level is, apparently, the level where one needs to take a stand. This is rarely stated explicitly, and it is rarely taken into account by news media or by conference organizers.

6. A Programme for AI Research

AI must somehow get to that stage where - as called for by KP and BH -- learning begins to accelerate due to the amount already known. Learning will not be an effective means to get to that stage, unfortunately; we shall have to hand-craft that large "seed" KB one piece at a time. In terms of the graph in Figure 4, all the programs that have ever been written, including AM and Eurisko, lie so far toward the left edge of the x-axis that the learning curve is more or less horizontal. Several of the more successful recent additions to the suite of ML techniques can be interpreted as pushes in the direction of adding more knowledge from which to begin the learning.

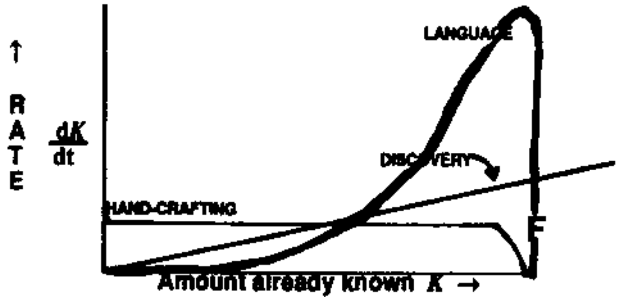

Figure 4. The rate at which one can learn new knowledge. One can also integrate these three curves wrt time, to see how the total amount known might grow over time.

The graph in Figure 3 shows learning by discovery constantly accelerating: the more one knows, the faster one can discover more. Once you speak fluently, learning by talking with other people is more efficient than rediscovery, until you cross the frontier of what humanity already knows (the vertical line at x=F), at which point there is no one to tell you the next piece of knowledge.

By contrast, the rate of hand coding of knowledge is fairly constant, though it, too, drops to zero once we cross the boundary of what is already known by humanity. The hand-coding rate may slope down a bit, since the time to find related concepts will increase perhaps as the log of the size of the KB. Or, instead, the hand-coding rate may slope *up* a bit, since copy & edit is a powerful technique for knowledge entry, and, as the KB grows, there will be more chance that some very similar concept is already present.

This is an example of EH (the Empirical Inquiry Hypothesis presented in section 5): Only by trying to hand-code the KB will we see which of those two counteracting factors outweighs the other, and by how much. Only by continued work on NL and ML will we determine whether or not there is a region, near where all three curves meet, where ML temporarily surpasses NL as a way to grow the KB. Only after our program crosses the frontier F will we find out if the discovery curve begins to slope up or down.

Figure 3 suggests a sweeping three-stage research programme for the coming three decades of AI research:
 * Slowly hand-code a large, broad knowledge base
 * When enough knowledge is present, it will be faster to acquire more through reading, assimilating data bases, etc.
 * To go beyond the frontier of human knowledge, the system will have to rely on learning by discovery, carrying out research and development projects to expand its KB.

Three decades, not three centuries? Yes! The usefulness and timeliness of the BH rests on this quantitative assumption:

> Breadth is Within Our Grasp: A KB of under a million frames will provide a significant performance increase, due to generalization and analogy; this will consume -»2 person-centuries of time, $50 million, and -1 decade. Why such a "small size"? That's about all that people know!

"Under a million things! What an insult!" you may say. "You just argued that the world is a complicated place. Surely we human beings each know an effectively infinite number of things! It's hopeless to try to represent an appreciable fraction of that, so we may as well settle on writing programs that know only 10-1000 specific things."

What goes on during the 200,000 hours between birth and age 21 ? Certainly most of it is spent gathering experiences, building up long term memories; some conscious time (and perhaps some sleep time) is spent generalizing and organizing one's memories. Much of what we're learning is quite specific to our selves, our home, our family, our friends, and our culture. (Humor results from thrusting someone into a different culture; e.g., Crocodile Dundee, A Connecticut Yankee, Beverly Hillbillies, The Gods Must Be Crazy.)

Three recent independent estimates of the number of concepts (frames) needed for full breadth of knowledge all came up with a figure of approximately one million:
(1) CYC: 30,000 articles x 30 frames per article
(2) EDR: 200k words x 1 frame for each of a few languages
(3) Minsky: 4 LTM entries/hour from birth to adulthood

Two other ways for bounding the "bits" a human brain can store lead to much larger numbers: (1) counting neurons and synapses; but its unclear how memories are stored in them; (2) counting pixels in our "mental images"; but controversy rages in Cognitive Psychology over whether mental imagery is just an illusion caused by the consistency and regularity in the world that lets us fill in missing pieces of memories - and of dynamic sensory experiences - with default values. So it's unclear what those larger numbers signify. (Also, though it's clearly an over-simplification, having a million entries means that there can be a trillion one-step inferences involving pairs of them.)

Here again is a case where various theories give various estimates, and the way to settle the issue - and, perhaps, achieve the goal of having the KB we want - is to go off and try to build the large KB. Along the way, it will no doubt become clear how big it is growing and what the actual obstacles are that must be overcome.

7. Differences with Other Positions

7.1. Our position regarding the Aesthetes

People do prefer -- and *should* prefer - the simplest consistent hypothesis about any phenomenon. That doesn't make them correct, of course. A geocentric cosmology *is* the proper one to have, until data to the contrary piles up. Chapter 1 of [Lakatos] presents the historical series of mathematicians' retreats from the initial form of the Euler-Descartes conjecture to increasingly longer, less elegant versions, with more and more terms to take obscure cases and exceptions into account

If only there were a Secret Ingredient for intelligence - the Maxwell's Equations of Thought. If only we could axiomatize the world, and deduce everything. If only our learning program could start from scratch. If only our neural net were big enough. If only the world were like that. But it isn't. The evidence indicates that almost all the power is in the bulk knowledge. As Whitehead remarked, *"God is in the details."*

This difference about the elegance and simplicity (or not) of the world leads to a deep methodological difference between our way of doing AI and the aesthetes'. Following the Difficult Problems Hypothesis, we are firmly convinced that the AI researcher must make a major time commitment to the domain(s) in which his/her programs are to be competent; e.g., the two years that Stefik and Friedland spent to learn about molecular biology before doing Molgcn; the decade-long time frame for CYC [Lenat *et al,* 86].

We may be exhausting the range of potent experimental AI theses that can be carried out in two years, by a student starting more or less from scratch; witness the trend to give the Computers and Thought Award to increasingly less recent graduates. The presence of a large, widely-accessable "testbed" KB should enable a new round of important theses.

Many AI researchers quest for an elegant solution in a desperate desire for scientific respectability. The name of our field - "Artificial Intelligence" - invites everyone to instantly form an opinion. Too bad it wasn't called quantum cognodynamics. "But perhaps, by interposing a layer of mathematical formalism, we can come to be accepted as hard scientists." Hence the physics-envy!

Formalizing has never driven any early science along. In designing new drug molecules, the biochemist knows it's too inefficient to apply Schroedinger's wave equation to compute the energy minimizations, hence from his/her point of view the fact that such a deep understanding even exists *is irrelevant* to the task at hand. S/he relies on crude design heuristics, and the drug companies using this methodology occasionaly are enriched. As Minsky remarked about the A* algorithm in 1970: *"Just because it's mathematical doesn't mean it deserves to be taught."*

Eventually, we *will* want layers of increasing "neatness." E.g., in physics, students learn each year that last year's equations were a special case. We always try to reason at the highest, most superficial, most efficient level at which we can, and delve down one level deeper when we are forced to. But devoting so much effort to the attempt at "neatness" today just drains time and intellectual energy away from the prime enterprise of the field.

One popular type of aestheticism that deserves mention is the trend to highly parallel (e.g., connectionistic) and ever faster devices. The trouble is that most difficult tasks in knowledge-rich areas can't be highly parallelized. If we were to set a million people to work trying to find a cure for cancer, we wouldn't find one in .2 seconds. Each cancer experiment, takes months or years to perform, and there are only a moderate number of promising experiments to do at any one time; their *results* will determine what the next round of promising experiments should be.

Parallelism is useful at one extreme for implementing very carefully engineered algorithms (e.g., Systolic algorithms), and at the other extreme for allowing a community of meaningfully-individuated agents act independently, asynchronously. For most technical tasks, until we understand the task very well, the size of such an actor community that we can design is typically only -100.

The time to perform a task often increases exponentially with its size (e.g., looking ahead *n* moves in chess.) Taking a microcoding approach or a parallelizing approach cuts off a constant factor; taking a knowledge based approach may add a constant overhead but more importantly, for the long run, it may chip at the *exponent.* Cf. Figure 2 again. On the other hand, it is worth remarking that there are some special tasks where the desired level of performance (x-coordinate) is fixed: beating all humans at chess, understanding spoken words in real time, tracking the space shuttle in real time, etc. In such a case, getting a large enough constant factor speedup really could solve the problem, with no need to apply the KP, BH, or EH. As our ambition to attack ever more difficult problems grows, though, the exponential nature of the search hurts worse.

7.2. Our position regarding Expert Systems

The KP underlies the current explosion of work on expert systems. Still, there are additional things our position argues for, that arc not yet realized in today's ES.

One major power source for ES, the reason they can be so readily constructed, is the synergistic additivity of many rules. Using a Blackboard [Erman *et al.]* or partitioned rule sets, it is possible to combine small packets of rules into mega-rules: knowledge sources for one large expert system.

The analogue at the next higher level would be to hook hundreds of large ES together, and achieve even greater synergy. That dream fails to materialize. As we increase the domain of each "element" we are trying to couple together, the "glue" we need gets to be larger and more sophisticated. It seems to us that it will require the large system mandated by the Breadth Hypothesis, before the true potential of ES technology will be realized.

> Plateau-hopping Requires Breadth: To couple together a large number of disparate expert systems will require something approaching full consensus reality - the million abstractions, models, facts, rules of thumb, representations, etc, that we all possess and that we assume everyone else does.

As we try to combine ES from various tasks, even somewhat related tasks, their particular simplifications and idiosyncracies prevent synergy. The simplifying was done in the interests of highly efficient and competent problem solving; breadth was not one of the engineering goals.

This naturally results in each ES being a separate, simplified, knowledge universe. When you sit down to build an ES for a task -- say scheduling machines on a factory floor -- you talk to the experts and find out the compiled knowledge they use, the ways they finesse things. For instance, how do they avoid general reasoning about time and belief? Probably they have a very simple, very specialized data structure that captures just the bits of information about time and belief that they need to solve their task. How do they deal with the fact that this milling machine M has a precise location, relative to all the others; that its base plate is a solid slab of metal of such and such a size and orientation; that its operator is a human; that only one operator at a time can use it; etc.?

If someone accidentally drills a hole through the base plate, most human beings would realize that the machine can still be used for certain jobs but not for others - e.g., it's OK if you want to mill a very large part, but not a very small one that might fall through the hole! People can fluidly shift to the next more detailed grain size, to reason out the impact of the hole in the base plate, even if they've never thought of it happening before; but the typical ES would have had just one particular level built in to it, so it couldn't adapt to using the crippled milling machine.

A remark that is appropriate both to ES and to the Logicians (Section 7.1) is that there is no need - and probably not even any possibility -- of achieving a global consistent unification of a large set of ES' KBs. Large, broad systems need *local* consistency -- what we call *coherence*. E.g., physics advanced for many decades with inconsistent particle and wave models for light. Knowledge space *in toto* is still a set of self-supporting buttes. In a coherent system, inferring an inconsistency is only slightly more serious than the usual sort of "dead-end"; the system should still just back up a bit and continue on.

8. Problems and Solutions

Problem 1: Possible "in-principle" Limitations. There are several extremes that one can point to where the Knowledge Principle and Breadth Hypothesis would be inapplicable or even harmful: perceptual and motor tasks; certain tasks which must be performed in small pieces of real time; tasks that involve things we don't yet know how to represent well (time, space, belief, mass nouns, counterfactuals,...); tasks for which an adequate algorithm exists; tasks so poorly understood that no one can do it well yet; and tasks involving large amounts of common sense.

Our response ~ in principle and in CYC - is to describe perception, emotion, motion, etc., down to some level of detail that enables the system to understand humans doing those things, and/or to be able to reason simply about them. As discussed under problem 2, below, we let a large body of examples dictate what sorts of knowledge, and to what depth, are required.

A similar answer applies to all the items which we don't yet know very clearly now to represent. In building CYC, e.g., a large amount of effort is being spent on capturing an adequate body of knowledge (including representations) for rime, space, and belief. We did not set out to do this, the effort is driven completely by need, empirically: looking at encyclopedia and newspaper articles, and developing machinery that can handle those cases encountered.

Tasks which can be done without knowledge, or which require some that no one yet possesses, should be shied away from. One does not use a hammer type with.

The huge KB mandated by the Breadth Hypothesis is AI's "mattress in the road". Knowing that we can go around it one more time, AI researchers build a system in six months that will perform adequately on a narrow version of task X; they don't pause for a decade to pull the mattress away. This research opportunity is finally being pursued; but until CYC or a similar project succeeds, the knowledge based approach must shy away from tasks that involve a great deal of wide-ranging common sense or analogy.

The remainder of the problems in this section are primarily pragmatic, engineering problems, dealing with the mechanics of constructing systems and making them more usable. As can be seen from our response to the in-principle limitations, we personally view Problem 1 in that very same category! That is a view based on the EH, of course.

Problem 2: How exactly do we get the knowledge? Knowledge must be extracted from people, from data bases, from the intelligent systems' KBs themselves (e.g., thinking up new analogies), and from Nature directly. Each source of knowledge requires its own special extraction methods.

In the case of the CYC project, the goal is to capture the full breadth of human knowledge. To drive that acquisition task, Lenat and his team are going through an encyclopedia, sentence by sentence. They aren't just entering the facts stated, but - much more importantly - are encoding what the writer of that sentence assumed the reader already knew about the world. They are the facts and heuristics which one would need in order to understand the sentence, things which would be insulting or confusing for the writer to have actually stated explicitly (e.g., if coke is consumed to turn ore into metal, then coke and ore must both be worth less than metal.) They also generalize each of these as much as possible (e.g., the products of commercial processes are more valuable than their inputs.) Another useful place they focus is the inter-sentential gap: in a historical article, what actions should the reader infer have happened between each sentence and the next one? Yet another focus: what questions should anyone be able to answer having just read that article? These foci drive the extraction process. Eventually, CYC itself should help add knowledge, e.g., by proposing analogues, extending existing analogies, and noticing gaps in nearly-symmetric structures.

This methodology will collect, e.g., all the facts and heuristics about Water that every article in the encyclopedia assumed its reader already knew; we expect this will be close to what everyone does know and needs to know about Water. This is in contrast to, for instance, Naive Physics and other approaches that aim to somehow capture a deeper theory of Water in all its various forms.

Problem 3: How do we adequately represent it? Human experts choose or devise representations that enable the significant features of the problem to remain distinguished, for the relevant connections to be quickly found, etc. Thus, one can reduce this to a special case of Problem 2, and try to elicit appropriate representations from human experts. CYC takes a pragmatic approach: when something proves awkward to represent, add new kinds of slots to make it compactly representable.

Problem 4: How will it be used? The representation chosen will of course impact on what inference methods are easy or difficult to implement. Our inclination is again to apply EH: when you find out that some kind of operation needs to be performed often, but it's very inefficient, then you need to adjust the representation, or the inference methods available, or both. As with Problem 3, there is a temptation to early specialization: it is a local optimum, like swerving around a mattress in the road. Pulling this mattress aside means assembling a large repertoire of reasoning methods, and heuristics for choosing, monitoring, and switching among them. Earlier, we sketched an opportunistic (non-monolithic) control structure which utilizes items in the control-strategy region of the KB.

To take a more specific version of this question: how do we expect to efficiently "index" -- find relevant partial matches? Our answer is to finesse it for now. Wait until our programs *are* finding many, far-flung analogies, e.g., but only through large searches. Then investigate what additional knowledge *people* bring to bear, to eliminate large parts of the search space in those cases. Codify the knowledge so extracted, and add it to the system. This is a combined application of the Difficult Problems Hypothesis and the EH. It is a claim that the true nature of the indexing problem will only become apparent in the context of a large problem running in an already very large KB.

Problem 5: How can someone interact "naturally" with KB systems? Knowledge based systems built so far share with their knowledge-free predecessors an intolerant rigidity of stylistic expression, vocabulary, and concepts. They rarely accept synonyms and pronouns, never metaphors, and only acknowledge users willing to wear a rigid grammatical straitjacket. The coming few years should witness the emergence of systems which begin to overcome this problem. As is only fitting, they will overcome it with knowledge: knowledge of the user, of the system's domain, of discourse, of metaphor. They will employ pictures and sound as well as text, as means of input and output. Many individual projects (Oncocin, CYC) and expert system tools (ART, KEE) are already moving in this direction.

Problem 6: How can you combine several enterersVsystems' knowledge? One solution is to sequentialize the entry, but it's not a good solution. Many Emycin-based programs designated someone to be the knowledge base czar, with whom all the other experts would discuss the knowledge to be entered. Eurisko, built on RLL, tried *explicitly* enforced semantics. Each slot would have a description of its intended use, constraints that could be checked statically or dynamically (e.g., each rule's If-maybe-relevant slot should take less cpu time to execute than its If-truly-relevant slot). When someone enters rules that violate that constraint, the system can complain to them, to get everyone back on track using the same semantics again. CYC extends this to *implicitly* enforced semantics: having such a large existing KB that copy&edit is the clear favorite way of entering new knowledge. When one copies&edits an existing frame, virtually all of its slots' semantics (and even most of their values!) carry right over.

We have already described the important "gluing" role that coherence (local consistency) will play. At a more exotic level, one can imagine mental immune systems providing (in the background) constant cross-checking, healthy skepticism, advice, and criticism.

Problem 7: How can the system builder, and the system user, not get lost? "Getting lost" is probably the right metaphor to extend here, because what they need to do is to successfully navigate their way through knowledge space, to find and/or extend the relevant parts. Many systems, including CYC, are experimenting with various exploration metaphors and orientation tools: helicoptering through semantic nets; exploring a museum with Alician entry into display cases and posters, etc. For more elaborately scripted interface metaphors, see [Vinge] or [Lenat 84b]. The latter suggests clip-on filters to shade or highlight certain aspects of what was seen; models of groups and each individual user; and simulated tour-guides with distinct personalities.

Problem 8: How big a fraction of the million pieces of "consensus reality" do you need to represent? We believe the answer is around 20-50%. Why? When communicating with an intelligent entity, having chosen some concept X, we would expect the "listener" to be familiar with X; if it fails several times in a row ~ often! - then it is missing too much of consensus reality. A similar argument applies to analogizing, and to generalizing. Now to have a 30% chance for the chosen analogue to be already known by the listener, he/she/it might have to know 30% of the concepts that are analogized to. But how *uniformly* are good analogues distributed in concept-space? Lacking more data, we assume that they are uniformly distributed, which means the system should embody 30% of the full corpus of consensus reality. The distribution is quite possibly *not* uniform, which is why (the EH again) we need to build the KB and see.

9. Conclusion: Beyond Local Maxima

Our position includes the statements that

- One must include *domain-specific* knowledge to solve difficult problems effectively
- One must also include both *very general* knowledge (to fall back on) and very *wide-ranging* knowledge (to analogize to), to cope with novel situations.
- We already have plenty of theories about mechanisms of intelligence; we need to proceed empirically: go off and build large testbeds for performing, analogizing, ML, NL...
- Despite the progress in learning, language understanding, and other areas of AI, *hand-crafting* is still the fastest way to get the knowledge into the program in the 80's.
- With a large KB of facts, heuristics, and methods, the fastest way would tip toward NL, and then toward ML.
- The hand-crafting and language-based learning phases may each take about one decade, culminating in a system with human-level breadth and depth of knowledge.

Each of those statements is more strongly believed than the one following it. There is overwhelming evidence for the KP and EH. There is strong evidence in favor of the BH. There is a moderate basis for our three-stage programme. And there is suggestive evidence that it may be possible to carry out the programme this century.

The Knowledge Principle is a mandate for humanity to concretize the knowledge used in solving hard problems in various fields. This *might* lead to faster training based on explicit knowledge rather than apprenticeships. It has *already* led to over a thousand profitable expert systems.

The Breadth Hypothesis is a mandate to spend the resources necessary to construct one immense knowledge base. It should extend horizontally across an encyclopedic span of human thought and experience, and vertically extend upward from a moderate level of detail all the way up to encompass the most sweeping generalities.

As a partial application of the Breadth Hypothesis, consider the task of building a knowledge based system covering most of engineering design. Interestingly, this task was chosen independently by the Japanese EDR project and by Bob Kahn's National Research Institute. Both groups see this task as a moderate-term (-1994) goal. It is certainly much broader than any single expert system, yet much narrower than the universal knowledge base mandated by the BH.

Slightly narrower "Lawyers' workstations" or "Botanists' workstations", etc., are similar sorts of compromises (partial applications of BH) worth working on. They would possess a crown of very general knowledge, plus their specific field's next level of generalities, useful representations, etc., and some detailed knowledge including, e.g., methods for extracting and using entries in that field's online databases. These have the nice side effect of enticing the experts to use them, and then modify them and expand them.

We are betting our professional lives, the few decades of useful research we have left in us, on KP, BH, and EH. That's a scary thought, but one has to place one's bets somewhere, in Science. It's especially scary because
(a) the hypotheses are not obvious to most AI researchers
(b) they are unpalatable even to us, their advocates!

Why are they not obvious? Most AI research focuses on very small problems, attacking them with machinery (both hardware and search methods) that overpower them. The end result is a program that "succeeds" with very little knowledge, and so KP, BH, and EH are irrelevant. One is led to them only by tackling problems in difficult "real" areas, with the world able to surprise and falsify.

Why are our three hypotheses not particularly palatable? Because they are unacsthetic! Until forced to them, Occam's Razor encourages us to theorize more elegant solutions.

Section 8 listed several limitations and problems. We do not sec any of them as insurmountable. The biggest hurdle has already been put well behind us: the enormous local maximum of *knowledge-free* systems; on the far side of that hill we found a much larger payoff, namely expert systems.

And yet we see expert systems technology, too, as just a local maximum. AI is finally beginning to move on beyond that threshold. This paper has presented what its authors glimpse on the far side of the expert systems hill: the promise of very large scale knowledge bases (VLSK), the promise of analogical reasoning and common sense knowledge.

The impact of systems mandated by the KP and BH cannot be overestimated. Public education, e.g., is predicated on the unavailability of an intelligent, competent tutor for each individual for each hour of their life. AI will change that. Our present entertainment industry is built largely on passive viewing. AI will turn "viewers" into "doers". What will happen to society as the cost of wisdom declines, and society routinely applies the best of what it knows? Will a *knowledge utility* arise, like the electric utility, and how might it (and other AI infrastructures) effect what will be economically affordable for personal use?

Man-Machine Synergy Prediction: In that "second era" of knowledge systems, the "system" will be reconceptualized as a kind of colleagular relationship between intelligent computer agents and intelligent people. Each will perform the tasks that he/she/it does best, and the intelligence of the system will be an emergent of the collaboration.

The interaction may be sufficiently seamless and natural that it will hardly matter to anyone which skills, which knowledge and which ideas resided where (in the head of the person or the knowledge structures of the computer.) It would be inaccurate to identify Intelligence, then, as being "in the program". From such man-machine systems will emerge intelligence and competence surpassing the unaided human's. Beyond that threshold, in turn, lie wonders which we (as unaided humans) literally cannot today imagine.

Acknowledgements

Many of the ideas in this paper emerged during a series of meetings with Mark Stefik, Marvin Minsky, Alan Kay, Bob Kahn, Bob Wilensky, Ron Brachman, and several others. The careful reader will also detect the welcome influence of many of our colleagues on our way of thinking: Woody Bledose, John Brown, John McDermott, Allen Newell, George Polya, Roger Schank, and Herbert Simon. Chuck Dement, Carl Hewitt, David Kirsch, and Bruce Porter provided very useful examples and critiques.

References

Buchanan, B.G., G. Sutherland, and E. Feigenbaum, "Heuristic Dendral: A Program for Generating Explanatory Hypotheses in Organic Chemistry"; in (Meltzer and Michie, eds.) Machine Intelligence 4, American Elsevier, New York, 1969, pp. 209-254.

EDR (Japan Electronic Dictionary Research Institute, LTD), Tokyo, Japan, personal communication, February 1987.

Erman, L., F. Hayes-Roth, V. Lesser, and D. Raj Reddy, "Hearsay-II Speech Understanding System," Comp.. Surv., 12,2,1980,224-225.

Feigenbaum, E., "The Art of Artificial Intelligence: Themes and Case Studies in Knowledge Engineering , UCAI-5,1977.

Genesereth, M. R., "The use of design descriptions in automated diagnosis", Artificial Intelligence 24,1984, pp. 411-436.

Lakatos, Imre, *Proofs and Refutations,* Cambridge TJ. Press, 1976.

Lakoff, G., & M. Johnson, *Metaphors We Live By,* U. Chicago, 1980.

Lenat, D. & J. S. Brown, "Why AM and Eurisko Appear to Work," Artificial Intelligence 23,1983, pp. 269-294.

Lenat, D., "Computer Software for Intelligent Systems," Scientific American 251, September 1984, pp. 204-213.

Lenat, D., A. Borning, D. McDonald, C. Taylor, and S. Weyer, "Knoesphere: ES with Encyclopedic Knowledge," UCAI-8,1984b.

Lenat, D., M. Shepherd, and M. Prakash, "CYC: Using Common Sense Knowledge to Overcome Brittleness and Knowledge Acquisition Bottlenecks," AI Magazine, Winter, 1986, pp. 65-84.

Minsky, M., *The Society ofMinad* Simon and Schuster, NY, 1985.

Newell, A., and H. A. Simon, *Human Problem Solving,* Prentice-Hall, Englewood Cliffs, N J., 1972.

Newell, A., "The Knowledge Level", AI Mag., Spring, 1980,1-4.

Stefik, M.. "The Knowledge Medium", AI Mag., Spring 1986,34-46.

Suisman, G. J., *A Computer Model of Skill Acquisition,* American Elsevier Publishing, Inc., New York, 1975.

Vinge, Verner, *True Names,* Bluejay, New York, 1984.

"What's News", Wall Street Journal, LXXIX, 65, Apr. 3&7,1987, p.l