The binding roots of symbolic AI: a brief review of the Cyc project

Deniz Yuret
MIT Artificial Intelligence Laboratory
545 Technology Square, Room 815
Cambridge, MA 02139, USA
e-mail: deniz@mit.edu

February 13, 1996

Abstract

Cyc is a monumental but controversial research effort for codifying the human consensus knowledge initiated by Doug Lenat in 1984. The human consensus knowledge is meant to be the background knowledge that a human is assumed to possess in order to understand, for example, newspaper articles or encyclopedia entries. The methodology chosen is to enter this knowledge explicitly in a large knowledge base. A team of knowledge enterers have been actively engaged in this process since the onset of the project. It is a multi-million dollar, decade-long, two person-century effort. The engineering goal of Cyc is to overcome the brittleness of conventional application programs by letting them fall back on background knowledge. The scientific goal of Cyc is to build a system that would exhibit human level common sense and understanding. I believe that Cyc has a respectable goal but inadequate methodology. I further believe that this inadequacy comes from insisting on three limiting principles: (1) representing knowledge explicitly, (2) representing knowledge in a single uniform framework, and (3) insisting on deduction as the main inference engine.

1 Introduction

The view of a typical AI researcher on the Cyc project seems to be

> "Cyc is generally viewed as a failed project. The basic idea of typing in a lot of knowledge is interesting but their knowledge representation technology seems poor."

R. V. Guha, who was the co-leader of the project with Lenat for years and the co-author of the book *Building large knowledge based systems* [Lenat and Guha, 1990] left the team in 1994. He is quoted saying in an interview [Stipp, 1995]

> "We were killing ourselves trying to create a pale shadow of what had been promised. Cyc may prove useful in commercial applications like data mining, but the goal of creating a system that would exhibit real common sense failed."

Lenat, who recently founded the privately-owned company Cycorp to continue the development of the Cyc project, still seems optimistic. His view is [Lenat, 1996]

> "The development of Cyc was a very long-term, high-risk gamble that has begun to pay off. Begun as a research project in 1984, Cyc is now a working technology with applications to many real-world business problems. Cyc's vast knowledge base enables it to perform well at tasks that are beyond the capabilities of other software technologies. The applications currently available, or in development include: natural-language processing, integration of heterogeneous databases, knowledge-enhanced retrieval of captioned information, distributed ai, www information retrieval..."

In this review I will try to discuss whether or not Cyc actually failed, and whether its basic approach has any hope of reaching human level intelligence. The disturbing thing about reviews such as this one, or in general papers about competing approaches to AI, is that the arguments never seem sound and conclusive. The reader should expect this as a natural outcome, though. The reason why there exists competing approaches in our field is that they cannot be dismissed by simple deductive reasoning. Each counter-example introduced for a methodology, can typically be met by a small patch. After any number of patches, another counter-example can be found. Such as the nature of things, how can one argue objectively for or against an approach? Below are some possible types of arguments:

Doability Argument Because almost every small machine we come up with happens to be Turing complete, it is meaningless to argue about the *impossibility* of achieving intelligence with a certain approach . However softer claims can be made such as "I will get there before you will." It is difficult to defend such claims though.

Naturality Argument Because the only instances of human level intelligence we see in the world are humans, it can be argued that we should try to build things like them. Although this argument seems to be very appealing to some people, it is repulsive for others, who have two defenses: (1) We don't know enough about the brain, who knows if the evidence can't be interpreted in some other way, and (2) We are interested in the concept of intelligence in general, not just some particular implementation of it. This topic seems to have a religious attribute, thus is not very useful in arguments.

Definitional Argument A common hack in artillery troops, to improve the hit statistics is to first shoot, and then determine the target. In our case, the target is intelligence, so it is common practice to change its definition to match the performance of one's own approach. This argument is hardly convincing though, the ultimate comparison point will be humans after all.

Pragmatic Argument This is the type of argument I find most objectively appealing. If an algorithm is proven to run faster than another, I have no problem adopting it. If a representation is more expressive than some other, the former can represent some things that the latter can not. A lot of problems attacked since the beginning of the field have been shown to be intractable. Thus the possibility of finding "the single algorithm" to learn, "the single representation" to think etc. is a dream. Still, intelligent beings have a way of solving intractable problems in the cases where it matters the most. I believe if we focus on our natural constraints, i.e. performing in real time, focusing on the relevant issue in the presence of a large amount of knowledge, deciding on "intuitively obvious" things in constant time, whereas being painfully slow in others, we can find the necessary clues to build systems that share our performance profile. Although this doesn't seem like the approach history directs us to take, keep in mind that most philosophers that have inspired AI research did not have the opportunity to learn complexity theory, or to imagine the existence of machines that can manipulate the most powerful data structures imaginable.

As a final point of this section, I would like to state my sympathy for Cyc. I believe that an hour of work is more rewarding than a thousand pages of argument. Thus I admire the attempt of launching such a grand scale project in contrast with a lot of other projects that produce more papers than pages of code. Typically, all that is left from small projects is their papers. The code gets lost in the researcher's files, and nothing can be built on top of it. Large scale work can be cumulative. Even if Cyc is built on a flawed methodology, maybe one day the machine with the right methodology can read its knowledge base like a book, to learn/verify its own knowledge. I happen to agree with Lenat's complaint [Stipp, 1995]

> "... Hal-like computers would fundamentally change the whole nature of existence. But AI is filled with little bump-on-a-log projects that couldn't possibly lead there."

The next section is going to describe the goal of Cyc and provide evidence for why it is a useful goal. Section three will describe Cyc's proposal to achieve this goal. Section four discusses the application of this proposal. It describes the evolution of Cyc during its twelve year history, and presents various design decisions made by Cyc and alternatives. Section five will focus on the historical roots of Cyc's methodology. The claim is that the reason behind Cyc's failures is not the particular design decisions made, but the basic principles Cyc inherits from the classical AI tradition. Section six elaborates on these principles, why it is necessary to go beyond them, and how to go there. The last section gives a short summary and concludes with a list of principles for building the next Cyc.

2 The problem statement of Cyc

Consider a toy deduction system, e.g. ZOOKEEPER from [Winston, 1992]. It has rules such as the following:

If *?x* Flies
?x LaysEggs
then *?x* is a Bird

It is a trivial fact that such systems don't know the meaning of the symbols that make up their sentences. So, for example, the ZOOKEEPER cannot answer questions such as:

Does *?x* touch the ground while it flies?
Where are the eggs before *?x* lays them?
even
Does *?x* lay eggs?

These questions can be answered by any five year-old. They do not require any extra knowledge beyond the simple comprehension of the meanings of the terms such as "fly" or "lay". So failing this simple test, we can say that the ZOOKEEPER does not understand what these terms mean. Actually for all it is concerned, the rule could have been:

If *?x* syvrf
?x ynlfrttf
then *?x* vf n oveq

And looking from its perspective, you wouldn't have any advantage over the ZOOKEEPER to answer questions such as:

Qbrf *?x* gbhpu gur tebhaq juvyr vg syvrf?
Jurer ner gur rttf orsber *?x* ynlf gurz?

However, you could, in principle, manipulate the symbols of this language, to make the inferences ZOOKEEPER makes. Am I repeating the Chinese room argument? [Searle, 1980] No. I am just stating a simple fact that this small system doesn't have any understanding of the symbols it is using.

What would a program be able to do, if it did have any understanding of its symbols? The Cyc book [Lenat and Guha, 1990] has many nice examples:

Brittleness One can imagine many examples of hypothetical situations where expert systems fall into ridiculous situations like deciding that a man is pregnant, or believing that a 20 year old has been working for 22 years. The bottom line of all these examples is that the meanings of the symbols are in the mind of the programmer, thus the constraints about these meanings are also in there. You cannot predict every possible circumstance that your program is going to get into, thus there will be some constraints that you forget to encode.

Communication Different programs can encode the same information in different ways. Typically their symbol names will not match. Even if they represent the same value, they may represent it in different ways. Consider three medical diagnosis programs that represent the parameter fever in different ways: the first one uses a scale of "low, medium, high", the second one uses "slight, moderate, extreme", and the last one uses numerical values. A human being looking at these programs can understand that these refer to the same scale, and convert between them easily, because he has access to the meaning of these symbols. The programs, on the other hand, will have a hard time sharing knowledge.

Language A lot of hard problems in language just cannot be solved syntactically. It comes down to understanding the meanings of words. Consider these typical examples of semantic disambiguation:

> Fred saw the plane flying over Zurich.
> Fred saw the mountains flying over Zurich.
>
> The police arrested the demonstrators because they feared violence.
> The police arrested the demonstrators because they advocated violence.

Knowledge retrieval Current indexing and keyword search techniques provide only a limited performance in accessing heterogeneous knowledge sources. A more advanced approach is to annotate the opaque information by simple sentences and fetch the piece of data whose annotation matches a query. This is still fairly limited in the following sense: Consider a picture in a library of news photographs, annotated "A soldier holding a gun to a woman's head." It is conceivable to retrieve this photograph by a query such as: "Show me the picture of someone holding a gun." But to fetch the same picture based on queries like: "Show me a picture of a frightened person", or "a man threatening a woman," requires knowing about the meaning of the symbols.

What do the symbols of our current programs mean? Think about the link between the symbol "fly" and the action of flying in the real world. The only connection seems to be the human who is using or writing the program. He is responsible from coming up with this concept in the first place, and then learning the assignment of this particular sign to it. In other words, the symbols in systems such as ZOOKEEPER take their meanings in the eye of the observer.

Cyc's goal is to provide a solution to exactly this problem. It is assumed that if enough background knowledge is explicitly collected, programs can use this source to answer trivial questions and make obvious inferences about their own symbols.

3 The proposal of Cyc

So the question is, what would it take a system to "understand" the meanings of symbols in the sense described above? What would we have to do in order to get ZOOKEEPER to answer obvious questions about "flying" and "laying eggs"?

It won't do to patch the system with a specific rule every time we come across another question. There are too many questions one can come up with. It is impractical to assume a system that carries the answers to every possible question it can face. Imagine a calculator that had to store the answers for all possible operations in its ROM. It wouldn't be cost effective, size effective, plus, within its life time only a tiny percent of it would get used.

So, presumably, intelligent systems carry a relatively small portion of "what they know" in their memory (which I will call the core), and then have some mechanisms to extend this core to the full radius of their knowledge. Here, then, I define "knowing" to be either explicitly having some fact in the memory, or being able to infer it so efficiently that we cannot distinguish it from the facts we have in our memory. I claim that, most of the time when I ask you obvious questions like the above, you just feel like you "know" the answers, and can't always tell whether you are pulling them out of the memory or subconsciously inferring them with lightning speed.

Then the two problems to be solved are (1) What is going to be carried around in your memory, and (2) How will you reach from this core to the full circle of "what you know." Different approaches to representation are indeed distinguished by the different answers they give to these questions.

Cyc's proposal is a straightforward one. We realize that the system does not understand some concept like "LaysEggs". We conclude that in order to understand such a concept, the system has to understand lower level concepts like "lays" and "eggs". To understand "lays", it probably needs to know about containers, and inside/outside relationships. To understand "eggs", it needs to know something about living things, the concept of crust vs interior, fragility, and objects in general. Going down like this, we define every concept in terms of the lower level ones. We explicitly assign a symbol to every concept, and connect these symbols via various relationships. So, if a question cannot be answered by a direct match in our knowledge base, i.e. if it is not in our core knowledge, then we can deduce the answer by walking down the chain of relationships to more primitive concepts.

4 The implementation of Cyc

In this section I will go through the twelve year evolution of Cyc, how the proposal was carried out. This will bring interesting issues about knowledge representation in general. I will distinguish between four phases in this evolution. The period of the project getting off the ground during 1984-1986 as depicted in [Lenat et al., 1986], the state of Cyc as it is depicted in the book mostly written before 1990 [Lenat and Guha, 1990], the midterm report [Guha and Lenat, 1990] and their AI journal discussion of the methodology [Lenat and Feigenbaum, 199 Smith, 1991], and the current state of affairs in 1996[Lenat, 1996]. I would like to have played with the system myself, to get an objective view of the performance. However the bureaucracy has been holding down my efforts for the last four months. All my observations, thus, are based on reading reviews. I will trace the evolution in three dimensions: content, representation and modularity.

4.1 Content

When Cyc was first envisioned, the idea was to start by encoding the contents of a one volume desk encyclopedia, hence the name Cyc. Issues such as the representation of time, substances, agents, and causality were largely ignored in favor of highly specific issues. However, soon after, the team realized that it wasn't the contents of the encyclopedia that was urgent, it was the complement of it, i.e. the underlying common sense knowledge that the writers of encyclopedias and newspaper articles assumed that their readers already possessed. This seems like a step in the right direction. However, as I will argue below, it wasn't a sufficient step. Other reviewers [McDermott, 1993, Neches, 1993] are also skeptical about the premise that what is holding AI back is programs' lack of explicitly represented knowledge. McDermott comments in his usual bold style that "you can't make the problems go away by burying them under a truckload of frames."

Throughout the articles and books of the Cyc team, there is an implicit dream that shows itself in random places. For example in the middle of a section about how many entries Cyc will need, they say

> "... probably another ten to fifty million entries would suffice for general intelligence (for example, the intelligence required for acquiring knowledge in school and in extra-curricular conversations). Quite a bit more may be needed for qualitatively super-human intelligence."[Lenat and Guha, 1990]

This is not a proposition that they officially defend, that Cyc is going to encompass general intelligence when loaded with enough knowledge. But every once in a while, you can see glimpses of the idea that makes you feel that this is their background hope.

A related hope is their anticipation of a kind of crossover [Guha and Lenat, 1990] from manual knowledge entry to primarily automatic entry via natural language understanding. It is observed that the role of human knowledge enterers today, manually entering assertion after assertion, is akin to teachers who must instruct by surgically manipulating brains. Reaching the crossover would mean that the role of humans would become much more like tutors, with a great deal of question answering going on in both directions. Cyc would start reading books and newspapers itself, initiating a dialogue with humans to clarify ambiguous points.

When I heard about the crossover point, and the quest for knowledge supporting general intelligence, I thought what they would focus on is the knowledge of a five year-old, or the knowledge and inference capabilities that seem to be common to human species throughout all world cultures. I thought they were after constructing an infrastructure, not the building itself. Apparently, I was wrong. Here is a typical example of a Cyc axiom [Guha and Lenat, 1993]

```
(#%ist #%LargeCorpInternalsMt
  (#%ForAll x (#%HumanResourcesDepartment #%allInstances)
    (#%actsInCapacity x #%mediatorInProcesses
      #%EmployeeHiring #%MainFunction)))
```

Translated into English, it says that the human resources department of a company plays the primary role of mediating the hiring of employees.

Certainly most of the Australian aboriginals, for example, probably haven't heard about large corporations. A five year-old might not know what a vice-president or a house speaker is. I certainly did not have much knowledge about what kind of a game baseball was before I came to this country, and I still don't understand how it works. But we are all considered intelligent. We have the capacity to learn about corporations, politics or sports. In addition, we all have some kind of common-sense (the kind that is common to all of us), i.e. we can all tell that a flying bird does not touch the ground, that after I enter a room I am inside the room, if I was able to enter, then probably the room was bigger than me etc. etc. We will all be able to answer a question such as "Can a lone king reach every square of an empty chess board?" after given a brief description of the rules. We will all have a hard time answering the same question for a knight [McAllester, 1991].

An argument can be made for this kind of common-sense reflecting the infrastructure for intelligence. An argument cannot be made for knowledge about large corporation internals being a necessary prerequisite for intelligence. That is static encyclopedic knowledge. If Cyc ever reaches the "crossover point", it can read about these things later. I presume that the pressure from the large corporations which were supporting the Cyc project forced the team to focus on such issues prematurely. It is unfortunate that this was at the expense of handling the infrastructure in an adequate way.

4.2 Representation

One thing that evolved significantly during the development of Cyc was the form of its symbolic representations. For the first five years there was an emphasis on frame based representations and almost a reaction to the findings of the formal community (as illustrated by the "neats versus scruffies" discussion [Lenat and Guha, 1990]). As most of the reviewers of the Cyc book point out [Elkan and Greiner, 1993, Skuce, 1993], they did not have much justification for this decision, other than the questionable claims for efficiency. In a sense, they were following history with a ten year lag, as Minsky had published his seminal work on frames in 1974 [Minsky, 1974].

Frames do have their own problems. The basic semantics of a frame x having value v under slot p is the same as saying $p(x, v)$ in logic. However there is a lot more that can be expressed in logic that is not so easy with frames. Embedded clauses are the standard example [Elkan and Greiner, 1993]. Mayor(Capital(Texas)) cannot be expressed without making Capital(Texas) into a new frame (reifying it). Another problem comes up when handling time. To express the fact that some object had a property over a particular time period, Cyc chose to define *subabstractions*. A subabstraction is a new frame that only represents the object over that time period, such as LenatAsAnInfant or FredWhileAsleep. So, for each assertion you have to make that is valid during a particular period of the subject's life, you need to create a new frame. You have to keep all these frames in synchrony in other aspects. They have tried the same solution for belief. So if Mary believes that Fred is a plumber, you create a Mary subabstraction of Fred as a separate frame and assert facts about it. These are special cases of the embedding problem. In a frame language, one is not allowed to say true-over-interval(1980, occupation(Fred, plumber)) [McDermott, 1993].

So Cyc started its life having a simple frame language. By the time the book was written (1989), on top of the frames there is a predicate-calculus like constraint language. Soon after, in their midterm report (1990), they express the need for a clean semantics, and the need to have a language that has the expressiveness of first order logic with some extensions to handle equality, default reasoning, skolemization and some second order features. Currently (1996), Cyc doesn't seem to have any connection to the frame architecture. The Cyc team thinks of their knowledge base instead as a sea of assertions, with each assertion being no more "about" one of the terms involved than another. The moral of the story, however, is that things that seem most natural to human thinking, time, space, beliefs, causality, different possible worlds, intentionality, etc. create the hardest problems for symbolic representations.

A rather surprising development is the introduction of analogue representations to Cyc (at least the effort thereof). In the midterm report they mention their failed attempts at developing a single general abstraction for space, and their decision to develop several abstractions: (1) simple diagram-like representations, (2) computer-aided-design like representations that build solids and surfaces out of a small number of primitives, and (3) device-level representations that primarily deal with the topology of a device by using a number of primitive components and using a small number of ports for each primitive and a small number of ways in which two primitives can be connected. A similar intention is mentioned in the book also. I don't know if these ideas ever got beyond the stage of intentions.

4.3 Modularity

Another dimension in the evolution was the gradual appreciation of modularity. As the knowledge base grew, so did the problem of inference efficiency. I will mention two methods they used to fight this problem.

4.3.1 Epistemological level / heuristic level distinction

The importance of this concept was first pointed out twenty years earlier by McCarthy and Hayes [McCarthy and Hayes, 1969]. There are two properties that knowledge based systems must satisfy. Epistemological adequacy means that we should be able to state whatever we want to state in the domain using the language of the system. Heuristic adequacy means that the system should be able to infer common conclusions in an efficient fashion. Typically there is a trade-off between the representational power and the inference efficiency. There is no a priori reason for using the same language for both requirements. Thus one can build a system on two levels. The epistemological level supports an expressive language. The heuristic level consists of a lot of special purpose inference engines that gain efficiency on a subclass of problems by using appropriate representations and algorithms.

The state of Cyc in the book, split between a frame language and a predicate calculus representation, is in fact a result of not appreciating this distinction from the first day of design. The frame language was adopted for its efficiency, while the predicate calculus was needed for its expressive power to represent constraints. As Cyc matured, the EL/HL difference was integrated into the system, and the need for different languages disappeared. In fact the current Cyc provides a uniform language for the users to make assertions and queries. Knowledge is represented redundantly at two levels. The epistemological level keeps a copy of the facts in the uniform user language. The heuristic level keeps its own copy in a number of different languages and data structures to make certain inferences efficient. A user query is automatically translated between these levels to make use of the most efficient algorithm to carry it out. The heuristic level contains dozens of specialized inference engines.

These inference engines, however, are based purely on the syntactic structure of the axioms. Typical examples include engines for inheritance, automatic classification, maintenance of inverse links and dependencies etc. The representations they manipulate are still symbolic representations, albeit expressed in a more convenient data structure. The inference they support is still deductive inference.

4.3.2 Microtheories

An orthogonal direction for modularity is using microtheories. A microtheory is an internally consistent collection of facts about a particular domain [Sowa, 1993]. For example the system can have a naive theory of physics, in addition to a more formal theory. They would be useful in different contexts. However they talk about the same concepts, and may include incompatible facts. Instead of trying to hold the whole knowledge base in synchrony, we can have independent microtheories floating, each of which is internally consistent.

Microtheories were introduced relatively late to Cyc. They are not mentioned in the book. Cyc already had a large number of assertions by the time the book was written. This means that the knowledge engineers managed to keep so many assertions consistent, which is remarkable. The microtheories are introduced in the midterm report. The contents of the ontology up to that point were collected in one default microtheory called MEM (most expressive microtheory). Most of the assertions at that point were in MEM. In recent reviews however, it is evident that microtheories started playing a central role in Cyc.

As a rough estimate of the current magnitude of knowledge in Cyc, there are more than 400,000 significant assertions of which less than 30,000 are rules for inference. There are over 500 microtheories defined [Whitten, 1996]. The size of the knowledge base has fluctuated over the years, in particular it has decreased in size when axioms have been generalized. In the midterm report (1990) Cyc was reported to have over a million assertions. The work being done on the Cyc knowledge base currently is largely in the form of the development of microtheories for topics at the level of transportation, human emotions, modern buildings and so on.

Of course microtheories have their own problems. After all Cyc was a decision to build a large knowledge base instead of a hundred expert systems. Dividing Cyc up into microtheories carries the risk of making each microtheory susceptible to the same criticisms they made about expert systems in the first place. How can different microtheories support each other without getting in each other's way? How does the system know which microtheory to select given a certain query? These and similar questions remain unanswered in the reviews.

One of the most fascinating things about human mind, and human memory in particular, is the fact that we don't run into the above problem. Imagine your long term memory, full of millions of small facts ranging from your mother's face, to your social-security number, to the players of your favorite football team. Yet when you are trying to come up with ideas to solve a particular question in a physics test, none of these things seem to show up.

Other than making Cyc focus on domains, microtheories have the additional function of keeping it consistent. Why are they trying to keep Cyc consistent? Why do they insist on checking every new assertion against everything else that is already in the knowledge base to make sure things do not clash? After all, a program could just remember every entry on a particular topic with who typed it when, and give all the different answers with their sources when probed with a query.

Deductive systems are extremely sensitive to inconsistency. That's why there are people in the world whose field of expertise is building truth maintenance systems. A deductive system happens to crush rather abruptly if you assert two contradictory statements:

$$\begin{array}{l} p \Rightarrow (p \vee q) \\ \neg p \vee p \vee q \\ (p \wedge \neg p) \Rightarrow q \end{array}$$

This small derivation starting from a tautology shows that a conjunction of an assertion with its negation can be used to deduce q, *no matter what* q *is.* In an inconsistent system every statement will have a proof. Thus the system will cease to be interesting, because it will fail its main function as a logic: the ability to separate the set of all statements into true and false.

This state of affairs is rather expected. I will argue in the next section that logic initially was invented as the science of *demonstrative arguments*, in which case finding out who is right and who is wrong is all that matters. When it got applied to the science of *thinking*, rather ugly things started showing themselves. As one typical example, new information was able to change the old conclusions. This is never the case in traditional logic. If a statement can be proven from already existing axioms, it remains true no matter what you add to the system. To cope with this, the field of "non-monotonic reasoning" was born. To handle further problems, frame axioms, closed world assumption, circumscription, default reasoning etc. had to be invented.

The immunity of humans to inconsistency is a remarkable fact. In a science fiction story by C. Cherniak [Hofstadter and Dennett, 1981], an artificial intelligence researcher enters a state of trance in front of his computer. His friends finally notice the problem after a few days and try to wake him up. He never comes out of the trance, though, and dies within a couple of days. Mysteriously, people who dig up his work, look at his files share the same terrible fate. At the conclusion it is revealed that the first victim actually had discovered the Gödel sentence for humans. Thank God, we are not created as deductive systems. Tuttle and Smith discuss various ways in which human thought can be different from logical automatons [Tuttle, 1993, Smith, 1991].

One wonders, when faced with solutions too complicated, whether one is even at the right search space.

5 The historical roots of Cyc

5.1 Cyc was a natural consequence of history

Building Cyc within the above design space may seem like an odd way of doing things. To a student of history, it will seem more like a natural outcome of the developments preceding it. Let's take a quick look at how things got there in two thousand years.

Logic is the representational framework with long roots in history. As with almost every other science, its beginnings can be tracked down to the dialogues of Socrates and the writings of Aristotle (4th century BC). At the time, the axiomatization of mathematics had just begun. They were preceded by Pythagoras by 200 years, and immediately followed by Euclid. Aristotle set down the rules for dialectic (the art of logical discussion employed in finding out the truth of a theory or opinion), he categorized syllogisms (the early form of deduction). Although the rules set down weren't as clearcut as the modern logicians would like them to be, a collection of regularities was discovered for making logical arguments [Stevenson, 1996].

Leibniz (1646-1716), is the founder of mathematical logic. Unfortunately he abstained from publishing his results, because he kept on finding evidence that Aristotle's doctrine of syllogism was wrong on some points, and respect for Aristotle made it impossible for him to believe this. He mistakenly supposed that the errors must be his own. Thus mathematical logic wasn't discovered for another century and a half. His inspiration was the hope of discovering a kind of generalized mathematics, which he called *Characteristica Universalis*, by means of which thinking could be replaced by calculation. He is quoted to say

> "If we had it, we should be able to reason in metaphysics and morals in much the same way as in geometry and analysis. If controversies were to arise, there would be no more need of disputation between two philosophers than between two accountants. For it would suffice to take their pencils in their hands, to sit down to their slates, and to say to each other: Let us calculate." [Russell, 1945]

With a little imagination we can replace the people with pencils with computers. Maybe we should think of Leibniz as the founder of AI. Almost two centuries later, inspired by the writings of Leibniz, Boole developed the theory of propositional logic, and modestly called it *The Laws of Thought* (1854) [Kurzweil, 1990].

Frege finally layed the mature foundations of modern logic as we know it (1879). From his work it followed that arithmetic, and pure mathematics generally, is nothing but a prolongation of deductive logic. He remained without recognition until Russell drew attention to him in 1903. Russell's own work with Whitehead culminated in *Principia Mathematica* which layed the foundation of mathematics from set theoretical principles and logic.

Logic went through its own evolution over the centuries. We can separate this evolution into three stages. The idea of the classical logic was to start from a limited number of axioms (the core), to show that each theorem follows logically from the axioms and the theorems which precede it according to a limited number of rules of inference. The idea of the modern logic is to represent axioms and theorems as sequences of opaque symbols, and specify the mechanical manipulations of these symbols that correspond to the chain of deductions in classical logic. The idea of meta-mathematics, originated by Hilbert at the turn of the century, is to construct the symbols and their mechanisms of manipulation, independently of any interpretation of it. The system can then be interpreted as representing a deductive system if a valid isomorphism can be found. At each of the above steps there is an abstraction away from the actual subject matter represented. This is important to be able to prove theorems about the symbol manipulation system itself. In fact, this effort culminated in Gödel's proof (1930) of the incompleteness theorem, which basically states that the two thousand year effort of formalizing mathematics can never be achieved completely.[Gödel, 1962]

Now let's look at how this development might have effected Artificial Intelligence. At around the same time as Gödel, Turing's theorems and abstract machines gave hint of the fundamental idea that the computer could be used to model the symbol-manipulating process. This was the bridge between Leibniz's dream and its computer implementation.

In about thirty years, the field of Artificial Intelligence was born. Being a new science, it drew from its ancestors. Psychology at the time did not provide an adequate foundation. It was a collection of simple learning rules, Gestalt principles, Freudian theories, non of which were sufficient as a root to build on. The mathematical logic seemed much closer to the computational way of thinking. The idea that one can specify a set of symbols and mechanical rules of manipulation, which can correspond to some external reality seemed appealing. This led to the first principle of the new science, the physical symbol system hypothesis.

Some people take the physical symbol system hypothesis as a form of the Church-Turing thesis. That a universal machine will be able to simulate anything in the universe given enough time and memory. I want to make clear that the physical symbol system hypothesis should not be understood in this trivial sense. Remember the point about the **Doability Argument** from the introduction. It is pointless to argue about the impossibility of an approach. If you push too hard, the physical symbol system hypothesis might include simulating each atom in someone's brain, in which case we have no argument. So what is meant by the hypothesis is that the trick pulled by the modern logicians to formalize mathematics can be applied to "thought" in general. i.e. Tying each separate "concept" (not atom or neuron) to a symbol, and then specifying some mechanical rules to push around these symbols is sufficient means for intelligent action. It basically boils down to the claim that a system like Cyc should work.

It should also be noted that this was not an original hypothesis. In fact, Frege is the first person who explicitly defended the role of symbolic formalism as a general representation to be applied to arbitrary domains. The definition of truth by Tarski [Tarski, 1935] which later led to the implementation of model-theoretic semantics in linguistics by Montague [Montague, 1974] pre-dates the Dartmouth Conference (1956).

Now let's trace the thirty years of the AI experience that led to Cyc. The first era of AI focused on isolated problems and search methodologies. Soon the need to attack real world problems as opposed to investigating toy domains arose. The link between the programs and the real world had to be more knowledge. In 1970's knowledge is power hypothesis gained popularity. The expert systems became popular. But by 1980's it was evident that the narrow sighted problem solvers were no more closer to achieving intelligence than their search based predecessors. An obvious difference between the programs that pushed meaningless symbols around and the real intelligent beings that interpreted them was the ability to access the meanings of those symbols. Thus Cyc was born.

5.2 Logic was meant to be a science of statements

The only reason I dwelled upon so long on history in the previous section is to show you the forced adoption of logic and related formalisms as theories of thinking. I will argue below that logic was created as a science of demonstrative statements rather than thinking.

We can start by asking the question, why create formalisms at all? To give an example from a different domain, consider the science of harmony and counterpoint in music. Today, you can take a class in these topics, and learn about the basic principles and rules by which "good music" is made. There are well formed rules for constructing chords, which are ultimately collections of pitches that sound nice when played together. In fact, more than being nice, you can predict the psychological effect of particular chords on the listener (happy, sad, tension creating etc.) The surprising fact for me was to learn that a lot of such rules were not known during the lives of great composers like Bach and Mozart. In fact, analysts later constructed these rules by finding the regularities in the compositions of such geniuses. This allowed mediocre non-geniuses like me to study and learn the basic principles.

Socrates was one of the best virtuoso of argument making. He believed that the nature of things could be discovered by argument instead of empirical observation. Looking at the dialogues by Plato, one can't help but think that he almost succeeded. Aristotle later layed the rules for "dialectic" and "rhetoric" which formed the foundations of logic. These fields captured the regularities in the verbal demonstrations of geniuses like Socrates. They were put down so that mediocre non-geniuses could learn the principles and not get stuck, or be misled by arguments. Thus logic was invented as a science for the means of persuasion. It assumed the existence of two intelligent parties arguing on some topic. It layed down the rules which encapsulated the minimum intersection between people with different views that could be used for persuasion. Meaning of symbols were agreed upon the participants using their common concepts. So there is no focus on where those concepts actually come from. And finally, it did not say anything about what went on in the heads of the speakers.

Even in Leibniz's dream I quoted in the previous section, one can see the same picture. He says "If controversies were to arise, there would be no more need of disputation between two philosophers than between two accountants." Logic is intended as a way to resolve disputes between two already intelligent participants. Again, the meaning of the symbols are taken for granted, because they exist in the heads of the participants. And nothing is said about what goes on in the heads of the participants, other than maybe the expectation that a small portion of it, namely deductive argumentation, might be simulated by mechanical calculation.

I do not mean to imply that classical AI is founded on mathematical logic. I do mean that they share some basic principles, namely

Explicit representation The effort to represent each little piece of knowledge explicitly in a declarative form.

Uniform representation The use of symbols as the only data structures that correspond to "concepts" of the domain.

Deductive reasoning The reliance on deductive inference as the main computational engine to expand from the core to the borders of knowledge.

To illustrate the power of these principles in guiding an AI researcher's thinking, consider this quote from Lenat:

> "First, we criticized the current expert systems for merely containing opaque tokens and pushing them around. Yet our example of having more general, flexible knowledge was nothing more than having more (and more general) tokens and pushing *them* around! Yes, all we're doing is pushing tokens around, but that's all that cognition is."[Lenat and Guha, 1990]

Saying that a particular theory was not intended for a particular application is certainly not a proof for its inadequacy. But it makes one think whether we are stretching too far. After all, the physical symbol system hypothesis believes in an explanation of what goes on in our brains, within a formalism intended as a theory of statements those brains can generate. A formalism that just concerns itself with the minimum common ground that those brains can use to resolve disputes. In the following section I will try to describe a theory of thinking that relaxes some of these bonds, and gets its inspiration from cognitive science.

6 Beyond the binding roots of symbolic AI

So far I have only talked about problems. I believe Cyc falls short both in expressive power and inference efficiency. This means it fails both on epistemological and heuristic grounds. I have tried to argue that the problems were typically due to insisting on some limiting principles like explicit and uniform representation, and relying mainly on deductive inference.

I should make clear that although Cyc seems to have different representations and inference engines at the heuristic level, they are just reimplementations of the general representation and inference framework of the epistemological level. The specific modules are separated on syntax based rather than domain based concerns. They do not implement different frameworks in the sense I will describe below.

I believe that these limiting factors were borrowed from a formalism that was not intended to be a theory of thinking, and buried into the foundational hypotheses by the fathers of AI. In this section I will try to demonstrate what it means to break these limitations.

6.1 What does it mean to understand RED?

In a previous section, I suggested the “asking obvious questions” test to measure understanding. So a system that understands the word “enter” for example, should know that when you enter a room, you are “inside” that room.

However, it might seem unnatural to talk about understanding without perception. Is it OK to say that an agent understands “red” when it does not have a “red sensor”? It is one thing to know what things are red, and that red is a color, and that color is physical phenomenon related to the spectrum of light. It is another thing to be able to tell a red thing from a green thing. This latter type of “understanding”, is considered an important requirement by advocates of “embodiment” and “grounding”.

Other people think that this is a “trivial” requirement, i.e. it doesn’t say anything about whether lack of grounding makes achieving human level intelligence impossible, or even difficult. It can be argued that it makes it actually easier by eliminating irrelevant detail. I find the arguments like the following not adequately answered by the proponents of embodiment:

Human with a teletype argument If you don’t believe that a teletype is a good enough I/O channel for an intelligent being, imagine yourself locked in a room, having to answer messages sent over a terminal. If we can build a snapshot of your mind as it is functioning right now, a teletype will be more than adequate to communicate.

Thinking with symbols argument I realize that the sensory system gives you a lot of information, and not all the details can be captured in a symbolic representation. However, it is not clear that humans retain much of that information either. The long term memory seems to convert what you see into symbolic representations (like picture frames). It is plausible that thinking is also done using such representations.

High level reasoning argument I don’t care about navigation and survival. These are things that get in the way of understanding intelligence. Redefining intelligence to be these things does not help. I am interested in high level reasoning.

I think the adequate answers to these objections would have to be pragmatic arguments. Doability arguments are not going to lead to conclusions until someone actually does achieve human level intelligence. Shifting the definition of intelligence does not help. We don't know enough about the brain to rule out different theories of organization. In the following sections I will try to build an argument on pragmatic grounds.

6.1.1 A grounded understanding of red

To facilitate discussion, I would like to describe a hypothetical color reader machine inspired by Hausser [Hausser, 1989]. This machine has a camera and a speaker. It can look at objects and tell what color they are. In that sense, it satisfies the basic premise of understanding given above.

Note that, there are at least five different levels of "red" in this picture:

(1) The red object out there in the real world.
(2) The image of the red object that falls on the camera.
(3) The concept red, that matches this image and presumably all other red things.
(4) The representation of the linguistic structure corresponding to the sentence "This is red".
(5) The symbol tokens out there in the real world "This", "is", and "red" carried by the sound wave.

The logical tradition concerns itself with the relationship between symbols (5) and objects (1) in the real world [Lakoff, 1987, Johnson, 1987]. One starts by linking symbols to objects, sets of objects, and tuples of objects. Other symbols are reserved to represent properties, relations and quantifiers. Syntactic rules are specified for constructing well formed formulae (wff). Assertions express propositions about the real world using wff's. Rules for algorithmic manipulation of these wff's for inference are determined [Davis, 1990]. There are two critical properties in such a system: that the inferences made by following the rules of symbol manipulation correspond to the external reality, and that things that are true in the real world can be inferred by symbol manipulation. These correspond to soundness and completeness respectively.

Note that nothing in this description include anything at the interpreter side of the picture. There is no mention of the camera, the intermediate concept, the linguistic system or the robot. In fact, it looks almost like the rules of a game. You and me sit down at a table with a game board. I throw some pebbles on the board and say, "This pebble corresponds to the bird, that pebble corresponds to flying, etc." Then I tell you the rules by which I am allowed to rearrange the pebbles, and how groups of pebbles correspond to propositions, and how the rearrangements map to valid rules of inference. If you just accept my mapping and the validity of my rules, then I should be able to convince you of anything that I can come up with using my pebbles. There is no need to worry about questions like "What is the real meaning of this pebble?", I just told you it corresponds to the bird.

This is exactly how Cyc is built. Symbols are selected to correspond to "things" in the universe. A language is specified at the epistemological level to express assertions about these “things”. Rules of inference are defined, and implemented efficiently at the heuristic level. The symbols in Cyc have no meaning attached to them. Lenat frequently makes the point that all of them could be replaced by gensyms. This means the only property by which we can distinguish the meaning of two symbols is their position relative to all the other symbols in a complicated web.

6.1.2 The philosophical problem

Ungrounded definitions of meanings have their philosophical problems. The mainstream linguistic semantics has what is called a model-theoretic definition of meaning. It is based on the truth of statements. However, you cannot simply define the meaning of a sentence as its truth value. In that case “MIT is in Massachusetts.” and “Ankara is the capital of Turkey.” would mean the same thing, both are true. So we have to be more careful. The model theoretic definition of meaning is a function which assigns a truth value to that sentence in each possible world. Thus it is equivalent to the set of possible worlds under which this sentence would be true. Putnam proved in 1981 [Putnam, 1981, Lakoff, 1987] that such a definition was inadequate to capture the concept of meaning. Putnam’s theorem shows that one can always come up with two different sentences consisting of words with different meanings, that would be mapped to the same “meaning” according to the above definition. Attempts to patch the model-theoretic definition of meaning have not been successful. The problem seems to arise from two factors: (1) the attempt to define meaning mainly in terms of truth, and (2) the attempt to define meaning objectively, i.e. independent of the mind which assigns the meanings to the symbols in the first place.

Although Putnam’s theorem may attract one’s attention to the importance of including the agent side of the above picture, it is a philosophical argument. It is like Gödel’s incompleteness theorem which did not stop people from using axiomatic systems to prove theorems about arithmetic. Just as a student who cannot prove some theorem in an arithmetic exam would be foolish to attribute his failure to Gödel, Putnam’s argument is not, by itself, suggesting that the model-theoretic approach is inadequate in practice. We need more hard evidence for that. It does make an ironic point however, that the system founded by people whose most important criteria for success was consistency is not consistent with the basic facts of its domain.

6.1.3 The practical problem

One thing the objectivist theory of meaning bypasses completely is how the universe gets divided up into concepts (the cosmic cookie cutter problem [Lenat and Guha, 1990]). We look at the color spectrum and divide a continuous space into discrete regions, which we call “red”, “blue”, etc. We look at orientations, and come up with three main concepts, “vertical”, “horizontal”, and “diagonal”. We have the ability to see an army troop as a single unit, and a car as a collection of interacting components. We look at a room, and see a chair, a table, a person etc. A chemist could see the room as a bunch of molecules, or a particle physicist could see the same room as continuous waves. Who knows how a fly sees it? Our divisions are in no way universal or most natural. Cyc, on the other hand does not look at a universe and try to make sense out of it. It passes the concept acquisition problem completely to its designers, and its units of thinking are predetermined.

One could use the human with a teletype picture to argue that what is being built in Cyc is the model of a mind frozen after reaching adulthood. So, the problems of concept acquisition has been solved. The system has learnt to see the room in terms of tables and chairs. All we need to do now is understand new information in terms of these concepts and make inference over them.

The argument would be valid if the machinery that supports perception and concept acquisition does not support learning and inference later. As I will try to show in the following sections, they seem to play a very important role.

6.2 The imagination perception loop

I criticized Cyc for having only an explicit representation, and relying on deductive inference as the main computational machinery. How else is it possible to represent things and do inference?

To illustrate the point, I will give an example from Herbert Simon, commenting on visual imagery to Gazzaniga [Gazzaniga, 1985]: “Imagine a rectangle. Draw a line from the top right-hand corner to the bottom left corner. Now draw a line from the middle of the diagonal to the bottom right corner. Now approximately one third of the distance from the top right corner along the top line, drop a perpendicular line down to the lower edge. How many lines do you intersect?”

Introspection is typically not considered valid scientific evidence in cognitive science. However if you felt like you were drawing the rectangle in your mental sketchpad while reading the question, it did not mislead you this time. There are conclusive brain imaging studies that show that some visual regions of the brain that are active during perception are also active during thinking and imagination.

Obviously the example was cooked up to fit exactly my point. Nevertheless it illustrates a radically different way of doing inference. You probably did not enter the first couple of assertions in an ontology, and then use resolution to get your answer. You simply used the reverse wiring in your brain to go from descriptions to sensations, and just ran your already existing perceptual machinery to look at the answer for you [Rao, 1995b].

This illustrates a powerful inference engine. To find out whether a flying bird touches ground, you might have used this machinery for a blink of a second to see that it does not. Maybe the picture was drawn for you while you were hearing the sentence. To “deduce” that the Pisa tower is an unbalanced structure, you can imagine your body tilted at that angle, and the balance sensors in your ears will tell you it is not very stable.

McDermott, in his review, points out that a computer system cannot effectively receive a representation of a piece of knowledge without an algorithm ready to process it with reasonable efficiency [McDermott, 1993]. He also adds in a footnote that if knowledge is not represented, but merely implicit in an efficient algorithm, this requirement will not come up.

The imagination-perception loop also illustrates the implicit representation. The fact that flying birds do not touch ground does not have to be represented anywhere in your knowledge. It is implicit in the procedures that convert the verbal description to a visual image, and the procedures which can look at the image and answer queries. Contrast this with Cyc, which has to link flying and touching ground with an explicit chain of declarative statements.

The number of facts you can deduce from what you know, is not restricted to the deductive closure of everything in your symbolic memory. You add to this the analogue information you have in the memories of your other representational systems. You further add the ability of one system being able to set up experiments to be run in another.

Actually, the number of facts Cyc can draw from its knowledge base (within practical time limits) is probably a lot smaller than the deductive closure. Inference is an expensive process, especially if you have so many rules. Deductions that require a few steps can be reached, but the search space grows exponentially as the number of steps increase. In contrast, the imagination-perception loop is constant time. You convert the description from one representation to another. You run the already optimized constant time visual machinery to look at the answer. You send it back to language. The equivalent number of deductive steps is irrelevant.

If you think that you are not using your imagination for some of these problems, don't worry. There is nothing wrong with you. First of all, simple facts like "flying things don't touch the ground" are probably cached in your symbolic memory, even if you had to use your perceptual machinery to figure them out in the first place. If you think you know the answer to the question "If A is taller than B, and B is taller than C, is C taller than A?", before you have to visualize anything, this just means that you have done similar inferences hundreds of times in your life, and they just became second nature to your symbolic system. After all, we are not arguing against any of the things Cyc does, or any of the representations Cyc uses. We are advocating the use of more representations, and more computational machinery.

Last year I built a chess program to try out a new move generation algorithm I designed. It was based on the idea of representing a lot of state instead of redoing the computation. I ended up using a data structure which was a two dimensional array of cells, which were connected to each other by doubly linked lists running in the horizontal and vertical directions, where each cell was actually a pushdown stack of structures. In the end I was proud to be the first person using a quadruple star operator in C. The point is, we are used to designing appropriate data structures and algorithms for our programs. It is hard to understand the resistance to using multiple frameworks when one is designing a mind.

6.3 If a picture is worth a thousand words, how many pictures is a word worth?

There has been an ongoing debate between British associationists and logical positivists in the philosophy world, on whether our minds use pictures or words for thinking. The same discussion continues today in the form of the imagery debate. Symbolic vs non-symbolic AI discussion is also a mirror of the same philosophical tradition.

The advocates of thinking with pictures say that a picture is worth a thousand words. There will always be details of a picture left that a symbolic description just fails to capture. There are always more inferences you can draw from a picture than from its description.

Other philosophers argued on the other hand, that it was impossible to represent disjunctions (a tall or a Chinese man), and negations (a room without a giraffe) with pictures. In fact, in general, it is impossible to represent sets of things. One can never visualize the concept of a triangle. Each time you try, you will see a specific triangle.

Another argument for symbols being the natural currency of thinking, comes from the way people store things in their long term memory. Without looking at a penny, try drawing the picture of one, seen from the head side. Typically people remember that there is a portrait of Abraham Lincoln from profile, somewhere it says “in God we trust”. Maybe you even remember that the year and the word “liberty” are also on this side. Did you finish the drawing? Now find a real penny and compare the results. The mistakes people typically make are drawing Lincoln looking at left rather than right, putting “in God we trust” to the bottom rather than top, switching the places of the year and “liberty”. No one draws a penny with the upper left side faded out, as you would expect from the degradation of a pixel representation.

This is a powerful argument for symbolic representations. However, it does not say anything about using images for doing inference, the imagination-perception loop. You may be reconstructing the image from its symbolic description, and this does not hinder the advantage of using images for certain kinds of inference. Furthermore, if you didn’t have any pictorial information in your memory, you wouldn’t be able to draw anything. This shows that our long term memory has to carry pictorial representations in addition to verbal ones. In fact, it can be argued that each representational system has its own long term memory. They each experiences the world within its own language, remember things from the past and learn from regularities within its own framework.

The answer to the question in the title of this section is not 1/1000. In fact it is a thousand also. This is an unusual situation. We are used to seeing “many to one” mappings. But typically this means the “one” set is much smaller than the “many” set. Imagine a situation where the relationship is symmetric. To completely capture the information in a picture, you need a lot of words. To completely capture what is represented by a particular concept, you need a lot of pictures. This is the power of multiple representational systems. One cannot substitute for the other. What you can say in a few bits in one, require infinite space in the other, and vice versa [Rao, 1996].

6.4 Plausible reasoning

In fact, as you might have suspected already, there is a bug with our scheme of inference using imagery. And this bug stems from the realization of the previous section, that we cannot substitute different representations for one another. When I started the description, “imagine a rectangle...”, you were imagining a specific rectangle. Your image did not capture the concept of a rectangle in general. The answer to that particular puzzle happens to hold for any rectangle, so you were able to solve it. But it didn’t have to be that way. The typical mistake of geometry students is to draw an equilateral triangle instead of "any" triangle and be misled by their own drawing into wrong conclusions. Good thing we brought this up. I was almost suggesting a constant time algorithm for the NP-hard problem of inference.

Whenever you pass information from one representational system to another, knowledge gets lost. You have to add defaults to visualize a concept, or you have to choose particular abstractions to verbalize a picture.

To conclusively prove a statement including abstract concepts like triangle using pictures, one would have to consider the picture of every possible triangle. Typically, however, a few carefully selected examples suffice to span the whole set and convince us of a proposition. How to select the right examples is a hard problem, because it partly depends on the question asked. This is the heuristic component of our "constant time" algorithm, and it may fail at times.

The fact that some mechanism of inference is not complete, does not render it useless. It may be better to have a system that works 99% of the time, and then try to focus on learning the exceptions.

This starts to look more like inductive inference rather than deductive inference. The difference from the standard idea of "induction" is that the mind has the ability to generate its own examples. Instead of going out to a field and try to find flying birds, you can just imagine them to infer they don't touch the ground.

In fact, if we were to do deduction based on explicit rules, we would have to acquire those rules using induction. We would have to see enough examples of flying birds and falling rocks etc. to compile the rules about them. If Cyc did not start with the idea of typing in the knowledge, but to discover the regularities watching the real world, it would have to come up with this mechanism in the first place.

Similarly, if the primitive symbols were not typed in initially but Cyc had to come up with its own concepts, the team would have to come up with perceptual systems. Then the implicit representations and the imagination perception loop would come more naturally.

Polya distinguishes between demonstrative reasoning and plausible reasoning, and further claims that plausible reasoning is the only means by which we can acquire new knowledge. For example, what you turn in with your mathematics problem set (hopefully) is demonstrative reasoning, but the activity you engaged in the night before, to understand new concepts from examples, trying out special cases, searching for similar solution patterns in your head is plausible reasoning. The problem of AI concerns the latter, dependence on pure symbols belongs to the former.

> "We secure our knowledge by demonstrative reasoning, but we support our conjectures by plausible reasoning. A mathematical proof is demonstrative reasoning, but the inductive evidence of the physicist, the circumstantial evidence of the lawyer, the documentary evidence of the historian, and the statistical evidence of the economist belong to plausible reasoning."
>
> "The difference between two kinds of reasoning is great and manifold. Demonstrative reasoning is safe, beyond controversy, and final. Plausible reasoning is

> hazardous, controversial and provisional. Demonstrative reasoning penetrates the sciences just as far as mathematics does, but it is in itself (as mathematics is in itself) incapable of yielding essentially new knowledge about the world around us. Anything new that we learn about the world involves plausible reasoning, which is the only kind of reasoning for which we care in everyday affairs. Demonstrative reasoning has rigid standards, codified and clarified by logic (formal or demonstrative logic), which is the theory of demonstrative reasoning. The standards of plausible reasoning are fluid, and there is no theory of such reasoning that could be compared to demonstrative logic in clarity or would command comparable consensus." [Polya, 1954]

6.5 The "high level" reasoning

Imagine standing in front of an ice pond, wearing flat-soled shoes and thinking "what would happen if I tried to walk on this pond?". Let's ask this question to Cyc and try to visualize the steps it would take. It would dig up what it knows about ice and shoes and walking. It would find out that ice typically has a low friction surface. It would see that shoes are what you wear when you walk. It would go down the chain of its concepts to as deep a level as necessary to get the information it needs. Somewhere in its ontology, there is an explanation of what friction is, and that it is needed for walking. If we are lucky, it would finally conclude that we would slip and fall if we walked on an ice pond with flat-soled shoes.

On the other hand, your previous experience had introduced you to the experience of falling down on ice. You can imagine the sequence of your foot sliding, a sense of losing balance, landing on your back. In fact you have low level machinery that supports how you take each of your steps when you walk on ice, rocky terrain, in crowd etc. This means that you have a native system whose primitives are things like walking, running, sliding and falling.

Cyc has a hierarchy in its ontology. High level concepts are defined in terms of lower level concepts. Cyc understands sliding and falling in terms of friction. We understood friction in high school in terms of our memories of sliding and falling. Lakoff characterizes this phenomenon by the term *basic level category* [Lakoff, 1987]. We tend to understand more abstract concepts in terms of our bodies. We also understand more primitive concepts (if you agree that friction is more primitive than sliding) in terms of our bodies. It is as if our bodily concepts cut an ontology like Cyc's horizontally in half, and we reduce both the upper branches, and the lower roots to the midline.

Why is this twisted organization of things, when there is a natural hierarchy of generalizations and specializations. The simple reason is that we have the computational machinery to evaluate things efficiently that come from the midline. We learn speaking before grammar, we learn walking before physics, we learn how to interact with our friends before sociology. It is only natural that we build the later layers of knowledge, on top of what already exists, irrespective of whether they are higher or lower in some hierarchy. Papert's *Mindstorms* has wonderful examples of this structure [Papert, 1980].

Humans come to life with powerful computers and optimized representations in a number of domains. The visuo-spatial system gives us powerful tools to perceive and think about the forms and positions of objects. The motor system senses and directs our posture, orientation and movements. The language system is specialized in syntactic manipulation. The socio-emotional system not only provides motivation for our behavior but also allows us to understand the behavior and internal state of others by putting us in their shoes and running simulations. And probably a conceptual system not unlike Cyc, organizes our explicit memories and leverages off the knowledge and simulation capabilities of all the other systems.

Each system has its own independent memory, its own regularity detection and learning mechanisms, its own representation of concepts. Consider the concept "vertical", for example. It means quite different things to your visual system (imagine a vertical bar), to your motor system (sensation of a vertical posture), and to your vestibular system (sense of balance, compare being vertical versus being tilted like the Pisa tower). When you think about "vertical", though, all these systems are ready to help you with their own special expertise of efficient inference.

What does this tell us about how to organize a system with common sense? One would first have to select some primitive domains as a basis for the system [Rao, 1995a]. There are two requirements a basis domain has to satisfy to be useful:

(1) It has to be expressive enough, so that other domains can be mapped on it.
(2) It has to have efficient inference engines.

A visual system satisfies both of the above criteria. A theory of mathematical groups satisfy neither.

Once the basis systems are built and are able to use each other as computational "demons", the system can start learning other domains by mapping them onto the existing ones. We frequently do this sort of mapping. For example the Venn diagram representations of events are often useful in understanding probability. Each time I forget Bayes' formula, I draw a Venn diagram of two events and read the formula off of it.

The mapping will typically not be perfect. For example, when there are too many events, the Venn diagrams become confusing. All the inferences you can draw from the basis domain might not be valid in the target domain. However, different basis domains can cover different parts of the inference space. If you have powerful engines handling a great deal of your inference, one can focus on the exceptions and structure them further easily.

There is a section on analogy in the Cyc book. Judging from the lack of any positive evidence in the recent publications, I conclude that it is one of the ideas that did not give satisfactory results. Typically "analogy", in the knowledge representation world means that you go and look for similarities between two existing structures. What I am suggesting in this section is a step further. You *define* a new domain as analogies or metaphors to existing ones. Thus analogies are already existing links in the system. We use spatial metaphors for quantities, not because it is a hot trend in the literary fashion, but because we understand quantities in terms of spatial primitives. Our inferences about them depend on the powerful engines of spatial representations that exist in our minds.

7 Conclusion

Cyc is a controversial project. It was a bold attempt to come forward and try to put whatever the field has to offer into a single direction. Breakthroughs can be achieved only by such concentrated efforts.

The problem that motivated Cyc was the brittleness of programs. Many problems in expert systems, knowledge sharing, natural language understanding and human interfaces were due to the shallow of support behind the programs' primitive symbols. Cyc proposed to fix this problem by providing a deeper framework, in which each symbol would be situated within a rich ontology from which it can drive meaning.

Cyc was a child of symbolic AI. Its approach to knowledge representation uses a purely explicit symbolic framework. Its inference engines are rooted in the formalist tradition. It pushed these frameworks to their limits. It uncovered many potential problems that arise from the increasing magnitude of knowledge.

As it pushed certain methodologies to their limits, it opened up a need for new approaches. The approach I advocated in this paper is based on removing certain binding constraints that come from the symbolic AI tradition. I will conclude by giving a list of principles I would use to build the next Cyc.

> Symbols are not the only kind of data structures that should flow in a system. There is room for other modalities.
>
> Deduction is not the only way to reach new facts. Using the appropriate modality to imagine situations and testing them by the efficient engines of the perceptual machinery can be the main inference engine.
>
> Complementary representational systems can express what each one by itself cannot express.
>
> Complementary representational systems can infer things that each one by itself could not infer efficiently.
>
> Finally, higher level domains can be built as analogical layers on primitive domains, and inherit their inference efficiency.

References

[Davis, 1990] Davis, E. (1990). *Representations of commonsense knowledge.* Morgan Kaufmann.

[Elkan and Greiner, 1993] Elkan, C. and Greiner, R. (1993). Book Review of *Building Large Knowledge-Based Systems: Representation and Inference in the Cyc Project* (D.B. Lenat and R.V. Guha). *Artificial Intelligence*, 61.

[Gazzaniga, 1985] Gazzaniga, M. S. (1985). *The social brain: discovering the networks of the mind.* Basic Books.

[Gödel, 1962] Gödel, K. (1992, c1962). *On formally undecidable propositions of principia mathematica and related systems.* Dover.

[Guha and Lenat, 1993] Guha, R. and Lenat, D. (1993). Re: Cycling paper reviews (response). *Artificial Intelligence*, 61.

[Guha and Lenat, 1990] Guha, R. V. and Lenat, D. B. (1990). Cyc: A midterm report. *AI Magazine*, 11(3).

[Hausser, 1989] Hausser, R. (1989). *Computation of language: an essay on syntax, semantics and pragmatics in natural man-machine communication.* Springer Verlag.

[Hofstadter and Dennett, 1981] Hofstadter, D. R. and Dennett, D. C. (1981). *The mind's I: fantasies and reflections on self and soul.* Basic Books.

[Johnson, 1987] Johnson, M. (1987). *The body in the mind: the bodily basis of meaning, imagination and reason.* University of Chicago Press.

[Kurzweil, 1990] Kurzweil, R. (1990). *The age of intelligent machines.* MIT Press.

[Lakoff, 1987] Lakoff, G. (1987). *Women, fire, and dangerous things: what categories reveal about the mind.* University of Chicago Press.

[Lenat and Feigenbaum, 1991] Lenat, D. and Feigenbaum, E. (1991). On the thresholds of knowledge. *Artificial Intelligence*, 47.

[Lenat, 1996] Lenat, D. B. (1996). Cycorp, inc. homepage. http://www.cyc.com/.

[Lenat and Guha, 1990] Lenat, D. B. and Guha, R. V. (1990). *Building large knowledge-based systems: representation and inference in the Cyc project.* Addison-Wesley.

[Lenat et al., 1986] Lenat, D. B., Prakash, M., and Shepherd, M. (1986). Cyc: using common sense knowledge to overcome brittleness and knowledge acquisition bottlenecks. *AI Magazine*, 6(4).

[McAllester, 1991] McAllester, D. (1991). Observations on cognitive judgements. *MIT AI Memo 1340.*

[McCarthy and Hayes, 1969] McCarthy, J. and Hayes, P. J. (1969). Some philosophical problems from the standpoint of artificial intelligence. In Meltzer, B. and Michie, D., editors, *Machine Intelligence*, volume 4. Edinburgh University Press.

[McDermott, 1993] McDermott, D. (1993). Book Review of *Building Large Knowledge-Based Systems: Representation and Inference in the Cyc Project* (D.B. Lenat and R.V. Guha). *Artificial Intelligence*, 61.

[Minsky, 1974] Minsky, M. L. (1974). A framework for representing knowledge. *MIT AI Memo 306.*

[Montague, 1974] Montague, R. (1974). *Formal Philosophy.* Yale University Press.

[Neches, 1993] Neches, R. (1993). Book Review of *Building Large Knowledge-Based Systems: Representation and Inference in the Cyc Project* (D.B. Lenat and R.V. Guha). *Artificial Intelligence*, 61.

[Papert, 1980] Papert, S. (1980). *Mindstorms: children, computers, and powerful ideas.* Basic Books.

[Polya, 1954] Polya, G. (1954). *Mathematics and plausible reasoning.* Princeton University Press.

[Putnam, 1981] Putnam, H. (1981). *Reason, truth and history.* Cambridge University Press.

[Rao, 1995a] Rao, S. (1995a). Multiple representational systems. Unpublished paper.

[Rao, 1995b] Rao, S. (1995b). A visuospatial representational system. MIT PhD proposal.

[Rao, 1996] Rao, S. (1996). Personal communication.

[Russell, 1945] Russell, B. (1945). *A history of western philosophy.* Simon and Schuster.

[Searle, 1980] Searle, J. (1980). Minds, brains, and programs. *Behavioral and brain sciences*, 3.

[Skuce, 1993] Skuce, D. (1993). Book Review of *Building Large Knowledge-Based Systems: Representation and Inference in the Cyc Project* (D.B. Lenat and R.V. Guha). *Artificial Intelligence*, 61.

[Smith, 1991] Smith, B. C. (1991). The owl and the electric encylopedia. *Artificial Intelligence*, 47.

[Sowa, 1993] Sowa, J. F. (1993). Book Review of *Building Large Knowledge-Based Systems: Representation and Inference in the Cyc Project* (D.B. Lenat and R.V. Guha). *Artificial Intelligence*, 61.

[Stevenson, 1996] Stevenson, D. C. (1996). The tech classics archive. http://thetech.mit.edu/Classics/.

[Stipp, 1995] Stipp, D. (1995). 2001 is just around the corner. Where's Hal? *Fortune.*

[Tarski, 1935] Tarski, R. (1935). Der wahrheitsbegriff in den formalisierten sprachen. *Studia Philosophica*, 1.

[Tuttle, 1993] Tuttle, M. S. (1993). Book Review of *Representations of Commonsense Knowledge* (E. Davis) and *Building Large Knowledge-Based Systems: Representation and Inference in the Cyc Project* (D.B. Lenat and R.V. Guha). *Artificial Intelligence*, 61.

[Whitten, 1996] Whitten, D. (1996). The unofficial, unauthorized cyc frequently asked questions information sheet. http://www.cais.com/wcmac/cyc/cyc-faq.text.

[Winston, 1992] Winston, P. H. (1992). *Artificial Intelligence.* Addison-Wesley, 3rd edition.