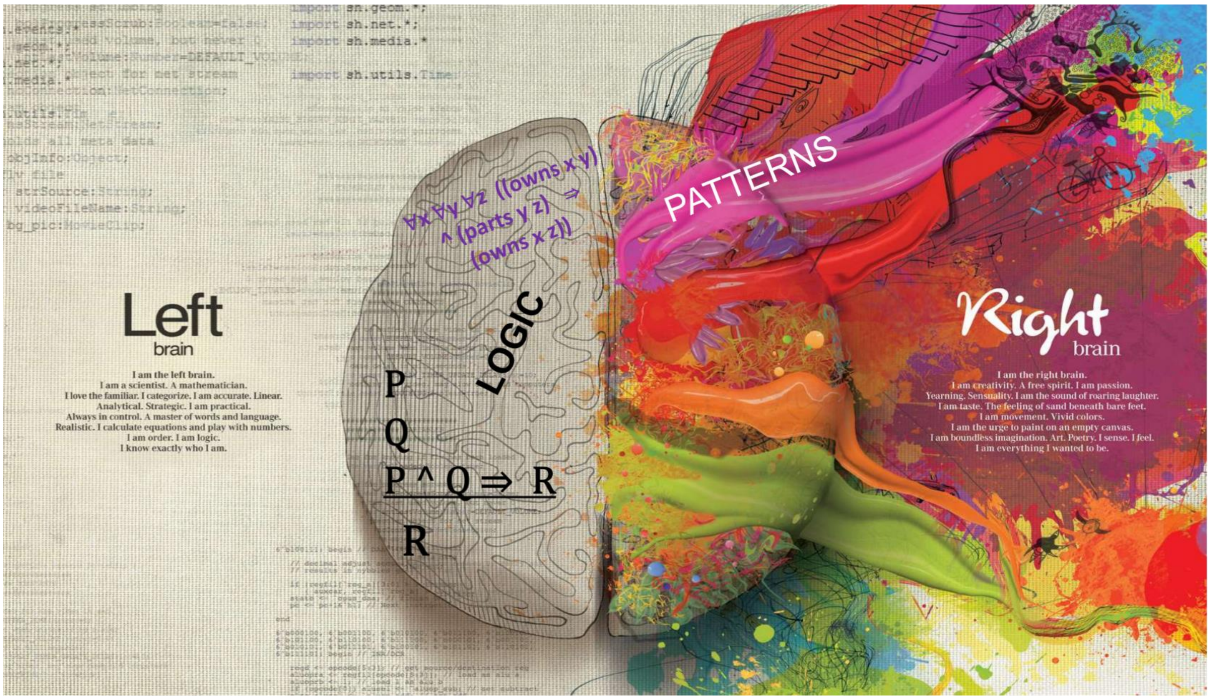

Cyc

Largest AI Project to Endow Computers with Common Sense

APTS YOERI, FAZELIAN PADIN, TIMMERMANS BART

"The leg of the syllogisers is of wood: a wooden leg is very infirm" – Rumi

ERASMUS HOGESCHOOL BRUSSEL – POSTGRADUATE ARTIFICIAL INTELLIGENCE

NOVEMBER 2021

Table Of Contents

Introduction

Spanning almost four decades, Cyc is the most ambitious long-running project in the history of artificial intelligence. Started in 1984, Cyc was a response by MCC – a consortium of major American computer and semiconductor manufacturers – to Japan's Fifth Generation Computer Systems project with the objective of instilling computers with human common sense and achieving artificial general intelligence.

The history of Cyc is closely tied to prominent AI researcher Douglas Lenat, who initiated the project in July 1984 at MCC as a Principal Scientist (1984–1994), and then in January 1995, founded a spin-off company, Cycorp, where he would continue the development of Cyc until today.

With the high ambitions of capturing common sense knowledge, Cyc focuses on implicit knowledge that people or other AI platforms take for granted. Cyc, meaning encyclopedia, aspires to endow computers with an encyclopedia of human common-sense knowledge. This implicit knowledge, metaphorically likened to the white spaces of a book, is in contrast with facts that can be retrieved or found on the Web. Cyc enables semantic reasoners to perform human-like reasoning and be less brittle when confronted with novel situations.

For over thirty-five years, AI researcher Doug Lenat and a small team at Cyc have devoted themselves to digitally codifying all of the world's common-sense knowledge into a set of rules. These rules include things like: "you can't be in two places at the same time," "you can't pick something up unless you're near it," and "when drinking a cup of coffee, you hold the open end up."

Drawing upon the world's largest general-purpose knowledge base of over 164,000 basic concepts (also called atomic terms or vocabularies) and 3,300,000 facts (rules and ground assertions) relating them, Cyc claims to have some degree of common sense (Curtis, 2005). Yet the number might be even counterintuitive, as Cyc has not led to artificial intelligence with common sense. As of 2017, it was estimated that the Cyc database contained close to 25 million rules and that Lenat's team had spent over 1,000 person-years on the project.

Cyc still lacks a quantitative measure of how far away its goal of capturing common sense is. In his latest interview in September 2021, Lenat is still hopeful that Cyc can do anything when it will have reached a certain level understanding of the world, if the system primes the pump enough to a level that self-reinforcing upward cascade becomes possible. Although passionately believing that Cyc will, one day, bootstrap itself through a combination of discovery and reading, he evades on giving a quantitative measure on the remaining path: "maybe it's the final 2%, maybe it's the final 99%."

Cyc is also well-known for being shrouded in secrecy and mystery. Lenat (2021) admits that the people behind Cyc keep "a very low profile", and do not publish or attend conferences, and the public information about the project is limited to the few publications they had decades ago.

The project's long haul, secrecy, and as-of-now failure to meet its ambitious goals have caused much criticism from the AI community, both from the thriving machine learning regarding it as the biggest failure in AI history, and knowledge representation communities.

History

The largest experiment yet in symbolic AI, Cyc came into existence in an era when expert systems were proliferating, and the developed countries were racing for funding AI projects. Regarded as the most enduring and ambitious project in the history of AI, Cyc has witnessed both springs and autumns of AI, and still pursues, in near secrecy, the ultimate goal of capturing human common-sense knowledge and artificial general intelligence.

In this section, we will take a panoramic view of the philosophical roots of symbolic reasoning, the commercial success of expert systems, the hiatus of the competing Connectionism school, and the trans-Pacific political competition over AI, all of which were crucial to giving birth to Cyc. We will then investigate how the development of Cyc continued despite the setbacks of expert systems, the abandonment of Japan's competing Fifth Generation Project, and the shift of AI paradigm in favor of machine learning.

Philosophical Roots

The basis of Symbolic AI is the Physical Symbol System Hypothesis, which is a strong hypothesis that states "A physical symbol system has the necessary and sufficient means for general intelligent action" (Newell & Simon, 1976, p.87). It assumes any intelligent agent to be necessarily a physical symbol system, and the human mind is a kind of symbol manipulation. It also assumes that a physical symbol system is all that is required for intelligent action, so machines can be intelligent because symbols are all a physical entity needs for intelligence.

In a sharp contrast to Plato's Meno, the Newell & Simon (1976) assertively argue that since the mid-1950s, computer science has been able to start grasping the ineffable human mind: "The study of logic and computers has revealed to us that intelligence resides in physical-symbol system" (p.108). Under this purely objectivist view, symbols, even though internalized in brains or computers, are indeed physical objects that constitute the real world, hence the name. "There is no magic or an as-yet-to-be-discovered quantum phenomenon required" (Poole & Mackworth, 2010, p.19).

The philosophical roots of the hypothesis are Hobbes's definition of reason as modelled on calculation, Leibniz' conviction that all human reasoning can be explained by mathematics and logical calculus, and Hume's epistemological view on distinct, distinguishable existence of all human perceptions.

With the earlier success of performing automated reasoning in their 1956 computer program Logic Theorist – dubbed as "the first artificial intelligence program" before the coinage of the term "artificial intelligence" – Allen Newell, Herbert A. Simon and Cliff Shaw introduced central concepts to symbolic AI research, such as reasoning as search, heuristics, and list processing.

Another pivotal figure in symbolic AI, John McCarthy, whose ideas would later become foundational in Cyc, believed that regardless of how human mind works, computers do not need to imitate human mind. McCarthy advocated declarative representations of knowledge in artificial intelligence and using formal logic to solve problems through, logical programming, such as Prolog and answer set programming (ASP). Formal logic represents statements and argument patterns symbolically using formal systems such as first order logic (or predicate logic).

The later development of symbolic AI, based on human-readable symbolic representation of problems, logic, and search, led to proliferation of expert systems, and would be abandoned in favour of the competing school Connectionism.

Proliferation of Expert Systems in 1980s

The first commercially successful expert system in 1978, XCON saved Digital Equipment Corporation $40 million throughout six years of operation (Simon & Kandel, 2018) and captured the attention of enterprises in emulating the decision-makings of human experts. Represented as a series of if-then rules, expert systems comprise of two divisions: the knowledge base, which represents facts and rules, and the inference engine, which reasons on stored facts using the rules to infer new facts.

In the 1980s, a whole new industry both in computer software and hardware grew to support the commercialization of expert systems. Specialized computers and workstations called Lisp machines were built to optimize Lisp, an expression-oriented programming language once preferred for AI.

The commercialization boom of symbolic AI led to the design and manufacture of the first commercial workstation known as Symbolics. The eponymous company Symbolics Inc. made these single-user, optimized computers to run Lisp and became one of the pioneers of software and hardware developments popular for commercial and research AI applications. The fact symbolics.com was the first registered .com domain attests to their immense commercial success. This would ultimately end with the collapse of Lisp workstation market in the late 1980s and early 1990s.

Cold Winter of Connectionism

The spring of symbolic reasoning and expert systems yet coincided with the winter of the competing school of AI: Connectionism.

A precursor to machine learning and larger neural networks, perceptron is a single neuron model resembling and simplifying a biological neuron. In 1969, the seminal book Perceptrons by Marvin Minsky and Seymour Paper was perceived as a harbinger of doom for connectionism and neural networks. This publication abruptly triggered a long hiatus on any connectionist endeavor in artificial intelligence, most of which had already abandoned in favor of exploring symbolic reasoning as the essence of intelligence.

It was not until mid-1980s that connectionist AI revived when Hopfield (1982) proposed that associative memories, in an analogy to spin glasses, can simulate neural networks.

Japan's Fifth Generation

By the late 1970s, Japan had already become the world leader in advanced chip design and manufacture thanks to a predecessor project: Promotion and Development of Technology for Next [Fourth] Generation Computers, or "VLSI Project."

In 1982, Japan's Ministry of International Trade and Industry took a massive research initiative called the Fifth Generation Computer Systems project (FGCS), designed to develop novel computer hardware using massively parallel computing and intelligent software developed using concurrent logic programming. The project would be funded with $850 million over the next decade (Hilts, 1983).

The joint government-industry research project aimed to create the peak performance of supercomputers that could serve as a platform for artificial intelligence ambitions. It was the first national, large-scale AI research and development project that was free from military influence and corporate profit motives.

As the 1980s began, the FGSC revived interest in artificial general intelligence (AGI) by setting out a ten-year timeline and ambitious goals: "Fifth Generation machines would carry on casual conversations, translate languages, interpret pictures, and reason like human beings" (Crevier, 1993, p.211). Although Japan was a close ally of the United States, and the FGCS did not plan any commercialized technologies, many American computer experts, most notably Edward Feigenbaum and Pamela McCorduck (1983), portrayed it as an economic threat to U.S. dominance in computing and the global economy.

Having already witnessed the commercial success of expert systems, the American government and industry thus pumped money back into the field of artificial intelligence in response to Japan's Fifth Generation Computer.

The Cyc project, ambitiously targeting artificial general intelligence in 1984, was one of these responses – the only one that has lasted to date.

However, at the end of the ten-year period, the FGSC had spent over $400 million and was terminated without having met its goals. In the late 1980s, the field of artificial intelligence similarly faced a confidence collapse due to how researchers grossly underestimated the limitations of AGI projects.

Birth of Cyc (1984)

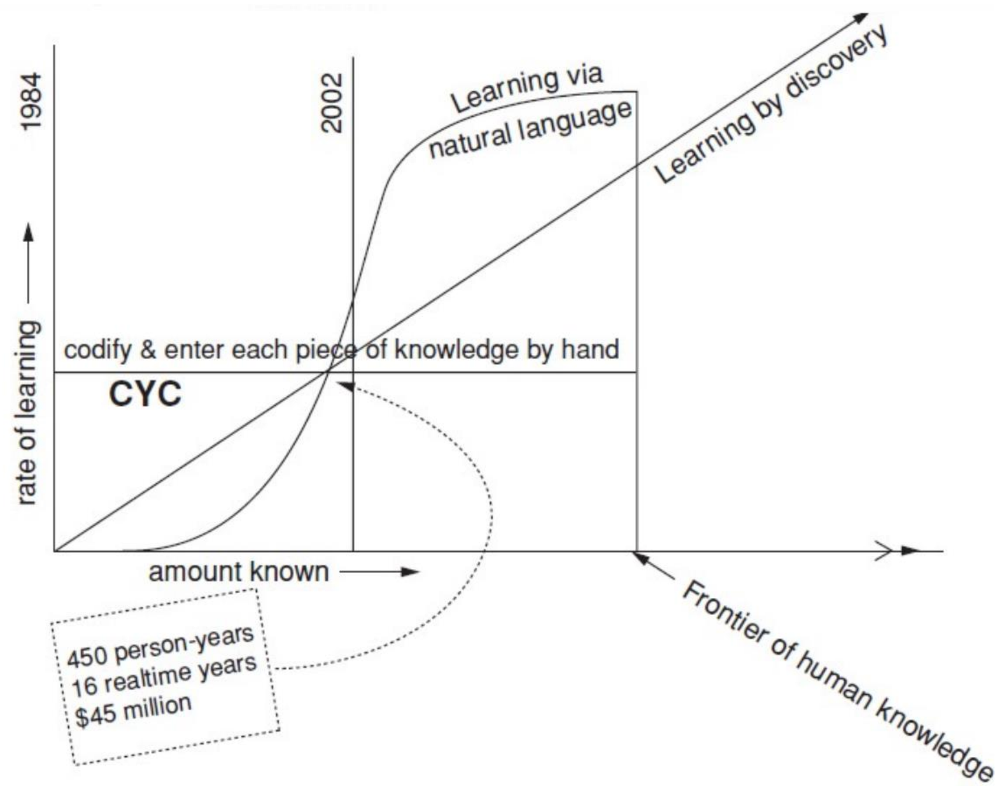

FIGURE 1 – THE GROWTH OF CYC AS ENVISAGED BY ITS CREATORS (EKBIA, 2008)

Many European and American computer companies saw this new Japanese initiative as an attempt to take full control of the world's high-end computer market. In 1984, Microelectronics and Computer Technology Corporation (MCC), under the US Congressional passage of National Cooperative Research Act, was created as a defensive move against that threat. The consortium came into existence on the support of Apple, Digital Equipment Corporation, Eastman Kodak Corporation, NCR Corporation and other large computer manufacturers and users.

In 1984 a US Congressional act authorizing formation of research consortiums enabled creation of MCC, the Microelectronics and Computer Consortium, as a defense to the Japanese technological threat. Later that year at MCC, the Cyc project started as a ten-year project to construct the largest expert system ever.

Much of Cyc's history is tied to Douglas Lenat, the key figure behind the project since 1984. In 1976, Douglas Lenat had been awarded with the prestigious International Joint Conference on Artificial Intelligence (IJCAI)'s Computers and Thought Award for his discovery system *Automated Mathematician (AM)*. The computer program had elaborate heuristics and creative behavior in the field of mathematics. Frustrated by its limitation of heuristics, Lenat developed his next program *Eurisko* in Lisp which consisted of heuristics describing how to use and change its own heuristics. Eurisko won a widely-recognized, sci-fi role-playing tournament for two consecutive years in 1981 and 1982. When the tournament officials announced that if Eurisko won another championship the competition would be abolished, Lenat retired Eurisko from the game.

The goal of this mega-expert-system project is unlike that of most traditional expert system approaches which focus on gathering a great deal of specialized information about a narrowly focused technical area – the most efficient way to deploy a telephone switching network, for example. Instead, the Cyc system is an attempt to collect and code common-sense reasoning: *A person cannot walk through a wall. Water falls downhill. All animals live, die and stay dead*. The goal of the knowledge base is the support of 100 million of these common-sense assertions, creating a system that is 10,000 times as large as an average expert system.

Similar to other expert systems, Cyc follows a knowledge-based architecture essentially composed of two sub-systems: the knowledge base and the inference engine. As a mega-expert-system, Cyc uses Ontology Classification that adds object classes to its knowledge base to make new types of reasoning possible.

Its initial proposed methodology was to encode the knowledge in 400 sample articles in a one-volume desk encyclopedia together with all the implicit background knowledge that a reader would need to understand the articles (hence the name) (Lenat, Prakash, & Shepherd, 1985).

Cycorp Spin-Off (1995)

In 1995, Douglas Lenat spun off the project as Cycorp Inc. Since then, Cycorp has remained a private company, and will not go public "because we want to have control over our future, over our state of being, so that we can continue to do this as until it's done and we're making progress and we're now so close to done" Lenat (Interview, 2021).

Over the past 37 years, most of the company's clients, the government and big corporations, have been shrouded in secrecy. Recently, Cyc has found a number of commercial applications in hospital chains for medical reasoning about patients, energy companies, and various manufacturers for reasoning about supply chains.

Lenat (Interview, 2021) revealed that "Five years ago, almost all of our money came from the government. Now, virtually none of it comes from the government. Almost all of it is from companies that are actually using it for something."

OpenCyc (2002-2017)

In 2002 Cycorp decided to open source parts of the knowledge base to the public under the name OpenCyc supported by the OpenCyc.org organization. The idea was to raise awareness for symbolic knowledge representation and to try to establish Cyc as the standard for knowledge representation, knowledge management, database integration and in general for intelligent software applications.

Besides the knowledge base, OpenCyc also contained a compiled version of the inference engine and several tools for interacting with the knowledge base and inference engine.

The most recent public version, OpenCyc 4.0, released in June 2012 contains 239,000 concepts and 2,039,000 facts, mostly taxonomic.

In 2017, OpenCyc was no longer publicly available.

ResearchCyc (2004-2019)

ResearchCyc started with an early beta release in 2004 with the goal to lower the barrier for researchers to have access to Cyc technology in order to attract interest in Cyc from the R&D community. Talks with leaders in R&D lead to a direction of modularization of Cyc, the creation of power mapping tools and easier integration capabilities (Lefkowitz, Curtis, & Witbrock, 2007).

ResearchCyc 1.0 launched in 2006 and was available upon written request, freely licensed for research purposes, contains 500,000 concepts and 5,000,000 facts. There was a fairly large interest of more than 225 research organizations being part of the ResearchCyc community in 2007. That continued to grow to about 600 research organizations in 2017. During that period, several new versions of ResearchCyc were released.

At the end of 2019, ResearchCyc is no longer supported.

Lucid AI (2008)

In 2008, Michael Stewart and Doug Lenat, started a new company called Lucid AI. “Part of the reason is the doneness of Cyc,” explains Lenat. “Not that there’s nothing else to do,” he says. But he notes that most of what is left to be added is relevant to a specific area of expertise, such as finance or oncology (Knight, 2016).

And that is exactly what Lucid AI is commercializing. Michael Stewart explains it as follows: “We interview subject-matter experts, and also use documentation of the company. We ingest that knowledge into Cyc much like you would with a human.” (Knight, 2016).

This results in the knowledge workers being able to interact in natural language, for example doctors input queries such as “Find patients with bacteria after a pericardial window.” Lucid should not only find the right candidate patients but provide a clear chain of logical reasoning for why it selected them.

How Does It Work

Humans are not born with common sense. We can describe common sense as the knowledge and understanding that we develop over the years and share with a particular group of people. Most of this knowledge is often not explicitly cited when making decisions as it is considered known be everybody and it would be awkward to state this common knowledge. Common sense always takes the context into consideration when reasoning. When we want a system to have common sense, all this implicit knowledge, often also referred to as the white space, would have to be made explicit and the system would need some sort of brain to reason based on this knowledge. To acquire all this knowledge, Cyc chose to grow a knowledge base containing all this knowledge and use an inference engine to reason with this knowledge (Matthias, AI and Society: 07a. CYC: What is common sense?, 2019).

Ontology (Knowledge Base)

Structure

Built over the course of the last 35 years, by 2000+ Ph.D. scientist-years' effort, Cyc's ever-growing common sense and domain-specific Knowledge Base (KB) is the broadest, deepest, most complete ever developed. It can understand real world contextual nuance, like culture, emotions, time, space, beliefs, and bias.

The Knowledge Base comprises:

- An ontology of about 1.5 million general concepts (e.g., taxonomically "placing" terms like eyes, sleep, night, person, unhappiness, hours, posture, being woken up, etc.)
- More than 25 million general rules and assertions involving those concepts (e.g., that most people sleep at night, for several hours at a time, lying down, with their eyes closed, they can be awakened by a loud noise but don't like that, etc.)
- Domain-specific extensions to the common sense ontology and knowledge base in areas such as healthcare, intelligence, defence, energy, transportation, and financial services (Cycorp, Technology overview, 2019)

The knowledge base has a pyramidal structure. On top, there are a set of abstract concepts aka the upper ontology. Further down the pyramid, we find core theories which are still very general, for example axioms about time, space, etc. Going down a level, we arrive at the domain specific theories which are much bigger in number and describe in a generic way the domain specifics. At the base of the pyramid, we have the actual real-world facts. This knowledge can be static or more dynamic in nature.

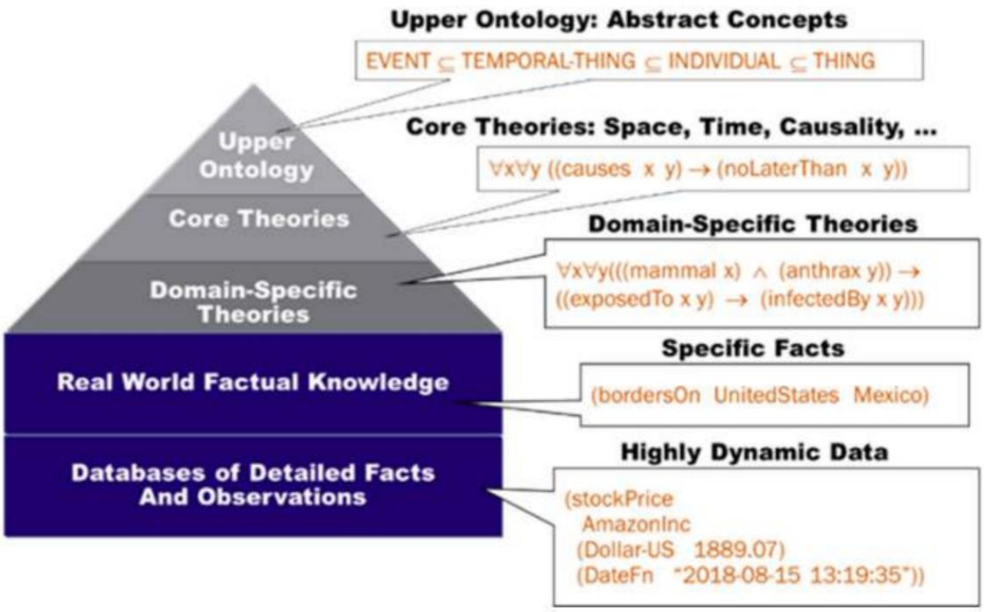

FIGURE 2 - ONTOLOGY PYRAMID

CycL - The Language

The choice of the representation language of this knowledge is crucial. It needs to be expressive, avoid ambiguity (I'm at the bank does not tell if you're in a financial institution or at a river bank) and would benefit from operators that can enable reasoning.

There are numerous possible representational languages (natural languages, Frames & slots, OO languages, logic-based languages). The Cyc team believed that a logic-based language would be the best fit to represent knowledge and they developed a language called **CycL**.

In order to understand CycL, we need to work ourselves through a few concepts. This is a compact summary based on the Cyc 101 Tutorial (Cycorp, Cyc 101 Tutorial, 2002) and (Matthias, AI and Society: 07c. How does CYC work?, 2019).

Constants

Constants, called CycLConstants in CycL, are specific individuals or collections and are written as a character string prefixed with '**#$**'.

A collection can in a way be compared to a class in OO. A collection has different instances. It is characterized by some common properties that all instances share. A collection can have other collections as instances, the collection #$Person contains a collection of #$MalePerson.

An individual is a concrete thing, an instance which can have parts.

Some examples:

Collections:

- *#$Cat => collection of cats*
- *#$TransportationDevice => collection of transportation devices*

Individuals

- 'partially' tangible: *#$BarackObama, #$RedDevilsTeam*

- relation: *#$and, #$isMotherOf, #$isa*
- Attribute values: *#$blueColor, #$greasy*

Formulas

Formulas or CycLFormulas, are a form of statement describing a concretization of a relation, which consists of the relation type and the subject of the relation. Syntactically formulas are written between parentheses.

Some examples:

- *(#$isa #$BarackObama #$Man) => Barack Obama in an instance of the collection of men*
- *(#$genls #$Cat #$Mammal) => every instance of the cat collection is an instance of the mammal collection or also known as the mammal collection is a generalization of the cat collection*
- *(#$isMarriedTo #$BarackObama #$Michelle) => Michelle is married to Barack Obama*
- *(#$ParliamentFn #$Flanders) => the parliament of Flanders*

Sentences

If we look carefully at the previous examples, we can see that the first two formulas express some sort of truth. We call these formulas **CycL Sentences** and they are characterized by a **truth function** as the first argument. Syntactically, a truth function starts with a lower-case letter. CycL sentences are used to form assertions and they can also be used in queries. A truth function is the more generic term for one of the three following:

- Predicate: a function describing the relation. As it is a truth function, the result will be True or False and typically for a predicate this result is a fact based coming from the represented world. In other words, in the real world we can say it's true that Barack Obama is married to Michelle.
- Logical connective: some logical relation like and, or, not. We will describe them in more detail later.
- Quantifier: some quantifying relation like everybody, somebody, at least 3. We will also describe them later on.

Non-Atomic Terms

The third formula does not immediately result into a truth but rather a new term, namely the parliament of Flanders. Such a formula is called a **CycL Non-Atomic Term (NAT)** and has a denotational function as its first argument. These functions describe a relation that can be applied to one or more subjects of this relation and result into something new, being it an individual or a collection. As a convention, these functions start with a capital letter and have a Fn suffix. Non-atomic terms can be used like any other terms in other formulas and greatly enhance expressiveness, reusability, and uniformity:

(#$residenceOfOrganisation (#$ParliamentFn #$Flanders) #$Brussels) => The parliament of Flanders resides (has its residence) in Brussels

Uniformity will be important as it can greatly improve inference efficiency for the engine. We can assert a lot of things on the new non-atomic terms based on the knowledge derived from the denotational function, but we'll describe the principals of the inference engine in more detail later on.

Constraints

Formulas give you a lot of flexibility in describing relations but these predicates or functions that describe the relation only make sense when used correctly. So CycL uses constraints to communicate the possibilities and expectations of the formulas.

Arity

Often a predicate of function that describes a relation expects a specific set of arguments. For example, for a denotational function #$MarriageFn you would expect two arguments. We call this type of constraint an arity constraint, described by the predicate *#$arity*. As you can expect, the constraint is written as a CycL sentence. In the example of the #$MarriageFn function, we express the constraint as

(#$arity #$MarriageFn 2) => the arity of #$MarriageFn is 2

Argument Type

Argument type constraints determine what kind of form an argument can take on. There are two argument type constraints.

The first one is ***#$argIsa*** describes that the argument is an instance of an individual or a collection. The template for argument type constraint looks like:

({argumentTypeConstraintFunction} {predicate} {argumentPosition} {term})

If we apply this to the #$MariageFn, we write

(#$argIsa #$MarriageFn 1 #$Person) => The first argument of #$MarriageFn must be an individual #$Person like #$BarackObama

The second one is ***#$argGenl*** describes that the argument must be some subtype of a particular collection.

(#$argGenl #$penaltyForInfraction 2 #$Event) => the second argument of #$penaltyForInfraction must be a type of #$Event,

such as the collection of illegal equipment use events in:

(#$penaltyForInfraction #$SportsEvent #$IllegalEquipmentUse #$Disqualification)

Complex Formulas

So far, we've seen formulas as standalone artifacts. The true power lies within the connecting these formulas into more complex interrelated relations. Having chosen a logic-based representation language, this is made possible through the use of logical connectives, variables and quantifiers.

Logical Connectives

A logical connective is a sentence having a truth function that takes a set of sentences as arguments. Some well-known logical operators as 'and', 'or', 'not' also exist in CycL as #$and, #$or, #$not with the same meaning as in other languages. Another logical connective uses the truth function #$implies and expresses an if-then relationship.

```
(#$implies
    (#colorOfObject #$Hair #$BlackColor)
```

(#$not (#$colorOfObject #$Hair #$WhiteColor))) => if the color of hair is black then it is not the case that the color of hair is white

Variables and Quantifiers

Variables and quantifiers allow us to generalize the assertions even more. It allows us to represent many pieces of ordinary knowledge. We can make assertions about everything, something, exactly x things, at least y things, etc. Again, we use (quantifier) predicates to describe these relations: *#$forAll, #$thereExists, #$thereExistsExactly*, etc.

(#$thereExistsAtLeast 3 ?LNG (#$isa ?LNG #$BelgianLanguage)) => there are at least 3 Belgian languages

Where ?LNG is the variable. *#$forAll* is called the universal quantifier (everybody...) and for brevity this can be left out when specifying formulas by just using the variable in the sentence

(#$implies

(#$isa ?A #$Cat)

(#$isa ?A #$Mammal)) => For everything, let's call it ?A, if it is a cat then it is a mammal

Microtheories

With CycL as a representation language, we are capable of writing a whole set of assertions to describe the world. There is however a problem. Writing assertions that are absolutely true or false is often very hard or quite often even impossible because of contradictions in truth depending on for example geographical location. Therefore, Cyc introduces the concept of a clear bounded context called a microtheory.

A microtheory is a bundle of assertions around a shared topic or shared source and it is important that within a single microtheory no monotonic contradictions are present. It is of course possible that assertions contradict each other in different microtheories. Having these clear boundaries also allows for simpler assertions which improve the speed of the knowledge building process and reduces the processing time within the engine.

The most important predicates to describe the relations between microtheories and formulas are #$ist and #$genlMt.

#$ist links a microtheory to a formula:

(#$ist #$FlandersBasketballMt (#$isa #$Dinos #$BasketballTeam)) => In the microtheory FlandersBasketballMt, it is true that Dinos is an instance of the BasketballTeam collection

#$genlMt links one microtheory to another in the sense that one microtheory inherits all assertions from the other. This #$genlMt is transitive:

(#$genlMt #$Basketball #$Sport) => Basketball inherits from Sport meaning that Basketball can see and use all the assertions from Sport.

(#$genlMt #$WomenBasketball #$Basketball) => As the predicate is transitive WomenBasketball can also use all the assertions from Sport

The transitive nature of #$genlMt is very powerful but also means that defining the boundaries of a microtheory and what assertions to put in it, is a daunting task as it needs the right balance between not too general but also not too specific.

The Quest for Knowledge

As humans, we learn every day. The world is always evolving. What we considered to be common sense 30 years ago, might raise an eyebrow or two today.

As a system willing to be able to reason about the world, it is of vital importance that the axioms, concepts, rules and relations are constantly evolving and open to be extended. From the early days on, feeding the knowledge base has been a primary concern.

Manual Feed

Cyc started with feeding the KB with information modelled by the Cyc personnel. These people were often very analytical philosophers laying the ground concepts of the ontology and the most abstract generalizations. At that time, the project had quite some traction and the Cyc team decided to provide an open source version of the knowledge base part of the system, OpenCyc, to gain more knowledge from the community.

In order to lower the barrier for entering knowledge into the KB, Cyc invested in Natural Language Processing (NLP). The idea was that to provide tools to translate back and forth between English and CycL (Panton, et al., 2006). At the core of these NLP capabilities is its English lexicon. The words and phrases and their syntactic properties are linked to Cyc concepts and rules in CycL.

In order to generate an English sentence based on Cyc concepts aka Natural Language Generation (NLG), Cyc uses generation templates which are part of KB and are intelligent in the sense that they can conjugate, etc.. It's easy to understand that the ability to generate natural language sentences from Cyc KB is a giant step to open up the knowledge to people and systems that are not familiar with CycL.

Natural Language Understanding (NLU) tries to process natural language and transform it into CycL. This can be as easy as mapping a string to a Cyc concept or more complex, translating sentences into CycL rules (Panton, et al., 2006).

It soon became clear that modelling the world by manually feeding, would take forever. Together with some ability to translate to and from natural language, new ways of feeding the knowledge base emerged.

Automated Import of Relatively Static Knowledge

There is an enormous amount of (structured) information available on the web in the form of for example web services or the semantic web. A lot of this information is fairly static over time and could expand the knowledge substantially. Performance wise it makes a lot of sense to store this data close to the inference engine. But mapping, merging, and integrating the different ontologies of all these services are a major obstruction for knowledge growth (Reed & Lenat, 2002). In the beginning, this process was an intense collaboration between trained ontologists and the domain experts of the respective service. Over the years, several integrations were done with Cyc and that lead to the creation of tools that enabled the domain experts to independently do the mapping the Cyc concepts and assertions.

CycL was extended to map external concepts to Cyc concepts. One of the simpler language extensions that illustrates this is

#$synonymousExternalConcept TERM SOURCE STRING

meaning that the Cyc concept TERM is synonymous with the concept named by STRING in the external data source SOURCE. This is the simplest sort of 1-to-1 term mapping (Reed & Lenat, 2002).

A good and successful illustration of extending the KB with data from external sources is the integration with the Federal Information Processing Standards (FIPS) where countries are mapped to their country codes. An (extremely simplified) assertion to map the concept might be

#$synonymousExternalConcept #$Belgium #$FIPS10-Information1995 BE

Where #$Belgium is the term already known in Cyc, *#$FIPS10-Information1995* is the source of the information and BE is the string as it is known in the external source.

Semantic Knowledge Source Integration (SKSI)

With the rise of more and more web ontology standards, structured machine based access to a broad set of dynamic data became available. In other to integrate these different forms of information, an alignment of the semantics of these sources independent of their syntax is inevitable (Masters & Güngörd, 2003). This can be done in a peer to peer manner or by using a third ontology which maps the sources to a common language. As Cyc has already a large set of concepts and rules in his knowledge base, it is a good candidate for this 'third ontology'. So Cyc engineers developed the Semantic Knowledge Source Integration (SKSI) technology which allows to map and integrate in a way that it extends the Cyc knowledge base with the knowledge of these external sources. The content is really seen as Cyc knowledge and therefore query-able and usable for inference while the actual content remains on the native source server (Reed & Lenat, 2002). For example, imagine an external database containing the birthdate of employees, the SKSI integration layer will map the column of the birthdate to the concept 'Birthdate' in CycL but the actual list of employees with their birthdate remains in the external database.

On a conceptual level, the ways to interact with (the KB of) Cyc are as shown below (Panton, et al., 2006)

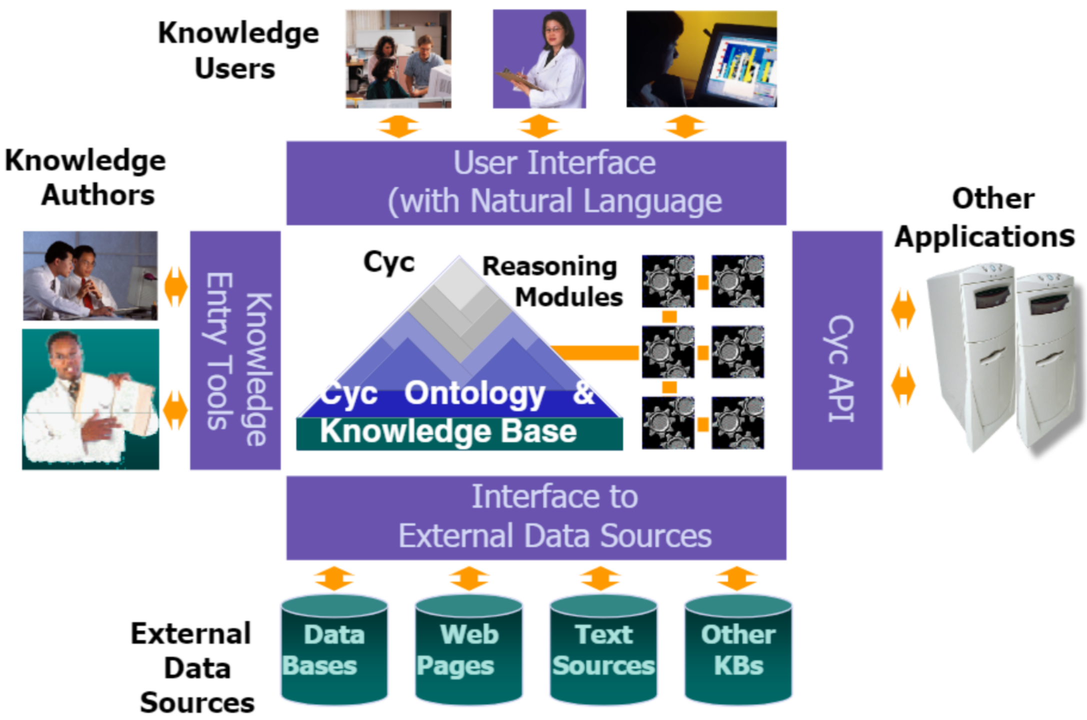

FIGURE 3 - INTERFACING WITH CYC

Inference Engine

Cyc has a huge knowledge base of more than 40 000 predicates, over a million concepts and more than 25 million explicitly stated axioms. All this knowledge is expressed in CycL which understand higher order logic and has an expressiveness which is relatively close to a natural language (English). All this knowledge serves the purpose of the original goal: to reason. Therefore, we need an additional component that can effectively use all this knowledge to reason: the inference engine.

A Higher Order Automated General Theorem Prover

The fundamentals of the inference engine are those of a higher order automated general theorem prover (Ramachandran, Reagan, & Goolsbey, 2005).

Higher order logic refers to the high degree of expressivity of the language and its interpretation capabilities. Ordered from least to most expressive, we can distinguish (Cycorp, Technology overview, 2019):

- **0th-order (Propositional Logic)**: Each rule deals with concrete logical objects and uses logical connectives like and, or, not and if/then. For example, take the premise: 'If it is raining, then it is cloudy'. If we know that 'it rains', then we can conclude that 'it is cloudy'.
- **Full 1st-order (Predicate Calculus)**: First-order logic uses quantified variables over non-logical objects, and allows the use of sentences that contain variables, so that rather than propositions such as "Socrates is a man", one can have expressions in the form "there exists x such that x is Socrates and x is a man", where "there exists" is a quantifier, while x is a variable. This distinguishes it from propositional logic, which does not use quantifiers or relations. In this sense, propositional logic is the foundation of first-order logic (Wikipedia, 2021).
- **2nd-order logic**: Second-order logic adds to first-order logic that variable can range over predicates (first-order logic variables range over individuals). It also add quantification over sets and predicates.
- **Higher-order logic (HOL)**: Higher-order logic takes it a few levels of abstraction higher so that the variables can range over statements, e.g., "Rule x was entered earlier than rule y, but by a novice". Modal operators are allowed, expressing that someone aims to make something true, or believes that it's true, or dreads it being true. Nested modal and temporal operators occur: "Fox News reported today that The US Supreme Court believed last year that high level White House staffers wanted to delay. . . ". If you can say something in English, you can capture its full meaning in HOL. I.e., any argument one might carry out in

English can be 100% captured and automatically carried out in HOL (Cycorp, Technology overview, 2019).

In general at macro level the process of finding proof can be described as in the figure below (Sundararajan, 2021):

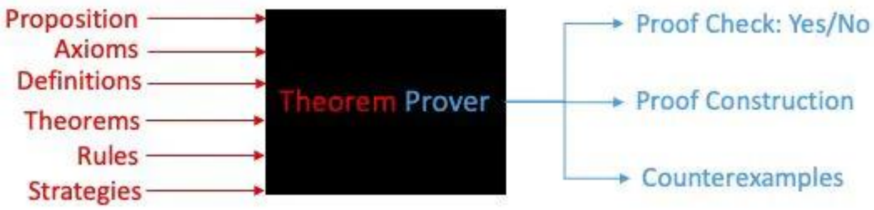

FIGURE 4 – MACRO LEVEL PROCESS FOR A THEOREM PROVER

Cyc is scarce with information of the exact inner workings of their inference engine. To give an idea of how an automated theorem prover (ATP) process can work, we describe the example of the ATP process from 'Thousand of Problems for Theorem Provers' aka TPTP (Sutcliffe, 2020)

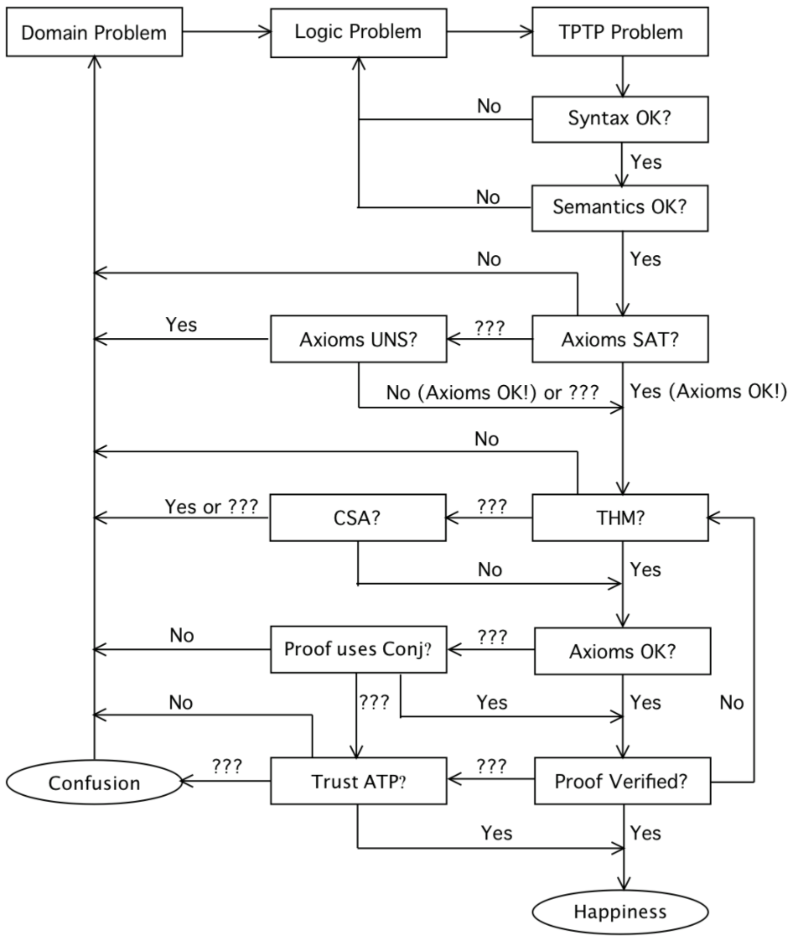

FIGURE 5 - PROCESS OF AN AUTOMATED THEOREM PROVER (ATP)

The following steps can be identified:

- Convert the domain problem into a machine form. This can be seen a preparing your inputs (and outputs) as shown in Figure 4 – Macro Level Process for a Theorem Prover
    - Start from domain problem into a more logical description to end up with a description of the problem in CycL (or for the TPTP ATP process in TPTP)
    - Good practice is to foresee in a verification of the syntax of the problem and a semantics check
- Check that the axioms are consistent and satisfiable or unsatisfiable. The system also provides a path for unknown outcomes.
- Establish a logical consequence based on the 'OK' axioms (the satisfiable and unsatisfiable ones)
    - If it's a theorem, go to the step of proving the theorem, otherwise let's see if we can prove it is not a theorem. If the answer is No to the question 'Is it not a theorem?', we can continue to process the proof
- Process the proof by checking if the axioms are known to be satisfiable or if the conjecture is used in the proof
- Verify the proof (using for example semantic derivation verification. Depending on the result, we're done (happiness) or have to try a different ATP system

This is a basic process for an ATP but the inference engine has to cope with a much broader and more complex set of knowledge.

A Multitude of Small Engines Working Together

Cyc claims that what we've described here as one inference engine is exactly a collection of more than 1100 smaller inference engines working together. These smaller engines are good at performing a specific task being it checking a rule in a database or traversing a graph on a generalization or performing a transitivity reasoning.

Cyc incorporates the following types of reasoning:

- **Deductive**: In this type of reasoning, we start with premises that result in a conclusion that is smaller than the premises. A common form is the syllogism in which a general broad premise is combined with a less broad premise to infer a conclusion (Bradford, 2017). For example:

  All basketball players are sportsmen. LeBron James is a basketball player so we can deduct that LeBron James is a sportsman.

  A conclusion from deductive inference is certain as long as the premises are true. Deductive reasoning is a very common practice.

- **Inductive**: Unlike deductive reasoning, inductive reasoning starts from one or more specific premises to make a more general conclusion (Bradford, 2017).

  Belgians wear a warm coat in winter therefore all people wear warm coats in winter.

  In other words, the premises give a certain probability that the conclusion is correct. Because this type of reasoning introduces a level of uncertainty it has long been contrived but is more accepted nowadays.

- **Abductive**: This type of reasoning chooses a hypothesis which if true, would best fit the evidence. So it starts from a number of facts and goes back and search for a hypothesis most likely resulting in the evidence (Bradford, 2017). For example:

An apartment on the 3rd floor has its door forced and has all drawers from the bedroom closet opened. Furthermore, the police caught a thief stealing in the apartment on the floor below. The likely hypothesis is that the thief also broke into the appartement on the 3rd floor

So, it is a useful type of reasoning to come up with different possible hypothesis and assert their validity. This can proof very helpful in belief revision in which beliefs are adapted based on new facts/information.

CycL can express practically everything that can be described in a natural language. This expressiveness makes it hard and resource intensive to reason about things. As the knowledge base is very large, finding your way in all the information in a fast and efficient manner is crucial. In order to overcome this hurdle, Cyc expresses its knowledge on two different levels:

- Epistemological Level (EL): this is the higher level which has the expressiveness of a natural language, alternatively also called the Constraint Language (CL)
- Heuristic Level (HL): this is the lower level which is easy to interpret into reason with in more traditional programming language like Java.

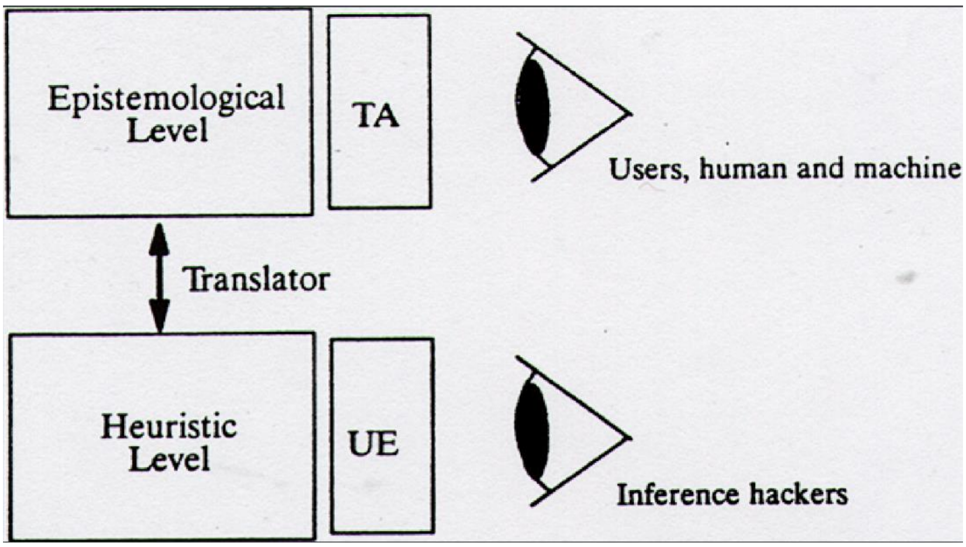

FIGURE 6 - EPISTEMOLOGICAL (EL) AND HEURISTIC LEVEL (HL)

The inference engine does all the hard work at the Heuristic Level.

A user can interact on the Epistemological Level with the system through a Tell/Ask interface (Derthick, 1990).

Asking can be done extensionally (based on existing examples in the KB) or 'intentionally' (based on proof, a matching rule). For example, if we ask if '*all basketball players are sportsmen*', and we only find basketball players in the KB that are also sportsmen, then asking extensionally will return a positive result. On the other hand, if we find a rule in the KB stating that '*all basketball players are sportsmen*' or some rule which can be considered equal, asking intentionally would give a positive result (Derthick, 1990).

Telling the interface is by definition intentionally and adds rules to the knowledge base. Whenever possible, these rules are also automatically translated to the heuristic level.

Certainly, in the early phases, skilled persons interacted with the Heuristic Level through the Unit Editor (UE) to enter rules directly at this level.

Because Cyc has all this knowledge at the HL, it can reach good performance because the operation for reasoning at this level require system operations at which computer are typically very well suited for. Let's take the rule “Cats are at least partially tangible”. At the EL, Cyc represent this rule as (genls Cat PartiallyTangible) (i.e., anything that is a cat is also at least partly tangible—being something with a weight and volume, etc.). This is a generalization that can be defined at the Heuristic Level in the genls hierarchy, and there exist multiple links between Cat and PartiallyTangible: Mammal, Animal, etc. A computer can efficiently travers this hierarchies. Furthermore, because transitive closures are precalculated and persisted in Cyc, this type of inference resolves very fast (sometimes in one step). By representing a cat and its rules in so many ways, there is a need for a system to select the correct rules. In fact, all the HL modules form a sort of team that are willing to work on a question: A query is offered to the HL modules and those who think they can help, tell the system that they might be useful in resolving the query. The system then picks a few to work on the query. That team member can then ask again for help on a part of the problem. All this is done recursively to resolve the query. Cyc spends roughly about 90% of its time in this type of problem solving.

The other 10% of its time is spent on another important task to make the reasoning process efficient: meta-reasoning. One can really see this as sitting back and rethinking the way it is currently trying to solve the current query. Should we use a different tactic? Maybe try to disprove instead of trying to prove? Cyc uses meta-meta-level rules to support this process. In contrast to other knowledge-based systems, Cyc is not afraid to add more and more meta- and meta-meta-knowledge to its knowledge base. Cyc strongly beliefs that this is crucial to reach expert level.

Applications

In this chapter we discuss commercial and academic applications of Cyc. If possible, we evaluate the application.

BELLA: A Learning-By-Teaching System

This paragraph is based on the paper "Reinforcing Math Knowledge by Immersing Students in a Simulated Learning-by-Teaching experience" (Lenat, 2014).

This paper discusses an application of Cyc to function as an Intelligent Tutoring System (ITS) for teaching mathematics, focused on pre-algebra, to 6th grade students. In the USA, 6th grade students are typically 11-12 years old.

In many teaching systems, the system takes the role of the teacher, but in this application the Cyc based system controls a teachable agent (tutee) which the user (the student) has to teach by helping it solve math questions. The user observes the agent and gives it advice, corrects its errors and mentors it. As the user gives good advice, Cyc allows the agent to make fewer mistakes and hence, from the user's point of view, it seems as if (s)he has taught something to the agent. This approach is referred to as learning by teaching.

User Experience

The main users of this system are 6th grade students and their teachers.

The learning environment is offered to the 6th grade student as a 3D sci-fi adventure game.

The game backstory situates the user as a 12-year-old living in a distant future. Elle is a robot companion permanently linked to the user mentally. The starship carrying Elle and tens of thousands of other inactive companions is attacked and crashes onto a planet's surface. All the equipment to activate them is damaged, but Elle has woken up just before the crash. So, Elle finds herself stranded with only a mental communications link to the use. The user must help Elle traverse multiple challenges at ever-increasing "scale": exploring and exploiting the crashed ship, the nearby area, adversaries, buried alien ruins, etc.—eventually triumphing by repairing the ship and awakening all the other robots.

The students navigate this 3D world and make progress by helping Elle solve pre-algebra puzzles. Each puzzle is presented to the user as a set of multiple-choice questions. By selecting an answer, the student tells Elle how to solve the puzzle, the student will not actually solve the puzzle for Elle; the student will learn by teaching Elle.

There is also a dashboard-like UI for teachers which enables them to see the evolution of each pupil on a predefined set of topics concerning knowledge of certain pre-algebra concepts, skills of using certain methods, handling of common mistakes etc.

System Architecture

This system is based on Cyc, with the following extras:

- The existing ontology (of 500.000 terms) was slightly extended
- The existing knowledge base (of 10.000.000 assertions and rules) was extended
- The representation language was extended

- Changes and additions were made to the inference engine: Special microtheories were developed to help manage the user, Elle, the storyline and other aspects of the system
- Extensions were made to its natural language generation and explanation subsystem

This extended version of Cyc has been named BELLA.

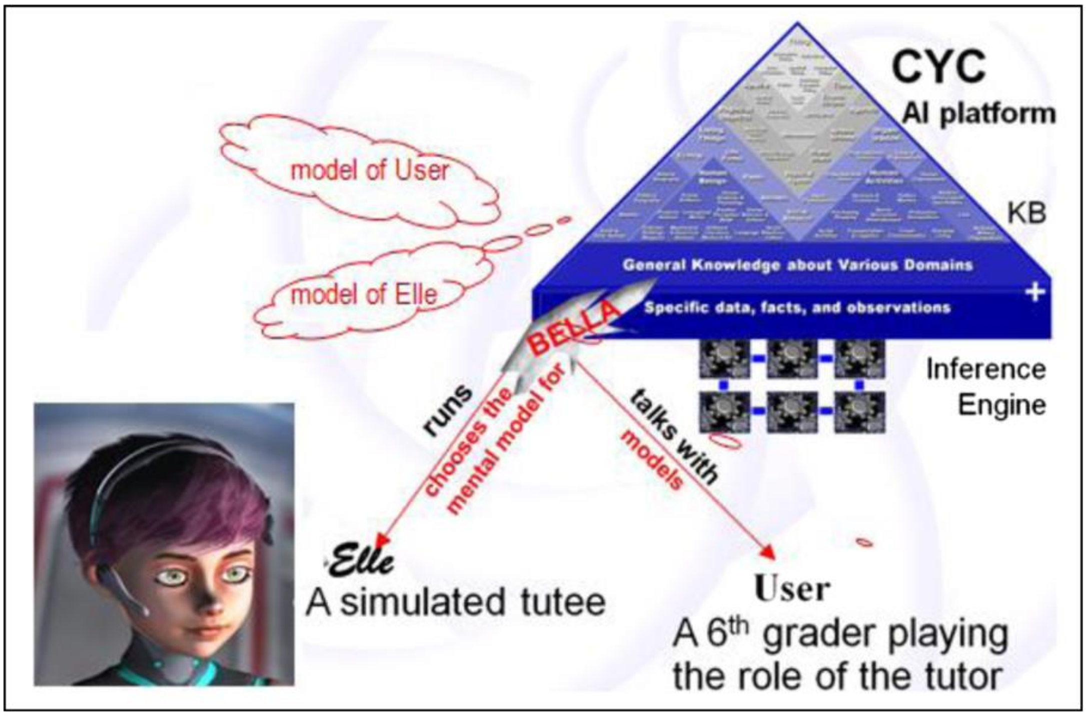

FIGURE 7 – MAIN COMPONENTS OF BELLA; CYC'S KB AND INFERENCE ENGINE; MODELS OF ELLE AND THE USER (LENAT, 2014)

BELLA works with a symbolic model of the human student and a model of Elle, the tutee robot.

Elle's mental model is started with knowledge of pre-algebra which is almost at the same level as the student. The paper does not exactly state how this model is initialized, it only alludes to some possibilities, like pre-testing the student in a traditional way, interviewing the teacher, starting at 'average' values etc...)

The student model comprises about 150 tuples, each of which corresponds to either a core concept (such as the associativity or commutativity of addition), or a core skill (such as how to multiply two fractions), or a specific type of "bug" or "missing heuristic" which students may have.

Each of these 150 model elements is scored as a value from 0 (meaning that the student does not understand this concept, cannot do this skill, never avoids this bug) to 100.

BELLA stores the entire history of every model parameter' score; the 0–100 value at which it started (for that student), and every incident, action or inaction by the user which caused that number to go up or down.

The teacher has access to that full history through a dashboard-style UI enabling him/her to follow the evolution of the student.

Main Use Case

BELLA maintains the student model and uses that model to determine the nature of the next problem to be presented. In addition to influencing the nature of each problem, the student model also determines Elle's problem-solving behavior and her interactions with the student.

Elle has multiple correct and incorrect parameterized models, and the choice of models (and parameters) is made by BELLA based on the current student model. Thus, strictly speaking, Elle does not learn and is not a teachable agent. Rather, BELLA selects a model for Elle to follow, deemed to be pedagogically advantageous, based on the state of the student model.

Cyc's KB is divided into microtheories each of which has a set of contextual assumptions and some body of content; *the content is internally consistent*, and the assumptions help prescribe the contexts in which that content is asserted to hold true. This context mechanism allows for the existence of different, contradictory mental models, sets of beliefs and opinions, different strategies, and tactics for tackling problems, etc.—all of which are used in BELLA.

Math word problems—and the in-game situations they describe—are represented in Cyc as declarative expressions with open variables in predicate calculus using the Cyc language (CycL).

Correct mental models for Elle were developed, and the various incorrect models are generated (in some cases by hand, in some cases automatically) by extirpating portions of the KB. For a particular math problem, BELLA can block Elle from solving it by going through each Cyc argument in each solution to the problem and either:

- removing at least one assertion
- adding in an incorrect assertion which overrides the correct still-present ones

In effect, these changes represent misunderstandings and errors in Elle's mental model.

When BELLA prepares to ask a question to a student, it uses Elle's 'incorrect' mental models to generate the 4 most probable answers. These expressions are then translated into English using Cyc's natural language generation subsystem and given to the student to answer.

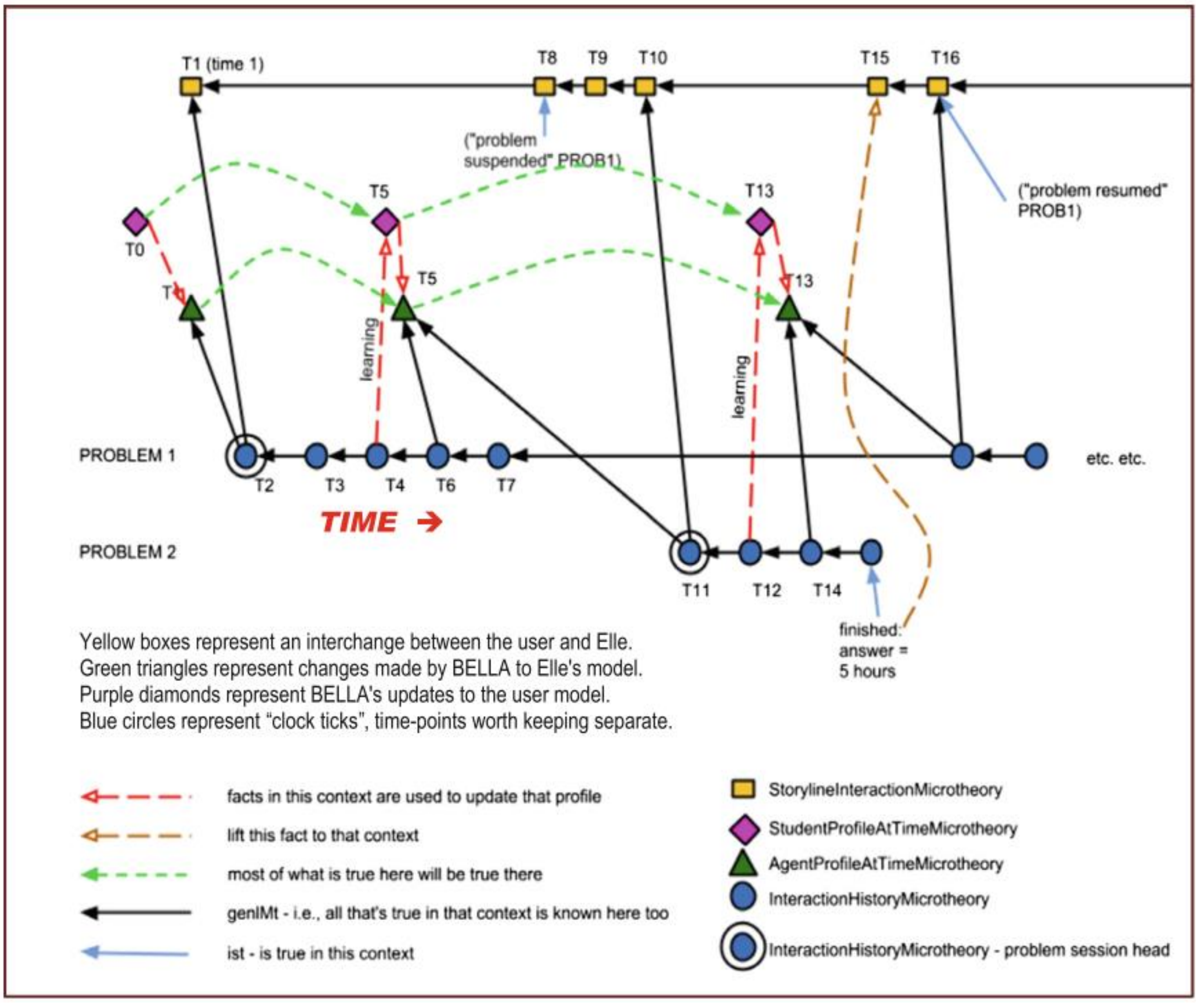

FIGURE 8 – USER AND AGENT (ELLE) AND OTHER MODEL EVOLUTIONS IN BELLA (LENAT, 2014)

The figure above illustrates how BELLA's models of the user and Elle evolve over time.

Evaluation

The reviewed paper (Lenat & Durlach, 2014) was published in 2014.

The paper states that, at that time, BELLA was not the first and not the only system that used AI in a learning-by-teaching setup. *Yet, some credit for this novel approach feels justified.*

The paper mentions that their system has only been tried by 26 sixth graders and their teacher and that they intend to start alpha-testing 'soon' and hope to roll out the system to real users. *In 2014 this system was clearly still a prototype.*

BELLA does not cover all mathematics usually taught at 6th grade level; it focuses on pre-algebra; basic arithmetic and solving linear equations with a single variable. Some attention is given to working with units of physical quantities, like for example 'Volts'. *In summary, the field of application is very limited.*

Today, in November 2021, seven years after this initial paper, there is little information to be found on the evolution of BELLA.

There is a website created by Cycorp which announces “Mathcraft” where students will learn math by teaching an android called ‘L’, an obvious reference to ‘Elle’ from the BELLA system (see https://cyc.com/mathcraft/).

The site claims that MathCraft is currently under development and will be commercially released within the next twelve months. Before the commercial release, a public beta-release will be made available to a wide range of players who can help find bugs and help give suggestions for improvement. *We signed up for this public beta but received no further information.*

We are very much looking forward to getting involved in this beta-release of MathCraft, but in the meantime *we remain skeptical as to its final capabilities.*

Cleveland Clinic Foundation

This paragraph is based on the article “Harnessing Cyc to Answer Clinical Researchers’ Ad Hoc Queries” (Lenat et al., 2010).

By extending Cyc’s ontology and knowledge base approximately 2 percent, Cycorp and Cleveland Clinic Foundation (CCF) have built a system to answer clinical researchers’ ad hoc queries. A query is at first parsed into a set of CycL higher-order logic fragments with open variables. After applying various constraints (medical domain knowledge, common sense discourse pragmatics, syntax), there is only one single way to fit those fragments together, and one semantically meaningful formal query is created.

To process this query, the Semantic Research Assistant (SRA) system dispatches a series of database calls and then combines, logically and arithmetically, their results into answers to the query.

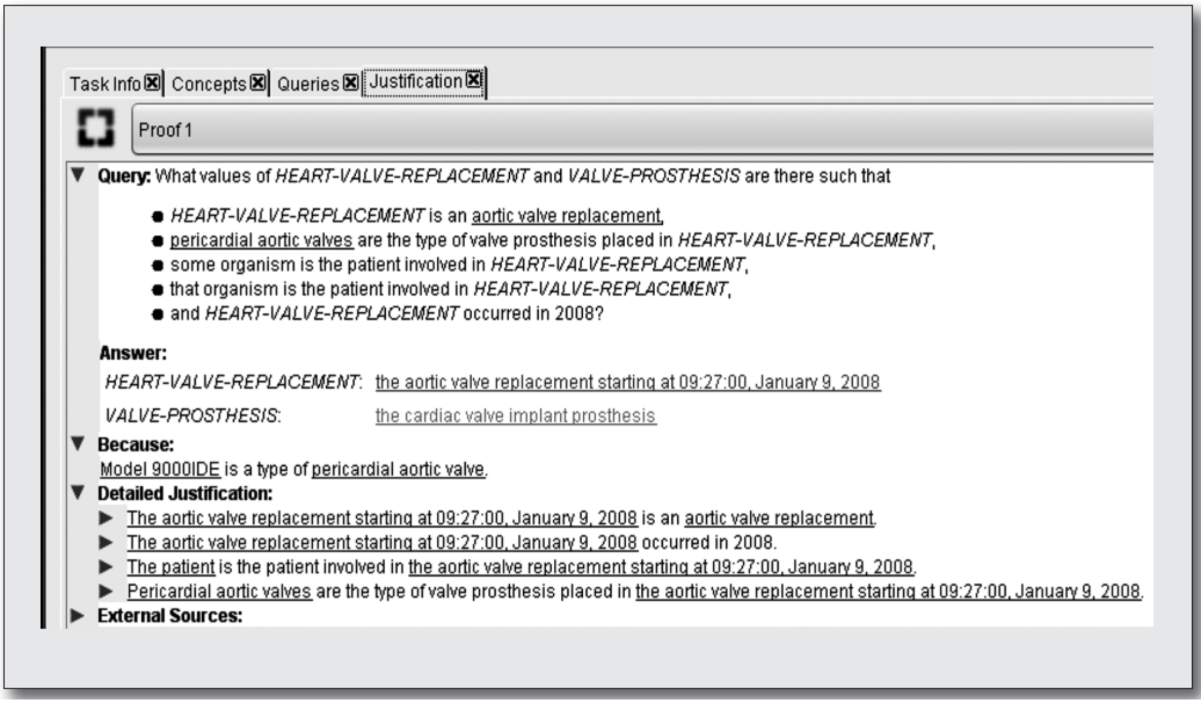

Figure 9 – The SRA Shows the Justification (Logical Proof) (Lenat et al., 2010)

A bundle of 275 queries is rerun quarterly by CCF to produce the data it needs to report to the Society of Thoracic Surgeons and another bundle covers the American College of Cardiology (ACC) reporting.

MySentient: A Commercial Question-Answering System

This paragraph is mainly based on the paper "On the Effective Use of Cyc in a Question Answering System " (Jon Curtis, 2005)

Cyc's role in the MySentient system heralds its first appearance in a commercial question-answering system. MySentient makes use of Cyc pervasively, to augment NLP-based QA, as the basis for a deductive question-answering module, and in other capacities, such as clarification and profiling (Jon Curtis, 2005)

"MySentient Answers 1.0" is a working question-answering system, designed by MySentient Ireland (R&D) Ltd. of Dublin, Ireland, and implemented in collaboration with Cycorp, Inc. of Austin, Texas, and the Center for Natural Language Processing in Syracuse, New York. MySentient Answers has been the subject of extensive demonstration to interested commercial parties and is expected to be available for public access in the near future. MySentient uses a carefully chosen subset of the full Cyc knowledge base with 137,000 concepts and 1,700,000 facts. (Jon Curtis, 2005)

Cyc is pervasively used in the MySentient system. For example, in its "Query Expansion' module". Query expansion is the process of altering an input question or a (quasi-)formal representation thereof, typically by adding or replacing terms: (Jon Curtis, 2005)

- Key phrases in the user's question are identified, translated into CycL, and placed in the discourse model. The User Profile Manager then reasons over this formal representation of the meanings of query-words to identify concepts that form the semantic basis for expansions (Jon Curtis, 2005)
- The highest-confidence strategy queries the knowledge base with a CycL representation of the user's question. As such, this strategy depends on the total success of the Natural Language Preprocessor module in mapping English to CycL. (Jon Curtis, 2005)

Evaluation

We found that:

- A trademark application for MySentient was filed on the 14th of January 2004, but by 26th July 2006, this application was abandoned (MYSENTIENT Trademark, n.d.)
- The company "MySentient Ireland (R&D) Ltd. of Dublin, Ireland" is also flagged as inactive (MYSENTIENT IRELAND (R&D) LIMITED INACTIVE, n.d.)
- The website for the Center for Natural Language Processing - Syracuse NY (https://cnlp.org) is no longer available.

In conclusion, it looks like MySentient has not been commercialized into any system or service that is actively used today.

Terrorism Knowledge Base

This paragraph is taken from the report "Terrorism Knowledge Base (TKB)" (Lenat & Deaton, 2008).

The objective of this project was to support intelligence analysts by developing a comprehensive Terrorism Knowledge Base (TKB) which included information about terrorist events and terrorist groups and their members and activities, as well as information captured by the analyst's use of the tool. Using that knowledge base, plus the knowledge base and inference engine of our company's Cyc(r) technology, the TKB was to exhibit sophisticated reasoning using domain knowledge, externally stored data, common sense knowledge and knowledge about what the analyst has considered relevant or irrelevant and templated question-answering with explanations. Analysts were to be able to use the TKB to pose terrorism-related queries, and to help them derive answers to those questions, integrate data, correlate observations, compose explanations, and, in general, augment their ability to effectively complete the reasoning tasks that they need to perform (Lenat & Deaton, 2008).

FACTory

FACTory was a web-based game that asked people to evaluate statements Cyc generated so it could extend its common-sense knowledge.

This game has been taken down from the web, but archived versions can be found:

> FACTory is a game that lets people enter knowledge into the Cyc Knowledge Base, a large store of common-sense knowledge.
> Cyc is trying to determine the truth or falsehood of a series of facts. It will ask about things someone has told it, things it has read about on the web, and things that it's just guessing about. You will be presented with these facts and asked whether you think they're true or false. As people play the game, Cyc accumulates votes; when enough people have agreed that a fact is true or not, Cyc considers it confirmed and stops asking about it. (How To Play, 2008)

Apparently, FACTory did not only offer 'true' and 'false' as possible answers, but players could also say the statements didn't make sense, or the user did not know the answer. (Katharina Siorpaes1 and Martin Hepp1, 2008)

Some anecdotal experience shared by a user:

> "I visited cyc.com and played the "FACTory" trivia game, where Cyc give the player the assumptions it has made from the facts in its database and asks if they are true, false, or don't make sense at all. One true assumption Cyc had made was, "Devices are typically located in toll booths," but I had to think about it. "Condominiums are typically located in modern homes," was an obviously false assumption, and "Ones are typically located in police stations," failed to make sense to me or any of the other players either." (Somma, 2008)

Position within AI Landscape

Since its inception, Cyc has had the ambitious goal of emulating human common sense through encyclopedic representation of human knowledge and formal logic reasoning. Douglas Lenat's optimistic prediction was that by 2015 "no one would dream of buying a machine without common sense, any more than anyone today would buy a personal computer that couldn't run spreadsheets [and] word processing programs" (Lenat, as cited by Copeland, 1993, p.103).

Spanning over 37 years, the Cyc project is the longest AI project in the history of artificial intelligence, enjoying consultations of influential AI figures such as Stuart Russell and John McCarthy. It was in embryo in the AI spring of expert systems proliferation and formal logic approaches, and it has survived the winters of expert system collapse paradigmatic shift to machine learning.

Missing Targets: It Must Be Something Philosophical

Cyc, as a mega-expert system and the largest symbolic AI effort in existence today, was supposed to tackle the brittleness of expert systems by capturing common sense. Lenat saw this brittleness as a symptom of lack of knowledge, and to him, the only general solution was to equip expert systems with common sense.

The goal of Cyc was to solve AI by entering into a computer all the necessary knowledge. In the beginning, Lenat, confidently predicted success within a decade. Thirty-seven years later, Cyc continues to grow without end in sight. "Maybe it's the final 2%, maybe it's the final 99% remaining," Lenat (2021) says in his latest interview.

No matter how many times Cyc would redefine or extend its targets, this ambition of creating artificial general intelligence inevitably runs into this basic problem: "the problem of dealing with the chaotic complexity of real-life" (Hankins, 2005).

It is due to this "unbounded complexity of real world" that an exception needs to be defined for every common-sense rule one can think of. The human mind somehow manages to grasp this common sense; "however it does it, it is not through a brute-force, hand-crafted knowledge base" (Toews, 2021). Oren Etzioni, CEO of the Allen Institute for AI puts it in this way: "Common sense is the dark matter of artificial intelligence. It's a little bit ineffable, but you see its effects on everything" (Toews, 2021).

The philosophical foundation of Cyc is based upon the Physical Symbol System Hypothesis discussed above. A prominent critic of this hypothesis is philosopher John Searle (1980) who believes that thought, or mentation as he calls it, is a biological experience. In his *Chinese Room Experiment,* he uses provocative scenarios to incite intuitions. Copeland (1993, pp.132-133) quotes Searle on this:

> *"Searle has another card in this sleeve. If the symbol system hypothesis is true, he says, then a computer made of toilet paper can think; so anyone inclined to believe the symbol system hypothesis must be prepared to accept that rolls of toilet paper are the right kind of stuff to think."*

For Searle, the idea that a brain could be made from such improbable substances is ludicrous. Similarly, American philosopher and renowned critic of artificial intelligence Hubert Dreyfus (1972) – by citing research on how brain neurons work – refuted the *Biological Assumption* that sees human brain as a physical symbol system, thus rejecting the Physical Symbol System Hypothesis. In a similar critique, Monberg (2006) believed that the project ignores the social ground of intelligence and meaning, rather emphasizing a calculating and all-controlling form of rationality that is one-dimensional.

Even if the biological assumption were true, Dreyfus (1993) rejected the *Epistemological Assumption* that symbol reasoning machines could represent knowledge of human beings, as human knowledge is not symbolic. Dreyfus also refuted the *Ontological Assumption* behind expert systems, arguing that human knowledge does not have a formal structure that can be fully described in terms of rules. In his seminal book *What Computer Can't Do*, Dreyfus (1993, p.xxii) specifically anticipated Cyc's failure:

> *The problem is that the rules and meta-rules are just more meaningless facts and so may well make matters worse. In the end, Lenat's faith that Cyc will succeed is based neither on arguments nor on actual successes but on the untested traditional assumption that human beings have a vast library of commonsense knowledge and somehow solve the scaling-up problem by applying further knowledge.*

Copeland (1993, p.120) had predicted that Cyc would fail to emulate human common sense because of the false grounds of the Physical Symbol System Hypothesis: "It would be a huge irony if the symbol system were true." However, he welcomed the idea that the Cyc project heralded a new rapprochement between AI and philosophy. As the major portion of the project's effort had gone into these three areas, Cyc was the first project that the 'hard' sciences would no longer sneer at the abstract fields of ontology (the theory of what there is), epistemology (the general theory of knowledge) and logic.

Lenat and Feigenbaum (1991) elucidated the philosophical foundations behind Cyc in their paper *On the Thresholds of Knowledge*, equating intelligence with knowledge (p.186) in their Knowledge Principle, which basically argues that minimum amount of knowledge is required for solving a problem:

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

An early critic of this Knowledge Principle is Smith (1991), who believed that artificial intelligence is nowhere near developing systems that are genuinely intelligent, and unlike what Lenat and Feigenbaum propose, knowledge and intelligence require participation in the world. Smith (1991, p.254) considers the Knowledge Principle as highly debatable:

> *As students of AI are increasingly realizing, there's no reason to believe that people formulate anything like all the problems they solve, even internally. Children happily charge around the world long before they acquire any conceptual apparatus (such as the notions of "route" and "destination") with which to formulate navigational problems. So too with language: fluent discourse*

is regularly conducted in complete absence of a single linguistic concept--
including "word" or "sentence".

Lenat (2021) still believes that if they prime the knowledge pump enough, Cyc can reach artificial general intelligence, or strong AI. Lenat insists that formal logic could be the key to creating an intelligent agent that can understand or learn any intellectual task that a human being can.

From a linguistic and terminological point of view, Temmerman (2000) argues that objectivists ignore the human capacity to understand and imagine, and see the reality has a rational structure which is independent of human understanding. This makes an objective system relatively blind to almost impossible to reason in unknown contexts (Winograd, 1990).

Goertzel (2007, p.15) believes that there are strong reasons to believe that classical formal logic is not suitable to play a central role in an AGI system: “It has no natural way to deal with uncertainty, or with the fact that different propositions may be based on different amounts of evidence. It leads to well-known and frustrating logical paradoxes. And it doesn’t seem to come along with any natural “control strategy” for navigating the combinatorial explosion of possible valid inferences.” This lack of dealing with uncertainty is important. Cyc does not use any sophisticated mechanism, such as Fuzzy logic, for uncertainty reasoning, and eschews numeric certainty factors. Cyc assumes each assertion is true by default and later decides on its value through additional meta-level assertions (Lenat,1995, p.33).

To Goertzel, Cyc is a traditional logic-based AI that attempts to map the mind rather than the brain. For Goertzel, the problem of Cyc is that it is built on the assumption that basically every aspect of mental process should be thought about as a kind of logical reasoning. Although Cyc knowledge base and its highly complex and specialized inference engine may potentially be useful eventually to a mature AGI system. it just scratches the surface of what is required for creating an intelligent mind.

The people working at Cycorp, Goertzel claims, are aware of this fundamental challenge. In 2006, a project called CognitiveCyc was led by former Cycorp employee Stephen Reed. As of today, this project is pursued by Ai-Blockchain with the mission of achieving artificial general intelligence with The Expert System knowledge base founded on the common sense OpenCyc ontology merged with the WordNet and Wiktionary English Lexicons.

In the knowledge representation community, however, opinions are divided. For Singh (2003), perhaps the greatest contribution of the Cyc project is not so much its knowledge base, but the vast ontology of terms and predicates that it uses to express its knowledge; Cyc likely has what is the most expressive such ontology in existence.

Singh envisions as Cyc begins to incorporate more such knowledge, it will come much closer to being a Society of Mind. Cyc is not yet a Society of Mind in the sense Minsky (1988) describes. Cyc focuses on the kinds of knowledge that might populate the A-brain of a Society of Mind, that is, it knows a great deal about the kinds of objects, events, and other entities that exist in the external world. However, it knows far less about cognitive processes themselves, knowledge about how to learn, reason, and reflect. Minsky (1993), in his seminal work *Society of Mind*, creates a dichotomy for two parts of the brain: A-Brain and B-Brain. While A-Brain uses external sensors to think about the external world, B-Brain has no external sensor and uses A-Brian’s description, thus not thinking so much about the outside world. This division of the mind into 'levels of reflection' is one of Minsky’s description of how human brain solves problems.

Difficulty of Evaluation: Is It Just a Beautiful Dream?

In his latest interview with Lex Fridman, answering to why people think that Cyc, despite being a beautiful dream, has been never materialized, Lenat (2021) said:

> *"We keep a very low profile. We don't attend many conferences, we don't give talks, we don't write papers, we don't play academic games at all. As a result, people only know about us because of a paper we wrote 10, 20 or 30 years ago or 37 years ago. They only know about us because of what someone else second hand or third hand said about us.*
>
> *[Lex Fridman]: So thank you for doing this podcast by the way!"*

For Lenat, there has never been a time he thought about quitting, and he still envisions that with priming the pump enough, Cyc can possess artificial general intelligence – which, in Lenat's view, will be the most important invention of humanity since the invention of language in the ancient times: "Not Internet; not mathematics; not computers; all the way back to the development of language."

As grand is the vision, the information is scarce. Indeed, Guha & Lenat (1990) was the last extensive description of Cyc – explaining mainly the philosophical foundation. There has been no follow-up of any kind in the last thirty years.

Ekbia (2008) is one of the few published AI researchers who had the opportunity to access the Cyc system during a workshop for potential customers in 2002. The participants manually fed the system with a short story, in a "long and tedious" encoding process. Users were required to create a microtheory for the whole events and individual parts, after which they had the chance to pose questions. Cyc's answers to the questions were unsatisfactory in a way that the user had to manually enter extra implicit information himself.

In a paper titled *Evaluating Cyc: Preliminary Notes*, Davis (2016) strongly calls for Cycorp to publish an overview of Cyc, and a substantial collection of interesting examples that it can actually handle.

> *On the whole, it is fair to say that the AI community regards CYC as a very elaborate failure. […] If the AI research community is unfairly and foolishly neglecting a valuable resource, then they should be made aware of that.*

Bleak Possibility of Open-Sourcing

OpenCyc, just a shadow of Cyc as Lenat (2021) claims, was released to only show AI researchers the magnificence of using an expressive representation (higher order logic) compared to triple store or knowledge graph type representation. The release was to entice researchers, programmers and enterprises to acquire a license to ResearchCyc or Full Cyc.

Fridman asks Lenat whether he believes that opening Cyc to the dormant giant open-source community is a leap of faith in creating a phase shift in the current landscape of AI – where now 99.9 percent of projects are machine learning, and Cyc is at the center of the symbolic AI. Lenat (2021), relatively unimpressed by open-source models, elaborates how the source codes of their inference engine is important: "Usually what we have done with people who approached us to collaborate on research is to say: we will make available to you the entire knowledge base and executable copies of all of the code, but only very, very limited source code access if you have some idea for how you might improve something or work with us on something so let me also."

Position in New Machine Learning Paradigm

In 1984, in a Stanford lecture Douglas Lenat on Eurisko, researchers would dismiss machine learning as “overly pragmatic” hacks (Nathan, 2016). This would change a decade later with the paradigmatic shift from symbolic AI machine learning.

Following a connectionist view, Minsky’s (1988), in *Society of Mind*, explains intellectual abilities using artificial neural networks as simplified models of the brain. Philosophers have become interested in connectionism because it promises to provide an alternative to the classical theory of the mind: the widely held view that the mind is something akin to a digital computer processing a symbolic language. Singh (2003) states that Minsky’s Society of Mind incorporated both symbolic and connectionist notions:

> “Since the Society of Mind was published there has been much work on developing 'connectionist' systems that recognize patterns, make inferences, and solve problems by distributing the solution process across societies of simple computational elements. One of Minsky's purposes in developing the Society of Mind theory was to establish a framework which naturally incorporated both symbolic and connectionist notions, yet despite the compatible presentation Minsky gives in the Society of Mind, the debate between symbolic and connectionist approaches continues to rage to this day.”

While Minsky (1993) sees this segregation of AI into machine learning and symbolic bastions as an apparent false dichotomy, Domingos criticizes Minsky as an ardent supporter of the Cyc project and his ideas as expressed in *the Society of Mind* Minsky.

Domingos (2015, p.35), a leading researcher in machine learning, lambasted Cyc as “the most notorious failure in the history of AI.” Domingo’s harsh comments unsurprisingly come from his stronghold of machine learning that sees the community of knowledge engineers as “machine learning’s perennial foe” (p.34).

To knowledge engineers, real knowledge cannot be learned automatically, but must be rather programmed into computer by human experts. Domingos (2015, p.36) resembles this approach to “Imagine if farmers had to engineer each cornstalk in turn, instead of sowing the seeds and letting them grow: we would all starve.”

It was during the 1980s, in the winter of machine learning, that knowledge engineers struck back with massive investments and promises of fulfilling artificial general intelligence. The knowledge engineering community, in Domingos’ view, are still unconvinced of the successes of machine learning, believe that “its limitations will soon become apparent, and the pendulum will swing back,” (p.35), similar to how the big promises of expert systems turned out to be disappointing, and a revival of machine learning occurred thanks to massive torrents of data. Domingos (p.35) writes:

> *Even if by some miracle we managed to finish coding up all the necessary pieces, our troubles would be just beginning. Over the years, a number of research groups have attempted to build complete intelligent agents by putting together algorithms for vision, speech recognition, language understanding, reasoning, planning, navigation, manipulation, and so on. Without a unifying framework, these attempts soon hit an insurmountable wall of complexity: too many moving*

parts, too many interactions, too many bugs for poor human software engineers to cope with.

As IBM's Watson is astonishing the world by showing some satisfactory degree of common sense in winning Jeopardy, its predecessor Cyc, the largest logic-based common sense project, is in eclipse (Havasi, 2014). In the 2010s, a new wave of AI research laboratories, this time under the paradigm of machine learning, have started ambitiously targeting. American OpenAI, founded in 2015, and British DeepMind, founded in 2010, are the most notable ones.

In June 2020, OpenAI released its GPT-3 that can generate coherent improvised texts, answer questions in natural language, and translate texts – all of this based on a language model trained on trillions of words from the Internet. Its GPT-4, five hundred times larger than its predecessor, aims to target Strong AI, or AGI.

Since the early 2010s, Cyc has embarked on using the tremendous benefits of machine learning. Domingos (2015, p.35) is sarcastic about Cyc's ability to harness information automatically: "Lenat has belatedly embraced populating Cyc by mining the web, not because Cyc can read, but because there's no other way."

Under the current paradigm, Lenat (2021) has redefined Cyc as the left brain, with machine learning being the right brain that takes a bunch of data and statistically infer. "But I would like to have my left brain: thinking more deeply and slowly; what Daniel Kahneman called 'thinking slowly'".

Cycorp latest report (2019) analogously likens ML-based software to right brain – which although excels at detecting correlations but fails to detect the underlying causation. Figure X shows how the report define Cycorp as the only company which can act as the left brain for providing causal reasoning.

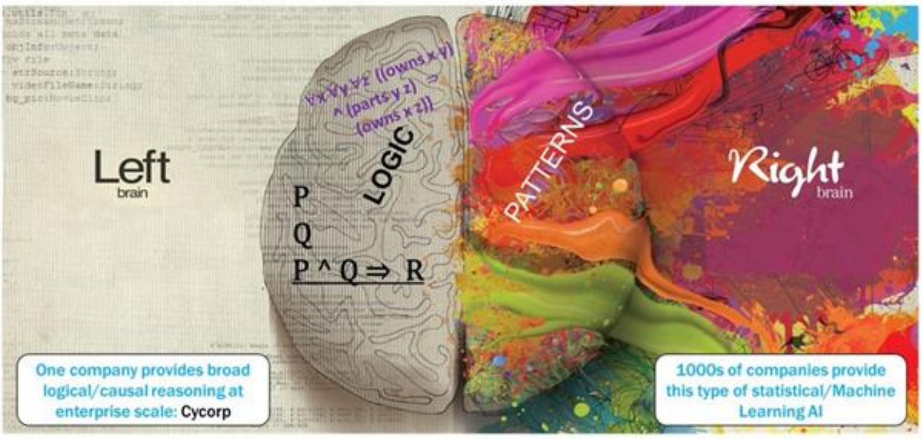

FIGURE 10 - CYC AS LEFT PART OF THE BRAIN

This proposed synergy of left and right hemispheres, Cycorp claims, has been fruitfully applied in several client applications, notably Cleveland Clinic and the National Library of Medicine. First, machine learning applications provide statistical correlations from GWASs (genome wide association studies), suggesting correlations between a patient's DNA mutations and early diagnosis of osteoporosis. Cyc then constructs plausible biochemical pathways to explain how this has led to osteoporosis.

In the latest report, Cyc has evaded expressing its long-held ambitions of targeting AGI, rather redefining its unique position as a commercial application within the current AI landscape.

Cyc Today

Cycorp

Cycorp, the company that started work on Cyc in the 1980s, is still in existence today. It is based in Austin, Texas, USA.

Doug Lenat is mentioned as Founder and Co-CEO of Cycorp and is today still the leader of Cycorps management team. Cycorp claims their technologies have had “$100M R&D dollars poured into it” over the past decades. (cyc, 2021)

According to Craft.co (craft.co, 2021), some 56 people work for Cycorp today. Cycorp seems to be actively hiring and is looking for ontology engineers, linguists, inference programmers, DevOps engineers etc. (cyc, 2021)

Cycorp targets large enterprise customers and offers Cyc integration in vertical markets for

- Hospital advisors
- Supply chain advisors
- Drilling Advisors

and they offer engineering services for a ‘Semantic Knowledge Source Integration’ (SKSI) where they will integrate Cyc’s knowledge base with the data available at a company allowing them to create an ‘Enterprise Knowledge Layer’ that adds ‘meaning’ to the companies' data (cyc, 2021)

Their website (https://cyc.com/) does not offer any explicit examples of products or clients; parties interested in their technologies can request a private demo.

Lucid

Lucid.ai, also based in Austin Texas, is another company building on the inheritance of Cyc.

Lucid.ai is led by Michael Stewart, who founded the company with Dough Lenat in 2008 (planettechnews.com, 2021). They claim to have a technical staff of more than 200 people of which more than half have PhDs. (lucid.ai, 2021)

Lucid.ai is the world’s largest and most complete general knowledge base and common-sense reasoning engine. When coupled with Big Data and Cloud-Level computing power, the result is an enterprise intelligence system unlike any other; one that brings human-like understanding and reasoning to computers, and unprecedented power, speed and scalability to human thought (lucid.ai, 2021).

Lucid.ai combines causal reasoning with machine learning and deep neural networks to create stronger AI (lucid.ai, 2021).

Today, they are focusing efforts on finance, healthcare-, energy- and retail-supply-chain- markets.

They claim success stories in (lucid.ai, 2021) :

- investment banking: a top financial institution was faced with increasing costs and resource needs to comply with increasing regulatory requirements. Lucid.ai developed a regulatory and compliance application and a comprehensive platform that identified employees possibly trading on insider information.

- Oil production: they developed an overlay to “well monitoring” software to enhance system engineers’ ability to drive well head efficiency, which can increase yields 10+% on all deep-water platforms”.
- Energy forecasting for smart grids; Lucid.ai is able to control, manage, analyze and predict behavior in an extensible manner on electric power grids.

Conclusion

Cyc is one of the largest attempts in history to create a system which has human-like common sense. In 2017, already more than 1000 men years went into the project, and now, 37 years later, the ambitious goal has not been reached yet. While Doug Lenat, pioneer of the project and CEO of Cycorp, still believes in reaching this goal, it is difficult to estimate how far they are from the final destination.

Critics of Cyc believe that capturing all the necessary rules around the concepts is not possible, even more because these are subjective in space, culture, time, and other contextual parameters (Matthias, AI and Society: 07d. CYC: Problems with CYC, 2019). Other criticisms build upon the previous that by definition, a symbolic system forms a symbolic representation of a situation and acts upon the rules and concepts defined within. Decontextualizing the concepts and rules is necessary (think microtheories) but the system is relatively blind to unknown contexts aka 'the blindness of representation' and as a consequence, it will be very hard if not impossible to reason in these contexts (Winograd, 1990). Or the even more drastic critics of Dreyfus (1993) arguing Cyc failure because human common sense cannot be put in some vast library.

By reducing the scope and contexts to (very) specific domains, the system would drastically reduce the chances on blind spots and could combine domain specific rules and concepts with existing knowledge in the Cyc KB to perform reasoning within the specific domain.

Over all these years Cyc has been sparse in sharing their projects and success stories with the world. Critics tend to attribute that to the absence of them. Doug Lenat's points out that their clients value working in strict confidentiality on projects with Cycorp. Fact is that Cycorp still exists and according to some (ex-)employees, they are mainly funded by projects for private customers (Unknown, 2019).

Some bigger tech companies used to invest in Cyc in the earlier days but all of them have shifted their AI focus towards machine learning and deep learning (Metz, 2016).

Doug Lenat's popular representation of Cyc being the left side of the brain and the current AI landscape the right side of the brain (mainly pattern matching according to him) did not pass the test of time and today a hybrid AI approach will do a lot more than just pattern matching jeopardizing the hopes of a growing Cyc adoption.

Nevertheless, the gigantic effort of writing an ontology and formalizing common sense formed a source of inspiration and knowledge for many other semantic projects and standards. Ramanathan V. Guha, co leader of Cyc during the early days, was later very active in the creation of several standards related to semantic web ontologies, among others: Really Simple Syndication and Resource Description Framework.

Is there a future for Cyc? In the AI community, It is difficult to find believers that the early goals of Cyc will ever be met.

This belief is strengthened by Cycorp and Lucid positioning the Cyc engine as an enterprise reasoning system quickly tailorable to reason about domain specific tasks. Their huge experience in understanding and formalizing the real world, combined with a multitude of tools for mapping and integration and doing some form of bidirectional NLP is a great trump card for certain projects. Depending on the specific nature of the domain, this might by a compelling offer but given the legacy and age of Cyc, more recent domain specific alternatives have a good chance of outperforming Cyc.

Figures

References

(2021, November). Retrieved from cyc: https::/cyc.com/

Bosselut, A., Rashkin, H., Sap, M., Malaviya, C., Celikyilmaz, A., & Choi, Y. (2019). Comet: Commonsense transformers for automatic knowledge graph construction. *arXiv preprint arXiv:1906.05317*.

Bradford, A. (2017, 7 25). *Deductive Reasoning vs. Inductive Reasoning.* Retrieved from https://www.livescience.com: https://www.livescience.com/21569-deduction-vs-induction.html

Copeland, J. (1993). *Artificial intelligence: A philosophical introduction*. John Wiley & Sons.

craft.co. (2021, November). *Cycorp.* Retrieved from craft.io: https://craft.co/cycorp

Crevier, D. (1993). *AI: the tumultuous history of the search for artificial intelligence*. Basic Books, Inc..

Curtis, J., Matthews, G., & Baxter, D. (2005, August). On the effective use of Cyc in a question answering system. In *IJCAI Workshop on Knowledge and Reasoning for Answering Questions* (pp. 61-70).

Cycorp. (2002). *Cyc 101 Tutorial.* Retrieved from http://www.opencyc.org: https://web.archive.org/web/20121025010325/http://www.opencyc.org/doc/tut/?expand_all=1

Cycorp. (2019). *Technology overview.* Retrieved from https://www.cyc.com: https://www.cyc.com/wp-content/uploads/2019/09/Cyc-Technology-Overview.pdf

Dascal, M. (1989). Artificial intelligence and philosophy: The knowledge of representation. *Systems Research*, , 6(1), 39-52.

Dascal, M. (1989). Artificial intelligence and philosophy: The knowledge of representation. *Systems Research*, *6*(1), 39-52.

Davis, E (July 9, 2016). *Evaluating Cyc: Perliminary Notes*.

Derthick, M. (1990, 2). *An Epistemological Level Interface for CYC.* Retrieved from http://www.cs.cmu.edu: http://www.cs.cmu.edu/~mad/CYC/act-cyc-084-90.pdf

Domingos, P. (2015). *The master algorithm: How the quest for the ultimate learning machine will remake our world*. Basic Books.

Dreyfus, H. L. (1972). *What computers can't do: A critique of artificial reason*. New York: Harper & Row.

Ekbia, H. R. (2008). *Artificial dreams: The quest for non-biological intelligence* (Vol. 200, No. 8). Cambridge: Cambridge University Press.

Feigenbaum, E. A., & McCorduck, P. (1983). *The fifth generation*. Addison-Wesley Pub.

Garvey, C. (2019). Artificial intelligence and Japan's fifth generation: the information society, neoliberalism, and alternative modernities. *Pacific Historical Review*, *88*(4), 619-658.

Guha, R. V., & Lenat, D. B. (1994). Enabling agents to work together. *Communications of the ACM*, *37*(7), 126-142.

Hankins, P. (2005, May 12). *Cycosaurus breaks free*. Cycosaurus breaks free. Retrieved November 13, 2021, from http://www.consciousentities.com/cyc.htm.

Hilts, P. J. (1983, April 7). *U.S. urged to aid development of new generation of computers*. The Washington Post. Retrieved November 11, 2021, from https://www.washingtonpost.com/archive/politics/1983/04/07/us-urged-to-aid-development-of-new-generation-of-computers/63cb65b5-5f5e-4403-a95a-e7b45df0bf99/.

How To Play. (2008, 6 19). Retrieved from game.cyc.com: https://web.archive.org/web/20080619202637/http:/game.cyc.com/helpfiles/HowToPlay.html

John McCarthy. "Programs with common sense". *Symposium on Mechanization of Thought Processes*. National Physical Laboratory. Teddington, England. 1958.

Jon Curtis, G. M. (2005). On the Effective Use of Cyc in a Question Answering System.

Katharina Siorpaes1 and Martin Hepp1, 2. (2008). *OntoGame: Weaving the Semantic Web by Online Games .* Munich.

Knight, W. (2016, 7). *An AI with 30 Years' Worth of Knowledge Finally Goes to Work.* Retrieved from https://www.technologyreview.com: https://www.technologyreview.com/2016/03/14/108873/an-ai-with-30-years-worth-of-knowledge-finally-goes-to-work/

Lefkowitz, L., Curtis, J., & Witbrock, M. (2007, 9). *Accessible Research Cyc.* Retrieved from https://www.researchgate.net/publication/235117637_Accessible_Research_Cyc

Lenat, D. B. (1995). CYC: A large-scale investment in knowledge infrastructure. *Communications of the ACM*, *38*(11), 33-38.

Lenat, D., & Deaton, C. (2008). *Terrorism Knowledge Base (TKB)*. CYCORP AUSTIN TX.

Lenat, D., Witbrock, M., Baxter, D., Blackstone, E., Deaton, C., Schneider, D., Scott, J., & Shepard, B. (2010). Harnessing Cyc to answer clinical researchers' ad hoc queries. *AI Magazine*, *31*(3), 13-32.

Lenat, D. B., & Durlach, P. J. (2014). Reinforcing math knowledge by immersing students in a simulated learning-by-teaching experience. *International Journal of Artificial Intelligence in Education*, *24*(3), 216-250.

Lenat, D., & Feigenbaum, E. (1991). On the thresholds of knowledge. Foundations of Artificial Intelligence, MIT Press, Cambridge, MA, 185-250.

lucid.ai. (2021, November). Retrieved from lucid.ai: https://lucid.ai/

Marshall, E. (2014). *The spiritual automaton: Spinoza's science of the mind*. OUP Oxford.

Masters, J., & Güngörd, Z. (2003). *Semantic Knowledge Source Integration: A Progress Report.* Retrieved from https://ieeexplore.ieee.org: https://ieeexplore.ieee.org/document/1245101

Matthias, A. (2019, 10 10). *AI and Society: 07a. CYC: What is common sense?* Retrieved from Youtube: https://www.youtube.com/watch?v=uWI2e7gUrWw

Matthias, A. (2019, 10 10).*AI and Society: 07c. How does CYC work?* Retrieved from Youtube: https://www.youtube.com/watch?v=gD2CNI_xLNQ&list=PLbI7KkMhkaPtmtp39rQwR3cPeZvFt-jC9&index=17

Matthias, A. (2019, 10 11). *AI and Society: 07d. CYC: Problems with CYC.* Retrieved from Youtube: https://www.youtube.com/watch?v=BzZZsCtmwfE

Matthias, A. (2019, 10 11). *AI and Society: 07e. Does CYC understand?* Retrieved from Youtube: https://www.youtube.com/watch?v=LURs18ipFxs

McCarthy, J. (1993). Notes on formalizing context.

Metz, C. (2016, 3 24). *One Genius' Lonely Crusade to Teach a Computer Common Sense.* Retrieved from https://www.wired.com/: https://www.wired.com/2016/03/doug-lenat-artificial-intelligence-common-sense-engine/

Minsky, M. (1988). *Society of mind*. Simon and Schuster.

Minsky, M. L. (1991). Logical versus analogical or symbolic versus connectionist or neat versus scruffy. *AI magazine*, *12*(2), 34-34.

Monberg, J. (2006). Conceptions of the Social that Stand behind Artificial Intelligence Decision Making. *Journal of Technology Studies*, 32(1), 15-22.

Monberg, J. (2006). Conceptions of the Social that Stand behind Artificial Intelligence Decision Making. *Journal of Technology Studies*, *32*(1), 15-22.

MYSENTIENT IRELAND (R&D) LIMITED INACTIVE. (n.d.). Retrieved from GlobalDatabase: https://ie.globaldatabase.com/company/mysentient-ireland-r-d-limited

MYSENTIENT Trademark. (n.d.). Retrieved from https://alter.com/trademarks/mysentient-78351664

Nathan, P. (2016, August 6). *Beyond the AI winter*. Medium. Retrieved November 11, 2021, from https://medium.com/derwen/beyond-the-ai-winter-941c0a66b4f5.

Neumann, D. (2020). Artificial Intelligence of the Human Mind: Finding Common Ground between Intelligent Machines and "mechanistic" Minds. In *Artificial Intelligence* (pp. 37-48). mentis.

Newel, A., & Simon, H. A. (1976). Computer science as empirical inquiry: Symbols and search. *Communications of the ACM*, *19*(3), 113-126.

Panton, K., Matuszek, C., Lenat, D., Schneider, D., Witbrock, M., Siegel, N., & Shepard, B. (2006, 8). *Common Sense Reasoning – From Cyc to Intelligent Assistant.* Retrieved from https://www.researchgate.net/: https://www.researchgate.net/publication/226813714_Common_Sense_Reasoning_-_From_Cyc_to_Intelligent_Assistant

planettechnews.com. (2021, November). *PlanetTech interviews Michael Stewart, founder, chairman and CEO of Lucid AI*. Retrieved from planettechnews.com: https://www.planettechnews.com/planettech-interviews-michael-stewart-founder-chairman-and-ceo-of-lucid-ai/

Poole, D. L., & Mackworth, A. K. (2010). *Artificial Intelligence: foundations of computational agents*. Cambridge University Press.

Ramachandran, D., Reagan, P., & Goolsbey, K. (2005). *First-Orderized ResearchCyc : Expressivity and Efficiency in a Common-Sense Ontology.* Retrieved from https://www.aaai.org/Papers/Workshops/2005/WS-05-01/WS05-01-006.pdf

Reed, S., & Lenat, D. (2002). *Mapping Ontologies into Cyc.* Retrieved from Researchgate: https://www.researchgate.net/publication/2917539_Mapping_Ontologies_into_Cyc

Searle, J. R. (1980). Minds, brains, and programs. *Behavioral and Brain Sciences, 3*(3), 417–424. https://doi.org/10.1017/S0140525X00005756

Singh, P. (2003). Examining the society of mind. *Computing and Informatics*, *22*(6), 521-543.

Smith, B. C. (1991). The owl and the electric enCyclopedia. Artificial Intelligence. 47(1-3), 251-288.

Smith, B. C. (1991). The owl and the electric encyclopedia. *Artificial Intelligence*, *47*(1-3), 251-288.

Smith, S., & Kandel, A. (2018). *Verification and validation of rule-based expert systems*. CRC Press.

Somma, R. (2008, Aptil). *T.L. A.I.* Retrieved from ideonexus.com: https://ideonexus.com/2008/04/23/rl-ai/

Sundararajan, S. (2021, 1 10). *The Story of Machine Proofs - Part II.* Retrieved from https://hackernoon.com: https://hackernoon.com/the-story-of-machine-proofs-part-ii-ojw34y4

Sutcliffe, G. (2020). *The ATP Process.* Retrieved from http://www.tptp.org: http://www.tptp.org/Seminars/TPTPWorldTutorial/ATPProcess.html

Temmerman, R. (2000). *Towards new ways of terminology description: The sociocognitive-approach* (Vol. 3). John Benjamins Publishing.

Toews, R. (2021, June 1). *What artificial intelligence still can't do*. Forbes. Retrieved November 11, 2021, from https://www.forbes.com/sites/robtoews/2021/06/01/what-artificial-intelligence-still-cant-do/.

Unknown. (2019, 12). Retrieved from Hackernews: https://news.ycombinator.com/item?id=21781597

Wikipedia. (2021, 10). *First-order logic.* Retrieved from Wikipedia: https://en.wikipedia.org/wiki/First-order_logic

Winograd, T. (1990). *Thinking machines.* Retrieved from http://hci.stanford.edu: http://hci.stanford.edu/~winograd/papers/thinking-machines.html