Fuller Theological Seminary

The Limits of formal Languages

Paper for the Class

PH852: Methods in Philosophy

Fall 1996

by

Christoph Lameter

December 6, 1996

Table of Contents:

Introduction

Humanity was always marked by curiosity and a desire to know. Knowledge was written down and communicated using natural languages but early on there was the recognition that language is a difficult medium to use for communication of concepts and rules due to fluidity and ambiguity in meaning inherent in natural languages. The ancient Greeks already made attempts to make language and the argumentation for these concepts unambiguous and thus more formal. Euclid's *Geometry* is a classic example of such an ancient piece. Many of the characteristics of current formal systems are foreshadowed in Euclid's work. Due to its compactness and clarity Euclid's work was the foundational text for Geometry for almost 2000 years.

The enlightenment with its drive for certain foundations and clarity in reasoning took such works as a *Vorlage* for how good reasoning and good science should proceed. An unprecedented development in logic and mathematics was the result culminating in Frege's and Russell's attempt to develop and use formal logic to clarify natural languages. With the development came the assumption that the formalized form of language expresses concepts belonging to the real reality. Natural language statements were prone to misunderstandings and inaccuracies. Russell and Whitehead later pioneered the basis for modern mathematics and logic in their epochal work *Principia Mathematica*.

In the 20th century most of science is predominantly communicated in some form of formal system. For example one of the greatest discoveries of Einstein was the equivalence of Mass and Energy made famous by the following formula: $E = m c^2$

Einstein's formula of the convertibility of mass to energy enabled the development of the atomic bomb in the second world war changing the course of history. The formula is a highly condensed form of Einstein's thinking. It is rather difficult to express the idea in natural language avoiding all references to terms developed through formal systems since *Mass* and *Energy* are not everyday categories we are used to handle.

Formal systems became even more important with the advent of the computer revolution. Languages based on formal systems are used to control computers, write programs and even major parts of the hardware of computers are designed using formal languages to describe the characteristics of the hardware to be build by engineers or by some advanced machine able to automatize the hardware design.

Formal languages have been vastly successful. It is doubtful whether any of the scientific accomplishments like for example landing humans on the moon would have been possible without formal systems. Quantum Mechanics would certainly be impossible and is largely done on the basis of formal systems. Analogies used frequently in the past break down and it seems that the way to talk intelligibly about Quantum Mechanics is in formal systems describing the characteristics of the elements in question.

Due to these successes formal systems have been idealized as the (if not *the only*) form of reliable knowledge. Other forms of communication in narrative, analogy and testimony by witnesses are seen as inferior. Religious forms of communication have been predominantly using non-formal systems instead of a systematic formalism. Based on the assumption that scientific formal systems are the only reality, religious statements have frequently been banned into the realm of the expression of emotions, into the unreal. The only real reality accepted is the scientific one which is commonly expressed in formal systems.

These assumptions came with the positivists like Russell due to their enormous contribution to the development of logic and mathematics. I would content though that some of their ideas about the nature of reality and about the power of their formal systems are unrealistic and too optimistic.

The question about the limits of formal systems is important to clarify the claims made on their behalf and to see if they fulfill the hopes invested in them. Thus this paper is investigating the limits of formal systems.

In this context also another important question is rising up: If formal languages are used to express our knowledge and if computers can be programmed using formalized systems then how far can computers be developed? Do formal systems set a limit on what Computers can do? Can computers provide Artificial Intelligence or Consciousness ?

In the following paper I am trying to simplify matters as much as possible and avoid the use of formalism in order to increase the comprehensibility. I will try to translate formal expressions into natural language whenever possible. Please do not expect a comprehensive coverage of the subject. I will try to only touch on those issues relevant to the questions under consideration.

What are formal languages?

Basic Structure

Formal systems generally consist of a set of **symbols** and rules for the formation of valid sentences (combination of symbols) commonly called a **grammar** or **formation rules**. The grammar determines which combination of symbols form a valid sentence (a *well formed sentence*) of the formal system (Audi 1995, 275).

It is customary to refer to formal systems also as **languages** due to the similarities with natural languages. There are important distinctions to natural languages though due to the rigorous structure of formal languages. The limitations incurred in formal systems through those rigorous structures is in contrast to the universal scope of natural languages. Formal systems are therefore also referred to as **artificial languages** or **formal languages**.

The basic structure of a formal system is usually not specified exhaustively except in teaching situations. There are customs on how formal languages are formed in different contexts that allow for the omission of the specification of the basic structure. Handling formal systems is often a daily routine and is done without much thinking. Only necessary elements to highlight a point are mentioned which might be confusing to a newcomer in the field.

Syntax and Semantics

Formal languages do not refer to outside of their system by themselves. They are after all self-contained little logical universes. Connections to outside entities have to be carefully established to give them meaning and significance.

I will use the term *syntax* for the formal language by itself drawing on the use of syntax from computer science. A problem on the syntactical level will refer to a problem with the formation of the formal system in itself not in regard to any function the formal system performs or is associated with.

To make formal systems useful an **interpretation** has to be assigned. Customarily an interpretation is accomplished by assigning meanings to the symbols of the language and truth conditions to the sentences of those systems. (Audi 1995, 275) And assignment of an interpretation does not necessarily mean that the interpretation is correct. The formal system might be correct in itself but the objects it is supposed to describe might work in a different way.

Formal systems have at least one interpretation given by the symbolic language used to describe it. It is a formal interpretation called the *standard-interpretation*. Multiple interpretations of formal systems are possible through the assignment of alternate objects in different interpretations.

The elements of the formal system are assigned a reference to some object. They can express significant things about reality and thus get semantical meaning. Thus semantics refers to the mapping of that formal system to perform something useful. Assignment of a meaning to formal systems necessarily involves using a different language to assign meaning to the formal system which is not possible with the formal language itself. That language is customarily called a **meta-language**. The formal language itself is sometimes referred to as the **object-language** in order to clearly distinguish it from the meta-language.

Natural languages are distinct from formal languages in that they provide meaning already through the symbols of the language. It is customary to use natural language at some point as the meta-language for a formal system.

A formal system might designate certain sentences to be **axioms** or add **inference rules**. The formal system equipped with such features is called a **logical system**.

Theoretical Limitations

Please note that all the issues discussed in this section are on the *syntactical level*. They are intrinsic to formal systems and are not depending on the reference or interpretation given to a formal system. Semantic issues are discussed later.

Russell's Paradox or the Vicious Circle Principle

Frege developed the mathematical foundations for set theory in his work *Grundgesetze.* Russell found that Frege's system could lead to sentences completely legal in set theory but which were not decidable within the Frege's framework. Decidable means here that set theory was unable to give a result of true or false to a question asked within the formal framework. Yet decidability was at the heart of the design of such a system avoiding the perceived ambiguities of natural languages.

Frege was devastated since Russell had thus shown an inconsistency in Frege's logical system and Russell himself was also very disturbed about the problems with the logical systems he was trying to construct. (Chihara 1973 ,1)

Russell's Paradox:

W = set of all sets that do not contain themselves.

Does W contain itself?

The question stated is not decidable. If W would contain itself then W is not a member of W. Since W is not a member of W it follows that W does not contain itself. The assumption we started with has been denied in the result we have gotten.

On the other hand if W does not contain itself then W would be a member of W. Consequently W would contain itself.

The question if W contains itself cannot be answered with either yes or no! It would not be a problem if the answer would be no. The point is that formal set theory *cannot give any answer* at all but becomes contradictory when trying to answer the question. The logical scheme of set theory seems to disintegrate and it becomes useless.

In essence Russell's Paradox is a variation (a formal version instead of a version in natural language) of the old Greek Paradox where a Cretan says “All Cretans are liars”. If the Cretan is lying then the sentence is a lie and thus Cretans are not lying. If the Cretan is telling the truth then we have one Cretan here who is not a liar and thus his proposition is not true.

The conclusion to be drawn from Russell's Paradox is that formal languages which are constructed in order to allow decisions cannot (in some situations) decide on a syntactical level if a sentence belongs to their own formal scheme of things. They do not perform the very purpose formal systems were designed to perform. Russell himself warns in a later writing: “If you proceed carelessly with formal logic, you can easily get into contradictions” (Russell 1956, 259). Thus in some situations formal systems suffer from the same indecidability as natural languages.

Russell later figured out why the paradoxes result. Paradoxes like Russell’s result out of the reference to a totality. A recursive (a self-referential) reference is the problem and Russell was able to resolve the problem through the *Vicious-Circle Principle* forbidding dangerous self-referential or recursive references like the above. Russell later developed a *Type Theory* (more about type theory later) to restrict formal systems in order to make dangerous recursive references impossible (Hofstadter 1979 , 24). The resulting formal systems can no longer produce undecidable sentences. But some parts of classical mathematics need recursive references violating Russell’s Vicious-Circle Principle.

Gödel’s Incompleteness Theorems

Kurt Gödel wrote a paper in 1931 titled “Über formal unentscheidbare Sätze” in which he reflected on the *Principia Mathematica* and developed the famous “incompleteness Theorems”. Gödel’s Theorems show for formal systems of some complexity (those formal systems must be sufficiently complex to express arithmetic within those systems) that there exists at least one sentence in the formal system that cannot be proven within the system itself (Audi 1995, 299, Chihara 1973, Chapter 2). The two Theorems result in the following insights:

Theorem 1: Proves the existence of a *valid* sentence in formal language but that same sentence cannot be proven to be valid by the formal language itself.

Theorem 2: No sentence can exist within a formal system that proves the consistency of the same formal system as a whole.

These Theorems put significant limits on formal methods and the formal systems possible. A formal language can never be self-contained because it cannot prove its own consistency. A formal system of sufficient complex nature can also never completely determine which sentences belong to the language itself.

A strange phenomenon develops: If additional rules are added to a formal system - which was found to be lacking the provability in some aspects - to make those indecidable sentences decidable then additional sentences will be showing up as unprovable. If on the other hand the system is restricted in order to be completely decidable then it will not be complete. *The incompleteness of a formal system is one of its core characteristics!* (Hofstadter 1979, 503)

The richness of rules in such a system essentially lead to its downfall. There seems to be a critical mass. Below the critical point the formal system is not able to prove its own sentences but above that critical point the system gains self-referential characteristics and develops chronic incompleteness. (Hofstadter 1979, 503). Either the language has facilities below the critical point and thus is so poor in what it can express that no self-referential expressions are possible (and the notion of truth is not definable within that context) or the language is powerful enough to talk about itself incurring self-referentialism and thus paradoxes and undecidable sentences (Scherb 1992, 23). Russell's Paradox might just be a sub-problem of the limitations unearthed by Gödel's Theorems.

Natural languages can be self-referential and thus are ambiguous if seen from a formal language perspective. Words get their meaning through use and associate freely within common knowledge. Consequently sentences in isolation are not definable if true or false. But meaning in a natural language can be expressed with a natural language in contrast to formal languages which customarily use a metalanguage. Unambiguity of language and meaning is not accomplished by restricting the context but through a community and a tradition using that language. The semantic reference is defined by the environment. Life itself is the basis for communication (Scherb 1992, 40-55)

Russell's solution: Type Theory or Metalanguages

In response to the Paradoxes Bertrand Russell initially developed the *Vicious Circle Principle* but later proposed the so-called *type-theory*. From Gödel's Theorems it follows that statements can be made about any language that are not expressible in the language itself. Some statements about a language may be made in the language in question but to avoid contradictions through undecidability or paradoxes a **metalanguage** is necessary to communicate about the characteristics of a given language. Russell (and later Tarski) proposes a hierarchy of languages to avoid self-references to a totality within the language itself. (Russell 1956, 371)

Each proposition has to be located at the proper level of the hierarchy and propositions at one level cannot refer to abstract notions on the same level or higher levels. Thus self-referential elements to totalities are contained. Similarly the truth or integrity of one language cannot be proven on the same level as the language. A metalanguage is necessary to prove the integrity of a language avoiding the impact of Gödel's Theorems.

It is needless to say that the hierarchization of formal languages in the Russellian way leads to a significant reduction of what these languages can express and necessitated the reformulation of some basic mathematical formal proofs to fit within Russell's type theory.

Nevertheless there were still paradoxes possible even within Russell's initially proposed type-theory. Russell developed another restriction for formal systems called the *ramified type theory* by a complex system of restrictions on the grammar of formal systems and improved the *Vicious Circle Principle* to account for these paradoxes as well.

It turned out that the ramified type theory did not allow adequate foundations for classical mathematics anymore leading to a modification of the ramified type theory to accommodate for necessary self-referential elements in basic mathematics. (Chihara 1973, Chapter 1; Audi 1995, 818)

Implications

Formal systems have problems on a theoretical level. They do not provide a perfect framework but can only incompletely develop a system. They are useful nevertheless since they are the best method we have been able to come up so far but it is unreasonable to believe that they are perfect in themselves. Formal systems should be taken as helpful devices for thinking keeping in mind Russell's warning about logical systems. They can be contradictory and are of necessity incomplete. Russell's type theory is complex and not easily applied to formal systems and consequently paradoxes cannot be easily avoided.

Russell's hierarchy of formal systems also means that the validity of a formal systems can never be proven within that formal system itself. A metalanguage will be used to talk about that language (Scherb 1992, 21). But is the *metalanguage* used valid and correct? The metalanguage needs another *meta-metalanguage* to be discussed. That meta-meta-language needs *another meta-meta-meta-language* and so on. The burden of the ultimate proof is shifted higher and higher. Never can any formal system be proven to be correct despite Russell's hierarchy. Type theory only removes paradoxes by restricting the ability of expression by rules of the formal system. Ultimately all proofs have to resort to natural language with all its ambiguities in order to avoid the infinite regress.

It is interesting to see the decadelong attempts of physicists to come up with the Grand Unified Theory that would explain everything in the light of the theoretical limitations of formal languages. Einstein was looking for such a theory and recently Stephen Hawking in his inaugural address at

Cambridge (Hawking 1993, 49-) and in his book *A short History of Time* is still hoping for the development of a such a Grand Unified Theory (GUT).

It might be bold by me to draw such inferences from Gödel's Theorems, its rather speculative since I am not sure that the Theorems are applicable (GUTs might not be covered by the Theorems if they only try to unify major laws of physics), but the application of the Theorems on GUTs would lead to the following interesting conclusions:

⇒ GUTs cannot exists since they attempt to be complete. Formal systems cannot be complete. Theories use formal systems. A GUT cannot be a Theory.

⇒ If a GUT would exists then we would not have the theoretical means to describe it since formal systems are inadequate to describe a complete entity.

⇒ If we try to describe a GUT then the theory will turn out to be contradictory and not consistent.

All attempts to develop GUTs have so far only been partially successful and turned out to be contradictory in one aspect or another. Certainly better theories than we have today can be developed and will be developed in the future but I think the hope for a Grand Unified Theory of *everything* might better be given up in light of the limitations we have in even giving expression to our knowledge.

Practical Issues

Semantics

The prior chapter was concerned with the internal issues within formal systems on a syntactical level. The question still remains how the formal systems are related to reality. How do these formal systems relate to external objects?

Semantics here are used not to express the intrinsic nature of logical systems but the use of a formal system to express something about science or reality.

In my research I have found lots of references to semantics of logic but rarely about how the formal systems relate or express objects or ideas. The notion of a truth function does not solve the issue since there is essentially just another formal system set up to verify the first one. The relationship of a formal system to what it is attempting to express is not formalizable. Most of the time I have seen the reference of formal systems being implicitly assumed or simply connected to terms defined by earlier usage of formalisms. Symbols are chosen to represent entities known to the audience the formal system is communicated to.

The same issues of reference as with natural languages exist in formal systems. Some associations have to be established to external systems to make a formal system useful. Due to the nature of formal systems the association has certain freedoms. Within a formal system new terms can be defined that do not have strong associations with external objects. Formal systems can build a kind of Quinean web having concepts on the outside referencing elements from the realm of our natural understanding and prior concepts and ideas already defined by conventions. Interior concepts are more independent and can be defined within certain boundaries.

Relation to raw data

In my undergraduate studies we had to perform experiments in physics. I would like to roughly summarize the method here to show how formal systems are intended to be resulting from data in a laboratory situation (We were of course not doing original research but only given to survey what others had already done before us):

I. **Assignment of a Research Project**. The project starts with an assignment given by the teacher to perform prescribed experiments and take certain measurements.

II. **Data Collection**. The experiment is repeated a couple of times. The parameters of the experiment are varied. Results are recorded.

III. **Speculation on relationship within the data collected**. The data is depicted as graphs or arranged in other fitting ways to find some correlation between the varied parameters and the results. A theory is developed on how the results relate to the parameters that were varied in the experiment.

IV. **Calculation of the error rate**. Statistics are consulted to figure out how much the results vary from the assumed relationship of the parameters. The speculation can be rejected if the variance is too high and the experiment might have to be redone or the speculation about the correlation between the data in the experiment might be reconsidered.

V. **Explanation**. The correlation between the data in the experiment is placed within the larger framework of established concepts and knowledge. The knowledge is extended appropriately if necessary. Already known formal systems are expanded or supplemented by the explanation.

Basically the above procedure means that the semantic meaning of formal systems (the interpretation) is confirmed by trial and error speculating on already present knowledge and testing if the assumed speculation is to be verified in the experiment. All concepts expressed in the interpretation of formal systems are the result of such procedures and as such are the result of a research tradition and do not have their source only in raw data. Those concepts are expressed in formal systems and consequently the formal systems are an expression of the research tradition. Research would be impossible without knowing how to do the experiments and how to measure things as Kuhn has shown before.

A Tradition of Knowledge

Formal systems are therefore formulated within a tradition of a body of formalized knowledge and thus form a tradition on their own. There is an established vocabulary in the tradition that common formal systems reflect on. Formal systems are used in the tradition to communicate and preserve the tradition. In contrast to normal life-style “traditions” that govern our way of life the tradition of formal systems is not determining or way of life but is a certain formalized way of expressing and communicating about knowledge which is much more effective than the natural languages since “precise” definitions have been developed in that tradition. Formal systems can in my opinion be regarded as another language (or a partial language) used for communication between specialists or between humans and machines. Formal systems are fit for the communication with machines due to their restricted context and association of the vocabulary within the system.

It is an illusion to believe formal languages and formalized knowledge to be a separate enlightenment style foundational building. Formal languages change over time and the assumption that change is always progress is certainly a mistake. Paradigms compete in science as Kuhn has shown. The predominant language of science is using formal language to describe reality. Formal languages and their interpretation are affected by the paradigm shifts experienced by science. One such example is the partial reworking of mathematical foundations as a result of Russell’s struggle with self-referentialism in formal systems.

Computing and Formal Systems

A Computer = a formal system?

Formal languages have been used in multiple ways in Computer Science and Computer technology. Formal systems in Mathematics were establishing the foundational concepts that made the first computers possible. Mathematicians in the early part of this century discussed a number of issues bearing directly on the first machines constructed around the period of the second world war. Formal systems are essential for specifying how to instruct, build and interact with a computer.

When I reviewed literature in philosophy on the issue of computing I was astonished of the remnants from that early mathematical period. There is a classic claim that computers are only formal systems and consequently limited by Gödel's Theorems. Hofstadter still has the view that computers are formal systems on the hardware level but claims that higher levels might not fit the character of a formal system anymore (Hofstadter 1979, 615). Scherb sees computing limited by the need to use formal languages (Scherb 1992, 66-80).

From my long years of dealing with computers I must say that none of this is an accurate reflection of current conditions in computer technology. Elaine Rich's book rightfully does not mention any theoretical limitations anymore but simply suggests a variety of ways to deal with potential problems.

Computer programs are written in a huge variety of computer languages. Formal languages are used to describe these languages. What the programs express is not limited by the formal systems. Only the description how the problem is to be solved is limited by formal languages.

Formal systems are combined in a variety of ways in order to gain functionality. For example the common programming language has customarily at least two basic formal systems nested together. One is governing the grammar describing how symbols can be arranged in such a language and another one on the so-called "lexical" level determining how the special kinds of variant symbols itself are to be formed. Thus the vocabulary of the first formal language is vastly extended by having a vocabulary of which a part has again its own formal language.

Current computer technology violates formal languages in many ways since all machines contain multiple processors. Each computer device invariably nowadays has its own microprocessor and the main Central Processing Unit is functioning more like an agent responding to communications from peripheral microprocessors arranging the flow of information and supervising the processing of the subunits.

I have no idea how a keyboard or a network interface card could fit within a formal language. Formal languages can be applied in order to explain certain aspects of the system but there will be no comprehensive formal system possible describing the whole machine.

Gödel's Theorem might apply to a single formal system but no to their combined power. If one language does not perform to the satisfaction of a programmer in a certain area (for example the language gets ambiguous or difficult to control) then the programmer will feel free to move on to use another computer language for that problem. The limitations of Gödel's Theorem are thus avoided.

Current computers are not only programmed in one computer language but there is a plethora of languages available and customarily one chooses the language according to the need. Programs written in different languages interact with each other complicating the situation even more. Computers processing software written in those languages are also more like busy messenger systems than formal automaton in the same way as the microprocessors on the hardware level Formal languages are used in computer technology whenever they are useful. But technology is not a theoretic field like mathematics or logic. Computer technicians always have felt free to break any logical rules if it could give them an edge over the competition. Pragmatically the limits are accepted. If there is a problem then creative ways have to be found around. If the result of specifying a problem results in an ambiguity then one or the other way will be found to make some decision or postpone the decision until ambiguities (or "Paradoxes") can be resolved when more information is available.

Determinism

Another surprising view is that computers are seen as deterministic automatons in philosophical literature. The parallel processing features of current computers make this claim impossible to uphold. All computers that I have ever known have some timing dependent elements in them requiring the non-deterministic interruption of processing in order to accomplish urgent tasks or tasks completely governed by undeterministic events.

Through the linking of millions of machines on the Internet another severe blow to determinism is made. A machine can never know when another machine tries to establish contact and exchange information. Machines have to react to indeterminate events. Determinism is a basic characteristic of computers in order to allow humans control and accomplish reliable operation but its readily abandoned whenever necessary. Computers even include devices to allow not only for the simulation but for real randomness. Randomness or indeterminacy has become an essential property for network operations in order to avoid all systems sending information at the same time and thus overloading the network. Timings need to be truly random to effectively use the network medium.

The Vicious Circle Principle

In the 1960s and 1970s some early computer languages did not support recursion (probably because of the concern for the issues arising out of self-referentialism in formal language theory) notably FORTRAN, COBOL and BASIC. Programmers were forced to find ways to avoid recursion in order to implement their algorithms leading to unnecessarily complex programs.

When languages like C and PASCAL became widely available supporting recursion (self-referential structures) it became evident that self-referentialism was a concept too powerful to omit from computer languages. The languages which did not support recursion have now all been extended to support recursion in one way or another. The theoretical prejudice coming from logic had to be abandoned.

The programmer was left to deal with the potential problems resulting from self-referential programming since it was impossible to implement translators for computer languages distinguishing the correct use of self-referentialism from Vicious Circle Style self-referentialism.

Programs will simply fail to perform their function if coded incorrectly. Commonly the programs will just stop responding or they will exhaust one or another form of computing resource on a computer. The event of such things happening is sufficiently rare that it is of no major concern for common computer operation and they are usually indistinguishable from regular bugs that the programmer has to deal with anyway.

The solutions found in Computer Science in their application of formal systems is thus a pragmatic one. The Vicious Circle problem is just a minor side issue not bearing on the central utility of languages. The solution is analogous to natural languages which also have no big issue with ambiguity.

Use of Metalanguages

The idea of evaluating one formal system by another has proven to be another theoretical ideal. In the practice of Computer Science formal meta-language systems are so complicated that a proof of correctness is not possible (disregarding the issue of the correctness of the metalanguage). Proofs of certain restricted aspects of formal systems have been successful and are commonly done by translators for high level programming languages.

Mathematics also has largely avoided using metalanguages to prove the correctness of formal systems. It is common to use natural languages to evaluate and discuss the correctness of formal systems and also of computer software.

Issues of Correctness

The connection of formal languages to reality is the biggest problem for computers. Essentially programmers are thrown back to trial and error to see if their algorithms work. Remember the huge testing enterprise that Microsoft went through last year in order to find out all the bugs in their new Windows95 operating system. Programs written have to be tested against all possible uses in reality to see if they really fulfill what they were designed to do. But even after all these tests there are still bugs showing up due to the complexity of the system and the inability to verify all potential uses of the software products.

I think a clear parallel can be drawn with regular science. Science develops models of reality but these models first have to be tested in order to accept their content. Einstein's equations and theories about the basic nature of reality for example were heavily disputed until the bending of light was observed during a solar eclipse. Models are not the reflection of reality but a useful device to control some aspect of reality.

Implications

In computing formal languages finally are confronted with reality. Computing takes formal languages for the functionality and power of expression that they bring and varies those languages according to the purpose or need. The use of formal languages is essentially pragmatic. No one worries about Gödel's Theorems or ambiguous logic too much. If ambiguities exist then one or the other technique or language will be found to work around it. Practical use of formal systems disregards the idealistic assumptions of completeness and correctness coming from theoretical logic and mathematics.

Artificial Intelligence (AI)

What is Artificial Intelligence?

As of today Artificial Intelligence is not unanimously defined. Artificial Intelligence is a buzz word in some areas of the computer market and the term is stretched by various agendas. Common understanding of what AI is include:

- **Intelligent Behavior**. The machine is able to solve logical problems and communicate using natural language (Audi 1995, 47). The machine performs comparable to what an intelligent human being would do. This is the standard definition which will be used here.

- **A machine doing things usually done by humans**. Systems have Artificial Intelligence at the borderline of the acknowledged capabilities of machines. As the abilities of machines are accepted by society so the definition of Artificial Intelligence changes (Rich 1983,1).

- **Tesler's Proposal: Artificial Intelligent is all that has not been done yet** (Hofstadter 1979, 640). A more extreme variant on the last version. Thus artificial intelligence does not exist and will always be an aim to strive for.

- **Machines having minds** in exactly the same way we do. Another name for this kind of Artificial Intelligence is **Strong Artificial Intelligence.** All other definitions of Artificial Intelligence have consequently to be classified as **Weak Artificial Intelligence**.

The Turing Test

One of the founding fathers of the computer age Alan Turing proposed the following method to verify if a machine is thinking (Rich 1983, 18): A human is made to interact with another person and then with a machine. The machine is considered to be a thinking machine if the person cannot tell the difference between machine and the other person.

In the 1970s programs were designed to fake human interaction through a typewriter style interface. Famous programs of that era include ELIZA (simulates a Rogerian Psychologist) and PARRY (simulates a paranoid person) (Rich 1983, 17) . Those programs were extremely limited in the way they interacted with humans. Usually those early programs were only looking for keywords triggering standard forms of responses having no notion at all of context. But for some people (especially the computer novices) the impression of interaction was established and some persons were able to construct long dialogues with those programs and some others even insisted on the privacy of the conversation with the program (Hofstadter 1979, 322).

Currently there are commercial products on the market such as *PC Therapist V* of *Thinking Machines*[1]. PC Therapist can respond through the speakers of a regular desktop computer but still needs the keyboard for input and is sensitive to the context of communication. It seems to be gradually getting the capability of standing up to the Turing test performed by experts and has the ability to lead long conversations.

Arguments against AI

Algorithms are static

Scherb's opinion on Artificial Intelligence: If machines are to be intelligent like humans are then the thinking processes of machines must be as powerful as the human thought which is using natural languages. Computers need to be programmable in natural language in order to reach intelligence which is impossible since formal languages can never have the same capabilities as natural languages. All formal computer languages need a metalanguage to verify correctness. If computers would be able to use natural language then no metalanguage would be available anymore. Even if a computer would be programmable in a natural language then the program would be static. Human language changes by use (drawing on Wittgenstein) and through the experiences which is not true for a program. Also if such a powerful computer programming language would be developed then it would suffer from incompleteness according to Gödel's theorem and would be unusable (Scherb 1992, 66-68).

I think there are three assumptions here which make the argument doubtful:

A. The assumption that human thinking works in language.

[1] URL http://pages.prodigy.com/thinkingsoftware/jsw2.html

B. Computers are not programmed in only one formal language but have a variety of mechanism usually described in terms of formal languages at their disposal. It is not clear if the combination of techniques available in a computer are inferior to natural languages by principle or theory.

C. The assumption that programs are static and the identification of language with a program is not right anymore. A meta-program can be written (and has been written to simulate neurological networks) which in turn keeps experiences in a knowledge base. Reactions can then be performed based on an interpretation of the knowledge base and a connected decision algorithm. Thus the "programming language" of the so equipped computer is dynamic and not static anymore.

Lady Lovelace's Argument

Machines do not create they do whatever we tell them to do.

There have been Artificial Intelligence Programs written (first one in the 1960s!) which have been able to come up with proofs for the first chapters of Russell's *Principia Mathematica* through a search algorithm exploring the possibilities of proofs (Rich 1983,137).

The argument draws on the determinacy issue of computers. Computers are able to be non-determinate through a variety of means and could use that information to explore new avenues of reasoning.

Theological Argument

God has given humans a soul but no machine. Thus machines cannot think.

It depends on what is meant by thinking and what exactly the differentiating quality of the soul is that God has given. Both these notions are rather vague at this time so nothing definite can be said about the argument based on scientific reasoning.

Neurological Argument

The human nervous system is not discrete like the conditions in a digital machine.

The human nervous system has been partially simulated by neural networks.

Predictions

It is possible to construct machines that interact in an intelligent way with humans if we can understand the way the human brain works good enough. All our knowledge can be coded into formal systems and so the knowledge we gain about the human brain will also be able to be used by an intelligent machine. The intelligence of machines is determined by our understanding of intelligence.

I cannot see any reason why the technical issues of a computer should be a limitation to providing intelligence. Our knowledge about intelligence is the limit. It is not the limits of our computational facilities that is the problem but the limits of our own mental understanding of intelligence which might be an insurmountable barrier to Artificial Intelligence.

It is doubtful if machines can be constructed that will exactly behave like humans since humans have gone through a history of experiences largely determined by having a body. If an intelligent machine does not have a body, then the way in which it can be capable to communicate is limited. Wittgenstein once said: If a lion would be able to talk, we would not be able to understand it. Thus for a machine to be like us it not only needs the reasoning capabilities that we have but also the experiences that we have had. I would propose therefore that *Strong Artificial Intelligence* is not possible.

Conclusion

Formal languages are nothing ideal although they are powerful tools to communicate procedures and knowledge. Formal languages have the same issues as regular language terms to relate to the outside world. The expectations of the logical positivists at the beginning of the century were mistaken. Chihara is right accusing the positivists like Russell of Platonism since they assume these abstract concepts to have a reality! (Chihara 1973, 62). The enlightenment in trying to abolish metaphysical assumptions finally erected its own metaphysical building which inevitably fell down under further careful consideration. Formal systems also have to be seen in the context of a tradition of research as every other bit of knowledge.

The attitude taken by the pragmatic view of computer science of formal systems is I think the right one. Formal Systems should be used whenever they fit our needs but an awareness should be kept that formal systems are not everything and other forms of reasoning definitely have their value.

Bibliography

Audi, Robert ed. *The Cambridge Dictionary of Philosophy* Cambridge University Press, 1995

Blackburn, Simon *The Oxford Dictionary of Philosophy* New York: Oxford University Press, 1994

Chihara, Charles S. *ONTOLOGY and the VICIOUS-CIRCLE PRINCIPLE* Ithaca / London: Cornell University Press, 1973

Hawking, Stephen *Black Holes and Baby Universes and other Essays* Bantam Books, 1993

Hofstadter, Douglas R. *Gödel, Escher, Bach ein Endlos Geflochtenes Band* Stuttgart: Klett-Cotta, 1985 [Original: *Gödel, Escher, Bach: an Eternal Golden Braid* New York: Basic Books, 1979]

Jeffrey, Richard C. *Formal Logic: Its Scope and Limits* New York: McGraw-Hill, 1967

Klemke, E.D. *Essays on Bertrand Russell* Chicago: Univerity of Illinois Press, 1970

Rich, Elaine *Artificial Intelligence* New York etc: McGraw-Hill, 1983

Russell, Bertrand *LOGIC AND KNOWLEDGE: Essays 1901-1950* London / New York: Routledge, 1994 (Originally published in Great Britain 1956)

Scherb, Martin *Künstliche und Natürliche Sprachen: Semantik bei Wittgenstein und Tarski* Hildesheim: Georg OLMS AG, 1992