HUMAN-LEVEL AI: THE LOGICAL ROAD

John McCarthy
Computer Science Department
Stanford University
Stanford, CA 94305
jmc@cs.stanford.edu
http://www-formal.stanford.edu/jmc/

Contents

Chapter 1

INTRODUCTION

Logical AI since 1958 has had one major theme. It is to formalize aspects of the common sense knowledge and reasoning that were previously discussed only informally. Many aspects of the *common sense informatic situation* are not conveniently formalized using exclusively the methods that were successful in mathematics and physical science. Many philosophers and other people took this as evidence that logic was unsuitable for expressing common sense knowledge and reasoning with it. Instead we find that additional logical methods are needed and have proved capable of making progress.

Here are some specifics.

- Human level AI must operate in the *common sense informatic situation* in which humans think. It contrasts with the *bounded informatic situations* of formal scientific theories and most AI systems. There is no a priori limit on what information is relevant. See chapter 4.

- Mathematical logic was invented for human convenience, specifically to make definite the notion of correct reasoning. Formal logical theories are imbedded in informal human knowledge and reasoning. We might try to invent an informal machine way of representing knowledge and reasoning in which logical reasoning would be imbedded. Unfortunately, there are no proposals for such an "informal" language. Therefore, logical AI has always been completely formal. Tarski has shown that metalanguage can be formalized, and we propose to do that also.

- Human reasoning is often nonmonotonic. Therefore, it was necessary

to develop formal logical nonmonotonic reasoning methods. Circumscription and default logic were the first steps.

- Particular facts and reasoning methods are often correct only in a context. People deal with contexts informally. Logical AI must deal with them formally.

- The concepts used in common sense reasoning are often only approximate. A logical treatment of intrinsically approximate, i.e. ill-defined, concepts is required.

This book concerns steps in realizing these goals.

Logical AI is an ambitious approach to making intelligent computer programs—including programs of human level intelligence. It involves understanding the world well enough to put enough information in the form of logical axioms so that a good enough reasoning program can decide what to do by logical reasoning.

Logical AI has proved difficult, but it is probably closer to the goal of human level AI than the other approaches that have been tried. It has also had many successes in limited domains.

Animal-level intelligence can be achieved by simpler systems than are required for human-level AI. Logical AI is not needed for representing simple stimulus-response behavior. To the extent that an animal can predict the consequences of a sequence of actions, logical AI may be useful in describing animal level intelligence.

[1]

[1]Logical AI continues a line of inquiry that perhaps starts with Aristotle (384–322 BC) who began the study of the rules of reasoning. It continues with the development of mathematical logic by Leibniz, Boole, De Morgan, Frege, Peirce, ...

The goal was best expressed by Leibniz who wanted to replace dispute by logical calculation. Since he built calculators, I assume his idea included both human calculation and machine calculation.

Leibniz didn't get far with mathematical logic, but George Boole and Augustus De Morgan, followed by Gottlob Frege and Charles Peirce, created modern logic. Zermelo-Fraenkel set theory, as formalized in first order logic, is adequate for expressing all formal mathematical statements. The properties of this logic, including its strengths and limitations, were established in the 20th century by Gödel, Tarski, Church, Turing, Kleene and many others.

While ZF and ZFC are adequate for expressing mathematical definitions and facts, the proof in the systems described in the textbooks are too long and difficult to construct.

We often refer to *robots*. For the purposes of this book a robot is a computer program with sensors and effectors and a continued existence from task to task. We include in the concept programs whose sensors and effectors are purely symbolic, e.g. robots that live on the web, but we are most interested in robots with physical sensors and effectors including mobility.

Logical robots represent what they know about the world *mainly* by sentences in suitable *logical languages*.

When we refer to intelligent machines we mean intelligent computer programs except in the few cases when non-computer machines are mentioned explicitly.

1.1 Human-Level AI

The ambitious goal of artificial intelligence research is to make computer programs with human-level intelligence. Alan Turing was the first to formulate it. In his (Turing 1950), he conjectured that a computer program would successfully pretend to be human against a sophisticated opponent by the year 2000. Obviously, we didn't make it. Elsewhere, (**?**), he thought it might take 100 years.

Allen Newell and Herbert Simon were explicitly more optimistic in the late 1950s, but it seems to me that this optimism was based on the conjecture that the means-end methodology of their General Problem Solver GPS was adequate for human-level intelligence. Unfortunately, the GPS idea was inadequate.

Human-level AI needs to be distinguished from human-like AI. AI can use computational speeds and storage unavailable to humans. Moreover, many features of human intelligence resulting from our evolution are unneeded for AI.

Human-like AI is interesting in itself, and its evaluation requires detailed comparison with human performance. Moreover, AI still has a lot to learn

---

Systems admitting short proofs, especially when computers can be used to carry out routine though lengthy processes, are being developed. One attempt is the Mizar project.

However, the goal expressed by Leibniz, Boole, and Frege of expressing common sense reasoning in logical form requires extensions to mathematical logic to deal formally with contexts, with concepts as objects and for nonmonotonic reasoning. Even the common sense reasoning expressed in natural language in mathematics books requires extensions to logic if it is to be expressed formally. Perhaps the kinds of extensions proposed in this book will be adequate.

from studying human performance.

One tool is introspection—e.g. figuring out how one performs a mental task. Introspection has a bad name in psychology, because much 19th century psychology drew wrong and vague conclusions from it. Indeed the results of introspection need to be verified by other means. One of these other means is a psychological experiment. More important for AI is verifying whether an introspected method works in a program. It can also be verified that an introspected idea can be formalized suitably in logic.

Chapter **??** proposes a path to human-level logical AI. To realize this path, or some other, conceptual advances are required. Ultimate success is likely, because understanding human level intelligence is a scientific problem like many others that scientific research has solved and is solving. How long it will take cannot presently be predicted. I have offered the range of estimates—five years to five hundred years. New scientific ideas are probably required—not just scaling up present approaches.

The present state of AI research permits many applications, but some of the applications that have attempted require aspects of intelligence that are not understood well enough to be programmed.

Contrast logical AI with approaches based on simulated evolution. The evolutionists propose to make a system that can evolve into one that behaves intelligently. The designers of the system need not even be able to understand how it works—any more than we have full understanding of how human intelligence works. This approach should also succeed eventually.

Logical AI is more ambitious. We propose to understand the facts of the common sense world well enough to make a system that can act intelligently and can learn explicitly from its experience. The designers of logical AI systems will understand the intelligent behavior our programs exhibit. This approach will also succeed.

It's a race.

1.2 Common sense

It turns out that the key problem for human level AI is the problem of giving the program common sense.

1.3 Mathematical Logical Reasoning

The founders of mathematical logic, Leibniz, Boole and Frege, all wanted to use it to express common sense information. For example, Leibniz hoped to replace argument by calculation. Boole called his book "The Laws of Thought". Frege discussed both sense and denotation. Mathematical logic, as created by these men and as improved by their 20th century successors is adequate for expressing reasoning in mathematics and in the mathematical formalizations of various sciences. However, as we shall see, it needs to be supplemented by nonmonotonic reasoning in order to deal with the common sense informatic situation.

The simplest reasoning is forward reasoning from a set of premises.

It starts from a set A of sentences and infers from it some conclusions, say p, q, and r. The idea is that if the premises are believed, there is reason to believe the conclusions. In the conventional case of mathematical logic, the conclusions are true *whenever*[2] the premises are true. Therefore, believing the premises ought to compel one to believe the conclusions. We call such reasoning *rigorous* and the process of doing the reasoning *deduction*. Section **??** tells more.

The logic of correct reasoning was studied by Aristotle and later by Leibniz and Boole. However, the first complete system was first proposed in by Frege in 1879, proved complete by Gödel in 1930 and well established by the 1930s. (See (Van Heijenoort 1967) for the basic papers.) Humans can do mathematical logical reasoning and aspire to the rigor of mathematical logic but often don't make it—from inability, from a need for speed, or from mixing correct deductive reasoning and other forms of reasoning.

Deductive reasoning suffices for formalizing proofs of theorems in mathematics, although it is inadequate to formalize the process of finding worthwhile theorems or finding the proofs.

Deductive reasoning is *monotonic* in the sense that the set of conclusions $\mathcal{C}(A)$ that follow from a set A of premises is a monotonic increasing function of the set A.

Formalizing common sense requires more than deductive reasoning as we discuss briefly in the following sections.

[2]The common sense meaning of the word "whenever" is a good approximation here. However, there is a more precise notion in symbolic logic based on the concept of an *interpretation* of a logical language. What we want is that the conclusion shall be true in all interpretations in which the premises are true. More about this later.

Formal mathematics relies entirely on purely deductive reasoning, although nonmonotonic reasoning is to be found in the natural language discussions in mathematical books and articles.

1.4 Nonmonotonic reasoning

Unfortunately, the world is not so kind as to always provide us, or the robots we may build, with sufficient premises to get the conclusions we need solely by monotonic reasoning. We and our robots need to be braver at the cost of occasional error and having to take back some conclusions. Indeed we humans are braver and would hardly ever be able to decide what to do if we weren't. These braver forms of reasoning are mostly *nonmonotonic*.

Nonmonotonic reasoning is necessary for both humans and machines even though the conclusions reached do not always follow from the premises and are therefore not always true when the premises are true. A conclusion reached by correct nonmonotonic reasoning may have to be withdrawn when more facts become available.

Nonmonotonically inferred conclusions should be true in all *preferred* interpretations in which the premises are true.

Here's the capsule example of nonmonotonic reasoning. Suppose I hire you to build a cage for my bird. You will correctly infer that you have to put a top on the cage to prevent my bird from escaping. This is nonmonotonic reasoning, because if I add the assertion that my bird is a penguin, you will withdraw the conclusion that the cage needs a top.

The inadequacy of purely deductive reasoning has long been known, and systems have been developed for probabilistic reasoning and inductive reasoning. However, these do not cover the main cases where we need to go beyond deduction. All the systems I know about assume known what propositions are to be assigned probabilities, given fuzzy values or tested according to the evidence.

AI research since the late 1970s has developed *formalized nonmonotonic reasoning*, abbreviated NMR. These systems form new sentences rather than just assigning probabilities or truth values to sentences already present.

1.5 Elaboration tolerance

A formalism is *elaboration tolerant* to the extent that it is convenient to modify a set of facts expressed in the formalism to take into account new phenomena or changed circumstances. Representations of information in natural language have good elaboration tolerance when used with human background knowledge. Human-level AI will require representations with much more elaboration tolerance than those used by present AI programs, because human-level AI needs to be able to take new phenomena into account.

Chapter **??** and (McCarthy 1999c) treat elaboration tolerance in some detail. Much of the "brittleness" of present AI systems can be explained by their lack of elaboration tolerance.

Formalized mathematical theories rarely have much elaboration tolerance, because elaborations are done in the mental processes of the author and then expressed in the natural language part of books and articles. The author then makes a completely new logical formalization for the elaborated theory.

This won't do for logical AI, because logical AI requires that the mental processes be carried out in the logical language.

1.6 Concepts as objects

1.7 Contexts as objects

1.8 Approximate objects and theories

1.9 Consciousness

Chapter 2

CONCEPTS OF LOGICAL AI

2.1 Introduction

Logical AI involves representing knowledge of an agent's world, its goals and the current situation by sentences in logic. The agent decides what to do by inferring that a certain action or course of action was appropriate to achieve the goals. The inference may be *monotonic*, but the nature of the world and what can be known about it often requires that the reasoning be *nonmonotonic*.

Logical AI has both *epistemological* problems and *heuristic* problems. The former concern the knowledge needed by an intelligent agent and how it is represented. The latter concerns how the knowledge is to be used to decide questions, to solve problems and to achieve goals. These are discussed in (McCarthy and Hayes 1969a). Neither the *epistemological problems* nor the *heuristic* problems of logical AI have been solved. The epistemological problems are more fundamental, because the form of their solution determines what the heuristic problems will eventually be like.[1]

This article has links to other articles of mine. I'd like to supplement the normal references by direct links to such articles as are available.

[1]Thus the heuristics of a chess program that represents "My opponent has an open file for his rooks." by a sentence will be different from those of a present program which at most represents the phenomenon by the value of a numerical co-efficient in an evaluation function.

2.2 A LOT OF CONCEPTS

The uses of logic in AI and other parts of computer science that have been undertaken so far do not involve such an extensive collection of concepts. However, it seems to me that reaching human level AI will involve all of the following—and probably more.

Logical AI Logical AI in the sense of the present article was proposed in (McCarthy 1959) and also in (McCarthy 1989). The idea is that an agent can represent knowledge of its world, its goals and the current situation by sentences in logic and decide what to do by inferring that a certain action or course of action is appropriate to achieve its goals.

Logic is also used in weaker ways in AI, databases, logic programming, hardware design and other parts of computer science. Many AI systems represent facts by a limited subset of logic and use non-logical programs as well as logical inference to make inferences. Databases often use only ground formulas. Logic programming restricts its representation to Horn clauses. Hardware design usually involves only propositional logic. These restrictions are almost always justified by considerations of computational efficiency.

Epistemology and Heuristics In philosophy, epistemology is the study of knowledge, its form and limitations. This will do pretty well for AI also, provided we include in the study common sense knowledge of the world and scientific knowledge. Both of these offer difficulties philosophers haven't studied, e.g. they haven't studied in detail what people or machines can know about the shape of an object the field of view, remembered from previously being in the field of view, remembered from a description or remembered from having been felt with the hands. This is discussed a little in (McCarthy and Hayes 1969a).

Most AI work has concerned heuristics, i.e. the algorithms that solve problems, usually taking for granted a particular epistemology of a particular domain, e.g. the representation of chess positions.

Bounded Informatic Situation Formal theories in the physical sciences deal with a *bounded informatic situation*. Scientists decide informally in advance what phenomena to take into account. For example, much celestial mechanics is done within the Newtonian gravitational theory

and does not take into account possible additional effects such as outgassing from a comet or electromagnetic forces exerted by the solar wind. If more phenomena are to be considered, scientists must make a new theories—and of course they do.

Most AI formalisms also work only in a bounded informatic situation. What phenomena to take into account is decided by a person before the formal theory is constructed. With such restrictions, much of the reasoning can be monotonic, but such systems cannot reach human level ability. For that, the machine will have to decide for itself what information is relevant, and that reasoning will inevitably be partly nonmonotonic.

One example is the "blocks world" where the position of a block x is entirely characterized by a sentence $At(x, l)$ or $On(x, y)$, where l is a location or y is another block.

Another example is the Mycin (Davis et al. 1977) expert system in which the ontology (objects considered) includes diseases, symptoms, and drugs, but not patients (there is only one), doctors or events occurring in time. See (McCarthy 1983) for more comment.

Common Sense Knowledge of the World As first discussed in (McCarthy 1959), humans have a lot of knowledge of the world which cannot be put in the form of precise theories. Though the information is imprecise, we believe it can still be put in logical form. The Cyc project (Lenat and Guha 1990) aims at making a large base of common sense knowledge. Cyc is useful, but further progress in logical AI is needed for Cyc to reach its full potential.

Common Sense Informatic Situation In general a thinking human is in what we call the *common sense informatic situation*, as distinct from the *bounded informatic situation*. The known facts are necessarily incomplete. We live in a world of middle-sized object which can only be partly observed. We only partly know how the objects that can be observed are built from elementary particles in general, and our information is even more incomplete about the structure of particular objects. These limitations apply to any buildable machines, so the problem is not just one of human limitations.[2]

[2]Science fiction and scientific and philosophical speculation have often indulged in the

In many actual situations, there is no *a priori* limitation on what facts are relevant. It may not even be clear in advance what phenomena should be taken into account. The consequences of actions cannot be fully determined. The *common sense informatic situation* necessitates the use of *approximate concepts* that cannot be fully defined and the use of *approximate theories* involving them. It also requires *nonmonotonic* reasoning in reaching conclusions. Many AI texts assume that the information situation is bounded—without even mentioning the assumption explicitly.

The common sense informatic situation often includes some knowledge about the system's mental state as discussed in (McCarthy 1996c).

One key problem in formalizing the common sense informatic situation is to make the axiom sets elaboration tolerant2.2.

Epistemologically Adequate Languages A logical language for use in the common sense informatic situation must be capable of expressing directly the information actually available to agents. For example, giving the density and temperature of air and its velocity field and the Navier-Stokes equations does not practically allow expressing what a person or robot actually can know about the wind that is blowing. We and robots can talk about its direction, strength and gustiness approximately, and can give a few of these quantitities numerical values with the aid of instruments if instruments are available, but we have to deal with the phenomena even when no numbers can be obtained. The idea of epistemological adequacy was introduced in (McCarthy and Hayes 1969a).

Robot We can generalize the notion of a robot as a system with a variant of the physical capabilities of a person, including the ability to move around, manipulate objects and perceive scenes, all controlled by a computer program. More generally, a robot is a computer-controlled system that can explore and manipulate an environment that is not part of the robot itself and is, in some important sense, larger than the robot. A robot should maintain a continued existence and not reset itself to a standard state after each task. From this point of view, we

Laplacian fantasy of super beings able to predict the future by knowing the positions and velocities of all the particles. That isn't the direction to go. Rather they would be better at using the information that is available to the senses.

can have a robot that explores and manipulates the Internet without it needing legs, hands and eyes. The considerations of this article that mention robots are intended to apply to this more general notion. The internet robots discussed so far are very limited in their mentalities.

Qualitative Reasoning This concerns reasoning about physical processes when the numerical relations required for applying the formulas of physics are not known. Most of the work in the area assumes that information about what processes to take into account are provided by the user. Systems that must be given this information often won't do human level qualitative reasoning. See (Weld and de Kleer 1990) and (Kuipers 1994).

Common Sense Physics Corresponds to people's ability to make decisions involving physical phenomena in daily life, e.g. deciding that the spill of a cup of hot coffee is likely to burn Mr. A, but Mr. B is far enough to be safe. It differs from qualitative physics, as studied by most researchers in *qualitative reasoning*, in that the system doing the reasoning must itself use common sense knowledge to decide what phenomena are relevant in the particular case. See (Hayes 1985) for one view of this.

Expert Systems These are designed by people, i.e. not by computer programs, to take a limited set of phenomena into account. Many of them do their reasoning using logic, and others use formalisms amounting to subsets of first order logic. Many require very little common sense knowledge and reasoning ability. Restricting expressiveness of the representation of facts is often done to increase computational efficiency.

Knowledge Level Allen Newell ((Newell 1982) and (Newell 1993)) did not advocate (as we do here) using logic as the way a system should represent its knowledge internally. He did say that a system can often be appropriately described as knowing certain facts even when the facts are not represented by sentences in memory. This view corresponds to Daniel Dennett's *intentional stance* (Dennett 1971), reprinted in (Dennett 1978), and was also proposed and elaborated in (McCarthy 1979b).

Elaboration Tolerance A set of facts described as a logical theory needs to be modifiable by adding sentences rather than only by going back

to natural language and starting over. For example, we can modify the missionaries and cannibals problem by saying that there is an oar on each bank of the river and that the boat can be propelled with one oar carrying one person but needs two oars to carry two people. Some formalizations require complete rewriting to accomodate this elaboration. Others share with natural language the ability to allow the elaboration by an addition to what was previously said.

There are degrees of elaboration tolerance. A state space formalization of the missionaries and cannibals problem in which a state is represented by a triplet (m c b) of the numbers of missionaries, cannibals and boats on the initial bank is less elaboration tolerant than a situation calculus formalism in which the set of objects present in a situation is not specified in advance. In particular, the former representation needs surgery to add the oars, whereas the latter can handle it by adjoining more sentences—as can a person. The realization of elaboration tolerance requires nonmonotonic reasoning. See (McCarthy 1997).

Robotic Free Will Robots need to consider their choices and decide which of them leads to the most favorable situation. In doing this, the robot considers a system in which its own outputs are regarded as free variables, i.e. it doesn't consider the process by which it is deciding what to do. The perception of having choices is also what humans consider as *free will*. The matter is discussed in (McCarthy and Hayes 1969a) and is roughly in accordance with the philosophical attitude towards free will called *compatibilism*, i.e. the view that determinism and free will are compatible.

Reification To refify an entity is to "make a thing" out of it (from Latin *re* for *thing*). From a logical point of view, things are what variables can range over. Logical AI needs to *reify* hopes, intentions and "things wrong with the boat". Some philosophers deplore reification, referring to a "bloated ontology", but AI needs more things than are dreamed of in the philosophers' philosophy. In general, reification gives a language more expressive power, because it permits referring to entities directly that were previously mentionable only in a metalanguage.

Ontology In philosophy, ontology is the branch that studies what things exist. W.V.O. Quine's view is that the ontology is what the variables

range over. Ontology has been used variously in AI, but I think Quine's usage is best for AI. "Reification" and "ontology" treat the same phenomena. Regrettably, the word "ontology" has become popular in AI in much vaguer senses. Ontology and reification are basically the same concept.

Approximate Concepts Common sense thinking cannot avoid concepts without clear definitions. Consider the welfare of an animal. Over a period of minutes, the welfare is fairly well defined, but asking what will benefit a newly hatched chick over the next year is ill defined. The exact snow, ice and rock that constitutes Mount Everest is ill defined. The key fact about approximate concepts is that while they are not well defined, sentences involving them may be quite well defined. For example, the proposition that Mount Everest was first climbed in 1953 is definite, and its definiteness is not compromised by the ill-definedness of the exact boundaries of the mountain. See (McCarthy 1999d).

There are two ways of regarding approximate concepts. The first is to suppose that there is a precise concept, but it is incompletely known. Thus we may suppose that there is a truth of the matter as to which rocks and ice constitute Mount Everest. If this approach is taken, we simply need weak axioms telling what we do know but not defining the concept completely.

The second approach is to regard the concept as intrinsically approximate. There is no truth of the matter. One practical difference is that we would not expect two geographers independently researching Mount Everest to define the same boundary. They would have to interact, because the boundaries of Mount Everest are yet to be defined.[3]

Approximate Theories Any theory involving approximate concepts is an approximate theory. We can have a theory of the welfare of chickens. However, its notions don't make sense if pushed too far. For example, animal rights people assign some rights to chickens but cannot define them precisely. It is not presently apparent whether the expression of approximate theories in mathematical logical languages will require any innovations in mathematical logic. See (McCarthy 1999d).

[3]Regarding a concept as intrinsically approximate is distinct from either regarding it as fully defined by nature or fully defined by human convention.

Ambiguity Tolerance Assertions often turn out to be ambiguous with the ambiguity only being discovered many years after the assertion was enunciated. For example, it is *a priori* ambiguous whether the phrase "conspiring to assault a Federal official" covers the case when the criminals mistakenly believe their intended victim is a Federal official. An ambiguity in a law does not invalidate it in the cases where it can be considered unambiguous. Even where it is formally ambiguous, it is subject to judicial interpretation. AI systems will also require means of isolating ambiguities and also contradictions. The default rule is that the concept is not ambiguous in the particular case. The ambiguous theories are a kind of approximate theory.

Causal Reasoning A major concern of logical AI has been treating the consequences of actions and other events. The *epistemological* problem concerns what can be known about the laws that determine the results of events. A theory of causality is pretty sure to be approximate.

Situation Calculus Situation calculus is the most studied formalism for doing causal reasoning. A situation is in principle a snapshot of the world at an instant. One never knows a situation—one only knows facts about a situation. Events occur in situations and give rise to new situations. There are many variants of situation calculus, and none of them has come to dominate. (McCarthy and Hayes 1969a) introduces situation calculus. (Gelfond et al. 1991) is a 1991 discussion.

Fluents Functions of situations in situation calculus. The simplest fluents are *propositional* and have truth values. There are also fluents with values in numerical or symbolic domains. *Situational fluents* take on situations as values.

Frame Problem This is the problem of how to express the facts about the effects of actions and other events in such a way that it is not necessary to explicitly state for every event, the fluents it does not affect. Murray Shanahan (Shanahan 1997) has an extensive discussion.

Qualification Problem This concerns how to express the preconditions for actions and other events. That it is necessary to have a ticket to fly on a commercial airplane is rather unproblematical to express. That it is necessary to be wearing clothes needs to be kept inexplicit unless it somehow comes up.

Ramification Problem Events often have other effects than those we are immediately inclined to put in the axioms concerned with the particular kind of event.

Projection Given information about a situation, and axioms about the effects of actions and other events, the projection problem is to determine facts about future situations. It is assumed that no facts are available about future situations other than what can be inferred from the "known laws of motion" and what is known about the initial situation. Query: how does one tell a reasoning system that the facts are such that it should rely on projection for information about the future.

Planning The largest single domain for logical AI has been planning, usually the restricted problem of finding a finite sequence of actions that will achieve a goal. (Green 1969a) is the first paper to use a theorem prover to do planning. Planning is somewhat the inverse problem to projection.

Narrative A narrative tells what happened, but any narrative can only tell a certain amount. What narratives can tell, how to express that logically, and how to elaborate narratives is given a preliminary logical treatment in (McCarthy 1995b) and more fully in (McCarthy and Costello 1998). (Pinto and Reiter 1993) and (R.S.Miller and M.P.Shanahan 1994) are relevant here. A narrative will usually give facts about the future of a situation that are not just consequences of projection from an initial situation. [While we may suppose that the future is entirely determined by the initial situation, our knowledge doesn't permit inferring all the facts about it by projection. Therefore, narratives give facts about the future beyond what follows by projection.]

Understanding A rather demanding notion is most useful. In particular, fish do not understand swimming, because they can't use knowledge to improve their swimming, to wish for better fins, or to teach other fish. See the section on understanding in (McCarthy 1996c). Maybe fish do learn to improve their swimming, but this presumably consists primarily of the adjustment of parameters and isn't usefully called understanding. I would apply understanding only to some systems that can do hypothetical reasoning—if p were true, then q would be true. Thus Fortran compilers don't understand Fortran.

Consciousness, awareness and introspection Human level AI systems will require these qualities in order to do tasks we assign them. In order to decide how well it is doing, a robot will need to be able to examine its goal structure and the structure of its beliefs from the *outside*. See (McCarthy 1996c).

Intention to do something Intentions as objects are discussed briefly in (McCarthy 1989) and (McCarthy 1996c).

Mental situation calculus The idea is that there are mental situations, mental fluents and mental events that give rise to new mental situations. The mental events include observations and inferences but also the results of observing the mental situation up to the current time. This allows drawing the conclusion that there isn't yet information needed to solve a certain problem, and therefore more information must be sought outside the robot or organism. (Scherl and Levesque 1993) treats this and so does (McCarthy 1996c).

Discrete processes Causal reasoning is simplest when applied to processes in which discrete events occur and have definite results. In situation calculus, the formulas $s' = result(e, s)$ gives the new situation s' that results when the event e occurs in situation s. Many continuous processes that occur in human or robot activity can have *approximate theories* that are discrete.

Continuous Processes Humans approximate continuous processes with representations that are as discrete as possible. For example, "Junior read a book while on the airplane from Glasgow to London." Continuous processes can be treated in the situation calculus, but the theory is so far less successful than in discrete cases. We also sometimes approximate discrete processes by continuous ones. (Miller 1996) and (Reiter 1996) treat this problem.

Non-deterministic events Situation calculus and other causal formalisms are harder to use when the effects of an action are indefinite. Often $result(e, s)$ is not usefully axiomatizable and something like $occurs(e, s)$ must be used.

Concurrrent Events Formalisms treating actions and other events must

allow for any level of dependence between events. Complete independence is a limiting case and is treated in (McCarthy 1995b).

Conjunctivity It often happens that two phenomena are independent. In that case, we may form a description of their combination by taking the conjunction of the descriptions of the separate phenomena. The description language satisfies *conjunctivity* if the conclusions we can draw about one of the phenomena from the combined description are the same as the conjunctions we could draw from the single description. For example, we may have separate descriptions of the assassination of Abraham Lincoln and of Mendel's contemporaneous experiments with peas. What we can infer about Mendel's experiments from the conjunction should ordinarily be the same as what we can infer from just the description of Mendel's experiments. Many formalisms for concurrent events don't have this property, but *conjunctivity* itself is applicable to more than concurrent events.

To use logician's language, the conjunction of the two theories should be a conservative extension of each of the theories. Actually, we may settle for less. We only require that the inferrable sentences about Mendel (or about Lincoln) in the conjunction are the same. The combined theory may admit inferring other sentences in the language of the separate theory that weren't inferrable in the separate theories.

Learning Making computers learn presents two problems—*epistemological* and *heuristic*. The epistemological problem is to define the space of concepts that the program can learn. The heuristic problem is the actual learning algorithm. The heuristic problem of algorithms for learning has been much studied and the epistemological mostly ignored. The designer of the learning system makes the program operate with a fixed and limited set of concepts. Learning programs will never reach human level of generality as long as this approach is followed. (McCarthy 1959) says, **"A computer can't learn what it can't be told."** We might correct this, as suggested by Murray Shanahan, to say that it can only learn what can be expressed in the language we equip it with. To learn many important concepts, it must have more than a set of weights. (Muggleton and De Raedt 1994) and (Bratko and Muggleton 1995) present some progress on learning within a logical language. The many kinds of learning discussed in

(Mitchell 1997) are all, with the possible exception of inductive logic programming, very limited in what they can represent—and hence can conceivably learn. (McCarthy 1999a) presents a challenge to machine learning problems and discovery programs to learn or discovery the reality behind appearance.

Representation of Physical Objects We aren't close to having an epistemologically adequate language for this. What do I know about my pocket knife that permits me to recognize it in my pocket or by sight or to open its blades by feel or by feel and sight? What can I tell others about that knife that will let them recognize it by feel, and what information must a robot have in order to pick my pocket of it?

Representation of Space and Shape We again have the problem of an epistemologically adequate representation. Trying to match what a human can remember and reason about when out of sight of the scene is more what we need than some pixel by pixel representation. Some problems of this are discussed in (McCarthy 1995a) which concerns the Lemmings computer games. One can think about a particular game and decide how to solve it away from the display of the position, and this obviously requires a compact representation of partial information about a scene.

Discrimination, Recognition and Description *Discrimination* is the deciding which category a stimulus belongs to among a fixed set of categories, e.g. decide which letter of the alphabet is depicted in an image. *Recognition* involves deciding whether a stimulus belongs to the same set, i.e. represents the same object, e.g. a person, as a previously seen stimulus. *Description* involves describing an object in detail appropriate to performing some action with it, e.g. picking it up by the handle or some other designated part. Description is the most ambitious of these operations and has been the forte of logic-based approaches.

Logical Robot (McCarthy 1959) proposed that a robot be controlled by a program that infers logically that a certain action will advance its goals and then does that action. This approach was implemented in (Green 1969b), but the program was very slow. Shortly greater speed was obtained in systems like STRIPS at the cost of limiting the

generality of facts the robot takes into account. See (Nilsson 1984), (Levesque et al. 1997), and (Shanahan 1996).

Declarative Expression of Heuristics (McCarthy 1959) proposes reasoning be controlled by domain-dependent and problem-dependent heuristics expressed declaratively. Expressing heuristics declaratively means that a sentence about a heuristic can be the result of reasoning and not merely something put in from the outside by a person. Josefina Sierra (Sierra 1998b), (Sierra 1998a), (Sierra 1998c), (Sierra 1999) has made some recent progress.

Logic programming Logic programming isolates a subdomain of first order logic that has nice computational properties. When the facts are described as a logic program, problems can often be solved by a standard program, e.g. a Prolog interpreter, using these facts as a program. Unfortunately, in general the facts about a domain and the problems we would like computers to solve have that form only in special cases.

Useful Counterfactuals "If another car had come over the hill when you passed that Mercedes, there would have been a head-on collision." One's reaction to believing that counterfactual conditional sentence is quite different from one's reaction to the corresponding material conditional. Machines need to represent such sentences in order to learn from not-quite-experiences. See (Costello and McCarthy 1998).

Formalized Contexts Any particular bit of thinking occurs in some context. Humans often specialize the context to particular situations or theories, and this makes the reasoning more definite, sometimes completely definite. Going the other way, we sometimes have to generalize the context of our thoughts to take some phenomena into account.

It has been worthwhile to admit contexts as objects into the ontology of logical AI. The prototype formula $ist(c, p)$ asserts that the proposition p is true in the context c. The formal theory is discussed in (McCarthy 1993), (McCarthy and Buvač 1998) and in papers by Saša Buvač, available in (Buvač 1995a).

Rich and Poor Entities A *rich entity* is one about which a person or machine can never learn all the facts. The state of the reader's body is a rich entity. The actual history of my going home this evening is a

rich entity, e.g. it includes the exact position of my body on foot and in the car at each moment. While a system can never fully describe a rich entity, it can learn facts about it and represent them by logical sentences.

Poor entities occur in plans and formal theories and in accounts of situations and events and can be fully prescribed. For example, my plan for going home this evening is a poor entity, since it does not contain more than a small, fixed amount of detail. Rich entities are often approximated by poor entities. Indeed some rich entities may be regarded as inverse limits of trees of poor entities. (The mathematical notion of inverse limit may or may not turn out to be useful, although I wouldn't advise anyone to study the subject quite yet just for its possible AI applications.)

Nonmonotonic Reasoning Both humans and machines must draw conclusions that are true in the "*best*" models of the facts being taken into account. Several concepts of *best* are used in different systems. Many are based on minimizing something. When new facts are added, some of the previous conclusions may no longer hold. This is why the reasoning that reached these conclusions is called nonmonotonic.

Probabilistic Reasoning Probabilistic reasoning is a kind of nonmonotonic reasoning. If the probability of one sentence is changed, say given the value 1, other sentences that previously had high probability may now have low or even 0 probability. Setting up the probabilistic models, i.e defining the sample space of "events" to which probabilities are to be given often involves more general nonmonotonic reasoning, but this is conventionally done by a person informally rather than by a computer.

In the open common sense informatic situation, there isn't any apparent overall sample space. Probabilistic theories may formed by limiting the space of events considered and then establishing a distribution. Limiting the events considered should be done by whatever nonmonotonic reasoning techniques are developed techniques for limiting the phenomena taken into account. (You may take this as a confession that I don't know these techniques.) In forming distributions, there would seem to be a default rule that two events e_1 and e_2 are to be taken as independent unless there is a reason to do otherwise. e_1 and e_2 can't be just

any events but have to be in some sense basic events.

Circumscription A method of nonmonotonic reasoning involving minimizing predicates (and sometimes domains). It was introduced in (McCarthy 1976), (McCarthy 1980) and (McCarthy 1986). An up-to-date discussion, including numerous variants, is (Lifschitz 1994).

Default Logic A method of nonmonotonic reasoning introduced in (Reiter 1980) that is the main survivor along with circumscription.

Yale Shooting Problem This problem, introduced in (Hanks and McDermott 1986), is a simple *Drosophila* for nonmonotonic reasoning. The simplest formalizations of causal reasoning using circumscription or default logic for doing the nonmonotonic reasoning do not give the result that intuition demands. Various more recent formalizations of events handle the problem ok. The Yale shooting problem is likely to remain a benchmark problem for formalizations of causality.

Design Stance Daniel Dennett's idea (Dennett 1978) is to regard an entity in terms of its function rather than in terms of its physical structure. For example, a traveller using a hotel alarm clock need not notice whether the clock is controlled by a mechanical escapement, the 60 cycle power line or by an internal crystal. We formalize it in terms of (a) the fact that it can be used to wake the traveller, and (b) setting it and the noise it makes at the time for which it is set.

Physical Stance We consider an object in terms of its physical structure. This is needed for actually building it or repairing it but is often unnecessary in making decisions about how to use it.

Intentional Stance Dennett proposes that sometimes we consider the behavior of a person, animal or machine by ascribing to it belief, desires and intentions. This is discussed in (Dennett 1971) and (Dennett 1978) and also in (McCarthy 1979b).

Relation between logic and calculation and various data structures Murray Shanahan recommends putting in something about this.

Creativity Humans are sometimes creative—perhaps rarely in the life of an individual and among people. What is creativity? We consider

creativity as an aspect of the solution to a problem rather than as attribute of a person (or computer program).

A creative solution to a problem contains a concept not present in the functions and predicates in terms of which the problem is posed. (McCarthy 1964) and (McCarthy 1999b)discuss the mutilated checkerboard problem.

The problem is to determine whether a checkerboard with two diagonally opposite squares can be removed can be covered with dominoes, each of which covers two rectilinearly adjacent squares. The standard proof that this can't be done is *creative* relative to the statement of the problem. It notes that a domino covers two squares of opposite color, but there are 32 squares of one color and 30 of the other color to be colored.

Colors are not mentioned in the statement of the problem, and their introduction is a creative step relative to this statement. For a mathematician of moderate experience (and for many other people), this bit of creativity is not difficult. We must, therefore, separate the concept of creativity from the concept of difficulty.

Before we can have creativity we must have some elaboration tolerance2.2. Namely, in the simple languagge of *A tough nut* ..., the colors of the squares cannot even be expressed. A program confined to this language could not even be told the solution. As discussed in (McCarthy 1996d), Zermelo-Frankel set theory is an adequate language. In general, set theory, in a form allowing definitions may have enough elaboration tolerance in general. Regard this as a conjecture that requires more study.

How it happened Consider an action like buying a pack of cigarettes on a particular occasion and the subactions thereof. It would be a mistake to regard the relation between the action and its subactions as like that between a program and its subroutines. On one occasion I might have bought the cigarettes from a machine. on a second occasion at a supermarket, and on a third occasion from a cigarettelegger, cigarettes having become illegal.

2.3 Alphabetical Index

Nonmonotonic Reasoning
Ontology
Physical Stance
Planning
Probabilistic Reasoning
Projection
Qualification Problem
Qualitative Reasoning
Ramification Problem
Reification
Relation between logic and calculation and various data structures
Representation of Physical Objects
Representation of Space and Shape
Rich and Poor Entities
Robotic Free Will
Situation Calculus
Understanding
Useful Counterfactuals
Yale Shooting Problem

Chapter 3

PHILOSOPHICAL AND SCIENTIFIC PRESUPPOSITIONS OF LOGICAL AI

Extinguished theologians lie about the cradle of every science as the strangled snakes beside that of Hercules.
—T. H. Huxley [1]

Abstract: Many ideas from philosophy, especially from recent analytic philosophy, are usable for AI. However, some philosophical points of view make assumptions that have the effect of excluding the possibility of AI. Likewise work on AI is not neutral with regard to philosophical issues. This chapter presents what we consider the presuppositions of logical AI and also some scientific presuppositions, i.e. some results of science that are relevant. We emphasize the relation to AI rather than philosophy itself.

3.1 Philosophical Presuppositions

Q. Why bother stating philosophical presuppositions? Why not just get on with the AI?

[1] Thomas H. Huxley (1825–1895), the major polemicist for the theory of evolution, was referred to as Darwin's bulldog. Progress in AI may help extinguish some philosophies, but don't stand on one foot.

A. AI shares many concerns with philosophy—with metaphysics, epistemology, philosophy of mind and other branches of philosophy. AI researchers who ignore philosophy don't notice some of these common problems. Many of thse problems are important for AI, especially AI research aimed at human-level AI. This is because AI concerns the creation of an artificial mind. However, AI has to treat these questions in more detail than philosophers customarily consider relevant.[2]

In principle, an evolutionary approach to AI need not involve philosophical presuppositions. However, many putative evolutionary approaches are crippled by impoverished philosophical assumptions. For example, the systems often only admit patterns in appearance and can't even represent reality behind appearance. (McCarthy 1999a) presents a challenge to learning systems to learn reality behind appearance.

AI research not based on stated philosophical presuppositions usually turns out to be based on unstated philosophical presuppositions. These are often so wrong as to interfere with developing intelligent systems.

That it should be possible to make machines as intelligent as humans involves some philosophical premises, although the possibility is probably accepted by a majority of philosophers. The way we propose to build intelligent machines, i.e. via logical AI, makes more presuppositions, some of which may be new.

This chapter concentrates on stating the presuppositions and their relations to AI without much philosophical argument. A later chapter presents arguments and discusses other opinions.

objective world The world exists independently of humans. The facts of mathematics and physical science are independent of there being people to know them. Intelligent Martians and robots will need to know the same facts.

A robot also needs to believe that the world exists independently of itself. Science tells us that humans evolved in a world which formerly did not contain humans. Given this, it is odd to regard the world as a human construct. It would be even more odd to program a robot

[2]For example, the treatment of counterfactual conditional sentences in (Costello and McCarthy 1999) goes into detail about how counterfactuals are to be inferred from non-counterfactuals and used to infer non-counterfactuals whereas that in (Lewis 1973) and in other philosophical literature is mainly concerned with the truth conditions of counterfactuals.

to regard the world as its own construct. What the robot believes about the world in general doesn't arise for the limited robots of today, because the languages they are programmed to use can't express assertions about the world in general. This limits what they can learn or can be told—and hence what we can get them to do for us.

correspondence theory of truth and reference A logical robot represents what it *believes* about the world by logical sentences. Some of these beliefs we build in; others come from its observations and still others by induction from its experience. Within the sentences it uses *terms* to refer to objects in the world.

In every case, we try to design it so that what it will believe about the world is as accurate as possible, though not usually as detailed as possible. Debugging and improving the robot includes detecting false beliefs about the world and changing the way it acquires information to maximize the correspondence between what it believes and the facts of world. The terms the robot uses to refer to entities need to correspond to the entities so that the sentences will express facts about these entities. We have in mind both material objects and other entities, e.g. plans.

Already this involves a philosophical presupposition—that which is called the *correspondence theory of truth*. AI also needs a *correspondence theory of reference* , i.e. that a mental structure can refer to an external object and can be judged by the accuracy of the reference.

As with science, a robot's theories are tested experimentally, but the concepts robots use are often not defined in terms of experiments. Their properties are partially axiomatized, and some axioms relate terms to observations.

It is possible to be too finicky about truth. Richard von Mises (von Mises 1951) states that whereas twelve is divisible by three and thirteen is not, "but a shepherd can no more divide twelve sheep than thirteen into three absolutely "equal" parts." The common sense robots must have should emphasize the sense of truth and the sense of division into parts in which it is true that twelve sheep can be divided into three equal parts.Actually, robots should distinguish among natural numbers, rationals,reals, complex numbers, so maybe von Mises's point is just beside the point.

The important consequence of the correspondence theory is that when we design robots, we need to keep in mind the relation between *appearance*, the information coming through the robot's sensors, and *reality*. Only in certain simple cases, e.g. the position in a chess game, does the robot have sufficient access to reality for this distinction to be ignored.

Some robots react directly to their inputs without memory or inferences. It is our scientific (i.e. not philosophical) contention that these are inadequate for human-level intelligence, because the world contains too many important entities that cannot be observed directly and must reason about hypothetical and future situations that cannot be immediately observed.

A robot that reasons about the acquisition of information must itself be aware of these relations. In order that a robot should not always believe what it sees with its own eyes, it must distinguish between appearance and reality. It must learn more about these relations from its experience, from being taught and from reading. (McCarthy 1999a) presents a challenge problem requiring the discovery of reality behind appearance.

The correspondence theory of truth may be contrasted with *pragmatic theories of truth* in which beliefs are regarded as true if they result in success in achieving goals. Each kind of theory has adherents among philosophers. Roughly speaking, pragmatic theories of truth correspond to making *reactive robots* that respond directly to inputs. Some behaviors can be programmed this way, but logical robots are appropriately designed to *do what they think will advance their goals.*

science Science is substantially correct in what it tells us about the world, and scientific activity is the best way to obtain more knowledge. 20th century corrections to scientific knowledge mostly left the old scientific theories as good approximations to reality.

Slogan: Atoms are as real as rocks. 102 years elapsed from Dalton's article on atoms in 1803 to Einstein's article on the Brownian motion in 1905 that convinced the last significant scientific holdout, Wilhelm Ostwald, that atoms are real. That this took a long time to discover and to convince scientists is a fact about the position of humans in the world and our opportunities to observe, not a fundamental fact about

the world. Alas it didn't convince the main philosophical holdout, Ernst Mach.

mind and brain The human mind is an activity of the human brain. This is a scientific proposition, supported by all the evidence science has discovered so far. However, the dualist intuition of separation between mind and body is related to the sometimes weak connections between thought and action. Dualist models may have some use as psychological abstractions.

common sense Common sense ways of perceiving the world and common opinion are also substantially correct. When general common sense errs, it can often be corrected by science, and the results of the correction may become part of common sense if their use doesn't require numerical observation or high speed numerical computation. Thus common sense has absorbed the notion of inertia. However, its mathematical generalization, the law of conservation of *linear* momentum has made its way into the common sense of only a small fraction of people—even among the people who have taken courses in physics.

From Socrates on philosophers have found many inadequacies in common sense usage, e.g. common sense notions of the meanings of words. The corrections are often elaborations, making distinctions blurred in common sense usage. Unfortunately, there is no end to philosophical elaboration, and the theories become very complex. However, some of the elaborations seem essential to avoid confusion in some circumstances. Here's a candidate for the way out of the maze.

Robots will need both the simplest common sense usages and to be able to tolerate elaborations when required. For this we have proposed two notions—contexts as formal objects (McCarthy 1993) and (McCarthy and Buvač 1997) and *elaboration tolerance* discussed in **??**[3]

[3]Hilary Putnam (Putnam 1975) discusses two notions concerning meaning proposed by previous philosophers which he finds inadequate. These are

> (I) That knowing the meaning of a term is just a matter of being in a certain "psychological state" (in the sense of "psychological state" in which states of memory and psychological dispositions are "psychological states"; no one thought that knowing the meaning of a word was a continuous state of consciousness, of course.)
>
> (II) That the meaning of a term (in the sense of "intension") determines

science embedded in common sense Science is embedded in common sense. Galileo taught us that the distance s that a dropped body falls in time t is given by the formula

$$s = \frac{1}{2}gt^2.$$

To use this information, the English (or its logical equivalent) is just as essential as the formula, and common sense knowledge of the world is required to make the measurements required to use or verify the formula. Without the embedding there is no difference between Galileo's formula and

$$E = \frac{1}{2}CV^2.$$

The English explains that the latter gives the energy stored in a capacitor as a function of the capacitance and the voltage.

possibility of AI According to some philosophers' views, artificial intelligence is either a contradiction in terms (Searle 1984) or intrinsically impossible (Dreyfus 1992) or (Penrose 1994). The methodological basis of these arguments has to be wrong and not just the arguments themselves. We will discuss some of this in Chapter 16.2 containing some polemics.

mental qualities treated individually AI has to treat mind in terms of components rather than regarding mind as a unit that necessarily has all the mental features that occur in humans. Thus we design some very simple systems in terms of the beliefs we want them to have and

---

its extension (in the sense that sameness of intension entails sameness of extension).

Suppose Putnam is right in his criticism of the general correctness of (I) and (II). We don't discuss his own more elaborate ideas.

It may be convenient for a robot to work mostly in contexts within a larger context C_{phil1} in which (I) and (II) (or something even simpler) hold. However, the same robot, if it is to have human level intelligence, must be able to *transcend* C_{phil1} when it has to work in contexts to which Putnam's criticisms of the assumptions of C_{phil1} apply.

It is interesting, and perhaps necessary, for AI at first, to characterize those contexts in which (I) and (II) are correct.

debug them by identifying erroneous beliefs. (McCarthy 1979b) treats this. Ascribing a few beliefs to thermostats has led to controversy. Most philosophers don't like it, but Daniel Dennett and I are on the same side of this issue.

third person point of view We ask "How does it (or he) know?", "What does it perceive?" rather than how do I know and what do I perceive. This presupposes the correspondence theory of truth. It applies to how we look at robots, but also to how we want robots to reason about the knowledge of people and other robots. Some philosophers, e.g. John Searle, insist on a first person point of view.

rich ontology Our theories involve many kinds of entity—material objects, situations, properties as objects, contexts, propositions, indivdual concepts, wishes, intentions. When one kind A of entity might be defined in terms of others, we will often prefer to treat A separately, because we may later want to change our ideas of its relation to other entities.

We often consider several related concepts, where others have tried to get by with one. Suppose a man sees a dog. Is seeing a relation between the man and the dog or a relation between the man and an appearance of a dog? Some purport to refute calling seeing a relation between the man and the dog by pointing out that the man may actually see a hologram or picture of the dog. AI needs the relation between the man and the appearance of a dog, the relation between the man and the dog and also the relation between dogs and appearances of them. Advocating one of these as what "seeing really is" is unfruitful.

natural kinds The entities the robot must refer to often are *rich* with properties the robot cannot know all about. The best example is a *natural kind* like a lemon. A child buying a lemon at a store knows enough properties of the lemons that occur in the stores he frequents to distinguish lemons from other fruits in the store. Experts know more properties of lemons, but no-one knows all of them. AI systems also have to distinguish between sets of properties that suffice to recognize an object in particular situations and the natural kinds of some objects.

To a child, all kinds are natural kinds, i.e. kinds about which the child is ready to learn more. The idea of a concept having an if-and-only-if definition comes later—perhaps at ages 10–13. Taking that further,

natural kind seems to be a context relative notion. Thus some part of income tax law is a natural kind to me, whereas it might have an if-and-only-if definition to an expert.

Curiously enough, many of the notions studied in philosophy are not natural kinds, e.g. proposition, meaning, necessity. When they are regarded as natural kinds, then fruitless arguments about what they really are take place. AI needs these concepts but must be able to work with limited notions of them.

approximate entities Many of the philosophical arguments purporting to show that naive common sense is hopelessly mistaken are wrong. These arguments often stem from trying to force intrinsically approximate concepts into the form of if-and-only-if definitions.

Our emphasis on the first class character of approximate entities may be new. It means that we can quantify over approximate entities and also express how an entity is approximate. (McCarthy 2000) treats approximate concepts and approximate theories.

compatibility of determinism and free will A logical robot needs to consider its choices and the consequences of them. Therefore, it must regard itself as having *free will* even though it is a deterministic device.

We discuss our choices and those of robots by considering non-determinist approximations to a determinist world—or at least a world more determinist than is needed in the approximation. The philosophical name for this view is *compatibilism*. I think compatibilism is a requisite for AI research reaching human-level intelligence.

In practice, regarding an observed system as having choices is necessary when ever a human or robot knows more about the relation of the system to the environment than about what goes on within the system. This is discussed in (McCarthy 1996c).

mind-brain distinctions I'm not sure whether this point is philosophical or scientific. The mind corresponds to software, perhaps with an internal distinction between program and knowledge. Software won't do anything without hardware, but the hardware can be quite simple. In particular, the hardware usually need not have any knowledge. Some hardware configurations can run many different programs concurrently,

i.e. there can be many minds in the same computer body. Software can also interpret other software. The distinction is less in humans than in computers.

3.2 Scientific Presuppositions

Some of the premises of logical AI are scientific in the sense that they are subject to scientific verification. This may also be true of some of the premises listed above as philosophical.

innate knowledge The human brain has important innate knowledge, e.g. that the world includes three dimensional objects that usually persist even when not observed. This was learned by evolution. Acquiring such knowledge by learning from sense data will be quite hard. It is better to build it into AI systems.

Different animals have different innate knowledge. Dogs can know about permanent objects and will look for them when they disappear behind an obstacle. Quite possibly, cockroaches don't know about objects and react only to what they sense..

Identifying human innate knowledge has been the subject of recent psychological research. See (Spelke 1994) and the discussion in (Pinker 1997) and the references Pinker gives. In particular, babies and dogs know innately that there are permanent objects and look for them when they go out of sight. We'd better build that in to our robots.

middle out Humans deal with middle-sized objects and develop our knowledge up and down from the middle. Formal theories of the world must also start from the middle where our experience informs us. Efforts to start from the most basic concepts, e.g. to make a basic ontology are unlikely to succeed as well as starting in the middle. The ontology must be compatible with the idea that the basic entities in the commonsense ontology are not the basic entities in the world. More basic entities in the world, e.g. atoms and quarks, are known less well and are less observable than the middle entities.

universality of intelligence Achieving goals in the world requires that an agent with limited knowledge, computational ability and ability to observe use certain methods. This is independent of whether the agent

is human, Martian or machine. For example, playing chess-like games effectively requires something like alpha-beta pruning.

universal expressiveness of logic This is a proposition analogous to the Turing thesis that Turing machines are computationally universal—anything that can be computed by any machine can be computed by a Turing machine. The *expressiveness thesis* is that anything that can be expressed, can be expressed in first order logic. Some elaboration of the idea is required before it will be as clear as the Turing thesis.[4]

sufficient complexity yields essentially unique interpretations A robot that interacts with the world in a sufficiently complex way gives rise to an essentially unique interpretation of the part of the world with which it interacts. This is an empirical, scientific proposition, but many people, especially philosophers (see (Quine 1969), (Putnam 1975), (Dennett 1971), (Dennett 1998)), take its negation for granted. There are often many interpretations in the world of short descriptions, but long descriptions almost always admit at most one.

The most straightforward example is that a simple substitution cipher cryptogram of an English sentence usually has multiple interpretations if the text is less than 21 letters and usually has a unique interpretation if the text is longer than 21 letters. Why 21? It's a measure of the redundancy of English. The redundancy of a person's or a robot's interaction with the world is just as real—though clearly much harder to quantify.

Second argument from cryptanalysis It would be nice if we could learn about the world as Descartes proposed to do by starting with nothing and building our ideas a step at a time, verifying each idea as it was introduced. Sometimes that's possible, but many important ideas are discovered as complex packages of theory and connections to possible observations or experiment. The verification is of the package as a whole, not of each part separately. A really simple example is Barbara Grosz's "The clock on the wall" when there are several clocks and several walls, but just one wall has a clock.

[4]First order logic isn't the best way of expressing all that can be expressed any more than Turing machines are the best way of expressing computations. However, with set theory, as formalized in first order logic, what can be expressed in stronger systems can apparently also be expressed.

It will not be easy to make AI systems that make up and verify complex theories as wholes.

We expect these philosophical and scientific presuppositions to become more important as AI begins to tackle human level intelligence.

Chapter 4

COMMON SENSE—INFORMAL

The main obstacle to getting computer programs with human level intelligence is that we don't understand yet how to give them common sense. Without common sense, no amount of computer power will give human level intelligence. Once programs have common sense, improvements in computer power and algorithm design will be directly applicable to making them more intelligent.

This chapter is an informal summary of various aspects of common sense. Formalisms will be given in later chapters.

4.1 What is common sense?

Common sense is a certain collection of reasoning abilities, perhaps other abilities, and knowledge.

In (McCarthy 1959) I wrote that the computer programs that had been written up to 1958 lacked common sense. Common sense has proved to be a difficult phenomenon to understand, and most programs of 2004 also lack common sense or have only a little. In the 1959 paper, I wrote "We shall therefore say that **a program has common sense if it automatically deduces for itself a sufficiently wide class of immediate consequences of anything it is told and what it already knows.**"[1]

[1]At least much that people consider obvious is not deduced.

Programs with common sense à la (McCarthy 1959) are still lacking, and, moreover, the ideas of that paper are not enough. Logical deduction is insufficient, and nonmonotonic reasoning is required. Common sense knowledge is also required.

Here's what I think is a more up-to-date formulation.

A program has common sense if it has sufficient common sense knowledge of the world and suitable inference methods to infer a sufficiently wide class of immediate consequences of anything it is told and what it already knows.

Requiring some intelligence as part of the idea of common sense gives another formulation.

A program has common sense if it can act effectively in the *common sense informatic situation*, using the available information to achieve its goals.

4.2 The common sense informatic situation

A program that decides what to do has certain information built in, gets other information from its inputs or observations; still other information is generated by reasoning. Thus it is in a certain *informatic situation*. If the information that has to be used has a common sense character, it will be in what we call the *common sense informatic situation*.

We need to contrast the *common sense informatic situation* with the less general *bounded informatic situations*.

Formal theories in the physical sciences deal with *bounded informatic situations*. A scientist decides informally in advance what phenomena to take into account. For example, much celestial mechanics is done within the Newtonian gravitational theory and does not take into account possible additional effects such as outgassing from a comet or electromagnetic forces exerted by the solar wind. If more phenomena are to be considered, scientists must make new theories—and of course they do.

Likewise present AI formalisms work only in a bounded informatic situations. What phenomena to take into account is decided by a person before the formal theory is constructed. With such restrictions, much of the reasoning can be monotonic, but such systems cannot reach human level ability. For that, the machine will have to decide for itself what information is relevant, and that reasoning will inevitably be partly nonmonotonic.

One example is the "blocks world" where the position of a block x is entirely characterized by a sentence $At(x, l)$ or $On(x, y)$, where l is a location or y is another block. The language does not permit saying that one block is partly on another.

Another example is the MYCIN (Davis et al. 1977) expert system in which the ontology (objects considered) includes diseases, symptoms, and drugs, but not patients (there is only one), doctors or events occurring in time. Thus MYCIN cannot be told that the previous patient with the same symptoms died. See (McCarthy 1983) for more comment.

Systems in a bounded informatic situation are redesigned from the outside when the set of phenomena they take into account is inadequate. However, there is no-one to redesign a human from the outside, so a human has to be able to take new phenomena into account. A human-level AI system needs the same ability to take new phenomena into account.

In general a thinking human is in what we call the *common sense informatic situation*. The known facts are necessarily incomplete. [2] There is no *a priori* limitation on what facts are relevant. It may not even be clear in advance what phenomena should be taken into account. The consequences of actions cannot be fully determined.

An astrophysicist can learn that computing the orbit of a comet during its passage near the sun requires taking into account the forces resulting from gases boiled of the comet by the sun's heat. He can either deal with this informally, regarding the orbit as somewhat uncertain, or he can make a new mathematical theory taking outgassing into account. In either case, he remains outside the theory. Not only that, he can look at his own thinking and say, "How dumb I was not to have noticed ...."

Should it be needed, a human manipulating blocks can discover or be told that it is necessary to deal with configurations in which one block is supported by two others.

[2] We live in a world of middle-sized objects which can only be partly observed. We only partly know how the objects that can be observed are built from elementary particles in general, and our information is even more incomplete about the structure of particular objects. These limitations apply to any buildable machines, so the problem is not just one of human limitations.

Science fiction and scientific and philosophical speculation have often indulged in the *Laplacian fantasy* of super-beings able to predict the future by knowing the positions and velocities of all the particles. That isn't the direction to go. Rather the super-beings would be better at using the information that is available to the senses—maybe having more and more sensitive senses.

MYCIN's lack of common sense is yet more blatant, because its language cannot represent facts about events occurring in time.

The common sense informatic situation has the following features.

1. The theory used by the agent is open to new facts and new phenomena.
2. The objects and other entities under consideration are incompletely known and are not fully characterized by what is known about them.
3. Most of the entities considered are intrinsically not fully defined.
4. In general the informatic situation itself is an object about which facts are known. This human capability is not used in most human reasoning, and very likely animals don't have it.

The difficulties imposed by these requirements are the reason why the goal of Leibniz, Boole and Frege to use logical calculation as the main way of deciding questions in human affairs has not yet been realized. Realizing their goal will require extensions to logic beyond those required to reason in bounded informatic situations. Computer programs operating in the common sense informatic situation also need tools beyond those that have been used so far.

Here are some of the characteristics of such systems and some of the tools.

elaboration tolerance The languages used require *elaboration tolerance*. It must be possible to add facts without scrapping the theory and starting over. Elaboration tolerance seems to impose requirements on the languages used, i.e. on the set of predicate and function symbols.

nonmonotonic reasoning Elaboration tolerance imposes one requirement on the logic, and this is the ability to do *nonmonotonic reasoning*. The system must reach conclusions that further facts not contradicting the original facts are used to correct.

Taking into account only some of the phenomena is a nonmonotonic reasoning step. It doesn't matter whether phenomena not taken into account are intentionally left out or if they are unknown to the reasoner.

While nonmonotonic reasoning is essential for both man and machine, it can lead to error when an important fact is not taken into account. These are the errors most often noticed.

[3]

approximate concepts and approximate objects The *common sense informatic situation* necessitates the use of *approximate concepts* that cannot be fully defined and the use of *approximate theories* involving them.

reasoning in contexts and about contexts In bounded theories, the context is fixed at the time the theory is created. Therefore, the reasoner doesn't have to switch contexts. For example, the theorists undertakes to decide on a consistent notation. There are exceptions to this in computer programming wherein information is encapsulated in classes, but the external behavior of the classes is prescribed by the designer.

An agent in the common sense informatic situation is often confronted with new contexts.

approximate objects

composition of objects Consider an object composed of parts. It is convenient logically when what we knew about the parts and how they are put together enables us to determine the behavior of the compound

[3]Here's an extended example from the history of science.

Starting in the middle of the 19th century, Lord Kelvin (William Thomson) undertook to set limits on the age of the earth. He had measurements of the rate of increase of temperature with depth and of the thermal conductivity of rock. He started with the assumption that the earth was originally molten and computed how long it would have taken for the earth to cool to its present temperature. He also took into account gravitational contraction as a source of energy. He obtained numbers like 25 million years. This put him into conflict with geologists who already had greater estimates based on counting annual layers in sedimentary rock.

Kelvin's calculations were correct but gave the wrong answer, because no-one until Becquerel's discovery in 1896 knew about radioactive decay. Radioactive decay is the main source of energy that keeps the earth hot.

Kelvin's reasoning was nonmonotonic. Namely, he took into account all the sources of energy the science of the day knew about.

Nonmonotonic reasoning is necessary in science as in daily life. There can always be phenomena we don't know about. Indeed there might be another source of energy in the earth besides radioactivity.

Experience tells us that careful nonmonotonic reasoning, taking into account all the sources of information we can find and understand, usually gives good results, but we can never be as certain as we can be of purely mathematical results.

object. Indeed this is often true in science and engineering and is often the goal of the search for a scientific theory. The common sense informatic situation is not so convenient logically. The properties of an object are often more readily available than the properties of the parts and their relations. Formalizations of facts about the relation between structured objects and their parts must not require that all facts about the objects be inferred from known facts about the parts.

A person knows that a certain object is composed of parts. He knows something about the structural relations and about the parts. Physically, the parts and their relations make up the object. If we knew all about these, we would know the object and its potential behavior. However, actual knowledge often runs the other way. We know more about the object than about its parts.

For example, a baseball has a visible and feelable surface, and we can see and feel the seams and can feel its compliance and its simplest heat transfer properties. We also know, from reading or from seeing a baseball disassembled, something about its innards. However, this knowledge of structure is less usable than the knowledge of the baseball as a whole.

It would be logically simpler if we knew about the structures of the objects in our environment and could establish the properties of the whole object from its structure. Thus it is quite pleasing that the properties of molecules follow from the properties of atoms and their interactions. Unfortunately, the common sense world is informatically more complex. We learn about complex objects and go from there to their structures and can only partly discover the structures.

This is not any kind of grand complementarity of the kind Bohr and his followers mistakenly tried to extend from quantum mechanics where it is one usable perspective. Moreover, this limitation on knowledge will apply to robots in a similar way as to humans.

This phenomenon, of often knowing more about the whole than about the parts, applies to more than physical objects. It can apply to processes. The phenomenon even existed in mathematics. Euclid's geometry was a powerful logical structure, but the basic concepts were fuzzy.

Formalization of facts about the relation between structured objects

and their parts must not require that all facts about the objects be inferred from known facts about the parts.

knowledge of physical objects

knowledge of regions in space

knowledge of other actors

introspective knowledge

bounded informatic situations in contexts Bounded informatic situations have an important relation to the common sense informatic situation. For example, suppose there are some blocks on a table. They are not perfect cubes and they are not precisely aligned. Nevertheless, a simple blocks world theory may be useful for planning building a tower by moving and painting blocks. The bounded theory of the simple blocks world in which the blocks are related only by the $on(x, y, s)$ relation is related to the common sense informatic situation faced by the tower builder. This relation is conveniently expressed using the theory of contexts as objects discussed in Chapter . The blocks world theory holds in a subcontext *cblocks* of the common sense theory c, and sentences can be *lifted* in either direction between c and *cblocks*.

4.3 Localization

Maybe this section should be moved closer to discussions of nonmonotonic reasoning.

We do not expect events on the moon to influence the physical location of objects on the table. However, we can provide for the possibility that an astronomer looking through a telescope might be so startled by seeing a meteorite collide with the moon that he would fall off his chair and knock an object off the table. Distant causality is a special phenomenon. We take it into account only when we have a specific reason.

Closer to hand, we do not expect objects not touching or connected through intermediate objects to affect each other. Perhaps there is a lot of common sense knowledge of the physical motion of table scale ojects and how they affect each other that needs to be expressed as a logical theory.

4.3.1 The objects that are present

In Section **??** on circumscription, we discussed what objects can fly. We have

$$\begin{array}{l} \neg Ab\ Aspect1\ x \rightarrow \neg flies\ x, \\ Bird\ x \rightarrow Ab\ Aspect1\ x, \\ Bird\ x \wedge \neg Ab\ Aspect2\ x \rightarrow flies\ x, \\ Penguin\ x \rightarrow Bird\ x, \\ Penguin\ x \rightarrow Ab\ Aspect2\ x \\ , Penguin\ x \wedge \neg Ab\ Aspect3\ x \rightarrow \neg flies\ x, \text{etc.} \end{array} \tag{4.1}$$

When we do the circumscription we certainly want to vary $flies$. If we leave the predicates *bird* and *penguin* constant and vary only $flies$, we conclude that the objects that fly are precisely the birds that are not penguins.

If we are reasoning within a sufficiently limited context this is ok. However, we can't have it in a general purpose common sense knowledge base, because it excludes bats and airplanes from fliers and ostriches from the non-fliers.

4.4 Remarks

Scientists and philosophers of science have often criticized *common sense* as inferior science. Indeed common sense notions of falling bodies not incorporating the discoveries of Galileo and his successors give wrong answers.[4]

Consider the fact that a glass dropped from a table will hit the floor, perhaps hard enough to shatter. Galileo tells us the falling will take a little less that 1/2 seconds, but we haven't much use for this fact in cautioning ourselves to avoid pushing the glass off the table. The common sense notions of physics are needed by a robot that can function in a home. Very likely if we could compute faster and estimate distances and times quantitatively, we could get substantial advantage from knowledge of elementary mechanics. Since we can't, humans have to make do with qualitative reasoning and practice skills that have some quantitative elements which are present but not explicit.

[4]One service mathematics has rendered to the human race. It has put common sense back where it belongs, on the topmost shelf next to the dusty canister labelled 'discarded nonsense'. —E. T. Bell. What did Bell mean by common sense?

A person or program has common sense if it can act effectively in the *common sense informatic situation*, using the available information to achieve its goals.

Achieving human-level AI faces a problem that is generally ignored in building scientific theories and in philosophy. The theory builder or the philosopher of science discusses theories from the outside. However, for an AI system of human level, there is no outside. All its reasoning, including its metamathematical reasoning must be done in its own language, just as we humans discuss human reasoning in our own languages.

Common sense operates in a world with the following characteristics.

Humans and robots that humans build are middle-sized objects in a physical world that contains other purposeful entities. We have common sense information about this world and our possibilities for action and we need to give our robots this common sense information. This information is partial and will usually be partly wrong. We are, and our robots will be in the ¡em¿ common sense informatic situation¡/em¿.

Distinguish between the facts about the world, which include atomic structure and how general relativity interacts with quantum mechanics and what facts are available for common sense use.

Basic facts about the Common sense informatic situation

1. The world in which common sense operates has the aspects described in the following items.

2. Situations are snapshots of part of the world.

3. Events occur in time creating new situations. Agents' actions are events.

4. Agents have purposes they attempt to realize.

5. Processes are structures of events and situations.

6. 3-dimensional space and objects occupy regions. Embodied agents, e.g. people and physical robots are objects. Objects can move, have mass, can come apart or combine to make larger objects.

7. Knowledge of the above can only be approximate.

8. The csis includes mathematics and physics, i.e. abstract structures and their correspondence with structures in the real world.

9. Common sense can come to include facts discovered by science. Examples are conservation of mass and conservation of volume of a liquid.

10. Scientific information and theories are imbedded in common sense information, and common sense is needed to use science.

The common sense informatic situation includes at least the following.

1. The facts that may have to be used to decide what to do are not limited to an initially given set. New facts may have to be obtained.

2. The set of concepts required to act intelligently in a situation may have to be extended.

3. The ability to take into account new facts without complete reorganization of the knowledge base is called ¡EM¿elaboration tolerance.¡/EM¿ The need for elaboration tolerance affects the logical formalisms.

4. Introspection about the methods one has been using may be required. Indeed the complete mental state up to the present moment sometimes has to be regarded as an object that can be reasoned about.

4.4.1 Refinement

Here are some truisms. They are all obvious, but formalisms must take them into account, and how to do this is often not obvious.

Consider a person's knowledge of his body.

He can see his arms, legs and other external parts. He cannot see his liver or his brain, but this is an accident of the senses humans happen to have. If we had ultrasonic detection abilities of bats, we might be able to see our internal organs. Indeed with artificial aids we can see them.

The organs are made up of tissues. Common sense knows little of them, but medical science knows a lot.

The tissues are made of cells. A lot is known about this.

Cells have a structure above the level of molecules, but not much is known about this.

Once we get down to the domain of molecules we are in chemistry which is based on atomic physics.

Below that is elementary particle physics. Whether that is the bottom is unknown.

Objects depend for their integrity and for their perceivable qualities on certain processes going on within them and between them and their neighbors. All material objects depend on the motion of their molecules for their temperatures, and their structural integrity often depends on temperature.

Human common sense can deal with all these phenomena. Some objects we can perceive well, others partly and others not at all. We are aware of objects as structured from parts.

Chapter 5

MATHEMATICAL LOGIC

5.1 Monotonic Mathematical Logic

A logical robot decides what to do by reasoning logically that a certain action will advance its goals. As we have emphasized, some of the reasoning will be nonmonotonic. However, conventional monotonic logic will do most of the reasoning and also provides the framework for developing nonmonotonic reasoning. Therefore, the basis of logical AI is conventional mathematical logic.

Mathematical logicians have developed elegant notations for logic, including the parts used in AI—mainly first order logic and set theory. They have optimized readability for people, having no more notation than necessary and convenient for informal proofs of metatheorems, e.g. completeness, etc. Computer scientists have also developed a wide variety of logical notations and have made many systems. In many of them, (semantic nets, Horn logic,) expressive power has been sacrificed to make the machine computation easier.

Logical AI needs different emphases from those appropriate for research in mathematical logic and also from those appropriate for most applications of logic in computer engineering, e.g. database languages, designing and verifying digital circuits.

1. A representation that makes programs that manipulate logical expressions easy to write. Lisp lists are good for this. They are also reasonably readable as is. If wanted, their readability and writability can be enhanced by using translators into a more optimally readable notation

on the screen or on paper. Less thought has gone into ease of writing than into ease of reading, but computers can help us here also.[1]

2. A rich vocabulary. Russell and Whitehead's *Principia Mathematica* had a richer vocabulary than later systems, because their objective was proving theorems within the system. Later this was replaced by the metamathematical objective of proving theorems about the system. Since a logical robot must do its reasoning within the system we install in it, that system must be rich in expressiveness. It is not good enough to give it a facility for making definitions if each defined concept must have its definition expanded before it is used.

3. Heavy duty set theory. [see writeup] /u/jmc/f87/heavy also checkerboard.

4. Proper treatment of comprehension. Frege's original comprehension axiom allowed the definition $P(x) \equiv <formula>$, where $<formula>$ does not contain P. This allows Russell's paradox defining $bad(x) \equiv \neg x(x)$ yielding the contradiction $bad(bad) \equiv \neg bad(bad)$. The immediate problem was fixed by Zermelo in his set theory. In the notation of set theory we would like to write $\{x|<formula>\}$ to define the set of x's satisfying $<formula>$, but this leads again to Russell's paradox. Zermelo allowed only the restricted comprehension principle that gives the set $\{x \in A|<formula>\}$, where A is a previously defined set. This proved a bit too weak, and Fraenkel added the stronger but more awkward replacement schema that gives the image of a set by a function formula as a set. This is the system ZF.

[1]The lisp expression (forall (x) (implies (p x) (q x))) is conveniently read in the form $\forall x.p(x) \rightarrow q(x)$ In a context of writing logical expressions involving predicates of one argument, it might be typed "ax.px i qx", thus reducing the amount of typing. In the pre-computer world, the notation chosen has to be a compromise between readability and writability. Once we have computers convenience of describing computations and speed of the programs that carry out the computations with symbolic expression are also considerations. (With computers built since 1960, the speed of translation to and from notations convenient for people is not an issue.)

Read and print programs can permit the most convenient notation for each purpose to be used. This assumes that the user takes the time to learn all three kinds of notation and doesn't forget them. Occasional users might be better off with a single notation. Actually Lisp programmers have chosen to write in Lisp list notation rather than in the more conventional mathematical notations that were offered in some early systems. This surprised me.

A logical robots must use comprehension principle as the main tool for extending its collection of sets. Present logical AI computational systems have mainly confined themselves to first order logic. In particular, they have not used any form of comprehension that permits making a set (or predicate) from an arbitrary property. I suppose this is mainly because it seems difficult to make a system that will invent properties. Inventing properties is a necessary step on the path to human-level AI. My article (McCarthy 1964) discusses the standard but creative solution to the mutilated checkerboard problem which involves introducing the colors of the squares, a property not present in the initial formulation of the problem.

5. In order to get convenient decideability in a subdomain, don't restrict the language. Instead restrict the reasoner to making appropriate inferences. There will be some cost, say a factor of 5, in speed, but the results of the restricted reasoning will then be available for more general reasoning. Reasoning about this requires that facts about the reasoning algorithms be expressible as sentences that can be reasoned with.

5.2 Set theory

The plan is to use Zermelo-Fraenkel set theory, abbreviated ZF, to express what a robot knows. Set theory is usually presented in first order logic, but we will go to higher order logic when it is convenient. In particular, second order logic is convenient for nonmonotonic reasoning. Our set theory will include as basic some functions of sets usually omitted from ZF, because they are definable.

The advantage for AI of set theory over just using first order logic is that the operations that form new sets from old ones are convenient for building new common sense concepts from old ones.

The most important operation is *comprehension* which forms the set of objects having a given property.

$\{x|\phi(x)\}$ is the set of xs having the property ϕ.

ϕ can be an arbitrary property. As we shall see, an application of comprehension is often the way of introducing a creative idea to reasoning. On the other hand, comprehension makes difficulty for theorem proving programs,

because they would have to choose among an arbitrary set of properties. My opinion is that this difficulty must be faced in order to achieve human-level AI.

Here are some examples.

1. Let $father(x, y)$ be the assertion that x is the father of y. We then have

$$\begin{array}{l} sons(x) = \{x|father(x, y)\} \\ number\text{-}sons(x) = card(sons(x)) \\ middle\text{-}sons(x) = \{y|father(x, y) \wedge (\exists zw) father(x, z) \wedge father(x, w) \\ \wedge older(z, y) \wedge older(y, w)\} \end{array} \tag{5.1}$$

2. The transitive closure of a relation $R(x, y)$ is the intersection of all transitive relations R' containing R, i.e. they must satisfy

$$\begin{array}{l} (\forall x\ y)(R(x, y) \rightarrow R'(x, y) \\ \wedge \\ (\forall x\ y\ z)(R'(x, y) \wedge R'(y, z) \rightarrow R'(x, z)). \end{array}$$

   However, this isn't enough to get the intersection.

3. The mutilated checkerboard problem.

   An 8 by 8 checkerboard with two diagonally opposite squares removed cannot be covered by dominoes each of which covers two rectilinearly adjacent squares. We present a set theory description of the proposition and an informal proof that the covering is impossible. While no present system that I know of will accept either the formal description or the proof, I claim that both should be admitted in any *heavy duty set theory.*[2]

   We have the definitions

$$Board = Z8 \times Z8. \tag{5.2}$$

[2]The Mizar proof checker accepts the definitions essentially as they are, but the first proof in Mizar is 400 lines.

In set theory $Z8$ is a standard name for the natural numbers from 0 to 7.

$$mutilated\text{-}board = Board - \{(0,0),(7,7)\}. \tag{5.3}$$

This formula uses three of the tools of set theory. $(0,0)$ and $(7,7)$ are ordered pairs. In ZF proper, the ordered pair is a defined operation. In a heavy duty set theory (HDST), forming an ordered pair will be built in. $\{(0,0),(7,7)\}$ is the two element set consisting of $(0,0)$ and $(7,7)$. Forming a set with given elements is a basic operation. Finally, we have used the minus sign for the operation of set difference. This again is a defined operation in ZF. Indeed we can write $X - Y = \{x | x \in X \wedge \neg(x \in Y)\}$. HDST will also have set difference built in.

$$\begin{array}{l} domino\text{-}on\text{-}board(x) \equiv (x \subset Board) \wedge card(x) = 2 \\ \wedge(\forall x1\ x2)(x = \{x1, x2\} \rightarrow adjacent(x1, x2)) \end{array} \tag{5.4}$$

$$\begin{array}{l} domino\text{-}on\text{-}board(x) \equiv x \subset Board \\ \wedge(\exists x1\ x2)(x = \{x1, x2\} \wedge adjacent(x1, x2)) \end{array} \tag{5.5}$$

and

$$\begin{array}{l} adjacent(x1, x2) \equiv |c(x1,1) - c(x2,1)| = 1 \\ \wedge c(x1,2) = c(x2,2) \\ \vee |c(x1,2) - c(x2,2)| = 1 \wedge c(x1,1) = c(x2,1). \end{array} \tag{5.6}$$

If we are willing to be slightly tricky, we can write more compactly

$$adjacent(x1, x2) \equiv |c(x1,1) - c(x2,1)| + |c(x1,2) - c(x2,2)| = 1, \tag{5.7}$$

but then the proof might not be so obvious to the program.

Next we have.

$$\begin{array}{l} partial\text{-}covering(z) \\ \equiv (\forall x)(x \in z \supset domino\text{-}on\text{-}board(x)) \\ \wedge(\forall x\ y)(x \in z \wedge y \in z \supset x = y \vee x \cap y = \{\}) \end{array} \tag{5.8}$$

Theorem:

$$\neg(\exists z)(\textit{partial-covering}(z) \wedge \bigcup z = \textit{mutilated-board}) \tag{5.9}$$

Proof:

We define

$$x \in \textit{Board} \supset \textit{color}(x) = \textit{rem}(c(x,1) + c(x,2), 2) \tag{5.10}$$

$$\begin{array}{l} \textit{domino-on-board}(x) \supset \\ (\exists u\ v)(u \in x \wedge v \in x \wedge \textit{color}(u) = 0 \wedge \textit{color}(v) = 1), \end{array} \tag{5.11}$$

$$\begin{array}{l} \textit{partial-covering}(z) \supset \\ \textit{card}(\{u \in \bigcup z | \textit{color}(u) = 0\}) \\ = \textit{card}(\{u \in \bigcup z | \textit{color}(u) = 1\}), \end{array} \tag{5.12}$$

$$\begin{array}{l} \textit{card}(\{u \in \textit{mutilated-board} | \textit{color}(u) = 0\}) \\ \neq \textit{card}(\{u \in \textit{mutilated-board} | \textit{color}(u) = 1\}), \end{array} \tag{5.13}$$

and finally

$$\neg(\exists z)(\textit{partial-covering}(z) \quad \wedge \quad \textit{mutilated-board} = \bigcup z) \tag{5.14}$$

A practical HDST needs a substantial initial theorem base—essentially those theorems a mathematician will use freely, often without even appealing to them formally.

We begin with Dedekind's recursion theorem of 1888. As expressed in (H.-D. Ebbinghaus 1991), the theorem is

Let A be an arbitrary set containing an element $a \in A$, and g a given mapping $g : A \to A$ of A into itself. Then there is one and only one mapping $\varphi : N \to A$ with the two properties $\varphi(0) = a$ and $\varphi \circ S = g \circ \varphi$. Here S is the successor operation on the set N of natural numbers. The proof is long enough so that it should not be repeated every time the theorem is used.

5.2.1 Fregean set theory

Human thinking, even in mathematics, often works by uninhibited speculation aiming at a promising result. If no promising result is obtained by a particular line of speculation, the investment that has to be abandoned is smaller than if the line of reasoning had been carried out rigorously. If the result of the speculation looks good, an effort is made to put the reasoning on a sound footing.

The logic of Frege's (**?**) *Begriffschrift* was inconsistent, as Bertrand Russell pointed out to him in 1901. We present a set theory based on this logic which is also inconsistent, but may be suitable for the exploratory phase of reasoning. [3]

Fregean set theory, call it FR, lets the reasoner speculate by allowing unrestricted comprehension. Thus it allows the term $\{x|\mathcal{E}(x)\}$, where $\mathcal{E}(x)$ is an arbitrary open sentence in x instead of just $\{x \in A|\mathcal{E}(x)\}$ with A a previously defined set. Fregean set theory is inconsistent, because we can form the set $\{x|x \notin x\}$ and get a contradiction by asking whether $\{x|x \notin x\} \in \{x|x \notin x\}$. However, our reasoner need not form that set or any other contradictory set.

Suppose it reaches a promising conclusion p. Because of the possibility that it got there via a contradiction, p is not certain, but it can try to validate the proof by replacing it by a proof in ZFC. This can save computation, because proof attempts that fail in FR, i.e. don't reach a valued conclusion, cost less computation than in ZFC. There will be many more failures than successes.

A lot depends on the theorem proving or problem solving program that

[3] Here's an example of correctable speculative reasoning from chess.

Consider the move h2-h3 (i.e. P-KR3) early in the game. It is usually a pointless move, but the programs all include it because it is sometimes good. Its usefulness is sometimes to drive away a black bishop on g4 and sometimes to prevent the bishop from moving to g4. Another use is to give the white king an escape square. Nevertheless, it is usually a bad move. Deep Blue may examine this move 10^9 times at various points in the move tree in deciding on a particular move, in almost all positions concluding that the move is bad.

Deep Blue, like all chess programs that I know about, scans the move tree once except for iterative deepening. Consider a chess program that ignores h2-h3 but can redo the calculation including the move. It would put the move in the new calculation if the preceding calculation had black profitably putting a bishop on g4 or profitably attacking the white king in a castled position on the king's side.

Chess programs could play with much less computation if they used bold speculation followed by careful checking of good lines.

uses FR. A perverse prover would immediately derive the contradiction and from that derive whatever might be wanted. Such proofs could not be repaired in ZFC.

FR has many fewer axioms than ZFC, because many of the axioms of ZFC assert the existence of sets that would be terms of FR.

Here are the axioms of ZFC and their FR counterparts.

extensionality $(\forall x\ y)(x = y \equiv (\forall z)(z \in x \equiv z \in y))$. This is still needed in FR.

null set Unnecessary: $\{x|\mathbf{false}\}$ is a perfectly good term.

pairing Unnecessary: $\{z|z = x \vee z = y\}$ is a term.

power set Unnecessary: $\{z|z \subset x\}$ is a term.

union Unnecessary: $\{z|(\exists y)(z \in y \wedge y \in x)\}$ is a term.

infinity needed, because the infinite set asserted to exist is not given by a term. I'd like a more ambitious axiom that immediately gives the existence of recursively defined functions and sets rather than requiring somewhat lengthy proofs required starting with the standard axiom.

foundation Needed if infinite descending chains of sets are to be forbidden. I'm presently neutral on this point.

choice seems to be needed (if wanted)

comprehension replaced by the more general set former $\{x|\mathcal{E}\{x\}\}$, where $\mathcal{E}\{x\}$ is an arbitrary open formula involving x and possibly other free variables.

replacement Unnecessary: $\{z|(\exists y)(y \in A \wedge z = f(y))\}$, where f is a function term is the term wanted. We need to be able to form function terms.

Fregean set theory is inconsistent because of Russell's paradox. Namely, consider the set π defined by $\pi = \{x|x \notin x\}$. We have immediately $\pi \in \pi \equiv \pi \notin \pi$.

Suppose we have a proof (or our program has a proof) in FR of some interesting result. We or it can't be sure the result is correct. However, the

proof may be translatable to a proof in ZFC. Occurrences of the Fregean set former may be translatable to the ZFC set formers—comprehension and replacement.

Thus FR has just the axioms of extensionality, infinity and choice and the Fregean set former $\{x|\mathcal{E}\{x\}\}$ are replaced by instances of $\{x \in A|\mathcal{E}\{x\}\}$, where A is a previously defined set. We can expect that the other axioms of ZFC, e.g. the power set axiom, will be required to get A. In some proofs we will have to use the replacement axiom schema.

It could turn out that it is just as convenient to have the program explore ZFC derivations in the first place, but it seems to me that FR is closer to the way mathematicians and other people think.

Since FR is inconsistent, there is no obvious way of giving it any semantics.

5.3 Why set theory?

5.4 Some topics: to be included in revised form

Here are some important topics.

1. Propositional calculus
2. Interpretations, models, soundness and completeness
3. First order logic
4. Second order and higher order logics
5. Set theory, simple and extended
6. Computability and computational complexity
7. Extensions: conditional expressions,
8. Logic of programs
9. Logic programs
10. Partial predicates and functions

11. Theorem provers and problem solvers

5.4.1 Second order logic

Second order formulas have proved convenient for formalizing many kinds of nonmonotonic reasoning. For example, a standard way of writing circumscription is

$$A(P, Z) \wedge (\forall pz)(A(p, z) \rightarrow \neg p < P). \tag{5.15}$$

The quantifier $(\forall pz)$ quantifies over the predicate variable p and the variable z which may also be a predicate variable. In simple cases, it is possible to get the full force of (5.15) by making suitable substitutions for p and z—but not in general.

It is often asserted, e.g. in (Reiter 2001) that the utility of second order logic in AI is limited by the fact that any axiom system for second order logic is necessarily incomplete. I think this is incorrect for formalizing common sense knowledge and reasoning. All the known sentences that are true in second order logic but unprovable are obtained ultimately from Gödel's theorems about the incompleteness of Peano arithmetic. No-one has found a theorem of conventional mathematics that one would want to prove for which the second order axiom systems are inadequate. I'll conjecture that the axiom systems for second order logic are "complete enough" in a sense yet to be determined.[4]

5.5 AI and the frustration theorems of logic

Starting with Gödel in 1930, mathematical logicians discovered a number of theorems that limit what can be formalized and what can be computed. The limitations apply to both people and computers. The theorems were formulated and proved by looking at mathematical formalisms from the outside, i.e. doing metamathematics. Getting around the limitations also requires looking at formalisms from the outside, and human-level AI will also have

[4]There are complications here. Thus the consistency of Peano arithmetic is equivalent to a certain set of Diophantine equations not having solutions in integers. Diophantine equations are quite ordinary mathematics. Somehow a theory of "complete enough" must be able to exclude some kinds of Diophantine equations.

to be able to do the kind of metamathematics involved in reasoning about these limitations.

In this section we will discuss

1. Russell's paradox (1900) and what it did to Cantor's set theory.
2. Gödel's theorems of 1930.
3. Turing's undecidability results of 1936
4. The theory of NP-completeness 1970?
5. Montague's theorems limiting syntactic treatments of modality.
6. The independence results of set theory

We propose three tools by which AI systems can work around these limitations—all suggested by what people do.

1. Treat whole theories as objects in metamathematical reasoning.
2. Use unjustifiably strong reasoning methods to get results that are subject to amplification in normal systems. For example, we propose "Fregean set theory" with unrestricted comprehension for deriving results with the derivations of interesting statements later to be translated into ZFC proofs.

nonmonotonic

Gödel's incompleteness theorem, Turing's proof of the unsolvability of the halting problem and Cook and Karp's discovery of NP-completeness put limits on what humans and machines can do. In spite of these limits, humans manage to be intelligent, and we can make intelligent machines. There is plenty of room for maneuver among the limitations.

Gödel's first incompleteness theorem shows that any consistent logical theory expressive enough for elementary arithmetic, i.e. with addition, multiplication and quantifiers could express true sentences unprovable in the theory.

Gödel's second incompleteness theorem tells that the consistency of the system is one of these unprovable sentences.

The basis of Gödel's proof was the fact that the syntactic computations involved in combining formulas and verifying that a sequence of formulas is a proof can be imitated by arithmetic computations on "Gödel numbers" of formulas. If we have axioms for symbolic computations, e.g. for Lisp computations, then the proofs of Gödel's theorems become much shorter. Shankar (Shankar 1986) has demonstrated this using the Boyer-Moore prover.

Among the unprovable true sentences is the statement of the theory's own consistency. We can interpret this as saying that the theory lacks self-confidence. Turing, in his PhD thesis, studied what happens if we add to a theory T the statement $consis(T)$ asserting that T is consistent, getting a stronger theory T'. While the new theory has $consis(T)$ as a theorem, it doesn't have $consis(T')$ as a theorem—provided it is consistent. The process can be iterated, and the union of all these theories is $consis^{\omega}(T)$. Indeed the process can again be iterated, as Turing showed, to any constructive ordinal number.

Solomon Feferman (Feferman 1962) introduced a more powerful iteration principle than Turing's. Feferman's principle, like Turing's represents expressions of confidence in what has already been done. The theory of self-confidence principles gets extremely technical at this point. The technicalities don't seem directly relevant to AI research, because AI programs don't need these high levels of self-confidence. However, they are relevant to arguments like those advanced by Penrose about what is possible in principle for computers. See Chapter 16.2 for more.

The discovery of NP-completeness and the subsequent development of the theory of computational complexity was the largest single contribution that computer science has made to mathematics. The theory has definite but limited application to logical AI.

Consider a problem that depends on a number n, e.g. inverting an $n \times n$ matrix or determining whether a propositional expression with of length n is satisfiable. The first computation can be done with a number of multiplications proportional to n^3 (the actual exponent being somewhat smaller). The class of problems for which the number of operations is polynomial in the size of the problem, n in this case, is called P.

The known algorithms for propositional satisfiability all take a number of operations exponential in n, but no-one has shown that there isn't an algorithm that does it in a number of operations polynomial in n.

A proposed assignment of truth values to a propositional expression can be checked in polynomial time. Moreover, if we imagine an algorithm that

could go both ways every time it came to two conditions that had to be verified, such an algorithm could test for satisfiability in polynomial time. Algorithms that can go both ways are call non-deterministic, and problems that can be solved in polynomial time by a non-determinstc algorithm are form a class called NP. Solution by a non-deterministic polynomial time algorithm amounts to proposed solutions being checkable in polynomial time.

Certain problems in NP, including propositional satisfiability, have been shown to be *NP-complete*. Let A be such a problem. This means that any problem in NP can be put into the form of an A problem in a polynomial number of operations. Thus any NP problem can be reduced to a satisfiability problem in a polynomial number of operations.

Maybe P = NP. This is a famous unsolved problem. Most computer scientists believe that NP is not the same as P, and that NP-complete problems require more than polynomial time in general.

There is a tendency to informally identify polynomial time with feasibility and NP-completeness with infeasibility. This is shakey in both directions. The power or the coefficients might be too large for feasibility in a polynomial time algorithm. Problems of interest sometimes have only small enough numbers of variables even though the problem is NP-complete.

5.5.1 What has computational complexity to do with AI?

Many problems that people solve and that we want computers to solve belong to classes that are NP-complete. Some researchers have jumped to the conclusion that problems people solve require exponential time when done by computers. What this misses is that the problems people actually need to solve may belong to polynomial time subclasses of the NP-complete class.

One result that is definitely relevant to AI is the discovery that satisfiability problems are easy when there are many satisfying assignments and also easy when the sentences are very incompatibility. The hard problems are those on the border. The relevance to AI is that most of the satisfiability problems that have come from common sense reasoning have been easy and readily solved by the programs.

One problem for AI research is to identify more such subclasses of easy problems.

Chapter 6

Nonmonotonic Reasoning

Humans and intelligent computer programs must often jump to the conclusion that the objects they can determine to have certain properties or relations are the only objects that do. *Circumscription* formalizes such conjectural reasoning.

6.1 INTRODUCTION. THE QUALIFICATION PROBLEM

(McCarthy 1959) proposed a program with "common sense" that would represent what it knows (mainly) by sentences in a suitable logical language. It would decide what to do by deducing a conclusion that it should perform a certain act. Performing the act would create a new situation, and it would again decide what to do. This requires representing both knowledge about the particular situation and general common sense knowledge as sentences of logic.

The "qualification problem", immediately arose in representing general common sense knowledge. It seemed that in order to fully represent the conditions for the successful performance of an action, an impractical and implausible number of qualifications would have to be included in the sentences expressing them. For example, the successful use of a boat to cross a river requires, if the boat is a rowboat, that the oars and rowlocks be present and unbroken, and that they fit each other. Many other qualifications can be added, making the rules for using a rowboat almost impossible to apply, and yet anyone will still be able to think of additional requirements not yet

stated.

Circumscription is a rule of conjecture that can be used by a person or program for "jumping to certain conclusions". Namely, *the objects that can be shown to have a certain property P by reasoning from certain facts A are all the objects that satisfy P*. More generally, circumscription can be used to conjecture that the tuples $< x, y..., z >$ that can be shown to satisfy a relation $P(x, y, ..., z)$ are all the tuples satisfying this relation. Thus we *circumscribe* the set of relevant tuples.

We can postulate that a boat can be used to cross a river unless "something" prevents it. Then circumscription may be used to conjecture that the only entities that can prevent the use of the boat are those whose existence follows from the facts at hand. If no lack of oars or other circumstance preventing boat use is deducible, then the boat is concluded to be usable. The correctness of this conclusion depends on our having "taken into account" all relevant facts when we made the circumscription.

Circumscription formalizes several processes of human informal reasoning. For example, common sense reasoning is ordinarily ready to jump to the conclusion that a tool can be used for its intended purpose unless something prevents its use. Considered purely extensionally, such a statement conveys no information; it seems merely to assert that a tool can be used for its intended purpose unless it can't. Heuristically, the statement is not just a tautologous disjunction; it suggests forming a plan to use the tool.

Even when a program does not reach its conclusions by manipulating sentences in a formal language, we can often profitably analyze its behavior by considering it to *believe* certain sentences when it is in certain states, and we can study how these *ascribed beliefs* change with time. See (McCarthy 1979a). When we do such analyses, we again discover that successful people and programs must jump to such conclusions.

6.2 THE NEED FOR NONMONOTONIC REASONING

We cannot get circumscriptive reasoning capability by adding sentences to an axiomatization or by adding an ordinary rule of inference to mathematical logic. This is because the well known systems of mathematical logic have the following *monotonicity property*. If a sentence q follows from a collection A

of sentences and $A \subset B$, then q follows from B. In the notation of proof theory: if $A \vdash q$ and $A \subset B$, then $B \vdash q$. Indeed a proof from the premisses A is a sequence of sentences each of which is a either a premiss, an axiom or follows from a subset of the sentences occurring earlier in the proof by one of the rules of inference. Therefore, a proof from A can also serve as a proof from B. The semantic notion of entailment is also monotonic; we say that A entails q (written $A \models q$) if q is true in all models of A. But if $A \models q$ and $A \subset B$, then every model of B is also a model of A, which shows that $B \models q$.

Circumscription is a formalized *rule of conjecture* that can be used along with the *rules of inference* of first order logic. *Predicate circumscription* assumes that entities satisfy a given predicate only if they have to on the basis of a collection of facts. *Domain circumscription* conjectures that the "known" entities are all there are. It turns out that domain circumscription, previously called *minimal inference*, can be subsumed under predicate circumscription.

We will argue using examples that humans use such "nonmonotonic" reasoning and that it is required for intelligent behavior. The default case reasoning of many computer programs (Reiter 1980) and the use of THNOT in MICROPLANNER (Sussman, et. al. 1971) programs are also examples of nonmonotonic reasoning, but possibly of a different kind from those discussed in this paper. (Hewitt 1972) gives the basic ideas of the PLANNER approach.

The result of applying circumscription to a collection A of facts is a sentence schema that asserts that the only tuples satisfying a predicate $P(x, ..., z)$ are those whose doing so follows from the sentences of A. Since adding more sentences to A might make P applicable to more tuples, circumscription is not monotonic. Conclusions derived from circumscription are conjectures that A includes all the relevant facts and that the objects whose existence follows from A are all the relevant objects.

A heuristic program might use circumscription in various ways. Suppose it circumscribes some facts and makes a plan on the basis of the conclusions reached. It might immediately carry out the plan, or be more cautious and look for additional facts that might require modifying it.

Before introducing the formalism, we informally discuss a well known problem whose solution seems to involve such nonmonotonic reasoning.

6.3 MISSIONARIES AND CANNIBALS

The *Missionaries and Cannibals* puzzle, much used in AI, contains more than enough detail to illustrate many of the issues. "Three missionaries and three cannibals come to a river. A rowboat that seats two is available. If the cannibals ever outnumber the missionaries on either bank of the river, the missionaries will be eaten. How shall they cross the river?"

Obviously the puzzler is expected to devise a strategy of rowing the boat back and forth that gets them all across and avoids the disaster.

Amarel (1971) considered several representations of the problem and discussed criteria whereby the following representation is preferred for purposes of AI, because it leads to the smallest state space that must be explored to find the solution. A state is a triple comprising the numbers of missionaries, cannibals and boats on the starting bank of the river. The initial state is 331, the desired final state is 000, and one solution is given by the sequence (331,220,321,300,311,110,221,020,031,010,021,000).

We are not presently concerned with the heuristics of the problem but rather with the correctness of the reasoning that goes from the English statement of the problem to Amarel's state space representation. A generally intelligent computer program should be able to carry out this reasoning. Of course, there are the well known difficulties in making computers understand English, but suppose the English sentences describing the problem have already been rather directly translated into first order logic. The correctness of Amarel's representation is not an ordinary logical consequence of these sentences for two further reasons.

First, nothing has been stated about the properties of boats or even the fact that rowing across the river doesn't change the numbers of missionaries or cannibals or the capacity of the boat. Indeed it hasn't been stated that situations change as a result of action. These facts follow from common sense knowledge, so let us imagine that common sense knowledge, or at least the relevant part of it, is also expressed in first order logic.

The second reason we can't *deduce* the propriety of Amarel's representation is deeper. Imagine giving someone the problem, and after he puzzles for a while, he suggests going upstream half a mile and crossing on a bridge. "What bridge", you say. "No bridge is mentioned in the statement of the problem." And this dunce replies, "Well, they don't say there isn't a bridge". You look at the English and even at the translation of the English into first

order logic, and you must admit that "they don't say" there is no bridge. So you modify the problem to exclude bridges and pose it again, and the dunce proposes a helicopter, and after you exclude that, he proposes a winged horse or that the others hang onto the outside of the boat while two row.

You now see that while a dunce, he is an inventive dunce. Despairing of getting him to accept the problem in the proper puzzler's spirit, you tell him the solution. To your further annoyance, he attacks your solution on the grounds that the boat might have a leak or lack oars. After you rectify that omission from the statement of the problem, he suggests that a sea monster may swim up the river and may swallow the boat. Again you are frustrated, and you look for a mode of reasoning that will settle his hash once and for all.

In spite of our irritation with the dunce, it would be cheating to put into the statement of the problem that there is no other way to cross the river than using the boat and that nothing can go wrong with the boat. A human doesn't need such an ad hoc narrowing of the problem, and indeed the only watertight way to do it might amount to specifying the Amarel representation in English. Rather we want to avoid the excessive qualification and get the Amarel representation by common sense reasoning as humans ordinarily do.

Circumscription is one candidate for accomplishing this. It will allow us to conjecture that no relevant objects exist in certain categories except those whose existence follows from the statement of the problem and common sense knowledge. When we *circumscribe* the first order logic statement of the problem together with the common sense facts about boats etc., we will be able to conclude that there is no bridge or helicopter. "Aha", you say, "but there won't be any oars either". No, we get out of that as follows: It is a part of common knowledge that a boat can be used to cross a river *unless there is something wrong with it or something else prevents using it*, and if our facts don't require that there be something that prevents crossing the river, circumscription will generate the conjecture that there isn't. The price is introducing as entities in our language the "somethings" that may prevent the use of the boat.

If the statement of the problem were extended to mention a bridge, then the circumscription of the problem statement would no longer permit showing the non-existence of a bridge, i.e. a conclusion that can be drawn from a smaller collection of facts can no longer be drawn from a larger. This nonmonotonic character of circumscription is just what we want for this kind of problem. The statement, *"There is a bridge a mile upstream, and the*

boat has a leak." doesn't contradict the text of the problem, but its addition invalidates the Amarel representation.

In the usual sort of puzzle, there is a convention that there are no additional objects beyond those mentioned in the puzzle or whose existence is deducible from the puzzle and common sense knowledge. The convention can be explicated as applying circumscription to the puzzle statement and a certain part of common sense knowledge. However, if one really were sitting by a river bank and these six people came by and posed their problem, one wouldn't take the circumscription for granted, but one *would* consider the result of circumscription as a hypothesis. In puzzles, circumscription seems to be a rule of inference, while in life it is a rule of conjecture.

Some have suggested that the difficulties might be avoided by introducing probabilities. They suggest that the existence of a bridge is improbable. The whole situation involving cannibals with the postulated properties cannot be regarded as having a probability, so it is hard to take seriously the conditional probability of a bridge given the hypotheses. More to the point, we mentally propose to ourselves the normal non-bridge non-sea-monster interpretation *before* considering these extraneous possibilities, let alone their probabilities, i.e. we usually don't even introduce the sample space in which these possibilities are assigned whatever probabilities one might consider them to have. Therefore, regardless of our knowledge of probabilities, we need a way of formulating the normal situation from the statement of the facts, and nonmonotonic reasoning seems to be required. The same considerations seem to apply to fuzzy logic.

Using circumscription requires that common sense knowledge be expressed in a form that says a boat can be used to cross rivers unless there is something that prevents its use. In particular, it looks like we must introduce into our *ontology* (the things that exist) a category that includes *something wrong with a boat* or a category that includes *something that may prevent its use*. Incidentally, once we have decided to admit *something wrong with the boat*, we are inclined to admit a *lack of oars* as such a something and to ask questions like, *"Is a lack of oars all that is wrong with the boat?*".

Some philosophers and scientists may be reluctant to introduce such *things*, but since ordinary language allows *"something wrong with the boat"* we shouldn't be hasty in excluding it. Making a suitable formalism is likely to be technically difficult as well as philosophically problematical, but we must try.

We challenge anyone who thinks he can avoid such entities to express in

his favorite formalism, *"Besides leakiness, there is something else wrong with the boat"*. A good solution would avoid counterfactuals as this one does.

Circumscription may help understand natural language, because if the use of natural language involves something like circumscription, it is understandable that the expression of general common sense facts in natural language will be difficult without some form of nonmonotonic reasoning.

6.4 THE FORMALISM OF CIRCUMSCRIPTION

Let A be a sentence of first order logic containing a predicate symbol $P(x_1, \ldots, x_n)$ which we will write $P(\bar{x})$. We write $A(\Phi)$ for the result of replacing all occurrences of P in A by the predicate expression Φ. (As well as predicate symbols, suitable λ-expressions are allowed as predicate expressions).

Definition. *The circumscription of P in $A(P)$* is the sentence schema

$$A(\Phi) \wedge \forall \bar{x}.(\Phi(\bar{x}) \supset P(\bar{x})) \supset \forall \bar{x}.(P(\bar{x}) \supset \Phi(\bar{x})). \tag{6.1}$$

(6.33) can be regarded as asserting that the only tuples $(\bar{x})$ that satisfy P are those that have to — assuming the sentence A. Namely, (6.33) contains a predicate parameter Φ for which we may subsitute an arbitrary predicate expression. (If we were using second order logic, there would be a quantifier $\forall\Phi$ in front of (6.33).) Since (6.33) is an implication, we can assume both conjuncts on the left, and (6.33) lets us conclude the sentence on the right. The first conjunct $A(\Phi)$ expresses the assumption that Φ satisfies the conditions satisfied by P, and the second $\forall \bar{x}.(\Phi(\bar{x}) \supset P(\bar{x}))$ expresses the assumption that the entities satisfying Φ are a subset of those that satisfy P. The conclusion asserts the converse of the second conjunct which tells us that in this case, Φ and P must coincide.

We write $A \vdash_P q$ if the sentence q can be obtained by deduction from the result of circumscribing P in A. As we shall see $\vdash_P$ is a nonmonotonic form of inference, which we shall call *circumscriptive inference*.

A slight generalization allows circumscribing several predicates jointly; thus jointly circumscribing P and Q in $A(P, Q)$ leads to

$$\begin{aligned} &A(\Phi, \Psi) \wedge \forall \bar{x}.(\Phi(\bar{x}) \supset P(\bar{x})) \wedge \forall \bar{y}.(\Psi(\bar{y}) \supset Q(\bar{y})) \\ &\supset \forall \bar{x}.(P(\bar{x}) \supset \Phi(\bar{x})) \wedge \forall \bar{y}.(Q(\bar{y}) \supset \Psi(\bar{y})) \end{aligned}$$

in which we can simultaneously substitute for Φ and Ψ. The relation $A \vdash_{P,Q} q$ is defined in a corresponding way. Although we don't give examples of joint circumscription in this paper, we believe it will be important in some AI applications.

Consider the following examples:

Example 1. In the blocks world, the sentence A may be

$$isblock\ A \wedge isblock\ B \wedge isblock\ C \tag{6.2}$$

asserting that A, B and C are blocks. Circumscribing *isblock* in (6.47) gives the schema

$$\Phi(A) \wedge \Phi(B) \wedge \Phi(C) \wedge \forall x.(\Phi(x) \supset isblock\ x) \supset \forall x.(isblock\ x \supset \Phi(x)). \tag{6.3}$$

If we now substitute

$$\Phi(x) \equiv (x = A \vee x = B \vee x = C) \tag{6.4}$$

into (6.3) and use (6.47), the left side of the implication is seen to be true, and this gives

$$\forall x.(isblock\ x \supset (x = A \vee x = B \vee x = C)), \tag{6.5}$$

which asserts that the only blocks are A, B and C, i.e. just those objects that (6.47) requires to be blocks. This example is rather trivial, because (6.47) provides no way of generating new blocks from old ones. However, it shows that circumscriptive inference is nonmonotonic since if we adjoin *isblock* D to (6.47), we will no longer be able to infer (6.35).

Example 2. Circumscribing the disjunction

$$isblock\ A \vee isblock\ B \tag{6.6}$$

leads to

$$(\Phi(A) \vee \Phi(B)) \wedge \forall x.(\Phi(x) \supset isblockx) \supset \forall x.(isblock\ x \supset \Phi(x)). \tag{6.7}$$

We may then substitute successively $\Phi(x) \equiv (x = A)$ and $\Phi(x) \equiv (x = B)$, and these give respectively

$$(A = A \vee A = B) \wedge \forall x.(x = A \supset isblock\ x) \supset \forall x.(isblock\ x \supset x = A), \tag{6.8}$$

which simplifies to

$$isblock\ A \supset \forall x.(isblock\ x \supset x = A) \tag{6.9}$$

and

$$(B = A \vee B = B) \wedge \forall x.(x = B \supset isblock\ x) \supset \forall x.(isblock\ x \supset x = B), \tag{6.10}$$

which simplifies to

$$isblock\ B \supset \forall x.(isblock\ x \supset x = B). \tag{6.11}$$

(6.9), (6.11) and (6.6) yield

$$\forall x.(isblock\ x \supset x = A) \vee \forall x.(isblock\ x \supset x = B), \tag{6.12}$$

which asserts that either A is the only block or B is the only block.

Example 3. Consider the following algebraic axioms for natural numbers, i.e., non-negative integers, appropriate when we aren't supposing that natural numbers are the only objects.

$$isnatnum\ 0 \wedge \forall x.(isnatnum\ x \supset isnatnum\ succ\ x). \tag{6.13}$$

Circumscribing *isnatnum* in (6.36) yields

$$\Phi(0) \wedge \forall x.(\Phi(x) \supset \Phi(succ\ x)) \wedge \forall x.(\Phi(x) \supset isnatnum\ x) \supset \forall x.(isnatnum\ x \supset \Phi(x)). \tag{6.14}$$

(6.37) asserts that the only natural numbers are those objects that (6.36) forces to be natural numbers, and this is essentially the usual axiom schema of induction. We can get closer to the usual schema by substituting $\Phi(x) \equiv \Psi(x) \wedge isnatnum\ x$. This and (6.36) make the second conjunct drop out giving

$$\Psi(0) \wedge \forall x.(\Psi(x) \supset \Psi(succ\ x)) \supset \forall x.(isnatnum\ x \supset \Psi(x)). \tag{6.15}$$

Example 4. Returning to the blocks world, suppose we have a predicate $on(x, y, s)$ asserting that block x is on block y in situation s. Suppose we have

another predicate $above(x, y, s)$ which asserts that block x is above block y in situation s. We may write

$$\forall xys.(on(x, y, s) \supset above(x, y, s)) \tag{6.16}$$

and

$$\forall xyzs.(above(x, y, s) \wedge above(y, z, s) \supset above(x, z, s)), \tag{6.17}$$

i.e. *above* is a transitive relation. Circumscribing *above* in (6.38)∧(6.39) gives

$$\begin{array}{l} \forall xys.(on(x, y, s) \supset \Phi(x, y, s)) \\ \wedge\forall xyzs.(\Phi(x, y, s) \wedge \Phi(y, z, s) \supset \Phi(x, z, s)) \\ \wedge\forall xys.(\Phi(x, y, s) \supset above(x, y, s)) \\ \supset \forall xys.(above(x, y, s) \supset \Phi(x, y, s)) \end{array} \tag{6.18}$$

which tells us that *above* is the transitive closure of *on*.

In the preceding two examples, the schemas produced by circumscription play the role of axiom schemas rather than being just conjectures.

6.5 DOMAIN CIRCUMSCRIPTION

The form of circumscription described in this paper generalizes an earlier version called *minimal inference*. Minimal inference has a semantic counterpart called *minimal entailment*, and both are discussed in (McCarthy 1977) and more extensively in (Davis 1980). The general idea of minimal entailment is that a sentence q is minimally entailed by an axiom A, written $A \models_m q$, if q is true in all *minimal models* of A, where one model if is considered less than another if they agree on common elements, but the domain of the larger many contain elements not in the domain of the smaller. We shall call the earlier form *domain circumscription* to contrast it with the *predicate circumscription* discussed in this paper.

The domain circumscription of the sentence A is the sentence

$$Axiom(\Phi) \wedge A^{\Phi} \supset \forall x.\Phi(x), \tag{6.19}$$

where A^{Φ} is the relativization of A with respect to Φ and is formed by replacing each universal quantifier $\forall x.$ in A by $\forall x.\Phi(x) \supset$ and each existential quantifier $\exists x.$ by $\exists x.\Phi(x)\wedge$. $Axiom(\Phi)$ is the conjunction of sentences $\Phi(a)$ for each constant a and sentences $\forall x.(\Phi(x) \supset \Phi(f(x)))$ for each function symbol f and the corresponding sentences for functions of higher arities.

Domain circumscription can be reduced to predicate circumscription by relativizing A with respect to a new one place predicate called (say) all, then circumscribing all in $A^{all} \wedge Axiom(all)$, thus getting

$$Axiom(\Phi) \wedge A^{\Phi} \wedge \forall x.(\Phi(x) \supset all(x)) \supset \forall x.(all(x) \supset \Phi(x)). \qquad (6.20)$$

Now we justify our using the name all by adding the axiom $\forall x.all(x)$ so that (6.67) then simplifies precisely to (6.66).

In the case of the natural numbers, the domain circumscription of **true**, the identically true sentence, again leads to the axiom schema of induction. Here $Axiom$ does all the work, because it asserts that 0 is in the domain and that the domain is closed under the successor operation.

6.6 THE MODEL THEORY OF PREDICATE CIRCUMSCRIPTION

This treatment is similar to Davis's (1980) treatment of domain circumscription. Pat Hayes (1979) pointed out that the same ideas would work.

The intuitive idea of circumscription is saying that a tuple $\bar{x}$ satisfies the predicate P only if it has to. It has to satisfy P if this follows from the sentence A. The model-theoretic counterpart of circumscription is *minimal entailment*. A sentence q is minimally entailed by A, if q is true in all minimal models of A, where a model is minimal if as few as possible tuples $\bar{x}$ satisfy the predicate P. More formally, this works out as follows.

Definition. Let $M(A)$ and $N(A)$ be models of the sentence A. We say that M *is a submodel* of N in P, writing $M \leq_P N$, if M and N have the same domain, all other predicate symbols in A besides P have the same extensions in M and N, but the extension of P in M is included in its extension in N.

Definition. A model M of A is called *minimal* in P if $M' \leq_P M$ only if $M' = M$. As discussed by Davis (1980), minimal models don't always exist.

Definition. We say that A *minimally entails* q *with respect to* P, written $A \models_p q$ provided q is true in all models of A that are minimal in P.

Theorem. Any instance of the circumscription of P in A is true in all models of A minimal in P, i.e. is minimally entailed by A in P.

Proof. Let M be a model of A minimal in P. Let P' be a predicate satisfying the left side of (6.33) when substituted for Φ. By the second conjunct of the

left side P is an extension of P'. If the right side of (6.33) were not satisfied, P would be a proper extension of P'. In that case, we could get a proper submodel M' of M by letting M' agree with M on all predicates except P and agree with P' on P. This would contradict the assumed minimality of M.

Corollary. If $A \vdash_P q$, then $A \models_P q$.

While we have discussed minimal entailment in a single predicate P, the relation $<_{P,Q}$, models minimal in P and Q, and $\models_{P,Q}$ have corresponding properties and a corresponding relation to the syntactic notion $\vdash_{P,Q}$ mentioned earlier.

6.7 MORE ON BLOCKS

The axiom

$$\forall xys.(\forall z.\neg prevents(z, move(x,y), s) \supset on(x, y, result(move(x,y), s))) \quad (6.21)$$

states that unless something prevents it, x is on y in the situation that results from the action $move(x, y)$.

We now list various "things" that may prevent this action.

$$\forall xys.(\neg isblock\, x \vee \neg isblock\, y \supset prevents(NONBLOCK, move(x,y), s)) \quad (6.22)$$

$$\forall xys.(\neg clear(x,s) \vee \neg clear(y,s) \supset prevents(COVERED, move(x,y), s)) \quad (6.23)$$

$$\forall xys.(tooheavyx \supset prevents(weightx, move(x,y), s)). \quad (6.24)$$

Let us now suppose that a heuristic program would like to move block A onto block C in a situation $s0$. The program should conjecture from (6.57) that the action $move(A, C)$ would have the desired effect, so it must try to establish $\forall z.\neg prevents(z, move(A,C), s0)$. The predicate $\lambda z.prevents(z, move(A,C), s0)$ can be circumscribed in the conjunction of the sentences resulting from specializing (6.58), (6.59) and (6.60), and this gives

$$\begin{array}{l}(\neg isblock\ A \vee \neg isblock\ C \supset \Phi(NONBLOCK)) \\ \wedge(\neg clear(A, s0) \vee \neg clear(C, s0) \supset \Phi(COVERED)) \\ \wedge(tooheavyA \supset \Phi(weightA)) \\ \wedge\forall z.(\Phi(z) \supset prevents(z, move(A,C), s0)) \\ \supset \forall z.(prevents(z, move(A,C), s0) \supset \Phi(z))\end{array} \quad (6.25)$$

which says that the only things that can prevent the move are the phenomena described in (6.58), (6.59) and (6.60). Whether (6.25) is true depends on how good the program was in finding all the relevant statements. Since the program wants to show that nothing prevents the move, it must set $\forall z.(\Phi(z) \equiv false)$, after which (6.25) simplifies to

$$(isblock\ A \wedge isblock\ B \wedge clear(A, s0) \wedge clear(B, s0) \wedge \neg tooheavy A \supset \forall z.\neg prevents(z, move(A, C), s0). \tag{6.26}$$

We suppose that the premisses of this implication are to be obtained as follows:

1. *isblock* A and *isblock* B are explicitly asserted.

2. Suppose that the only *on*ness assertion explicitly given for situation $s0$ is $on(A, B, s0)$. Circumscription of λx y.on(x,y,s0) in this assertion gives

$$\Phi(A, B) \wedge \forall xy.(\Phi(x, y) \supset on(x, y, s0)) \supset \forall xy.(on(x, y, s0) \supset \Phi(x, y)), \tag{6.27}$$

and taking $\Phi(x, y) \equiv x = A \wedge y = B$ yields

$$\forall xy.(on(x, y, s0) \supset x = A \wedge y = B). \tag{6.28}$$

Using

$$\forall xs.(clear(x, s) \equiv \forall y.\neg on(y, x, s)) \tag{6.29}$$

as the definition of *clear* yields the second two desired premisses.

3. $\neg tooheavy(x)$ might be explicitly present or it might also be conjectured by a circumscription assuming that if x were too heavy, the facts would establish it.

Circumscription may also be convenient for asserting that when a block is moved, everything that cannot be proved to move stays where it was. In the simple blocks world, the effect of this can easily be achieved by an axiom that states that all blocks except the one that is moved stay put. However, if there are various sentences that say (for example) that one block is attached to another, circumscription may express the heuristic situation better than an axiom.

6.8 REMARKS AND ACKNOWLEDGEMENTS

1. Circumscription is not a "nonmonotonic logic". It is a form of nonmonotonic reasoning augmenting ordinary first order logic. Of course, sentence schemata are not properly handled by most present general purpose resolution theorem provers. Even fixed schemata of mathematical induction when used for proving programs correct usually require human intervention or special heuristics, while here the program would have to use new schemata produced by circumscription. In (McCarthy 1979b) we treat some modalities in first order logic instead of in modal logic. In our opinion, it is better to avoid modifying the logic if at all possible, because there are many temptations to modify the logic, and it would be very difficult to keep them compatible.

2. The default case reasoning provided in many systems is less general than circumscription. Suppose, for example, that a block x is considered to be on a block y only if this is explicitly stated, i.e. the default is that x is not on y. Then for each individual block x, we may be able to conclude that it isn't on block A, but we will not be able to conclude, as circumscription would allow, that there are no blocks on A. That would require a separate default statement that a block is clear unless something is stated to be on it.

3. The conjunct $\forall \bar{x}.(\Phi(\bar{x}) \supset P(\bar{x}))$ in the premiss of (6.33) is the result of suggestions by Ashok Chandra (1979) and Patrick Hayes (1979) whom I thank for their help. Without it, circumscribing a disjunction, as in the second example in Section 4, would lead to a contradiction.

4. The most direct way of using circumscription in AI is in a heuristic reasoning program that represents much of what it believes by sentences of logic. The program would sometimes apply circumscription to certain predicates in sentences. In particular, when it wants to perform an action that might be prevented by something, it circumscribes the prevention predicate in a sentence A representing the information being taken into account.

Clearly the program will have to include domain dependent heuristics for deciding what circumscriptions to make and when to take them back.

5. In circumscription it does no harm to take irrelevant facts into account. If these facts do not contain the predicate symbol being circumscribed, they will appear as conjuncts on the left side of the implication unchanged. Therefore, the original versions of these facts can be used in proving the left side.

6. Circumscription can be used in other formalisms than first order logic. Suppose for example that a set a satisfies a formula $A(a)$ of set theory. The

circumscription of this formula can be taken to be

$$\forall x.(A(x) \wedge (x \subset a) \supset (a \subset x)). \tag{6.30}$$

If a occurs in $A(a)$ only in expressions of the form $z \in a$, then its mathematical properties should be analogous to those of predicate circumscription. We have not explored what happens if formulas like $a \in z$ occur.

7. The results of circumscription depend on the set of predicates used to express the facts. For example, the same facts about the blocks world can be axiomatized using the relation *on* or the relation *above* considered in section 4 or also in terms of the heights and horizontal positions of the blocks. Since the results of circumscription will differ according to which representation is chosen, we see that the choice of representation has epistemological consequences if circumscription is admitted as a rule of conjecture. Choosing the set of predicates in terms of which to axiomatize a set of facts, such as those about blocks, is like choosing a co-ordinate system in physics or geography. As discussed in (McCarthy 1979a), certain concepts are definable only relative to a theory. What theory admits the most useful kinds of circumscription may be an important criterion in the choice of predicates. It may also be possible to make some statements about a domain like the blocks world in a form that does not depend on the language used.

8. This investigation was supported in part by ARPA Contract MDA-903-76-C-0206, ARPA Order No. 2494, in part by NSF Grant MCS 78-00524, in part by the IBM 1979 Distinguished Faculty Program at the T. J. Watson Research Center, and in part by the Center for Advanced Study in the Behavioral Sciences.

6.9 References

Amarel, Saul (1971). On Representation of Problems of Reasoning about Actions, in D. Michie (ed.), *Machine Intelligence 3*, Edinburgh University Press, pp. 131–171.

Chandra, Ashok (1979). Personal conversation, August.

Davis, Martin (1980). Notes on the Mathematics of Non-Monotonic Reasoning, *Artificial Intelligence 13* (1, 2), pp. 73–80.

Hayes, Patrick (1979). Personal conversation, September.

Hewitt, Carl (1972). *Description and Theoretical Analysis (Using Schemata) of PLANNER: a Language for Proving Theorems and Manipulating Models in a Robot*, MIT AI Laboratory TR-258.

McCarthy, John (1959). Programs with Common Sense, *Proceedings of the Teddington Conference on the Mechanization of Thought Processes*, London: Her Majesty's Stationery Office. (Reprinted in this volume, pp. 000–000).

McCarthy, John and Patrick Hayes (1969)[1]. Some Philosophical Problems from the Standpoint of Artificial Intelligence, in B. Meltzer and D. Michie (eds), *Machine Intelligence 4*, Edinburgh University. (Reprinted in B. L. Webber and N. J. Nilsson (eds.), *Readings in Artificial Intelligence*, Tioga, 1981, pp. 431–450; also in M. J. Ginsberg (ed.), *Readings in Nonmonotonic Reasoning*, Morgan Kaufmann, 1987, pp. 26–45. Reprinted in (McCarthy 1990).

McCarthy, John (1977). Epistemological Problems of Artificial Intelligence, *Proceedings of the Fifth International Joint Conference on Artificial Intelligence*, M.I.T., Cambridge, Mass. (Reprinted in B. L. Webber and N. J. Nilsson (eds.), *Readings in Artificial Intelligence*, Tioga, 1981, pp. 459–465; also in M. J. Ginsberg (ed.), *Readings in Nonmonotonic Reasoning*, Morgan Kaufmann, 1987, pp. 46–52. Reprinted in (McCarthy 1990).

McCarthy, John (1979a). Ascribing Mental Qualities to Machines[2] , *Philosophical Perspectives in Artificial Intelligence*, Martin Ringle, ed., Humanities Press. Reprinted in (McCarthy 1990).

McCarthy, John (1979b). First Order Theories of Individual Concepts and Propositions[3] in Michie, Donald (ed.) *Machine Intelligence 9*, Ellis Horwood. Reprinted in (McCarthy 1990).

McCarthy, John (1990). *Formalizing Common Sense*, Ablex.

Reiter, Raymond (1980). A Logic for Default Reasoning, *Artificial Intelligence 13* (1, 2), pp. 81–132.

Sussman, G.J., T. Winograd, and E. Charniak (1971). *Micro-Planner Reference Manual, AI Memo 203*, M.I.T. AI Lab.

[1]http://www-formal.stanford.edu/jmc/mcchay69.html
[2]http://www-formal.stanford.edu/jmc/ascribing.html
[3]http://www-formal.stanford.edu/jmc/concepts.html

ADDENDUM: CIRCUMSCRIPTION AND OTHER NONMONOTONIC FORMALISMS

Circumscription and the nonmonotonic reasoning formalisms of McDermott and Doyle (1980) and Reiter (1980) differ along two dimensions. First, circumscription is concerned with minimal models, and they are concerned with arbitrary models. It appears that these approaches solve somewhat different though overlapping classes of problems, and each has its uses. The other difference is that the reasoning of both other formalisms involves models directly, while the syntactic formulation of circumscription uses axiom schemata. Consequently, their systems are incompletely formal unless the metamathematics is also formalized, and this hasn't yet been done.

However, schemata are applicable to other formalisms than circumscription. Suppose, for example, that we have some axioms about trains and their presence on tracks, and we wish to express the fact that if a train may be present, it is unsafe to cross the tracks. In the McDermott-Doyle formalism, this might be expressed

$$\mathbf{M}\, on(train, tracks) \supset \neg safe\text{-}to\text{-}cross(tracks), \tag{1}$$

where the properties of the predicate *on* are supposed expressed in a formula that we may call $Axiom(on)$. The **M** in (1) stands for "is possible". We propose to replace (1) and $Axiom(on)$ by the schema

$$Axiom(\Phi) \wedge \Phi(train, tracks) \supset \neg safe\text{-}to\text{-}cross(tracks), \tag{2}$$

where Φ is a predicate parameter that can be replaced by any predicate expression that can be written in the language being used. If we can find a Φ that makes the left hand side of (2) provable, then we can be sure that $Axiom(on)$ together with $on(train, tracks)$ has a model assuming that $Axiom(on)$ is consistent. Therefore, the schema (2) is essentially a consequence of the McDermott-Doyle formula (1). The converse isn't true. A predicate symbol may have a model without there being an explicit formula realizing it. I believe, however, that the schema is usable in all cases where the McDermott-Doyle or Reiter formalisms can be practically applied, and, in particular, to all the examples in their papers.

(If one wants a counter-example to the usability of the schema, one might look at the membership relation of set theory with the finitely axiomatized Gödel-Bernays set theory as the axiom. Instantiating Φ in this case would

amount to giving an internal model of set theory, and this is possible only in a stronger theory).

It appears that such use of schemata amounts to importing part of the model theory of a subject into the theory itself. It looks useful and even essential for common sense reasoning, but its logical properties are not obvious.

We can also go frankly to second order logic and write

$$\forall\Phi.(Axiom(\Phi) \wedge \Phi(train, tracks) \supset \neg safe\text{-}to\text{-}cross(tracks)). \quad (6.31)$$

Second order reasoning, which might be in set theory or a formalism admitting concepts as objects rather than in second order logic, seems to have the advantage that some of the predicate and function symbols may be left fixed and others imitated by predicate parameters. This allows us to say something like, "For any interpretation of P and Q satisfying the axiom A, if there is an interpretation in which R and S satisfy the additional axiom A', then it is unsafe to cross the tracks". This may be needed to express common sense nonmonotonic reasoning, and it seems more powerful than any of the above-mentioned nonmonotonic formalisms including circumscription.

The train example is a nonnormal default in Reiter's sense, because we cannot conclude that the train is on the tracks in the absence of evidence to the contrary. Indeed, suppose that we want to wait for and catch a train at a station across the tracks. If there might be a train coming we will take a bridge rather than a shortcut across the tracks, but we don't want to jump to the conclusion that there is a train, because then we would think we were too late and give up trying to catch it. The statement can be reformulated as a normal default by writing

$$\mathbf{M}\neg safe\text{-}to\text{-}cross(tracks) \supset \neg safe\text{-}to\text{-}cross(tracks), \quad (6.32)$$

but this is unlikely to be equivalent in all cases and the nonnormal expression seems to express better the common sense facts.

Like normal defaults, circumscription doesn't deal with possibility directly, and a circumscriptive treatment of the train problem would involve circumscribing $safe\text{-}to\text{-}cross(tracks)$ in the set of axioms. It therefore might not be completely satisfactory.

6.10 INTRODUCTION AND NEW DEFINITION OF CIRCUMSCRIPTION

(McCarthy 1980) introduces the circumscription method of nonmonotonic reasoning and gives motivation, some mathematical properties and some examples of its application. The present paper is logically self-contained, but motivation may be enhanced by reading the earlier paper. We don't repeat its arguments about the importance of nonmonotonic reasoning in AI, and its examples are instructive.

Here we give a more symmetric definition of circumscription and applications to the formal expression of common sense facts. Our long term goal (far from realized in the present paper) is to express these facts in a way that would be suitable for inclusion in a general purpose database of common sense knowledge. We imagine this database to be used by AI programs written after the initial preparation of the database. It would be best if the writers of these programs didn't have to be familiar with how the common sense facts about particular phenomena are expressed. Thus common sense knowledge must be represented in a way that is not specific to a particular application.

It turns out that many such common sense facts can be formalized in a uniform way. A single predicate *ab*, standing for "abnormal" is circumscribed with certain other predicates and functions considered as variables that can be constrained to achieve the circumscription subject to the axioms. This also seems to cover the use of circumscription to represent default rules.

6.11 A NEW VERSION OF CIRCUMSCRIPTION

Definition. Let $A(P)$ be a formula of second order logic, where P is a tuple of some of the free predicate symbols in $A(P)$. Let $E(P,x)$ be a wff in which P and a tuple x of individual variables occur free. The circumscription of $E(P,x)$ relative to $A(P)$ is the formula $A'(P)$ defined by

$$A(P)\wedge\forall P'.[A(P')\wedge[\forall x.E(P',x)\supset E(P,x)]\supset[\forall x.E(P',x)\equiv E(P,x)]]. \quad (6.33)$$

[We are here writing $A(P)$ instead of $A(P_1,\ldots,P_n)$ for brevity and likewise writing $E(P,x)$ instead of $E(P_1,\ldots,P_n,x_1,\ldots,x_m)$]. Likewise the quantifier

$\forall x$ stands for $\forall x_1 \ldots x_m$. $A(P)$ may have embedded quantifiers. Circumscription is a kind of minimization, and the predicate symbols in $A(P)$ that are not in P itself act as parameters in this minimization. When we wish to mention these other predicates we write $A(P;Q)$ and $E(P;Q,x)$ where Q is a vector of predicate symbols which are not allowed to be varied.

There are two differences between this and (McCarthy 1980). First, in that paper $E(P,x)$ had the specific form $P(x)$. Here we speak of circumscribing a wff and call the method *formula circumscription*, while there we could speak of circumscribing a predicate. We still speak of circumscribing the predicate P when $E(P,x)$ has the special form $P(x)$. Formula circumscription is more symmetric in that any of the predicate symbols in P may be regarded as variables, and a wff is minimized; the earlier form distinguishes one of the predicates themselves for minimization. However, formula circumscription is reducible to predicate circumscription provided we allow as variables predicates besides the one being minimized.

Second, in definition (6.33) we use an explicit quantifier for the predicate variable P' whereas in (McCarthy 1980), the formula was a schema. One advantage of the present formalism is that now $A'(P)$ is the same kind of formula as $A(P)$ and can be used as part of the axiom for circumscribing some other wff.

In some of the literature, it has been supposed that nonmonotonic reasoning involves giving all predicates their minimum extension. This mistake has led to theorems about what reasoning cannot be done that are irrelevant to AI and database theory, because their premisses are too narrow.

6.12 A TYPOLOGY OF USES OF NONMONOTONIC REASONING

Before proceeding to applications of circumscription I want to suggest a typology of the uses of nonmonotonic reasoning. Each of the several papers that introduces a mode of nonmonotonic reasoning seems to have a particular application in mind. Perhaps we are looking at different parts of an elephant. The orientation is towards circumscription, but I suppose the considerations apply to other formalisms as well.

Nonmonotonic reasoning has several uses.

1. As a communication convention. Suppose A tells B about a situation

involving a bird. If the bird cannot fly, and this is relevant, then A must say so. Whereas if the bird can fly, there is no requirement to mention the fact. For example, if I hire you to build me a bird cage and you don't put a top on it, I can get out of paying for it even if you tell the judge that I never said my bird could fly. However, if I complain that you wasted money by putting a top on a cage I intended for a penguin, the judge will agree with you that if the bird couldn't fly I should have said so.

The proposed Common Business Communication Language (CBCL) (McCarthy 1982) must include nonmonotonic conventions about what may be inferred when a message leaves out such items as the method of delivery.

2. As a database or information storage convention. It may be a convention of a particular database that certain predicates have their minimal extension. This generalizes the closed world assumption. When a database makes the closed world assumption for all predicates it is reasonable to imbed this fact in the programs that use the database. However, when only some predicates are to be minimized, we need to say which ones by appropriate sentences of the database, perhaps as a preamble to the collection of ground sentences that usually constitute the main content.

Neither 1 nor 2 requires that most birds can fly. Should it happen that most birds that are subject to the communication or about which information is requested from the data base cannot fly, the convention may lead to inefficiency but not incorrectness.

3. As a rule of conjecture. This use was emphasized in (McCarthy 1980). The circumscriptions may be regarded as expressions of some probabilistic notions such as "most birds can fly" or they may be expressions of standard cases. Thus it is simple to conjecture that there are no relevant present material objects other than those whose presence can be inferred. It is also a simple conjecture that a tool asserted to be present is usable for its normal function. Such conjectures sometimes conflict, but there is nothing wrong with having incompatible conjectures on hand. Besides the possibility of deciding that one is correct and the other wrong, it is possible to use one for generating possible exceptions to the other.

4. As a representation of a policy. The example is Doyle's "The meeting will be on Wednesday unless another decision is explicitly made". Again probabilities are not involved.

5. As a very streamlined expression of probabilistic information when numerical probabilities, especially conditional probabilities, are unobtainable. Since circumscription doesn't provide numerical probabilities, its probabilis-

tic interpretation involves probabilities that are either infinitesimal, within an infinitesimal of one, or intermediate — without any discrimination among the intermediate values. The circumscriptions give conditional probabilities. Thus we may treat the probability that a bird can't fly as an infinitesimal. However, if the rare event occurs that the bird is a penguin, then the conditional probability that it can fly is infinitesimal, but we may hear of some rare condition that would allow it to fly after all.

Why don't we use finite probabilities combined by the usual laws? That would be fine if we had the numbers, but circumscription is usable when we can't get the numbers or find their use inconvenient. Note that the general probability that a bird can fly may be irrelevant, because we are interested in the facts that influence our opinion about whether a particular bird can fly in a particular situation.

Moreover, the use of probabilities is normally considered to require the definition of a sample space, i.e. the space of all possibilities. Circumscription allows one to conjecture that the cases we know about are all that there are. However, when additional cases are found, the axioms don't have to be changed. Thus there is no fixed space of all possibilities.

Notice also that circumscription does not provide for weighing evidence; it is appropriate when the information permits snap decisions. However, many cases nominally treated in terms of weighing information are in fact cases in which the weights are such that circumscription and other defaults work better.

6. Auto-epistemic reasoning. "If I had an elder brother, I'd know it". This has been studied by R. Moore. Perhaps it can be handled by circumscription.

7. Both common sense physics and common sense psychology use nonmonotonic rules. An object will continue in a straight line if nothing interferes with it. A person will eat when hungry unless something prevents it. Such rules are open ended about what might prevent the expected behavior, and this is required, because we are always encountering unexpected phenomena that modify the operation of our rules. Science, as distinct from common sense, tries to work with exceptionless rules. However, this means that common sense reasoning has to decide when a scientific model is applicable, i.e. that there are no important phenomena not taken into account by the theories being used and the model of the particular phenomena.

Seven different uses for nonmonotonic reasoning seem too many, so perhaps we can condense later.

6.13 MINIMIZING ABNORMALITY

Many people have proposed representing facts about what is "normally" the case. One problem is that every object is abnormal in some way, and we want to allow some aspects of the object to be abnormal and still assume the normality of the rest. We do this with a predicate *ab* standing for "abnormal". We circumscribe $ab\ z$. The argument of *ab* will be some aspect of the entities involved. Some aspects can be abnormal without affecting others. The aspects themselves are abstract entities, and their unintuitiveness is somewhat a blemish on the theory.

The idea is illustrated by the examples of the following sections.

6.14 WHETHER BIRDS CAN FLY

Marvin Minsky challenged us advocates of formal systems based on mathematical logic to express the facts and nonmonotonic reasoning concerning the ability of birds to fly.

There are many ways of nonmonotonically axiomatizing the facts about which birds can fly. The following axioms using *ab* seem to me quite straightforward.

$$\forall x.\neg ab\ aspect1\ x \supset \neg flies\ x. \tag{6.34}$$

Unless an object is abnormal in $aspect1$, it can't fly. (We're using a convention that parentheses may be omitted for functions and predicates of one argument, so that (6.34) is the same as $\forall x.(\neg ab(aspect1(x)) \supset \neg flies(x))$.)

It wouldn't work to write $ab\ x$ instead of $ab\ aspect1\ x$, because we don't want a bird that is abnormal with respect to its ability to fly to be automatically abnormal in other respects. Using aspects limits the effects of proofs of abnormality.

$$\forall x.bird\ x \supset ab\ aspect1\ x. \tag{6.35}$$

$$\forall x.bird\ x \wedge \neg ab\ aspect2\ x \supset flies\ x. \tag{6.36}$$

Unless a bird is abnormal in $aspect2$, it can fly.

A bird is abnormal in $aspect1$, so (6.34) can't be used to show it can't fly. If (6.35) were omitted, when we did the circumscription we would only be

able to infer a disjunction. Either a bird is abnormal in $aspect1$ or it can fly unless it is abnormal in $aspect2$. (6.35) expresses our preference for inferring that a bird is abnormal in $aspect1$ rather than $aspect2$. We call (6.35) a *cancellation of inheritance* axiom.

$$\forall x.ostrich\ x \supset ab\ aspect2\ x. \tag{6.37}$$

Ostriches are abnormal in $aspect2$. This doesn't say that an ostrich cannot fly — merely that (6.36) can't be used to infer that it does. (6.37) is another cancellation of inheritance axiom.

$$\forall x.penguin\ x \supset ab\ aspect2\ x. \tag{6.38}$$

Penguins are also abnormal in $aspect2$.

$$\forall x.ostrich\ x \wedge \neg ab\ aspect3\ x \supset \neg flies\ x. \tag{6.39}$$

$$\forall x.penguin\ x \wedge \neg ab\ aspect4\ x \supset \neg flies\ x. \tag{6.40}$$

Normally ostriches and penguins can't fly. However, there is an out. (6.39) and (6.40) provide that under unspecified conditions, an ostrich or penguin might fly after all. If we give no such conditions, we will conclude that an ostrich or penguin can't fly. Additional objects that can fly may be specified. Each needs two axioms. The first says that it is abnormal in $aspect1$ and prevents (6.34) from being used to say that it can't fly. The second provides that it can fly unless it is abnormal in yet another way. Additional non-flying birds can also be provided for at a cost of two axioms per kind.

We haven't yet said that ostriches and penguins are birds, so let's do that and throw in that canaries are birds also.

$$\forall x.ostrich\ x \supset bird\ x. \tag{6.41}$$

$$\forall x.penguin\ x \supset bird\ x. \tag{6.42}$$

$$\forall x.canary\ x \supset bird\ x. \tag{6.43}$$

Asserting that ostriches, penguins and canaries are birds will help inherit other properties from the class of birds. For example, we have

$$\forall x.bird\ x \wedge \neg ab\ aspect5\ x \supset feathered\ x. \tag{6.44}$$

So far there is nothing to prevent ostriches, penguins and canaries from overlapping. We could write disjointness axioms like

$$\forall x.\neg ostrich\ x \vee \neg penguin\ x, \tag{6.45}$$

but we require n^2 of them if we have n species. It is more efficient to write axioms like

$$\forall x.ostrich\ x \supset species\ x = {}'ostrich, \tag{6.46}$$

which makes the n species disjoint with only n axioms assuming that the distinctness of the names is apparent to the reasoner. This problem is like the unique names problem.

If these are the only facts to be taken into account, we must somehow specify that what can fly is to be determined by circumscribing the wff $ab\ z$ using ab and $flies$ as variables. Why exactly these? If ab were not taken as variable, $ab\ z$ couldn't vary either, and the minimization problem would go away. Since the purpose of the axiom set is to describe what flies, the predicate $flies$ must be varied also. Suppose we contemplate taking $bird$ as variable also. In the first place, this violates an intuition that deciding what flies follows deciding what is a bird in the common sense situations we want to cover. Secondly, if we use exactly the above axioms and admit bird as a variable, we will further conclude that the only birds are penguins, canaries and ostriches. Namely, for these entities something has to be abnormal, and therefore minimizing $ab\ z$ will involve making as few entities as possible penguins, canaries and ostriches. If we also admit $penguin$, $ostrich$, and $canary$ as variable, we will succeed in making $ab\ z$ always false, and there will be no birds at all.

However, if the same circumscriptions are done with additional axioms like $canary\ Tweety$ and $ostrich\ Joe$, we will get the expected result that Tweety can fly and Joe cannot even if all the above are variable.

While this works it may be more straightforward, and therefore less likely to lead to subsequent trouble, to circumscribe birds, ostriches and penguins with axioms like

$$\forall x.\neg ab\ aspect6\ x \supset \neg bird\ x, \tag{6.47}$$

We have not yet specified how a program will know what to circumscribe. One extreme is to build it into the program, but this is contrary to the

declarative spirit. However, a statement of what to circumscribe isn't just a sentence of the language because of its nonmonotonic character. Another possibility is to include some sort of metamathematical statement like

$$circumscribe(ab\ z\,; ab, flies, bird, ostrich, penguin) \tag{6.48}$$

in a "policy" database available to the program. (6.48) is intended to mean that $ab\ z$ is to be circumscribed with ab, $flies$, $bird$, $ostrich$ and $penguin$ taken as variable. Explicitly listing the variables makes adding new kinds awkward, since they will have to be mentioned in the *circumscribe* statement. Section 11 on *simple abnormality theories* presents yet another possibility.

6.15 THE UNIQUE NAMES HYPOTHESIS

Raymond Reiter (1980b) introduced the phrase "unique names hypothesis" for the assumption that each object has a unique name, i.e. that distinct names denote distinct objects. We want to treat this nonmonotonically. Namely, we want a wff that picks out those models of our initial assumptions that maximize the inequality of the denotations of constant symbols. While we're at it, we might as well try for something stronger. We want to maximize the extent to which distinct terms designate distinct objects. When there is a unique model of the axioms that maximizes distinctness, we can put it more simply; two terms denote distinct objects unless the axioms force them to denote the same. If we are even more fortunate, as we are in the examples to be given, we can say that two terms denote distinct objects unless their equality is provable.

We don't know a completely satisfactory way of doing this. Suppose that we have a language L and a theory T consisting of the consequences of a formula A. It would be most pleasant if we could just circumscribe equality, but as Etherington, Mercer and Reiter (1985) point out, this doesn't work, and nothing similar works. We could hope to circumscribe some other formula of L, but this doesn't seem to work either. Failing that, we could hope for some other second order formula taken from L that would express the unique names hypothesis, but we don't presently see how to do it.

Our solution involves extending the language by introducing the names themselves as the only objects. All assertions about objects are expressed as assertions about the names.

We suppose our theory is such that the names themselves are all provably distinct. There are several ways of doing this. Let the names be n_1, n_2, etc. The simplest solution is to have an axiom $n_i \neq n_j$ for each pair of distinct names. This requires a number of axioms proportional to the square of the number of names, which is sometimes objectionable. The next solution involves introducing an arbitrary ordering on the names. We have special axioms $n_1 < n_2, n_2 < n_3, n_3 < n_4$, etc. and the general axioms $\forall xy.x < y \supset x \neq y$ and $\forall xyz.x < y \wedge y < z \supset x < z$. This makes the number of axioms proportional to the number of names. A third possibility involves mapping the names onto integers with axioms like $index\ n_1 = 1, index\ n_2 = 2$, etc. and using a theory of the integers that provides for their distinctness. The fourth possibility involves using string constants for the names and "attaching" to equality in the language a subroutine that computes whether two strings are equal. If our names were quoted symbols as in LISP, this amounts to having $'a \neq 'b$ and all its countable infinity of analogs as axioms. Each of these devices is useful in appropriate circumstances.

From the point of view of mathematical logic, there is no harm in having an infinity of such axioms. From the computational point of view of a theorem proving or problem solving program, we merely suppose that we rely on the computer to generate the assertion that two names are distinct whenever this is required, since a subroutine can easily tell whether two strings are the same.

Besides axiomatizing the distinctness of the constants, we also want to axiomatize the distinctness of terms. This may be accomplished by providing for each function two axioms. Letting foo be a function of two arguments we postulate

$$\forall x_1 x_2 y_1 y_2.foo(x_1, y_1) = foo(x_2, y_2) \supset x_1 = x_2 \wedge y_1 = y_2 \tag{6.49}$$

and

$$\forall xy.fnamefoo(x, y) = 'foo. \tag{6.50}$$

The first axiom ensures that unless the arguments of foo are identical, its values are distinct. The second ensures that the values of foo are distinct from the values of any other function or any constant, assuming that we refrain from naming any constant $'foo$.

These axioms amount to making our domain isomorphic to an extension of the Herbrand universe of the language.

Now that the names are guaranteed distinct, what about the objects they denote? We introduce a predicate $e(x, y)$ and axiomatize it to be an equivalence relation. Its intended interpretation is that the names x and y denote the same object. We then formulate all our usual axioms in terms of names rather than in terms of objects. Thus $on(n_1, n_2)$ means that the object named by n_1 is on the object named by n_2, and $bird\ x$ means that the name x denotes a bird. We add axioms of substitutivity for e with regard to those predicates and functions that are translates of predicates referring to objects rather than predicates on the names themselves. Thus for a predicate on and a function foo we may have axioms

$$\forall n_1 n_2 n_1' n_2'.e(n_1, n_1') \wedge e(n_2, n_2') \supset (on(n_1, n_2) \equiv on(n_1', n_2')) \tag{6.51}$$

and

$$\forall x_1 x_2 y_1 y_2.e(x_1, x_2) \wedge e(y_1, y_2) \supset e(foo(x_1, y_1), foo(x_2, y_2)). \tag{6.52}$$

If for some class C of names, we wish to assert the unique names hypothesis, we simply use an axiom like

$$\forall n_1 n_2.n_1 \in C \wedge n_2 \in C \supset (e(n_1, n_2) \equiv n_1 = n_2). \tag{6.53}$$

However, we often want only to assume that distinct names denote distinct objects when this doesn't contradict our other assumptions. In general, our axioms won't permit making all names distinct simultaneously, and there will be several models with maximally distinct objects. The simplest example is obtained by circumscribing $e(x, y)$ while adhering to the axiom

$$e(n_1, n_2) \vee e(n_1, n_3)$$

where n_1, n_2, and n_3 are distinct names. There will then be two models, one satisfying $e(n_1, n_2) \wedge \neg e(n_1, n_3)$ and the other satisfying $\neg e(n_1, n_2) \wedge e(n_1, n_3)$.

Thus circumscribing $e(x, y)$ maximizes uniqueness of names. If we only want unique names for some class C of names, then we circumscribe the formula

$$x \in C \wedge y \in C \supset e(x, y). \tag{6.54}$$

An example of such a circumscription is given in Appendix B. However, there seems to be a price. Part of the price is admitting names as objects. Another part is admitting the predicate $e(x, y)$ which is substitutive for predicates and

functions of names that really are about the objects denoted by the names. $e(x, y)$ is not to be taken as substitutive for predicates on names that aren't about the objects. Of these our only present example is equality. Thus we don't have

$$\forall n_1 n_2 n_1' n_2'.e(n_1, n_1') \wedge e(n_2, n_2') \supset (n_1 = n_2 \equiv n_1' = n_2').$$

The awkward part of the price is that we must refrain from any functions whose values are the objects themselves rather than names. They would spoil the circumscription by not allowing us to infer the distinctness of the objects denoted by distinct names. Actually, we can allow them provided we don't include the axioms involving them in the circumscription. Unfortunately, this spoils the other property of circumscription that lets us take any facts into account.

The examples of the use of circumscription in AI in the rest of the paper don't interpret the variables as merely ranging over names. Therefore, they are incompatible with getting unique names by circumscription as described in this section. Presumably it wouldn't be very difficult to revise those axioms for compatibility with the present approach to unique names.

6.16 TWO EXAMPLES OF RAYMOND REITER

Reiter asks about representing, "Quakers are normally pacifists and Republicans are normally non-pacifists. How about Nixon, who is both a Quaker and a Republican?" Systems of nonmonotonic reasoning that use non-provability as a basis for inferring negation will infer that Nixon is neither a pacifist nor a non-pacifist. Combining these conclusions with the original premiss leads to a contradiction. We use

$$\forall x.quaker\ x \wedge \neg ab\ aspect1\ x \supset pacifist\ x, \tag{6.55}$$

$$\forall x.republican\ x \wedge \neg ab\ aspect2\ x \supset \neg pacifist\ x \tag{6.56}$$

and

$$quaker\ Nixon \wedge republican\ Nixon. \tag{6.57}$$

When we circumscribe $ab\ z$ using these three sentences as $A(ab, pacifist)$, we will only be able to conclude that Nixon is either abnormal in $aspect1$ or in $aspect2$, and we will not be able to say whether he is a pacifist. Of course, this is the same conclusion as would be reached without circumscription. The point is merely that we avoid contradiction.

Reiter's second example is that a person normally lives in the same city as his wife and in the same city as his employer. But A's wife lives in Vancouver and A's employer is in Toronto. We write

$$\forall x.\neg ab\ aspect1\ x \supset city\ x = city\ wife\ x \tag{6.58}$$

and

$$\forall x.\neg ab\ aspect2\ x \supset city\ x = city\ employer\ x. \tag{6.59}$$

If we have

$$city\ wife\ A = Vancouver \wedge city\ employer\ A = Toronto \wedge Toronto \neq Vancouver, \tag{6.60}$$

we will again only be able to conclude that A lives either in Toronto or Vancouver. In this circumscription, the function *city* must be taken as variable. This might be considered not entirely satisfactory. If one knows that a person either doesn't live in the same city as his wife or doesn't live in the same city as his employer, then there is an increased probability that he doesn't live in the same city as either. A system that did reasoning of this kind would seem to require a larger body of facts and perhaps more explicitly metamathematical reasoning. Not knowing how to do that, we might want to use $aspect1\ x$ in both (6.58) and (6.59). Then we would conclude nothing about his city once we knew that he wasn't in the same city as both.

6.17 A MORE GENERAL TREATMENT OF AN *IS-A* HIERARCHY

The bird example works fine when a fixed *is-a* hierarchy is in question. However, our writing the inheritance cancellation axioms depended on knowing exactly from what higher level the properties were inherited. This doesn't correspond to my intuition of how we humans represent inheritance. It would seem rather that when we say that birds can fly, we don't necessarily have in mind that an inheritance of inability to fly from things in general is being

cancelled. We can formulate inheritance of properties in a more general way provided we reify the properties. Presumably there are many ways of doing this, but here's one that seems to work.

The first order variables of our theory range over classes of objects (denoted by c with numerical suffixes), properties (denoted by p) and objects (denoted by x). We don't identify our classes with sets (or with the classes of Gödel-Bernays set theory). In particular, we don't assume extensionality. We have several predicates:

$ordinarily(c, p)$ means that objects of class c ordinarily have property p. $c1 \leq c2$ means that class $c1$ ordinarily inherits from class $c2$. We assume that this relation is transitive. $in(x, c)$ means that the object x is in class c. $ap(p, x)$ means that property p applies to object x. Our axioms are

$$\forall c1c2c3.c1 \leq c2 \wedge c2 \leq c3 \supset c1 \leq c3, \tag{6.61}$$

$$\forall c1c2p.ordinarily(c2, p) \wedge c1 \leq c2 \wedge \neg ab\, aspect1(c1, c2, p) \supset ordinarily(c1, p), \tag{6.62}$$

$$\forall c1c2c3p.c1 \leq c2 \wedge c2 \leq c3 \wedge ordinarily(c2, not\ p) \supset ab\, aspect1(c1, c3, p), \tag{6.63}$$

$$\forall xcp.in(x, c) \wedge ordinarily(c, p) \wedge \neg ab\ aspect2(x, c, p) \supset ap(p, x), \tag{6.64}$$

$$\forall xc1c2p.in(x, c1) \wedge c1 \leq c2 \wedge ordinarily(c1, not\ p) \supset ab\, aspect2(x, c2, p). \tag{6.65}$$

Axiom (6.61) is the afore-mentioned transitivity of $\leq$. (6.62) says that properties that ordinarily hold for a class are inherited unless something is abnormal. (6.63) cancels the inheritance if there is an intermediate class for which the property ordinarily doesn't hold. (6.64) says that properties which ordinarily hold actually hold for elements of the class unless something is abnormal. (6.65) cancels the effect of (6.64) when there is an intermediate class for which the negation of the property ordinarily holds. Notice that this reification of properties seems to require imitation boolean operators. Such operators are discussed in (McCarthy 1979).

6.18 THE BLOCKS WORLD

The following set of "situation calculus" axioms solves the frame problem for a blocks world in which blocks can be moved and painted. Here $result(e, s)$ denotes the situation that results when event e occurs in situation s. The formalism is approximately that of (McCarthy and Hayes 1969).

$$\forall xes.\neg ab\, aspect1(x, e, s) \supset location(x, result(e, s)) = location(x, s). \quad (6.66)$$

$$\forall xes.\neg ab\ aspect2(x, e, s) \supset color(x, result(e, s)) = color(x, s). \quad (6.67)$$

Objects change their locations and colors only for a reason.

$$\forall xls.ab\ aspect1(x, move(x, l), s) \quad (6.68)$$

and

$$\forall xls.\neg ab\ aspect3(x, l, s) \supset location(x, result(move(x, l), s)) = l. \quad (6.69)$$

$$\forall xcs.ab\ aspect2(x, paint(x, c), s) \quad (6.70)$$

and

$$\forall xcs.\neg ab\ aspect4(x, c, s) \supset color(x, result(paint(x, c), s)) = c. \quad (6.71)$$

Objects change their locations when moved and their colors when painted.

$$\forall xls.\neg clear(topx, s) \vee \neg clear(l, s) \vee tooheavy\ x \vee l = top\ x \quad (6.72)$$

$$\supset ab\ aspect3(x, l, s). \quad (6.73)$$

This prevents the rule (6.68) from being used to infer that an object will move if its top isn't clear or to a destination that isn't clear or if the object is too heavy. An object also cannot be moved to its own top.

$$\forall ls.clear(l, s) \equiv \neg\exists x.(\neg trivial\ x \wedge location(x, s) = l). \quad (6.74)$$

A location is clear if all the objects there are trivial, e.g. a speck of dust.

$$\forall x.\neg ab\ aspect5\ x \supset \neg trivial\ x. \quad (6.75)$$

Trivial objects are abnormal in $aspect5$.

6.19 AN EXAMPLE OF DOING THE CIRCUMSCRIPTION

In order to keep the example short we will take into account only the following facts from the earlier section on flying.

$$\forall x.\neg ab\ aspect1\ x \supset \neg flies\ x. \tag{6.2}$$

$$\forall x.bird\ x \supset ab\ aspect1\ x. \tag{6.3}$$

$$\forall x.bird\ x \wedge \neg ab\ aspect2\ x \supset flies\ x. \tag{6.4}$$

$$\forall x.ostrich\ x \supset ab\ aspect2\ x. \tag{6.5}$$

$$\forall x.ostrich\ x \wedge \neg ab\ aspect3\ x \supset \neg flies\ x. \tag{6.7}$$

Their conjunction is taken as $A(ab, flies)$. This means that what entities satisfy ab and what entities satisfy $flies$ are to be chosen so as to minimize $ab\ z$. Which objects are birds and ostriches are parameters rather than variables, i.e. what objects are birds is considered given.

We also need an axiom that asserts that the aspects are different. Here is a straightforward version that would be rather long were there more than three aspects.

$$\begin{array}{r}(\forall xy.\neg(aspect1\ x = aspect2\ y))\\ \wedge(\forall xy.\neg(aspect1\ x = aspect3\ y))\\ \wedge(\forall xy.\neg(aspect2\ x = aspect3\ y))\\ \wedge(\forall xy.aspect1\ x = aspect1\ y \equiv x = y)\\ \wedge(\forall xy.aspect2\ x = aspect2\ y \equiv x = y)\\ \wedge(\forall xy.aspect3\ x = aspect3\ y \equiv x = y).\end{array}$$

We could include this axiom in $A(ab, flies)$, but as we shall see, it won't matter whether we do, because it contains neither ab nor $flies$. The circumscription formula $A'(ab, flies)$ is then

$$A(ab, flies) \wedge \forall ab'\, flies'.[A(ab', flies') \wedge [\forall x.ab'\ x \supset ab\ x] \supset [\forall x.ab\ x \equiv ab'\ x]], \tag{6.41}$$

which spelled out becomes

$$
\begin{aligned}
&[\forall x.\neg ab\ aspect1\ x \supset \neg flies\ x] \\
&\wedge[\forall x.bird\ x \supset ab\ aspect1\ x] \\
&\wedge[\forall x.bird\ x \wedge \neg ab\ aspect2\ x \supset flies\ x] \\
&\wedge[\forall x.ostrich\ x \supset ab\ aspect2\ x] \\
&\wedge[\forall x.ostrich\ x \wedge \neg ab\ aspect3\ x \supset \neg flies\ x] \\
&\wedge\forall ab'\ flies'.\ [[\forall x.\neg ab'\ aspect1\ x \supset \neg flies'\ x] \\
&\qquad\qquad \wedge[\forall x.bird\ x \supset ab'\ aspect1\ x] \\
&\qquad\qquad \wedge[\forall x.bird\ x \wedge \neg ab'\ aspect2\ x \supset flies'\ x] \\
&\qquad\qquad \wedge[\forall x.ostrich\ x \supset ab'\ aspect2\ x] \\
&\qquad\qquad \wedge[\forall x.ostrich\ x \wedge \neg ab'\ aspect3\ x \supset \neg flies'\ x] \\
&\qquad\qquad \wedge[\forall z.ab'\ z \supset ab\ z] \\
&\supset [\forall z.ab\ z \equiv ab'\ z]].
\end{aligned}
\tag{6.42}
$$

$A(ab, flies)$ is guaranteed to be true, because it is part of what is assumed in our common sense database. Therefore (6.42) reduces to

$$
\begin{aligned}
&\forall ab' flies'.[[\forall x.\neg ab'\ aspect1\ x \supset \neg flies'\ x] \\
&\wedge[\forall x.bird\ x \supset ab'\ aspect1\ x] \\
&\wedge[\forall x.bird\ x \wedge \neg ab'\ aspect2\ x \supset flies'\ x] \\
&\wedge[\forall x.ostrich\ x \supset ab'\ aspect2\ x] \\
&\wedge[\forall x.ostrich\ x \wedge \neg ab'\ aspect3\ x \supset \neg flies'\ x] \\
&\wedge[\forall z.ab'\ z \supset ab\ z] \\
&\supset [\forall z.ab\ z \equiv ab'\ z]].
\end{aligned}
\tag{6.43}
$$

Our objective is now to make suitable substitutions for ab' and $flies'$ so that all the terms preceding the $\supset$ in (6.43) will be true, and the right side will determine ab. The axiom $A(ab, flies)$ will then determine $flies$, i.e. we will know what the fliers are. $flies'$ is easy, because we need only apply wishful thinking; we want the fliers to be just those birds that aren't ostriches. Therefore, we put

$$flies'\ x \equiv bird\ x \wedge \neg ostrich\ x. \tag{6.44}$$

ab' isn't really much more difficult, but there is a notational problem. We define

$$ab'\ z \equiv [\exists x.bird\ x \wedge z = aspect1\ x] \vee [\exists x.ostrich\ x \wedge z = aspect2\ x], \tag{6.45}$$

which covers the cases we want to be abnormal.

Appendix A contains a complete proof as accepted by Jussi Ketonen's (1984) interactive theorem prover EKL. EKL uses the theory of types and therefore has no problem with the second order logic required by circumscription.

6.20 SIMPLE ABNORMALITY THEORIES

The examples in this paper all circumscribe the predicate *ab*. However, they differ in what they take as variable in the circumscription. The declarative expression of common sense requires that we be definite about what is circumscribed and what is variable in the circumscription. We have the following objectives.

1. The general facts of common sense are described by a collection of sentences that are not oriented in advance to particular problems.

2. It is prescribed how to express the facts of particular situations including the goals to be achieved.

3. The general system prescribes what is to be circumscribed and what is variable.

4. Once the facts to be taken into account are chosen, the circumscription process proceeds in a definite way resulting in a definite theory — in general second order.

5. The conclusions reached taking a given set of facts into account are intuitively reasonable.

These objectives are the same as those of (McCarthy 1959) except that that paper used only monotonic reasoning.

The examples of this paper suggest defining a *simple abnormality formalism* used as follows.

1. The general facts include *ab* and a variety of aspects.

2. The specific facts do not involve *ab*.

3. The circumscription of *ab* is done with all predicates variable. This means that the axioms must be sufficient to tie them down.

I had hoped that the simple abnormality formalism would be adequate to express common sense knowledge. Unfortunately, this seems not to be the case. Consider the following axioms.

$$\neg ab\ aspect1\ x \supset \neg flies\ x,$$

$$bird\ x \supset ab\ aspect1\ x,$$

$$bird\ x \wedge \neg ab\ aspect2\ x \supset flies\ x,$$

$$canary\ x \wedge \neg ab\ aspect3\ x \supset bird\ x,$$

$$canary\ Tweety.$$

We ask whether Tweety flies. Simply circumscribing *ab* leaves this undecided, because Tweety can either be abnormal in *aspect*1 or in *aspect*3. Common sense tells us that we should conclude that Tweety flies. This can be achieved by preferring to have Tweety abnormal in *aspect*1 to having Tweety abnormal in *aspect*3. It is not yet clear whether this can be done using the *simple circumscriptive formalism*. Our approach to solving this problem is discussed in the following section on prioritized circumscription. However, simple abnormality theories may be adequate for an interesting set of common sense axiomatizations.

6.21 PRIORITIZED CIRCUMSCRIPTION

An alternate way of introducing formula circumscription is by means of an ordering on tuples of predicates satisfying an axiom. We define $P \leq P'$ by

$$\forall PP'.P \leq P' \equiv \forall x.E(P,x) \supset E(P',x). \tag{6.46}$$

That $P0$ is a relative minimum in this ordering is expressed by

$$\forall P.P \leq P0 \supset P = P0, \tag{6.47}$$

where equality is interpreted extensionally, i.e. we have

$$\forall PP'.P = P' \equiv (\forall x.E(P,x) \equiv E(P',x)). \tag{6.48}$$

Assuming that we look for a minimum among predicates P satisfying $A(P)$, (6.46) expands to precisely to the circumscription formula (6.33). In some earlier expositions of circumscription this ordering approach was used, and Vladimir Lifschitz in a recent seminar advocated returning to it as a more fundamental and understandable concept.

I'm beginning to think he's right about it being more understandable, and there seems to be a more fundamental reason for using it. Namely, certain common sense axiomatizations are easier to formalize if we use a new kind

of ordering, and circumscription based on this kind of ordering doesn't seem to reduce to ordinary formula circumscription.

We call it *prioritized circumscription.*

Suppose we write some bird axioms in the form

$$\forall x.\neg ab\ aspect1\ x \supset \neg flies\ x \tag{6.49}$$

and

$$\forall x.bird\ x \wedge \neg ab\ aspect2\ x \supset ab\ aspect1\ x. \tag{6.50}$$

The intent is clear. The goal is that being a bird and not abnormal in $aspect2$ prevents the application of (6.49). However, circumscribing $ab\ z$ with the conjunction of (6.49) and (6.50) as $A(ab)$ doesn't have this effect, because (6.50) is equivalent to

$$\forall x.bird\ x \supset ab\ aspect1\ x \vee ab\ aspect2\ x, \tag{6.51}$$

and there is no indication that one would prefer to have $aspect1\ x$ abnormal rather than to have $aspect2\ x$ abnormal. Circumscription then results in a disjunction which is not wanted in this case. The need to avoid this disjunction is why the axioms in section 6.14 (page 97) included cancellation of inheritance axioms.

However, by using a new kind of ordering we can leave (6.49) and (6.50) as is, and still get the desired effect.

We define two orderings on ab predicates, namely

$$\forall ab\ ab'.ab\ \leq_1 ab' \equiv \forall x.ab\ aspect1\ x \supset ab'\ aspect1\ x \tag{6.52}$$

and

$$\forall ab\ ab'.ab\ \leq_2 ab' \equiv \forall x.ab\ aspect2\ x \supset ab'\ aspect2\ x. \tag{6.53}$$

We then combine these orderings lexicographically giving $\leq_2$ priority over $\leq_1$ getting

$$\forall ab\ ab'.ab\ \leq_{1<2} ab' \equiv ab\ \leq_2 ab' \wedge ab\ =_2 ab' \supset ab\ \leq_1 ab'. \tag{6.54}$$

Choosing $ab0$ so as to minimize this ordering yields the result that exactly birds can fly. However, if we add

$$\forall x.ostrich\ x \supset ab\ aspect2\ x, \tag{6.55}$$

we'll get that ostriches (whether or not ostriches are birds) don't fly without further axioms. If we use

$$\forall x.ostrich\ x \wedge \neg ab\ aspect3\ x \supset ab\ aspect2\ x \tag{6.56}$$

instead of (6.55), we'll have to revise our notion of ordering to put minimizing $ab\ aspect3\ x$ at higher priority than minimizing $aspect2\ x$ and $a\ fortiori$ at higher priority than minimizing $aspect1$.

This suggests providing a partial ordering on aspects giving their priorities and providing axioms that permit deducing the ordering on ab from the sentences that describe the ordering relations. Lifschitz (1985) further develops the idea of prioritized circumscription.

I expect that *prioritized circumscription* will turn out to be the most natural and powerful variant.

Simple abnormality theories seem to be inadequate also for the blocks world described in section 11. I am indebted to Lifschitz for the following example. Consider

$$S2 = result(move(B, top\ A), result(move(A, top\ B), S0)), \tag{6.57}$$

where $S0$ is a situation with exactly blocks A and B on the table. Intuitively, the second action $move(B, top\ A)$ is unsuccessful, because after the first action A is on B, and so B isn't clear. Suppose we provide by a suitable axiom that when the block to be moved is not clear or the destination place is not clear, then the situation is normally unchanged. Then $S2$ should be the same situation as $S1 = result(move(A, B), S0)$. However, simple circumscription of ab won't give this result, because the first move is only normally successful, and if the first move is unsuccessful for some unspecified reason, the second move may succeed after all. Therefore, circumscription of ab only gives a disjunction.

Clearly the priorities need to be arranged to avoid this kind of unintended "sneak disjunction". The best way to do it by imposing priorities isn't clear at the time of this writing.

6.22 GENERAL CONSIDERATIONS AND REMARKS

1. Suppose we have a data base of facts axiomatized by a formalism involving the predicate ab. In connection with a particular problem, a program takes a

subcollection of these facts together with the specific facts of the problem and then circumscribes *ab z*. We get a second order formula, and in general, as the natural number example of (McCarthy 1980) shows, this formula is not equivalent to any first order formula. However, many common sense domains are axiomatizable in such a way that the circumscription is equivalent to a first order formula. In this case we call the circumscription collapsible. For example, Vladimir Lifschitz (1985) has shown that this is true if the axioms are from a certain class he calls "separable" formulas. This can presumably be extended to other cases in which the ranges and domains of the functions are disjoint, so that there is no way of generating an infinity of elements.

Circumscription is also collapsible when the predicates are all monadic and there are no functions.

2. We can then regard the process of deciding what facts to take into account and then circumscribing as a process of compiling from a slightly higher level nonmonotonic language into mathematical logic, especially first order logic. We can also regard natural language as higher level than logic. However, as I shall discuss elsewhere, natural language doesn't have an independent reasoning process, because most natural language inferences involve suppressed premisses which are not represented in natural language in the minds of the people doing the reasoning.

Reiter has pointed out, both informally and implicitly in (Reiter 1982) that circumscription often translates directly into Prolog program once it has been decided what facts to take into account.

3. Circumscription has interesting relations to Reiter's (1980a) logic of defaults. In simple cases they give the same results. However, a computer program using default logic would have to establish the existence of models, perhaps by constructing them, in order to determine that the sentences mentioned in default rules were consistent. Such computations are not just selectively applying the rules of inference of logic but are metamathematical. At present this is treated entirely informally, and I am not aware of any computer program that finds models of sets of sentences or even interacts with a user to find and verify such models.

Circumscription works entirely within logic as Appendices A and B illustrate. It can do this, because it uses second order logic to import some of the model theory of first order formulas into the theory itself. Finding the right substitution for the predicate variables is, in the cases we have examined, the same task as finding models of a first order theory. Putting everything into the logic itself is an advantage as long as there is neither a good theory

of how to construct models nor programs that do it.

Notice, however, that finding an interpretation of a language has two parts — finding a domain and interpreting the predicate and function letters by predicates and functions on the domain. It seems that the second is easier to import into second order logic than the first. This may be why our treatment of unique names is awkward.

4. We are only part way to our goal of providing a formalism in which a database of common sense knowledge can be expressed. Besides sets of axioms involving ab, we need ways of specifying what facts shall be taken into account and what functions and predicates are to be taken as variable.

Moreover, some of the circumscriptions have unwanted conclusions, e.g. that there are no ostriches if none are explicitly mentioned. Perhaps some of this can be fixed by introducing the notion of present situation. An axiom that ostriches exist will do no harm if what is allowed to vary includes only ostriches that are present.

5. Nonmonotonic formalisms in general, and circumscription in particular, have many as yet unrealized applications to formalizing common sense knowledge and reasoning. Since we have to think about these matters in a new way, what the applications are and how to realize them isn't immediately obvious. Here are some suggestions.

When we are searching for the "best" object of some kind, we often jump to the conclusion that the best we have found so far is the best. This process can be represented as circumscribing $better(x, candidate)$, where $candidate$ is the best we have found so far. If we attempt this circumscription while including certain information in our axiom $A(better, P)$, where P represents additional predicates being varied, we will succeed in showing that there is nothing better only if this is consistent with the information we take into account. If the attempt to circumscribe fails, we would like our reasoning program to use the failure as an aid to finding a better object. I don't know how hard this would be.

6.23 APPENDIX A

CIRCUMSCRIPTION IN A PROOF CHECKER

At present there are no reasoning or problem-solving programs using circumscription. A first step towards such a program involves determining what

kinds of reasoning are required to use circumscription effectively. As a step towards this we include in this and the following appendix two proofs in EKL (Ketonen and Weening 1984), an interactive theorem prover for the theory of types. The first does the bird problem and the second a simple unique names problem. It will be seen that the proofs make substantial use of EKL's ability to admit arguments in second order logic.

Each EKL step begins with a command given by the user. This is usually followed by the sentence resulting from the step in a group of lines each ending in a semicolon, but this is omitted for definitions when the information is contained in the command. We follow each step by a brief explanation. Of course, the reader may skip this proof if he is sufficiently clear about what steps are involved. However, I found that pushing the proof through EKL clarified my ideas considerably as well as turning up bugs in my axioms.

1. $(DEFINE\ A$
$|\forall AB\ FLIES.A(AB, FLIES) \equiv$
$(\forall X.\neg AB(ASPECT1(X)) \supset \neg FLIES(X)) \wedge$
$(\forall X.BIRD(X) \supset AB(ASPECT1(X))) \wedge$
$(\forall X.BIRD(X) \wedge \neg AB(ASPECT2(X)) \supset FLIES(X)) \wedge$
$(\forall X.OSTRICH(X) \supset AB(ASPECT2(X))) \wedge$
$(\forall X.OSTRICH(X) \wedge \neg AB(ASPECT3(X)) \supset \neg FLIES(X))|NIL)$

This defines the second order predicate $A(ab, flies)$, where ab and $flies$ are predicate variables. Included here are the specific facts about flying being taken into account.

$; labels : SIMPINFO$
2. $(AXIOM$
$|(\forall X\ Y.\neg ASPECT1(X) = ASPECT2(Y)) \wedge$
$(\forall X\ Y.\neg ASPECT1(X) = ASPECT3(Y)) \wedge$
$(\forall X\ Y.\neg ASPECT2(X) = ASPECT3(Y)) \wedge$
$(\forall X\ Y.ASPECT1(X) = ASPECT1(Y) \equiv X = Y) \wedge$
$(\forall X\ Y.ASPECT2(X) = ASPECT2(Y) \equiv X = Y) \wedge$
$(\forall X\ Y.ASPECT3(X) = ASPECT3(Y) \equiv X = Y)|)$

These facts about the distinctness of aspects are used in step 20 only. Since axiom 2 is labelled SIMPINFO, the EKL simplifier uses it as appropriate

when it is asked to simplify a formula.

3. $(DEFINE\ A1$
$|\forall AB\ FLIES.A1(AB, FLIES) \equiv$
$A(AB, FLIES) \wedge$
$(\forall AB1\ FLIES1.A(AB1, FLIES1) \wedge (\forall Z.AB1(Z) \supset AB(Z)) \supset$
$(\forall Z.AB(Z) \equiv AB1(Z)))|NIL)$

This is the circumscription formula itself.

4. $(ASSUME\ |A1(AB, FLIES)|)$
deps : (4)

Since EKL cannot be asked (yet) to do a circumscription, we assume the result. Most subsequent statements list line 4 as a dependency. This is appropriate since circumscription is a rule of conjecture rather than a rule of inference.

5. $(DEFINE\ FLIES2|\forall X.FLIES2(X) \equiv BIRD(X) \wedge \neg OSTRICH(X)|NIL)$

This definition and the next say what we are going to substitute for the bound predicate variables.

6. $(DEFINE\ AB2$
$|\forall Z.AB2(Z) \equiv (\exists X.BIRD(X) \wedge Z = ASPECT1(X)) \vee$
$(\exists X.OSTRICH(X) \wedge Z = ASPECT2(X))|NIL)$

The fact that this definition is necessarily somewhat awkward makes for some difficulty throughout the proof.

7. $(RW\ 4\ (OPEN\ A1))$
$A(AB, FLIES) \wedge (\forall AB1\ FLIES1.A(AB1, FLIES1) \wedge$
$(\forall Z.AB1(Z) \supset AB(Z)) \supset (\forall Z.AB(Z) \equiv AB1(Z)))$
deps : (4)

This step merely expands out the circumscription formula. RW stands for "rewrite a line", in this case line 4.

8.$(TRW\ |A(AB, FLIES)|(USE\ 7))$
$A(AB, FLIES)$
deps : (4)

We separate the two conjuncts of 7 in this and the next step.

9. $(TRW\ |\forall AB1\ FLIES1.A(AB1, FLIES1)\wedge$
$(\forall Z.AB1(Z) \supset AB(Z)) \supset (\forall Z.AB(Z) \equiv AB1(Z))|$
$(USE\ 7))$
$\forall AB1\ FLIES1.A(AB1, FLIES1) \wedge (\forall Z.AB1(Z) \supset AB(Z)) \supset$
$(\forall Z.AB(Z) \equiv AB1(Z))$
deps : (4)

10. $(RW\ 8\ (OPEN\ A))$
$(\forall X.\neg AB(ASPECT1(X)) \supset \neg FLIES(X))\wedge$
$(\forall X.BIRD(X) \supset AB(ASPECT1(X)))\wedge$
$(\forall X.BIRD(X) \wedge \neg AB(ASPECT2(X)) \supset FLIES(X))\wedge$
$(\forall X.OSTRICH(X) \supset AB(ASPECT2(X)))\wedge$
$(\forall X.OSTRICH(X) \wedge \neg AB(ASPECT3(X)) \supset \neg FLIES(X))$
deps : (4)

Expanding out the axiom using the definition *a* in step 1.

11. $(ASSUME\ |AB2(Z)|)$
deps : (11)

Our goal is step 15, but we need to assume its premiss and then derive its conclusion.

12. $(RW11\ (OPEN\ AB2))$
$(\exists X.BIRD(X) \wedge Z = ASPECT1(X))\vee$
$(\exists X.OSTRICH(X) \wedge Z = ASPECT2(X))$
deps : (11)

We use the definition of *ab*.

13. $(DERIVE\ |AB(Z)|\ (12\ 10)\ NIL)$
$AB(Z)$
deps : (4 11)

This is our first use of EKL's DERIVE command. It is based on the notion of direct proof of (Ketonen and Weyhrauch 1984). Sometimes it can do rather complicated things in one step.

14. $(CI\ (11)\ 13\ NIL)$
$AB2(Z) \supset AB(Z)$
deps : (4)

We discharge the assumption 11 with the "conditional introduction" command.

15. $(DERIVE\ |\forall Z.AB2(Z) \supset AB(Z)|\ (14)\ NIL)$
$\forall Z.AB2(Z) \supset AB(Z)$
$deps : (4)$

Universal generalization.

16. $(DERIVE$
$|(\forall X.\neg AB2(ASPECT1(X)) \supset \neg FLIES2(X)) \wedge$
$(\forall X.BIRD(X) \supset AB2(ASPECT1(X))) \wedge$
$(\forall X.BIRD(X) \wedge \neg AB2(ASPECT2(X)) \supset FLIES2(X)) \wedge$
$(\forall X.OSTRICH(X) \supset AB2(ASPECT2(X))) \wedge$
$(\forall X.OSTRICH(X) \wedge \neg AB2(ASPECT3(X)) \supset$
$\neg FLIES2(X))|()(OPEN\ AB2\ FLIES2))$
$;(\forall X.\neg AB2(ASPECT1(X)) \supset \neg FLIES2(X)) \wedge$
$;(\forall X.BIRD(X) \supset AB2(ASPECT1(X))) \wedge$
$;(\forall X.BIRD(X) \wedge \neg AB2(ASPECT2(X)) \supset FLIES2(X)) \wedge$
$;(\forall X.OSTRICH(X) \supset AB2(ASPECT2(X))) \wedge$
$;(\forall X.OSTRICH(X) \wedge \neg AB2(ASPECT3(X)) \supset \neg FLIES2(X))$

This is another rather lengthy computation, but it tells us that $ab2$ and $flies2$ satisfy the axioms for ab and flies.

17. $(UE\ ((AB.|AB2|)\ (FLIES.|FLIES2|))\ 1\ NIL)$
$;A(AB2, FLIES2) \equiv$
$;(\forall X.\neg AB2(ASPECT1(X)) \supset \neg FLIES2(X)) \wedge$
$;(\forall X.BIRD(X) \supset AB2(ASPECT1(X))) \wedge$
$;(\forall X.BIRD(X) \wedge \neg AB2(ASPECT2(X)) \supset FLIES2(X)) \wedge$
$;(\forall X.OSTRICH(X) \supset AB2(ASPECT2(X))) \wedge$
$;(\forall X.OSTRICH(X) \wedge \neg AB2(ASPECT3(X)) \supset \neg FLIES2(X))$

Now we substitute $ab2$ and $flies2$ in the definition of A and get a result we can compare with step 16.

18. $(RW\ 17\ (USE\ 16))$
$;A(AB2, FLIES2)$

We have shown that $ab2$ and $flies2$ satisfy A.

19. $(DERIVE\ |\forall Z.AB(Z) \equiv AB2(Z)|\ (9\ 15\ 18)\ NIL)$
$;\forall Z.AB(Z) \equiv AB2(Z)$
$;deps : (4)$

9 was the circumscription formula, and 15 and 18 are its two premisses, so we can now derive its conclusion. Now we know exactly what entities are abnormal.

20. $(RW\ 8\ ((USE\ 1\ MODE:\ EXACT)$
$((USE\ 19\ MODE:\ EXACT)\ (OPEN\ AB2))))$
$;(\forall X.\neg(\exists X1.BIRD(X1) \wedge X = X1) \supset \neg FLIES(X)) \wedge$
$;(\forall X.BIRD(X) \wedge \neg(\exists X2.OSTRICH(X2) \wedge X = X2) \supset FLIES(X)) \wedge$
$;(\forall X.OSTRICH(X) \supset \neg FLIES(X))$
$;deps : (4)$

We rewrite the axiom now that we know what's abnormal. This gives a somewhat awkward formula that nevertheless contains the desired conclusion. The occurrences of equality are left over from the elimination of the aspects that used the axiom of step 2.

21. $(DERIVE\ |\forall X.FLIES(X) \equiv$
$BIRD(X) \wedge \neg OSTRICH(X)|\ (20)\ NIL)$
$;\forall X.FLIES(X) \equiv BIRD(X) \wedge \neg OSTRICH(X)$
$;deps : (4)$

DERIVE straightens out 20 to put the conclusion in the desired form. The result is still dependent on the assumption of the correctness of the circumscription made in step 4.

Clearly if circumscription is to become a practical technique, the reasoning has to become much more automatic.

6.24 APPENDIX B

Here is an annotated EKL proof that circumscribes the predicate $e(x, y)$ discussed in section 6.

$(proof\ unique\ names)$

What the user types is indicated by the numbered statements in lower case. What EKL types is preceded by semicolons at the beginning of each line and is in upper case. We omit EKL's type-out when it merely repeats what the command asked it to do, as in the commands DERIVE, ASSUME and DEFINE.

1. $(axiom\ |index\ a = 1 \wedge index\ b = 2 \wedge index\ c = 3 \wedge index\ d = 4|)$

Since EKL does not have attachments to determine the equivalence of names, we establish a correspondence between the names in our domain and some natural numbers.

2. $(derive|\neg(1 = 2) \wedge \neg(1 = 3) \wedge \neg(2 = 3) \wedge \neg(1 = 4) \wedge \neg(2 = 4) \wedge \neg(3 = 4)|)$

EKL does know about the distinctness of natural numbers, so this can be derived.

$(der - slow)$

We have to tell EKL to use the properties of equality rather than regarding it as just another predicate symbol in order to do the next step. Sometimes this leads to combinatorial explosion.

3. $(derive\ |a \neq\ b|\ (1\ 2))$

This shows that two names themselves are distinct.

4. $(define\ equiv\ |\forall e.equiv\ e \equiv (\forall x.e(x,x)) \wedge (\forall x\ y.e(x,y) \supset e(y,x)) \wedge (\forall x\ y\ z.e(x,y) \wedge e(y,z) \supset e(x,z))|)$

Here we use second order logic to define the notion of equivalence relation. The first word after "define" is the entity being defined and included between vertical bars is the defining relation. EKL checks that an entity satisfying the relation exists.

5. $(define\ ax\ |\forall e.ax\ e \equiv e(a,b) \wedge equiv\ e|)$

We define ax as a predicate we want our imitation equality to satisfy. We have chosen a very simple case, namely making a and b "equal" and nothing else.

6. $(define\ ax1|\forall e.ax1\ e \equiv ax\ e \wedge \forall e1.(ax\ e1 \wedge (\forall\ x\ y.e1(x,y) \supset e(x,y))$
$\supset (\forall x\ y.e(x,y) \equiv e1(x,y)))|)$

This defines $ax1$ as the second order predicate specifying the circumscription of ax.

7. $(assume\ |ax1(e0)|)$
$(label\ circum)$

We now specify that $e0$ satisfies $ax1$. It takes till step 17 to determine what $e0$ actually is. When EKL includes circumscription as an operator, we may be able to write something like circumscribe$(e0, ax1)$ and make this step occur. For now it's just an ordinary assumption.

8. $(define\ e2\ |\forall x\ y.e2(x,y) \equiv (x = a \wedge y = b) \vee (x = b \wedge y = a)$
$\vee x = y|)$

The predicate $e2$ defined here is what $e0$ will turn out to be.

9. $(derive\ |equive2|\ nil\ (open\ equiv)\ (open\ e2))$

Now EKL agrees that $e2$ is an equivalence relation. This step takes the KL-10 about 14 seconds.

10. $(derive\ |ax\ e2|\ (9)\ (open\ ax)\ (open\ e2))$
$(label\ ax\ e2)$

Moreover it satisfies ax.

11. $(rw\ circum\ (open\ ax1))$
$; AX(E0) \wedge (\forall\ E1.AX(E1) \wedge (\forall X\ Y.E1(X,Y) \supset E0(X,Y)) \supset$
$; (\forall X\ Y.E0(X,Y) \equiv E1(X,Y)))$
$; deps : (CIRCUM)$

A trivial step of expanding the definition of $ax1$. EKL tells us that this fact depends on the assumption CIRCUM. So do many of the subsequent lines of the proof, but we omit it henceforth to save space.

12. $(trw\ |ax\ e0|\ (use\ 11))$
; $AX(E0)$

The first conjunct of the previous step.

13. $(rw\ 12\ (open\ ax\ equiv))$
$(label\ fact1)$
; $E0(A,B) \wedge (\forall X.E0(X,X)) \wedge (\forall X\ Y.E0(X,Y) \supset E0(Y,X)) \wedge$
; $(\forall X\ Y\ Z.E0(X,Y) \wedge E0(Y,Z) \supset E0(X,Z))$

We expand $ax(e0)$ according to the definitions of ax and $equiv$.

14. $(derive\ |\forall p\ q\ r.(p \vee q \supset r) \equiv (p \supset r) \wedge (q \supset r)|)$
$(label\ rewrite\ by\ cases)$

This is a fact of propositional calculus used as a rewrite rule in the next step. A program that can use circumscription by itself will either need to generate this step or systematically avoid the need for it.

15. $(trw\ |e2(x,y) \supset e0(x,y)|$
$((open\ e2)\ (use\ rewrite\ by\ cases\ mode:\ always)\ (use\ fact1)))$
; $E2(X,Y) \supset E0(X,Y)$

This is the least obvious step, because rewrite by cases is used after some preliminary transformation of the formula.

16. $(derive\ |\forall x\ y.e0(x,y) \equiv e2(x,y)|\ (ax\ e2\ 11\ 15))$

DERIVE is substituting $e2$ for the variable $e1$ in step 11 and using the fact $ax(e2)$ and step 15 to infer the conclusion of the implication that follows the quantifier $\forall e$.

17. $(rw\ 16\ (open\ E2))$
; $\forall X\ Y.E0(X,Y) \equiv X = A \wedge Y = B \vee X = B \wedge Y = A \vee X = Y$
; $deps:\ (CIRCUM)$

Expanding the definition of $e2$ tells us the final result of circumscribing $e0(x,y)$. A more complex $ax(e0)$ — see step 5 — would give a more complex result upon circumscription. However, it seems that the proof would be similar. Therefore, it could perhaps be made into some kind of macro.

6.25 Acknowledgments

I have had useful discussions with Matthew Ginsberg, Benjamin Grosof, Vladimir Lifschitz and Leslie Pack[4]. The work was partially supported by NSF and by DARPA. I also thank Jussi Ketonen for developing EKL and helping me with its use. In particular he greatly shortened the unique names proof.

[4]Leslie P. Kaelbling

Chapter 7

COMMON SENSE THEORIES OF THE WORLD

7.1 Ontology—or what there is

There are more things in heaven and earth, Horatio,
Than are dreamt of in your philosophy. - Shakespeare

I suspect that there are more things in heaven and earth than are dreamed of, or can be dreamed of, in any philosophy. - J. B. S. Haldane

For us ontology concerns what kinds of entities our theories will consider. There are many more than have been used in AI theories or physical theories up to now. Many important kinds of entity are ill-defined and approximate. Shakespeare and Haldane may be right, but we'll do our best.
[1]

[1]Ontology began as a branch of philosophy concerning what entities exist, e.g. whether material objects, ideas, or numbers exist. The philosopher W.V.O. Quine defined the ontology of a theory to be the range of its bound variables. Quine's notion also works well for logical AI and computer science generally.

Recently ontology has become the focus of much study in computer science, especially in AI. The result has been interaction among computer scientists, linguists and philosophers.

Philosophers, and now computer scientists, often try to minimize the kinds of objects admitted to exist and build other entities from these. Common sense thinking uses redundant ontologies. I suspect common sense is right.

In ontology common sense is quite liberal in three important senses, and logical AI needs to be equally liberal.

The ontology of common sense includes *rich entities*. A rich entity cannot be completely described, because it involves an infinite amount of information. While rich entities cannot be completely described we can know facts about them. It's just that however many facts we have, it isn't a complete description. The prototypical rich entity is a situation as discussed in (McCarthy and Hayes 1969a). In that article as situation is taken to be the complete state of the world at some time. We can't know it, but we can know a lot about it. A person is also a rich entity. No matter how much we know about a person, there is more to be learned.

All kinds of concepts are represented in natural language by nouns and many are very approximate. Indeed, outside of mathematics and the mathematical formalizations of other sciences, almost no entities are completely precisely defined.

Of course, imprecision often leads people into trouble, but total precision is impossible, and efforts to make a concept more precise than necessary are often lead to useless hairsplitting.

> Our discussion will be adequate if it has as much clearness as the subject matter admits of, for precision is not to be sought for alike in all discussions, any more than in all the products of the crafts.— Aristotle, *Nicomachean Ethics*

The usual ontology of the *blocks world* includes situations, blocks and often locations. However, if a program is to design structures in the blocks world, then the ontology must contain structures, designs for structures and specifications that a structure must meet.

Intentional objects like beliefs, purposes and intentions are needed if the agent is to ascribe beliefs, etc. to other agents or to think about its own methods.

Ontology relevant to specific areas of AI are conveniently discussed in the sections devoted to them.

I am not an adherent of any of the doctrines that have been proposed and am dubious about the current efforts for standardization. Until more has been said about what kinds of axioms are wanted, proposals concerning what objects to admit into the ontology are hard to evaluate.

7.2 Situations and events

7.2.1 Situation calculus - the language

Much intelligent activity is the achievement of goals by an agent, call it *Joe*, performing a sequence of actions—or, more generally, by a strategy of action. In general, there are other agents and there are natural events. Also some of the actions and other events are continuous and concurrent. We begin with the simplest case—in which there is one agent and the actions are discrete and have definite results.

Situation calculus is a formalism for describing the changes in situations resulting from events, especially actions of agents. We first discuss *simple situation calculus* (sitcalc) for which the basic equation is

$$s' = Result(e, s), \tag{7.1}$$

giving the new situation s' that results when event e occurs in situation s. Notice that events are regarded as discrete and having a single new situation as a result. This is a very important case, perhaps the most important, but not the most general. In particular, it doesn't conveniently deal with continuous change or concurrent events for which modified formalisms are required. We use *sitcalc* to refer to this simple situation calculus.

This section is devoted to the language of sitcalc. Reasoning in sitcalc is in the following sections.

The basic entities of sitcalc are situations and *fluents*. In the original papers, situations were taken to be snapshots of the whole world at a given time. However, this is not the only useful interpretations, so we will treat sitcalc as an abstract theory.

A *fluent* is a function whose domain is a space of situations. Three kinds of fluent appear in the literature—according to the range of the function. *Propositional* fluents have truth values as values. These are the most used, and some treatments of situation calculus use only propositional fluents. *Term* fluents have values in other spaces, e.g. numbers or locations. *Situational* fluents have situations as values. [2]

[2]The situation calculus was first proposed in (McCarthy 1963) and in the much more widely read (McCarthy and Hayes 1969a). It was not named in those papers. The term *fluent* is taken from Newton, and the sitcalc usage isn't too far from his. [3] Many AI researchers take actions as the only allowed first argument of the *Result*. Considering events other than actions offers no formal difficulties.

Even though time is continuous, and events are often concurrent, and there are many actors in general, sitcalc is useful. It is a special case, but human reasoning often involves this special case. For example, in planning, a trip often is regarded as a sequence of actions by the traveller. Each step of the journey or auxiliary action like buying a ticket can conveniently be considered as resulting in a single new situation. A travel agent usually provides only sequential plans, although there may be minor concurrencies like reconfirming a flight. Such plans rely on the *benevolence of institutions*. For example, airlines always provide information about how to find the gate for a continuing flight.

However, the human analog of sitcalc has an *elaboration tolerance* that most axiomatic sitcalc formalisms lack. Suppose all the events in the travel plan are sequential except that the traveler's laundry is washed by the hotel while the traveler is out buying a ticket or attending to some other business. The laundry is promised to be available at 6pm. This one concurrent consideration doesn't require a complete way for a person to think about the whole trip, and AI also requires formalisms that tolerate this kind of elaboration while preserving the simplicity of sitcalc in treating the parts of the trip that do not require concurrency.

7.2.2 Ontologies for the situation calculus

Here are some kinds of entity.

situation The domain $\mathcal{S}$ of situations is both the second argument and the range of $Result(e, s)$. But what is a situation? There are various useful interpretations. According to (McCarthy 1963) and (McCarthy and Hayes 1969b), a situation is a snapshot of the world at an instant of time. However, none of the situation calculus theories enforce this interpretation and some don't even allow it. The original idea was that a situation is a *rich* entity, and neither a person nor a robot could ever completely know a situation, although one could know facts about a situation. It is still often useful to regard situations as rich entities.

The most limited notion of situation is as a set of values of a known finite set of variables, e.g. the set of locations of 5 blocks. In this interpretation $s' = Result(e, s)$ is a new set of locations of the blocks.

Variable ranging over situations are denoted by s, with decorations such as primes and subscripts. Situation constants are denoted by $S0$,

$S1$, etc. In general, lower case is used for variables and upper case for constants, both for individuals and for functions and predicates.

event Events occur in situations and result in new situations. The letter e is used. Events also may be rich entities, but in most formalizations they are taken from a finite or parametrized set.

propositional fluent Propositional fluents take as arguments situations and some other entitities. They may be written as predicates directly as in $On(Block1, Block2, s)$ asserting that $Block1$ is on $Block2$ in situation s. Or they may be *reified* leading to writing $Holds(On(Block1, Block2), s)$.

term fluent It is often convenient to use terms like $Location(Block1, s)$. The possible notations are analogous to those for propositional fluents. We can write $Location(Block1, s) = Block2$ or $Location(Block1, s) = Top(Block2)$ if we want make a clean distinction between blocks and locations. Reifying further, we get $Value(Location(Block1), s) = Top(Block2)$.

1. $On(Block1, Block2, s)$ is a propositional fluent interpreted as asserting that $Block1$ is on $Block2$ in situation s. In theories with reified fluents, it is replaced $Holds(On(Block1, Block2), s)$.

2. $Holds(On(Block1, Top(Block2)), s)$

3. $Result(Move(Block1, Top(Block3)), s)$

The other formulas have the same interpretation but differ in the extent of *reification*.

7.2.3 Towers of Blocks and other structures

AI study of the blocks world has concentrated (maybe entirely) on planning the rearrangement of some piles of blocks on a table into another collection of piles of blocks. We want to go beyond that and consider building, moving and dismantling compound objects. Here are some considerations.

1. Suppose we define a *traffic light* as tower consisting of a red block above a green block with the latter at least three blocks above the ground. Besides the blocks specified to be red or green, no other blocks are red or green.

2. An obvious task is to build a traffic light. The simplest planner uses a greedy algorithm. It looks for a block to be the base which must be neither red nor green. Then it looks for the second block, etc. When it chooses the first block, it may place the block right away or continue with the construction plan and finish it before moving any blocks.

3. We can therefore consider the design of a traffic light, making a construction plan for a traffic light and actually building it as distinct tasks. Less concrete than a design is the *specification* of a traffic light.

4. There are also partially built traffic lights and damaged traffic lights. Maybe we shall also want to consider *future traffic lights*.

5. We can move a traffic light from one location to another. Our axioms may allow moving them as a unit or it may require dismantling the traffic light and moving it block by block. Even if dismantling is required, we can still have the action of moving it in our ontology.

6. It would seem that these entities are not independently defined but rather are derived from a basic concept of traffic light by some operations.

7. The design of a traffic light may be carried out by a sequence of operations specified in situation calculus. However, the operations available in design are different from those available in construction. For example, it is possible to decide that a red block goes on top before deciding about the blocks that go below it.

Our goal is to formalize the specification, design, construction planning, building, dismantling and moving of compound objects, i.e. objects made of parts connected in specific ways. Future objects, broken object and partially constructed objects may also be wanted.

Patrick Winston, (Winston 1970) and (Winston 1992), wrote programs to recognize arches, where an arch consists of two blocks supporting a third. We will also consider designing and building Winston type arches.

We begin with the design of a traffic light.

One approach is to define a traffic light as a sentence, e.g.

$$\begin{aligned} &On(A1, Top(A2)) \wedge Red(A1) \wedge Green(A2) \wedge On(A2, Top(A3)) \wedge \\ &Brown(A3) \wedge On(A3, Top(A4)) \wedge Brown(A4) \wedge On(A4, Top(A5)) \wedge \\ &Brown(A5) \wedge On(A5, Table). \end{aligned} \tag{7.2}$$

Within a set theoretic formalization of the blocks world, we can define a traffic light by

$$TrafficLight(obj) \equiv Islist(obj) \wedge (\forall x)(x\ In\ obj \rightarrow On(x, qifx = last(obj)qthenTableqelseNext(x, obj))). \tag{7.3}$$

Here we are assuming that the set theory has operations on lists.

Definition: A *simple structure* is a set of ground atomic formulas.

Simple structures occur in model theory where they are called diagrams and are part of the metamathematics. In computer science, they are called *relational databases* when they are explicitly represented in a computer system. We consider them more abstractly,i.e. not necessarily explicitly represented.

We need simple structures as objects in the theory itself, i.e. not just in the metamathematics. If ss is a simple structure, $Objects(ss)$ is the set of arguments of the atoms. We do not require that $Objects(ss)$ be finite or that it be known. In our applications, $Objects(ss)$ will usually be finite, but will sometimes be unknown or variable. $Relations(ss)$ is the set of relation symbols in ss. It also need not be finite or known, although in our applications it will be finite and known. We have the wff $SimpleStructure(ss)$.

Finite simple structure constants are written like

$$Ss1 = \{ON(A1, A2), ON(A2, A3), ON(A3, A4), ON(A4, A5), ON(A5, TABLE),\\ RED(A1), GREEN(A2), BROWN(A3), BROWN(A4), BROWN(A5), A6\}.$$

Here $A6$ is an object in $Ss1$ that doesn't happen to be involved in any of the relations of $Ss1$.

$$Cb1: \qquad On(x, y, S1) \equiv Application(ON, Name(x), Name(y)) \in Ss1.$$

Why all the fuss about On and ON, etc.?

The idea is the $On(x, y)$ is a relation between blocks in the world as asserted in a certain context, whereas $ON(A1, A2)$ is a member of an abstract structure.

7.2.4 Narratives

> In M-, an important town in northern Italy, the widowed Marquise of O-, a lady of unblemished reputation and the mother of several well-brought-up children, inserted the following announcement in the newspapers: that she had, without knowledge of the cause, come to find herself in a certain situation; that she would like the father of the child she was expecting to disclose his identity to her; and that she was resolved, out of consideration for her family, to marry him.

—the first sentence of *The Marquise of O-* by Heinrich von Kleist, 1821.

> Jane saw the puppy in the window of the pet store. She pressed her nose against the glass.

—a sentence giving trouble to a natural language understanding program.

A narrative is an account of what happened. We treat it by giving some situations and some events and some facts about them and their relations. Situations in a narrative are partially ordered in time. The real situations may be totally ordered, but the narrative does not include full information about this ordering. Thus the temporal relations between situations in different places need only be described to the extent needed to describe their interactions.

7.2.5 Induction in situation calculus

We do not wish to presume a unique initial situation $S0$ from which all situations are reachable by results of some events. Therefore, our induction is in terms of a predicate $reachable(s0, s)$. The axiom is the second order sentence

$$\begin{array}{l}(\forall s0)(reachable(s0, s0) \\ \wedge \\ (\forall s\ e)(reachable(s0, s) \rightarrow reachable(s0, result(e, s))) \\ \wedge \\ (\forall P)(P(s0) \\ \wedge(\forall s\ e)(P(s) \rightarrow P(result(e, s))) \\ \rightarrow (\forall s)(reachable(s0, s) \rightarrow P(s))))\end{array} \tag{7.4}$$

7.2.6 Two dimensional entities

The world is three dimensional, but we can know more about two dimensional objects.

As a major example, we consider the relation between the surface of the earth (or of limited regions), maps and what people can learn and know about the surface with limited opportunities to observe.

How to express the fact that any region, e.g. California, has more detail than any map can express?

7.3 Three dimensional objects

We and our robots live in a world of three dimensional objects. What we[4] can know about objects is severely limited by our ability to observe them and to represent information about them internally. What we can do with these objects is also limited. This poses severe problems for AI systems to describe and manipulate objects. We begin with an informal treatment.

What can people do with 3-dimensional objects that we can make robots do?

An object has three important and related aspects.

1. What it is. This includes parts that cannot be observed, either for the present or for the indefinite future.

2. What we can perceive about it and learn about it.

3. What we can remember and communicate about it.

Here are some considerations.

1. What we humans can discover about an object from observation is limited by the senses we happen to have. Humans have vision over a limited range of wavelengths. Even within that range an infinite dimensional space of reflectivities as a function frequency is compressed into a three dimensional space. Humans have touch and a very limited (compared to many animals) sense of smell. Humans are slightly sensitive to infrared through our skins but don't have anything the pit viper's "third eye". We don't have a bat's sonar abilities. Imagine what

[4] "We" sometimes refers both to people and robots.

it would be like to see each others' innards. "Oh, you can recognize George by his large flat liver".

Robots need not be limited to human senses. We can give them a wider electromagnetic spectrum, sensitivity to particular lines, and more than three dimensions of color resolution. They can have eyes further apart and eyes on the ends of arms for taking a close look. They can have sonar at a variety of frequencies and phased arrays of microphones for ears. They can have mechanically sensitive antennas at resolutions all the way down to that of the atomic force microscope. They can have gigabytes and perhaps eventually moles of memory to store their sensory informations and what they compute with it.

However, present robots are even more limited sensorily than people. In particular, no-one has gotten much from the sense of touch, e.g. what permits me to pick particular objects out of a pocket.

2. Our perception depends on memory of the object in question and of similar objects. More generally, it is dependent on theories of objects, e.g. the relations between a person's hands and his feet.

3. The objects in the environment are often quite complex, and are sometimes even more complex than they usually are. Human object perception depends on the objects we successfully perceive and manipulate having features that commonly exist and are helpful. These features include

   - Reflectance properties resulting an apparent color that is reasonably discriminable under normally varied lighting.
   - Reasonable shapes that don't change too fast.

7.3.1 Representation of 3-dimensional objects

There are two cases. We may need to represent an object located in space or reperesent the object apart from any particular locatition.

1. A point in space. Even if we represent the object by a point, we can attach attributes to it, e.g. color, material and even size.

2. A rectangular parallelopiped with sides parallel to some axes we are using.

3. A rectangular parallelopiped in no particular orientation can be appropriate when the object is not located in space.

4. Sometimes objects, especially artificial objects, have definite geometric shapes - or are as close to those shapes as the purpose warrants. This is an easy case, especially for computers.

5. A hard case is an unfamiliar animal—say a badger.

Chapter 8

Situation Calculus

[This chapter Latexed May 30, 2004]

8.1 What is situation calculus?

The term *situation calculus*, abbreviated *sitcalc*, has been used variously in the AI literature. Moreover, various formalisms have been developed under that name.

Here's a characterization that covers all the usages I know about, but many of the proposed definitions are more narrow.

The ontology includes situations, fluents and events. Usually the events are actions of some agent, but an action is just a special kind of event.

Situations are states of affairs. In the original formulation, (McCarthy 1963) and (McCarthy and Hayes 1969a), they were advertised as states of the whole world. As such situations could not be fully described; one could only say something about a situation, and this is done with fluents.[1]

There are propositional fluents that can be true or false in a situation, e.g. It-is-raining(s) asserts that it is raining in situation s. Term fluents have values in a situation, e.g. Color(B1, s) = Red or Height(B1, s) = 3.5cm. The occurrence of events in a situation produces new situations. In most sitcalc theories situations are not so grand as to be states of the world but rather partial states of some part of it.

[1]Entities about which one can have only partial information are called *rich*. There is a general discussion of them in Chapter 4.

Convention: Symbols denoting constants are capitalized and symbols denoting variables are lower case. This applies to objects, to functions and to predicates. Free variables in formulas are generally taken to be universally quantified unless the formula itself is the object of discussion.

Situation calculus is used to express theories about states of affairs and the effects of actions and other events.

Such theories were first proposed in (McCarthy 1963) and further developed in (McCarthy and Hayes 1969a), but the term *situation calculus* was not defined in those papers, so I can't claim authority for my precise use of the term. There is a good treatment in (Shanahan 1997).

As characterized above, situation calculus can include concurrent events and continuously acting events, and we will cover these topics in this chapter. However, most situation calculus theories in the AI literature have been much narrower and have conformed to what I now want to call *sequential situation calculus*.

Sequential situation calculus is based on a function Result, where Result(e, s) is the new situation that results when event e occurs in situation s. In the literature, e is usually an action, and in that case we write

$$s' = \text{Result}(a, s). \tag{8.1}$$

We begin with sequential situation calculus.

Raymond Reiter (Reiter 2001) defined an important form of situation calculus, I'll call Bbsitcalc. Bbsitcalc, there is a base situation $S0$, and all situations arise from $S0$ by iteration of the function $Result(a, s)$. Moreover, situations arising from distinct sequences of actions are regarded as distinct.[2] Reiter defined a planning system called Golog that has been used for robot control by the Toronto group.

8.2 Basics of the sequential situation calculus

Situations can be regarded as states of affairs. In (McCarthy and Hayes 1969a) they were taken as states of the whole world. A key idea is that you can't fully describe a state of the world, but you can say something about a state

[2]Reiter gave his version the name "situation calculus", but it's not the only important possibility. Therefore, I call his verision "big bang sitcalc", taking $S0$ as analogous to the "big bang" of current cosmological theories."

of the world. It is sometimes convenient to make theories in which situations represent much more modest states. A key point is that one nevertheless expresses the theories in terms of facts about situations rather than by fully specifying the situation even when this is possible. This makes the theory more elaboration tolerant, because new fluents can be added by adding sentences rather than by modifying a definition of situation.

Facts about situations are expressed using *fluents*, a term taken from Newton, who, it seems to me, meant something similar. A *propositional fluent* is a predicate on situations.

Here are some examples.

1. It-is-raining(s) asserts that it is raining in the situation s. Clearly this makes sense only when the situations refer to a somewhat localized state of affairs.

2. $On(B1, B2, s)$ asserts that block 1 is on block 2 in situation s.

3. $At(block, l, s)$ asserts that *block* is at location l in situation s.

4. $Red(B1, s)$ asserts that $B1$ is red in situation s.

We also have ordinary predicates and functions of objects. When we assert $Red(B1)$ we are making an assertion that is not situation dependent. Intuitively, $B1$ is red in all situations. $Color(B1) = Red$ is another way of making the same assertion.

Truth values of propositions do not change in situation calculus. When something changes, the fact that it depends on the situation must be explicit. When fluents and ordinary predicates are used in the same theory, we will use different symbols for them. Thus $Red(B1)$ and $Red(B1, s)$ will not appear in the same theory, although in principle, predicates of one argument and predicates of two arguments could use the same names without ambiguity. We will have use for the function $Top(block)$ designating the top of a block as a locations.

Term fluents take values other than truth values. Which kinds of values are allowed depends on the specific situation calculus theory.

Here are some examples.

1. $Location(B1, s)$. With this and the function Top, we can write equations like $Location(B1, s) = Top(B2)$. Using Top may strike some

people as finicky. We might just write $Location(B1, s) = B2$ to mean the same thing. However, we often have good reason to consider locations and blocks to be of different sorts; sometimes we need to quantify over the set of blocks and sometimes over the set of locations.

2. $Color(B1, s) = Red$ makes a clear assertion. We can also have $Height(B1, s) =$ 3.2**cm**.

Events occur in situations, and we use e for a variable ranging over events. In the simplest sitcalc theories that have been studied, the events are actions. In this case we use the variable a. In these simple theories, there is only one actor, so we can write actions like $Move(B1, Top(B2))$ for an action in the much studied blocks world theories. If we want to distinguish actors, we can have the event $Does(George, Move(B1, Top(B2)))$.

Here are further examples of events and actions.

1. $Paint(B1, Red)$ is the action of painting $B1$ red.

The main function of situation calculus is $Result(e, s)$.

$$s' = Result(e, s) \tag{8.2}$$

asserts that s' is the situation that results when the event e occurs in situation s.

Sequential sitcalc theories are based on the function *Result* and present very restricted models of events. Each event produces a single new situation, there are no continuous processes, and events are not concurrent. Nevertheless, sequential sitcalc, which most writers call situation calculus without qualification, has been a powerful tool in AI. Moreover, it corresponds to the way people think about many problems. "Doing $A1$ creates the preconditions for action $A2$, which in turn creates the precondition for $A3$, etc. (Reiter 2001) presents a situation calculus theory for which there is a problem solving system called Golog.

8.2.1 A simple blocks world theory $\mathcal{T}_{bl1}$

There are three blocks, $B1$, $B2$, and $B3$.

There are locations $Table$, $Top(B1)$, $Top(B2)$, $Top(B3)$.

There is an initial situation $S0$. The locations of the blocks are characterized by the sentence

$$At(B1, Table, S0) \wedge At(B2, Table, S0) \wedge At(B3, Top(B2), S0). \quad (8.3)$$

The blocks also have colors that depend on the situation. These are given by the sentence

$$Color(B1, S0) = Red \wedge Color(B2, S0) = Red \wedge Color(B3, S0) = Black. \quad (8.4)$$

We have two actions $Move(block, location)$ and $Paint(block, color)$.

Their effects are *partly* characterized by the equations

$$PrecondMove(block, l, s) \rightarrow At(block, l, Result(Move(block, l), s)) \quad (8.5)$$

and

$$PrecondPaint(block, s) \rightarrow Color(block, Result(Paint(block, color), s)) = color. \quad (8.6)$$

Suppose our goal is to make a traffic light, consisting of a red block on a green block on a black block.

Consider the situation

$$\begin{array}{l} S' = Result(Move(B1, Top(B2)), Result(Move(B2, Top(B3)), \\ Result(Move(B3, Table), Result(Paint(B2, Green), S0)))). \end{array} \quad (8.7)$$

We can also use an abbreviated notation

$$\begin{array}{l} S' = Result(\\ \qquad Paint(B2, Green); \\ \qquad Move(B3, Table); \\ \qquad Move(B2, Top(B3)); \\ \qquad Move(B1; Top(B2)) \\ \qquad\quad , S0). \end{array} \quad (8.8)$$

The abbreviated notation is easier to read and write but would require a more complicated axiomatization if it were part of the language. Therefore, we consider it just as a *syntactic abbreviation*. Then we don't have to axiomatize it but merely to provide for expanding any instances of it into the official

notation. Thus (8.8) expands to (8.7), and we only prove statements about (8.8). [3]

We want S' to have the desired properties, namely

$$\begin{array}{l} Color(B1, S') = Red \wedge Color(B2, S') = Green \wedge Color(B3, S') = Black \\ \wedge On(B1, B2, S') \wedge On(B2, B3, S') \wedge On(B3, Table, S').\wedge \end{array} \tag{8.9}$$

However, (8.9) won't follow logically until we make quite a few additional assumptions.

Here are some assumptions.

1. The block names—$B1$, $B2$ and $B3$—denote distinct, unique objects. We can express this with the sentence

$$B1 \neq B2 \wedge B1 \neq B3 \wedge B2 \neq B3. \tag{8.10}$$

   It is customary to use the syntactic abbreviation $UNA(B1, B2, B3)$, due to Baker, which expands to (8.10) to express distinction. (UNA stands for "unique names assumption".). UNA is not a function in the logic, because $UNA(B1, B2, B3)$ does not depend on the values of the symbols $B1$, $B2$ and $B3$ but is an assertion about the symbols themselves. Thus $UNA(A, A)$ does not make sense, because it expands to $A \neq A$.

   $UNA(\alpha_1, \ldots, \alpha_n)$ abbreviates the conjunction of $\frac{1}{2}n(n-1)$ sentences. If we have n objects to discuss, it is tedious to write $\frac{1}{2}n(n-1)$ sentences. [4]

[3] Suppose $A1$, $A2$, $A3$, and $A4$ are four actions. It will always be true that

$$Result(A3; A4, Result(A1; A2), s) = Result(A1; A2; A3; A4, s),$$

but the only way to prove it is to expand the formulas into compositions of single actions.

[4] We can get by with a number of sentences proportional to n in any several ways.

1. Introduce a relation G(x,y) satisfying the axioms

$$\begin{array}{l} G(x, y \wedge G(y, z) \rightarrow G(x, z), \\ \neg G(x, x) \\ G(B_1, B_2) \wedge G(B_2, B_3) \wedge \ldots \wedge G(B_{n-1}, B_n). \end{array} \tag{8.11}$$

This forces the B_i to be all different.

2. We often need to delimit the set of blocks. The most straightforward way to do this is to write

$$b = B1 \vee b = B2 \vee b = B3, \tag{8.15}$$

where b is a variable of sort **Block**. This assumes we are using a sorted

2. It can also be done with a sentence

$$Index(B_1) = 1 \wedge \ldots \wedge Index(B_n) = n, \tag{8.12}$$

where we presuppose axioms for the natural numbers assuring that they are all different.

3. If we use a function $B : Z^+ \rightarrow Blocks$, and denote blocks by $B(1)$, $B(2)$, etc., we get by with asserting

$$i \neq j \rightarrow B(i) \neq B(j)$$

and the axioms required to assure that the natural numbers are all distinct, which we are likely to want anyway.

4. If we use set theory, we can write

$$Card(\{B1, B2, B3, B4, B5\}) = 5. \tag{8.13}$$

Here $\{B1, B2, B3, B4, B5\}$ denotes the set whose elements are $B1$,$B2$,$B3$,$B4$, and $B5$, and $Card(u)$ denotes the cardinality of the set u.

5. We also need axioms like

$$Top(b1) = Top(b2) \rightarrow b1 = b2$$

and

$$Top(b) \neq Table.$$

6. More generally, we need axioms to insure that expressions that should denote distinct objects actually do.

Prolog has unique names built into its structure and doesn't require unique names axioms. This simplifies some Prolog programs and complicates those where different symbols may denote the same object. Consider the sentence

$$Card(\{B1, B2, B3, B4, B5\}) = 4, \tag{8.14}$$

which tells us that two of the blocks are the same but doesn't assert which two. This tells us that it is sometimes better to assert unique names by axioms than by syntactic convention.

Assuming that we want set theory anyway, the set theoretic formula (8.13) is the most compact way of asserting unique names.

logic. If we don't want to assume a sorted logic, this becomse

$$IsBlock(b) \rightarrow b = B1 \vee b = B2 \vee b = B3. \tag{8.16}$$

However, this assumes that $B1$, $B2$, and $B3$ are all the blocks in the world, which precludes elaborating $\mathcal{T}^{bl1}$ to include another table with additional blocks. Therefore, we prefer to write

$$\text{Block}(x, bl1) \rightarrow x = B1 \vee x = B2 \vee x = B3; \tag{8.17}$$

With set theory we can write

$$\text{Blocks}(bl1) = \{B1, B2, B3\}. \tag{8.18}$$

In either case, that a block be present needs to be part of the precondition for moving it, painting it and for its top to be a location to which things can be moved.

3. Our axioms for moving blocks will often require that the block's top be clear and also that the location to which we wish to move the block be clear. In the given example with blocks $B1$, $B2$ and $B3$, we can assure that the $B1$ and $B2$ are clear in several ways.

   1. The most straightforward way is to introduce a predicate $Clear(b)$ asserting that block b is clear and axiomatizing it in general by

   $$Clear(b) \equiv (\forall y)(\neg At(y, Top(b)). \tag{8.19}$$

   and in the particular case by

   $$Clear(B1) \wedge Clear(B2). \tag{8.20}$$

   2. $Clear(x)$ can also be introduced as a syntactic abbreviation for $(\forall y)(\neg At(y, Top(b))$, which keeps the model theory more compact, but I prefer the explicit predicate.

4. The precondition for moving a block x to a location l is that both $Top(x)$ and l be clear, i.e.

$$Precond(Move(x,l),s) \equiv Clear(Top(x),s) \wedge Clear(l,s). \quad (8.21)$$

There is no precondition for painting a block in the simple blocks world $\mathcal{T}^{bl1}$, so we always have $Precond(Paint(x,c),s)$.

5. We need more axioms to ensure that fluents that an action doesn't explicitly change retain their values. In $\mathcal{T}^{bl1}$ it is convenient to write these *frame axioms* as

$$\begin{gathered} Color(x, Result(Move(y,l),s)) = Color(x,s), \\ x \neq y \rightarrow Color(x, Result(Paint(y,c),s)) = Color(x,s), \\ At(x,l,Result(Paint(y,c),s)) \equiv At(x,l,s), \\ \text{and} \\ x \neq y \rightarrow At(x,l',Result(Move(y,l),s)) \equiv At(x,l',s). \end{gathered} \quad (8.22)$$

Because we have only two fluents $At(x,l,s)$ and $Color(x,s)$ and two actions $Move(x,l)$ and $Paint(x,c)$, (8.22) consists of just four sentences. With m fluents and n actions, this method requires mn sentences. There is a further disadvantage. It is convenient to state the facts about moving blocks once and for all. It is inconvenient to be required to state additional facts about the non-effect of moving blocks on the colors of blocks, their smell or on whether it is raining.

This is called the frame problem and its the subject of the next section.

However, we have done enough to prove the desired facts (8.9) about S'.

8.3 The frame problem

The frame problem has two aspects. The first is to be able to compactly describe the effects of some events on some fluents without having to include for each action-fluent pair an explicit sentence when the action does not change the value of the fluent. Many systems satisfy this goal. The second objective is to maintain *elaboration tolerance* of the addition of new fluents and actions to the formalism. Systems that meet the first goal may not meet the second.

8.4 Beyond sequential situation calculus

8.5 Elaborating sequential sitcalc to handle concurrency

Consider two sequential sitcalc theories. Call them D and J for Daddy and Junior, respectively. Let each of them be formalized with sequential sitcalc sentences of the form

$$s' = \text{Result}(e, s). \tag{8.23}$$

We have

$$\begin{array}{lll} D & : & At(Daddy, NY, S0) \\ J & : & At(Junior, Glasgow, S0). \end{array} \tag{8.24}$$

The fact that the term $S0$ occurs in both formulas does not mean that it designates the same situation. The context prefixes D and J keep them separated. We can reason separately about Daddy and Junior in the two contexts.

Let's now introduce an outer context c. We have

$$\begin{array}{lll} c:D & : & At(Daddy, NY, S0) \\ c:J & : & At(Junior, Glasgow, S0). \end{array} \tag{8.25}$$

We can now express the distinctness of the two occurrences of $S0$ by writing

$$c : \quad value(D, S0) \neq value(J, S0). \tag{8.26}$$

We can write the whole of (8.25) in the outer context as

$$\begin{array}{lll} c & : & Value(D, At)(Value(D, Daddy), Value(D, NY), Value(D, S0)), \\ c & : & Value(J, At)(Value(J, Junior), Value(J, Glasgow), Value(J, S0)). \end{array} \tag{8.27}$$

[5] However, we are more likely to want to keep the value of all these entities except for $S0$ the same and write

$$\begin{aligned} c &: \quad At(Daddy, NY, Value(D, S0)) \\ c &: \quad At(Junior, Glasgow, Value(J, S0)). \end{aligned} \tag{8.28}$$

Now suppose we want to combine the Daddy and Junior stories with a view to introducing an interaction between Daddy and Junior as discussed in (McCarthy 1995b) and (McCarthy and Costello 1998). We introduce a new context DJ (for Daddy-Junior). We need to translate sentences and terms from the D and J contexts into DJ.

Consider the sentence

$$J : \quad Has\text{-}ticket(x, y, s) \wedge At(x, s) \rightarrow At(y, \text{Result}(Fly(x, y), s)). \tag{8.29}$$

It requires several changes before it can meaningfully inhabit DJ, and these seem to involve giving up some of the advantages sequential sitcalc, i.e. of the Result(e, s) formalism. Here are some of those advantages.

1. Applications of Result can be composed, e.g. to get

$$\text{Result}(a3, \text{Result}(a2, \text{Result}(a1, s))),$$

representing the effect of three successive actions.

2. In sequential sitcalc, there is only one actor, so the actor can remain implicit. We will have to replace $Fly(x, y)$ by $Does(Junior, Fly(x, y))$ when we *lift* to a context with more than one actor.

Before combining two contexts, e.g. D and J, we need to go from the Result to one involving $occurs(e, s)$ and $s < s'$. We also need $time(s)$ and the axioms

$$time(\text{Result}(e, s)) > time(s) \tag{8.30}$$

and

$$s < s' \rightarrow time(s') > time(s). \tag{8.31}$$

[5] Making *at* context dependent would require using a higher order logic. We can contemplate this with equanimity, especially as we don't have immediate plans to make *at* context dependent.

Here's the main axiom for going from Result to $occurs$.

$$\begin{array}{l}\pi(s, \text{Result}(e, s)) \wedge occurs(e, s) \wedge \neg ab1(e, s) \\ \rightarrow (\exists s')(s < s' \wedge \pi(s, s').\end{array} \tag{8.32}$$

I expect that the formalism with $occurs$ and $<$ is more readily liftable than the Result formalism. I'm assuming, as with the previous axioms, that all the variables are universally quantified, but I'm not sure that this is the correct formula.

I used $\pi(s, s')$ in order to accomodate (8.30) and (8.31).

It looks like we should do the unabbreviations in D and J separately. Thus we have

$$\begin{array}{ll}J: & has(x, s) \equiv has(Junior, x, s) \\ J: & at(x, s) \equiv at(Junior, x, s) \\ J: & fly(x, y) = does(Junior, fly(x, y)) \\ D: & has(x, s) \equiv has(Daddy, x, s) \\ D: & send(x, y) = does(Daddy, send(x, y))\end{array} \tag{8.33}$$

Here we have escaped contradiction by the coincidence that the different occurrences of has, etc. take different numbers of arguments. In general, we will have to change some names.

Consider combining the Daddy and Junior narratives or combining the planning problems. Let's take a stripped down version of story in (McCarthy 1996b) and (McCarthy and Costello 1998). We'll say that Junior can buy a ticket but mention the need for money and not provide him with any. We can suppose that Daddy has money and can send it to Junior. Maybe we can put that action in the context D, i.e. with the axiom

$$has(Junior, ticket\text{-}money, \text{Result}(send(ticket\text{-}money, Junior, s))). \tag{8.34}$$

Junior is now a character in the Daddy narrative, i.e. is in the ontology of the context D, and we will have to identify him with the Junior of J when we combine the narratives. There are a few other problems.

1. We have to get the right time relations. There is no *a priori* relation between the times of D and J.

2. I don't want to say that in the combined context DJ

$$DJ: \qquad S0 = \text{Result}(send(ticket\text{-}noney, Junior, s)).$$

We need to put the time of arrival of the money before the time when Junior buys the ticket. Therefore, we need an additional persistence axiom to say that if Junior has the money as a result of Daddy sending it, he will still have it later unless something happens to get rid of it.

We'll first consider what sentences we want to appear in DJ and then consider how to get them there from D, J, from reasonable axioms about combining the contexts and from reasonable axioms about the combined context itself.

"Mann muss immer umkehren" - Jacobi to his students. Planning is the inverse of projection. There are two ways in which the nonmonotonic inertia laws can be seen to fail. (1) The block is too heavy, so the action of moving it fails. The second is that Block1 is observed to be not on top of Block2, so something must have happened. The first is the straightforward intent of making the law of inertia nonmonotonic. The second has offered difficulties.

Chronological minimization. We express inertia by minimizing ab at each situation. The variable predicate has two arguments f and e. The important clause is

$$(\forall ab' vars)\neg(ab'(f,a) < ab(f,a,s)) \tag{8.35}$$

8.6 Relations among event formalisms

Chapter 9

FORMALIZING CONTEXTS AS OBJECTS

> Let my name be Ishmael.
> Let the ship's name be Pequod.
> Let the captain's name be Ahab.
> Let the whale's name be as given in the title.
> - from *If a mathematician had written Moby Dick*

A mouse click in a context lifts to an assertion or command in a larger language.

Chapter 10

ELABORATION TOLERANCE

Abstract of this chapter

A formalism is *elaboration tolerant* to the extent that it is convenient to modify a set of facts expressed in the formalism to take into account new phenomena or changed circumstances. Representations of information in natural language have good elaboration tolerance when used with human background knowledge. Human-level AI will require representations with much more elaboration tolerance than those used by present AI programs, because human-level AI needs to be able to take new phenomena into account.

The simplest kind of elaboration is the addition of new formulas. We'll call these *additive elaborations*. Next comes changing the values of parameters. Adding new arguments to functions and predicates represents more of a change. However, elaborations not expressible as additions to the object language representation may be treatable as additions at a meta-level expression of the facts.

Elaboration tolerance requires nonmonotonic reasoning. The elaborations that are tolerated depend on what aspects of the phenomenon are treated nonmonotonically. Representing contexts as objects in a logical formalism that can express relations among contexts should also help.

We use the missionaries and cannibals problem and about 20 variants as our *Drosophila* in studying elaboration tolerance in logical AI.

The present version has only some parts of a situation calculus formalization. However, the English language elaborations listed are enough to serve

as a challenge to logical AI formalisms claiming elaboration tolerance.

10.1 Introduction

In several papers, e.g. (McCarthy 1988) and (McCarthy 1989), I discussed the *common sense informatic situation* and contrasted it with the information situation within a formal scientific theory. In the latter, it is already decided what phenomena to take into account. In the former, any information possessed by the agent is available and potentially relevant to achieving its goals. *Elaboration tolerance* seems to be the key property of any formalism that can represent information in the common sense informatic situation. [1]

Elaboration tolerance[2] is the ability to accept changes to a person's or a computer program's representation of facts about a subject without having to start all over. Often the addition of a few sentences describing the change suffices for humans and should suffice for computer programs.

Humans have considerable elaboration tolerance, and computers need it to reach human-level AI. In this article we study elaboration tolerance in terms of logical AI. However, researchers pursuing other AI methodologies will also have to face the problem of elaboration tolerance; maybe they just haven't noticed it yet. The relation to *belief revision* will be discussed briefly in section 12.7.

Humans represent information about the world in natural language and use background knowledge not ordinarily expressed in natural language and which is quite difficult to express. [3] The combination of linguistic and non-linguistic knowledge is what gives us humans our elaboration tolerance. Unfortunately, psychological research hasn't yet led to enough understanding of the background knowledge, so it is hard to study elaboration tolerance in humans. However, it is easy to give plenty of examples of human elaboration tolerance, e.g. those in this article.

[1] 2004 Jan: I now contrast the common sense informatic situation with a *bounded informatic situation*.

[2] The concept was first mentioned in (McCarthy 1988).

[3] The non-linguistic background knowledge has been emphasized in connection with physical skills by Hubert Dreyfus and others (Dreyfus 1992), but there is important non-linguistic knowledge also when the skill is purely symbolic. Even though a mathematician or a stock broker operates in a purely symbolic domain, he still cannot verbalize his full set of skills.

The *Drosophila*[4] of this article is the missionaries and cannibals problem (MCP).

After describing the original MCP, we give a large number of elaborations. Humans tolerate these elaborations in the sense that we can use the sentences expressing one of the elaborations to get a modified problem. People will agree on what the modified problem is and will agree on whether a proposed solution is ok.

Then we consider logical formalizations of the original problem and discuss which elaborations different formalisms tolerate. Our goal—not achieved in this article—is a formalism for describing problems logically that is as elaboration tolerant as English and the associated background knowledge. However, some logical languages are more elaboration tolerant than others.[5]

10.2 The Original Missionaries and Cannibals Problem

The missionaries and cannibals problem (abbreviated MCP):

> **Three missionaries and three cannibals come to a river and find a boat that holds two. If the cannibals ever outnumber the missionaries on either bank, the missionaries will be eaten.**
>
> **How shall they cross?**

We call this original version of the problem MCP0.

Saul Amarel proposed (Amarel 1971): Let a state (*mcb*) be given by the numbers of missionaries, cannibals and boats on the initial bank of the river. The initial situation is represented by 331 and the goal situation by 000.

[4] *Drosophilas* are the fruit flies that have been used by geneticists to study inheritance since 1910. Their short generation times, large chromosomes and the ability to keep 1,000 of them in a bottle make them valuable, even though the *Drosophilas* of today are no better than those of 1910. The utility of suitable *Drosophilas* for scientific research in AI needs to be emphasized, because of a recent fad for demanding that all research promise a practical payoff on a three year schedule. They aren't getting their payoffs and are learning much less than a more scientific approach would get them.

[5] 2004 Jan 26: The introduction of concepts and propositions as objects as in (McCarthy 1979c) can be handled as an elaboration. I think this corresponds to what people do. More on this in a new section.

Most AI texts that mention the problem accept this formulation and give us the solution:

$$331 \rightarrow 310 \rightarrow 321 \rightarrow 300 \rightarrow 311 \rightarrow 110 \rightarrow 221 \rightarrow 020 \rightarrow 031 \rightarrow 010 \rightarrow 021 \rightarrow 000.$$

The state space of the Amarel repreentation has 32 elements some of which are forbidden and two of which are unreachable. It is an elementary student exercise to write a program to search the space and get the above sequence of states, and people are always solving it without a computer or without even a pencil. Saul Amarel (Amarel 1971) points out that this representation has fewer states than a representation with named missionaries and cannibals.

What more does this problem offer AI?

If one indeed begins with the Amarel representation, the problem is indeed trivial. However, suppose we want a program that begins, as people do, with a natural language presentation of the problem. It is still trivial if the program need only solve the missionaries and cannibals problem. The programmer can then cheat as much as he likes by making his program exactly suited to the MCP. The extreme of cheating is to make a program that merely prints

$$331 \rightarrow 310 \rightarrow 321 \rightarrow 300 \rightarrow 311 \rightarrow 110 \rightarrow 221 \rightarrow 020 \rightarrow 031 \rightarrow 010 \rightarrow 021 \rightarrow 000.$$

Readers will rightly complain that this cheats, but it isn't clear what does and doesn't count as cheating when a method for solving a single problem is asked for.

The way to disallow cheating is to demand a program that can solve any problem in a suitable set of problems. To illustrate this we consider a large set of elaborations of MCP. It won't be trivial to make a program that can solve all of them unless the human sets up each of them as a state space search analogous to the original MCP. We demand that the program use background common sense knowledge like that about rivers and boats that is used by a human solver.

We skip the part about going from an English statement of the problem to a logical statement for two reasons. First, we don't have anything new to say about parsing English or about the semantics of English. Second, we don't yet have the logical target language that the parsing program should aim at. Progress toward establishing this language is the goal of the paper.

The problem is then to make a program that will solve any of the problems using logically expressed background knowledge. The background knowledge should be described in a general way, not specifically oriented to MCP and related problems.

This much was already proposed in (McCarthy 1959). What is new in the present paper is spelling out the idea of *elaboration tolerance* that was distantly implicit in the 1959 paper. We require a formulation of MCP that readily tolerates elaborations of the problem and allows them to be described by sentences added to the statement of the problem rather than by surgery on the problem. We can call these *additive elaborations.* English language formulations allow this, but the Amarel-type formulations do not. AI requires a logical language that allows elaboration tolerant formulations.

We begin a few examples of English language elaboration tolerance. After discussing situation calculus formalisms, there will be a lot more.

- The boat is a rowboat. (Or the boat is a motorboat). This elaboration by itself should not affect the reasoning. By default, a tool is usable. Later elaborations make use of specific properties of rowboats.

- There are four missionaries and four cannibals. The problem is now unsolvable.

- There is an oar on each bank. One person can cross in the boat with just one oar, but two oars are needed if the boat is to carry two people.

- One of the missionaries is Jesus Christ. Four can cross. Here we are using cultural literacy. However, a human will not have had to have read Mark 6: 48–49 to have heard of Jesus walking on water.

- Three missionaries with a lone cannibal can convert him into a missionary.

A later section discusses the formal problems of these and other elaborations.

10.3 Nonmonotonic reasoning

Elaboration tolerance clearly requires nonmonotonic reasoning. For example, elaborating MCP0 with a requirement for oars adds preconditions to the

action of going somewhere in the boat. If oars are not mentioned, nonmonotonic reasoning prevents such additional preconditions.

However, it is still not clear how to formulate the nonmonotonic reasoning so as to obtain tolerance of a wide class of elaborations, such as those of section 10.7. We propose to use some variant of circumscription, but this still leaves open what is to be circumscribed.

(McCarthy 1980) discusses several nonmonotonic aspects of the human understanding of MCP0. They all have a Gricean (Grice 1989) character. They all concern the non-existence of features of the problem that should have been mentioned, were they supposed to exist. What can be inferred from such contexts includes the *Gricean implicatures*. Very likely, the formal theory of contexts (McCarthy 1993) can be used, but that is beyond the scope of this article.

Here are a few nonmonotonic inferences that come up. Each of them seems to present its own formal problems.

- If there were a bridge, it should have been mentioned. A puzzle problem like MCP is given in a context.
- The river can't be forded, and there isn't an extra boat.
- There isn't a requirement for a permit or visa to cross the river.
- There is nothing wrong with the boat. In general, when a tool is mentioned, it is supposed to be usable in the normal way.
- The group of missionaries and cannibals is minimized. Mentioning that one of the missionaries is Jesus Christ will include him in the number otherwise inferrable rather than adding him as an additional missionary. Of course, assertions about him in the general database should have no effect unless he is mentioned in the problem.
- If you keep transporting cannibals to the island (in the variant with an island) they will eventually all be at the island.

These kinds of nonmonotonic reasoning were anticipated in (McCarthy 1980) and have been accomodated in situation calculus based nonmonotonic formalisms, although the Yale shooting problem and others have forced some of the axiomatizations into unintuitive forms.

The elaborations discussed in this article mainly require the same kinds of nonmonotonic reasoning.

10.4 A Typology of Elaborations

There are many kinds of elaborations a person can tolerate, and they pose different problems to different logical formalizations. Here are some kinds of elaborations.

irrelevant actors, actions and objects Sentences establishing the existence of such entities should not vitiate the reasoning leading to a solution.

adding preconditions, actions, effects of actions and objects The example of the oars adds a precondition to rowing and adds the action of picking up the oars. Several situation calculus and event calculus formalisms allow this—assuming sentences are added before the nonmonotonic reasoning is done. Tolerating added preconditions is a criterion for good solutions of the qualification problem, and tolerating adding effects relates similarly to the ramification problem.

changing a parameter This is needed when the numbers of missionaries and cannibals are changed from 3 to 4. In English, this is accomplished by an added sentence. Doing it that way in logic requires a suitable *belief revision* method as part of the basic logical formalism. At present we must use minor brain surgery to replace certain occurrences of the number 3.

making a property situation dependent Whether x is a missionary is not situation dependent in MCP0, but we can elaborate to a missionary becoming a cannibal. It is tempting to say that all properties should be situation dependent from the beginning, and such a formalism would admit this elaboration easily. I think this might lead to an infinite regress, but I can't formulate the problem yet.

specialization In one situation calculus formalization we have the action $Move(b1, b2)$. If there are guaranteed to be exactly two places, we can replace this action by $Move(b)$, regarding this as $Move(b, Opp(b))$, where $Opp(b)$ designates the opposite bank and satisfies $Opp(Opp(b)) = b$. We regard this kind of specialization as an easy kind of elaboration.

generalization Some of our elaborations can be composed of an generalization of the language—replacing a function by a function of more

arguments, e.g. making whether a person is a cannibal or missionary situation dependent or replacing going from a bank b to the opposite bank $Opp(b)$ by going from $b1$ to $b2$. Many elaborations consist of a generalization followed by the addition of sentences, e.g. adding preconditions or effects to an action.

unabbreviation This is a particular case of generalization. Suppose we write $(\forall a \in Actions)Abbreviates[a, Does(person, a)]$. We mean to use it in elaborating $Result(a, s)$ to $Result(Does(person, a), s)$ in sentences where it is somehow clear which person is referred to. The square brackets mean that this is a metalinguistic statement, but I don't presently understand precisely how unabbreviation is to work.

going into detail An event like the action of crossing the river is made up of subactions. However, the relation between an event and its subevents is not often like the relation between a program and its subroutines, because asserting that the action occurs does not imply specific subactions. Rowing is a detail of crossing a river when rowboats are used, but rowing is not a part of the general notion of crossing a river.. Bailing if necessary is another detail. Getting oars or a bailing can are associated details. There is more about this apparently controversial point in (McCarthy 1995b).

m...s and c...s as agents MCP and almost all of the elaborations we have considered take a god-like view of the actions, e.g. we send a cannibal to get an oar. We can also elaborate in the direction of supposing that the actions of cannibals and missionaries are sometimes determined by the situation. In this case, it may be convenient to use a predicate $Occurs(event, s)$ and let one possible event be $Does(C_1, Enter\text{-}Boat)$. The situation calculus treatment has to be altered and looks more like event calculus.

simple parallel actions If one of the missionaries is Jesus Christ, we can transport 4 missionaries and 4 cannibals. We get 3 cannibals on the far bank and one on the initial bank. Then two ordinary missionaries and Jesus cross, the ordinaries in the boat and Jesus walking on water. The rest of the solution is *essentially the same* as in MCP0. A missionary and a cannibal row back and now the remaining two missionaries cross. We then send a cannibal to ferry the remaining two cannibals. We

haven't tackled the problem of being able to say "essentially the same" in logic.

The formalization must permit Jesus to cross in parallel with the other missionaries so that the missionaries are never outnumbered. This isn't the same as having Jesus cross as a separate action.

full parallelism This is what permits requiring that the boat be bailed.

events other than actions The simple $Result(a, s)$ doesn't allow for events other than actions. To handle them we have used (McCarthy 1995b) a predicate $Occurs(e, s)$ asserting that the event e occurs in the situation s. Then $Result(e, s)$ can be used.

comparing different situations This works ok in situation calculus, but some other formalisms don't allow it or make it awkward. Thus we can have $s <_{better} Result(e, s)$ to say that the situation is better after event e occurs. We may also want $Result(a1, s) <_{better} Result(a2, s)$, comparing the result of doing $a1$ with the result of doing $a2$.

splitting an entity Sometimes an entity, e.g. a node in a graphy, an edge, or a concept needs to be split into two entities of the same type so that separate properties can be assigned to each subentity. Thus we may split cannibals into strong and weak cannibals.

continuous time and discrete time If Achilles runs enough faster than the tortoise, there is a time when Achilles catches up. We use the fluent $Future(\pi, s)$ to assert that the situation s will be followed in the future by a situation satisfying π. We do not require formalizing real numbers to express the Achilles catching up sentence

$$Future((\lambda s)(Value(Distance\text{-}covered\text{-}by(Achilles), s) = Value(Distance\text{-}covered\text{-}by(Tortoise), s)), S0).$$

Concepts and propositions as objects The ideas of (McCarthy 1979c) are directly usable. Let's add some sentences which are a conservative extension of the original theory.

$$\begin{array}{l} Missionaries = \{Tom, Dick, Harry\} \\ Distinct(Tom, Dick, Harry) \\ Jesus \in Missionaries \\ \text{hence} Jesus = Tom \vee Jesus = Dick \vee Jesus = Harry. \end{array} \tag{10.1}$$

Now suppose we want to say that Sam knows that Jesus is Tom.

We introduce

$$\begin{aligned} &denot(JJesus) = Jesus \\ &denot(TTom) = Tom \end{aligned} \tag{10.2}$$

Here we have had to change the notation of (McCarthy 1979c), because that paper uses capitalization to distinguish between concepts and ordinary objects, whereas this paper uses capitalization for constants. What we do is double the first letter to denote concepts. Equations (10.1) and (10.2) are just a conservative extension of the original theory, because they serve just to define new terms.

Now we write

$$Know(Sam, EEqual(JJesus, TTom). \tag{10.3}$$

If we use the theory of (McCarthy 1979c), this is a non-conservative extension, because it permits inferring

$$Jesus = Tom. \tag{10.4}$$

The main point is that the whole apparatus of concepts and knowledge is just an elaboration of a theory without it.

10.5 Formalizing the Amarel Representation

We use logic and set theory to formalize MCP0 and call the formalization MCP0a. In this formalization we are not concerned with elaboration tolerance. My opinion is that set theory needs to be used freely in logical AI in order to get enough expressiveness. The designers of problem-solving programs will just have to face up to the difficulties this gives for them.

$$States = Z4 \times Z4 \times Z2$$

$$\begin{aligned} &(\forall state)(Ok(state) \equiv \\ &\quad Ok1(P1(state), P2(state)) \wedge Ok1(3 - P1(state), 3 - P2(state))) \end{aligned}$$

Here $Z2 = \{0, 1\}$ and $Z4 = \{0, 1, 2, 3\}$ are standard set-theory names for the first two and the first four natural numbers respectively, and $P1$, $P2$ and $P3$ are the projections on the components of the cartesian product $Z4 \times Z4 \times Z2$.

Note that having used $3 - P1(state)$ for the number of missionaries on the other bank put information into posing the problem that is really part of solving it, i.e. it uses a law of conservation of missionaries.

$$(\forall m\ c)(Ok1(m, c) \equiv m \in Z4 \wedge c \in Z4 \wedge (m = 0 \vee m \geq c))$$

$$Moves = \{(1, 0), (2, 0), (0, 1), (0, 2), (1, 1)\}$$

$$\begin{array}{ll} & (\forall move\ state) \\ (Result(move, state) = Mkstate(& P1(state) - (2P3(state) - 1)P1(move), \\ & P2(state) - (2P3(state) - 1)P2(move), \\ & 1 - P3(state))), \end{array}$$

where $Mkstate(m, c, b)$ is the element of $States$ with the three given components.

$$(\forall s1\ s2)(Step(s1, s2) \equiv (\exists move)(s2 = Result(move, s1) \wedge Ok(s2)))$$

$$Attainable1 = Transitive\text{-}closure(Step)$$

$$Attainable(s) \equiv s = (3, 3, 1) \vee Attainable1((3, 3, 1), s)$$

Notice that all the above sentences are definitions, so there is no question of the existence of the required sets, functions, constants and relations. The existence of the transitive closure of a relation defined on a set is a theorem of set theory. No fact about the real world is assumed, i.e. nothing about rivers, people, boats or even about actions.

From these we can prove

$$attainable((0, 0, 0)).$$

The applicability of MCP0a to MCP must be done by postulating a correspondence between states of the missionaries and cannibals problem and states of MCP0a and then showing that actions in MCP have suitable corresponding actions in MCP0a. We will postpone this until we have a suitable elaboration tolerant formalization of MCP.

MCP0a has very little elaboration tolerance, in a large measure because of fixing the state space in the axioms. Situation calculus will be more elaboration tolerant, because the notation doesn't fix the set of all situations.

10.6 Situation Calculus Representations

The term *situation calculus* is used for a variety of formalisms treating situations as objects, considering *fluents* that take values in situations, and events (including actions) that generate new situations from old.

At present I do not know how to write a situation calculus formalization that tolerates all (or even most) of the elaborations of section 10.7. Nevertheless, I think it is useful to give some formulas that accomplish some elaborations and discuss some issues of elaboration tolerance that these formulas present.

10.6.1 Simple situation calculus

We begin with some axioms in a formalism like that of (McCarthy 1986) using a *Result* function, a single abnormality predicate and aspects. This suffers from the Yale shooting problem if we simply minimize *Ab*. However, as long as the problem requires only projection, i.e. predicting the future from the present without allowing premises about the future, *chronological minimization* of *Ab* (Shoham 1988) avoids the Yale shooting problem. It is certainly a limitation on elaboration tolerance to not allow premises about the future.

Here are some axioms and associated issues of elaboration tolerance.

The basic operation of moving some people from one bank to the other is conveniently described without distinguishing between missionaries and cannibals.

$$\begin{aligned}
&\neg Ab(Aspect1(group, b1, b2, s)) \rightarrow \\
&\qquad Value(Inhabitants(b1), Result(Cross(group, b1, b2), s)) = \\
&\qquad\qquad Value(Inhabitants(b1), s) \setminus group \\
&\wedge \\
&\qquad Value(Inhabitants(b2), Result(Cross(group, b1, b2), s)) = \\
&\qquad\qquad Value(Inhabitants(b2), s) \cup group,
\end{aligned} \tag{10.5}$$

where $\setminus$ denotes the difference of sets.

The fact that (10.5) can't be used to infer the result of moving a group if some member of the group is not at $b1$ is expressed by

$$\neg(group \subset Value(Inhabitants(b1), s)) \rightarrow Ab(Aspect1(group, b1, b2, s)).$$

We extend the notion of an individual being at a bank to that of a group being at a bank.

$$Holds(At(group, b), s) \equiv (\forall x \in group) Holds(At(x, b), s).$$

$$\begin{aligned} &\neg Ab(Aspect2(group, b1, b2, s)) \wedge Crossable(group, b1, b2, s) \\ &\quad \rightarrow \neg Ab(Aspect1(group, b1, b2, s)) \end{aligned} \tag{10.6}$$

relates two abnormalities.

$$Crossable(group, b1, b2, s) \rightarrow 0 < Card(group) < 3 \tag{10.7}$$

tells us that the boat can't cross alone and can't hold more than two.

$Card(u)$ denotes the cardinality of the set u.

We can sneak in Jesus by replaceing (10.7) by

$$Crossable(group, b1, b2, s) \rightarrow 0 < Card(group \setminus \{Jesus\}) < 3, \tag{10.8}$$

but this is not in the spirit of elaboration tolerance, because it isn't an added sentence but is accomplished by a precise modification of an existing sentence (10.7) and depends on knowing the form of (10.7). It's education by brain surgery.

It's bad if the cannibals outnumber the missionaries.

$$\begin{aligned} &Holds(Bad(bank), s) \\ &\equiv \\ &0 < Card(\{x | x \in Missionaries \wedge Holds(At(x, bank), s)\}) \\ &< Card(\{x | x \in Cannibals \wedge Holds(At(x, bank), s)\}) \end{aligned} \tag{10.9}$$

and

$$Holds(Bad, s) \equiv (\exists bank) Holds(Bad(bank), s). \tag{10.10}$$

Many unique names axioms will be required. We won't list them in this version.

10.6.2 Not so simple situation calculus

The notion of *Bad* in the previous subsection avoids any actual notion of the missionaries being eaten. More generally, it avoids any notion that in certain situations, certain events other than actions will occur. We can put part of this back.

We would like to handle the requirement for oars and the ability of Jesus Christ to walk on water in a uniform way, so that we could have either, both or neither of these elaborations.

To say that the missionaries will be eaten if the cannibals outnumber them can be done with the formalism of (McCarthy 1995b).

$$\begin{aligned} &Holds(Bad(bank), s) \rightarrow \\ &\quad (\forall x)(x \in Missionaries \\ &\qquad \wedge Holds(At(x, bank), s) \rightarrow Occurs(Eaten(x), s)). \end{aligned} \tag{10.11}$$

As sketched in (McCarthy 1995b), the consequences of the occurrence of an event may be described by a predicate $Future(f, s)$, asserting that in some situation in the future of the situation s, the fluent f will hold. We can write this

$$Future(f, s) \rightarrow (\exists s')(s <_{time} s' \wedge Holds(f, s')), \tag{10.12}$$

and treat the specific case by

$$occurs(Eaten(x), s) \rightarrow F(Dead\text{-}soon(x), s) \tag{10.13}$$

To say that something will be true in the future of a situation is more general than using *Result*, because there is no commitment to a specific next situation as the result of the event. Indeed an event can have consequences at many different times in the future. The $Result(e, s)$ formalism is very convenient when applicable, and is compatible with the formalism of $Occurs$ and F. We have

$$\neg Ab(Aspect2(e, s)) \wedge Occurs(e, s) \rightarrow Future((\lambda s')(s' = Result(e, s)), s), \tag{10.14}$$

where something has to be done to replace the lambda-expression $(\lambda s')(s' = Result(e, s))$ by a syntactically proper fluent expression. One way of doing that is to regard $Equal(Result(e, s))$ as a fluent and write

$$\neg Ab(Aspect2(e, s)) \wedge Occurs(e, s) \rightarrow Future(Equal(Result(e, s))). \tag{10.15}$$

We may get yet more mileage from the *Result* formalism. Suppose $Result(e, s)$ is taken to be a situation after all the events consequential to e have taken place. We then have one or more consequences of the form $Past(f, Result(e, s))$, and these permit us to refer to the consequences of e that are distributed in time. The advantage is that we can the use $Result(e, s)$ as a base situation for further events.

10.6.3 Actions by Persons and Joint Actions of Groups

When there is more than one actor acting, we can consider three levels of complexity. The simplest level is when the actors act jointly to achive the goal. The second level is when one actor (or more than one) does something to motivate the others, e.g. one person pays another to do something. This generalizes to a hierarchy of influence. The hard level is when the actors have competing motivations and must negotiate or fight. This is the subject of game theory, and we won't pursue it in this article.

As MCP was originally formulated, the missionaries and cannibals are moved like pieces on a chessboard. Let's consider elaborations in which the actions of individual missionaries and cannibals are considered. One eventual goal might be to allow a formalization in which a cannibal has to be persuaded to row another cannibal across the river and bring the boat back. However, our discussion starts with simpler phenomena.

We now consider an action by a person as a particular kind of event. What we have written $Result(a, s)$ we now write $Result(Does(person, a), s)$. If there is only one person, nothing is gained by the expansion.

Consider a proposition $Can\text{-}Achieve(person, goal, s)$, meaning that the person *person* can achieve the goal *goal* starting from the situation s. For the time being we shall not say what goals are, because our present considerations are independent of that decision. The simplest case is that there is a sequence of actions $\{a_1, \ldots, a_n\}$ such that

$$Result(Does(person, a_n), Result(\ldots Result(Does(person, a_1), s)\ldots))$$

satisfies *goal*.

Now let's consider achievement by a group. We will say $Can\text{-}Achieve(group, goal, s)$ provided there is a sequence of events $\{Does(person_1, a_1), \ldots, Does(person_n, a_n)\}$, where each $person_i$ is in *group*, and the $person_i$s are not assumed to be distinct, and such that

$$Result(Does(person_n, a_n), Result(\ldots Result(Does(person_1, a_1), s)\ldots))$$

satisfies *goal*.

We can now introduce a simple notion of a person leading a group, written $leads(person, group)$ or more generally $leads(person, group, s)$. We want the axioms

$$leads(person, group) \wedge Can\text{-}Achieve(group, goal, s) \rightarrow Can\text{-}Achieve(person, s)$$

Thus a leader of a group can achieve whatever the group can achieve. Note that *person* need not be a member of *group* for this definition to work.

We could give the same definition for $leads(person, group, s)$, but maybe it would be better to make a definition that requires that *person* maintain his leadership of *group* in the succeeding situations.

$Leads(person, group)$ is too strong a statement in general, because the members of a group only accept leadership in some activities.

10.7 Formalizing some elaborations

1. The boat is a rowboat. (Or the boat is a motorboat). By itself this is a trivial elaboration. Adding it should not affect the reasoning. By default, a tool, i.e. the boat, is usable. Further elaborations might use specific properties of rowboats.

2. The missionaries and cannibals have hats, all different—another trivial elaboration. These hats may be exchanged among the missionaries and cannibals. In all the elaborations mentioned below, exchanging hats is an action irrelevant to crossing the river. There are two demands on the reasoner. Epistemologically, whatever reasoning that establishes a plan for crossing the river without the hats should be valid with the hats. This includes any nonmonotonic reasoning.

   Heuristically, the problem may not be trivial. Why should it be obvious that exchanging hats is of no use? Certainly we can make elaborations in which it is of use, e.g. we can assert that if the smallest missionary wears the hat belonging to the largest missionary, the largest cannibal won't eat him even if they go together.

   However, it should be possible to tell a problem solver: Look for a solution that has no hat change actions. After that, the reasoner should find the solution as easily as it would if hats were never mentioned.

3. There are four missionaries and four cannibals. The problem is now unsolvable. In ordinary logic, adding sentences that there are four of each produces a contradiction. Belief revision systems ought to make the correct change. It seems to me that people take a metalinguistic stance, just saying "Change the numbers of missionaries and cannibals to four", thus regarding the original statement of the problem as an object. Actually what is regarded as an object is the *sense* of the original statement, since people ordinarily don't remember the words used.

   Proofs of impossibility take the following form. Choose a predicate formula $\phi(s)$ on situations. Show $\phi(S0)$ and $(\forall s)(\phi(s) \rightarrow \neg Goal(s))$. Also show

   $$(\forall s\ a)(\phi(s) \rightarrow Bad(Result(a, s)) \vee \phi(Result(a, s))).$$

   Thus you can't get out of the situations satisfying ϕ, and the goal isn't included. The simplest $\phi(s)$ is a disjunction of specific locations of the missionaries and cannibals in the reachable situations, but this disjunction is long, and it is very likely possible to do better.

   We can regard the argument that four can't cross as a kind of elaboration. A formalism that doesn't permit expressing the best argument is then deficient in elaboration tolerance.

4. The boat can carry three. Four can cross but not five. If the boat can carry four an arbitrary number can cross. [2003 Sept: This is mistaken. Joohyung Lee showed that if the boat holds three, five can cross.]

5. There is an oar on each bank. One person can cross in the boat with just one oar, but two oars are needed if the boat is to carry two people. We can send a cannibal to get the oar and then we are reduced to the original problem. [6]

   A formalism using preconditions can accept this elaboration as just adding a precondition for rowing, the action of putting an oar in the boat and adding facts about the locations of the oars in $S0$.

[6] It was not mentioned before that the boat was a rowboat. Once oars are mentioned, it is a Gricean implicature that the boat is a rowboat. The philosopher Paul Grice (Grice 1989) studied what can be inferred from statements under the assumption that the person posing the problem is not trying to be misleading. That the boat is a rowboat follows, because the speaker should have said so if it wasn't.

The oar-on-each-bank elaboration can be expressed by conjoining to (10.16),

$$Card(group) > Card(\{x|Oar(x) \wedge Holds(In(x, Boat), s)\}) \\ \rightarrow Ab(Aspect1(group, b1, b2, s)),$$

but this looks a bit *ad hoc*. In particular, it wouldn't tolerate the further elaboration of making the boat hold three if that elaboration were expressed as the single sentence

$$Crossable(group, b1, b2, s) \rightarrow 0 < Card(group) < 4$$

In order to admit the reasoning that getting the oar reduces the problem to MCP0, we will need a notion of one problem reducing to another—or one theory reducing to another.

6. Only one missionary and one cannibal can row. The problem is still solvable. Before this elaboration, we did not need to distinguish among the missionaries or among the cannibals. An elaboration tolerant language must permit this as an addition. We use

$$\neg(\exists x)(x \in group \wedge Rower(x)) \rightarrow Ab(Aspect1(group, b1, b2, s)). \quad (10.16)$$

and

$$(\exists! x \in Cannibals)Rower(x) \wedge (\exists! x \in Missionaries)Rower(x) \quad (10.17)$$

7. The missionaries can't row. This makes the problem impossible, since any solution requires two missionaries in the boat at some time. The formalism must admit the statement and proof of this lemma.

   For this we need (10.16) and $(\forall x \in Missionaries)\neg Rower(x)$.

8. The biggest cannibal cannot fit in the boat with another person. The problem is solvable. However, if the biggest missionary cannot fit in the boat with another person the problem becomes unsolvable. We can imagine having to elaborated in the direction of saying what sets

of people can fit in the boat. The elaborations are $BigC \in Cannibals$ and

$$Crossable(group) \wedge BigC \in group \rightarrow group = \{BigC\}. \quad (10.18)$$

Note that the defining property of the biggest cannibal is unnecessary to make the elaboration work. I assume we'd pay for this shortcut, were further elaboration necessary.

The corresponding elaboration about the biggest missionary is formalized in the same way; only the conclusion is different.

9. If the biggest cannibal is isolated with the smallest missionary, the latter will be eaten. A solution to the basic problem can be specialized to avoid this contingency. We have the Gricean implicature that the cannibals aren't all the same size, and need to have language for referring to an individual as the biggest cannibal and not just language to refer to him by name. We have

$$group = \{BigC, SmallM\} \rightarrow \neg Crossable(group, b1, b2, s), \quad (10.19)$$

and

$$Inhabitants(bank, s) = \{BigC, SmallM\} \rightarrow Holds(Bad(bank), s). \quad (10.20)$$

10. One of the missionaries is Jesus Christ. Four can cross. Here we are using cultural literacy. However, a human will not have had to have read Mark 6: 48–49 to have heard of Jesus walking on water. The formalism of Section 10.6 permits this elaboration just by adjoining the sentence

$$Crossable(group, b1, b2, s) \rightarrow Crossable(group \cup \{Jesus\}, b1, b2, s). \quad (10.21)$$

However, this elaboration says nothing about walking on water and therefore seems to be a cheat.

11. Three missionaries alone with a cannibal can convert him into a missionary. The problem for elaboration tolerance is to change a predicate that doesn't depend on situation or time to one that does. Note that a sorted logical language with missionaries and cannibals as distinct sorts would freeze the intolerance into the language itself.

12. The probability is 1/10 that a cannibal alone in a boat will steal it. We can ask what is the probability that a given plan will succeed, say the Amarel plan. The formalism of (McCarthy 1979a) treats *propositions* as objects. Using that formalism $Pr(p) = 1/10$ can be expressed for any proposition p. I see at least two problems. The language of propositions as objects needs to be rich enough to express notions like the probability of a cannibal stealing the boat on an occasion—or of being a thief who always steals boats if alone. The second problem is that we need to be able to assert independence or joint distributions without letting the entire formalism be taken over by its probabilistic aspects. In MCP0, cannibals have to be alone in the boat several times. We can write a formula that states that probabilities are independent by default.

    We now need to infer that the probability of successfully completing the task is 0.9.

13. There is a bridge. This makes it obvious to a person that any number can cross provided two people can cross at once. It should also be an *obvious* inductive argument in the sense of McAllester (McAllester). This is a straightforward elaboration in situation calculus formalisms, since adding the bridge is accomplished just by adding sentences. There is no need to get rid of the boat unless this is part of the elaboration wanted.

14. The boat leaks and must be bailed concurrently with rowing. Elaboration tolerance requires that treating a concurrent action be a small change in the statement of the problem, and this will show the limitations of some versions of situation calculus.

15. The boat may suffer damage and have to be taken back to the left bank for repair. This may happen at any time. This requires that the formalism permit splitting the event of crossing the river into two parts.

16. There is an island. Then any number can cross, but showing it requires inductive arguments. Though inductive, these arguments should be obvious. Defining the three stages—moving the cannibals to the island, moving the missionaries to the opposite bank and then moving the cannibals to the opposite bank—is an easy three step problem, provided moving the sets of missionaries and cannibals can be regarded as tasks. Whether the elaboration is easy depends on the original representation.

    There may be a nonmonotonic rule that if you keep getting closer to a goal and there is no inferrable obstacle you will achieve the goal. Zeno's "paradox" of Achilles and the tortoise involves noting that this rule doesn't always hold, i.e. is nonmonotonic. Such a rule would make the above induction easy and maybe obvious.

17. There are four cannibals and four missionaries, but if the strongest of the missionaries rows fast enough, the cannibals won't have gotten so hungry that they will eat the missionaries. This could be made precise in various ways, but the information is usable even in vague form.[7]

18. There are four missionaries and four cannibals, but the cannibals are not hungry initially, and the missionaries have a limited amount of cannibal food. They can tell if a cannibal is hungrier than he was and can avoid trouble by giving the food to the cannibal who has got hungrier. This requires comparing a situation and a successor situation.

19. There are two sets of missionaries and cannibals too far apart along the river to interact. The two problem should be solvable separately without considering interleaving actions at the two sites. If the two problems are different elaborations, the work required and the length of the proof should be the sum of the lengths for the separate problems plus a small constant.

    The theory of two sets of missionaries should be a *conservative extension* of each of the subtheories. We have called this property *conjunctivity*.

    There are N sites along the river with identical conditions. The reasoning should be able to do one site, or a generalized site, and, with a constant amount of additional reasoning, say that all N crossings are the same.

[7] "Pull, pull, my good boys", said Starbuck.—Moby Dick, XLVIII

20. After rowing twice, a person becomes too tired to row any more. [Added 2003 April 1].

10.8 Remarks and Acknowledgements

1. The English language elaborations don't refer to an original English text. If someone has read about the problem and understands it, he usually won't be able to quote the text he read. Moreover, if he tells the problem to someone else more than once, he is unlikely to use the same words each time. We conclude from this that that a person's understanding of MCP is represented in the brain in some other way than as an English text. For the purposes of this paper we don't need to speculate about how it is represented, since the formal elaboration tolerance applies to logical formulations.

2. Some commonly adopted conventions in theories of actions interfere with elaboration tolerance. An example is identifying situations or events with intervals of time. You can get away with it sometimes, but eventually you will be sorry. For example, you may want to say that a good move is one that leads to a better situation with

$$Good(a, s) \equiv s <_{good} Result(a, s).$$

3. *Elaboration tolerance* and *belief revision* have much in common, but we are looking at the problem from the opposite direction from researchers in belief revision. Belief revision studies have mainly concerned the effect of adding or removing a given sentence, whereas our treatment of elaboration tolerance concerns what you must add or change to get the effect you want. Moreover, the effect of an elaboration can involve changing the first order language and not just replacing one expression in the language by another.

4. Elaboration tolerance is rather straightforward when the theory to be changed has the structure of a cartesian product, and the elaboration can be describes as giving some components of the product new values. (McCarthy 1979b) discusses theories with cartesian product structures in connection with counterfactuals, and (McCarthy 1962c) discusses the semantics of assignment, i.e. the semantics of changing components of a state.

5. Murray Shanahan (Shanahan 1997) considers many issues of elaboration tolerance in his discussions of action formalisms. In particular, his solutions for the frame problem are considerably elaboration tolerant. I qualified the above, because I consider elaboration tolerance an open ended problem.

6. I suspect that elaboration tolerance requires a proper treatment of *hypothetical causality* and this involves *counterfactual conditional* sentences. Counterfactuals will be treated in a shortly forthcoming paper by Tom Costello and John McCarthy. For example, we need a non-trivial interpretation of "If another car had come over the hill while you were passing, there would have been a head-on collision" that is compatible with the fact that no car came. By non-trivial interpretation, I mean one that could have as a consequence that a person should change his driving habits, whereas no such conclusion can be reached from sentences of the form $p \rightarrow q$ when p is false.

7. We can distinguish between a formalism admitting a particular elaboration and the consequences of the elaboration being entirely determined. For example, the Jesus Christ elaboration could be given alternate interpretations and not just the one about his ability to walk on water.

   Another example (suggested by Tom Costello) has the original story say that the capacity of the boat is one less than the number of missionaries. Then changing the number of missionaries and cannibals to 4 leaves the problem still solvable, even though the set of logical consequences of the sentences of the two formalisms is the same. This tells us that if we translate the English to logic and take all logical consequences, information that determines the effects of elaborations can be lost.

This chapter has benefitted from discussions with Eyal Amir, Tom Costello, Aarati Parmar and Josephina Sierra.

Chapter 11

THEORIES OF APPROXIMATE OBJECTS

11.1 Difficulties with semantics

Consider the blocks world. A simple blocks world theory $T1$ has relations like $On(Block1, Block2)$. The language cannot express anything about the precise location of $Block1$ on $Block2$. Suppose now that we have a richer theory $T2$ in which one block can be displaced by a vector (x, y) from being centered on another, but the edges are still required to be parallel. For this we can use a predicate $On(b1, b2, x, y)$. $T1$ is then an aproximation to $T2$, and we could regard a description of a table with blocks in $T1$ to be an approximation to a description in $T2$. A still richer theory $T3$ allows one block to be at an angle on another, whereas $T4$ lets the blocks be rectangular parallelopipeds of arbitrary dimensions.

We can have an equation like

$$Ist(T1, On(b1, b2)) \equiv (\exists x\ y)(Ist(T2, On(b1, b2, x, y))). \tag{11.1}$$

If we allow ourselves to change the predicate symbols in order to merge $T1$ and $T2$, we can write the simpler formula

$$On1(b1, b2) \equiv (\exists x\ y)On2(b1, b2, x, y). \tag{11.2}$$

However, we prefer the formulation using contexts, because it allows keeping $T1$ and $T2$ intact and relating them via context.

Now consider the semantics of $T1$, supposing that we designed $T1$ first. It has a relation $On(b1, b2)$ and doesn't know about x and y. The semantics of $T2$ is quite different. How should we relate them?

This Berkeley abstract attracted quite a few requests for the paper.

Chapter 12

Consciousness in AI systems

Conscious knowledge and other information is distinguished from unconscious information by being observable, and its observation results in conscious knowledge about it. We call this introspective knowledge.

A robot will need to use introspective knowledge in order to operate in the common sense world and accomplish the tasks humans will give it.

Many features of human consciousness will be wanted, some will not, and some abilities not possessed by humans have already been found feasible and useful in limited domains.

We give preliminary fragments of a logical language a robot can use to represent information about its own state of mind.

A robot will often have to conclude that it cannot decide a question on the basis of the information in memory and therefore must seek information externally.

Programs with much introspective consciousness do not yet exist.

Thinking about consciousness with a view to designing it provides a new approach to some of the problems of consciousness studied by philosophers. One advantage is that it focusses on the aspects of consciousness important for intelligent behavior. If the advocates of qualia are right, it looks like robots won't need them to exhibit any behavior exhibited by humans.

12.1 Introduction

For the purposes of this article a robot is a continuously acting computer program interacting with the outside world and not normally stopping. What

physical senses and effectors or communication channels it has are irrelevant to this discussion except as examples.

This article discusses consciousness with the methodology of logical AI. (McCarthy 1989) contains a recent discussion of logical AI. AI systems that don't represent information by sentences can have only limited introspective knowledge.

12.1.1 About Logical AI

(McCarthy 1959) proposed programs with common sense that represent what they know about particular situations and the world in general *primarily* by sentences in some language of mathematical logic. They decide what to do *primarily* by logical reasoning, i.e. when a logical AI program does an important action, it is usually because it inferred a sentence saying it should. There will usually be other data structures and programs, and they may be very important computationally, but the main decisions of what do are made by logical reasoning from sentences explicitly present in the robot's memory. Some of the sentences may get into memory by processes that run independently of the robot's decisions, e.g. facts obtained by vision. Developments in logical AI include situation calculus in various forms, logical learning, nonmonotonic reasoning in various forms ((McCarthy 1980), (McCarthy 1986), (**?**), (Lifschitz 1994)), theories of concepts as objects (McCarthy 1979c) and theories of contexts as objects (McCarthy 1993), (McCarthy and Buvač 1998). (McCarthy 1959) mentioned self-observation but wasn't specific.

There have been many programs that decide what do by logical reasoning with logical sentences. However, I don't know of any that are *conscious* of their own ongoing mental processes, i.e. bring sentences *about* the sentences generated by these processes into memory *along with them*. We hope to establish in this article that some consciousness of their own mental processes will be required for robots to reach a level intelligence needed to do many of the tasks humans will want to give them. In our view, **consciousness of self, i.e. introspection, is essential for human level intelligence and not a mere epiphenomenon**. However, we need to distinguish which aspects of human consciousness need to be modelled, which human qualities need not and where AI systems can go beyond human consciousness.

12.1.2 Ascribing mental qualities to systems

A system, e.g. a robot, can be ascribed beliefs provided sentences expressing these beliefs have the right relation to the system's internal states, inputs and output and the goals we ascribe to it. (Dennett 1971) and (Dennett 1978) calls such ascriptions the *intentional stance*. The beliefs need not be explicitly represented in the memory of the system. Also Allen Newell, (**?**) regarded some information not represented by sentences explicitly present in memory as nevertheless representing sentences or propositions believed by the system. Newell called this the *logic level*. I believe he did not advocate general purpose programs that represent information primarily by sentences.[1] I do.

(McCarthy 1979b) goes into detail about conditions for ascribing belief and other mental qualities.

> To ascribe certain *beliefs*, *knowledge*, *free will*, *intentions*, *consciousness*, *abilities* or *wants* to a machine or computer program is *legitimate* when such an ascription expresses the same information about the machine that it expresses about a person. It is *useful* when the ascription helps us understand the structure of the machine, its past or future behavior, or how to repair or improve it. It is perhaps never *logically required* even for humans, but expressing reasonably briefly what is actually known about the state of a machine in a particular situation may require ascribing mental qualities or qualities isomorphic to them.

(McCarthy 1979b) considers systems with very limited beliefs. For example, a thermostat may usefully be ascribed one of exactly three beliefs—that the room is too cold, that it is too warm or that its temperature is ok. This is sometimes worth doing even though the thermostat may be completely understood as a physical system.

Tom Costello pointed out to me that a simple system that doesn't use sentences can sometimes be ascribed some introspective knowledge. Namely, an electronic alarm clock getting power after being without power can be said to know that it doesn't know the time. It asks to be reset by blinking its display. The usual alarm clock can be understood just as well by the design

[1] Newell, together with Herbert Simon and other collaborators used logic as a domain for AI in the 1950s. Here the AI was in programs for making proofs and not in the information represented in the logical sentences.

stance as by the intentional stance. However, we can imagine an alarm clock that had an interesting strategy for getting the time after the end of a power failure. In that case, the ascription of knowledge of non-knowledge might be the best way of understanding that part of the state.

12.1.3 Consciousness and introspection

We propose to design robot *consciousness* with explicitly represented beliefs as follows. At any time a certain set of sentences are directly available for reasoning. We call these the robot's *awareness*. Some of them, perhaps all, are available for observation, i.e. processes can generate sentences about these sentences. These sentences constitute the robot's *consciousness*. In this article, we shall consider the awareness and the consciousness to coincide; it makes the discussion shorter.

Some sentences come into consciousness by processes that operate all the time, i.e. by *involuntary subconscious processes*. Others come into *consciousness* as a result of *mental actions*, e.g. observations of its consciousness, that the robot *decides* to take. The latter are the results of *introspection* and constitute *self-consciousness*.

Here's an example of human introspection. Suppose I ask you whether the President of the United States is standing, sitting or lying down at the moment, and suppose you answer that you don't know. Suppose I then ask you to think harder about it, and you answer that no amount of thinking will help. (Kraus et al. 1991) has one formalization. A certain amount of introspection is required to give this answer, and robots will need a corresponding ability if they are to decide correctly whether to think more about a question or to seek the information they require externally. [2]

We discuss what forms of consciousness and introspection are required

[2]Here's an ancient example of observing one's likes and not knowing the reason.

> "Non amo te, Zabidi, nec possum dicere quare;
> Hoc tantum possum dicere, non amo te."

by Martial which Tom Brown translated to

> I do not like thee, Dr. Fell
> The reason why I cannot tell,
> But this I know, I know full well,
> I do not like thee, Dr. Fell.

for robots and how some of them may be formalized. It seems that the designer of robots has many choices to make about what features of human consciousness to include. Moreover, it is very likely that useful robots will include some introspective abilities not fully possessed by humans.

Two important features of consciousness and introspection are the ability to infer nonknowledge and the ability to do nonmonotonic reasoning.

12.2 What Consciousness does a Robot Need?

12.2.1 Easy introspection

In some respects it is easy to provide computer programs with more powerful introspective abilities than humans have. A computer program can inspect itself, and many programs do this in a rather trivial way by computing check sums in order to verify that they have been read into computer memory without modification.

It is easy to make available for inspection by the program the manuals for the programming language used, the manual for the computer itself and a copy of the compiler. A computer program can use this information to simulate what it would do if provided with given inputs. It can answer a question like: "Would I print "YES" in less than 1,000,000 steps for a certain input? A finitized version of Turing's argument that the *halting problem* is unsolvable tells us that that a computer cannot in general answer questions about what it would do in n steps in less than n steps. If it could, we (or a computer program) could construct a program that would answer a question about what it would do in n steps and then do the opposite.

We humans have rather weak memories of the events in our lives, especially of intellectual events. The ability to remember its entire intellectual history is possible for a computer program and can be used by the program in modifying its beliefs on the basis of new inferences or observations. This may prove very powerful.

Very likely, computer programs can be made to get more from reading itself than we presently know how to implement.

The dual concept to programs reading themselves is that of programs modifying themselves. Before the invention of index registers (B-lines) at Manchester, programs did indexing through arrays and telling subroutines where to return by program modification. It was sometimes stated that self-

modification was one of the essential ideas of using the same memory for programs and data. This idea went out of fashion when major computers, e.g. the IBM 704 in 1955, had index registers.

As AI advances, programs that modify themselves in substantial ways will become common. However, I don't treat self-modification in this article.

Unfortunately, these easy forms of introspection are insufficient for intelligent behavior in many common sense information situations.

12.2.2 Serious introspection

To do the tasks we will give them, a robot will need many forms of self-consciousness, i.e. ability to observe its own mental state. When we say that something is *observable*, we mean that a suitable *action* by the robot causes a sentence and possibly other data structures giving the result of the observation to appear in the robot's consciousness.

This section uses two formalisms described in previous papers.

The first is the notion of a context as a *first class object* introduced in (**?**) and developed in (McCarthy 1993) and (McCarthy and Buvač 1998). As first class objects, contexts can be the values of variables and arguments and values of functions. The most important expression is $Ist(c, p)$, which asserts that the proposition p is true in the context c. Propositions true in *subcontexts* need not be true in *outer contexts*. The language of a subcontext can also be an abbreviated version of the language of an outer context, because the subcontext can involve some assumptions not true in outer contexts. A reasoning system can *enter* a subcontext and reason with the assumptions and in the language of the subcontext. If we have $Ist(c, p)$ in an outer context $c0$, we can write

$$c : \quad p,$$

and reason directly with the sentence p. Much human reasoning, maybe all, is done in subcontexts, and robots will have to do the same. There is no most general context. The outermost context used so far can always be *transcended* to a yet outer context. A sentence $Ist(c, p)$ represents a kind of introspection all by itself.

The second important formalism is that of a *proposition* or *individual concept* as a first class object distinct from the truth value of the proposition or the value of the individual concept. This allows propositions and individual

concepts to be discussed formally in logical language rather than just informally in natural language. One motivating example from (McCarthy 1979c) is given by the sentences

$$\begin{array}{l} denotation(Telephone(Person)) = telephone(denotation(Person)) \\ denotation(Mike) = mike \\ telephone(mike) = telephone(mary) \\ knows(pat, Telephone(Mike)) \\ \neg knows(pat, Telephone(Mary)). \end{array} \tag{12.1}$$

Making the distinction between concepts and their denotation allows us to say that Pat knows Mike's telephone number but doesn't know Mary's telephone number even though Mary's telephone number is the same as Mike's telephone number. (McCarthy 1979c) uses capitalized words for concepts and lower case for objects. This is contrary to the convention in the rest of this paper that capitalizes constants and uses lower case for variables.

We will give tentative formulas for some of the results of observations. In this we take advantage of the ideas of (McCarthy 1993) and (McCarthy and Buvač 1998) and give a context for each formula. This makes the formulas shorter. What $Here$, Now and I mean is determined in an outer context.

- Observing its physical body, recognizing the positions of its effectors, noticing the relation of its body to the environment and noticing the values of important internal variables, e.g. the state of its power supply and of its communication channels. Already a notebook computer is aware of the state of its battery.

$$\ldots : C(Here, Now, I) : Lowbattery \wedge In(Screwdriver, Hand3) \tag{12.2}$$

  [No reason why the robot shouldn't have three hands.]

- Observing that it does or doesn't know the value of a certain term, e.g. observing whether it knows the telephone number of a certain person. Observing that it does know the number or that it can get it by some procedure is likely to be straightforward. However, observing that it doesn't know the telephone number and cannot infer what it is involves getting around Gödel's second incompleteness theorem. The reason we have to get around it is that showing that any sentence is

not inferrable says that the theory is consistent, because if the theory is inconsistent, all sentences are inferrable. Section 12.5 shows how do this using Gödel's idea of relative consistency. Consider

$$C(Now, I) : \neg Know(Telephone(Clinton)) \tag{12.3}$$

and

$$C(Now, I) : \neg Know\text{-}whether(Sitting(Clinton)). \tag{12.4}$$

Here, as discussed in (McCarthy 1979c), $Telephone(Clinton)$ stands for the *concept* of Clinton's telephone number, and $Sitting(Clinton)$ is the *proposition* that Clinton is sitting.

Deciding that it doesn't know and cannot infer the value of a telephone number is what should motivate the robot to look in the phone book or ask someone.

- The robot needs more than just the ability to observe that it doesn't know whether a particular sentence is true. It needs to be able to observe that it doesn't know anything about a certain subject, i.e. that anything about the subject is possible. Thus it needs to be able to say that the members of Clinton's cabinet may be in an arbitrary configuration of sitting and standing. This is discussed in Section 12.5.1.

- Reasoning about its abilities. "I think I can figure out how to do this". "I don't know how to do that."

- Keeping a journal of physical and intellectual events so it can refer to its past beliefs, observations and actions.

- Observing its goal structure and forming sentences about it. Notice that merely having a stack of subgoals doesn't achieve this unless the stack is observable and not merely obeyable. This lets it notice when a subgoal has become irrelevant to a larger goal and then abandon it.

- The robot may *intend* to perform a certain action. It may later infer that certain possibilities are irrelevant in view of its intentions. This requires the ability to observe intentions.

- It may also be able to say, "I can tell you how I solved that problem" in a way that takes into account its mental search processes and not just its external actions.

- The obverse of a goal is a constraint. Maybe we will want something like Asimov's science fiction laws of robotics, e.g. that a robot should not harm humans. In a sufficiently general way of looking at goals, achieving its other goals with the constraint of not harming humans is just an elaboration of the goal itself. However, since the same constraint will apply to the achievement of many goals, it is likely to be convenient to formalize them as a separate structure. A constraint can be used to reduce the space of achievable states before the details of the goals are considered.

- Observing how it arrived at its current beliefs. Most of the important beliefs of the system will have been obtained by nonmonotonic reasoning, and therefore are usually uncertain. It will need to maintain a critical view of these beliefs, i.e. believe meta-sentences about them that will aid in revising them when new information warrants doing so. It will presumably be useful to maintain a pedigree for each belief of the system so that it can be revised if its logical ancestors are revised. *Reason maintenance systems* maintain the pedigrees but not in the form of sentences that can be used in reasoning. Neither do they have introspective subroutines that can observe the pedigrees and generate sentences about them.

- Not only pedigrees of beliefs but other auxiliary information should either be represented as sentences or be observable in such a way as to give rise to sentences. Thus a system should be able to answer the questions: "Why do I believe p?" or alternatively "Why don't I believe p?".

- Regarding its entire mental state *up to the present* as an object, i.e. a context. (McCarthy 1993) discusses contexts as formal objects. The ability to *transcend* one's present context and think about it as an object is an important form of introspection. The restriction to *up to the present* avoids the paradoxes of self-reference and still preserves the useful generality.

- Knowing what goals it can currently achieve and what its choices are for action. (McCarthy and Hayes 1969b) showed how a robot could think about its own "free will" by considering the effects of the actions it might take, not taking into account its own internal processes that decide on which action to take.

- A simple (and basic) form of free will is illustrated in the situation calculus formula that asserts that John will do the action that John thinks results in the better situation for him.

$$\begin{array}{l} Occurs(Does(John, \\ \quad \textbf{if} \\ \text{Thinks-better}(John, Result(Does(John, a1), s), Result(Does(John, a2), s)) \\ \quad \textbf{then } a1 \\ \quad \textbf{else } a2 \\ \qquad), s). \end{array} \tag{12.5}$$

  Here Thinks-better$(John, s1, s2)$ is to be understood as asserting that John thinks $s1$ is better for him than $s2$.

- Besides specific information about its mental state, a robot will need general facts about mental processes, so it can plan its intellectual life.

- There often will be auxiliary goals, e.g. curiosity. When a robot is not otherwise occupied, we will want it to work at extending its knowledge.

- Probably we can design robots to keep their goals in a proper hierarchy so that they won't ever have to say, "I wish I didn't want to smoke."

The above are only some of the needed forms of self-consciousness. Research is needed to determine their properties and to find additional useful forms of self-consciousness.

12.2.3 Understanding and Awareness

We do not offer definitions of understanding and awareness. Instead we discuss which abilities related to these phenomena robots will require.

Consider fish swimming. Fish do not understand swimming in the following senses.

- A fish cannot, while not swimming, review its previous swimming performance so as to swim better next time.
- A fish cannot take instruction from a more experienced fish in how to swim better.
- A fish cannot contemplate designing a fish better adapted to certain swimming conditions than it is.

A human swimmer may understand more or less about swimming. [3]

We contend that intelligent robots will need understanding of how they do things in order to improve their behavior in ways that fish cannot. Aaron Sloman (**?**) has also discussed understanding, making the point that understanding is not an all-or-nothing quality.

Consider a robot that swims. Besides having a program for swimming with which it can interact, a logic-based robot needs to use sentences about swimming in order to give instructions to the program and to improve it. This includes sentences about how fast or how long it can swim.

The *understanding* a logical robot needs then requires it to use appropriate sentences about the matter being understood. The understanding involves both getting the sentences from observation and inference and using them appropriately to decide what to do.

Awareness is similar. It is a process whereby appropriate sentences about the world and its own mental situation come into the robot's consciousness, usually without intentional actions. Both understanding and awareness may be present to varying degrees in natural and artificial systems. The swimming robot may understand some facts about swimming and not others, and it may be aware of some aspects of its current swimming state and not others.

12.3 Formalized Self-Knowledge

We assume a system in which a robot maintains its information about the world and itself primarily as a collection of sentences in a mathematical logical language. There will be other data structures where they are more

[3] One can understand aspects of a human activity better than the people who are good at doing it. Nadia Comenici's gymnastics coach was a large, portly man hard to imagine cavorting on a gymnastics bar. Nevertheless, he *understood* women's gymnastics well enough to have coached a world champion.

compact or computationally easier to process, but they will be used by programs whose results become stored as sentences. The robot decides what to do by logical reasoning, by deduction using rules of inference and also by nonmonotonic reasoning.

We do not attempt a full formalization of the rules that determine the effects of mental actions and other events in this paper. The main reason is that we are revising our theory of events to handle concurrent events in a more modular way. This is discussed in the draft (McCarthy 1995b) and further in (McCarthy and Costello 1998).

Robot consciousness involves including among its sentences some about the robot itself and about subsets of the collection of sentences itself, e.g. the sentences that were in consciousness just previous to the introspection, or at some previous time, or the sentences about a particular subject. [4]

We say subsets in order to avoid self-reference as much as possible. References to the totality of the robot's beliefs can usually be replaced by references to the totality of its beliefs up to the present moment.

12.3.1 Mental Situation Calculus

The *situation calculus*, initiated in (McCarthy 1963) and (McCarthy and Hayes 1969a), is often used for describing how actions and other events affect the world. It is convenient to regard a robot's state of mind as a component of the situation and describe how mental events give rise to new situations. (We could use a formalism with a separate mental situation affected only by mental events, but this doesn't seem to be advantageous.) We contemplate a system in which what *holds* is closed under deductive inference, but *knowledge* is not.

The relevant notations are:

- $Holds(p, s)$ is the assertion that the proposition p holds in the situation s. We shall mainly be interested in propositions p of a mental nature.

[4] Too much work concerned with self-knowledge has considered self-referential sentences and getting around their apparent paradoxes. This is mostly a distraction for AI, because human self-consciousness and the self-consciousness we need to build into robots almost never involves self-referential sentences or other self-referential linguistic constructions. A simple reference to oneself is not a self-referential linguistic construction, because it isn't done by a sentence that refers to itself.

- Among the propositions that can hold are $Know(p)$ and $Believe(p)$, where p again denotes a proposition. Thus we can have

$$Holds(Know(p), s). \tag{12.6}$$

- As we will shortly see, sentences like

$$Holds(Know(Not\ Know(p), s) \tag{12.7}$$

  are often useful. The sentence(12.7) asserts that the robot knows it doesn't know p.

- Besides knowledge of propositions we need a notation for knowledge of an *individual concept*, e.g. a telephone number. (McCarthy 1979c) treats this in some detail. That paper has separate names for objects and concepts of objects and the argument of knowing is the latter. The symbol $mike$ denotes Mike himself, the function telephone takes a person into his telephone number. Thus $telephone(mike)$ denotes Mike's telephone number. The symbol $Mike$ is the concept of Mike, and the function $Telephone$ takes a the concept of a person into the concept of his telephone number. Thus we distinguish between Mike's telephone number, denoted by $telephone(mike)$ and the concept of his telephone number denoted by $Telephone(Mike)$.

  The convention used in this section of *telephone* and $Telephone$ is different from the convention in the rest of the article of using capital letters to begin constants (whether individual, functional or predicate constants) and using symbols in lower case letters to denote variables.

  This enables us to say

$$Holds(Knows(Telephone(Mike)), s) \tag{12.8}$$

  to assert knowledge of Mike's telephone number and

$$Holds(Know(Not(Knows(Telephone(Mike)))), s) \tag{12.9}$$

  to mean that the robot knows it doesn't know Mike's telephone number. The notation is somewhat ponderous, but it avoids the unwanted

inference that the robot knows Mary's telephone number from the facts that her telephone number is the same as Mike's and that the robot knows Mike's telephone number.[5] Having the sentence (12.9) in consciousness might stimulate the robot to look in the phone book.

12.3.2 Mental events, especially mental actions

Mental events change the situation just as do physical events.

Here is a list of some mental events, mostly described informally.

- In the simplest formalisms mental events occur sequentially. This corresponds ot a *stream of consciousness*. Whether or not the idea describes human consciousness, it is a design option for robot consciousness.

- $Learn(p)$. The robot learns the fact p. An obvious consequence is

$$Holds(Know(p), Result(Learn(p), s)) \tag{12.10}$$

  provided the effects are definite enough to justify the $Result$ formalism. More likely we'll want something like

$$Occurs(Learn(p), s) \rightarrow Holds(F\ Know(p), s), \tag{12.11}$$

  where $Occurs(event, s)$ is a *point fluent* asserting that *event* occurs (instantaneously) in situation s. $F(p)$ is the proposition that the proposition p will be true at some time in the future. The *temporal function* F is used in conjunction with the function $next$ and the axiom

$$Holds(F(p), s) \rightarrow Holds(p, Next(p, s)). \tag{12.12}$$

  Here $Next(p, s)$ denotes the next situation following s in which p holds. (12.12) asserts that if $F(p)$ holds in s, then there is a next situation in which p holds. (This $Next$ is not the $Next$ operator used in some temporal logic formalisms.)

[5]Some other formalisms give up the law of substitution in logic in order to avoid this difficulty. We find the price of having separate terms for concepts worth paying in order to retain all the resources of first order logic and even higher order logic when needed.

- The robot learning p has an effect on the rest of its knowledge. We are not yet ready to propose one of the many *belief revision* systems for this. Indeed we don't assume logical closure.

- What about an event $Forget(p)$? Forgetting p is definitely not an event with a definite result. What we can say is

$$Occurs(Forget(p), s) \rightarrow Holds(F(Not(Know(p))), s) \qquad (12.13)$$

  In general, we shall want to treat forgetting as a side-effect of some more complex event. Suppose Foo is the more complex event. We'll have

$$Occurs(foo, s) \rightarrow Occurs(Forget(p), s) \qquad (12.14)$$

- The robot may decide to do action a. This has the property:

$$Occurs(Decide\text{-}to\text{-}do\ a, s) \rightarrow Holds(Intend\text{-}to\text{-}do\ a, s). \qquad (12.15)$$

  The distinction is that *Decide* is an event, and we often don't need to reason about how long it takes. *Intend-to-do* is a fluent that persists until something changes it. Some call these *point fluents* and *continuous fluents* respectively.

- The robot may decide to assume p, e.g. for the sake of argument. The effect of this action is not exactly to believe p, but rather involves *entering a context* $Assume(c, p)$ in which p holds. This formalism is described in (McCarthy 1993) and (McCarthy and Buvač 1998).

- The robot may infer p from other sentences, either by deduction or by some nonmonotonic form of inference.

- The robot may see some object. One result of seeing an object may be knowing that it saw the object. So we might have

$$Occurs(See\ o, s) \rightarrow Holds(F\ Knows\ Did\ See\ o, s). \qquad (12.16)$$

  Formalizing other effects of seeing an object require a theory of seeing that is beyond the scope of this article.

It should be obvious to the reader that we are far from having a comprehensive list of the effects of mental events. However, I hope it is also apparent that the effects of a great variety of mental events on the mental part of a situation can be formalized. Moreover, it should be clear that useful robots will need to observe mental events and reason with facts about their effects.

Most work in logical AI has involve theories in which it can be shown that a sequence of actions will achieve a goal. There are recent extensions to concurrent action, continuous action and strategies of action. All this work applies to mental actions as well.

Mostly outside this work is reasoning leading to the conclusion that a goal cannot be achieved. Similar reasoning is involved in showing that actions are safe in the sense that a certain catastrophe cannot occur. Deriving both kinds of conclusion involves inductively inferring quantified propositions, e.g. "whatever I do the goal won't be achieved" or "whatever happens the catastrophe will be avoided." This is hard for today's automated reasoning techniques, but Reiter (**?**) and his colleagues have made important progress.

12.4 Logical paradoxes, Gödel's theorems, and self-confidence

You can't always get what you want,
But if you try, sometimes, you just might find,
You get what you need.
— Rolling Stones

Logical discoveries, mainly of the 20th century, impose limitations on the formalisms we can used without paradox. Other discoveries place limitations on what can be computed. In essence, the limitations apply to both people and machines, and intelligence can live within the limitations.

12.4.1 The paradoxes

It has precursors, but Russell's paradox of 1901 shows that the obvious set theory, as proposed by Frege has to be modified in unpleasant ways. Frege's

basic idea is to let us define the set of all objects having a given property, in more modern notation

$$\{x|\mathcal{P}(x)\},$$

giving the set of all x with the property $\mathcal{P}$. Thus the set of all red dogs is denoted by $\{x|dog(x) \wedge red(x)\}$, or if the set of dogs is denoted $dogs$ and the set of red objects as $reds$, we can also write $\{x|x \in dogs \wedge x \in reds\}$. This notation for forming sets is very convenient and is much used in mathematics. The principle is called *comprehension*.

Bertrand Russell in his 1901 letter to Gottlob Frege pointed out that forming the set

$$rp = \{x|\neg(x \in x)\},$$

i.e. the set of all sets that are not members of themselves, leads promptly to a contradiction. We get $rp \in rp \equiv \neg rp \in rp$.

There are many ways of restricting set theory to avoid the contradiction. The most commonly chosen is that of Zermelo, whose set theory Z allowed only writing $\{x \in A|\mathcal{P}(x)\}$, where A is a previously defined set. This turned out to be not quite enough to represent mathematics and Fraenkel introduce a further axiom schema of *replacement* giving a system now called ZF.

ZF is less convenient than Frege's inconsistent system because of the need to find the set A, and the unrestricted comprehension schema is often used when it is clear that the needed A could be found. [6]

A more direct inconvenience for giving robots consciousness is the paradox discovered by Richard Montague (Montague 1963) concerning a set of desirable axioms for knowledge of sentences.

We might denote by $knows(person, sentence)$ the assertion that $person$ knows $sentence$ and consider this as holding at some time t in in some situation s. However, Montague's paradox arises even when there is only one knower, and we write Kp for the knower knowing the sentence p. Montague's paradoxes arise under the assumption that the language of the sentences p is rich enough for "elementary syntax", i.e. allows quantifiers and operations on sentences or on Gödel numbers standing for sentences.

[6] For AI it might be convenient to use unrestricted comprehension as a default, with the default to the limited later by finding an A if necessary. This idea has not been explored yet.

The axioms are

$$Kp \rightarrow p, \tag{12.17}$$

$$Kp \rightarrow KKp, \tag{12.18}$$

and

$$K(Kp \wedge K(p \rightarrow q) \rightarrow Kq). \tag{12.19}$$

Intuitively these axioms state that if you know something, it's true, if you know something, you know you know it, and you can do modus ponens. Added to this are schemas saying that you know some sentences of elementary logic.

From these, Montague constructed a version of the paradox of the liar. Hence they must be weakened, and there are many weakenings that restore consistency. Montague preferred to leave out elementary syntax, thus getting a form of modal logic.

I think it might be better to weaken (12.18) by introducing a hierarchy of *introspective knowledge operators* on the idea that knowing that you know something is knowledge at an introspective level.

Suppose that we regard knowledge as a function of time or of the situation. We can slither out of Montague's paradox by changing the axiom $Kp \rightarrow KKp$ to say that if you knew something in the past, you now know that you knew it. This spoils Montague's recursive construction of the paradox.

None of this has yet been worked out for an AI system.

12.4.2 The incompleteness theorems

Gödel's first incompleteness theorem shows that any consistent logical theory expressive enough for elementary arithmetic, i.e. with addition, multiplication and quantifiers could express true sentences unprovable in the theory.

Gödel's second incompleteness theorem tells that the consistency of the system is one of these unprovable sentences.

The basis of Gödel's proof was the fact that the syntactic computations involved in combining formulas and verifying that a sequence of formulas is a proof can be imitated by arithmetic computations on "Gödel numbers" of formulas. If we have axioms for symbolic computations, e.g. for Lisp computations, then the proofs of Gödel's theorems become much shorter. Shankar (Shankar 1986) has demonstrated this using the Boyer-Moore prover.

Among the unprovable true sentences is the statement of the theory's own consistency. We can interpret this as saying that the theory lacks self-confidence. Turing, in his PhD thesis, studied what happens if we add to a theory T the statement $consis(T)$ asserting that T is consistent, getting a stronger theory T'. While the new theory has $consis(T)$ as a theorem, it doesn't have $consis(T')$ as a theorem—provided it is consistent. The process can be iterated, and the union of all these theories is $consis^{\omega}(T)$. Indeed the process can again be iterated, as Turing showed, to any constructive ordinal number.

12.4.3 Iterated self-confidence

Gödel's second incompleteness theorem (**?**) tells us that a consistent logical theory $T0$ strong enough to do Peano arithmetic cannot admit a proof of its own consistency. However, if we believe the theory $T0$, we will believe that it is consistent. We can add the statement $consis(T0)$ asserting that $T0$ is consistent to $T0$ getting a stronger theory $T1$. By the incompleteness theorem, $T1$ cannot admit a proof of $consis(T1)$, and so on. Adding consistency statement for what we already believe is a *self–confidence prinicple.*

Alan Turing (**?**) studied iterated statements of consistency, pointing out that we can continue the iteration of self-confidence to form $T\omega$, which asserts that all the Tn are consistent. Moreover, the iteration can be continued through the *recursive ordinal numbers.* Solomon Feferman (Feferman 1962) studied a more powerful iteration principle than Turing's called *transfinite progressions of theories.*

There is no single computable iterative self-confidence process that gets everything. If there were, we could put it in a single logical system, and Gödel's theorem would apply to it.

For AI purposes, $T1$, which is equivalent to induction up to the ordinal ϵ_0 may suffice.

The relevance to AI of Feferman's transfinite progressions is at least to refute naive arguments based on the incompleteness theorem that AI is impossible.

A robot thinking about self-confidence principles is performing a kind of introspection. For this it needs not only the iterates of $T0$ but to be able to think about theories in general, i.e. to use a formalism with variables ranging over theories.

12.4.4 Relative consistency

When we cannot prove a theory consistent, we can often show that it is consistent provided some other theory, e.g. Peano arithmetic or ZF is consistent.

In his (**?**), Gödel proved that if Gödel-Bernays set theory is consistent, then it remains consistent when the axiom of choice and the continuum hypothesis are added to the axioms. He did this by supposing that set theory has a model, i.e. there is a domain and an $\in$ predicate satisfying GB. He then showed that a subset of this domain, the constructible sets, provided a model of set theory in which the axiom of choice and the continuum hypothesis are also true. Paul Cohen proved in 1963 that if set theory has any models it has models in which the axiom of choice and the continuum hypothesis are false.

12.5 Inferring Non-knowledge

[This section and the next have a lot of redundancy. This will be fixed.]

Let p be a proposition. The proposition that the robot does not know p will be written $Not\ Know(p)$, and we are interested in those mental situations s in which we have $Holds(Not\ Know(p), s)$. If $Not\ p$ is consistent with the robot's knowledge, then we certainly want $Holds(Not\ Know(p), s)$.

How can we assert that the proposition *not p* is consistent with the robot's knowledge? Gödel's theorem tells us that we aren't going to do it by a formal proof using the robot's knowledge as axioms. [7] The most perfunctory approach is for a program to try to prove $Holds(not\ p, s)$ from the robot's knowledge and fail. Logic programming with negation as failure does this for Horn theories.

However, we can often do better. If a person or a robot regards a certain collection of facts as all that are relevant, it suffices to find a model of these facts in which p is false. [8]

[7] We assume that our axioms are strong enough to do symbolic computation which requires the same strength as arithmetic. I think we won't get much joy from weaker systems.

[8] A conviction of about what is relevant is responsible for a person's initial reaction to the well-known puzzle of the three activists and the bear. Three Greenpeace activists have just won a battle to protect the bears' prey, the bears being already protected. It was hard work, and they decide to go see the bears whose representatives they consider themselves to have been. They wander about with their cameras, each going his own way.

Consider asserting ignorance of the value of a numerical parameter. The simplest thing is to say that there are at least two values it could have, and therefore the robot doesn't know what it is. However, we often want more, e.g. to assert that the robot knows nothing of its value. Then we must assert that the parameter could have any value, i.e. for each possible value there are models of the relevant facts in which it has that value. Of course, complete ignorance of the values of two parameters requires that there be a model in which each pair of values is taken.

It is likely to be convenient in constructing these models to assume that arithmetic is consistent, i.e. that there are models of arithmetic. Then the set of natural numbers, or equivalently Lisp S-expressions, can be used to construct the desired models. The larger the robot's collection of theories postulated to have models, the easier it will be to show ignorance.

Making a program that reasons about models of its knowledge looks difficult, although it may turn out to be necessary in the long run. The notion of *transcending* a context may be suitable for this.

For now it seems more straightforward to use second order logic. The idea is to write the axioms of the theory with predicate and function variables and to use existential statements to assert the existence of models. Here's a proposal.

Suppose the robot has some knowledge expressed as an axiomatic theory and it needs to infer that it cannot infer *that* President Clinton is sitting down. We immediately have a problem with Gödel's incompleteness theorem, because if the theory is inconsistent, then every sentence is inferrable, and therefore a proof of non-inferrability of any sentence implies consistency. We get around this by using another idea of Gödel's—*relative consistency.*[9]

For example, suppose we have a first order theory with predicate symbols $\{P_1, \ldots, P_n, Sits\}$ and let $A(P_1, \ldots, P_n, Sits)$ be an axiom for the theory.

Meanwhile a bear wakes up from a long sleep very hungry and heads South. After three miles, she comes across one of the activists and eats him. She then goes three miles West, finds another activist and eats her. Three miles North she finds a third activist but is too full to eat. However, annoyed by the incessant blather, she kills the remaining activist and drags him two miles East to her starting point for a nap, certain that she and her cubs can have a snack when she wakes.

What color was the bear?

At first sight it seems that the color of the bear cannot be determined from the information given. While wrong in this case, jumping to such conclusions about what is relevant is more often than not the correct thing to do.

[9]Our approach is a variant of that used by (Kraus et al. 1991).

The second order sentence

$$(\exists P'_1, \ldots, P'_n\ sits')A(P'_1, \ldots, P'_n, sits') \tag{12.20}$$

expresses the consistency of the theory, and the sentence

$$(\exists P'_1, \ldots, P'_n\ sits')(A(P'_1, \ldots, P'_n, sits') \wedge \neg sits'(Clinton, s)) \tag{12.21}$$

expresses the consistency of the theory with the added assertion that Clinton is not sitting in the situation s. [In the above, we use upper case of the predicate constant $Sits$ and lower case for the variable $sits'$.

Then

$$(12.20) \rightarrow (12.21) \tag{12.22}$$

is then the required assertion of relative consistency.

Sometimes we will want to assert relative consistency under fixed interpretations of some of the predicate symbols. This would be important when we have axioms involving these predicates but do not have formulas for them, e.g. of the form $(\forall x\ y)(P(x, y) \equiv \ldots)$. Suppose, for example, that there are three predicate symbols $(P_1, P_2, Sits)$, and P_1 has a fixed interpretation, and the other two are to be chosen so as to satisfy the axiom. Then the assertion of consistency with Clinton sitting takes the form

$$(\exists P'_2 P'_3)A(P_1, P'_2, sits') \wedge sits'(Clinton, s). \tag{12.23}$$

The straightforward way of proving (12.23) is to find substitutions for the predicate variables P'_2 and $sits'$ that make the matrix of (12.23) true. The most trivial case of this would be when the axiom $A(P_1, P_2, Sits)$ does not actually involve the predicate $Sits$, and we already have an interpretation $P_1, \ldots, P_n, Sits$ in which it is satisfied. Then we can define

$$sits' = (\lambda x\ ss)(\neg(x = Clinton \wedge ss = s) \vee Sits(x, ss))), \tag{12.24}$$

and (12.23) follows immediately. This just means that if the new predicate does not interact with what is already known, then the values for which it is true can be assigned arbitrarily.

12.5.1 Existence of parameterized sets of models

Relative consistency provides a reasonable way of handling single cases of non-knowledge. However, we may want more. For example, suppose we want to say that we know nothing about whether any member of Clinton's cabinet is standing or sitting except (for example) that none of them sits when Clinton is standing in the same room.

The theory should then have lots of models, and we can parameterize them by a set of the standees that is arbitary except for the above condition. Here's a formula using non-knowledge.

$$\begin{aligned}&(\forall f)(f \in \{t,f\}^{\text{Clinton-cabinet}}\\ &\rightarrow (\forall x)(x \in \text{Clinton-cabinet}\\ &\quad \rightarrow \neg Know(Sits(x) \equiv f(x) = t)))\end{aligned} \tag{12.25}$$

but this only tells us that for each member of the cabinet, we don't know whether he is sitting.

We want the stronger formula

$$\begin{aligned}&(\forall f)(f \in \{t,f\}^{\text{Clinton-cabinet}}\\ &\neg Know(\neg(\forall x)(x \in \text{Clinton-cabinet}\\ &\quad Sits(x) \equiv f(x) = t)))\end{aligned} \tag{12.26}$$

which asserts that for all we know, Clinton's cabinet could be standing or sitting in an arbitrary pattern. Here we have had to take a quantifier inside the $Know$ function. (McCarthy 1979c) discusses difficulties in formalizing this and doesn't offer a satisfactory solution.

(**?**) gives a simple way of parameterizing the set of models of a propositional sentence. However, there can be no neat way of parameterizing the models of an arbitary first order theory. Thus parameterizing the set of axioms for group theory would amount to parameterizing the set of all groups, and group theory tells us that there is no straightforward parameterization.

12.5.2 Non-knowledge as failure

A system based on Horn clauses, e.g. a Prolog program, may treat non-knowledged as failure. Thus if both an attempt to prove Clinton to be sitting and an attempt to prove him standing fail, the system can infer that it doesn't know whether he is sitting or standing. This is likely to be easier than establishing that it is possible that he is standing and possible that he is sitting by finding models.

12.6 Humans and Robots

Human consciousness is undoubtedly more complicated than the design we propose for robots, but it isn't necessarily better.

The main complication I see is that human self observation, like human vision, is spotty. I pursue the analogy, because much more is accssible to observation and experiment with vision than with self observation.

Subjectively a person feels that he has a visual field with everything in the field accessible with approximately equal resolution. We also feel that colors are associated with points in the visual field. In fact, a person has a blind spot, resolution is much better in the small fovea than elsewhere, the perceived color of an object in the field has no simple relation to the light striking a corresponding point on the retina.

All this is because nature has evolved a vision system that finds out as much as possible about the world with very limited apparatus. For example, the usual objects have colors that can be recognized under varied lighting conditions as being the same color.

We have much less ability to observe human consciousness. However, it would be too good to be true if it consisted of a definite set of observable sentences.

12.6.1 A conjecture about human consciousness and its consequences for robots

There is a large difference between the human mind and the ape mind, and human intelligence evolved from ape-like intelligence in a short time as evolution goes. Our conjecture is that besides the larger brain, there is one qualitative difference—consciousness. The evolutionary step consisted of making more of the brain state itself observable than was possible for our ape-like ancestors. The consequence was that we could learn procedures that take into account the state of the brain, e.g. previous observations, knowledge or lack of it, etc.

The consequence for AI is that maybe introspection can be introduced into problem solving in a rather simple way—letting actions depend on the state of the mind and not just on the state of the external world as revealed by observation.

This suggests designing logical robots with observation as a subconscious process, i.e. mainly taking place in the background rather than as a result

of decisions. Observation results in sentences in consciousness. Deliberate observations should also be possible. The mental state would then be one aspect of the world that is subconsciously observed.

We propose to use contexts as formal objects for robot context, whereas context is mainly subconscious in humans. Perhaps robots should also deal with contexts at least partly subconsciously. I'd bet against it now.

[Much more to come when I get it clear.]
2002 July: It's still not sufficiently clear.

12.6.2 Robots Should Not be Equipped with Human-like Emotions

Human emotional and motivational structure is likely to be much farther from what we want to design than is human consciousness from robot consciousness.[10]

Some authors, (**?**), have argued that sufficiently intelligent robots would automatically have emotions somewhat like those of humans. However, I think that it would be possible to make robots with human-like emotions, but it would require a special effort distinct from that required to make intelligent robots. In order to make this argument, it is necessary to assume something, as little as possible, about human emotions. Here are some points.

1. Human reasoning operates primarily on the collection of ideas of which the person is immediately conscious.

2. Other ideas are in the background and come into consciousness by various processes.

3. Because reasoning is so often nonmonotonic, conclusions can be reached on the basis of the ideas in consciousness that would not be reached if certain additional ideas were also in consciousness. [11]

[10] Cindy Mason in her Emotional Machines home page (http://www.emotionalmachines.com/) expresses a different point of view.

[11] These conclusions are true in the simplest or most standard or otherwise minimal models of the ideas taken in consciousness. The point about nonmonotonicity is absolutely critical to understanding these ideas about emotion. See, for example, (McCarthy 1980) and (McCarthy 1986)

4. Human emotions influence human thought by influencing what ideas come into consciousness. For example, anger brings into consciousness ideas about the target of anger and also about ways of attacking this target.

5. According to these notions, paranoia, schizophrenia, depression and other mental illnesses would involve malfunctions of the chemical mechanisms that gate ideas into consciousness. A paranoid who believes the CIA is following him and influencing him with radio waves can lose these ideas when he takes his medicine and regain them when he stops. Certainly his blood chemistry cannot encode complicated paranoid theories, but they can bring ideas about threats from wherever or however they are stored.

6. Hormones analogous to neurostransmitters open synaptic gates to admit whole classes of beliefs into consciousness. They are analogs of similar substances and gates in animals.

7. A design that uses environmental or internal stimuli to bring whole classes of ideas into consciousness is entirely appropriate for a lower animals. We inherit this mechanism from our animal ancestors.

8. Building the analog of a chemically influenced gating mechanism would require a special effort.

These facts suggest the following design considerations.

1. We don't want robots to bring ideas into consciousness in an uncontrolled way. Robots that are to react against people (say) considered harmful, should include such reactions in their goal structures and prioritize them together with other goals. Indeed we humans advise ourselves to react rationally to danger, insult and injury. "Panic" is our name for reacting directly to perceptions of danger rather than rationally.

2. Putting such a mechanism, e.g. panic, in a robot is certainly feasible. It could be done by maintaining some numerical variables, e.g. level of fear, in the system and making the mechanism that brings sentences into consciousness (short term memory) depend on these variables. However, such human-like emotional structures are not an automatic byproduct of human-level intelligence.

3. Another aspect of the human mind that we shouldn't build into robots is that subgoals, e.g. ideas of good and bad learned to please parents, can become independent of the larger goal that motivated them. Robots should not let subgoals come to dominate the larger goals that gave rise to them.

4. It is also practically important to avoid making robots that are reasonable targets for either human sympathy or dislike. If robots are visibly sad, bored or angry, humans, starting with children, will react to them as persons. Then they would very likely come to occupy some status in human society. Human society is complicated enough already.

12

12.7 Remarks

1. In (**?**), Thomas Nagel wrote *"Perhaps anything complex enough to behave like a person would have experiences. But that, if true, is a fact that cannot be discovered merely by analyzing the concept of experience."*. This article supports Nagel's conjecture, both in showing that complex behavior requires something like conscious experience, and in that discovering it requires more than analyzing the concept of experience.

2. Already (**?**) disposes of "the claim that a machine cannot be the subject of its own thought". Turing further remarks

> By observing the results of its own behavior it can modify its own programs so as to achieve some purpose more effectively. These are possibilities of the near future rather than Utopian dreams.

[12]2001: The Steven Spielberg movie, *Artificial Intelligence* illustrates dangers of making robots that partly imitate humans and inserting them into society. I say "illustrates" rather "than provides evidence for", because a movie can illustrate any proposition the makers want, unrestricted by science or human psychology. In the movie, a robot boy is created to replace a lost child. However, the robot does not grow and is immortal and therefore cannot fit into a human family, although they depict it as programmed to love the bereaved mother. It has additional gratuitous differences from humans.

The movie also illustrates Spielberg's doctrines about environmental disaster and human prejudice against those who are different.

We want more than than Turing explicitly asked for. The machine should oberve its processes in action and not just the results.

3. The preceding sections are not to be taken as a theory of human consciousness. We do not claim that the human brain uses sentences as its primary way of representing information.

   Of course, logical AI involves using actual sentences in the memory of the machine.

4. Daniel Dennett (**?**) argues that human consciousness is not a single place in the brain with every conscious idea appearing there. I think he is partly right about the human brain, but I think a unitary consciousness will work quite well for robots. It would likely also work for humans, but evolution happens to have produced a brain with distributed consciousness.

5. John H. Flavell, (**?**) and (**?**), and his colleagues describe experiments concerning the introspective abilities of people ranging from 3 years old to adulthood. Even 3 year olds have some limited introspective abilities, and the ability to report on their own thoughts and infer the thoughts of others grows with age. Flavell, et. al. reference other work in this area. This is apparently a newly respectable area of experimental psychology, since the earliest references are from the late 1980s.

6. Francis Crick (**?**) discusses how to find *neurological correlates* of consciousness in the human and animal brain. I agree with all the philosophy in his paper and wish success to him and others using neuroscience. However, after reading his book, I think the logical artificial intelligence approach has a good chance of achieving human-level intelligence sooner. They won't tell as much about human intelligence, however.

7. What about *the unconscious?* Do we need it for robots? Very likely we will need some intermediate computational processes whose results are not appropriately included in the set of sentences we take as the *consciousness* of the robot. However, they should be observable when this is useful, i.e. sentences giving facts about these processes and their results should appear in consciousness as a result of mental actions

aimed at observing them. There is no need for a full-fledged Freudian unconscious with purposes of its own.

8. Should a robot hope? In what sense might it hope? How close would this be to human hope? It seems that the answer is yes and quite similar.. If it hopes for various things, and enough of the hopes come true, then the robot can conclude that it is doing well, and its higher level strategy is ok. If its hopes are always disappointed, then it needs to change its higher level strategy.

   To use hopes in this way requires the self observation to remember what it hoped for.

   Sometimes a robot must also infer that other robots or people hope or did hope for certain things.

9. The syntactic form is simple enough. If p is a proposition, then $Hope(p)$ is the proposition that the robot hopes for p to become true. In mental situation calculus we would write

$$Holds(Hope(p), s) \tag{12.27}$$

   to assert that in mental situation s, the robot hopes for p.

   Human hopes have certain qualities that I can't decide whether we will want. Hope automatically brings into consciousness thoughts related to what a situation realizing the hope would be like. We could design our programs to do the same, but this is more automatic in the human case than might be optimal. Wishful thinking is a well-known human malfunction.

10. A robot should be able to wish that it had acted differently from the way it has done. A mental example is that the robot may have taken too long to solve a problem and might wish that it had thought of the solution immediately. This will cause it to think about how it might solve such problems in the future with less computation.

11. A human can wish that his motivations and goals were different from what he observes them to be. It would seem that a program with such a wish could just change its goals. However, it may not be so simple if different subgoals each gives rise to wishes, e.g. that the other subgoals were different.

12. Programs that represent information by sentences but generate new sentences by processes that don't correspond to logical reasoning present similar problems to logical AI for introspection. Approaches to AI that don't use sentences at all need some other way of representing the results of introspection if they are to use it at all.

13. Psychologists and philosophers from Aristotle on have appealed to association as the main tool of thought. It is clearly inadequate to draw conclusions. We can make sense of their ideas by regarding association as the main tool for bringing facts into consciousness, but requiring reasoning to reach conclusions.

14. Some conclusions are reached by deduction, some by nonmonotonic reasoning and some by looking for models—alternatively by reasoning in second order logic.

15. Case based reasoning. Cases are *relatively rich* objects—or maybe we should say *locally rich*.

12.8 Acknowledgements

This work was partly supported by ARPA (ONR) grant N00014-94-1-0775 and partly done in 1994 while the author was Meyerhoff Visiting Professor at the Weizmann Institute of Science, Rehovot, Israel.

More recently, this research has been partly supported by ARPA contract no. USC 621915, the ARPA/Rome Laboratory planning initiative under grant (ONR) N00014-94-1-0775 and ARPA/AFOSR under (AFOSR) grant # F49620-97-1-0207.

Thanks to Yoav Shoham and Aaron Sloman for email comments and to Saša Buvač, Tom Costello and Donald Michie for face-to-face comments.

Chapter 13

PROBLEM SOLVING

Acting intelligently involves problem solving, i.e. finding an x such that $P(x)$. We can generalize this to finding $x_1, \ldots, x_n$ such that $P(x_1, \ldots, x_n)$, but this can be considered only superficially more general, because a single x can be a compound object.

We can formalize this a bit more by writing $Find(x, P(x))$ as the name of a problem. (or $Find(x_1, \ldots, x_n, P(x_1, \ldots, x_n))$) as the name of a problem-solving task.

13.0.1 Prolog and logic programming

Logic programming is the most straightforward form of problem solving.

13.1 Heuristics

When a computer executes a billion instructions to obtain a result a human gets in five minutes, the reason is not that the presumed parallelism of human wetware trumps the integrated circuits. The fault is not with the machine designers but with the programmers.

We don't have a proof of this contention, and it is certainly disputed. However, when one looks at what the computer is doing when it executes all these instructions, we can usually see that the program is looking at possibilities that it shouldn't.

13.2 Search

Chapter 14

A PLAN (NON-LINEAR) FOR HUMAN-LEVEL AI

In previous chapters we discussed theories of actions and change (using situation calculus), nonmonotonic reasoning, concepts as objects, contexts as objects, heuristic search, and theories with approximate objects.

How can we put all these topics into a plan for achieving human-level logical AI?

Acting intelligently involves problem solving, i.e. finding an x such that $P(x)$. We can generalize this to finding $x_1, \ldots, x_n$ such that $P(x_1, \ldots, x_n)$, but this can be considered only superficially more general, because a single x can be a compound object.

We can formalize this a bit more by writing $Find(x, P(x))$ as the name of a problem. (or $Find(x_1, \ldots, x_n, P(x_1, \ldots, x_n))$) as the name of a problem-solving task.

The most important x to find is a strategy of action to achieve a goal. Thus we have $Find(strat, Achieves(strat, g))$. In general a strategy has conditionals, loops and recursion, but almost all AI research has concentrated on the most straightforward case of finding a sequence of actions that achieves the goal g from an initial situation $S0$. Thus we have

$$Find(a_1, \ldots, a_n, g(Result(a_n, \ldots Result(a_1, S0) \ldots)))$$

as the typical problem. In general the situation has mental components, e.g. knowledge, as well as the physical components. Thus the some of the actions may be mental actions, e.g. observing or inferring, affecting the mental

component of the situation. Therefore, we need to plan mental actions, probably using a mental situation calculus or situation calculus with mental components.

Mental state includes beliefs and other *intentional entities*.

14.1 Creativity

Making computers creative is a part of achieving advanced human-level AI. I say "advanced", because no-one is creative most of the time, and some quite effective people seem not to be creative at all. One of the most commonly held intuitions that AI is impossible comes from a feeling that machines are intrinsically uncreative.

What is creativity?

I begin by restricting the inquiry to intellectual creativity, because I don't have enough intuition for artistic creativity to form an opinion about it. Maybe it involves the same mechanisms, and maybe it doesn't.

What is intellectual creativity?

We don't give credit for creativity to someone who starts from a previously known goal and works backwards by means-ends analysis to a sequence of steps that achieves the goal. Note that selecting a particular goal may involve creativity.

Creativity requires coming up with a new idea—one that is apparently not apparent in the statement of the problem. I say "apparently not apparent", because we can imagine that some apparently unapparent ideas are really apparent from a routine act of putting the problem into a somewhat larger context. I know this is a bit vague, but I hope that the examples to be given will make it clearer.

Creativity is a hard problem, but we can chip off a part of it.

We ask:

What is a creative solution to a problem?

Now we are talking about a creative solution and no longer about a creative person or program.

A creative solution involves introducing entities not present in the language required to state the problem.

Example: the mutilated checkerboard problem.

Remark:

Different human intellectual capabilities involve different mechanisms. Presumably the highest capabilities involve them all.1

Maybe there is a distinction between ordinary human level intelligence and high human intelligence. Perhaps ordinary intelligence does not involve significant use of introspection.

Does human level logical AI (hllai) require contexts as objects? On one hand, people switch contexts or use the appropriate context without such machinery. On the other hand, people speak of contexts as objects when convenient. Perhaps contexts as objects are only a special case of reification, i.e. all sorts of entities are treated as objects when convenient. What sorts of entities?

Does hllai require concepts as objects?

Do any other animals than humans use goal stacks in the solution of particular problems? In order to do A I must do B first; in order to B I must do C first; in order to do C I must do D first.

What's the role of reification? What is this thing called love?

Chapter 15

Miscellaneous

Most of the stuff here will go in the book somewhere. I'm keeping it in the text, so it will appear in my proof copies. If I give you a copy, you may read this chapter, but don't expect it to make sense.

15.1 Embedding science in situation calculus

Example: The operation of an automobile engine can be describe in situation calculus. Indeed the informal description in a high school or college course discusses sequences of events and their causal relations. However, the description doesn't take the form of a sitcalc problem solver where there is a goal to be achieved. One is not interested in a specific initial situation $S0$ and a goal situation. The interest is in describing the causal relation of an arbitrary process in a certain class of processes, in way that permits this process to operate concurrently with other processes.

Axiomatizing the IC engine in sitcalc should have the following features.

15.2 Generality

A situation calculus formalism that can express what people know about situations, narratives, general laws and events must have at least the following capabilities.

Big concurrency

15.3 projects

1. What is the simplest system that can genuinely be said to exhibit understanding of some class of phenomena?

2. What is the simplest case of learning about hidden entities, about structured entities?

15.4 Polemics

In this chapter we present arguments for some statements made rather baldly earlier and argue with other points at view. At present the section consists of squibs that are later to be developed systematically or deleted.

Here are some assumptions often made, usually without explicit notice of their limitations.

concept as a boolean combination An old book (Jerome S. Bruner 1956) defines a concept as a boolean combination of elementary concepts and interprets experiments in which subjects learned such concepts. Since most human concepts do not have this form, this theory was a Procrustean bed. Unfortunately, most machine learning research described in (Mitchell 1997) makes the same assumption.

15.4.1 Remarks on Psychology

People are very bad in observing their own reasoning, even when the premises and results of the reasoning are directly observable. Every chess player uses the alpha-beta heuristic in his own play. Yet the early designers of chess programs didn't notice this fact. Recent discoveries about how to handle domain constraints also correspond to observable, but often not observed, mental processes.

People may not be quite as bad logicians as the experiments in which people accept fallacies indicate. When a person makes a logical mistake, he may nevertheless accept correction from another person who points out the mistake. There need to be experiments about the extent to which people

who make logical mistakes are corrigible with regard to the mistake made on the specific occasion. Perhaps people are generally corrigible, and perhaps they will resist attempts at correction. If people are corrigible, it shows that in some sense they do know logic.

Moreover, much of the reasoning people do is nonmonotonic. They will even do their customary nonmonotonic reasoning when more careful thought would tell them that the question they are asked has a definite answer. Someone should study human nonmonotonic reasoning experimentally.

Can a system learn or even be told about a lemon as a natural kind having many properties beyond those used to identify it.

Arthur R. Jensen (Jensen 1998), a leading researcher in human intelligence, suggests "as a heuristic hypothesis" that all normal humans have the same intellectual mechanisms and that differences in intelligence are related to "quantitative biochemical and physiological conditions". I see them as speed, short term memory, and the ability to form accurate and retrievable long term memories.

Whether or not Jensen is right about human intelligence, the situation in AI is the reverse.

Computer programs have plenty of speed and memory but their abilities correspond to the intellectual mechanisms that program designers understand well enough to put in programs. Some abilities that children normally don't develop till they are teenagers may be in, and some abilities possessed by two year olds are still out. The matter is further complicated by the fact that the cognitive sciences still have not succeeded in determining exactly what the human abilities are. Very likely the organization of the intellectual mechanisms for AI can usefully be different from that in people.

Whenever people do better than computers on some task or computers use a lot of computation to do as well as people, this demonstrates that the program designers lack understanding of the intellectual mechanisms required to do the task.

Will an artificial system of human-level intelligence report qualia? I think it would report something similar, e.g. an abstraction of seeing red apart from seeing a red object, but I can't prove it.

Q. Are you claiming that logical AI corresponds to how people think?

A. No. Mathematical logic doesn't correspond well to the reasoning people actually do. It is rather a corrected version of what people do. It corresponds better to what people will accept after argument.

Logical AI is intended an improved version of what people do, taking

into account the need for nonmonotonic reasoning, reasoning within ccntexts and relating contexts, and reasoning with approximate entities. However, we don't yet know how people reason, i.e we don't yet know what we might be hoping to improve.

15.5 What AI can get from philosophy

Philosophers have done a lot of research in trying understand many ideas. A little of this work is immediately relevant to AI. More will be relevant when AI advances further. However, a lot of the research is unlikely to be useful, because AI systems will solve the problems in a different way.

We illustrate this using the idea of definite descriptions.

15.5.1 Definite descriptions

Definite descriptions were discussed by Frege, but the logical notation is due to Russell.

$(\iota x)\phi(x)$ denotes the unique x satisfying $\phi(x)$. In English, it may be read, "The x such that $\phi(x)$. [1]

Robots will need to use definite descriptions, and not just in communications with humans and with other robots.

Robots can use definite descriptions even when they are not intended for communication and reside in its private knowledge base. Suppose it decides, "I'll go climb the tallest tree in the grove, but to get to the grove, it will be necessary to go out of sight of the tree."

There is an extensive philosophical literature about definite descriptions. The philosophers pretty much agree about the interpretation of $(\iota x)\phi(x)$ when there is a unique x satisfying ϕ, although there are some odd cases.

Here are some of the philosophical examples. They all refer to communication.

[1] The literature I have seen about definite descriptions is based on English usage of the particle "the". However, the Russian language, like Latin, does not have a definite article. Normally whether a description refers to a definite object is determined by context. However, circumlocutions are used when the context is inadequate.

Vladimir Lifschitz gave me example of elaboration of Russian sentence to make a description definite. "Kniga lezhit na tom stolye, kotorii stoyit v uglu." for "The book is on the table in the corner." The Russian amounts to "The book is on that table, which is in the corner." The definiteness of "corner" still comes from context.

1. Keith Donellan distinguishes between attributive and referential uses of definite descriptions. The example is "The murderer must be insane" uttered on two kinds of occasions. The first is that on learning of the murder, the speaker uses the sentence to assert that, whoever the murderer is, he must be insane. The second is that on seeing Jones acting up in court and thinking Jones to be the murderer, the description "the murderer" is used to refer to the person before him. The speaker is making an assertion about the person before him that he would still avow even if it turned out that this person were not the murderer.

2. Saul Kripke uses "The man over there drinking champagne is happy", and the hearer agrees. Unbeknownst to them, the man they see is not drinking champagne but pure water, and another man over there whom they don't see is drinking champagne and is unhappy. The two have agreed about the man they see and misdescribe. The philosophers therefore see the need to distinguish between what the phrase, e.g. "the man over there drinking champagne" means and what the speakers mean by it.

The above comments are not intended to do justice to 145 pages of philosophy reprinted in (Garfield and Kiteley 1991)—only to illustrate a few of the phenomena that have been studied.

Which of them are important for AI—now and in the future?

Donellan's distinction may come up in the activity of a logical robot, but it should not be built into the formalism. Instead the mechanisms for dealing with ambiguity tolerance via approximate entities should be able to handle the distinction on an *ad hoc* basis if it comes up. The same consideration applies to Kripke's example.

Robots will mainly need the straightforward use of definite descriptions.

15.6 The Road to Human-Level Logical AI

Reaching human-level AI requires solving many problems. The problems are somewhat different for logic oriented approaches and biologically oriented approaches.

The biological approach is straightforward. If only we understood the physiology of the human brain well enough, we might hope to imitate it. There are conceptual difficulties, but the main difficulty is experimental.

Present technology cannot observe brains in action well enough. The technological difficulties may be overcome, and the biological approach to AI may succeed. The conceptual difficulties show up in the inadequacy of AI approaches based on neural nets and connectionist systems. We will discuss them somewhat in Chapter 16.2.

The logical approach is based on understanding the problems the world presents to intelligence and devising reasoning methods that can solve the problems at least as well as people do.

We propose to use the following tools.

representation of facts by logical formulas

logical deduction

nonmonotonic logical inference

theories including approximate entities

contexts as objects

The information situation of a person (or robot) in the world is more complex than just believing a theory in a logical language in at least the following respects.

At any moment, thinking, observing and acting is within a certain limited context. What is immediately meaningful is determined by that context. For logical AI this requires associating a language with a context. Within a context, other contexts may be referred to, and logical AI can do this by regarding contexts as objects. We use the predicate $Ist(c, p)$ as explained in chapter **??**. We also use operations of *entering* and *exiting* contexts to switch contexts.

Leibniz, Boole, Frege and Peirce all expected the symbolic logic they were developing to be applicable to the *common sense informatic situation*[2]. For example, Leibniz hoped to replace disputation by calculation. Unfortunately, deductive mathematical logic is inadequate for reasoning in the CSIS, and psychologist and others have had little difficulty in showing that much human reasoning is not in accordance with deduction.[3]

[2]Of course, they didn't use the term, because they didn't make the distinction between that and the situation of mathematical reasoning.

[3]Often they exaggerate the difference; much human reasoning is deductive and mathematical logic represents a formalization of that part of human reasoning.

One reaction to the difficulties is despair and the opinion that something entirely different is required. There have been many candidates for "something entirely different", but they haven't been even as successful as logic.

There are good reasons not to give up. Deductive logic, as pioneered by Aristotle and polished off by Gödel's completeness theorem, has told us what constitutes a rigorous argument in which the conclusions are guaranteed when the premises are true. That is a valuable achievement, and we can try to extend it.

The use of deductive logic has had plenty of success in AI and in computer science more generally. xxx details this. However, two kinds of limitations have appeared—expressive limitations and computational limitations.

There are two reactions to the limitations.

The most common reaction to the expressive limitations is to live with them and work on problems where they don't show themselves. This allows many applications, even if it won't lead to human-level AI.

The computational limitations are taken more seriously. There is a substantial theory of computational complexity. I think this theory is misleading.

The theory of computational complexity and the theory of computability are misleading when applied to AI.

15.6.1 Hard problems of AI

Three dimensional objects

Natural languages seem to lack names for textures that are analogous to the names for colors. They might be useful for robots and maybe even for people. Suppose we want to tell a person or robot how to pick an object of a kind unknown to that agent out of a pocket. If we had names for textures we could tell him the texture of various surfaces. Thus the texture of a surface of an object could be observed by one person and its name giveqq, and another person could use the name to find the designated surface.

15.7 Essays in AI

The main part of this book presents formalisms intended for use by reasoning programs. This chapter presents considerations relevant to human-level AI that are not presently in shape to be fully formalized. It would be nice if this chapter were considerably shorter.

15.7.1 Notes on the evolution of intelligence

Primitive animals react rather directly to stimuli in the environment combined with internal stimuli, e.g. whether the animal is hungry. For example, a frog sticks out its tongue to catch small black objects, which might be flies. A person swatting flies will distinguish them from other small dark objects. (I don't know if a frog put in an environment with both flies and small bad tasting black disks would learn to distinguish them. Maybe flies already know how to distinguish flies from wasps and poisonous spiders. Some of that could have been learned by evolution.)

However, the frog's S-R response to flies evolved in a world in which flies are three-dimensional objects. Therefore it may be more sophisticated than the learning theorists' 2-d world.

15.7.2 Remarks

Much research, some of it quite good, might be banished from AI proper. Here's the criterion for banishing a topic. Human-level AI can be reached without the topic, and a human-level AI program could then read a 1990s book about the topic and use it where applicable. An example, which only a few would include in AI, is linear programming. A human-level program could read a book about it and write programs to optimize the product mix of oil refineries.

Most work on algorithms for machine learning is likely to fare similarly. This is a snide remark that needs to be justified or omitted.

The step from chimpanzee intelligence to human intelligence is probably a short one. Candidates are speech, episodic memory and the ability to form concepts using episodic memory. The question is important for AI; at least it would help if research in chimpanzee intelligence found out more about the difference. Maybe also goal stack.

Why was GPS (General Problem Solver) inadequate?

The AI industry: expert systems.

15.7.3 Philosophical remarks

While some philosophical attitudes seem important for the development of AI, AI research suggests some philosophical ideas not directly required for AI science.

In the past, science has substantially influenced philosophy. Here's another shot at it. Suppose we believe that human intelligence is a product of evolution, and perhaps we are interested in a theory of the evolution of intelligence within causal systems, i.e. some subsystems develop intelligence. It is unlikely to be a theorem of this theory that all aspects of the world, i.e. the system as a whole, are definable or even discoverable in terms of the experience of the intelligent subsystem. Some aspects of the world may be undiscoverable.

We regard this scientific or mathematical proposition as a refutation of the philosophical proposition (or dogma) that a statement about the world is meaningful only if it is potentially testable.

15.8 Notes on logic chapter, in response to Pat Hayes

I am not decided which of them to put in the chapter. Some of it probably should go elsewhere in the book.

A logical robot decides what to do by reasoning logically that a certain action will advance its goals. //As I think Ive said elsewhere, theres a central snag in this idea, which is that reasoning is itself doing something. So if we take this statement at face value and literally, then the logical robot can't ever get started, since it first has to decide what reasoning to do...//

This requires a collection of sentences from which what to do can be inferred and a program to do the inferences. We must decide what part of the intelligence to put in the sentences and what part to put in the program.

There are two extreme cases

- Put everything in the program. Many AI systems have this character.
- Begin with a minimal program, and put everything, including a description of the program, in the sentences.

The bias of this book is toward beginning with a minimal program. The initial program has three capabilities.

- It forward chains from sentences in its database.

- When it infers a sentences saying that it should perform an action, it does the action.
- Some of the actions it can perform add to or modify the program.

Besides its knowledge of outside world, the knowledge base includes the following.

- A description of the program.
- Facts enabling reasoning about what the program will do in some class of situations. This must include a description of the programming language.
- Facts about modifying the program.

Advantages of Imperative Sentences

1. A procedure described in imperatives is already laid out and is carried out faster.
2. One starts with a machine in a basic state and does not assume previous knowledge on the part of the machine.

Advantages of Declarative Sentences

1. Advantage can be taken of previous knowledge.
2. Declarative sentences have logical consequences and it can be arranged that the machine will have available sufficiently simple logical consequences of what it is told and what it previously knew.
3. The meaning of declaratives is much less dependent on their order than is the case with imperatives. This makes it easier to have afterthoughts.
4. The effect of a declarative is less dependent on the previous state of the system so that less knowledge of this state is required on the part of the instructor.

15.9 Temporary Notes

These are just stuff that will go in somewhere.

Topics: introduction, philosophical premises, logic and set theory, nonmonotonic reasoning, situation calculus, elaboration tolerance, approximate objects, control of reasoning, connection to lower level processes, contexts, concepts as objects, counterfactual reasoning, reasoning about space, vision and hearing, speech recognition, recommendations for AI research, AI as science and AI as engineering, concepts of logical AI fits in somewhere, opportunities for society, goals of robots, plan to reach human-level logical AI, computability and computational complexity, criticisms and off-hand opinions, appearance and reality, ontology, creativity and creative ideas, squibs, unsolved problems, logic level

Human universals vs. world universals

Chapters: intro, mechanisms and structures of intelligence, logic (including nmr, contexts, approximate objects, reified concepts, counterfactuals, elaboration tolerance), basic situation calculus, heuristics (including planning), robots in the world, mental qualities, essays (philosophical basis, learning, pruning AI)

The causality cone, maybe inferential cone, so as to be logical and not just temporal. Backwards and forwards. Conceptually it is like the light cone, but what is outside it is inferred by default. The default is that events in one story do not affect events in another. Maybe it isn't a precise analogy, because the default is for each event in the other place. Even when we do have Daddy and Junior interacting, Junior's purchase of a new ticket doesn't affect Daddy's blocks.

Other work: Lifschitz, Reiter, Levesque, Lin, Shanahan, Costello, Sierra, Poole, Baker, Shoham, Etherington, Moore, Konolige, Saša Buvač Denecker et al.

Hypotheses:

One of those sheep out there is a wolf in sheep's clothing. That explains the loss of a sheep, but I don't presently know how to tell which one is the wolf.

Consider the hypothesis that one of those sheep is not what it seems to be.

Any kind of spying involves going from appearance to phenomena.

1. estimating quantity from a few observed serial numbers.
2. inferences about units from aerial photos of tank tracks

The evidence from the test suggests that the student does not understand about disguised quadratic equations.

There were three chairs at the table and three bowls of porridge.

Other people's likes and dislikes may be inferred concepts—although they may be innate.

Tue Aug 4 11:40:25 1998 What generality is possible? Consider

1. Unsupported objects fall.

2. When an object moves it is no longer where it was previously. On the other hand information can be transferred without the source forgetting it.

3. Objects placed on a flat horizontal surface stay there.

Locations of objects and occupants of places are dual notions. A good formalism will have both, accepting the fact that when one is changed, the other has to be updated.

/u/jmc/f94/reactive.tex The Garden Path from Reaction to Planning

Loves(A,B), the love of A for B is greater than the love of C for D. for every boy there is a girl who loves only him.

$$(\forall x)(Boy(x) \rightarrow (\exists y)(Girl(y) \wedge Loves(y,x) \wedge (\forall z)(Loves(y,z) \rightarrow z = x))) \quad (15.1)$$

$$(\forall boy)(\exists girl)(Loves\text{-}Only(girl, boy))) \quad (15.2)$$

If Pat knows Mike's telephone number he can dial it. If Pat asks, "What is Mike's telephone number?" and is answered responsively, he will then know the number.

Chapter 16

POLEMICS

16.1 Opponents of AI

16.1.1 AI is impossible in principle

The too simple counterexample

Quine's doctrine of "the indeterminacy of radical translation" relies on overly simple counterexamples.

Consider the cryptogram XYZ in a simple substitution cipher. XYZ could stand for "dog" and equally well for "cat". However, when a simple subsitution cryptogram passes 21 letters in length, its interpretation will almost always be unique. In cryptography 21 letters is the *unicity distance* of English language simple substitution ciphers.

In an article in *Scientiic American*, John Searle writes that the dialog of the "Chinese room" could equally well be the score of a chess game or an article predicting the stock market. This is mistaken for dialogs of reasonable length; 100 characters should suffice for Chinese dialogs, chess scores or articles about the stock market. A rule for translating Chinese dialogs into either of these would be long and would consume inordinate computer time. I don't have a precise statement of a conjectured theorem.

If we order the Chinese dialogs and the chess scores, we can make a one-one-correspondence. However, both languages have compositional structure, so extremely probably there is no 1-1 correspondence that preserves structure. Maybe even the Markov structures cannot be matched.

What is the simplest task you think computers cannot be programmed

to do?

16.1.2 Opponents of logical AI

16.1.3 Those who think computer speed will do it

Moravec, Kurzweil
/u/jmc/w99/miscellaneous.tex=476

16.2 Polemics

In this chapter we present arguments for some statements made rather baldly earlier and argue with other points at view. At present the section consists of squibs that are later to be developed systematically or deleted.

Here are some assumptions often made, usually without explicit notice of their limitations.

Inadequate paradigms for AI

Each of the following paradigms dominating fields of AI research must be transcended in order to reach human level AI.

concept as a boolean combination An old book (Jerome S. Bruner 1956) defines a concept as a boolean combination of elementary concepts and interprets experiments in which subjects learned such concepts. Since most human concepts do not have this form, this theory was a Procrustean bed. Unfortunately, most machine learning research described in (Mitchell 1997) makes the same assumption.

present concepts of machine learning

unary ontology It is usual in AI discussions of ontology to only use unary predicates as elements in the ontology. Relations among the elements are discussed, but relations themselves are not elements. Thus *child* and *parent* are elements, but the relation $parent(x, y)$ is not an element. Philsophers, starting with Aristotle but must emphatically Leibniz, make the same mistake. (**?**) discusses Leibniz's mistake in denying relations.

evolution of trivial systems

theorem provers not allowing domain dependent heuristics

limited notions of planning

Markov models

make it like physics

it must model neurology

Each bad paradigm needs an exemplifying quotation. Maybe each should be a separate article if the arguments are to have maximum effect. Each also needs a problem it won't solve that is in the domain the paradigm is advertised to solve. The killer would be the simplest example of a program solving one of the problems beyond the paradigm.

These paradigms are really mini-paradigms, i.e. not as grand scientific sociological phenomena as Thomas Kuhn postulated. Nevertheless, a paradigm with hundreds of papers, several journals and many books in which the paradigm is explicitly presumed to define the problem of e.g. learning or evolution or planning, is worthy of calling at least a mini-paradigm and worth criticizing from the outside.

Arguments against AI

meat machine

argument against AI from Gödel's theorem

argument from practice, Dreyfusses

The immorality of AI is occasionally argued, e.g. by Weizenbaum and by Joy. However, we haven't suffered from the intensity of ignorant and fanatical attacks like those levelled at nuclear energy and genetic engineering. Therefore, this book ignores this kind of criticism in the hope that it won't amount to much until human level AI is a lot closer.

Working index

learning /u/jmc/w99/miscellaneous.tex=2603 /u/jmc/w99/miscellaneous.tex=14517 would a fly learn

bad paradigms /u/jmc/w99/polemics.tex=781

Bibliography

Amarel, S. 1971. On representation of problems of reasoning about action. In D. Michie (Ed.), *Machine Intelligence 3*, 131–171. Edinburgh University Press.

Bratko, I., and S. Muggleton. 1995. Applications of inductive logic programming. *Communications of the ACM* 38(11):65–70.

Brewka, G. 1991. *Nonmonotonic Reasoning: Logical Foundations of Common Sense*. Cambridge University Press.

Buvač, S. 1995a. Saša Buvač's Web page[1].

Buvač, S. 1995b. Quantificational logic of context. In *Proceedings of the Workshop on Modeling Context in Knowledge Representation and Reasoning (held at the Thirteenth International Joint Conference on Artificial Intelligence)*.

Buvač, S., V. Buvač, and I. A. Mason. 1995. Metamathematics of contexts. *Fundamenta Informaticae* 23(3).

Carnap, R. 1956. *Meaning and Necessity*. University of Chicago Press.

Costello, T., and J. McCarthy. 1998. Useful Counterfactuals and Approximate Theories. In *AAAI Spring Symposium on Prospects for a Commonsense theory of Causation*. AAAI Press. A longer version will not appear in Proc. National Conference on Artificial Intelligence (AAAI '98).

Costello, T., and J. McCarthy. 1999. Useful Counterfactuals[2]. *Electronic Transactions on Artificial Intelligence*.

[1] http://www-formal.stanford.edu/buvac/

[2] http://www-formal.stanford.edu/jmc/counterfactuals.html

Crick, F. 1995. *The Astonishing Hypothesis: The Scientific Search for Soul*. Scribners.

Davis, R., B. Buchanan, and E. Shortliffe. 1977. Production rules as a representation for a knowledge-based consultation program. *Artificial Intelligence* 8(1):15–45.

Dennett, D. 1978. *Brainstorms: Philosophical Essays on Mind and Psychology*. Cambridge: Bradford Books/MIT Press.

Dennett, D. 1991. *Consciousness Explained*. Boston: Little, Brown and Co.

Dennett, D. 1998. *Brainchildren: Essays on Designing Minds*. MIT Press.

Dennett, D. C. 1971. Intentional systems. *The Journal of Philosophy* 68(4):87–106.

Dreyfus, H. 1992. *What Computers still can't Do*. M.I.T. Press.

Feferman, S. 1962. Transfinite recursive progressions of axiomatic theories. *J. Symbolic Logic* 27:259–316.

Flavell, J. H., and A. K. O'Donnell. 1999. Development of intuitions about mental experiences. *Enfance*. in press.

Frege, G. 1892. Uber sinn und bedeutung. *Zeitschrift für Philosophie und Philosophische Kritik* 100:25–50. Translated by H. Feigl under the title "On Sense and Nominatum" in H. Feigl and W. Sellars (eds.) *Readings in Philosophical Analysis*. New York 1949. Translated by M. Black under the title "On Sense and Reference" in P. Geach and M. Black, *Translations from the Philosophical Writings of Gottlob Frege*. Oxford, 1952.

Garfield, J. L., and M. Kiteley (Eds.). 1991. *Meaning and Truth: The Essential Readings in Modern Semantics*. Paragon Issues in Philosophy.

Gelfond, M., V. Lifschitz, and A. Rabinov. 1991. What are the limitations of the situation calculus? In R. Boyer (Ed.), *Automated Reasoning: Essays in Honor of Woody Bledsoe*, 167–179. Dordrecht: Kluwer Academic.

Gödel, K. 1940. *The Consistency of The Axiom of Choice and of the Generalized Continuum-Hypothesis with the Axioms of Set Theory*. Princeton University Press.

Gödel, K. 1965. On undecidable propositions of formal mathematical systems. In M. Davis (Ed.), *The Undecidable*. Raven Press. This is the famous 1931 paper.

Green, C. 1969a. Applications of theorem proving to problem solving. In *Proceedings IJCAI 69*, 219–240.

Green, C. 1969b. Theorem-proving by resolution as a basis for question-answering systems. In B. Meltzer, D. Michie, and M. Swann (Eds.), *Machine Intelligence 4*, 183–205. Edinburgh, Scotland: Edinburgh University Press.

Grice, P. 1989. *Studies in the Way of Words*. Harvard University Press.

H.-D. Ebbinghaus, H. Hermes, F. H. M. K. K. M. J. N. A. P. R. R. 1991. *Numbers*. Springer.

Hanks, S., and D. McDermott. 1986. Default reasoning, nonmonotonic logics and frame problem. In *Proceedings of AAAI-86*, 328–333. Morgan Kaufmann.

Hayes, P. J. 1985. The second naive physics manifesto. In H. J.R. and M. R.C. (Eds.), *Formal Theories of the Commonsense World*, 1–36. Ablex.

Jensen, A. R. 1998. Does IQ matter? *Commentary* 20–21. The reference is just to Jensen's comment—one of many.

Jerome S. Bruner, Jacqueline J. Goodnow, G. A. A. 1956. *A Study of Thinking*. Wiley.

John H. Flavell, F. L. G., and E. R. Flavell. 2000. Development of children's awareness of their own thoughts. *Journal of Cognition and Development* 1:97–112.

Kaplan, D. 1969. Quantifying in. In D. Davidson and J. Hintikka (Eds.), *Words and Objections: Essays on the Work of W.V. Quine*, 178–214. Dordrecht-Holland: D. Reidel Publishing Co. Reprinted in (Linsky 1971).

Kaplan, D., and R. Montague. 1960. A paradox regained. *Notre Dame Journal of Formal Logic* 1:79–90. Reprinted in (Montague 1974).

Kraus, S., D. Perlis, and J. Horty. 1991. Reasoning about ignorance: A note on the Bush-Gorbachev problem. *Fundamenta Informatica* XV:325–332.

Kuipers, B. 1994. *Qualitative Reasoning*. MIT Press.

Lenat, D. B., and R. V. Guha. 1990. *Building Large Knowledge-Based Systems: Representation and Inference in the CYC Project*. Addison-Wesley.

Levesque, H. J., R. Reiter, I. Lesprance, F. Lin, and R. B. Scherl. 1997. Golog: A logic programming language for dynamic domains. *Journal of Logic Programming* 31(1–3):59–83.

Lewis, D. 1973. *Counterfactuals*. Harvard University Press.

Lifschitz, V. 1994. Circumscription. In *Handbook of Logic in Artificial Intelligence and Logic Programming, Volume 3: Nonmonotonic Reasoning and Uncertain Reasoning*. Oxford University Press.

Linsky, L. (Ed.). 1971. *Reference and Modality*. Oxford University Press.

Maida, A. S., and S. C. Shapiro. 1982. Intensional concepts in propositional semantic networks. *Cognitive Science* 6(4):291–330. Reprinted in R. J. Brachman and H. J. Levesque, eds. Readings in Knowledge Representation, Morgan Kaufmann, Los Altos, CA, 1985, 170-189.

McAllester, D. n.d. Some Observations on Cognitive Judgements[3], booktitle = "aaai-91", publisher = "morgan kaufmann publishers", month = jul, year = "1991", pages = "910–915",.

McCarthy, J. 1959. Programs with Common Sense[4]. In *Mechanisation of Thought Processes, Proceedings of the Symposium of the National Physics Laboratory*, 77–84, London, U.K. Her Majesty's Stationery Office. Reprinted in (McCarthy 1990).

[3]http://www.research.att.com/ dmac/aaai91a.ps
[4]http://www-formal.stanford.edu/jmc/mcc59.html

McCarthy, J. 1962a. Towards a mathematical science of computation. In *Information Processing '62*, 21–28. North-Holland. Proceedings of 1962 IFIP Congress.

McCarthy, J. 1962b. Towards a mathematical science of computation. In *Information Processing '62*, 21–28. North-Holland. Proceedings of 1962 IFIP Congress.

McCarthy, J. 1962c. Towards a mathematical science of computation. In *Information Processing '62*, 21–28. North-Holland. Proceedings of 1962 IFIP Congress.

McCarthy, J. 1963. A Basis for a Mathematical Theory of Computation[5]. In P. Braffort and D. Hirschberg (Eds.), *Computer Programming and Formal Systems*, 33–70. Amsterdam: North-Holland.

McCarthy, J. 1964. **a tough nut for theorem provers**[6]. Stanford AI Memo 16—now on the web.

McCarthy, J. 1976. Epistemological Problems in Artificial Intelligence[7]. reprinted in (McCarthy 1990).

McCarthy, J. 1979a. First order theories of individual concepts and propositions. In D. Michie (Ed.), *Machine Intelligence*, Vol. 9. Edinburgh: Edinburgh University Press. Reprinted in (McCarthy 1990).

McCarthy, J. 1979b. Ascribing mental qualities to machines[8]. In M. Ringle (Ed.), *Philosophical Perspectives in Artificial Intelligence*. Harvester Press. Reprinted in (McCarthy 1990).

McCarthy, J. 1979c. First Order Theories of Individual Concepts and Propositions[9]. In D. Michie (Ed.), *Machine Intelligence*, Vol. 9. Edinburgh: Edinburgh University Press. Reprinted in (McCarthy 1990).

McCarthy, J. 1980. Circumscription—A Form of Non-Monotonic Reasoning[10]. *Artificial Intelligence* 13:27–39. Reprinted in (McCarthy 1990).

[5] http://www-formal.stanford.edu/jmc/basis.html
[6] http://www-formal.stanford.edu/jmc/nut.html
[7] http://www-formal.stanford.edu/jmc/epistemological.html
[8] http://www-formal.stanford.edu/jmc/ascribing.html
[9] http://www-formal.stanford.edu/jmc/concepts.html
[10] http://www-formal.stanford.edu/jmc/circumscription.html

McCarthy, J. 1983. Some Expert Systems Need Common Sense[11]. In H. Pagels (Ed.), *Computer Culture: The Scientific, Intellectual and Social Impact of the Computer*, Vol. 426. Annals of the New York Academy of Sciences.

McCarthy, J. 1986. Applications of Circumscription to Formalizing Common Sense Knowledge[12]. *Artificial Intelligence* 28:89–116. Reprinted in (McCarthy 1990).

McCarthy, J. 1987. Generality in artificial intelligence. *Communications of the Association for Computing Machinery* 30:1030–1035. Reprinted in (McCarthy 1990).

McCarthy, J. 1988. Mathematical logic in artificial intelligence. *Daedalus* 117(1):297–311.

McCarthy, J. 1989. Artificial Intelligence, Logic and Formalizing Common Sense[13]. In R. Thomason (Ed.), *Philosophical Logic and Artificial Intelligence*. Klüver Academic.

McCarthy, J. 1990. *Formalizing Common Sense: Papers by John McCarthy*. Ablex Publishing Corporation.

McCarthy, J. 1993. Notes on Formalizing Context[14]. In *IJCAI-93*.

McCarthy, J. 1995a. Partial Formalizations and the Lemmings Game[15]. Technical report, Stanford University, Formal Reasoning Group.

McCarthy, J. 1995b. Situation Calculus with Concurrent Events and Narrative[16]. Web only, partly superseded by (McCarthy and Costello 1998).

McCarthy, J. 1996a. **elephant 2000**[17]. Technical report, Stanford Formal Reasoning Group. Available only as http://www-formal.stanford.edu/jmc/elephant.html.

[11] http://www-formal.stanford.edu/jmc/someneed.html
[12] http://www-formal.stanford.edu/jmc/applications.html
[13] http://www-formal.stanford.edu/jmc/ailogic.html
[14] http://www-formal.stanford.edu/jmc/context.html
[15] http://www-formal.stanford.edu/jmc/lemmings.html
[16] http://www-formal.stanford.edu/jmc/narrative.html
[17] http://www-formal.stanford.edu/jmc/elephant.html

McCarthy, J. 1996b. From Here to Human-Level AI[18]. In *KR96 Proceedings*. Available as http://www-formal.stanford.edu/jmc/human.html.

McCarthy, J. 1996c. Making Robots Conscious of their Mental States[19]. In S. Muggleton (Ed.), *Machine Intelligence 15*. Oxford University Press. Appeared in 2000. The web version is improved from that presented at Machine Intelligence 15 in 1995.

McCarthy, J. 1996d. **the mutilated checkerboard in set theory**[20]. presented at a 1996 conference in Warsaw.

McCarthy, J. 1997. Elaboration Tolerance[21]. In *McCarthy's web page*.

McCarthy, J. 1999a. **appearance and reality**[22]. *web only for now, and perhaps for the future*. not fully publishable on paper, because it contains an essential imbedded applet.

McCarthy, J. 1999b. **creative solutions to problems**[23]. *web only for now*. given at AISB Workshop on AI and Scientific Creativity, Edinburgh, 1999 April.

McCarthy, J. 1999c. Elaboration tolerance[24]. *to appear*.

McCarthy, J. 1999d. Logical theories with approximate objects[25]. *superseded by (McCarthy 2000)*.

McCarthy, J. 1999e. Parameterizing models of propositional calculus formulas[26]. *web only for now*.

McCarthy, J. 2000. Approximate objects and approximate theories[27]. In A. G. Cohn, F. Giunchiglia, and B. Selman (Eds.), *KR2000: Principles*

[18]http://www-formal.stanford.edu/jmc/human.html
[19]http://www-formal.stanford.edu/jmc/consciousness.html
[20]http://www-formal.stanford.edu/jmc/checkerboard.html
[21]http://www-formal.stanford.edu/jmc/elaboration.html
[22]http://www-formal.stanford.edu/jmc/appearance.html
[23]http://www-formal.stanford.edu/jmc/creative.html
[24]http://www-formal.stanford.edu/jmc/elaboration.html
[25]http://www-formal.stanford.edu/jmc/approximate.html
[26]http://www-formal.stanford.edu/jmc/parameterize.html
[27]http://www.formal.stanford.edu/jmc/approximate.html

of Knowledge Representation and Reasoning,Proceedings of the Seventh International conference, 519–526. Morgan-Kaufman.

McCarthy, J., and S. Buvač. 1997. Formalizing context (expanded notes). In A. Aliseda, R. v. Glabbeek, and D. Westerståhl (Eds.), *Computing Natural Language*. Center for the Study of Language and Information, Stanford University.

McCarthy, J., and S. Buvač. 1998. Formalizing Context (Expanded Notes). In A. Aliseda, R. v. Glabbeek, and D. Westerståhl (Eds.), *Computing Natural Language*, Vol. 81 of *CSLI Lecture Notes*, 13–50. Center for the Study of Language and Information, Stanford University.

McCarthy, J., and T. Costello. 1998. Combining narratives. In *Proceedings of Sixth Intl. Conference on Principles of Knowledge Representation and Reasoning*, 48–59. Morgan-Kaufman.

McCarthy, J., and P. J. Hayes. 1969a. Some Philosophical Problems from the Standpoint of Artificial Intelligence[28]. In B. Meltzer and D. Michie (Eds.), *Machine Intelligence 4*, 463–502. Edinburgh University Press. Reprinted in (McCarthy 1990).

McCarthy, J., and P. J. Hayes. 1969b. Some Philosophical Problems from the Standpoint of Artificial Intelligence[29]. In B. Meltzer and D. Michie (Eds.), *Machine Intelligence 4*, 463–502. Edinburgh University Press. Reprinted in (McCarthy 1990).

Miller, R. S. 1996. A case study in reasoning about actions and continuous change. In *Proceedings ECAI 96*, 624–628.

Mitchell, T. 1997. *Machine Learning*. McGraw-Hill.

Montague, R. 1963. Syntactical treatments of modality, with corollaries on reflexion principles and finite axiomatizability. *Acta Philosophica Fennica* 16:153–167. Reprinted in (Montague 1974).

Montague, R. 1974. *Formal Philosophy*. Yale University Press.

[28] http://www-formal.stanford.edu/jmc/mcchay69.html
[29] http://www-formal.stanford.edu/jmc/mcchay69.html

Moore, R. C. 1995. A formal theory of knowledge and action. In *Logic and Representation*, chapter 3. Center for the Study of Language and Information, Stanford, California: CSLI Publications.

Muggleton, S., and L. De Raedt. 1994. Inductive logic programming: Theory and methods. *Journal of Logic Programming* 19,20:629–679.

Nagel, T. 1974. What is it like to be a bat? *Philosophical Review* 83(4):435–50.

Newell, A. 1980. Physical symbol systems. *Cognitive Science* 4:135–183.

Newell, A. 1982. The knowledge level. *AI* 18(1):87–127.

Newell, A. 1993. Reflections on the knowledge level. *Artificial Intelligence* 59(1-2):31–38.

Nilsson, N. J. 1984. Shakey the robot, sri technical note no. 323. Technical report, SRI International, Menlo Park, California.

Penrose, R. 1994. *Shadows of the Mind: A Search for the Missing Science of Consciousness*. Oxford: Oxford University Press.

Pinker, S. 1997. *How the Mind Works*. Norton.

Pinto, J., and R. Reiter. 1993. Temporal reasoning in logic programming: A case for the situation calculus. In *Proceedings of the Tenth International Conference on Logic Programming*, 203–221.

Putnam, H. 1975. The meaning of "meaning". In K. Gunderson (Ed.), *Language, Mind and Knowledge*, Vol. VII of *Minnesota Studies in the Philosophy of Science*, 131–193. University of Minnesota Press.

Quine, W. V. O. 1969. Propositional objects. In *Ontological Relativity and other Essays*. Columbia University Press, New York.

Quine, W. 1956. Quantifiers and propositional attitudes. *Journal of Philosophy* 53. Reprinted in (Linsky 1971).

Quine, W. 1961. *From a Logical Point of View*. Harper and Row.

Reiter, R. 1993. Proving properties of states in the situation calculus. *Artificial Intelligence* 64:337–351. available from http://www.cs.utoronto.ca/ cogrobo.

Reiter, R. 1996. Natural actions, concurrency and continuous time in the situation calculus. In *Proceedings KR96*, 2–13. Morgan Kaufmann.

Reiter, R. 1980. A Logic for Default Reasoning[30]. *Artificial Intelligence* 13 (1–2):81–132.

Reiter, R. 2001. *Knowledge in Action*. M.I.T. Press.

R.S.Miller, and M.P.Shanahan. 1994. Narratives in the situation calculus. *Journal of Logic and Computation* 4(5):513–530.

Scherl, R., and H. Levesque. 1993. The frame problem and knowledge producing actions. In *Proceedings AAAI 93*, 689–695.

Searle, J. R. 1984. *Minds, Brains, and Science*. Cambridge, Mass.: Harvard University Press.

Shanahan, M. P. 1996. Robotics and the common sense informatic situation. In *Proceedings ECAI 96*, 684–688.

Shanahan, M. 1997. *Solving the Frame Problem, a mathematical investigation of the common sense law of inertia*. M.I.T. Press.

Shankar, N. 1986. *Proof-Checking Metamathematics*. PhD thesis, Computer Science Department, University of Texas at Austin.

Shoham, Y. 1988. Chronological ignorance: Experiments in nonmonotonic temporal reasoning. *Artificial Intelligence* 36(3):279–331.

Sierra, J. 1998a. *Declarative Formalization of Heuristics*. chapter 11, 1–8.

Sierra, J. 1998b. Declarative formalization of strategies for action selection. In *Seventh International Workshop on Nonmonotonic Reasoning, NM98*, 21–29.

Sierra, J. 1998c. Declarative formalization of strips. In *Thirteenth European Conference on Artificial Intelligence, ECAI-98*, 509–513.

[30]Sent a request to Ray for the paper

Sierra, J. 1999. Declarative formalization of heuristics (taking advice in the blocks world). In *International Conference on Computational Intelligence for Modelling Control and Automation*, 221–228.

Sloman, A. 1985. What enables a machine to understand? In *Proceedings 9th International Joint Conference on AI*, 995–1001. Morgan-Kaufman.

Sloman, A., and M. Croucher. 1981. Why robots will have emotions. In *Proceedings 7th International Joint Conference on AI*. Morgan-Kaufman.

Spelke, E. 1994. Initial knowlege: six suggestions. *Cognition* 50:431–445.

Turing, A. 1950a. Computing machinery and intelligence. *Mind*.

Turing, A. 1950b. Computing machinery and intelligence. *Mind*.

Turing, A. M. 1939. Systems of logic based on ordinals. *Proc Lond Math Soc (2)* 45.

Van Heijenoort, J. (Ed.). 1967. *From Frege to Gödel; a source book in mathematical logic, 1879-1931*. Harvard University Press.

von Mises, R. 1951. *Positivism*. Dover (1968 reprint).

Weld, D. S., and J. de Kleer (Eds.). 1990. *Readings in Qualitative Reasoning about Physical Systems*. Morgan-Kaufmann.

Winston, P. 1970. Learning structural descriptions from examples. In P. Winston (Ed.), *Psychology of Computer Vision*. MIT PRess. based on Winston's 1970 MIT PhD thesis.

Winston, P. 1992. *Artificial Intelligence, Third edition*. Addison-Wesley.

Yap, C. K. 1977. A semantical analysis of intensional logics. Technical report, IBM Thomas J. Watson Research Center, Yorktown Heights, New York. RC 6893 (#29538), 12/13/77, 47pp.