Stanford Artificial Intelligence Laboratory
Memo AIM-334

February 1980

Computer Science Department
Report No. STAN-CS-80-788

CIRCUMSCRIPTION - A FORM OF NON-MONOTONIC REASONING

by

John McCarthy

Research sponsored by

Advanced Research Projects Agency
and
National Science Foundation

COMPUTER SCIENCE DEPARTMENT
Stanford University

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

(14) **REPORT DOCUMENTATION PAGE**

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER
AIM-334 (STAN-CS-80-788) , AIM-334

2. GOVT ACCESSION NO.

3. RECIPIENT'S CATALOG NUMBER
AD-A086 574

4. TITLE (and Subtitle)
(6) Circumscription - a form of non-monotonic reasoning.

5. TYPE OF REPORT & PERIOD COVERED
(9) technical rept., February 1980

6. PERFORMING ORG. REPORT NUMBER
AIM-334 (STAN-CS-80-788)

7. AUTHOR(s)
(10) John McCarthy

8. CONTRACT OR GRANT NUMBER(s)
(15) MDA 903-76-C-0206, ARPA Order-3494

9. PERFORMING ORGANIZATION NAME AND ADDRESS
Department of Computer Science
Stanford University
Stanford, California 94305 USA

10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS

11. CONTROLLING OFFICE NAME AND ADDRESS
Defense Advanced Research Projects Agency
Information Processing Techniques Office
1400 Wilson Avenue, Arlington, Virginia 22209

12. REPORT DATE
(11) February 1980

13. NO. OF PAGES
15 (12) 18

14. MONITORING AGENCY NAME & ADDRESS (if diff. from Controlling Office)
Mr. Philip Surra, Resident Representative
Office of Naval Research, Durand 165
Stanford University

15. SECURITY CLASS. (of this report)
Unclassified

15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Abstract: 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.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

094120

Stanford Artificial Intelligence Laboratory
Memo AIM-334

February 1980

Computer Science Department
Report No. STAN-CS-80-788

CIRCUMSCRIPTION - A FORM OF NON-MONOTONIC REASONING

by

John McCarthy

ABSTRACT

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.

This research was supported by the Advanced Research Projects Agency of the Department of Defense under ARPA Order No. 2494, Contract MDA903-76-C-0206 and National Science Foundation.

The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of Stanford University, or any agency of the U. S. Government.

CIRCUMSCRIPTION - A FORM OF NON-MONOTONIC REASONING

by John McCarthy

Stanford University

Abstract: 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.

1. Introduction - The Qualification Problem

(McCarthy 1960) 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 $<x, y \ldots, z>$ that can be shown to satisfy a relation $P(x, y, \ldots, z)$ 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.

2. The Need for Non-Monotonic 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 "non-monotonic" 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 non-monotonic 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 $P(x, \ldots, z)$ 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 non-monotonic reasoning.

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 (331,220,321,300311,110,221,020,031,010,021,000).

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 non-monotonic 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 non-monotonic 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 non-monotonic reasoning.

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

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

(1) can be regarded as asserting that the only tuples $(\bar{x})$ that satisfy P are those that have to - assuming the sentence A. Namely, (1) 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 (1)). Since (1) is an implication, we can assume both conjuncts on the left, and (1) 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 non-monotonic 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

2) $A(\Phi, \Psi) \wedge \forall \bar{x}.(\Phi(\bar{x}) \supset P(\bar{x})) \wedge \forall \bar{y}.(\Psi(\bar{y}) \supset Q(\bar{y})) \supset \forall \bar{x}.(P(\bar{x}) \supset \Phi(\bar{x})) \wedge \forall \bar{y}.(Q(\bar{y}) \supset \Psi(\bar{y}))$

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:

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

3) $isblock\ A \wedge isblock\ B \wedge isblock\ C$

asserting that A, B and C are blocks. Circumscribing $isblock$ in (3) gives the schema

4) $\Phi(A) \wedge \Phi(B) \wedge \Phi(C) \wedge \forall x.(\Phi(x) \supset isblock\ x) \supset \forall x.(isblock\ x \supset \Phi(x))$.

If we now substitute

5) $\Phi(x) \equiv (x = A \vee x = B \vee x = C)$

into (4) and use (3), the left side of the implication is seen to be true, and this gives

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

which asserts that the only blocks are A, B and C, i.e. just those objects that (3) requires to be blocks. This example is rather trivial, because (3) provides no way of generating new blocks from old ones. However, it shows that circumscriptive inference is non-monotonic since if we adjoin $\mathit{isblock}\ D$ to (3), we will no longer be able to infer (6).

2. Circumscribing the disjunction

7) $\mathit{isblock}\ A \vee \mathit{isblock}\ B$

leads to

8) $(\Phi(A) \vee \Phi(B)) \wedge \forall x.(\Phi(x) \supset \mathit{isblock}\ x) \supset \forall x.(\mathit{isblock}\ x \supset \Phi(x)).$

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

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

which simplifies to

10) $\mathit{isblock}\ A \supset \forall x.(\mathit{isblock}\ x \supset x{=}A)$

and

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

which simplifies to

12) $\mathit{isblock}\ B \supset \forall x.(\mathit{isblock}\ x \supset x{=}B).$

(10), (12) and (7) yield

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

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

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.

14) $\mathit{isnatnum}\ 0 \wedge \forall x.(\mathit{isnatnum}\ x \supset \mathit{isnainum}\ \mathit{succ}\ x).$

Circumscribing $\mathit{isnatnum}$ in (14) yields

15) $\Phi(0) \wedge \forall x.(\Phi(x) \supset \Phi(succ\ x)) \wedge \forall x.(\Phi(x) \supset isnatnum\ x) \supset \forall x.(isnatnum\ x \supset \Phi(x)).$

(15) asserts that the only natural numbers are those objects that (14) 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 (14) make the second conjunct drop out giving

16) $\Psi(0) \wedge \forall x.(\Psi(x) \supset \Psi(succ\ x)) \supset \forall x.(isnatnum\ x \supset \Psi(x)).$

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

17) $\forall x\ y\ s.(on(x, y, s) \supset above(x, y, s))$

and

18) $\forall x\ y\ z\ s.(above(x, y, s) \wedge above(y, z, s) \supset above(x, z, s)),$

i.e. *above* is a transitive relation. Circumscribing *above* in (17)∧(18) gives

19)
$$\begin{aligned}&\forall x\ y\ s.(on(x, y, s) \supset \Phi(x, y, s))\\ &\wedge \forall x\ y\ z\ s.(\Phi(x, y, s) \wedge \Phi(y, z, s) \supset \Phi(x, z, s))\\ &\wedge \forall x\ y\ s.(\Phi(x, y, s) \supset above(x, y, s))\\ &\qquad\qquad \supset \forall x\ y\ s.(above(x, y, s) \supset \Phi(x, y, s))\end{aligned}$$

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.

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

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

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

$$21) \qquad Axiom(\Phi) \wedge A^{\Phi} \wedge \forall x.(\Phi(x) \supset all(x)) \supset \forall x.(all(x) \supset \Phi(x)).$$

Now we justify our using the name *all* by adding the axiom $\forall x.all(x)$ so that (21) then simplifies precisely to (20).

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. 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, iff 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 $M \le_P N$, 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 iff $M' \le_P M$ 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 (1) when substituted for Φ. By the second conjunct of the left side P is an extension of P'. If the right side of (1) 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.

7 More on Blocks

The axiom

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

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.

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

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

25) $\forall x\, y\, s.(tooheavy\ x \supset prevents(weight\ x, move(x,y), s))$.

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 (22) 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 (23), (24) and (25), and this gives

26)
$$
\begin{aligned}
&(\neg isblock\ A \vee \neg isblock\ C \supset \Phi(NONBLOCK))\\
&\wedge (\neg clear(A, s0) \vee \neg clear(C, s0) \supset \Phi(COVERED))\\
&\wedge (tooheavy\ A \supset \Phi(weight\ A))\\
&\wedge \forall z.(\Phi(z) \supset prevents(z, move(A,C), s0))\\
&\qquad \supset \forall z.(prevents(z, move(A,C), s0) \supset \Phi(z))
\end{aligned}
$$

which says that the only things that can prevent the move are the phenomena described in (23), (24) and (25). Whether (26) 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 (26) simplifies to

27)
$$
\begin{aligned}
&(isblock\ A \wedge isblock\ B \wedge clear(A, s0) \wedge clear(B, s0) \wedge \neg tooheavy\ A\\
&\qquad \supset \forall z.\neg prevents(z, move(A,C), s0).
\end{aligned}
$$

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 *onness* assertion explicitly given for situation $s0$ is $on(A,B,s0)$. Circumscription of $\lambda x\, y.on(x,y,s0)$ in this assertion gives

28) $\Phi(A,B) \wedge \forall x\, y.(\Phi(x,y) \supset on(x,y,s0)) \supset \forall x\, y.(on(x,y,s0) \supset \Phi(x,y)),$

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

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

Using

30) $\forall x\, s.(clear(x,s) \equiv \forall y.\neg on(y,x,s))$

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.

8. Remarks and Acknowledgements

1. Circumscription is not a "non-monotonic logic". It is a form of non-monotonic 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 (1) is the result of suggestions by Ashok Chandra (August 1979) and Patrick Hayes (September 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

31) $\forall x.(A(x) \wedge (x \subset a) \supset (a \subset x))$.

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 as 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.

ADDENDUM: CIRCUMSCRIPTION AND OTHER NON-MONOTONIC FORMALISMS

Circumscription and the non-monotonic 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

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

where the properties of the predicate *on* are supposed expressed in a formula that we may call *Axiom*(*on*). The *M* in (32) stands for "is possible". We propose to replace (32) and *Axiom*(*on*) by the schema

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

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 (33) 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 (33) is essentially a consequence of the McDermott-Doyle formula (32). 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

34) $\exists\Phi.(Axiom(\Phi) \wedge on(train, tracks) \supset \neg safe\text{-}to\text{-}cross(tracks))$.

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 maybe needed to express common sense non-monotonic reasoning and it seems more powerful than any of the above-mentioned non-monotonic formalisms including circumscription.

The train example is a non-normal 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

35) $M \neg \mathit{safe\text{-}to\text{-}cross}(\mathit{tracks}) \supset \neg \mathit{safe\text{-}to\text{-}cross}(\mathit{tracks})$,

but this is unlikely to be equivalent in all cases and the non-normal 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-to-cross(tracks)* in the set of axioms. It therefore might not be completely satisfactory.

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*, this issue.

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 (1960): "Programs with Common Sense", *Proceedings of the Teddington Conference on the Mechanization of Thought Processes*, H.M. Stationery Office, 1960.

McCarthy, John (1960): "Programs with Common Sense", *Proceedings of the Teddington Conference on the Mechanization of Thought Processes*, H.M. Stationery Office, 1960.

McCarthy, John and Patrick Hayes (1969): "Some Philosophical Problems from the Standpoint of Artificial Intelligence," in D. Michie (ed), *Machine Intelligence* 4, American Elsevier, New York, NY.

McCarthy, John (1977): "Epistemological Problems of Artificial Intelligence", *Proceedings of the Fifth International Joint Conference on Artificial Intelligence*, M.I.T., Cambridge, MA.

McCarthy, John (1979a): "Ascribing Mental Qualities to Machines", *Philosophical Perspectives in Artificial Intelligence*, Martin Ringle, ed., Harvester Press, July 1979.

McCarthy, John (1979b): "First Order Theories of Individual Concepts and Propositions", in Michie, Donald (ed.) *Machine Intelligence* 9, University of Edinburgh Press.

McDermott, Drew and Jon Doyle (1980): "Non-Monotonic Logic I", *Artificial Intelligence*, this issue.

Reiter, Raymond (1980): "A Logic for Default Reasoning", *Artificial Intelligence*, this issue.

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

This version of CIRCUM.W80[S79,JMC] PUBbed at 10:20 on February 1, 1980.