FORMAL THEORIES OF ACTION
(PRELIMINARY REPORT)

Vladimir Lifschitz

Computer Science Department
Stanford University
Stanford, CA 94305, USA

Abstract

We apply circumscription to formalizing reasoning about the effects of actions in the framework of the situation calculus. The axiomatic description of causal connections between actions and changes allows us to solve the qualification problem and the frame problem using only simple forms of circumscription. The method is applied to the Hanks—McDermott shooting problem and to a blocks world in which blocks can be moved and painted.

1. Introduction

We consider the problem of formalizing reasoning about action in the framework of the *situation calculus* of McCarthy and Hayes (1969). A situation is the complete state of the universe at an instant of time. A *fluent* is a function denned on situations. For instance, in the blocks world, the location of a given block x is a fluent whose values are the possible locations of blocks. In the language of the situation calculus, the value of this fluent at *s* for a block *x* is denoted by *location(x,s).* If *s* is a situation and *a* an *action* then *result(a,s)* stands for the situation that results when the action *a* is carried out starting in the situation *s.*

We are primarily interested in the form of reasoning which Hanks and McDermott (1986) call "temporal projection": given some description of the current situation, some description of the effects of possible actions, and a sequence of actions to be performed, predict the properties of the world in the resulting situation.

A number of difficulties arise in attempts to describe the effects of actions in the situation calculus and in similar formalisms. One of them, known as the qualification *problem,* is related to the fact that the successful performance of an action may depend on a large number of qualifications. As a consequence, the axiom describing the effect of such an action has to include a long list of assumptions. It is desirable to treat each qualification as a separate fact and describe it by a separate axiom.

This research was partially supported by DARPA under Contract N0039-82-C-0250.

Another difficulty is that, in addition to the axioms describing how the properties of the current situation change when an action is performed, we need a large number of axioms listing the properties which are *not* affected by a given action. This is called the frame *problem.* For instance, when a robot is moving a block, appropriate "frame axioms" must guarantee that all the properties of the situation which are not related to the positions of the robot and the block (for instance, the positions of all other objects) remain the same.

John McCarthy proposed approaching the qualification problem and the frame problem using *circumscription* (McCarthy 1980, 1986). We assume that the reader has some familiarity with this concept; for the definition, see (McCarthy 1986). With regard to the qualification problem, the idea is, instead of listing all the qualifications in the antecedent of one axiom, to condition the axiom on the fact that (a certain aspect of) the action is not "abnormal" in the situation in which the action is performed. For each qualifying condition, a separate axiom postulates that the condition implies the abnormality of this aspect of the action. Circumscribing abnormality should ensure that the action has its expected effect whenever we are not in any of those exceptional situations which, according to the axioms, make the action abnormal.

For the frame problem, similarly, McCarthy proposes an axiom which guarantees that the location of *x* in the situation *result(a,s)* is the same as its location in the situation *s* unless a certain aspect of *a* is abnormal. Then circumscribing abnormality should allow us to prove that *x* does not change its position if the action consists of moving a block other than x, or in painting a block, etc., so that all "frame axioms" should become provable.

Further analysis has shown, however, that the result of minimizing abnormality relative to McCarthy's axiom set is not quite as strong as necessary for the successful formalization of the blocks world. The difficulty can be illustrated by the following example (McCarthy 1986). Consider the situations

$$51 = result(move(A, top\ B), SQ)$$

and

$$S2 = result(move(B,top\ A), S1),$$

where S0 is a situation with exactly blocks *A* and *B* on the table. We expect circumscription to guarantee that the first action, *move(A,top B),* is "normal", so that *A* is on top of *B* in situation S1. In view of that, the second action is "abnormal" (so that the positions of the blocks in situation S2 will be impossible to predict on the basis of McCarthy's axioms). However, if the first move is "abnormal" (for some unspecified reason), the positions of the blocks after the first action may remain unchanged, so that the second move may succeed after all. This observation shows that the intended model of McCarthy's axioms is not the only one in which the set of abnormal objects is minimal. In the intended model, the first action is normal, and the second is not; in the other model, it is the other way around. Using circumscription, we can only prove the common properties of all minimal models; minimizing abnormality will only give a disjunction.

Another example of such an "overweak disjunction", for an even simpler system of actions, is discussed in (Hanks and McDermott 1986). There are three possible actions: *load* (a gun), *wait,* and *shoot* (an individual, whose name, according to some accounts, is Fred). Normally, waiting does not cause any changes in the world, but shooting leads to Fred's death, provided, of course, that the gun is loaded. Assume that all three actions are performed, in the given order. An axiom set similar to McCarthy's axioms for the blocks world is shown to have an unintended minimal model, in which the gun mysteriously gets unloaded on the waiting stage, so that the shooting does not kill Fred. In the intended model, *wait* does not change the world, but *shoot* does; in the unintended model, it is the other way around. In either case, abnormality is minimal. Here again, the result of circumscription is too weak.

It is easy to notice that, in both examples, there is a conflict between minimizing abnormality at two instants of time. In the intended model, minimization at the earlier instant is preferred. But circumscription knows nothing about time; it interprets minimality of a predicate as the minimality of its extension relative to set inclusion. This, according to Hanks and McDermott, is the reason why circumscription does not live up to our expectations in application to temporal reasoning: "... the class of models we want our logic to select is not the "minimal models" in the set-inclusion sense of circumscription, but the "chronologically minimal" (a term due to Yoav Shoham): those in which normality assumptions are made in chronological order, from earliest to latest, or, equivalently, those in which abnormality occurs as late as possible" (Hanks and McDermott 1986).

The paper by Hanks and McDermott was presented at the AAAI conference in 1986, and three more talks given at the same conference, (Kautz 1986), (Lifschitz 1986) and (Shoham 1986), met their challenge by formalizing, in somewhat different contexts, the idea of chronological minimization. As could be expected, all of them had to use forms of logical minimization more general than circumscription in the sense of (McCarthy 1986).

In this paper we argue that formalizing chronological minimization is not the easiest way to solve the temporal projection problem. We start with another view on the difference between the intended and the unintended models of the axioms for actions: in the intended model, *all changes in the values of fluents* are *caused by actions,* whereas in other models this is not the case. To express this distinction formally, we add new primitive predicates to the language.* With this different choice of primitive concepts, the effects of actions can be characterized by simple axioms in the language of the situation calculus plus traditional circumscription. The axiomatic treatment of causality is the main distinctive feature of our approach.**

The method is illustrated by formalizing two examples: the shooting problem and the blocks world. The formalizations are reasonably complete, in the sense that they allow us to predict the effect of the execution of any sequence of actions. They are also computationally tractable, in the sense that, in both cases, the result of circumscription can be determined by simple syntactic methods (such as predicate completion). Proofs of these facts will be given in the full version of the paper (Lifschitz 1987).

2. The Yale Shooting Problem

To formalize the shooting story from the Hanks—McDermott paper, we use object variables of four sorts: for truth values *(v),* for actions (a), for situations *(s),* and for truth-valued fluents (*f*). The object constants are: truth values *false* and *true,* actions *load, wait* and *shoot,* situation S0 and fluents *loaded* and *alive.* The binary situation-valued function *result* must have an action term and a situation term as arguments. Finally, there are three predicate constants: *holds(f,s)* expresses that *f* is true in the situation s; *causes(a,f,v)* expresses that

* An early attempt to describe causation by axioms was made by Patrick Hayes (1971); see also (McDermott 1982).

** A similar solution was independently found by Brian Haugh (1987).

a causes the fluent f to take on the value v. *precond(f, a)* expresses that f is a precondition for the successful execution of a.*

Our axioms for the shooting problem can be classified into three groups. The first group describes the initial situation:

$$holds(alive, S0), \qquad (Y1.1)$$

$$\neg holds(loaded, S0). \qquad (71.2)$$

The second group tells us how the fluents are affected by actions:

$$causes(load, loaded, true), \qquad (72.1)$$

$$causes(shoot, loaded, false), \qquad (72.2)$$

$$causes(shoot, alive, false). \qquad (72.3)$$

These axioms describe the effects of *successfully performed* actions; they do not say *when* an action can be successful. This information is supplied separately:

$$precond(loaded, shoot). \qquad (72.4)$$

The last group consists of two axioms of a more general nature. We use the abbreviations:

$$success(a,s) \equiv \forall f(precond(f,a) \supset holds(f,s)),$$

$$affects(a,f,s) \equiv success(a,s) \wedge \exists v\, causes(a,f,v).$$

One axiom describes how the value of a fluent changes after an action affecting this fluent is carried out:

$$success(a,s) \wedge causes(a,f,v) \supset (holds(f, result(a,s)) \equiv v = true). \qquad (y3.1)$$

(Recall that v can take on two values here, *true* and *false;* the equivalence in the consequent of Y3.1 reduces to *holds(f,result(a,s))* in the first case and to the negation of this formula in the second.) If the fluent is not affected then its value remains the same:

$$\neg affects(a,f,s) \supset (holds(f, result(a,s)) \equiv holds(f,s)). \qquad (73.2)$$

* Notice that, syntactically, the variables for actions are object variables, not function variables, even though each action a has a function associated with it: the situational fluent s *result(a,s).* Similarly, the variables for truth-valued fluents are object variables, not predicate variables. The value of f at s is represented by *holds(f,s).*

This axiom set obviously does not include a number of assumptions that may be essential. The axioms do not tell us, for instance, whether *false* and *true* are different from each other and whether there are any truth values other than these two; we do not know whether *load, shoot* and *wait* are three different actions, etc. In this preliminary discussion, instead of making all these assumptions explicit, we limit our attention to the *term models* of the axioms, in which every element of the universe is represented by a ground term of the language, and different terms represent different objects. (Such models are also called *Herbrand models* in case of universal theories). We will identify the universe of a term model with the set of ground terms.

It remains to specify how circumscription is used. We circumscribe *causes* and *precond,* with the remaining predicate *holds* allowed to vary, relative to the conjunction 7 of the axioms 71.1—73.2. This will lead us to the conclusion that *causes* and *precond* are true only when this is required by the axioms of the second group, 72. The minimization of *causes* will imply, in view of axiom 73.2, that the only changes taking place in the world are those corresponding to axioms 72.1 -72.3. This solves the frame problem. The minimization of *precond,* in view of the definition of *success* will imply that the only unsuccessful actions are those in which precondition 72.4 is violated. This solves the qualification problem.

The main technical difference between this formulation and other attempts to apply circumscription to the situation calculus is that the predicates which we circumscribe, *causes* and *precond,* are *not* fluents, they do not have situation arguments. The predicate *success,* which does have a situation argument, is *defined* in terms of the minimized primitive *precond.* Since the minimized predicates have no situation arguments, the conflict between minimizing in earlier and later instants of time simply cannot arise.

It is not difficult to prove that this circumscription 7 has a unique term model Mo- In this sense, it completely determines whether any given sentence is true or false. In particular, the formula

$$holds(alive, result(shoot, result(wait, result(load, S0))))$$

is false in Mo. The predicates *causes* and *precond* can be described in Mo by explicit definitions:

$$causes(a,p,f) \equiv [(a = load \wedge p = loaded \wedge f = true) \vee (a = shoot \wedge p = loaded \wedge f = false) \vee (a = shoot \wedge p = alive \wedge f = false)],$$

$$precond(f,a) \equiv (f = loaded \wedge a = shoot).$$

The definitions can be obtained by applying the predicate completion procedure to axioms *Y2.* They can be also characterized as the result of circumscribing the predicates *causes* and *prccond* relative to Y'2. These predicates occur both in *Y2* and in Y3, but we see that axioms *Y3* can be ignored when the result of circumscription is determined.

The interpretation of *holds* in Mo can be viewed as the description of the work of a deterministic finite automaton *A* with 4 internal states, corresponding to all possible combinations of values of the fluents *loaded* and *alive.* The initial state of *A* is *(false,true).* Its input symbols are *load, wait, shoot.* Strings of input symbols are "plans", and the goal condition of a planning problem would define a set of final states. For example, the goal *-alive* corresponds to selecting *(false,false)* and *(true, false)* as the final states, and the plan *load, wait, shoot* is accepted by *A* with this set of final states. Each state of *A* corresponds to a set of situations which are indistinguishable in terms of the fluents *loaded* and *alive.*

3. An Alternative Formulation

The formalization of the Yale shooting problem proposed above has two defects. First, we were only able to show that actions lead to the expected results in *term models* of the circumscription, not in arbitrary models. Second, we exploited some special features of the problem that are not present in more complex problems of this kind. Now we will consider an improved, though somewhat less economical, formalization of the shooting story. Here are the main differences between the two versions.

1. Since it is essential in the first solution that, in a term model, different ground terms have different values, we will need some *uniqueness of names* axioms, expressing this property. The following notation will be useful: if o, . . . , *c„* are constants of the same sort then *U[c.\,...* ,c„] stands for the set of axioms c, *c,* (1 < *i* < *j* < n). For instance, *U[load, wait, shoot]* is

load # watt, load # shoot, wait # shoot.

2. It is also essential in the first solution that every situation in the term model is the result of executing a sequence of actions in 50.* We see several ways to deal with this problem, all of them, admittedly, somewhat unintuitive. In the new solution we choose to restrict axioms *Y3* to a subclass of *relevant* situations; we will postulate that 50 is relevant and that the result of executing an action in a relevant situation is relevant also.

* This fact allows us to define the interpretation of *holds* in Mo by recursion on the second argument.

3. The only precondition in the Yale shooting problem (given by axiom *Y2A)* is extremely simple: *loaded* is one of the "coordinate fluents[11] defining the state of the corresponding automaton (see the end of Section 2). Even in the blocks world, we need preconditions of a more complex nature, such as *clear l* and *clear top b* for the action *movc(b,l),* where *clear* is explicitly defined in terms of primitive fluents. Accordingly, in the second solution we explicitly distinguish between fluents in the general sense and primitive fluents. Only a primitive fluent can occur as the second argument of *causes,* but preconditions do not have to be primitive.

4. The value which *a* causes / to take on may depend on the situation in which *a* is executed. For instance, toggling a switch may cause its status to change in two different ways, depending on its current status.* Accordingly, we will allow a truth-valued fluent (not necessarily primitive), rather than a truth value, to be the last argument of *causes]* the value of that fluent in the situation ,s is the value that f will take on in the situation *result(a,s),* provided *a* is successful. The effect of toggling a switch can be described then by an axiom such as

causes(toggle, on, not ON)

(not is the function which transforms a fluent into its negation).

In the new formalization of the Yale shooting we have an additional sort of variables, primitive fluents. On the other hand, we do not need variables for truth values any more. Thus the language has variables for actions (a), for situations *(s),* for truth-valued fluents (/), and for primitive truth-valued fluents *(p).* This last sort is considered a subtype of fluents, so that a term representing a primitive fluent can be used wherever the syntax requires a fluent. (Accordingly, in a structure for the language the domain of primitive fluents must be a subset of the domain of fluents).

The object constants are the same as in the language of Section 2. As before, *load, wa.it* and *shoot* are action constants and 50 a situation constant; *loaded* and *alive* are primitive fluents and *false* and *true* are fluents (intuitively, the identically false fluent and the identically true fluent). The only function constant, *result,* and the predicate constants *holds* and *precond* are used as in the first formulation. The arguments of *causes* must be an action, a primitive fluent and a fluent. In addition, we

* This problem was pointed out by Michael Georgeff.

need the unary predicate *relevant,* whose argument must be a situation term.

The axioms of the first two groups, *Y* and *Y2* (Section 2) are included in the new formulation without change. (It is easy to see that all of them are well-formed formulas of the new language). The definitions of *success* and *affects* remain the same, except that variables of different sorts are required now in the definition of *affects:*

$$affects(a,p,s) \equiv success(a,s) \wedge \exists f\ causes(a,p,f).$$

The counterpart of Y3.1 is the following Law *of Change:*

$$relevant\ s \wedge success(a,s) \wedge causes(a,p,f) \supset (holds(p, result(a,s)) \equiv holds(f,s)). \quad (LC)$$

Y3.2 becomes the Commonsense Law of Inertia:

$$relevant\ s \wedge \neg affects(a,p,s) \supset (holds(p, result(a,s)) \equiv holds(p,s)). \quad (LI)$$

In addition, we need a few axioms which have no counterparts in the first version. Let *Act* be the list of actions *load, wait, shoot,* and let *Fl* be the list of fluents

loaded, alive, false, true.

We need the following uniqueness of names axioms:

$$U[Act], \quad (VI)$$

$$U[Fl], \quad (U2)$$

$$result(a1,s1) = result(a2,s2) \supset a1 = a2 \wedge s1 = s2. \quad (U3)$$

Axiom *U3* represents the assumption that a situation includes a complete description of its history, i.e., uniquely defines what action has led to it and in what situation the action was performed.

We also need to assume that the non-primitive fluent constants

false,true

denote objects which do not belong to the subdomain of primitive fluents. If this list is denoted by *NPF* then this assumption can be expressed by

$$p \neq t \qquad (t \in NPF). \quad (u4)$$

The next group of axioms consists of the fluent *definitions* for the non-primitive fluents:

$$holds(true, s), \quad (FD.true)$$

$$\neg holds(false, s). \quad \{FD.false)$$

Finally, there are 3 axioms for the new predicate *relevant:*

relevant 50, *(RI)*

$$relevant\ s \supset relevant\ result(a,s), \quad (R2)$$

$$result(a,s) = S0 \supset \neg relevant\ s. \quad (R2)$$

Axioms RI, *R2* imply

relevant s

for any situation term s without variables. Axiom *R3* guarantees, in the presence of *RI* and *R2,* that a sequence of actions cannot possibly lead from 50 back to 50 (i.e., 50 is not "in the future" of 50).

By Y' we denote the conjunction of the axioms *Y\, Y2, LC, LI, U, FD, R.* The predicates *causes* and *precond* are circumscribed now relative to *Y'* with the remaining predicates *holds* and *relevant* allowed to vary. This circumscription has many non-isomorphic models. But in all of them the predicates *causes* and *precond* satisfy the following conditions (essentially identical to their definitions in the model Mo of *Y):*

$$\begin{aligned} causes(a,p,f) \equiv\ & (a = load \wedge p = loaded \wedge f = true) \\ & \vee (a = shoot \wedge p = loaded \wedge f = false) \\ & \vee (a = shoot \wedge p = alive \wedge f = false), \end{aligned}$$

$$precond(f,a) \equiv (f = loaded \wedge a = shoot).$$

Moreover, every ground atom is either true in all models of this circumscription, or false in all of them. Thus our "circumscriptive theory" is sufficiently strong to decide any sentence without variables.

4. The Blocks World

Now we are ready to consider a slightly larger example: a blocks world in which blocks can be moved and painted, as in (McCarthy 1986). Even though the object domain is now entirely different, the primitive concepts *result, holds, causes* and *precond* will be used again, and some axioms (notably, the law of change and the law of inertia) will be included in the axiom set for the blocks world in exactly the same form. These axioms represent the general properties of actions which are likely to be useful for constructing axiomatic descriptions of many different object domains.

As before, we have variables for actions (a), for situations *(s),* for truth-valued fluents (/), and for primitive truth-valued fluents (p), the last sort being again a subtype of fluents. in addition, there are 3 "domain-specific" sorts: blocks (6), locations (/) and colors (c).

Object constants: So, *true, false,* blocks $Block_1$, ..., $Block_N$ (where N is a fixed positive integer), location *Table,* and, finally, colors *Red, White* and *Blue.* Here is the list of function constants, along with the sorts of their arguments and values:

top	$(b \rightarrow l)$,
at	$(b, l \rightarrow p)$,
color	$(b, c \rightarrow p)$,
clear	$(l \rightarrow f)$,
move	$(b, l \rightarrow a)$,
paint	$(b, c \rightarrow a)$,
result	$(a, s \rightarrow s)$.

The predicate constants are the same as in Section 3: *holds, causes, precond* and *relevant.*

The first group of axioms contains some general facts about blocks, their locations and colors:

$$b = Block_1 \vee \ldots \vee b = Block_N, \qquad (B0.1)$$

$$holds(at(b, l1), s) \wedge holds(at(b, l2), s) \supset l1 = l2, \qquad (B0.2)$$

$$holds(color(b, c1), s) \wedge holds(color(b, c2), s) \supset c1 = c2. \qquad (B0.3)$$

Initially, all blocks are white and on the table:

$$holds(at(b, Table), S0), \qquad (B1.1)$$

$$holds(color(b, White), S0). \qquad (B1.2)$$

The effects of actions:

$$causes(move(b, l), at(b, l), true), \qquad (B2.1)$$

$$l \neq l1 \supset causes(move(b, l), at(b, l1), false), \qquad (B2.2)$$

$$causes(paint(b, c), color(b, c), true), \qquad (B2.3)$$

$$c \neq c1 \supset causes(paint(b, c), color(b, c1), false), \qquad (B2.4)$$

$$l \neq Table \supset precond(clear\ l, move(b, l)), \qquad (B2.5)$$

$$precond(clear\ top\ b, move(b, l)), \qquad (B2.6)$$

$$precond(false, move(b, top\ b)). \qquad (B2.7)$$

(It is impossible to move a block onto its own top). Axioms $B2.2$ and $B2.4$ may seem redundant: we already know from axioms $B0.2$ and $B0.3$ that a block cannot possibly be at two places or have two colors simultaneously. But our method requires that all changes of all primitive fluents caused by an action be described explicitly. It does not solve the "ramification problem" in the sense of (Finger 1986) and (Ginsberg and Smith 1986).*

* I am indebted to Matthew Ginsberg for this important observation.

The laws of change and inertia *LC* and *LI* are formulated as in *Y'* above. To state the counterparts of the uniqueness of names axioms from Y, we define *Act* and *Fl* for the blocks world language as follows:

$$Act = (move, paint),$$

$$Fl = (at, color, true, false, clear).$$

Now the list *Act* consists of the symbols for *functions* returning actions, not of object constants, as in *Y'. Fl* contains both object constants and function constants. We will extend the definition of *U[...]* to the case when some or all of its arguments are functions. We can treat constants as 0-ary functions and thus define the meaning of $U[f_1, \ldots, f_n]$, where $f_1, \ldots, f_n$ are functions returning values of the same sort. By definition, this stands for the following set of axioms:

$$f_i x_1 \ldots x_k \neq f_j y_1 \ldots y_l$$

for all $i < j$, and

$$f_i x_1 \ldots x_k = f_i y_1 \ldots y_k \supset (x_1 = y_1 \wedge \ldots x_k = y_k),$$

for all /, of arity $k > 0$, where $x_1, \ldots, y_1, \ldots$ are variables of appropriate sorts. These axioms express that f_1f.*1*..., /^{n} are injections with disjoint ranges. If each /, is an object constant then we get the special case defined in Section 3 above. Example: Axiom *VZ* can be written in this notation as *U[result).*

NPF is defined as the list of terms

false, true, clear l.

Now axioms U_1—$U4$ are expressed exactly as in Section 3. In addition, we have 3 domain-specific uniqueness axioms:

$$U[B_1, \ldots, B_N], \qquad (U.block)$$

$$U[Table, top], \qquad (U.location)$$

$$U[Red, White, Blue]. \qquad (U.color)$$

The fluent definitions include, in addition to *FD.true* and *FD.false,* the definition of *clear:*

$$holds(clear\ l, s) \equiv \forall b \neg holds(at(b, l), s). \qquad (FD.clear)$$

Finally, the axioms *R* are formulated as in *Y'.*

By *B* we denote the conjunction of all these axioms. As before, we circumscribe *causes* and *precond,* with *holds* and *relevant* allowed to vary. It turns out that in each model of this circumscription the minimized predicates satisfy the formulas obtained by "completing" axioms $B2$: $causes(a,p,f)$ is equivalent to

$$\exists b,l,l1,c,c1[(a = move(b,l) \wedge p = at(b,l) \wedge f = true) \vee (a = move(b,l) \wedge move(b,l1) \wedge l \neq l1 \wedge f = false) \vee (a = paint(b,c) \wedge p = color(b,c2) \wedge f = true) \vee (a = paint(b,c) \wedge move(b,c1) \wedge c \neq c1 \wedge f = false)],$$

and $precond(f,a)$ is equivalent to

$$\exists b,l[a = move(b,l) \wedge ((f = clear\ l \wedge l \neq Table) \vee f = clear\ top\ b \vee (l = top\ b \wedge f = false))].$$

Every ground atom is either true in all models of this circumscription, or false in all of them, so that this "circumscriptive theory", like the theory from Section 3, is complete on the level of atomic formulas.

Acknowledgements

This paper is an outgrowth of ideas developed by John McCarthy in his research on commonsense and nonmonotonic reasoning, and I am grateful for the opportunity to discuss my work with him. I have also benefitted from discussions with David Etherington, Michael Gelfond, Michael Georgeff, Matthew Ginsberg, Robert Givan, Brian Haugh, Patrick Hayes, Robert Kowalski, Drew McDermott, Nils Nilsson, Donald Perlis, Raymond Reiter and Yoav Shoham.

References

1. Finger, J., *Exploiting Constraunts in Design Synthesis,* PhD thesis, Stanford University, Stanford, CA, 1986.

2. Ginsberg, M., and Smith, D., Reasoning about action I: a possible world approach, *Proc. Workshop on the Frame Problem in Artificial Intelligence,* Morgan Kauffman, 1987 (to appear).

3. Hanks, S. and McDermott, D., Default reasoning, nonmonotonic logics, and the frame problem, *Proc. AAAI-86* 1, 1986, 328-333.

4. Haugh, B., Simple causal minimizations for temporal persistence and projection, *Proc. AAAI-87* (to appear).

5. Hayes, P., A logic of actions, in: Meltzer, B. and Michie, D. (Eds.), *Machine Intelligence* 6, Edinburgh University Press, Edinburgh, 1971, 495-520.

6. Kautz, H., The logic of persistence, *Proc. AAAI-86* 1, 1986, 401-405.

7. Lifschitz, V., Pointwise circumscription: preliminary report, *Proc. AAAI-86* 1, 1986, 406-410.

8. Lifschitz, V., Formal theories of action, *Proc. Workshop on the Frame Problem in Artificial Intelligence,* Morgan Kauffman, 1987 (to appear).

9. McCarthy, J., Circumscription -- a form of non-monotonic reasoning, Artiflciai *Intelligence* 13 (1980), 27-39.

10. McCarthy, J., Applications of circumscription to formalizing commonsense knowledge, *Artificial Intelligence* 28 (1986), 89-118.

12. McCarthy, J. and Hayes, P., Some philosophical problems from the standpoint of artificial intelligence, in: Meltzer, B. and Michie, D. (Eds.), Machine *Intelligence* 4, Edinburgh University Press, Edinburgh, 1969, 463-502.

13. McDermott, D., A temporal logic for reasoning about processes and plans, Cognitive *Science* 6 (1982), 101-155.

14. Shoham, Y., Chronological ignorance: time, nonmonotonicity, necessity and causal theories, *Proc. AAAI-86* 1, 1986, 389-393.