THE LOGIC OF PERSISTENCE

Henry A. Kautz

Department of Computer Science
University of Rochester
Rochester, New York 14627

ABSTRACT

A recent paper [Hanks1985] examines temporal reasoning as an example of default reasoning. They conclude that all current systems of default reasoning, including non-monotonic logic, default logic, and circumscription, are inadequate for reasoning about persistence. I present a way of representing persistence in a framework based on a generalization of circumscription, which captures Hanks and McDermott's procedural representation.

1. Persistence

The frame problem is that of representing a dynamic world so that one can formally infer the facts whose truth values are not changed by a given action. A temporal world model allows one to assert that various actions occur at various times, and to be silent about other times. When one reasons with such a model, the frame problem is generalized to the persistence problem: given that no relevant action, or perhaps no action at all, occurred over a stretch of time, one may need to infer that certain facts do not change their truth values over that time. In other words, one needs to represent the "inertia" of the world, the moment to moment persistence of many of its properties.

Examples of persistence abound in everyday reasoning. Sitting in my office, I can infer that my car is in the parking lot, because that is where I left it this morning. [Hanks1985] examines the following example, here simplified. Assume a simple linear, discrete model of time, containing instants 1, 2, 3, etc. At time 1 John is alive, and a gun aimed at John is loaded. At time 3 the gun is fired. We know that if the gun is loaded when it is fired, John will die at the next moment of time. We would like to conclude that John is not alive at time 4. In order to do so, we must make the persistence inference that the gun stays loaded from times 1 to 3. (See figure 1.)

This report describes work done in the Department of Computer Science at the University of Rochester. It was supported in part by National Science Foundation grant DCR-8502481.

2. Problems with Default Reasoning

We would like to find some simple rule of default inference which captures persistence reasoning. Hanks and McDermott describe "obvious" solutions to the persistence problem using Reiter's default logic, McCarthy's circumscription operation, and McDermott and Doyle's non-monotonic logic. In default logic, for example, one would include an rule which stated that if a fact held at a time T1, and it was consistent that it held over an interval immediately following time T1, then infer that it does hold over T1. In the circumscriptive approach, one could define a "clipping event" which occurs whenever a fact changes truth value. Persistence is indirectly asserted by circumscribing (minimizing) the predicate which holds of all clipping events.

While intuitively appealing, these approaches do not work. The basic problem, Hanks and McDermott point out, is that default inferences are not prioritized by each system. For example, applying default rules in different orders yields different extensions; in circumscription, many different models of the axioms may be minimal in the "clipping" predicate. Yet only some of these extensions or minimal models correspond to the intuitive understanding of persistence.

Consider the gun example. The axioms have a minimal model (or corresponding extension) in which the fact ALIVE persists, but the fact LOADED is (mysteriously) clipped between times 1 and 3. (See figure 2.) Therefore simply circumscribing clipped (or adding default rules) does not sanction inferences about persistence.

3. A Procedural Solution

Hanks provides a temporal-assertion management program which computes persistences. Hanks's program functions by computing persistences in temporal order, from the past to the future. For example, the persistence of LOADED is computed before the persistence of ALIVE, and so the program concludes that John dies. The program reflects our intuitions in many cases because it captures the temporal order of causality: the gun being loaded can cause John to die, and so has precedence over it.

Hanks is not optimistic about the ability of any default logic to handle this reasoning properly: "...If a significant part of defeasible reasoning can't be represented by default logics, and if in the cases where the logics fail we have no better way of describing the reasoning process than by a direct procedural characterization (like our program or its inductive definition), then logic as an AI representation language begins to look less and less attractive." Such pessimism may be premature. It is possible to represent many kinds of ordered defaults in an declarative representation. We show how this can be done in a circumscriptive framework.

4. Model Theory

The semantics of circumscription are based on the idea of minimal entailment. One statement entails another if all models of the first are also models of the second. Suppose a partial order is defined over class of models. The minimal models of a statement are those which have no strict predecessor in the partial order. Then one statement minimally entails another if all minimal models of the first are also models of the second.

McCarthy's original formulation of circumscription [McCarthy1980] defined the partial order over models in terms of the extension of some predicate, say P. A model M1 would be less than a model M2 if the extension of P in M1 is a subset of its extension in M2, and M1 and M2 are otherwise the same. Newer work [McCarthy1985] has refined this definition, largely concentrating on the role of the non-circumscribed predicates in the minimization. But many other variations on circumscription are possible.

Let facts (such as LOADED) be represented by terms, and the atom

$$\mathrm{Hold}(t,f)$$

be used to assert that fact f holds at time t. The predicate Clip holds of a time and a fact if the fact becomes false at that time; otherwise, the truth-value of the fact persists from the earlier instant. That is:

$$\mathrm{Hold}(t,f) \supset (\mathrm{Hold}(t+1,f) \oplus \mathrm{Clip}(t+1,f))$$

(The symbol $\oplus$ represents exclusive or.) Suppose we are given some assertions about when various facts hold. We wish to define a partial order over models of these sentences which reflects our intuitions about persistence.

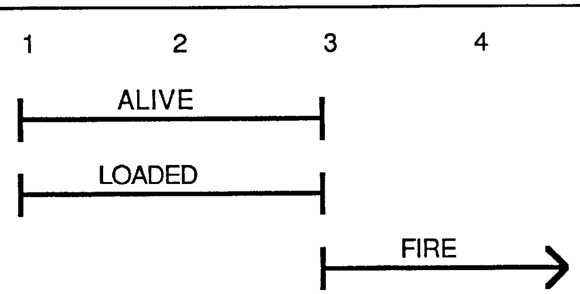

Figure 1

"Good" models, Hanks and McDermott suggest, are ones in which earlier facts persist as long as possible, and so should fall at the beginning of the ordering. Where M1 and M2 are models, M1 is as good or better than M2 if every clipping in M1 is either matched by an identical clipping in M2, or by an earlier clipping in M2 (possibly of some different fact) which does not also appear in M1.

The less than or equal relation between models is formally defined as follows. Where M1 is a model and P is a predicate, the expression M1[P] yields the extension of P in M1. The extension of a binary predicate such as Clip is a set of pairs, where the pair of x and y is written $\langle x, y \rangle$. Models can be compared only if they interpret constant, function, and predicate symbols other than Clip or Hold in the same way. In particular, this means that the models agree on the predicate "<", which is used to order time instances. Because models may be compared even if they do not agree on the predicate Hold, that predicate (as well as Clip) is said to *vary* during the minimization.

$M1 \leq M2$ if and only if

(i) M1 and M2 have the same domain

(ii) Every constant, function, and predicate symbol other than Clip and Hold receives the same interpretation in M1 and M2.

(iii) The following (meta-theoretic) statement is true:

$$\begin{aligned} &\langle f, t \rangle \in M1[\mathrm{Clip}] \supset \\ &\quad \langle f, t \rangle \in M2[\mathrm{Clip}] \vee \\ &\quad \exists\, t', f' \,.\, \langle f', t' \rangle \in M2[\mathrm{Clip}]\ \& \\ &\qquad \langle f', t' \rangle \notin M1[\mathrm{Clip}]\ \& \\ &\qquad \langle t', t \rangle \in M1[<] \end{aligned}$$

The final clause in this formula means that the time instant t' is before the time instant t.

A model M1 is strictly better than M2 ($M1<M2$) just in case $M1 \leq M2$ and it is not the case that $M2 \leq M1$. From this definition one can prove that if $M1<M2$, then (in terms of the Clip predicate) M1 and M2 are identical up to some time t'; at t', the set of clippings in M2 properly includes the set of clippings in M1.

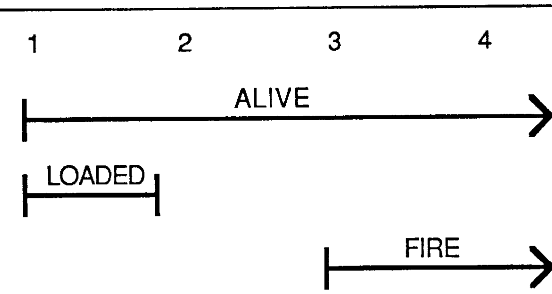

Figure 2

The minimal models are those M such that there is no M' such that M'<M. It is important to understand that the set of models is only *partially* ordered; there will be many minimal models.

The set of minimal models may be empty if there is an infinite chain of models, M1 > M2 > M3 > ... . This can occur if we minimize an existential statement of the form, "f will eventually be clipped", with no upper bound placed on the time of clipping. The preference order will attempt to postpone the clipping for an infinite period of time. This problem does not occur if such an unknown time is given a (skolem) constant name, however, due to the fact that constants and functions do not vary in the minimization.

5. Proof Theory

McCarthy's circumscription formula is a statement in 2nd-order logic which entails those statements true in all the minimal models of a predicate. The following persistence circumscription formula entails those statements true in models minimal in the partial order defined above. Let K(Clip,Hold) be our initial set of temporal assertions. We write an expression such as K(Foo,Bar) to stand for the set of sentences obtained by substituting the predicates Foo and Bar for every occurrence of Clip and Hold in K(Clip,Hold) respectively. The variables c and h range over predicates.

$$\begin{aligned}&\forall\, c,h\,.\\&\quad \{K(c,h)\ \&\\&\quad\quad \forall\, t,f\,.\ c(t,f) \supset\\&\quad\quad\quad [Clip(t,f) \vee\\&\quad\quad\quad \exists\, t2,f2\,.\ t2<t\ \&\ Clip(t2,f2)\ \&\ \neg c(t2,f2)]\}\\&\quad \supset \forall\, t,f\,.\ Clip(t,f) \equiv c(t,f)\end{aligned}$$

The formula can be informally understood as follows. Suppose that c and h are arbitrary predicates which satisfies all the constraints placed by the knowledge base on the predicates Clip and Holds respectively. Furthermore, suppose whenever c holds of a particular time and a particular fact, then either Clip also holds of that time and fact, or Clip holds of some earlier time and fact which are not in the extension of c. The conclusion is that c and Clip are identical; the second alternative is never the case. There cannot be a predicate which satisfies all the constraints on Clip, yet allows some fact to persist for a longer time, without having to clip some other fact at that time. The predicate-variable h was introduced in order to allow Holds to vary during the minimization of Clip.

In order to use this formula, we must select particular instantiations for the variables c and h, such that the initial set of assertions K(Clip,Hold) entails the main antecedent (in curly braces). Typically c is instantiated as a lambda expression which enumerates the the desired set of clippings. The variable h is instantiated by a lambda expression which describes which and when facts hold in the corresponding minimal models. Modus ponens then allows us to conclude that the extension of Clip is precisely the desired set of clippings: Clip(t,f) $\equiv$ c(t,f).

It is possible to show that this formula is valid in all models minimal in the above sense. As with standard circumscription, the formula is inconsistent if there are no minimal models. [Lifschitz1985] develops a generic circumscription-like formula based on pre-orders. The formula above is easy to express in Lifschitz' compact and elegant notation.

6. Example

The gun example illustrates the use of persistence circumscription. K(Clip,Hold) is the following set of statements. Not shown are unique name axioms, such as LOADED $\neq$ ALIVE, etc.

$$Hold(t,f) \supset (Hold(t+1,f) \oplus Clip(t+1,f))$$

$$\begin{aligned}&Hold(t,FIRE)\ \&\ Hold(t,LOADED) \supset\\&\quad \neg Hold(t+1,LOADED)\ \&\ \neg Hold(t+1,ALIVE)\end{aligned}$$

$$Hold(1,LOADED)$$

$$Hold(1,ALIVE)$$

$$Hold(3,FIRE)$$

The goal is to prove that ¬Hold(4,ALIVE). (A more complete set of axioms would also state that if something is a fact, and it does not hold at a time, then its negation holds at that time. This complication would not materially change our solution.)

Our intuitions tell us that the only (required) clipping event occurs at time 4, when both LOADED and ALIVE become false (as in figure 1). The instantiation for c is therefore:

$$c = \lambda\, t,f\,.\ t=4\ \&\ (f=LOADED \vee f=ALIVE)$$

When do various facts hold? Again referring to figure 1, we see that LOADED and ALIVE hold between times 1 and 3, and FIRE begins holding at time 3 (and persists thereafter). For h we can thus choose:

$$\begin{aligned}h = \lambda\, &t,f\,.\\&(f=LOADED \supset 1\leq t\leq 3)\ \&\\&(f=ALIVE \supset 1\leq t\leq 3)\ \&\\&(f=FIRE \supset t\geq 3)\ \&\\&(f=LOADED \vee f=ALIVE \vee f=FIRE)\end{aligned}$$

These expressions are placed in the persistence circumscription formula, which is then simplified. This involves proving that the main antecedent of the formula:

$$\{K(c,h)\ \&\ \forall t,f\,.\,c(t,f) \supset [Clip(t,f) \lor \exists t2,f2\,.\,t2<t\ \&\ Clip(t2,f2)\ \&\ \neg c(t2,f2)]\} \supset \forall t,f\,.\,Clip(t,f) \equiv c(t,f)$$

must be true, where c and h are defined as above. This is done by showing (i) that K(Clip,Hold) entails K(c,h), and (ii) that K(Clip,Hold) entails:

$$\star \quad \forall t,f\,.\,c(t,f) \supset [Clip(t,f) \lor \exists t2,f2\,.\,t2<t\ \&\ Clip(t2,f2)\ \&\ \neg c(t2,f2)]$$

The first part of the proof involves substituting c and h for Clip and Holds in the initial knowledge base and simplifying, which is straightforward but tedious. For example, the formula Hold(1,LOADED) becomes h(1,LOADED), which is:

$$[\lambda\, t,f\,.\,(f=LOADED \supset 1 \leq t \leq 3)\ \&\ (f=ALIVE \supset 1 \leq t \leq 3)\ \&\ (f=FIRE \supset t \geq 3)\ \&\ (f=LOADED \lor f=ALIVE \lor f=FIRE)]\,(1,LOADED)$$

This expression reduces to:

$$(LOADED=LOADED \supset 1 \leq 1 \leq 3)\ \&\ (LOADED=ALIVE \supset 1 \leq 1 \leq 3)\ \&\ (LOADED=FIRE \supset 1 \geq 3)\ \&\ (LOADED=LOADED \lor LOADED=ALIVE \lor LOADED=FIRE)$$

which, given the unique name axioms mentioned above, is a tautology.

The second step, as mentioned above, is to show that K(Clip,Hold) entails the statement marked with a (★). The antecedent of (★) is false, and the statement therefore true, except when t=4, and f=LOADED or f=ALIVE. Therefore we must show that:

$$Clip(4,LOADED) \lor \exists t2,f2\,.\,t2<4\ \&\ Clip(t2,f2)\ \&\ \neg c(t2,f2)$$

and

$$\dagger \quad Clip(4,ALIVE) \lor \exists t2,f2\,.\,t2<4\ \&\ Clip(t2,f2)\ \&\ \neg c(t2,f2)$$

Consider the sentence involving ALIVE, marked with a (†). We can show this statement is true by showing that if the second main disjunct is false, then the first disjunct must be true. So suppose that

$$\exists t2,f2\,.\,t2<4\ \&\ Clip(t2,f2)\ \&\ \neg c(t2,f2)$$

is false. This means that there is no clipping event before time 4. K(Clip,Hold) includes the statements Hold(1,ALIVE) and Hold(1,LOADED). The axiom

$$Hold(t,f) \supset (Hold(t+1,f) \oplus Clip(t+1,f))$$

can therefore be applied for times t=1 and t=2, giving the conclusion

$$Hold(3,ALIVE)\ \&\ Hold(3,LOADED)$$

Since Hold(3,FIRE), the axiom about firing loaded guns tells us that ¬Hold(4,ALIVE). Since Hold(3,ALIVE), we finally conclude that Clip(4,ALIVE), the first disjunct of (†), is true. Therefore (†) is true. The sentence (just before (†)) involving LOADED can be proven in a similar manner.

Thus the statement (★) is true, the main antecedent of the instantiated persistence circumscription formula is true, and so

$$Clip(t,f) \equiv c(t,f)$$

Since c(4,ALIVE), it must the case that Clip(4,ALIVE), and so ¬Hold(4,ALIVE).

Discussion

Several morals can be drawn from this exercise. One is that in reasoning about time, and probably most other applications, default inferences must be properly ordered. Another is that we may need to step beyond the incremental progression of circumscriptive techniques, from predicate circumscription, to circumscription with variables, to formula circumscription, and view circumscription as a general framework for expressing inference in terms of various classes of minimal models. A final moral is that by thinking about default inference in terms of relationships between models, we may more readily see the inadequacies of our own purported solutions.

The particular formulas just presented do not solve in the persistence problem in general. Recall the example using persistence to infer that my car is in the parking lot. Suppose I learn at time 1000 that my car is gone. Using the techniques just described, I can infer that the car was in the parking lot up to the shortest possible time before I knew it was gone. This is clearly an unreasonable inference. Someone could have stolen it five minutes after I left it there; I have no reason to prefer an explanation in which it vanished five seconds before I glanced out my office window. The inadequacy is ontological: we can't handle persistence properly until we have a richer theory of causation. The purely temporal solution often works because the flow of time reflects the order of physical causation. When the full story of causation is told, we then require an efficient algorithm for performing the necessary deductions, such as Hanks's, and a clear model theory, such as that provided by generalized circumscription, to explain and justify the whole process.

References

Hanks1985.
: Steve Hanks and Drew McDermott, "Temporal Reasoning and Default Logics," YALEU/CSD/RR #430, Yale University, Department of Computer Science, Oct 1985.

Lifschitz1985.
: Vladimir Lifschitz, "Some Results on Circumscription," in *Proceedings from the Non-Monotonic Reasoning Workshop*, AAAI, Oct 1985.

McCarthy1985.
: John McCarthy, "Applications of Circumscription to Formalizing Common Sense Knowledge," in *Proceedings from the Non-Monotonic Reasoning Workshop*, AAAI, Oct 1985.

McCarthy1980.
: John McCarthy, "Circumscription -- A Form of Non-Monotonic Reasoning," *Artificial Intelligence*, vol. 13, no. 1, pp. 27-38, 1980.