Things That Change by Themselves

Vladimir Lifschitz
Arkady Rabinov

Computer Science Department
Stanford University
Stanford, California 94305

Abstract

Some fluents (for instance, *time)* change even after events that are not assumed to affect them (such as the action *wait).* We propose a formalization of the "commonsense law of inertia" in the situation calculus that allows us to describe such fluents.

1 Introduction

The frame problem in Artificial Intelligence is the problem of specifying formally "what doesn't change when an event occurs" in such a way that the formal system "be ready to accept descriptions of new kinds of events and new kinds of fluents whose values are in general not affected by events whose descriptions don't mention them" [McCarthy, 1987].p The assertion that the value of a fluent / does not change when an action *a* is performed in a situation *s* can be expressed by the equation

$$value(f, result(a, s)) = value(f, s). \tag{1}$$

The question is how to express that this is true "in general," with the exceptions corresponding to the cases when other axioms postulate a new value for /. The default principle according to which the values of fluents are presumed to remain unchanged is sometimes called the "commonsense law of inertia." Formalizing this principle is considered one of the central problems in formal nonmonotonic reasoning.

The proposal of [McCarthy, 1986] was to restrict (1) to the t r i p: (f, a, s) a t do not satisfy an "abnormality" predicate ab:

$$\neg ab(f, a, s) \supset [value(f, result(a, s)) = value(f, s)]. \tag{2}$$

Circumscription is used to ensure that the extent of *ab* is minimal. Some difficulties were uncovered in [Hanks and McDermott, 1986], Section 3, and [McCarthy, 1986], Section 12. Several modifications that fix these problems have been found; for references and a critical discussion, see [Hanks and McDermott, 1987].

One of the proposed solutions [Lifschitz, 1987] describes the effects of actions using the predicate *causes.* For instance, the effect of moving a block *b* to a location / can be expressed by the axiom

$$causes(move(b, l), at(b, l), true).$$

The commonsense law of inertia can be postulated then in the following form:

$$\neg \exists v\ causes(a, f, v) \supset value(f, result(a, s)) = value(f, s). \tag{3}$$

The predicate *causes* is circumscribed; as a result, the antecedent of (3) is "generally true," and (3) can serve as the desired weakening of (1).

Axiom (3) asserts that the value of a fluent does not change after an action is p e r f o1 *unless h a t action causes the fluent to take on some value.*[2] In this paper we argue that, in many domains, this interpretation of "commonsense interia" is inappropriate. There are fluents that may change no matter what action is performed. The simplest example is the integer-valued fluent *time,* characterized by the equation

$$value(time, result(a, s)) = value(time, s) + 1.$$

The value of *time* in the situation *result (wait, s)* is different from its value in the situation *s* in spite of the fact that *wait* is not assumed to cause any changes whatsoever. We call fluents that can change even after an action like *wait* "dynamic."

We discuss here an alternative, more flexible description of commonsense inertia, that allows us to formalize reasoning about dynamic fluents. This description uses a new binary predicate *nonin*$(f, a)$the fluent / is noninertial relative to the action a"). We will write the commonsense law of inertia as follows:

$$\neg noninertial(f, a) \supset value(f, result(a, s)) = value(f, s). \tag{4}$$

The predicate *noninertial* will be circumscribed, like *ab* ' " < 2) -

It is clear that (4) is quite close to (2); the only difference (other than the more suggestive name of the predicate) is that in (4) the antecedent loses its situation argument. By changing the original formulation of

*This research was partially supported by DARPA under Contract NOO039-84-C-O211.

We use the terminology and notation of the situation calculus [McCarthy and Hayes, 1969].

The formulation proposed in [Haugh, 1987] is based essentially on the same assumption.

the Hanks-McDermott counterexample in this way, with few additional modifications, we get a particularly simple theory that eliminates all unwanted models. (In fact, it is strange that this method was not proposed among the first responses to the "Yale shooting" challenge.) The original form (2) can be thought of as a formalization of the assertion:

Normally, $value(f, result(a, s)) = value(f, s)$.

Formula (4) says something slightly different:

Normally, $\forall s[value(f, result(a, s)) = value(f, s)]$.

In other words, the *existence of a situation* in which the execution of a changes the value of / is considered exceptional.

It should be pointed out, however, that the new method, like the causality-based formalisms, does not address the problem of "ramifications," or indirect effects of an action.[3]

First we discuss the use of *noninertial* in the shooting example (Sections 2, 3), and then apply the method to some examples of dynamic fluents (Sections 4, 5).

2 The Hanks-McDermott Example

We assume that the reader is familiar with [Hanks and McDermott, 1986] or with another exposition of that counterexample. The language we use is essentially the same as in the original formulation, except that the ternary predicate *ab* is replaced by the binary predicate *noninertial.* The fluent constants are *loaded* and *alive;* the action constants are *load, wait* and *shoot;* the only situation constant is 50.

Since all fluents in this example are propositional, it is convenient to use the predicate *holds* instead of the function *value.* Accordingly, axiom (4) takes the form

$$\neg noninertial(f, a) \supset holds(f, result(a, s)) \equiv holds(f, s). \quad (5)$$

In addition to the law of inertia (5), the axiom set includes the following formulas, describing the effects of actions:

$$holds(loaded, result(load, s)), \quad (6)$$

$$noninertial(loaded, load), \quad (7)$$

$$\neg holds(loaded, result(shoot, s)), \quad (8)$$

$$noninertial(loaded, shoot), \quad (9)$$

$$holds(loaded, s) \supset \neg holds(alive, result(shoot, s)), \quad (10)$$

$$noninertial(alive, shoot), \quad (11)$$

$$\neg holds(loaded, s) \supset holds(alive, result(shoot, s)) \equiv holds(alive, s). \quad (12)$$

The main axioms are (6), (8) and (10). Each of these axioms represents a case when the law of inertia is cancelled—this is expressed by axioms (7), (9) and (11). Since axiom (10) has a precondition, we need to explain separately what happens when the precondition is not satisfied. This is done in axiom (12).

We also have the usual initial conditions:

$$holds(loaded, S0), \quad (13)$$

$$holds(alive, S0). \quad (14)$$

(Actually, adding these particular initial conditions makes axiom (7) redundant: It follows from axioms (5), (6) and (13).)

Formulas (5)-(14) are all the axioms we need, if we restrict attention to *term* models, i.e., to the models in which every element of the universe is represented by exactly one ground term of the corresponding sort. Such a model is determined, up to an isomorphism, by the set of ground atoms that are true in it.

Let A_1 be the axiom set (5)—(14). By a *minimal* model we understand a model in which the predicate *noninertial* is circumscribed, with *holds* varied.

Proposition 1. *The axiom set A_1 has a unique minimal term model. This model satisfies the condition*

$$noninertial(f, a) \equiv [(f = loaded \wedge a = load) \vee (f = loaded \wedge a = shoot) \vee (f = alive \wedge a = shoot)]. \quad (15)$$

Proof. Consider the structure M whose universe consists of the ground terms of the language, in which *holds* has its intended meaning, and *noninertial* is defined by (15). It is clear that M is a model of A_1. Axioms (7), (9) and (11) show, first, that M is minimal, and, second, that any minimal model satisfies (15) also. Furthermore, it is easy to check by induction that formulas (5), (6), (8), (10), (12)-(15) completely determine the extent of *holds* in any term model. Consequently, no term model other than M is minimal.

3 The Qualification Problem

The formalization given above does not address the qualification problem—the problem of leaving the lists of preconditions "open," so that it would be possible to incorporate new preconditions by adding new axioms. According to axiom (6), for example, loading has no preconditions; it always gives the desired effect. We may wish to make the axiomatization slightly more realistic, and formalize the fact that one cannot load the gun if it is locked in a safe, or if bullets are unavailable. It is impossible to do that by simply adding axioms; we would have to replace (6) by a weaker axiom, with the preconditions listed in the antecedent. Similarly, *loaded* is the only precondition included in (10); should we decide to incorporate other preconditions, it will be necessary to change that axiom.

An action can be unsuccessful in two different ways: It can be physically impossible, or it can merely fail to produce a particular effect [Pednault, 1988], [Gelfond *et al.*, 1989]. In this paper, we assume for simplicity that any action is physically possible in any situation,[4]

[3] Important ideas related to the ramification problem are proposed in the forthcoming paper [Baker, 1989].

[4] Formalizing actions that can be physically impossible is discussed in [Gelfond *et al.*, 1989].

and restrict our attention to "weak" preconditions—preconditions for particular effects. In Section 2, for instance, *loaded* is treated as a precondition for the success of *shoot* in the weak sense; if the gun is not loaded in a situation s, then the expression *result(shoot, s)* represents a physically meaningful situation, but, according to (12), the value of *alive* in that situation is the same as its value in the situation *s.*

In this section we describe an enhancement of *Al* that provides an improved treatment of weak preconditions. The assumption that *loaded* should hold in order for *shoot* to affect *alive* will be represented by the axiom

$$precond(loaded, shoot, alive), \tag{16}$$

where *precond* is a new predicate constant.[5] This predicate will be circumscribed along with *noninertial.* Its main property is expressed by the axiom:

$$precond(p,a,f) \wedge \neg holds(p,s) \supset value(f, result(a,s)) = value(f,s),$$

where *p* is a variable for propositional fluents. If / ranges over propositional fluents also, then we can write instead:

$$precond(p,a,f) \wedge \neg holds(p,s) \supset holds(f, result(a,s)) \equiv holds(f,s). \tag{17}$$

Notice that when the shooting example is reformulated in this way, axiom (12) will no longer be necessary, because it follows from (16) and (17). Axiom (17) has the same consequent as the commonsense law of inertia (5). It can be viewed as the formalization of an aspect of inertia not captured in (5).

An action *a* succeeds in affecting the value of / if all preconditions for that are satisfied; accordingly, we introduce the following abbreviation:

$$succeeds(a,f,s) \equiv \forall p[precond(p,a,f) \supset holds(p,s)].$$

The qualification problem can be solved by including a *succeeds* assumption in the antecedent of each axiom describing the effect of an action. For instance, axioms (6), (8) and (10) will be replaced by:

$$succeeds(load, loaded, s) \supset holds(loaded, result(load, s)), \tag{18}$$

$$succeeds(shoot, loaded, s) \supset \neg holds(loaded, result(shoot, s)), \tag{19}$$

$$succeeds(shoot, alive, s) \supset \neg holds(alive, result(shoot, 5)). \tag{20}$$

Axiom (17) can be rewritten as

$$\exists p[precond(p,a,f) \wedge \neg holds(p,s)] \supset holds(f, result(a,s)) \equiv holds(f,s)$$

or

$$\neg succeeds(a,f,s) \supset holds(f, result(a,s)) \equiv holds(f,s).$$

This is similar to the treatment of preconditions in [Lifschitz, 1987], except that *precond* had only two arguments there. The idea of enhancing *precond* in this way was suggested to us by Michael Gelfond and Michael Georgeff.

This observation shows that (17) is complementary to such axioms as (18)-(20): It shows how to determine the new value of a fluent when the *succeeds* condition is violated.

The new formulation of the shooting example includes the following postulates. General axioms: (5), (17). Effects of actions: (18)-(20), (7), (9), (11). Preconditions: (16). Initial conditions: (13), (14). We denote this axiom set by *A2.* By a *minimal* model we understand now a model in which *noninertial* and *precond* are circumscribed in parallel, with *holds* varied.

Proposition 2. *Tiie axiom set A2 has a unique minimal term model This model satisfies (15) and*

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

The proof is completely analogous to the proof of Proposition 1.

4 Dynamic Fluents

Recall that a "dynamic" fluent is, intuitively, a fluent that may change its value even after an action that is not assumed to have any causal effects, like *wait* The example given in the introduction is *time.*

Here is a more interesting example. Consider the process of filling a pool with water, regulated by opening and closing a valve [Hendrix, 1973]. The state of the system can be described by two numeric (for simplicity, integer-valued) fluents: *volume* (the volume of water in the pool, in cubic meters) and *inflow* (the current inflow of water, in cubic meters per minute). The following actions are available:

1. *setvolume n:* Bring the volume of water in the pool to n, and close the valve.
2. *setinflow* n: Turn the valve to bring the inflow of water to n.
3. *wait:* Do nothing.

We assume that every action other than *setvolume* is practically instantaneous and is followed by a 1 minute wait period. In particular, *wait* means "wait for 1 minute." Then the relation between *volume* and *inflow* can be expressed by the axiom

$$\neg\exists n(a = setvolume\ n) \supset value(volume, result(a,s)) = value(volume, s) + value(inflow, result(a,s)). \tag{21}$$

The fluent *volume* is dynamic: Its value may change even if the action being performed is *wait.*

The effects of *setinflow* and *setvolume* can be described by the axioms:

$$succeeds(setinflow\ n, inflow, s) \supset value(inflow, result(setinflow\ n, s)) = n,$$

$$succeeds(setvolume\ n, volume, s) \supset value(volume, result(setvolume\ n, s)) = n,$$

$$succeeds(setvolume\ n, inflow, s) \supset value(inflow, result(setvolume\ n, s)) = 0.$$

Consider now a situation s such that

$$\text{value (inflow, s)} \neq 0.$$

What can we say about the values of inflow and volume in the situation result(wait, s)? Axiom (21) implies that at least one of these fluents will have a value different from its value in the situation s. It follows that there is tension between minimizing nonineriial(inflow, wait) on the one hand, and nonineriial(volume, wait) on the other. The axioms given above have unintended minimal models.

We can eliminate these models by adding an axiom which says that the fluent volume is dynamic. The general concept of a dynamic fluent is defined as follows:

$$\text{dynamic } f \equiv \forall a \text{ nonineriial}(f, a).$$

Then the additional axiom needed in the flowing water example can be written as

dynamic volume.

5 Momentary Fluents

In this section we discuss a special case of dynamic fluents—the propositional fluents that have the tendency to become false. Consider the following example.

Striking one object against another produces noise:

$$\text{succeeds (strike , noise , s)} \supset \text{holds (noise , result (strike , s))}.$$

In the absence of preconditions for strike , the antecedent of this axiom is identically true, and we get:

holds (noise , result (strike ,5)).

But the law of inertia implies then that the noise will continue for a long time:

$$\text{holds (noise , result(wait, result(strike , 5)))}, \quad (22)$$

$$\text{holds (noise , result (wait, result (wait, result (strike , s))))}, \quad (23)$$

etc. We need to postulate that the noise is momentary, rather than continuous, that it comes to an end by itself.

The basic property of "momentary" propositional fluents is that, by default, they take on the value false:

$$\text{momentary } f \wedge \neg ab(f, s) \supset \text{-\^{}holds}(f, s).$$

Moreover, momentary fluents are dynamic:

$$\text{momentary } f \supset \text{dynamic } f.$$

Both momentary and the abnormality predicate ab are circumscribed.[6]

The additional postulate needed in the noise example can be written as

momentary noise.

These axioms allow us to prove the negations of (22) and (23).

Acknowledgements

We are grateful to Andrew Baker, Michael Gelfond, Matthew Ginsberg, John McCarthy, Paul Morris, Karen Myers and Yoav Shoham for valuable comments and criticisms.

The predicate ab should be circumscribed at a lower priority than the predicates that have no situation arguments (nonineriial, precond, momentary).

References

[Baker, 1989] Andrew Baker. A simple solution to the Yale Shooting problem. In Proc. KR'89, 1989. To appear.

[Gelfond et ai, 1989] Michael Gelfond, Vladimir Lifschitz, and Arkady Rabinov. A theory of concurrent actions. In preparation, 1989.

[Hanks and McDermott, 1986] Steve Hanks and Drew McDermott. Default reasoning, nonmonotonic logics, and the frame problem. In Proc. AAAI-86, pages 328-333, 1986.

[Hanks and McDermott, 1987] Steve Hanks and Drew McDermott. Nonmonotonic logic and temporal projection. Artificial Intelligence, 33(3):379-412, 1987.

[Haugh, 1987] Brian Haugh. Simple casual minimization for temporal persistence and projection. In Proc. AAAL87, pages 218-223, 1987.

Hendrix, 1973] Gary Hendrix. Modeling simultaneous actions and continuous processes. Artificial Intelligence, 4:145-180, 1973.

[Lifschitz, 1987] Vladimir Lifschitz. Formal theories of action. In The Frame Problem, Proc. of the 1987 Workshop, pages 35-37, 1987.

McCarthy and Hayes, 1969] John McCarthy and Patrick Hayes. Some philosophical problems from the standpoint of artificial intelligence. In B. Meltzer and D. Michie, editors, Machine Intelligence, pages 463-502, Edinburgh, 1969. Edinburgh University Press.

[McCarthy, 1986] John McCarthy. Applications of circumscription to formalizing common-sense knowledge. Artificial Intelligence, 26(3):89-118, 1986.

[McCarthy, 1987] John McCarthy. The frame problem today. In The Frame Problem, Proc. of the 1987 Workshop, page 3, 1987.

[Pednault, 1988] Edwin Pednault. Extending conventional planning techniques to handle actions with context-dependent effects. In Proc. AAAI-88, pages 55-59, 1988.