In *Proceedings of the 13th International Joint Conference on Artificial Intelligence (IJCAI 93)*, pages 712-718, Morgan Kaufmann, 1993.

Soundness and Completeness Theorems for Three Formalizations of Action*

G. Neelakantan Kartha
Department of Computer Sciences
University of Texas at Austin
Austin, 78712
email: kartha@cs.utexas.edu

Abstract

Instead of trying to compare methodologies for reasoning about action on the basis of specific examples, we focus here on a general class of problems, expressible in a declarative language $\mathcal{A}$. We propose three translations, P, R and B from $\mathcal{A}$, representing respectively the first order methods of reasoning about action proposed by Pednault and Reiter and the circumscriptive approach of Baker. We then prove the soundness and completeness of these translations relative to the semantics of $\mathcal{A}$. This lets us compare these three methods in a mathematically precise fashion. Moreover, we apply the methods of Baker in a general setting and prove a theorem which shows that if the domain of interest can be expressed in $\mathcal{A}$, circumscription yields results which are intuitively expected.

1 Introduction

Most of the past work in reasoning about action has been done using nonmonotonic logics. Several nonmonotonic formalisms have appeared in the literature [McCarthy, 1980; Reiter, 1980; McDermott and Doyle, 1980; Moore, 1985] and a number of technical results about these have been obtained. Moreover, there has been a flurry of formalisms which are variations of the above which were designed to handle instances in which the original formalisms yield counterintuitive results[1]. However, the focus of most of this work has been on formalizing specific problems. This makes it very difficult to compare the different approaches as to the range of applicability of each.

There also have been attempts recently to reason about action in first order logic [Pednault, 1989; Schubert, 1990; Reiter, 1991]. Again, most of the work has been in terms of specific examples.

In this paper, we focus on a general class of problems, which are expressible in a simple declarative language $\mathcal{A}$, introduced in [Gelfond and Lifschitz, 1992]. In that paper, the authors also define a translation from $\mathcal{A}$ into the language of extended logic programs and prove its soundness. Our work is similar in that we provide simple translations from $\mathcal{A}$ into three different methodologies which have been proposed in the literature, but we prove both the soundness and completeness of the three translations relative to the semantics of $\mathcal{A}$. Our motivation is to precisely characterize the strengths and limitations of various formalizations of action and in that, our work is is related to that in [Lifschitz, 1991; Lin and Shoham, 1991].

The first two translations we provide use the first order methods for reasoning about action, suggested by Pednault [Pednault, 1989] and Reiter [Reiter, 1991]. The third one is the nonmonotonic formalism of circumscription. The soundness and completeness of the three translations is interesting since it lets us compare these three methodologies in a mathematically precise fashion: we get the result that they are equivalent in the sense that they all faithfully capture the semantics of $\mathcal{A}$. Many problems which have been discussed extensively in the literature, such as the Yale shooting problem, are special cases of the formalization presented here. Thus, our work yields three classes of successful formalizations of wide applicability. Unlike other approaches like that in [Shoham, 1988], in addition to reasoning forward in time, the formalizations support reasoning backwards in time as well. They also let us deal with incompletely specified initial situations.

The technically most difficult part of this paper is the formalization based on circumscription. It is based on the following idea of Baker [Baker, 1991]: When the abnormality predicate Ab is circumscribed, the *Result* function and the situation constant corresponding to the initial situation, S_0 are allowed to vary. A theorem is proved which shows that the circumscription yields results which are intuitively expected. This result shows how to apply the methods in [Baker, 1991] in a general setting. The theorem is proved in a fashion which makes it plausible that the methods of proof can be employed in other situations as well.

In the next section, we describe the language $\mathcal{A}$ which is used to formalize the domain of actions. Section 3 describes Pednault's approach and a translation of actions described in $\mathcal{A}$ which uses his scheme. Section 4 presents

*This work was partially supported by National Science Foundation under grant IRI-9101078.

[1]The most notorious among such instances is the Yale shooting problem [Hanks and McDermott, 1987], for which a large number of solutions have been proposed.

a translation based on Reiter's approach. In Section 5, we provide a translation which uses the circumscriptive approach. Section 6 contains concluding remarks. The proofs of all the theorems are omitted.

2 The Language $\mathcal{A}$

In this section, we describe the languge $\mathcal{A}$ introduced in [Gelfond and Lifschitz, 1992]. The reader who is interested in more examples and motivation is referred to that paper.

Consider two disjoint nonempty sets of symbols, called *fluent names* and *action names*. A *fluent expression* is a fluent name possibly preceded by $\neg$. A *v-proposition* specifies the value of a fluent after performing zero or more actions from the initial situation. It is of the form

$$F \textbf{ after } A_1; \ldots ; A_m \tag{1}$$

where F is a fluent expression, and $A_1,\ldots,A_m$ $(m \geq 0)$ are action names. If $m = 0$, we will write (1) as

$$\textbf{initially } F.$$

An *e-proposition* describes the effect of an action on a fluent and is of the form

$$A \textbf{ causes } F \textbf{ if } P_1, \ldots , P_n, \tag{2}$$

where A is an action name, and each of $F, P_1, \ldots , P_n$ $(n \geq 0)$ is a fluent expression. Let us denote by $|F|$ the fluent name occurring in the fluent expression F. If $|F|$ occurs unnegated (negated) in (2), we call it a *positive (negative) e-proposition* and say that A *affects* $|F|$ *positively (negatively)*. We also say that this e-proposition *describes the effect of* A *on* F, and that $P_1, \ldots , P_n$ are its *preconditions*. If $n = 0$, we will drop **if** and write simply

$$A \textbf{ causes } F.$$

A *proposition* is a v-proposition or an e-proposition. A *domain description*, or simply *domain*, is a set of propositions (not necessarily finite).

Example 1. The Blocks World domain D_B, motivated by problem $D4$ from [Lifschitz, 1989], is defined as follows. The fluent names are *Ontable* and *Inhand*; the action names are *Putdown*, *Pickup*, *Lower* and *Wait*. The propositions of D_B are:

initially $\neg$*Ontable*,
Ontable **after** *Wait;Putdown*,
Putdown **causes** *Ontable* **if** *Inhand*,
Putdown **causes** $\neg$*Inhand* **if** *Inhand*,
Pickup **causes** *Inhand* **if** *Ontable*,
Pickup **causes** $\neg$*Ontable* **if** *Ontable*,
Lower **causes** *Ontable* **if** *Inhand*.

The following definitions will be useful in describing the semantics of $\mathcal{A}$.

A *state* is a set of fluent names. Given a fluent name F and a state σ, we say that F *holds* in σ if $F \in \sigma$; $\neg F$ *holds* in σ if $F \notin \sigma$. A *transition function* is a mapping Φ of the set of pairs (A, σ), where A is an action name and σ is a state, into the set of states. A *structure* is a pair (σ_0, Φ), where σ_0 is a state (the *initial state* of the structure), and Φ is a transition function.

For any structure M and any action names $A_1, \ldots , A_m$, by $M^{A_1;\ldots;A_m}$ we denote the state

$$\Phi(A_m, \Phi(A_{m-1}, \ldots , \Phi(A_1, \sigma_0) \ldots)),$$

where Φ and σ_0 are the transition function and the initial state of M. We say that a v-proposition (1) is *true* in a structure M if F holds in the state $M^{A_1;\ldots;A_m}$, and that it is *false* otherwise.

A structure (σ_0, Φ) is a *model* of a domain description D if every v-proposition from D is true in (σ_0, Φ) and for every action name A, every fluent name F, and every state σ, the following two conditions are satisfied:

(i) if D includes an e-proposition describing the effect of A on F $(\neg F)$ whose preconditions hold in σ, then F $(\neg F)$ holds in the state $\Phi(A, \sigma)$.

(ii) otherwise, F holds in the state $\Phi(A, \sigma)$ iff F holds in the state σ.

Note that there can be at most one transition function Φ, satisfying conditions (i) and (ii). Hence different models of the same domain differ only by their initial states.

A v-proposition is *entailed* by a domain description D if it is true in every model of D.

A domain description is *consistent* if it has a model.

Example 1 (continued). The domain description D_B is consistent. Its only model is defined by the equations

$$\sigma_0 = \{Inhand\},$$

$$\Phi(Putdown, \sigma) = \begin{cases} \sigma \cup \{Ontable\} \setminus \{Inhand\}, & \text{if } Inhand \in \sigma; \\ \sigma, & \text{otherwise;} \end{cases}$$

$$\Phi(Pickup, \sigma) = \begin{cases} \sigma \cup \{Inhand\} \setminus \{Ontable\}, & \text{if } Ontable \in \sigma; \\ \sigma, & \text{otherwise;} \end{cases}$$

$$\Phi(Lower, \sigma) = \begin{cases} \sigma \cup \{Ontable\}, & \text{if } Inhand \in \sigma; \\ \sigma, & \text{otherwise;} \end{cases}$$

$$\Phi(Wait, \sigma) = \sigma.$$

It is easy to see, for instance, that

initially *Inhand*,
Ontable **after** *Lower; Wait*,
$\neg$*Inhand* **after** *Lower; Wait;Putdown*,

is entailed by D_B.

3 Translation into Pednault's Scheme

3.1 Review of Pednault's Proposal

In this section, we review Pednault's proposal for generating the frame axioms. We will conform for the most part to the notation introduced in [Reiter, 1991].

Pednault assumes that corresponding to each fixed fluent F and each fixed action A there are two effect axioms, a positive effect axiom of the form[2]:

$$\pi_A(s) \wedge \varepsilon_F^+(s) \supset Holds(F, Result(A, s)), \qquad (3)$$

and a negative effect axiom of the form

$$\pi_A(s) \wedge \varepsilon_F^-(s) \supset \neg Holds(F, Result(A, s)). \qquad (4)$$

Here, $\pi_A(s)$ denotes the *action preconditions* of the action A. These are the prerequisites that must be satisfied for the execution of action A. They depend only on A. $\varepsilon_F^+(s)$ ($\varepsilon_F^-(s)$) denotes the *positive (negative) fluent preconditions* under which action A, if performed, will lead to F becoming true (false) in the resulting situation $Result(A, s)$. We assume that in (3) and (4), s is universally quantified.

Pednault observes that he can obtain the frame axioms from the effect axioms systematically if a completeness assumption for the fluent preconditions is made. The assumption is that the positive (negative) fluent preconditions specify *all* the conditions under which an action if performed will lead to the truth (falsity) of the fluent in the resulting situation. Using this assumption, we can say that if in situation s, we have $\pi_A(s)$, then the only way that F can change from false in situation s to true in situation $Result(A, s)$ is if $\varepsilon_F^+(s)$ were true. This can be formalized as

$$\pi_A(s) \wedge \neg Holds(F, s) \wedge Holds(F, Result(A, s)) \supset \varepsilon_F^+(s).$$

Rewriting the above as

$$\pi_A(s) \wedge \neg Holds(F, s) \wedge \neg \varepsilon_F^+(s) \supset \neg Holds(F, Result(A, s)),$$

we see that this is a frame axiom. Similarly, we can generate the frame axiom

$$\pi_A(s) \wedge Holds(F, s) \wedge \neg \varepsilon_F^-(s) \supset Holds(F, Result(A, s))$$

from (4).

3.2 The Translation $P(D)$

Let D be a finite domain. The following notation will be used: If $A_1, \ldots, A_m$ are action names, $[A_1; \ldots; A_m]$ stands for the ground term

$$Result(A_m, Result(A_{m-1}, \ldots, Result(A_1, S_0), \ldots)).$$

If t is a term,

$$Holds(\neg F, t) \text{ stands for } \neg Holds(F, t).$$

The translation of a v-proposition α of the form (1), denoted $P(\alpha)$, is

$$Holds(F, [A_1; \ldots; A_m]). \qquad (5)$$

Since corresponding to an F, A pair (where F is a fluent name and A is an action), Pednault has just two effect axioms, e-propositions are not translated individually. Rather the set of all e-propositions describing the effect of A on F or $\neg F$, $S_{F,A}$, is partitioned into two subsets $S_{F,A}^+$ and $S_{F,A}^-$. $S_{F,A}^+$ ($S_{F,A}^-$) consists of all the e-propositions describing the effect of A on F ($\neg F$). The e-propositions in each of these two sets are then translated as a group. To express this formally, define for any e-proposition E of the form (2),

$$pre_E(s) = \bigwedge_{i=1}^{n} Holds(P_i, s).$$

Also define,

$$precond_{F,A}^+(s) = \bigvee_{E \in S_{F,A}^+} pre_E(s),$$

$$precond_{F,A}^-(s) = \bigvee_{E \in S_{F,A}^-} pre_E(s).$$

Now, we can translate the collection of e-propositions in $S_{F,A}$ as

$$precond_{F,A}^+(s) \supset Holds(F, Result(A, s)), \qquad (6)$$

$$precond_{F,A}^-(s) \supset \neg Holds(F, Result(A, s)). \qquad (7)$$

Using the notation of the previous section we see that $\pi_A(s)$ can be identified with *true*, $\varepsilon_F^+(s)$ can be identified with $precond_{F,A}^+(s)$ and $\varepsilon_F^-(s)$ can be identified with $precond_{F,A}^-(s)$.

Let $\mathbf{F}_D$ and $\mathbf{A}_D$ be, respectively, the set of all the fluent names and action names mentioned in the domain description D. Then we will also add the following axioms:

1. The domain closure axiom (DCA) for fluents:
$$\bigvee_{F \in \mathbf{F}_D} f = F.$$
2. The domain closure axiom (DCA) for actions:
$$\bigvee_{A \in \mathbf{A}_D} a = A.$$
3. The unique name assumption (UNA) axiom for fluents:
$$\bigwedge_{\substack{F_1, F_2 \in \mathbf{F}_D \\ F_1, F_2 \text{ distinct}}} F_1 \neq F_2.$$
4. The unique name assumption (UNA) axiom for actions:
$$\bigwedge_{\substack{A_1, A_2 \in \mathbf{A}_D \\ A_1, A_2 \text{ distinct}}} A_1 \neq A_2.$$

By $P(D)$, we denote the collection of axioms which consist of these four axioms, the axioms of the form (5), (6) and (7) and the frame axioms obtained from the effect axioms by the process outlined in the previous section. That is, we add for each fluent-action pair F, A, the two frame axioms

$$\neg Holds(F, s) \wedge \neg precond_{F,A}^+(s) \supset \neg Holds(F, Result(A, s)), \qquad (8)$$

[2] The form of the effect axioms in our review is more special than the ones actually used by Pednault: we deal with individual actions and fluents rather than families parameterized by x and y. We use such a simplification in our review of Reiter's proposal also.

$$Holds(F,s) \wedge \neg precond^-_{F,A}(s) \supset Holds(F, Result(A,s)). \quad (9)$$

Example 1 (continued). $P(D_B)$ consists of the following axioms 1-11 and sixteen frame axioms (including two trivial ones) of the form (8) and (9).

1. $\neg Holds(Ontable, S_0)$.
2. $Holds(Ontable, Result(Putdown, Result(Wait, S_0)))$
3. $Holds(Inhand, s) \supset$
   $Holds(Ontable, Result(Putdown, s))$.
4. $Holds(Inhand, s) \supset$
   $\neg Holds(Inhand, Result(Putdown, s))$.
5. $Holds(Ontable, s) \supset$
   $Holds(Inhand, Result(Pickup, s))$.
6. $Holds(Ontable, s) \supset$
   $\neg Holds(Ontable, Result(Pickup, s))$.
7. $Holds(Inhand, s) \supset$
   $Holds(Ontable, Result(Lower, s))$.
8. $f = Ontable \vee f = Inhand$.
9. $a = Putdown \vee a = Pickup \vee$
   $a = Lower \vee a = Wait$.
10. $Ontable \neq Inhand$.
11. $Putdown \neq Pickup \wedge Putdown \neq Lower \wedge$
    $Putdown \neq Wait \wedge Pickup \neq Lower \wedge$
    $Pickup \neq Wait \wedge Lower \neq Wait$.

3.3 Soundness and Completeness

We can relate the notion of entailment in the sense of first order logic and entailment relation for the language $\mathcal{A}$ using the following theorem.

Theorem 1 *Let D be any finite, consistent domain, and let $P(D)$ be defined as before. For any v-proposition α,*

$$P(D) \models P(\alpha) \iff D \text{ entails } \alpha.$$

Theorem 1 can be thought of as expressing the soundess and completeness of the translation P. Note that it gives a very strong characterization of what can be proved from $P(D)$. For instance, using this theorem, we can conclude in the case of Example 1 that $P(D) \models Holds(Inhand, S_0)$. Note that this corresponds to reasoning about the past. Similarly, we can conclude that $P(D) \models \neg Holds(Inhand, [Lower; Wait; Putdown])$. This corresponds to reasoning about the future, when the initial situation is incompletely specified. Domains in which each v-proposition has the form **initially** F, are just a simple special case of this theorem. Such domains correspond to temporal projection problems, which have been the focus of much attention in the literature.

In the subsequent sections, we shall propose similar translations from D into other domains. In each case, we will prove theorems of exactly the same form as Theorem 1, the only change being that instead of the translation P, we will have other translation schemes. Thus all the remarks of the previous paragraph apply to these translation schemes as well.

4 Translation into Reiter's Scheme

4.1 Review of Reiter's Proposal

Like Pednault, Reiter also has two effect axioms for each fluent F. But in these effect axioms, the actions are universally quantified (following the proposal by Schubert [Schubert, 1990]). So he calls them *general* effect axioms. They are of the form

$$Poss(a,s) \wedge \gamma^+_F(a,s) \supset Holds(F, Result(a,s)), \quad (10)$$

$$Poss(a,s) \wedge \gamma^-_F(a,s) \supset \neg Holds(F, Result(a,s)). \quad (11)$$

The predicate $Poss(a,s)$ intuitively means that a is possible in situation s. $Poss$ is defined by action precondition axioms of the form

$$\pi_A(s) \supset Poss(A,s)$$

for each action A, where $\pi_A(s)$ stands for the action A's preconditions.

Reiter also makes a completeness assumption, according to which (10) and (11) account for *all* the conditions under which action a can affect the value of F. As before, this completeness assumption leads to what Reiter calls *explanation closure axioms*. They are equivalent to frame axioms of the following form:

$$Poss(a,s) \ \wedge \ \neg Holds(F,s) \ \wedge \ \neg\gamma^+_F(a,s) \supset \neg Holds(F, Result(a,s)),$$

$$Poss(a,s) \ \wedge \ Holds(F,s) \ \wedge \ \neg\gamma^-_F(a,s) \supset Holds(F, Result(a,s)).$$

Comparing these frame axioms with the ones obtained by Pednault, we see that the reification of actions allows Reiter to express the frame axioms in a more compact way.

4.2 The Translation $R(D)$

The translation of the domain D into Reiter's scheme, $R(D)$, is very similar to the translation $P(D)$ into Pednault's scheme: the translation of v-propositions is exactly the same and the UNA and DCA for fluents and actions are included in the translation, as before. The only differences are in the translation of the e-propositions and in the form of the added frame axioms. These differences arise due to the fact that here, actions in the general effect statements are universally quantified.

With a fixed fluent name F, we associate two sets: the set A^+_F of action names which affect it positively and the set A^-_F of those which affect it negatively. Note that it is possible for a particular action name to be in both these sets or to be in neither of them.

Now we can give the translation of e-propositions into Reiter's formalism as follows:

$$\bigvee_{A \in A^+_F} (a = A \ \wedge \ precond^+_{F,A}(s)) \supset Holds(F, Result(a,s)), \quad (12)$$

$$\bigvee_{A \in A^-_F} (a = A \ \wedge \ precond^-_{F,A}(s)) \supset \neg Holds(F, Result(a,s)). \quad (13)$$

In comparison to the notation used in Section 4.1, $Poss(a,s)$ corresponds to $true$, $\gamma_F^+(a,s)$ to $\bigvee_{A \in A_F^+}(a = A \wedge precond_{F,A}^+(s))$, and $\gamma_F^-(a,s)$ to $\bigvee_{A \in A_F^-}(a = A \wedge precond_{F,A}^-(s))$.

$R(D)$ consists of the translation of all the v-propositions and e-propositions, that is, all sentences of the form (5), (12) and (13), the UNA and DCA for fluents and actions, and the frame axioms generated by the process outlined in the previous section.

We can now prove a theorem about $R(D)$ exactly similar to Theorem 1.

Theorem 2 *Let D be any finite, consistent domain and let $R(D)$ be defined as before. For any v-proposition α,*

$$R(D) \models R(\alpha) \iff D \text{ entails } \alpha.$$

In [Elkan, 1992], Elkan proposed a similar scheme for reasoning about action. We can easily prove a theorem of the same form as Theorem 2 for this scheme also.

5 Translation into Circumscriptive Scheme

In this section we will propose a translation into a circumscriptive scheme, based on the method of [Baker, 1991] and state again the soundness and completeness of the translation. This result is of independent interest, since it shows that circumscription can handle reasoning about actions elegantly.

5.1 Review of Circumscription

Circumscription was introduced by McCarthy [McCarthy, 1980; McCarthy, 1986] as a formalism for nonmonotonic reasoning. Intuitively, circumscribing a predicate in a sentence means assuming that the extent of the predicate is as small as possible. We formally define this as follows:

For any predicate symbols P, Q of the same arity, let $P = Q$ stand for $\forall x(P(x) \equiv Q(x))$ and $P \leq Q$ stand for $\forall x(P(x) \supset Q(x))$. Let Z stand for the tuple $Z_1, Z_2, \ldots Z_m$ of object, function, and/or predicate constants. Let $A(P, Z)$ be a sentence containing Z and a predicate constant P.

The *circumscription of P in A with Z varied*, denoted by $CIRC(A; P; Z)$ is the sentence

$$A(P, Z) \wedge \neg \exists p, z[A(p, z) \wedge p < P].$$

Here p is a predicate variable of the same arity as P, z stands for an m-tuple of variables which matches the m-tuple Z in arity and type, and $p < P$ stands for $(p \leq P) \wedge \neg(p = P)$.

In reasoning about action using circumscription, we first introduce an axiom for the "commonsense law of inertia"

$$\neg Ab(f, a, s) \supset (Holds(f, s) \equiv Holds(f, Result(a, s))). \tag{14}$$

This says that the values of fluents normally persist, after an action is performed. The predicate $Ab(f, a, s)$ intuitively means the following: f is abnormal with respect to a in situation s.

Traditionally [McCarthy, 1986], the formula B' consisting of the conjunction of the universal closures of all the domain axioms is circumscribed with respect to Ab while varying the predicate $Holds$. But this does not work, as noted by Hanks and McDermott in [Hanks and McDermott, 1987]. A simple solution, proposed by Baker in [Baker, 1991] is to vary the function $Result$ instead. For this method to work correctly, we need an "existence of situation" axiom, which intuitively says that, corresponding to any subset of fluents, there is a situation in which exactly these hold. This can be formulated as follows: Let σ range over states, that is, subsets of the set of all fluents. Define

$$St_\sigma(s) = \bigwedge_{F \in \sigma} Holds(F, s) \wedge \bigwedge_{F \notin \sigma} \neg Holds(F, s).$$

Now, the existence of situations axiom can be written as

$$\bigwedge_\sigma \exists s St_\sigma(s). \tag{15}$$

Baker [Baker, 1991] suggests ways to write this axiom in a more compact form.

5.2 The Translation $B(D)$

Let us first define an auxiliary translation $T_B(D)$ as a preliminary step before defining $B(D)$. $T_B(D)$ is similar to $P(D)$ in that the translations of the v-propositions are exactly the same as in $P(D)$, and in that $T_B(D)$ includes the UNA and DCA for actions and the DCA for fluents. Each e-proposition is translated individually in $T_B(D)$. That is, the translation of an e-proposition E of the form (2), which describes the effect of A on F is simply

$$Holds(P_1, s) \wedge \ldots \wedge Holds(P_n, s) \supset Holds(F, Result(A, s)). \tag{16}$$

In addition to the above, $T_B(D)$ contains axioms (14) and (15). (Note that we do not need to add the UNA for fluents, since it follows from (15).)

Let B' be the conjunction of the universal closures of the axioms of $T_B(D)$. Let us denote the formula $CIRC(B'; Ab; Result, S_0)$ by $B(D)$. The following theorem about $B(D)$ can now be proved:

Theorem 3 *Let D be any finite, consistent domain and let $B(D)$ be defined as before. For any v-proposition α,*

$$B(D) \models B(\alpha) \iff D \text{ entails } \alpha.$$

Note that this theorem is of the same form as Theorems 1 and 2. Using this theorem, we can prove a "restricted monotonicity" result [Lifschitz, 1993] for Baker's approach to formalizing action.

The proof of Theorem 3 is based on the following lemma about the result of circumscribing Ab.

The Main Lemma *Let D be a finite, consistent domain. Then $B(D)$ is equivalent to the conjunction of B' and*

$$\forall f, a, s\,[Ab(f, a, s) \equiv \bigvee_{F \in \mathbf{F}_D, A \in \mathbf{A}_D}(f = F \wedge a = A \wedge \mathit{affected}_{F,A}(s))]$$

where

$$affected_{F,A}(s) = (precond^{+}_{F,A}(s) \wedge \neg Holds(F,s)) \vee \\ (precond^{-}_{F,A}(s) \wedge Holds(F,s))$$

Note that this lemma characterizes Ab completely.

6 Summary and Discussion

In this paper, we have shown that the methods in [Pednault, 1989; Reiter, 1991; Baker, 1991] are applicable to a large class of problems. These methods support reasoning forward as well as backward in time. We have also compared these methods in a mathematically precise fashion. Moreover, we have shown how to apply the methods in [Baker, 1991] to reason about actions using circumscription, in a general setting. The results of this paper, in combination with the soundness theorem for the translation into logic programming from [Gelfond and Lifschitz, 1992], suggest that the computational mechanism of logic programming can be used for implementing the theories of action proposed by Pednault, Reiter and Baker.

One major assumption we make is that the problem domain under consideration can be formalized in $\mathcal{A}$. As is pointed out in [Gelfond and Lifschitz, 1992], $\mathcal{A}$ is rather limited in its expressive power. For instance, we assume here that the fluents are all independent. This means that we are spared the task of coping with the "ramification problem" . Another issue we do not address here is the qualification problem.

It seems, however, that the methods outlined in this paper will prove capable of being extended to more complex domains. This is the topic of our ongoing research.

Acknowledgements

I wish to thank Vladimir Lifschitz for discussing this work with me and for his help with the contents and presentation of this paper. I also wish to thank Michael Gelfond, Norm McCain, Jeff Rickel and Thomas Woo for comments on an earlier draft.

References

[Baker, 1991] Andrew Baker. Nonmonotonic reasoning in the framework of situation calculus. *Artificial Intelligence*, 49:5–23, 1991.

[Elkan, 1992] Charles Elkan. Reasoning about action in first-order logic. In *Proceedings of the Ninth Biennial Conference of the Canadian Society for Computational Studies of Intelligence (CSCSI'92), Vancouver, Canada.* Morgan Kaufmann, 1992.

[Gelfond and Lifschitz, 1992] Michael Gelfond and Vladimir Lifschitz. Describing action and change by logic programs. In *Proc. Joint Int'l Conf. and Symp. on Logic Programming*, 1992.

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

[Lifschitz, 1989] Vladimir Lifschitz. Benchmark problems for formal non-monotonic reasoning, version 2.00. In Michael Reinfrank, Johan de Kleer, Matthew Ginsberg, and Erik Sandewall, editors, *Non-Monotonic Reasoning: 2nd International Workshop (Lecture Notes in Artificial Intelligence 346)*, pages 202–219. Springer-Verlag, 1989.

[Lifschitz, 1991] Vladimir Lifschitz. Towards a metatheory of action. In James Allen, Richard Fikes, and Erik Sandewall, editors, *Proc. of the Second Int'l Conf. on Principles of Knowledge Representation and Reasoning*, 1991.

[Lifschitz, 1993] Vladimir Lifschitz. Restricted monotonicity. In *Proc. of the Eleventh National Conference of Artificial Intelligence*, 1993. To appear.

[Lin and Shoham, 1991] Fangzhen Lin and Yoav Shoham. Provably correct theories of action. In *Proc. of the Ninth National Conference of Artificial Intelligence*, pages 349–354, 1991.

[McCarthy, 1980] John McCarthy. Circumscription—a form of non-monotonic reasoning. *Artificial Intelligence*, 13(1,2):27–39,171–172, 1980.

[McCarthy, 1986] John McCarthy. Applications of circumscription to formalizing common sense knowledge. *Artificial Intelligence*, 26(3):89–116, 1986.

[McDermott and Doyle, 1980] Drew McDermott and Jon Doyle. Nonmonotonic logic I. *Artificial Intelligence*, 13(1,2):41–72, 1980.

[Moore, 1985] Robert Moore. Semantical considerations on nonmonotonic logic. *Artificial Intelligence*, 25(1):75–94, 1985.

[Pednault, 1989] Edwin Pednault. ADL: Exploring the middle ground between STRIPS and the situation calculus. In Ronald Brachman, Hector Levesque, and Raymond Reiter, editors, *Proc. of the First Int'l Conf. on Principles of Knowledge Representation and Reasoning*, pages 324–332, 1989.

[Reiter, 1980] Raymond Reiter. A logic for default reasoning. *Artificial Intelligence*, 13(1,2):81–132, 1980.

[Reiter, 1991] Raymond Reiter. The frame problem in the situation calculus: A simple solution (sometimes) and a completeness result for goal regression. In Vladimir Lifschitz, editor, *Artificial Intelligence and Mathematical Theory of Computation*, pages 359–380. Academic Press, 1991.

[Schubert, 1990] L.K Schubert. Monotonic solution of the frame problem in the situation calculus: an efficient method for worlds with fully specified actions. In H.E Kyberg, R.P Loui, and G.N Carlson, editors, *Knowledge Representation and Defeasible Reasoning*, pages 23–67. Kluwer Academic Press, Boston, Mass., 1990.

[Shoham, 1988] Yoav Shoham. Chronological ignorance: experiments in nonmonotonic temporal reasoning. *Artificial Intelligence*, 36:279–331, 1988.