Explaining Explanation Closure

Patrick Doherty[1] and Witold Łukaszewicz[2] and Andrzej Szałas[2]

[1] Department of Computer and Information Science
Linköping University, S-58183 Linköping, Sweden
patdo@ida.liu.se
[2] Institute of Informatics
Warsaw University, 00-913 Warsaw 59, Poland
(witlu,szalas)@mimuw.edu.pl

Abstract. Recently, Haas, Schubert, and Reiter, have developed an alternative approach to the frame problem which is based on the idea of using *explanation closure axioms*. The claim is that there is a monotonic solution for characterizing nonchange in serial worlds with fully specified actions, where one can have both a succinct representation of frame axioms and an effective proof theory for the characterization. In the paper, we propose a circumscriptive version of explanation closure, PMON, that has an effective proof theory and works for both context dependent and nondeterministic actions. The approach retains representational succinctness and a large degree of elaboration tolerance, since the process of generating closure axioms is fully automated and is of no concern to the knowledge engineer. In addition, we argue that the monotonic/nonmonotonic dichotomy proposed by others is not as sharp as previously claimed and is not fully justified.

`Accepted for publication at the 9th Int'l Symposium on Methodologies for Intelligent Systems, Zakopane, Poland, June 10-13, 1996.`

1 Introduction

Recently, Haas [6], Schubert [16], and Reiter [13], have developed an alternative approach to the frame problem which is based on the idea of using *explanation closure axioms*. The claim is that for characterizing nonchange in serial worlds with fully specified actions, one can have both a succinct representation of frame axioms and an effective proof theory for the characterization. In Schubert's case, the downside is that the explanation closure axioms which are essential to the approach, must be generated manually. In fact, Schubert claims that since the closure axioms are generally domain dependent there is little chance of automating their generation. Reiter fares somewhat better in this respect, because he can generate closure axioms for a restricted class of problems by using a meta theoretic-assumption of completeness together with syntactic transformations applied to action effect axioms. On the other hand, both approaches are limited to deterministic actions and are subject to limitations inherent in the situation calculus.

Briefly, the proposal suggests that rather than generating one frame axiom for each action-fluent pair in a theory, one can generate one (two in the case of Reiter) explanation closure axiom per fluent. Each axiom characterizes the only explanations for a fluent changing value. The explanations are characterized in terms of the actions which potentially effect the fluent. In Schubert [16], an explanation closure axiom is added to an action theory for each fluent. For example, in a robot scenario, given the fluent *Holding* and a number of actions which include *Putdown* and *Drop*, Schubert ([16], p. 27) proposes the following closure axiom,

$$(\forall a, x, s, s')[[Holding(R, x, s) \wedge \neg Holding(R, x, s') \wedge s' = Result(a, s)] \\ \supset a \in \{Putdown(R, x), Drop(R, x)\}], \quad (1)$$

where $a \in \{a_1, \ldots, a_n\}$ abbreviates $a = a_1 \vee, \ldots, \vee a = a_n$. This states that the only explanations for the robot R ceasing to hold an object x are the actions *Putdown* and *Drop*.

The basic idea is that when given an action theory describing the effects of actions, and possibly including domain constraints, one manually constructs the necessary closure axioms and adds them to the theory. The original theory together with the closure axioms allows one to reason monotonically about the action scenario characterized. The claim is that not only does one avoid the use of nonmonotonic logics, but one also avoids the space complexity associated with the original approach. Throughout the paper, we will focus on Schubert's work, but much of the discussion should apply to any approach using explanation closure that claims to provide a monotonic solution to the frame problem.

Schubert claims to provide "evidence that explanation closure axioms provide a succinct encoding of nonchange in serial worlds with fully specified actions". He also claims that "they also offer advantages over circumscriptive and nonmonotonic approaches, in that they relate nonchange to intuitively transparent explanations for change, retain an effective proof theory, and avoid unwarranted persistence inferences".

In this paper, we provide evidence against the claims concerning circumscriptive and nonmonotonic approaches in the following manner. We first present a slightly modified version of PMON ([15], [2]), a logic of action and change, which uses circumscription and is nonmonotonic. PMON can be viewed as a circumscriptive presentation of explanation closure, although the original proposal was made independently of the explanation closure approaches. The implicit nonmonotonicity inherent in the "monotonic" solution to the frame problem characterized by explanation closure, is made explicit in the context of PMON. We show that PMON

1. relates nonchange to intuitively transparent explanations for change;
2. retains an effective proof theory;
3. and avoids unwarranted persistence inferences,

unlike a number of other circumscriptive approaches. In addition,

- the equivalent of explanation closure axioms are automatically generated by reducing the circumscription axiom used in PMON to a logically equivalent first-order formula;
- our approach works for a class of problems which includes both non-deterministic and context-dependent actions;
- a smaller number of general axioms may be used due to our use of fluent variables.

Finally, we claim that the distinction between the explanation closure approach as being "monotonic", and circumscriptive approaches as being "non-monotonic" is not fully justified. In this respect, we agree with Lifschitz [10], and also supply evidence "that a circumscriptive presentation of explanation closure may lead to a generalization of this method that will be applicable to nondeterministic actions".[3]

2 Action Scenarios and $\mathcal{L}(FL)$

The formal syntax for specifying scenario descriptions is defined in terms of a surface language $\mathcal{L}(SD)$, consisting of action occurrence statements (`ac1,ac2`), action (law) schemas (`acs1,acs2`), and observation statements (`obs1`). In what follows, all expressions occurring in scenario descriptions will be prefixed. We shall use the symbols "obs", "ac" and "acs" to denote observation statements, action occurrence statements and action schemas, respectively.

Example 1. The following is the Yale shooting scenario (below al and l are fluent constants standing for *alive* and *loaded*, respectively, while $Load$ and $Fire$ are action symbols).

obs1 [0] $al \land \neg l$
ac1 [2,4] $Load$
ac2 [5,6] $Fire$
acs1 $[t_1, t_2]$ $Load \leadsto [t_1, t_2]\ l := T$
acs2 $[t_1, t_2]$ $Fire \leadsto ([t_1]\ l \rightarrow [t_1,t_2](al \land l) := F)$.

Given a scenario description Υ, consisting of statements in the surface language $\mathcal{L}(SD)$, these statements can be translated into formulas in the language $\mathcal{L}(FL)$ via a two-step process. In the first step, action schemas in Υ are instantiated with action occurrence statements, resulting in what are called *schedule statements*. The resulting schedule statements replace the action schemas and action occurrence statements. The result is an *expanded (action) scenario description* Υ', consisting of both schedule and observation statements. In the second step, abbreviation definitions are used to translate statements in Υ' into formulas in $\mathcal{L}(FL)$.

The language $\mathcal{L}(FL)$ is a sorted first-order language with sorts for fluents, actions, and temporal entities. The surface language $\mathcal{L}(SD)$ serves as a convenient set of macros for representing action scenarios. Formal reasoning is done

[3] [10], p. 11.

in $\mathcal{L}(FL)$. The notation for an action scenario in $\mathcal{L}(FL)$ is

$$\Gamma_C = \Gamma_{OBS} \cup \Gamma_{SCD} \cup \Gamma_{UNA},$$

where Γ_{OBS} and Γ_{SCD} are translations of the observation and schedule statements in the surface language into $\mathcal{L}(FL)$, respectively, while Γ_{UNA} are the unique names axioms for the respective sorts in $\mathcal{L}(FL)$.

The use of $\mathcal{L}(FL)$ and $\mathcal{L}(SD)$ should be clear from the examples in Section 5. See [2] and [1] for detailed definitions of both languages and the translation process.

3 PMON Circumscription

PMON was originally proposed by Sandewall [15], in terms of a model theoretic preferential semantics. It has been assessed correct using the Features and Fluents framework for the $\mathcal{K} - IA$ class of reasoning problems which include nondeterministic actions, actions with duration, partial or complete specification of any state in a scenario, including the first, and incomplete specification of the timing and order of actions. Doherty ([2],[5]) developed PMON by translating Sandewall's representation into a conventional sorted FOPC, providing a circumscription axiom for the PMON logic of preferential entailment and then showing that for the $\mathcal{K} - IA$ class, the circumscription axiom can be reduced to a first-order formula. Consequently, standard classical theorem provers for monotonic FOPC can be used to reason about action scenarios in the $\mathcal{K} - IA$ class. The logic described in this section is a slightly modified version of that in [2]. The main difference is a new sort for actions which allows for their reification.

In the following, $Circ_{SO}(\Gamma; \ldots)$ and $Circ_{PW}(\Gamma; \ldots)$ denote standard 2nd-order and pointwise circumscription as described in [11] and [9], respectively.

3.1 Occlusion

Associated with each action type is a subset of fluents that are influenced by the action. If the action has duration, then during its performance, it is not known in general what value the influenced fluents have. Since the action performance can potentially change the value of these fluents at any time, all that can generally be asserted is that at the end of the duration the fluent is assigned a specific value. To specify such behavior, an *Occlude* predicate is introduced and used in the definition of reassignment expressions. The occlusion predicate is used as part of the definition of a reassignment expression which in turn is used as part of the definition of an action schema.

The predicate *Occlude* takes an action, fluent and timepoint as argument. For example, if the $[t,t']Load$ action is performed then the formula $\forall t''.t < t'' \leq t' \rightarrow Occlude(load, t'', loaded)$ represents the fact that *loaded* will be occluded from t to t', the duration of the *Load* action. The definition for a reassignment expression $[\mathbf{s},\mathbf{t}]\delta := T$ used in an action occurrence statement with action α is

$$(\exists t.\mathbf{s} \leq t < \mathbf{t} \wedge \forall t'.(t < t' \leq \mathbf{t} \rightarrow Holds(t',\delta))) \\ \wedge(\forall t''.(\mathbf{s} < t'' \leq \mathbf{t} \rightarrow Occlude(\alpha, t'', \delta)))$$

The definition for $[\mathbf{s},\mathbf{t}]\delta := F$ is similar, but the $Hold$'s atom is negated. Technically, occlusion is a device which is used to mask fluent changes from influencing choice of preferred models in the minimization process.

3.2 The Nochange Axiom

Let Γ_{NCG} denote the following *nochange axiom set*:

$$\{\forall f, t.Holds(t, f) \oplus Holds(t + 1, f) \supset \exists a.Occlude(a, t + 1, f)\}, \qquad (2)$$

where the connective $\oplus$ is an abbreviation for the exclusive-or connective. This axiom asserts that for any fluent f and time-point t, if the value of f changes from time-point t to $t+1$, then there is an action a which causes f to be occluded from t to $t + 1$. The nochange axiom implicitly asserts a persistence assumption which is observed by taking the contraposition of Γ_{NCG}:

$$\{\forall a, f, t.\neg Occlude(a, t + 1, f) \supset Holds(t, f) \equiv Holds(t + 1, f)\}. \qquad (3)$$

Relation to Explanation Closure It is clear that the nochange axiom

$$\{\forall f, t.Holds(t, f) \oplus Holds(t + 1, f) \supset \exists a.Occlude(a, t + 1, f)\}, \qquad (4)$$

provides an explanation for a fluent f changing value from time t to $t+1$ in terms of actions, provided one has both the necessary and sufficient conditions for a tuple $\langle a, t, f\rangle$ having the property $Occlude$. The schedule axioms provide the sufficient conditions, whereas the minimization of $Occlude$ in Γ_{SCD}, discussed in the next section, provides the necessary conditions.

In PMON, the EC axiom corresponding to (1) would be derived in two stages. First, instantiate formula (4), with the fluent in question:

$$\forall t.H(t, h(R, o)) \oplus H(t + 1, h(R, o)) \supset \exists a.Occlude(a, t + 1, h(R, o)), \qquad (5)$$

where $h(R, o)$ is a fluent constant representing the fact that "Robot R is holding object o" and H is $Holds$. We can of course extend the fluent sort to deal with complex fluents, but will avoid these complications in this paper. Secondly, minimize $Occlude$, but only relative to the schedule axioms. The derived definition of $Occlude$, together with (5), would then be used to show that $a = Putdown(R, o)$ or $a = Drop(R, o)$.

3.3 Filtered Preferential Entailment

Filtered preferential entailment is a technique originally introduced by Sandewall [14] for dealing with postdiction. The filtering technique is based on distinguishing between different types of formulas in a scenario, In this particular case, between schedule and observation axioms. Given a scenario description $\Gamma_C = \Gamma_{OBS} \cup \Gamma_{SCD} \cup \Gamma_{UNA}$, the basic idea is to minimize only the schedule axioms Γ_{SCD} relative to the *Occlude* predicate and then use the intersection of the *Occlude* minimal models with the models for the observation axioms and the nochange axiom as the class of preferred models.

3.4 PMON Circumscription

The PMON minimization policy combines the occlusion concept, nochange premises and the filtering technique in the following manner. Given a scenario description Υ, and the corresponding formulas Γ_C in $\mathcal{L}(FL)$, the *Occlude* predicate will be minimized globally relative to Γ_{SCD} and then filtered with Γ_{NCG} and Γ_{OBS}. Let $Circ_{SO}(\Gamma_{SCD}(Occlude); Occlude) =$

$$\Gamma_{SCD}(Occlude) \wedge \forall \Phi . \neg[\Gamma_{SCD}(\Phi) \wedge \Phi < Occlude] \tag{6}$$

denote the PMON circumscription axiom with *Occlude* minimized and *Holds* fixed. PMON circumscription is then defined as

$$\Gamma_{NCG} \wedge \Gamma_C \wedge Circ_{SO}(\Gamma_{SCD}(Occlude); Occlude).$$

Observe that the circumscription policy is surprisingly simple, yet at the same time is assessed correct for the very broad ontological class $\mathcal{K} - IA$.

4 Reduction to the First-Order Case

Although $Circ_{SO}(\Gamma_{SCD}(Occlude); Occlude)$ is a second-order formula, it can be shown that it is equivalent to a first-order formula using two results by Lifschitz [9], and the fact that *Occlude*-atoms only occur positively in Γ_{SCD}. Lifschitz's results allow us to show that for any Γ_C with the required restrictions on *Occlude*-atoms, the PMON circumscription of Γ_C is equivalent to the following first-order formula,

$$\begin{aligned} &\Gamma_{NCG} \wedge \Gamma_C \wedge \forall a, t, f . \neg[Occlude(a, t, f) \wedge \\ &\Gamma_{SCD}(\lambda a', t', f' . (Occlude(a', t', f') \wedge \langle a', t', f' \rangle \neq \langle a, t, f \rangle))]. \end{aligned} \tag{7}$$

More recently, we have shown in Doherty [1] that standard predicate completion can be used to derive a definition of *Occlude*, which is not so surprising considering the form of schedule axioms. More importantly, we have recently proposed an efficient algorithm for reducing a large class of circumscription axioms to logically equivalent 1st-order formulas via quantifier elimination techniques ([3]). The DLS algorithm can be used for the PMON circumscription axiom. Consequently, we have an automatic method for "compiling" away the 2nd-orderness of the circumscription axiom and generating the necessary conditions for a tuple being a member of *Occlude*.

Such reductions are very useful in the sense that one can reason about any scenario description in the $\mathcal{K}-IA$ class using standard theorem provers for monotonic FOL. In addition, since the temporal structure is linear discrete time with $+$, $<$, and $=$, existing logic-based constraint packages could be used to increase efficiency of the implementation. These results provide not only an alternative to, but an explanation for the role of nonmonotonicity in the explanation closure approach.

5 Some Examples

5.1 Yale Shooting Problem

Example 2. This example is due to Hanks and McDermott [7] The YSP scenario description is (below al, l, lo, and fi are fluent constants standing for *alive*, *loaded*, *load*, and *fire*, respectively, while *Load* and *Fire* are action symbols),

obs1 $[0]\ al \wedge \neg l$
ac1 $[2,4]\ Load$
ac2 $[5,6]\ Fire$
acs1 $[t_1, t_2]\ Load \rightsquigarrow [t_1, t_2]\ l := T$
acs2 $[t_1, t_2]\ Fire \rightsquigarrow ([t_1]\ l \rightarrow [t_1,t_2](al := F \wedge l := F)$.

The corresponding formulas in $\mathcal{L}(FL)$ are,

obs1 $Holds(0, al) \wedge \neg Holds(0, l)$
scd1 $\exists t.2 \leq t < 4 \wedge \forall t'(t < t' \leq 4 \supset Holds(t', l))$
$\wedge \forall t'.(2 < t' \leq 4 \supset Occlude(lo, t', l)$
scd2 $Holds(5, l) \supset$
$([\exists t.5 \leq t < 6 \wedge \forall t'(t < t' \leq 6 \supset \neg Holds(t', al))] \wedge$
$\wedge \forall t'.(5 < t' \leq 6 \supset Occlude(fi, t', al)] \wedge$
$[\exists t.5 \leq t < 6 \wedge \forall t'(t < t' \leq 6 \supset \neg Holds(t', l))])$
$\wedge \forall t'.(5 < t' \leq 6 \supset Occlude(fi, t', l)]$.

For the YSP scenario, $\Gamma_C = \Gamma_{OBS} \cup \Gamma_{SCD} \cup \Gamma_{UNA}$, where

$$\Gamma_{OBS} = \{obs1\}, \Gamma_{SCD} = \{scd1, scd2\}, \Gamma_{UNA} = \{l \neq al \wedge lo \neq fi\},$$

$$\Gamma_{NCG} = \{\forall f, t, a.\neg Occlude(a, t+1, f) \supset Holds(t, f) \equiv Holds(t+1, f)\}.$$

After circumscribing *Occlude* in Γ_{SCD}, the following definition for *Occlude* can be derived using predicate completion,

$$\begin{aligned} \forall a, t, f. \quad & [(2 < t \leq 4 \wedge f = l \wedge a = lo) \vee \\ & (Holds(5, l) \wedge 5 < t \leq 6 \wedge f = al \wedge a = fi) \vee \\ & (Holds(5, l) \wedge 5 < t \leq 6 \wedge f = l \wedge a = fi)] \equiv Occlude(a, t, f). \end{aligned} \tag{8}$$

The derived formula (8) succinctly describes the explanations for a fluent f possibly changing value at a timepoint t. For example, the only actions that can change the value of fluent al are fi. The only actions that can change the value of fluent l are lo, and fi. In order to find the actions a which can change a fluent fl's value, simply look at each disjunct on the left side of (8) where $f = fl$. Each ac in the associated subformula $a = ac$ provides a potential explanation for the fluent changing value. In addition, both the temporal constraints and preconditions can be listed by considering the left side of (8).

Generating Explanations In the following, we will demonstrate the derivation of explanations. Given the scenario above, we can derive that

$$Holds(5, l) \wedge \neg Holds(6, l). \tag{9}$$

Suppose we would like to explain why this is the case. By (2),

$$\exists a. Occlude(a, 6, l). \tag{10}$$

Which action a occludes l at timepoint 6? Since $f = l$ and $t = 6$, the first two disjuncts on the LHS of (8) are false. Consequently,

$$\forall a. Holds(5, l) \wedge 5 < 6 \leq 6 \wedge l = l \wedge a = fi \equiv Occlude(a, 6, l). \tag{11}$$

It follows that

$$\forall a. Holds(5, l) \wedge a = fi \equiv Occlude(a, 6, l). \tag{12}$$

This states that if the precondition $Holds(5, l)$ to the action fi is true then $Occlude(fi, 6, l)$. Since $Holds(5, l)$ is true, the action fi provides an explanation for the fluent l changing value from timepoint 5 to 6.

5.2 The Fragile Example

The following example, described in ([16], p. 30), claims to show that one can not automatically generate EC axioms via circumscription or "biconditionalization". The claim is that in the general case, using circumscription together with abnormality theories or causal theories is too strong and would sanction unwarranted inferences. The problem exhibited by the example, is essentially one having to do with context dependent actions. Although we agree that in the general case, where ramification is taken into account, it may not be possible to completely automate generation of EC axioms due to qualification and domain specificity, we do not agree that context dependency is a problem with our particular circumscriptive approach. To be fair to Schubert, his claim of inadequacy is made for particular approaches, (see [12],[8]). On the other hand, PMON has similarities to both approaches. What distinguishes PMON from these approaches, is the ability to fine-tune the application of persistence to particular fluent-timepoint pairs via the use of the $Occlude$ predicate. Consequently, PMON does not suffer from overzealous application of the persistence assumption where disjunction is involved even though circumscription of the $Occlude$ predicate can be interpreted as a form of biconditionalization.

5.3 Fragile Problem

Example 3. The following is a modified version of Schubert's fragile example. (below $br(c)$, $ho(r,c)$, and $fr(c)$ are fluent constants denoting the features $broken(c)$, $holding(r,c)$ and $fragile(c)$, respectively, while $dr(r,c)$ is a fluent constant denoting the action $drop(r,c)$.)

obs1 $[0]\ \neg br(c)$
obs2 $[0]\ ho(r,c)$
ac1 $[0,1]\ Drop(r,c)$
acs1 $[t_1,t_2]\ Drop(r,c) \leadsto$
$([t_1]\ ho(r,c) \rightarrow [t_1,t_2]\ ho(r,c) := F) \wedge$
$([t_1]\ ho(r,c) \wedge fr(c) \rightarrow [t_1,t_2]\ br(c) := T)$.

The corresponding formulas in $\mathcal{L}(FL)$ are,

obs1 $\neg Holds(0, br(c))$
obs2 $Holds(0, ho(r,c))$
scd1 $(Holds(0, ho(r,c)) \supset$
$[\exists t.0 \leq t < 1 \wedge \forall t'(t < t' \leq 1 \supset \neg Holds(t', ho(r,c)))]$
$\wedge \forall t'.(0 < t' \leq 1 \supset Occlude(dr(r,c), t', ho(r,c)))) \wedge$
$(Holds(0, ho(r,c)) \wedge Holds(0, fr(c)) \supset$
$[\exists t.0 \leq t < 1 \wedge \forall t'(t < t' \leq 1 \supset Holds(t', br(c)))]$
$\wedge \forall t'.(0 < t' \leq 1 \supset Occlude(dr(r,c), t', br(c))))$.

For the fragile scenario, $\Gamma_C = \Gamma_{OBS} \cup \Gamma_{SCD} \cup \Gamma_{UNA}$, where

$$\Gamma_{OBS} = \{obs1, obs2\}, \Gamma_{SCD} = \{scd1\},$$

$$\Gamma_{UNA} = \{ho(r,c) \neq br(c) \wedge ho(r,c) \neq fr(c) \wedge \ldots\},$$

$$\Gamma_{NCG} = \{\forall f, t, a. \neg Occlude(a, t+1, f) \supset Holds(t, f) \equiv Holds(t+1, f)\}.$$

After circumscribing $Occlude$ in Γ_{SCD}, the following definition for $Occlude$ can be derived using the output of the DLS algorithm [3]:

$$\begin{aligned} \forall a, t, f. \quad & [(Holds(0, h(r,c)) \wedge 0 < t \leq 1 \wedge f = h(r,c) \wedge a = dr(r,c)) \vee \\ & (Holds(0, h(r,c)) \wedge Holds(0, fr(c)) \wedge 0 < t \leq 1 \wedge f = br(c) \wedge a = dr(r,c))] \\ & \equiv Occlude(a, t, f). \end{aligned} \tag{13}$$

It follows from Γ_C and (13) that we can neither derive $Holds(x, br(c))$ nor $\neg Holds(x, br(c))$ for $x > 0$.

Note that the explanation closure axiom analogous to (A5,[16]) is derived from the schedule axiom analogous to (A4,[16]) by a systematic and general principle, not dependent on the particular domain. The systematic and general principle is simply the automatic generation of the necessary conditions for a tuple being in $Occlude$ via the circumscription of the schedule axioms.

6 Discussion

Both Schubert and Reiter deal with a class of problems more general than that discussed in this paper. They consider ramification and concurrency and provide evidence that the explanation closure approach generalizes, at least for the situation calculus, to cover this expanded class of problems. PMON is still not equipped to deal with ramification or concurrency, but see [4] for an attempt at extending PMON for ramification. It remains to be investigated just how much one can generalize the reduction results and automatic generation of closure axioms for these expanded classes in the context of PMON.

We have provided a case for a circumscriptive version of explanation closure that has an effective proof theory and is applicable to both context dependent and nondeterministic actions. It also retains a large degree of representational succinctness and elaboration tolerance, since the process of generating closure axioms is fully automated. In addition, we feel some evidence has been provided that the monotonic/nonmonotonic dichotomy is not as clear cut as previously assumed and is not fully justified.

References

1. P. Doherty. Notes on PMON circumscription. Technical Report LITH-IDA-94-43, Department of Computer and Information Science, Linköping University, Linköping, Sweden, December 1994.
2. P. Doherty. Reasoning about action and change using occlusion. In *Proceedings of the 11th European Conference on Artificial Intelligence, Aug. 8-12, Amsterdam*, pages 401–405, 1994.
3. P. Doherty, W. Łukaszewicz, and A. Szałas. Computing circumscription revisited: Preliminary report. In *Proceedings of the 14th Int'l Joint Conference on Artificial Intelligence*, volume 2, pages 1502–1508, 1995. Extended version to appear in Journal of Automated Reasoning.
4. P. Doherty and P. Peppas. A comparison between two approaches to ramification: PMON(R) and $\mathcal{AR}_0$. In *Proceedings of the 8th Australian Joint Conference on Artificial Intelligence*, 1995.
5. P. Doherty and W. Łukaszewicz. Circumscribing features and fluents. In D. Gabbay and H. J. Ohlbach, editors, *Proceedings of the 1st International Conference on Temporal Logic*, volume 827 of *Lecture Notes in Artificial Intelligence*, pages 82–100. Springer, 1994.
6. A. R. Haas. The case for domain-specific frame axioms. In F. M. Brown, editor, *The Frame Problem in Artificial Intelligence*. Morgan Kaufmann, 1987.
7. S. Hanks and D. McDermott. Nonmonotonic logic and temporal projection. *Artificial Intelligence*, 33, 1987.
8. V. Lifschitz. Formal theories of action. In F. M. Brown, editor, *The Frame Problem in Artificial Intelligence*. Morgan Kaufmann, 1987.
9. V. Lifschitz. Pointwise circumscription. In M. Ginsberg, editor, *Readings in Nonmonotonic Reasoning*, pages 179–193. Morgan Kaufmann, 1988.
10. V. Lifschitz. Nested abnormality theories. *Artificial Intelligence*, 1995. To appear.
11. W. Łukaszewicz. *Non-Monotonic Reasoning – Formalization of Commonsense Reasoning*. Ellis Horwood Series in Artificial Intelligence. Ellis Horwood, 1990.

12. J. McCarthy. Applications of circumscription to formalizing common-sense knowledge. *Artificial Intelligence*, 28:89–116, 1986.
13. R. Reiter. The frame problem in the situation calculus: A simple solution (sometimes) and a completeness result for goal regression. In V. Lifschitz, editor, *Artificial Intelligence and Mathematical Theory of Computation*, pages 359–380. Academic Press, 1991.
14. E. Sandewall. Filter preferential entailment for the logic of action and change. In *Proc. Int'l Joint Conf. on Artificial Intelligence, (IJCAI-89)*, 1989.
15. E. Sandewall. *Features and Fluents: A Systematic Approach to the Representation of Knowledge about Dynamical Systems.* Oxford University Press, 1994.
16. L. Schubert. Monotonic solution of the frame problem in situation calculus. In H. E. Kyburg, R. P. Loui, and G. N. Carlson, editors, *Knowledge Representation and Defeasible Reasoning*, pages 23–67. Kluwer, 1990.

This article was processed using the LaTeX macro package with LLNCS style