A Closed-Form Solution to the Ramification Problem (Sometimes)[*] [†]

Sheila A. McIlraith [‡]
Department of Computer Science
University of Toronto
Toronto, M5S 3G4. Canada
E-mail: mcilrait@cs.utoronto.ca

October 19, 1997

Abstract

Our general concern is with how to integrate a representation of action into an existing set of state constraints. As has been observed in the literature, state constraints implicitly define indirect effects of actions as well as indirectly imposing further preconditions on the performance of actions. Thus, any representation scheme we propose must address the ramification and qualification problems, as well as the frame problem. In this paper we achieve such a representation for a syntactically restricted class of situation calculus theories.

This paper presents two major technical contributions. The first contribution is provision of an axiomatic closed-form solution to the frame, ramification and qualification problems for a common class of ramification constraints. The solution is presented in the form of an automatable procedure that compiles a syntactically restricted set of situation calculus ramification constraints and effect axioms into a set of successor state axioms. The second major contribution of this paper is provision of an independent semantic justification for this closed-form solution. In particular, we present a semantic specification for a solution to the frame and ramification problems in terms of a prioritized minimization policy, and show that the successor state axioms of our closed-form solution adhere to this specification. Observing that our minimization policy is simply an instance of prioritized circumscription, we exploit results of Lifschitz on computing circumscription [6] to show that computing the prioritized circumscription yields our successor state axioms. In the special case where there are no ramification constraints, computing the circumscription yields Reiter's earlier successor state axiom solution to the frame problem [17].

[*]I would like to acknowledge Fangzhen Lin for providing helpful comments on the circumscription results, and particularly for mathematically articulating my intuitions regarding the existence of a lemma that was essential to the proof of Theorem 3.

[†]A subset of the material presented in this paper also appears in Representing Actions and State Constraints in Model-Based Diagnosis, *Proceedings of Fourteenth National Conference on Artificial Intelligence* © AAAI Press, Menlo Park, CA, 1997.

[‡]Current Address: Xerox PARC, 3333 Coyote Hill Road, Palo Alto, CA 94304, E-mail: mcilrait@parc.xerox.com, Phone: 415-812-4332, Fax: 415-812-4334, and Knowledge Systems Laboratory, Stanford University, Gates Sciences Building 2A-248, Stanford, CA 94305-9020, E-mail: sam@hpp.stanford.edu.

1 Introduction

This paper presents an axiomatic closed-form solution to the frame, ramification and qualification problems for what we argue to be a commonly occurring class of state constraints. The results in this paper are motivated by and contribute towards addressing the following more general problem.

> Given a set of state constraints describing some aspect of the world which we henceforth refer to as the *system*, how do we integrate a representation of action and change, so that we can reason about the effects of an agent's[1] actions on the system, and the effect of the system on performing those actions.

This general problem arises in the context of many applications of artificial intelligence (AI). For example, in the case of diagnostic problem solving, we might have a set of state constraints representing the behaviour of some device, such as a power plant or a motor vehicle. We might then wish to integrate a representation of actions in order to perform such tasks as system maintenance, testing, repair or contingency planning. In contrast, in an active vision application, the state constraints might represent the ontology of objects that could occur in a scene, and we might wish to integrate a representation of actions in order to contemplate the effects of moving the camera or acting upon objects in the scene in order to achieve our goal of image understanding.

In the context of our general problem, state constraints serve two purposes. On the one hand, they define consistent states of our system. In this role, state constraints have traditionally been used to reason about the system; for example in the case of diagnosis, to conjecture diagnoses. In the context of a theory of action and change, state constraints have an additional role. They also serve as ramification constraints and qualification constraints, implicitly defining indirect effects of actions, and further constraining when actions can be performed, respectively. As a consequence, addressing our general problem must preserve the original role of our state constraints while providing a solution to the frame, ramification and qualification problems.

We achieve our objective by exploiting the language of the situation calculus, and integrating a situation calculus representation of action with our state constraints. This paper presents two major technical contributions to this end. In Sections 4 through 6, we show that for an arguably common class of ramification constraints, we can provide an axiomatic closed-form solution to the frame and ramification problems. Providing a closed-form solution means that our solution is present in the axiomatization as opposed to requiring computation. This solution is presented via an automatable procedure that compiles a set of situation calculus ramification constraints and effect axioms into a set of successor state axioms. To address the qualification problem, we appeal to existing results [9], compiling our qualification constraints, necessary conditions for action and successor state axioms into action precondition axioms.

A shortcoming in the justification of our closed-form solution is that it relies on an informal appeal to a completeness assumption. To overcome this shortcoming, the second major contribution of this paper is to provide independent semantic justification for our solution. We describe these results in Section 7. In order to achieve this semantic justification, we first define a prioritized minimization policy following the intuition followed by our closed-form solution. Appealing to this minimization policy we provide semantic specification for a solution to the frame and ramification problems. Further we show that under a consistency assumption, our successor state axioms

[1] An agent can be a human, another system, a robot, or nature.

are indeed a solution with respect to this specification. Observing that our minimization policy is simply an instance of prioritized circumscription, we exploit results by Lifschitz on computing circumscription [6] to show that computing the prioritized circumscription yields our successor state axioms. Finally, we show that when there are no ramifications, computing the circumscription results in the set of successor state axioms Reiter proposed as a solution to the frame problem [17]. This provides further justification for his solution to the frame problem.

2 The Situation Calculus

The situation calculus language we employ to represent our domains is a sorted first-order language with equality. The language consists of sorts *actions, situations,* and *domain*. Each action is represented as a (parameterized) first-class object within the language. The evolution of the world can be viewed as a tree rooted at the distinguished initial situation S_0. The branches of the tree are determined by the possible future situations that could arise from the realization of particular sequences of actions. As such, each situation along the tree is simply a history of the sequence of actions performed to reach it. The function symbol do maps an action term and a situation term into a new situation term. For example, $do(turn\_on\_pump, S_0)$ is the situation resulting from performing the action of turning on the pump in situation S_0. The distinguished predicate $Poss(a, s)$ denotes that an action a is possible to perform in situation s (e.g., $Poss(turn\_on\_pump, S_0)$). As such, $Poss$ determines the subset of the situation tree consisting of situations that are possible in the world. Finally, those properties or relations whose truth value can change from situation to situation are referred to as *fluents*. For example, the property that the pump is on in situation s could be represented by the fluent $on(Pump, s)$. In addition to the first-order language we use to axiomatize our domain, the situation calculus also consists of a set of foundational axioms, Σ_{found} which establish properties of our situations and situation tree [9]. Included in these axioms is definition of the binary relation $<$ which provides a partial ordering over situations in the subset of the situation tree that is $Poss$-ible. Finally, note that the situation calculus language we employ in this paper is restricted to primitive, determinate actions. Our language does not include a representation of time, concurrency or complex actions, but we intend to extend our results to more expressive dialects of the situation calculus (e.g., [18]) in future work.

3 Domain Axiomatization: An Example

Once again, our problem assumes an existing set of system state constraints and our task is to incorporate a representation of action, solving the frame, ramification and qualification problems. In this paper, we forgo preliminary discussion on transforming our original system state constraints into situation calculus state constraints (see [13] for such a discussion) and assume that our axiomatizer has given us a situation calculus domain axiomatization comprising the following sets of axioms

$$T_{SC} \cup T_{ef} \cup T_{nec} \cup T_{UNA} \cup T_{S_0}, \tag{1}$$

which we describe below. Note that this axiomatization does *not* solve the frame, ramification and qualification problems.

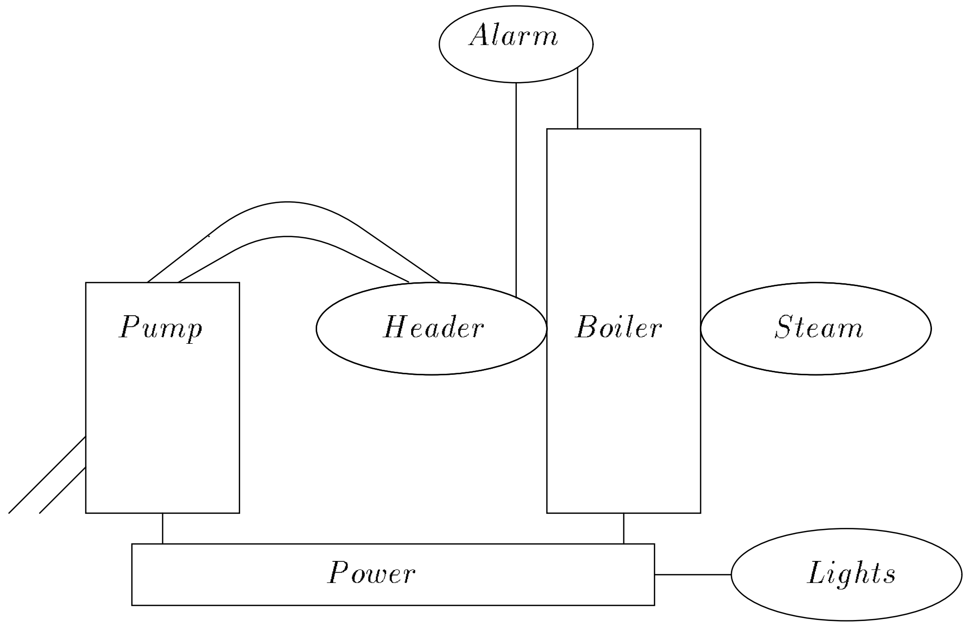

Figure 1: Power Plant Feedwater System

We illustrate this axiomatization with a simplified power plant feedwater system, depicted in Figure 1 and used for applications of diagnostic problem solving [4]. The system consists of three potentially malfunctioning components: a power supply ($Power$); a pump ($Pump$); and a boiler ($Boiler$). The power supply provides power to both the pump and the boiler. The pump fills the header with water, ($water\_entering\_header$), which in turn provides water to the boiler, producing steam. Alternately, the header can be filled manually ($manual\_fill$). To make the example more interesting, we assume that once water is entering the header, a siphon is created and water will only stop entering the header when the siphon is stopped. The system also contains lights and an alarm. (See [13] for a detailed description.)

Notation: all formulae are universally quantified from the outside, unless stated otherwise.

- T_{SC} is a set of situation calculus state constraints. These incorporate the existing system state constraints, indexed where appropriate with a situation term, s. T_{SC} is in turn comprised of sets of ramification constraints T_{ram}, qualification constraints T_{qual}, and domain constraints T_{domain}.
  - T_{ram}, the set of ramification constraints for our feedwater example is as follows:

$$\neg AB(Power, s) \wedge \neg AB(Pump, s) \wedge on(Pump, s) \supset water\_entering\_header(s) \quad (2)$$

$$manual\_fill(s) \supset water\_entering\_header(s) \quad (3)$$

$$AB(Power, s) \supset lights\_out(s) \quad (4)$$

$$\neg AB(Power, s) \supset \neg lights\_out(s) \quad (5)$$

$$water\_entering\_header(s) \wedge \neg AB(Power, s) \wedge \neg AB(Boiler, s) \wedge on(Boiler, s) \supset steam(s) \quad (6)$$

$$\neg\ (water\_entering\_header(s) \wedge \neg AB(Power, s) \wedge \neg AB(Boiler, s) \wedge on(Boiler, s)) \supset \neg steam(s) \quad (7)$$

$$\neg water\_entering\_header(s) \wedge on(Boiler, s) \supset alarm(s) \quad (8)$$

$$AB(Boiler, s) \supset alarm(s). \quad (9)$$

Axiom (2) states that if the power and pump are operating normally and if the pump is on, then it implies that water will be entering the header. As an aside, note that axioms (4) and (5) could be defined as one *iff* statement. We have written them as noted in order to simplify designation of the criteria for our forthcoming transformation procedure.

- T_{qual}, the set of qualification constraints for our feedwater example is as follows:

$$\neg(on(Pump, s) \wedge manual\_fill(s)). \tag{10}$$

- T_{domain}, the set of domain constraints for our feedwater example is as follows:

$$Power \neq Pump \neq Boiler. \tag{11}$$

Actions are axiomatized as a set of effect axioms T_{ef}, necessary conditions for actions T_{nec}, and unique names for actions T_{UNA}.

- T_{ef} is the set of effect axioms representing the changes in the truth values of fluents as a result of performing actions. For each fluent F in our language, we may have both positive and negative effect axioms of the following syntactic form,

$$Poss(a, s) \wedge \gamma_F^+(\vec{x}, a, s) \supset F(\vec{x}, do(a, s)) \tag{12}$$

$$Poss(a, s) \wedge \gamma_F^-(\vec{x}, a, s) \supset \neg F(\vec{x}, do(a, s)), \tag{13}$$

where $\gamma_F^+(\vec{x}, a, s)$ and $\gamma_F^-(\vec{x}, a, s)$ are simple formulas[2] whose free variables are among $\vec{x}, a, s$.
The following axioms compose T_{ef} for our feedwater example.

$$Poss(a, s) \wedge a = turn\_on\_pump \supset on(Pump, do(a, s)) \tag{14}$$

$$Poss(a, s) \wedge a = turn\_off\_pump \supset \neg on(Pump, do(a, s)) \tag{15}$$

$$Poss(a, s) \wedge a = turn\_on\_boiler \supset on(Boiler, do(a, s)) \tag{16}$$

$$Poss(a, s) \wedge a = turn\_off\_boiler \supset \neg on(Boiler, do(a, s)) \tag{17}$$

$$Poss(a, s) \wedge a = power\_fail \supset AB(Power, do(a, s)) \tag{18}$$

$$Poss(a, s) \wedge a = aux\_power \supset \neg AB(Power, do(a, s)) \tag{19}$$

$$Poss(a, s) \wedge a = power\_fix \supset \neg AB(Power, do(a, s)) \tag{20}$$

$$Poss(a, s) \wedge a = pump\_burn\_out \supset AB(Pump, do(a, s)) \tag{21}$$

$$Poss(a, s) \wedge a = pump\_fix \supset \neg AB(Pump, do(a, s)) \tag{22}$$

$$Poss(a, s) \wedge a = boiler\_blow \supset AB(Boiler, do(a, s)) \tag{23}$$

$$Poss(a, s) \wedge a = boiler\_fix \supset \neg AB(Boiler, do(a, s)) \tag{24}$$

$$Poss(a, s) \wedge a = turn\_on\_manual\_fill \supset manual\_fill(do(a, s)) \tag{25}$$

$$Poss(a, s) \wedge a = turn\_off\_manual\_fill \supset \neg manual\_fill(do(a, s)) \tag{26}$$

$$Poss(a, s) \wedge a = stop\_siphon \supset \neg water\_entering\_header(do(a, s)) \tag{27}$$

$$Poss(a, s) \wedge a = turn\_on\_alarm \supset alarm(do(a, s)) \tag{28}$$

$$Poss(a, s) \wedge a = turn\_off\_alarm \supset \neg alarm(do(a, s)). \tag{29}$$

[2] A simple formula with respect to s is one in which only domain specific predicate symbols are mentioned (i.e., they do not mention $Poss$ or $<$), in which fluents do not include the function symbol do, in which there is no quantification over sort $situations$, and in which there is at most one free $situations$ variable.

Axiom (14) states that if action a is possible in situation s, and a is the $turn\_on\_pump$ action, then the pump will be on in the situation resulting from performing action a in situation s.

- T_{nec} is the set of axioms representing the necessary conditions for an action to be performed. For each action prototype A in our language, necessary conditions are of the following form:

$$Poss(A(\vec{x}), s) \supset \pi_A^i, \tag{30}$$

where π_A^i is a simple formula with respect to s, whose free variables are among $\vec{x}, s$.
The following axioms compose *some* of T_{nec} for our feedwater example.

$$Poss(turn\_on\_pump, s) \tag{31}$$

$$Poss(turn\_off\_alarm, s) \supset (water\_entering\_header(s) \vee \neg on(Boiler, s)) \tag{32}$$

$$Poss(turn\_off\_alarm, s) \supset \neg AB(Boiler, s) \tag{33}$$

$$\ldots$$

$$Poss(turn\_on\_manual\_fill, s) \supset \neg alarm(s) \tag{34}$$

Axiom (34) states that if it is possible to turn on the manual filling then the alarm must be off.

- T_{UNA} is the set of unique names axioms for actions. They state that identical actions have identical arguments, and every action name refers to a distinct action.
The following axioms compose *some* of T_{UNA} for our feedwater example.

$$turn\_on\_pump \neq turn\_off\_pump \neq \ldots \neq turn\_off\_alarm \tag{35}$$

- T_{S_0} is the initial database. It captures what is known of the initial state of the world.
The following axioms might compose T_{S_0} for our feedwater example.

$$\neg AB(Power, S_0) \wedge \neg manual\_fill(S_0) \wedge \neg AB(Pump, S_0) \wedge \neg water\_entering\_header(S_0) \tag{36}$$

$$\neg on(Boiler, S_0) \wedge \neg on(Pump, S_0) \wedge \neg AB(Boiler, S_0) \tag{37}$$

4 The Frame and Ramification Problems

Once again, our domain axiomatization comprises the sets of axioms defined in (1). The job of the axiomatizer is done, but as previously observed, these axioms do not provide a solution to the frame, ramification and qualification problems. In this section, we propose a solution to the frame and ramification problems for what we argue to be a common class of ramification constraints. The qualification problem is discussed in a subsequent section.

Lin and Reiter [9] proposed a definition for a solution to the frame and ramification problems in our situation calculus language using circumscription and minimal model semantics. This solution has its limitations. Sometimes there is no minimal model. In other cases, there are multiple minimal models, some of which do not reflect the intended interpretation of the ramification and effect axioms. Most importantly, there is no guaranteed procedure to produce a closed-form solution.

Our contribution in this section is to provide an automatic procedure for generating a closed-form solution to the frame and ramification problems for a common class of state constraints. This solution is distinguished because it is closed-form and because it captures the *intended* interpretation of T_{SC} with respect to the theory.

4.1 The Problem

We illustrate our problem with a subset of the feedwater system example. Consider the ramification constraints, (2) and (3) above. The effect axioms, necessary conditions for actions and initial conditions are as defined in the previous section. Assume for the sake of simplifying the example that $\forall a, s. Poss(a, s)$, i.e., that all actions are possible in all situations.
Assume the action $turn\_on\_pump$ is performed in S_0, resulting in situation $S_1 = do(turn\_on\_pump, S_0)$. From effect axiom (14), we infer that $on(Pump, S_1)$. What do our ramification constraints tell us about the indirect effect of this action? Under Lin and Reiter's minimization policy to maximize persistence, three minimal models[3] are apparent.

$$\mathcal{M}_1 : \{\neg AB(Power, S_1), \neg AB(Pump, S_1), water\_entering\_header(S_1)\}$$
$$\mathcal{M}_2 : \{AB(Power, S_1), \neg AB(Pump, S_1), \neg water\_entering\_header(S_1)\}$$
$$\mathcal{M}_3 : \{\neg AB(Power, S_1), AB(Pump, S_1), \neg water\_entering\_header(S_1)\}$$

Clearly, the intended model is $\mathcal{M}_1$. Turning on the pump results in water entering the header. It does not result in an abnormal power supply, or an abnormal pump. We intuitively know this to be the intended model, because we have a basic understanding of machinery. More importantly, the axiomatizer has communicated the intended interpretation through the syntactic form of the ramification constraints, as we explain below.

Recall that our state constraints serve two purposes. On the one hand, they define consistent states of the world; However, in the context of a theory of action and change, state constraints have an additional role. They also serve as ramification and qualification constraints, indirectly constraining the effects of actions and further constraining the preconditions for actions

When employing the ramification constraints to infer the indirect effects of actions, the implication connective is interpreted as causal or *definitional*, in the logic programming sense. Following [5], we say that a fluent is **defined** in an axiom or set of axioms if it appears on the right-hand side of an implication connective in that axiom or set of axioms. Thus, it follows that an effect axiom for fluent F also serves to define fluent F.

If we assume that a fluent only changes value according to the effect axioms and the ramification constraints that *define* it, then the ramification constraints above only provide information about changes in the truth value of fluent $water\_entering\_header(s)$. With this assumption, we can conclude that the consequence of performing $turn\_on\_pump$ in S_0 is captured by model $\mathcal{M}_1$.

In the section to follow, we use this intuition to generate successor state axioms that reflect the intended interpretation of the ramification constraints and effect axioms, for a syntactically restricted class of theories.

4.2 A Closed-form Solution

In this section we provide a closed-form solution to the frame and ramification problems for axiomatizations whose syntactic representation of ramification constraints and effect axioms, collectively form a *solitary stratified theory*.

We combine the notion of solitary theory [6] and stratified logic program (e.g., [5]) to define the notion of a solitary stratified theory. Note that unlike stratified logic programs, we use a strictly

[3]We only list the relevant portion of the models here.

$<$ relation to distinguish the strata of our theories. Intuitively, a solitary stratified theory is a stratified logic program that allows negation in the consequent. If such a theory were represented as a dependency graph, the graph would have no cycles. The stratification of a solitary stratified theory need not be unique and we could write a procedure to determine a stratification automatically.

Definition 1 (Solitary Stratified Theory)
Suppose T is a theory in the language of the situation calculus with domain fluents, $\mathcal{L}$. Then T is a solitary stratified theory with **stratification** $(T_1, T_2, \ldots, T_n)$, *and* **partition** $(\mathcal{L}_1, \mathcal{L}_2, \ldots, \mathcal{L}_n)$ *if,*

- *for $i = 1, \ldots, n$, $\mathcal{L}_i$ is the set of fluents F_i that are defined in stratum T_i; and $\mathcal{L}_1 \cup \mathcal{L}_2 \cup \ldots \cup \mathcal{L}_n = \mathcal{L}$, and*
- *T is the union $T_1 \cup T_2 \cup \ldots T_n$ of sets of axioms T_i where for each stratum, T_i is solitary with respect to $\mathcal{L}_i$; i.e., each T_i can be written as the union $(\mathcal{D}_i \leq \neg\mathcal{L}_i) \cup (\mathcal{E}_i \leq \mathcal{L}_i)$, where*
  1. *$\mathcal{L}_i$ is the set of fluents, F_i such that $[\neg]F_i$ is defined in T_i;*
  2. *$\mathcal{D}_i \leq \neg\mathcal{L}_i$, is a set of formulae of the form $(D_i \supset \neg F_i)$, – at most one for each fluent $F_i \in \mathcal{L}_i$, where each D_i is a formula containing no fluents drawn from $\mathcal{L}_i \cup \ldots \cup \mathcal{L}_n$.*
  3. *$\mathcal{E}_i \leq \mathcal{L}_i$, is a set of formulae of the form $(E_i \supset F_i)$, – at most one for each fluent $F_i \in \mathcal{L}_i$, where each E_i is a formula containing no fluents drawn from $\mathcal{L}_i \cup \ldots \cup \mathcal{L}_n$.*

Example:
In our feedwater example, $T = T_{ram} \cup T_{ef}$ is a solitary stratified theory with stratification (T_1, T_2, T_3).

- T_1 comprises Effect Axioms (14) – (26),
- T_2 comprises Ramification Constraints (2) – (7), and Effect Axiom (27).
- T_3 comprises Ramification Constraints (8) – (9), and Effect Axioms (28) and (29).

In what follows, we define a seven step syntactic manipulation procedure which results in a closed-form solution to the frame and ramification problems for solitary stratified theory $T = T_{ef} \cup T_{ram}$. The solution is predicated on an appeal to a completeness assumption which enables us to generate explanation closure axioms.

Transformation Procedure

Let $T = T_{ram} \cup T_{ef}$ be a solitary stratified theory, with stratification $(T_1, T_2, \ldots, T_n)$.

Step 1. *For every fluent F_i defined in an effect axioms of T_i, generate at most one general positive and one general negative effect axiom as per axioms (12) and (13) above.*

Step 2. *For every fluent F_i defined in a ramification constraint of T_i, generate general positive and negative ramification axioms, relativized to situation $(do(a, s))$.*
General Ramification Axioms

$$v^+_{F_i}(do(a, s)) \supset F_i(do(a, s))^4 \tag{38}$$

$$v^-_{F_i}(do(a, s)) \supset \neg F_i(do(a, s)) \tag{39}$$

$v^+_{F_i}(do(a, s))$ and $v^-_{F_i}(do(a, s))$ are formulae whose free variables are among a, s, and any state or action arguments.

[4] Henceforth, action and state arguments, $\vec{x}$ will not be explicitly represented in canonical formulae.

Step 3. *Combine the two sets of axioms above, to define extended positive and negative effect axioms, for every fluent F_i.*
Extended Effect Axioms

$$Poss(a,s) \wedge (\gamma^+_{F_i}(a,s) \vee \upsilon^+_{F_i}(do(a,s))) \supset F_i(do(a,s)) \tag{40}$$

$$Poss(a,s) \wedge (\gamma^-_{F_i}(a,s) \vee \upsilon^-_{F_i}(do(a,s))) \supset \neg F_i(do(a,s)) \tag{41}$$

Example:
Extended positive and negative effect axioms for the fluent $(on(Pump, do(a,s))$, defined in T_1,

$$Poss(a,s) \wedge a = turn\_on\_pump \supset on(Pump, do(a,s)) \tag{42}$$

$$Poss(a,s) \wedge a = turn\_off\_pump \supset \neg on(Pump, do(a,s)). \tag{43}$$

For the fluent $water\_entering\_header(do(a,s))$, defined in T_2,

$$(\neg AB(Power, do(a,s)) \wedge \neg AB(Pump, do(a,s)) \wedge \; on(Pump, do(a,s)))$$
$$\vee \; manual\_fill(do(a,s)) \supset water\_entering\_header(do(a,s)) \tag{44}$$

$$Poss(a,s) \wedge a = stop\_siphon \supset \neg water\_entering\_header(do(a,s)). \tag{45}$$

Step 4. *Make the following completeness assumption regarding the effects and ramifications.*
All the conditions underwhich an action a can lead, directly or indirectly, to fluent F becoming true or false in the successor state are characterized in the extended positive and negative effect axioms for fluent F.

Step 5. *From the completeness assumption, generate explanation closure axioms.* We argue that if action a is possible in s and if the truth value of fluent F_i changes from $true$ to $false$ upon doing action a in situation s, then either $\gamma^-_{F_i}(a,s)$ is $true$ or $\upsilon^-_{F_i}(do(a,s))$ is $true$. An analogous argument can be made when the truth value of fluent F changes from $false$ to $true$ upon doing action a in situation s. This assumption is captured in the following positive and negative explanation closure axioms. For every fluent F_i,
Explanation Closure Axioms

$$Poss(a,s) \wedge F_i(s) \wedge \neg F_i(do(a,s)) \supset \gamma^-_{F_i}(a,s) \vee \upsilon^-_{F_i}(do(a,s)) \tag{46}$$

$$Poss(a,s) \wedge \neg F_i(s) \wedge F_i(do(a,s)) \supset \gamma^+_{F_i}(a,s) \vee \upsilon^+_{F_i}(do(a,s)). \tag{47}$$

Step 6. *From the extended positive and negative effect axioms and the explanation closure axioms, define intermediate successor state axioms for each fluent F_i.* We distinguish them as *intermediate* because, in the next step, we simplify them through a further syntactic transformation. For every fluent F_i,
Intermediate Successor State Axioms

$$Poss(a,s) \supset [F_i(do(a,s)) \equiv \Phi^*_{F_i}], \tag{48}$$

$$\Phi^*_{F_i} \equiv \gamma^+_{F_i}(a,s) \vee \upsilon^+_{F_i}(do(a,s))$$
$$\vee \; (F(s) \wedge \neg(\gamma^-_{F_i}(a,s) \vee \upsilon^-_{F_i}(do(a,s)))). \tag{49}$$

The set of intermediate successor state axioms, $T_{ISS} = \bigcup_{i=1,\ldots,n} T_{ISS_i}$, where T_{ISS_i} is the set of axioms for fluents F_i, defined in stratum T_i.

Example:
Intermediate successor state axioms for the fluent $on(Pump, do(a,s))$ defined in T_1 and for the fluent $water\_entering\_header(do(a,s))$ defined in T_2:

$$Poss(a,s) \supset [on(Pump, do(a,s)) \equiv a = turn\_on\_pump \vee \ (on(Pump,s) \wedge a \neq turn\_off\_pump)] \quad (50)$$

$$\begin{aligned} Poss(a,s) \supset [& water\_entering\_header(do(a,s)) \equiv \\ & manual\_fill(do(a,s)) \\ & \vee (\neg AB(Power, do(a,s)) \wedge \neg AB(Pump, do(a,s)) \wedge \ on(Pump, do(a,s))) \\ & \vee water\_entering\_header(s) \wedge a \neq stop\_siphon] \end{aligned} \quad (51)$$

Step 7. *By regressing[5] the intermediate successor state axioms, generate (final) successor state axioms.* These axioms are simple formulae containing no reference to fluents indexed by the situation term $do(a,s)$. For every fluent F_i,
Successor State Axioms

$$Poss(a,s) \supset [F_i(do(a,s)) \equiv \Phi_{F_i}] \quad (52)$$

where Φ_{F_i} is the following simple formula, $\mathcal{R}^{i-1}_{SS}[\Phi^*_{F_i}]$, i.e.,

$$\begin{aligned} \Phi_{F_i} \ \equiv \ & \gamma^+_{F_i}(a,s) \vee \mathcal{R}^{i-1}_{SS}[\upsilon^+_{F_i}(do(a,s))] \\ & \vee (F(s) \wedge \neg(\gamma^-_{F_i}(a,s) \vee \mathcal{R}^{i-1}_{SS}[\upsilon^-_{F_i}(do(a,s))])) \end{aligned} \quad (53)$$

and $\mathcal{R}^{i-1}_{SS}[\phi]$ is the regression of formula ϕ under successor state axioms $T_{SS_1}, \ldots, T_{SS_{i-1}}$.

The set of successor state axioms is $T_{SS} = \bigcup_{i=1,\ldots,n} T_{SS_i}$, where T_{SS_i} is the set of axioms for fluents $F_i \in \mathcal{L}_i$.

Example:
(50) is both the intermediate and the final successor state axiom for fluent $on(Pump, do(a,s))$. The intermediate successor state axiom (51) transforms into the following successor state axiom.

$$\begin{aligned} Poss(a,s) \supset [& water\_entering\_header(do(a,s)) \equiv \\ & a = turn\_on\_manual\_fill \\ & \vee \ (manual\_fill(s) \wedge a \neq turn\_off\_manual\_fill) \\ & \vee \ [(a \neq power\_fail \wedge (\neg AB(Power,s) \vee a = aux\_power \vee a = power\_fix)) \\ & \quad \wedge \ (a \neq pump\_burn\_out \wedge (\neg AB(Pump,s) \vee a = pump\_fix)) \\ & \quad \wedge \ (a = turn\_on\_pump \vee (on(Pump,s) \wedge a \neq turn\_off\_pump))] \\ & \vee \ (water\_entering\_header(s) \wedge a \neq stop\_siphon)]. \end{aligned} \quad (54)$$

[5] Regression (e.g., [21]) is a recursive rewriting procedure used here to reduce the nesting of the do function in situation terms. If F is a fluent with successor state axiom $Poss(a,s) \supset F(\vec{x}, do(a,s)) \equiv \Phi_F$ in T_{SS} then $\mathcal{R}_{SS}[F(t_1,\ldots,t_n, do(\alpha,\sigma))] = \Phi_F \mid^{x_1,\ldots,x_n,a,s}_{t_1,\ldots,t_n,\alpha,\sigma}$.

Proposition 1 *Suppose $T = T_{ef} \cup T_{ram}$ is a solitary stratified theory of the form described above. Then for every fluent $F_1 \in \mathcal{L}_1$, the successor state axiom for F_1 is identical to the intermediate successor state axioms for F_1, and is of the following general form.*

$$Poss(a,s) \supset [F_1(do(a,s)) \equiv \gamma^+_{F_1}(a,s) \vee (F_1(s) \wedge \neg\gamma^-_{F_1}(a,s)) \equiv \Phi^*_{F_1} \equiv \Phi_{F_1}]$$

Further, for any formula ϕ, $\mathcal{R}^i_{ISS}[\phi] = \mathcal{R}^i_{SS}[\phi]$, and for any fluent $F_i(\vec{x}, do(a,s))$,

$$\mathcal{R}^i_{ISS}[F_i(\vec{x}, do(a,s))] = \mathcal{R}^i_{SS}[F_i(\vec{x}, do(a,s))] = \mathcal{R}_{SS_i}[F_i(\vec{x}, do(a,s))]$$

where $\mathcal{R}^i_{ISS}$ denotes regression under the intermediate successor state axioms $T_{ISS_1} \cup \ldots \cup T_{ISS_i}$, $\mathcal{R}^i_{SS}$ denotes regression under the successor state axioms $T_{SS_1} \cup \ldots \cup T_{SS_i}$, and $\mathcal{R}_{SS_i}$ denotes regression under the successor state axioms T_{SS_i}

Our successor state axioms provide a closed-form solution to the frame and ramification problems. Since we have compiled T_{ef} and T_{ram} into T_{SS}, we can replace T_{ef} and T_{ram} by T_{SS} and $T^{S_0}_{ram}$ in (1), where $T^{S_0}_{ram}$ is the set of ramification constraints, relativized to S_0. We prove the legitimacy of this claim in Section 7.2.

5 The Qualification Problem

Our domain theory,

$$T_{UNA} \cup T_{SS} \cup T_{S_0} \cup T^{S_0}_{ram} \cup T_{qual} \cup T_{domain} \cup T_{nec} \tag{55}$$

now provides a solution to the frame and ramification problems. It remains to address the qualification problem. As previously observed the qualification constraints in T_{qual} can further restrict those situations s in which an action a is $Poss$-ible. We propose to use Lin and Reiter's solution [9], to determine a set of action precondition axioms T_{AP}. It transforms the necessary conditions for actions, T_{nec} and the qualification constraints, T_{qual} into a set of action precondition axioms T_{AP}. Following their results, we add one more step to our procedure.

Step 8. *Define one action precondition axiom for each action prototype $A(\vec{x})$ as follows.*
Action Precondition Axioms

$$Poss(A(\vec{x}), s) \equiv \Pi_A \wedge \bigwedge_{C \in T_{qual}} \Pi_C, \tag{56}$$

where,

$$\Pi_C \equiv \mathcal{R}_{SS}[C(do(A(\vec{x}), s))] \tag{57}$$

$\Pi_A \equiv \pi^1_A \vee \ldots \vee \pi^n_A$ for each π^i_A of (30) in T_{nec}. $\mathcal{R}_{SS}$ is the regression operator under the successor state axioms, T_{SS}.

Example: Consider (10) of T_{qual}, and (34) and (31) of T_{nec}. The action precondition axioms for $turn\_on\_manual\_fill$ and $turn\_on\_pump$ following Step 8 of our procedure are:

$$Poss(turn\_on\_manual\_fill, s) \equiv \neg alarm(s) \wedge \neg on(Pump, s) \tag{58}$$

$$Poss(turn\_on\_pump, s) \equiv \neg manual\_fill(s). \tag{59}$$

The action precondition axioms provide a closed-form solution to the qualification problem. Since we have compiled T_{nec} and T_{qual} into T_{AP}, we can replace T_{nec} and T_{qual} by T_{AP} and $T_{qual}^{S_0}$ in our theory, where $T_{qual}^{S_0}$ is the set of qualification constraints relativized to situation S_0. Lin and Reiter's solution also requires a domain closure axiom for actions, T_{DCA}.

6 Discussion of the Closed-Form Solution

Incorporating the results of the previous sections yields the following final domain theory which integrates our syntactically restricted state constraints and a representation of action, while solving the frame, ramification and qualification problems:

$$T_{UNA} \cup T_{DCA} \cup T_{SS} \cup T_{AP} \cup T_{S_0} \cup T_{SC}^{S_0} \cup T_{domain}. \tag{60}$$

This representation can be viewed as an executable specification because it is easily realized in Prolog by exploiting Prolog's completion semantics and simply replacing the equivalence signs, characteristic of T_{SS} and T_{AP}, by implications. The Lloyd-Topor transformation [11] must then be applied, to convert the resultant theory into Prolog clausal form. Indeed, as an interesting aside, in the sections to follow we show that our successor state axioms are semantically characterized as the outcome of computing a particular prioritized circumscription. Perfect models in logic programs have a prioritized circumscription semantics [16], thus the logic program produced from translation of our successor state axioms also has a perfect model semantics. See [13] for further details.

The state constraints that play the role of ramification constraints with respect to our theory of actions are compiled into successor state axioms, one for every fluent in our theory. When state constraints are absent, as in the case of Reiter's solution to the frame problem [17], successor state axioms provide a parsimonious representation for frame and effect axioms. In the presence of ramification constraints, the successor state axioms can, under certain conditions, grow exceedingly long. This presents the problem of trying to find the best trade-off between precompilation and runtime computation; a problem that many AI researchers face, and one that is often best addressed with respect to the specific domain. Fortunately, in our case we have an ideal compromise in those cases where T_{SS} proves to be unwieldy, that is to employ the *intermediate* successor state axioms as our representation. The axioms in T_{ISS} capture the intended interpretation of our domain but are only partially compiled, and thus don't risk the length concerns associated with the axioms in T_{SS}. Further, T_{ISS} preserves the compositionality of our representation, which is important in some model-based reasoning applications.

Recall that our closed-form solution to the frame and ramification problems is restricted to the class of solitary stratified theories. The syntax of such theories provides a mechanism for communicating the causal dependency relationship between fluents. Since the dependency graph dictated by the stratification of the fluents contains no cycles, the propagation of change from one fluent to another is singularly defined. While we can make no definitive claims about the frequency of occurrence of solitary stratified theories in general, they appear to occur quite commonly in the representations of industrial artifacts without feedback loops, e.g., the power plant feedwater system. In these systems, the causal dependency between fluents often reflects physical connectivity of components and subcomponents.

As we have observed, our closed-form solution appeals to a completeness assumption in order to generate explanation closure axioms. While this completeness assumption may not be valid for all domains, it is, for example, viewed as a reasonable assumption in the case of industrial artifacts, where the number of components is fixed, and where the environment is controlled. To more formally justify our closed-form solution, Section 7 provides an independent semantic justification. From these results we also show that our solution is predicated on a consistency condition.

7 Semantic Justification

In previous sections, we presented a closed-form solution to the frame and ramification problems for syntactically restricted ramification constraints and effect axioms that collectively form a solitary stratified theory. Our solution involved compiling effect axioms and ramification constraints into successor state axioms. Unfortunately, the compilation procedure, and as a consequence, our closed-form solution are predicated on a loose appeal to a completeness assumption. In the rest of this paper we provide an independent semantic justification for our closed-form solution. In particular we show how to specify and compute a solution to the frame and ramification problems using minimal model semantics and circumscription. This represents the second major technical contribution of this paper.

We achieve our semantic justification as follows. Exploiting the natural stratification of solitary stratified theories, we specify a nonmonotonic solution to the frame and ramification problems in terms of a prioritized minimization policy. We show that under a consistency assumption, our successor state axioms (52) are solutions to the frame and ramification problems with respect to the specification. We also show that any solution with respect to our specification is also a solution with respect to Lin and Reiter's specification [9]. In Section 7.2, we observe that our minimization policy is equivalent to a particular instance of prioritized circumscription. Through simple syntactic renaming and by exploiting results from Lifschitz on computing circumscription (e.g., [6]), we show that under a consistency assumption, computing this prioritized circumscription results in the computation of our successor state axioms. This result establishes the correctness of our closed-form solution with respect to our nonmonotonic specification. Finally, we use these results to show that, in the case where there are no ramification constraints, computing the circumscription results in the successor state axioms defined by Reiter in his solution to the frame problem [17].

7.1 Minimization Policy

In this section we define a prioritized minimization policy and use it to specify what counts as a solution to the frame and ramification problems for solitary stratified theories. To solve the frame problem, we wish to capture the intuition that things normally stay the same, and that when they do not, it is abnormal. We express the notion of abnormality through the distinguished predicate $ab_{F_i}(a, s)$[6], one for each fluent $F_i \in \mathcal{L}_i$, $i = 1, \ldots, n$. The predicate $ab_{F_i}(a, s)$ is an abbreviation for $\neg[F_i(s) \equiv F_i(do(a, s))]$.

We wish to minimize $ab_{F_i}(a, s)$, and in so doing, capture the intuition that in the absence of something abnormal, the truth value of a fluent persists after an action is performed. In order to

[6] The predicate ab is distinguished from the predicate AB, which is commonly used in model-based diagnosis representations to denote that a component of a system is behaving abnormally in a situation.

define our minimization policy, we must differentiate between an initial situation and the situation resulting from performing an action, which we will refer to henceforth as the resulting situation. Like the minimization policies advocated by Lin and Shoham [10] and Lin and Reiter [9], our policy minimizes $ab_{F_i}(a, s)$ with $Poss(a, s)$ and the truth status of fluents in the initial situation, $F_i(\vec{x}, s)$ remaining fixed. Fluents in the resulting situation, $F_i(do(a, s))$ are allowed to vary.

While we share basic minimization principles with previously advocated solutions to the frame and ramification problems, our minimization policy is distinguished because it places a priority ordering over the minimization of the predicate $ab_{F_i}(a, s)$. The ordering is

$$Ab_1 > Ab_2 > \ldots > Ab_n,$$

where Ab_i is a tuple containing the abnormality predicate $ab_{F_i}(a, s)$ for each fluent $F_i \in \mathcal{L}_i$. The priority ordering corresponds to the stratification of the solitary stratified theory, $T = T_{ef} \cup T_{ram}$. $Ab_1(a, s)$ is assigned the highest priority for minimization, and $Ab_n(a, s)$ is assigned the lowest priority.

Under this prioritized minimization policy, each $ab_{F_i}(a, s)$ is minimized, even at the expense of increasing the extent of predicates $ab_{F_{i+1}}(a, s), \ldots, ab_{F_n}(a, s)$ and fluents $F_k(do(a, s))$, $k = 1, \ldots, n$. The intuition behind this prioritized minimization hinges on the fact that our theory is solitary stratified. Recall from the discussion in Section 4.1 that the intended interpretation we wish to capture with our solitary stratified theories T is that a fluent F_i is defined by the axioms in stratum T_i, using fluents drawn from $\mathcal{L}_1, \ldots, \mathcal{L}_{i-1}$. Any mention of F_i in stratum T_j, $j > i$ contributes no further towards the definition of F_i, but rather serves to define the fluent F_j of that stratum. This captures the intended interpretation of our ramification constraints and effect axioms. As a result, to address the frame and ramification problems, we minimize each $ab_{F_i}(a, s)$ allowing $ab_{F_j}(a, s)$, $j > i$ to vary as well as allowing all the fluents in resulting situations, $F_k(do(a, s))$, $k = 1, \ldots, n$ to vary. As we will see, this prioritized minimization policy captures our intended solution to the frame and ramification problems for solitary stratified theories.

The definition of the minimization policy follows. Let s and a denote variables of sort *situation* and *action* respectively. Further, let σ_s, σ_a and σ_d denote assignment functions from free variables of sorts *situation*, *action* and *domain*, respectively.

Definition 2 (Prioritized Model Preference) *Suppose, T is a solitary stratified theory with stratification $(T_1, \ldots, T_n)$, domain fluents $\mathcal{L}$, and partition $(\mathcal{L}_1, \mathcal{L}_2, \ldots, \mathcal{L}_n)$, where $\mathcal{L} = \bigcup_{i=1}^{n} \mathcal{L}_i$. Suppose $ab_{F_i}(\vec{x}, a, s)$[7] abbreviates $\neg[F_i(\vec{x}, s) \equiv F_i(\vec{x}, do(a, s))]$ and $\mathcal{M}$ and $\mathcal{M}'$ are models of T.*

Model $\mathcal{M}'$ is preferred over model $\mathcal{M}$ with respect to variable assignment to situations, σ_s, (denoted by $\mathcal{M}' <_{\sigma_s} \mathcal{M}$), iff the following conditions hold.

1. *$\mathcal{M}$ and $\mathcal{M}'$ have the same universe of discourse.*
2. *$\mathcal{M}$ and $\mathcal{M}'$ agree on their interpretation of everything, including $Poss$, with the potential exception of domain fluents.*

3(a). *$\mathcal{M}$ and $\mathcal{M}'$ agree on the extensions of every fluent $F_i(\vec{x}, s)$, in every stratum T_i, $i = 1, \ldots, n$.*
I.e., for any assignment σ_a and σ_d, and any fluent $F_i(\vec{x}, s)$, $i = 1, \ldots, n$,
$\mathcal{M}', \sigma_s, \sigma_d \models F_i(\vec{x}, s)$ *iff* $\mathcal{M}, \sigma_s, \sigma_d \models F_i(\vec{x}, s)$

[7] For the purposes of this definition only, we explicitly include action and predicate arguments $\vec{x}$.

3(b). *For some* i, $1 \leq i \leq n$, $\mathcal{M}$ *and* $\mathcal{M}'$ *agree on the extensions of every* $ab_{F_j}(\vec{x}, a, s)$ *in stratum* T_j, $j = 1, \ldots, i-1$, *and the extensions of* $ab_{F_i}(\vec{x}, a, s)$ *in* $\mathcal{M}'$ *are a subset of the extensions of* $ab_{F_i}(\vec{x}, a, s)$ *in* $\mathcal{M}$.
I.e., for some i *and any assignment* σ_a *and* σ_d*, and any fluent* $F_j(\vec{x}, s)$, $j = 1, \ldots, i-1$,
$\mathcal{M}', \sigma_s, \sigma_a, \sigma_d \models Poss(a, s) \wedge \neg ab_{F_j}(\vec{x}, a, s)$ *iff* $\mathcal{M}, \sigma_s, \sigma_a, \sigma_d \models \neg ab_{F_j}(\vec{x}, a, s)$
and for some fluent $F_i(\vec{x}, s)$*, there are two assignments* σ_a *and* σ_d *such that,*
$\mathcal{M}, \sigma_s, \sigma_a, \sigma_d \models Poss(a, s) \wedge ab_{F_i}(\vec{x}, a, s)$ *but* $\mathcal{M}', \sigma_s, \sigma_a, \sigma_d \models \neg ab_{F_i}(\vec{x}, a, s)$

$\mathcal{M}$ is a minimal model of T if there is no $\mathcal{M}'$ and no variable assignment to situations σ_s such that $\mathcal{M}' <_{\sigma_s} \mathcal{M}$.

From our prioritized model preference, we provide a semantic specification for a solution to the frame and ramification problems for our syntactically restricted theories. In particular, we specify that under the prioritized minimization policy, the minimal models of our restricted theories prescribe solutions to the frame and ramification problem. Recall that Σ_{found} is the set of foundational axioms of the situation calculus [9].

Definition 3 (Semantic Specification) *Suppose* $\Sigma = \Sigma_{found} \cup T_{UNA} \cup T_{ef} \cup T_{ram}$ *where* $T = T_{ef} \cup T_{ram}$ *is a solitary stratified theory, with stratification* $(T_1, T_2, \ldots, T_n)$*, domain fluents* $\mathcal{L}$*, and partition* $(\mathcal{L}_1, \mathcal{L}_2, \ldots, \mathcal{L}_n)$*, such that* $\mathcal{L} = \bigcup_{i=1}^{n} \mathcal{L}_i$*. Finally suppose* $\mathcal{M}$ *is a minimal model of* Σ.

Then $\mathcal{M}$ *captures a solution to the frame and ramification problem for* Σ.

As observed in Section 4.1, Lin and Reiter previously defined a solution to the frame and ramification problem as the minimal models of our same Σ under a similar non-prioritized minimization policy [9]. Interestingly, our minimization policy collapses to Lin and Reiter's policy when $n = 1$.

Proposition 2 *If* $\mathcal{M}$ *is a minimal model of* Σ*, then* $\mathcal{M}$ *is also a minimal model of* Σ *under Lin and Reiter's minimization policy, outlined in [9].*

Remark 1 *If* $\mathcal{M}$ *captures a solution to the frame and ramification problem for* Σ *as specified in Definition 3, then it also meets Lin and Reiter's general specification for a solution to the frame and ramification problem, as outline in [9].*

To contrast our minimization policy to Lin and Reiter's, recall that their specification provides criteria for a solution to the frame and ramification problems. Unlike our specification which is limited to a syntactically restricted class of theories, their specification need not yield a minimal model, and indeed can yield multiple minimal models, some of which will not reflect the intended interpretation of the effect axioms and ramification constraints. Further, as we show in the pages to follow, our specification for our restricted theories guarantees a procedure to generate an axiomatic solution, whereas Lin and Reiter's does not.

Next we demonstrate the relationship between our semantically specified solution to the frame and ramification problem and the successor state axioms we defined in (52). This relationship is predicated on a consistency assumption. The consistency assumption, ensures that either an action is impossible to perform in situation s, or that performing the action will not result in a situation where a fluent is determined to be both true and false by some combination of effect axioms and ramification constraints. The unique names axioms, T_{UNA} ensure that no action has the effect of making a fluent both true and false in the same situation. The necessary conditions for actions, T_{nec} dictate that an action is impossible to perform in a situation if performing the action results in an inconsistency between the effect of the action and the intended effects of ramification constraints.

Under the consistency assumption, the successor state axioms combined with the foundational axioms, the unique names axioms for actions and the ramification constraints relativized to S_0 collectively entail the ramification constraints relativized to the situations accessible from S_0. This enables us to exclude T_{ram} and T_{ef} from our theory, provided $T_{ram}^{S_0}$ is present.

Theorem 1 *Suppose Σ is the theory defined in Definition 3 and T_{SS} is the set of successor state axioms as per (52). Further, assume that the following consistency condition holds,*

$$T_{UNA} \cup T_{nec} \models (\forall a, s).Poss(a, s) \supset \neg\, [(\gamma_{F_i}^{+}(a, s) \vee \mathcal{R}[\upsilon_{F_i}^{+}(do(a, s))]) \wedge (\gamma_{F_i}^{-}(a, s) \vee \mathcal{R}[\upsilon_{F_i}^{-}(do(a, s))])]. \quad (61)$$

Then for every ramification constraint $(\forall s).C(s) \in T_{ram}$,

$$\Sigma_{found} \cup T_{UNA} \cup T_{SS} \cup T_{ram}^{S_0} \models (\forall s).S_0 \leq s \supset C(s) \quad (62)$$

where $T_{ram}^{S_0}$ is the set of ramification constraints relativized to S_0,

$$T_{ram}^{S_0} = \{C(S_0) \mid (\forall s).C(s) \in T_{ram}\}.$$

To paraphrase, we make a consistency assumption (61) about our theory, which says that either an action is impossible, or if it is possible, that it is never the case that the direct effects or ramifications of an action (γ's and υ's, respectively) can make a fluent both false and true in the same situation. Under this assumption, (62) says that if we replace the effect axioms, T_{ef} and ramification constraints, T_{ram} by the successor state axioms of (52), T_{SS} and the ramification constraints relativized to the initial situation, $T_{ram}^{S_0}$, that the resulting theory will entail the ramification constraints, not only at situation S_0, but via the successor state axioms, at every situation s that follows S_0 on the tree of $Poss$-ible situations, i.e., those situations s such that $S_0 \leq s$.

The following theorem proves that, under the stated consistency assumption, the successor state axioms provide a solution to the frame and ramification problems, in keeping with our specification. Later, we will see that the results in this theorem are subsumed by Theorem 4.

Theorem 2 *Suppose Σ is the theory defined in Definition 3 and T_{SS} is the set of successor state axioms derived from T_{ef} and T_{ram} of Σ as per (52). Finally assume that the following consistency condition holds,*

$$T_{UNA} \cup T_{nec} \models Poss(a, s) \supset \neg[(\gamma_{F_i}^{+}(a, s) \vee \mathcal{R}[\upsilon_{F_i}^{+}(do(a, s))]) \wedge (\gamma_{F_i}^{-}(a, s) \vee \mathcal{R}[\upsilon_{F_i}^{-}(do(a, s))])].$$

Then if $\mathcal{M}$ is a model of $\Sigma_{found} \cup T_{UNA} \cup T_{SS} \cup T_{ram}^{S_0}$ then $\mathcal{M}$ is a minimal model of Σ and $\mathcal{M}$ captures a solution to the frame and ramification problems under Definition 3.

The models of these theories are not equivalent because the successor state axioms, T_{SS} only characterize the effects of $Poss$-ible actions, not all actions. Replacing the ramification constraints by $T_{ram}^{S_0}$ and T_{SS} is insufficient. To be complete, we must somehow express that the ramification constraints hold for the situations that are not accessible from S_0 using $Poss$. We can address this issue mathematically, but for most of our applications we are only interested in considering the subset of the situation tree that is $Poss$-ible, and so instead we simply restrict further discussion to this subset of all situations.

7.2 Computing Minimal Models using Circumscription

In this section we observe that semantic entailment in the minimal models of our prioritized model preference can be captured by circumscription and that, for the class of theories we are studying, the result of circumscription is first-order definable. We further show that for our class of theories, the successor state axioms defined in (52) are equivalent to those generated by computing our circumscription. Indeed, under a consistency assumption, we show that our circumscription computes the explanation closure axioms, and in turn the successor state axioms. This result formally establishes the equivalence between a monotonic theory which includes the successor state axioms of (52), and our nonmonotonic specification of a solution to the frame and ramification problems.

The objective of our circumscriptive policy is to minimize the difference between the truth value of fluents in an initial situation and a resultant situation. For any situation S, our circumscription minimizes $ab_{F_i}(a, S)$ with $Poss(a, S)$ and $F_i(S)$ fixed and with $F_i(do(a, S))$ allowed to vary.

To simplify the computation of this circumscription, we transform our theory Σ_{ram} into a simpler theory, Σ^*_{ram}. The circumscription is then computed with respect to Σ^*_{ram} by exploiting results of Lifschitz on computing circumscription (e.g., [6], [7]). Our objective in transforming our theory is three-fold.

- To make the literal ab explicit in our theory.
- To remove all mention of the situation term s, since our minimization policy and corresponding circumscription is defined with respect to a fixed situation S.
- To syntactically distinguish between F in $F(do(a, s))$ and F in $F(s)$ so that we can exploit results on computing circumscription, and in particular so that we can easily compute the predicate completion of fluents, F in our resultant situation, fixing fluents, F in the initial situation.

The transformation and results are not complex, although the notation may be a little off-putting. To illustrate the transformation, consider the effect axioms and ramification constraints for the fluent $alarm(s)$, as originally defined in our feedwater example.

$$Poss(a, s) \wedge a = turn\_on\_alarm \supset alarm(do(a, s))$$
$$Poss(a, s) \wedge a = turn\_off\_alarm \supset \neg alarm(do(a, s))$$
$$\neg water\_entering\_header(s) \wedge on(Boiler, s) \supset alarm(s)$$
$$AB(Boiler, s) \supset alarm(s)$$
$$\neg water\_entering\_header(do(a, s)) \wedge on(Boiler, do(a, s)) \supset alarm(do(a, s))$$
$$AB(Boiler, do(a, s)) \supset alarm(do(a, s))$$

Our first step is to distinguish the predicate $ab_{F_i}(a, s)$ into $ab^+_{F_i}(a, s) \wedge ab^-_{F_i}(a, s)$, and to make them explicit in our theory by adding positive and negative generic frame axioms, one for each fluent $F_i \in \mathcal{L}$. We refer to these frame axioms collectively as T_{frame}. In our example, our frame axioms are as follows.

$$Poss(a, s) \wedge alarm(s) \wedge \neg ab^-_{alarm}(a, s) \supset alarm(do(a, s))$$
$$Poss(a, s) \wedge \neg alarm(s) \wedge \neg ab^+_{alarm}(a, s) \supset \neg alarm(do(a, s))$$

Next, we rewrite our theory $\Sigma_{ram} \cup T_{frame}$ as a new theory, Σ^*_{ram}. To do so, we extend our language by the addition of a new predicate $Poss^*$ and new predicates F_i^*, F_i^{**}, $ab^{*+}_{F_i}$, and $ab^{*-}_{F_i}$, one for each fluent $F_i \in \mathcal{L}$. Next, for every axiom in $\Sigma_{ram} \cup T_{frame}$, we replace each occurrence of $Poss(a,s)$, $F(s)$, $F(do(a,s))$, $ab^+_{F_i}$, and $ab^-_{F_i}$ with the corresponding occurrence of $Poss^*$, F_i^*, F_i^{**}, $ab^{*+}_{F_i}$, and $ab^{*-}_{F_i}$. In our example above, the axioms are transformed as follows.

$$\begin{aligned}
Poss^*(a) \wedge a = turn_on_alarm \supset alarm^{**}(a) \\
Poss^*(a) \wedge a = turn_off_alarm \supset \neg alarm^{**}(a) \\
\neg water_entering_header^* \wedge on^*(Boiler) \supset alarm^* \\
AB^*(Boiler) \supset alarm^* \\
\neg water_entering_header^{**}(a) \wedge on^{**}(Boiler, a) \supset alarm^{**}(a) \\
AB^{**}(Boiler, a) \supset alarm^{**}(a) \\
Poss^*(a) \wedge alarm^* \wedge \neg ab^{*-}_{alarm}(a) \supset alarm^{**}(a) \\
Poss^*(a) \wedge \neg alarm^* \wedge \neg ab^{*+}_{alarm}(a) \supset \neg alarm^*(a)
\end{aligned}$$

Using analogous notation to that employed in Σ_{ram}, we refer to

$$\begin{aligned}
a = turn_on_alarm \quad & as \quad \gamma^{*+}_{alarm}(a) \\
a = turn_off_alarm \quad & as \quad \gamma^{*-}_{alarm}(a) \\
(\neg water_entering_header^* \wedge on^*(Boiler)) \vee AB^*(Boiler) \quad & as \quad v^{*+}_{alarm} \\
(\neg water_entering_header^{**}(a) \wedge on^{**}(Boiler, a)) \vee AB^{**}(Boiler, a) \quad & as \quad v^{**+}_{alarm}(a).
\end{aligned}$$

There is no v^{*-}_{alarm} and no $v^{**-}_{alarm}(a)$.

Generalizing, the transformed theory, Σ^*_{ram} is produced from $\Sigma_{ram} \cup T_{frame}$ as follows.

Definition 4 (Σ^*) *Suppose Σ is the theory defined in Definition 3. Define Σ^* to be the theory*

$$\Sigma_{found} \cup T_{UNA} \cup T^*_{ef} \cup T^*_{ram} \cup T^*_{frame}$$

where T_{frame} is the set of positive and negative frame axioms, one each for each fluent $F_i \in \mathcal{L}$,

$$Poss(a,s) \wedge F_i(\vec{x}, s) \wedge \neg ab^-_{F_i}(\vec{x}, a, s) \supset F_i(\vec{x}, do(a,s)) \tag{63}$$

$$Poss(a,s) \wedge \neg F_i(\vec{x}, s) \wedge \neg ab^+_{F_i}(\vec{x}, a, s) \supset \neg F_i(\vec{x}, do(a,s)) \tag{64}$$

and T^*_{frame}, T^*_{ef} and T^*_{ram} are T_{frame} above, and T_{ef}, T_{ram} drawn from Σ with

- *each occurrence of $F_i(\vec{x}, s)$ replaced by $F_i^*(\vec{x})$,*
- *each occurrence of $F_i(\vec{x}, do(a,s))$ replaced by $F_i^{**}(\vec{x}, a)$,*
- *each occurrence of $ab^{[+/-]}_{F_i}(\vec{x}, a, s)$ replaced by $ab^{*[+/-]}_{F_i}(\vec{x}, a)$, and*
- *each occurrence of $Poss(a,s)$ replaced by $Poss^*(a)$.*

Lemma 1 below establishes that our nonmonotonic specification of a solution to the frame and ramification problems can be captured by prioritized circumscription in our transformed theory. The results follow directly from the semantic definition of prioritized circumscription (e.g., [6]), and the definition of our prioritized model preference.

Lemma 1 *Suppose Σ is the theory defined in Definition 3 and Σ^* is the theory defined in Definition 4. Then $\mathcal{M}$ is a minimal model of Σ with respect to the prioritized model preference of Definition 2 iff $\mathcal{M}'$ is a model of*

$$\forall s. CIRC^+(\Sigma^*; Ab_1 > \ldots > Ab_n; F_1^{**}, \ldots, F_n^{**}),$$

where each Ab_i is a tuple containing the abnormality predicates $ab_{F_i}^{*+}(a)$ and $ab_{F_i}^{*-}(a)$, and where $CIRC^+$ is the circumscription $CIRC(\Sigma^*; Ab_1 > \ldots > Ab_n; F_1^{**}, \ldots, F_n^{**})$ with

- *each occurrence of $ab^{*[+/-]}(a)$ replaced by the corresponding $ab_{F_i}^{[+/-]}(a, s)$,*
- *each occurrence of F_i^* replaced by $F_i(s)$,*
- *each occurrence of $F_i^{**}(a)$ replaced by $F_i(do(a, s))$, and*
- *each occurrence of $Poss^*(a)$ replaced by $Poss(a, s)$.*

Lifschitz proved some very nice results identifying when circumscription is first-order definable, and when we can actually compute the axioms that result from a circumscription (e.g., [6], [7]). In the theorem to follow, we exploit these results to show that, under a consistency assumption, our prioritized circumscription of Ab_i with respect to Σ^* leads to the creation of explanation closure axioms, which when combined with effect axioms and ramification axioms, are equivalent to successor state axioms.

The consistency assumption upon which we predicate our theorem is the transformation of the assumption employed in the previous section. Recall that the objective of the consistency assumption is to ensure that $F_i(do(a, s))$ and $\neg F_i(do(a, s))$ never co-occurred. Since we have added generic frame axioms to Σ^*, we must reflect this addition in the consistency assumption.

Assumption 1 (Consistency Assumption) *For each fluent F_i, assume*

$$T_{UNA} \cup T_{nec}^* \models Poss^*(a) \supset \neg\, [(\gamma_{F_i}^{*+}(a) \vee \upsilon_{F_i}^{**+}(a) \vee (F_i^* \wedge \neg ab_{F_i}^{*-}(a))) \wedge\, (\gamma_{F_i}^{*-}(a) \vee \upsilon_{F_i}^{**-}(a) \vee (\neg F_i^* \wedge \neg ab_{F_i}^{*+}(a)))] \quad (65)$$

where T_{nec}^* is the transformation of T_{nec} as described in Definition 4.

Theorem 3 *Suppose Σ^* as defined in Definition 4 and Consistency Assumption 1 holds. Then*

$$\begin{aligned} CIRC(\Sigma^*; Ab_1 > \ldots > Ab_n; F_1^{**}(a), \ldots, F_n^{**}(a)) &\equiv \Sigma_{found} \cup T_{UNA} \cup T_{ef}^* \cup T_{ram}^* \cup T_{EC}^* \cup T_{Ab-equivs}^* \\ &\equiv \Sigma_{found} \cup T_{UNA} \cup T_{SS}^* \cup T_{Ab-equivs}^* \end{aligned}$$

- *$T_{EC}^* = \bigcup_{i=1}^n T_{EC_i}^*$ is the set of explanation closure axioms for theory Σ^*.*
  Each $T_{EC_i}^*$ is a set of formulae of the following form, one each for every $F_i \in \mathcal{L}_i$.

$$Poss^*(a) \wedge F_i^* \wedge \neg F_i^{**}(a) \supset \gamma_{F_i}^{*-}(a) \vee \upsilon_{F_i}^{**-}(a)$$
$$Poss^*(a) \wedge \neg F_i^* \wedge F_i^{**}(a) \supset \gamma_{F_i}^{*+}(a) \vee \upsilon_{F_i}^{**+}(a).$$

- *$T_{SS}^* = \bigcup_{i=1}^n T_{SS_i}^*$ is the set of successor state axioms for theory Σ^*.*
  Each $T_{SS_i}^*$ is a set of formulae of the following form, one for every $F_i \in \mathcal{L}_i$.

$$Poss^*(a) \supset [F_i^{**}(a) \equiv \gamma_{F_i}^{*+}(a) \vee \mathcal{R}^{i-1}[\upsilon_{F_i}^{**+}(a)] \vee (F_i^*(s) \wedge \neg(\gamma_{F_i}^{*-}(a) \vee \mathcal{R}^{i-1}[\upsilon_{F_i}^{**-}(a)]))]$$

  where $\mathcal{R}^{i-1}$ is the regression operator under the successor state axioms, $T_{SS_1}^* \cup \ldots \cup T_{SS_{i-1}}^*$.

- $T^*_{Ab-equivs} = \bigcup_{i=1}^{n} T^*_{Ab-equivs_i}$ *is the set of circumscribed definition of* $ab^{*+}_{F_i}(a)$ *and* $ab^{*-}_{F_i}(a)$. $T^*_{Ab-equivs_i}$ *is a set of formulae of the following form, one formula for every* $F_i \in \mathcal{L}_i$.

$$(ab^{*+}_{F_i}(a) \equiv Poss^*(a) \wedge \neg F^*_i \wedge (\gamma^{*+}_{F_i}(a) \vee \upsilon^{**+}_{F_i}(a))$$
$$(ab^{*-}_{F_i}(a) \equiv Poss^*(a) \wedge F^*_i \wedge (\gamma^{*-}_{F_i}(a) \vee \upsilon^{**-}_{F_i}(a)))$$

We have shown that our circumscription computes our successor state axioms in our transformed theory. In what follows we easily relate the results of Theorem 3 back to the successor state axioms of our original language.

Proposition 3 *Suppose* Σ *is the theory defined in Definition 3 and* Σ^* *is the theory defined in Definition 4 and assume that Consistency Assumption 1 holds. Then*

$$\forall s. CIRC^+(\Sigma^*; Ab_1 > \ldots > Ab_n; F_1^{**}, \ldots, F_n^{**}) \equiv \Sigma_{found} \cup T_{UNA} \cup T_{SS} \cup T_{Ab-equivs}$$

where

- $CIRC^+$ *is as defined in Lemma 1.*
- T_{SS} *is the set of successor state axioms for fluents* $F_i \in \mathcal{L}$ *of* Σ. *They are of the form of (52).*
- $T_{Ab-equivs}$ *is* $T^*_{Ab-equivs}$, *of Theorem 3, with each occurrence of* $ab^{*[+/-]}(a)$, F^*_i, $F^{**}_i(a)$, *and* $Poss^*(a)$ *replaced by the corresponding* $ab^{[+/-]}_{F_i}(a,s)$, $F_i(s)$, $F_i(do(a,s))$, *and* $Poss(a,s)$.

Finally, in the theorem to follow, we show that if we restrict our consideration to the situations that are $Poss$-ible in the world, (i.e., s, s.t. $S_0 \leq s$, using notation from Σ_{found}), that the nonmonotonic theory $\Sigma_{found} \cup T_{UNA} \cup T_{ef} \cup T_{ram}$ is equivalent to the monotonic theory $\Sigma_{found} \cup T_{UNA} \cup T_{SS}$.

Theorem 4 *Suppose* Σ *is the theory defined in Definition 3 and assume that the following consistency condition holds:*

$$T_{UNA} \cup T_{nec} \models Poss(a,s) \supset \neg[(\gamma^+_{F_i}(a,s) \vee \mathcal{R}[\upsilon^+_{F_i}(do(a,s))]) \wedge (\gamma^-_{F_i}(a,s) \vee \mathcal{R}[\upsilon^-_{F_i}(do(a,s))])].$$

Suppose $\mathcal{M}$ *is a model of* Σ. *Then for variable assignment* σ_s *to* s *such that,* $S_0 \leq s$,

$\mathcal{M}$ is a minimal model of Σ with respect to the prioritized model preference of Definition 2

iff

$\mathcal{M}'$ is a model of $\Sigma_{found} \cup T_{UNA} \cup T_{SS}$.

Using similar rewriting tricks, we can apply these results to Reiter's successor state axiom solution to the frame problem to establish that in the case where there are no ramification constraints, our prioritized minimization policy, and also Lin and Reiter's minimization policy [9] both compute Reiter's successor state axioms, and hence his closed-form solution to the frame problem. These results confirm the syntactic form of Reiter's successor state axiom solution.

Theorem 5 *Suppose* Σ *is the theory defined in Definition 3 and that* $T_{ram} = \{\}$. *Further, assume the following consistency assumption holds,*

$$T_{UNA} \models \neg(\gamma^+_{F_i}(a,s) \wedge \gamma^-_{F_i}(a,s)). \quad (66)$$

Suppose $\mathcal{M}$ *is a model of* Σ. *Then for variable assignment* σ_s *to* s *such that,* $S_0 \leq s$,

$\mathcal{M}$ is a minimal model of Σ with respect to the prioritized model preference of Definition 2 iff
$\mathcal{M}'$ is a model of $\Sigma_{found} \cup T_{UNA} \cup T_{SSF}$,
where T_{SSF} is the set of successor state axioms of the following form.

$$Poss(a, s) \supset [F_i(do(a, s)) \equiv \gamma^{+}_{F_i}(a, s) \vee (F_i(s) \wedge \neg\gamma^{-}_{F_i}(a, s))] \quad (67)$$

(66) states that an action cannot make a fluent both true and false in the same situation. It captures the same intuition as our previous consistency assumption without the need to discuss ramifications, and consequently, without the need to restrict ourselves to those situations that are $Poss$-ible. The successor state axioms, T_{SSF} defined in (67) are the successor state axioms Reiter identified as his solution to the frame problem [17].

This concludes the independent semantic justification for our closed-form solution. Proofs of theorems can be found in [13]. In what follows, we briefly discuss related work.

8 Related Work

The dialect of the situation calculus language used in this paper originates with the Cognitive Robotics Group at the University of Toronto. Our compilation approach to solving the ramification problem, and more specifically our appeal to a completeness assumption to generate explanation closure axioms is derivative of Reiter's [17], Schubert's [19] and Pednault's [14] approaches to solving the frame problem. The basic minimization policy we employed in our semantic justification is derivative of Lin and Shoham [10] and Lin and Reiter [9], with the important addition of making the minimization prioritized. This enabled us to define a closed-form solution for our restricted theories. The intuition behind our solution to the frame and ramification problems – the notion of interpreting our ramification constraints as definitional in nature, was influenced by research on the semantics of normal logic programs and deductive databases (e.g., [16]), and is related to preliminary work on this problem by Pinto [15]. Indeed the spirit of this solution – the notion of imposing a *directional* interpretation on our implication connective in our ramification constraints, is akin to the intuition behind proposed solutions to the ramification problem which advocate minimizing an explicitly represented notion of causality (e.g., [8], [12], [3], [20], [2]). Indeed the author suspects that for the syntactically restricted case studied here, all our different proposed solutions may produce the same results, just as many of the independent solutions to the frame problem prove to be identical under certain conditions [1]. What distinguishes this work in particular is that it provides an axiomatic closed-form solution and it retains the dual role played by our state constraints.

9 Contributions

This paper addressed the problem of integrating a theory of action with a pre-existing set of state constraints. The first major contribution of this paper was provision of an axiomatic closed-form solution to the frame, ramification and qualification problems for an arguably common class of theories, which we referred to as solitary stratified theories. The solution was presented as an automatable procedure that compiled effect axioms and ramification constraints into a set of successor state axioms. The benefit of our solution over many previous solutions is that the axiomatic closed-form solution enables us to use monotonic reasoning machinery to perform inference, rather than having to reason nonmonotonically.

The second major contribution of this paper was an independent semantic justification for our solution. Limiting our attention to solitary stratified theories, we proposed a semantic specification for a solution to the frame and ramification problems in terms of a prioritized minimization policy, proving that the successor state axioms of our closed-form solution agreed with this specification. Establishing our minimization policy as an instance of prioritized circumscription, we observed that this circumscription was first-order definable and showed that computing the prioritized circumscription yielded exactly our successor state axioms. We also showed that in the special case where there are no ramification constraints, computing the circumscription produced exactly Reiter's earlier successor state axiom solution to the frame problem. These results provide solid support for the syntactic form of our closed-form solution, and as a side effect, Reiter's as well.

10 Acknowledgements

I would like to thank Fangzhen Lin and Ray Reiter for their insightful comments on the work that is presented in this paper. I would also like to thank Yves Lespérance, Hector Levesque and Allan Jepson for reviewing this work and for posing some thought provoking questions.

References

[1] T. Costello. Solutions to the ramification problem. In *Working Papers for the Second Symposium on Logical Formalizations of Commonsense Reasoning*, pages 32–39, 1993.

[2] E. Giunchiglia. Determining ramifications in the situation calculus. In *Proceedings of the Fifth International Conference on Principles of Knowledge Representation and Reasoning (KR'96)*, pages 76–86, Cambridge, Massachusetts, USA., 1996.

[3] G. Neelakantan Kartha and V. Lifschitz. Actions with indirect effects (preliminary report). In *Proceedings of the International Conference on Principles of Knowledge Representation and Reasoning (KR'94)*, pages 341–350, 1994.

[4] B. Kramer and J. Mylopolous et al. Developing an expert system technology for industrial process control: An experience report. In *Proceedings of the Conference of the Canadian Society for Computational Studies of Intelligence (CSCSI'96)*, pages 172–186, 1996.

[5] G. Levi. *Advances in Logic Programming Theory*. Oxford Science Publications, 1994.

[6] V. Lifschitz. Computing Circumscription. In *Proceedings of the Ninth International Joint Conference on Artificial Intelligence (IJCAI-85)*, pages 121–127, 1985.

[7] V. Lifschitz. Circumscription. In D.M. Gabbay, C.J. Hogger, and J.A. Robinson, editors, *The Handbook of Logic in AI and Logic Programming*, volume 3, pages 298–352. Oxford University Press, 1994.

[8] F. Lin. Embracing causality in specifying the indirect effects of actions. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI-95)*, pages 1985–1991, 1995.

[9] F. Lin and R. Reiter. State constraints revisited. *Journal of Logic and Computation*, 4(5):655–678, 1994. Special Issue on Action and Processes.

[10] F. Lin and Y. Shoham. Provably correct theories of action. *Journal of the ACM*, 42(2):293–320, 1995.

[11] J.W. Lloyd. *Foundations of Logic Programming*. Springer Verlag, second edition, 1987.

[12] N. McCain and H. Turner. A causal theory of ramifications and qualifications. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI-95)*, pages 1978–1984, 1995.

[13] S. McIlraith. *Towards a Formal Account of Diagnostic Problem Solving*. PhD thesis, Department of Computer Science, University of Toronto, Toronto, Ontario, Canada, 1997.

[14] E. Pednault. Synthesizing plans that contain actions with context-dependent ef fects. *Computational Intelligence*, 4:356–372, 1988.

[15] J. Pinto. *Temporal Reasoning in the Situation Calculus.* PhD thesis, Department of Computer Science, University of Toronto, Toronto, Ontario, Canada, February 1994. Also published as Technical Report, Dept. of Computer Science, University of Toronto (KRR-TR-94-1), Feb. 1994.

[16] T.C. Przymusinski. On the declarative and procedural semantics of logic programs. *Journal of Automated Reasoning*, 5:167–205, 1989.

[17] R. Reiter. *The Frame Problem in the Situation Calculus: A Simple Solution (sometimes) and a completeness result for goal regression*, pages 359–380. Artificial Intelligence and Mathematical Theory of Computation: Papers in Honor of J. McCarthy. Academic Press, San Diego, CA, 1991.

[18] R. Reiter. Natural actions, concurrency and continuous time in the situation calculus. In L.C. Aiello, J. Doyle, and S.C. Shapiro, editors, *Proceedings of the Fifth International Conference on Principles of Knowledge Representation and Reasoning (KR'96)*, pages 2–13, Cambridge, Massachusetts, USA., November 1996. Morgan Kaufmann.

[19] 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, 1990.

[20] M. Thielscher. Computing ramifications by postprocessing. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI-95)*, pages 1994–2000, 1995.

[21] R. Waldinger. Achieving several goals simultaneously. In E. Elcock and D. Michie, editors, *Machine Intelligence 8*, pages 94–136. Ellis Horwood, Edinburgh, Scotland, 1977.