ELSEVIER
Artificial Intelligence 116 (2000) 87–121
Artificial Intelligence
www.elsevier.com/locate/artint

Integrating actions and state constraints: A closed-form solution to the ramification problem (sometimes)

Sheila A. McIlraith [1]

Knowledge Systems Laboratory, Department of Computer Science, Stanford University, Stanford, CA 94305-9020, USA

Received 15 July 1998

Abstract

Integrating actions and state constraints is a central problem in knowledge representation. State constraints are commonly used to represent the relationship between objects in the world. When a representation of action is integrated, state constraints implicitly define indirect effects of actions and impose further preconditions on the performance of actions. Thus, a semantically correct integration of actions and state constraints must address the ramification and qualification problems, as well as the frame problem. In this paper we achieve such an integration for a syntactically restricted class of situation calculus theories.

This paper presents two major technical contributions. The first contribution is an axiomatic closed-form solution to the frame, ramification and qualification problems for a common class of theories. 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 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 (1985) on computing circumscription 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 (1991) earlier successor state axiom solution to the frame problem.

Keywords: Ramification problem; Frame problem; Qualification problem; Reasoning about action; State constraints; Planning; Model-based reasoning

[1] Email: sam@ksl.stanford.edu.

0004-3702/00/$ – see front matter
PII: S0004-3702(99)00087-9

1. Introduction

This paper presents an axiomatic closed-form solution to the frame, ramification and qualification problems for a commonly occurring class of state constraints. The results in this paper are motivated by and contribute towards addressing the following 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, so that we can reason about the effects of an agent's [2] actions on the system, and the effect of the system on the agent's ability to perform 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 action in order to perform such tasks as monitoring, system maintenance, intrusive testing, repair, contingency planning or supervisory control. In an e-commerce application, we might have a set of state constraints representing the ontology of companies' products, their compatibility, their component parts, the suppliers of those parts, and the current inventory. We might wish to integrate a representation of the actions of various agent programs that interact automatically with the system to, for example, configure a system from component parts, or to buy or sell products. Finally, in an active vision application, the state constraints might represent the properties of and relationships between 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 support of image understanding.

Integrating actions and state constraints presents several knowledge representation challenges. In the context of a representation of action, state constraints play two roles. On the one hand, they capture the relationship between objects in the world, and hence the consistent states of a system. In this role, state constraints have traditionally been used to reason about system state; for example, in the case of diagnostic problem solving, to conjecture diagnoses. When integrated with a representation of action, state constraints play an additional role. They serve as ramification constraints and qualification constraints, implicitly defining indirect effects of actions, and further constraining when actions can be performed, respectively. Consequently, in addressing the general problem of integrating actions and state constraints, we must preserve the original role of our state constraints while providing a solution to the frame, ramification and qualification problems.

The frame, ramification and qualification problems are three classical problems that arise in reasoning about action using formal logic [6]. Put simply, the frame problem, first posed by McCarthy and Hayes [22], is the problem of characterizing what does *not* change when an action is performed. The ramification problem, so named by Finger [5], is the problem of characterizing the indirect effects of actions; a problem that can arise when a theory of action is integrated with a set of state constraints. Finally, the qualification problem, also

[2] We consider an agent to be a human, a computer program, a robot, or nature.

attributed to McCarthy [20], is the problem of characterizing the preconditions for actions; a problem that is aggravated by the existence of state constraints. Shanahan outlined three criteria for evaluating proposed solutions to the frame problem [40]. These criteria apply equally well to the ramification and qualification problems. Most fundamental is the criterion of *representational parsimony*, which requires that the representation of the solution be compact. Second is the criterion of *expressive flexibility*, which requires that the solution be sufficiently robust to deal with complex domain features, e.g., nondeterminism or concurrency. Finally the solution should be *elaboration tolerant* [21], that is, intuitively, the representation should be easily amenable to the addition of new information. A number of researchers have examined the frame, ramification and qualification problems over the years (e.g. [4,6,14,34,39,40,42], to name but a few); however a general solution has proven elusive.

In this paper, we address the general problem of integrating a representation of action with an existing set of state constraints by exploiting the language of the situation calculus and integrating a situation calculus representation of action with a set of first-order logic state constraints. This paper presents two major technical contributions to this end. First, for an arguably common class of state constraints, we provide an axiomatic closed-form solution to the frame and ramification problems. Our closed-form solution is captured by a parsimonious set of first-order logic axioms, that compose part of the domain axiomatization. We contrast this to solutions requiring nonmonotonic reasoning or a non-classical consequence relation. A closed-form solution such as ours is appealing for time-critical applications, as well as for tasks such as diagnosis, where solving the frame and ramification problems is not an end in itself. Our 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. A shortcoming in the justification of our closed-form solution is that it relies on an informal appeal to a completeness assumption and a causal interpretation of the material implication connective. To overcome these shortcoming, the second major contribution of this paper is to provide independent semantic justification for our solution.

The paper is organized as follows. In Section 2 we overview the specific situation calculus language we employ. In Section 3 we describe our starting point, a domain axiomatization that includes both state constraints and a situation calculus representation of action, but that does not address the frame, ramification or qualification problems. This is illustrated with respect to a simplified power plant feedwater system, which we refer to throughout the paper. In Section 4 we examine the ramification problem in more detail, showing that a previous solution to the frame and ramification problems in our language is not sufficiently discriminating to capture the *intended interpretation* of our domain axiomatization, and outlining our intuitions for a solution. In Section 4.2 we describe our proposal for a closed-form solution to the frame and ramification problems for a class of syntactically restricted state constraints, which we contend occur commonly in applications of AI. The solution comprises a simple syntactic manipulation which compiles ramification constraints and effect axioms into a set of successor state axioms, under a completeness assumption and a causal interpretation of the material implication connective. These successor state axioms capture the intended interpretation of our theory. To complete the solution to our problem, Section 5 outlines a solution to the qualification problem which appeals to existing results [15], compiling our qualification constraints, necessary conditions for action and successor state axioms into action precondition axioms. Section 6 discusses some advantages and disadvantages of our approach.

In Section 7 we provide important independent semantic justification for the solution to the frame and ramification problems, presented in the first half of the paper. We first define a prioritized minimization policy following the intuition exploited 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 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 [12] 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 [33]. This provides further justification for his solution to the frame problem. The paper concludes with a discussion of related work and a summary of our contributions.

2. The situation calculus

The situation calculus was first proposed by John McCarthy in the early 1960s as a logical representation scheme for reasoning about action and change [19]. 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 includes a set of foundational axioms, Σ_{found} which establish properties of our situations and situation tree ([15], and more recently [32]). Included in these axioms is the 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 are currently extending our results to more expressive dialects of the situation calculus (e.g., [35]).

Throughout this paper, we adopt the following notational convention. All formulae are universally quantified with maximum scope, unless stated otherwise. Variables begin with lower-case letters and constants begin with upper-case letters.

3. Domain axiomatization: An example

Once again, the problem we address in this paper assumes the existence of a set of system state constraints and our task is to integrate 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 [25] for such a discussion), and assume that our axiomatizer has given us a situation calculus domain axiomatization comprising the following sets of axioms:

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

where

- T_{SC} is a set of state constraints, comprised of T_{ram}, T_{qual}, and T_{domain}:
  - T_{ram} is a set of ramification constraints,
  - T_{qual} is a set of qualification constraints,
  - T_{domain} is a set of other domain constraints,
- T_{ef} is a set of effect axioms,
- T_{nec} is a set of axioms describing the necessary conditions for actions,
- T_{UNA} is a set of unique names axioms for actions, and
- T_{S_0} is a set of axioms describing what is known of the initial situation, S_0.

We describe these sets of axioms in further detail below. This axiomatization, while combining state constraints and a representation of action, does *not* solve the frame, ramification and qualification problems, and herein lies the problem we address in this paper.

We illustrate the form of these axioms, and many of the concepts in this paper with a simplified power plant feedwater system, depicted in Fig. 1. This example was extracted from a real-world diagnosis problem [27]. 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, (*Wtr_entering_hdr*), which in turn provides water to the boiler, producing steam. Alternately, the header can be filled manually (*Mnl_filling*). 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 disrupted. The system also contains lights and an alarm which is triggered under certain conditions. The predicate *Ok* is used throughout this example to designate that the component is operating normally. This is an artifact of the diagnostic problem solving task from which this example was drawn and has no bearing on our solution to the frame and ramification problems. The axiomatization of the power plant feedwater systems presented in this paper was designed to be the simplest representation that would suffice to illustrate important

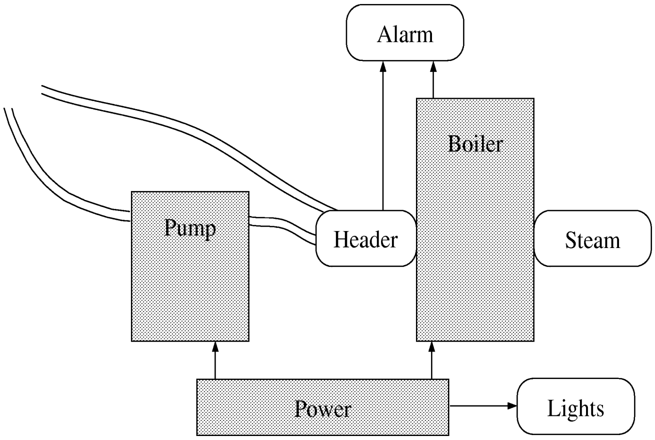

Fig. 1. Power plant feedwater system.

concepts. In order to reduce the number of literals and fluents in our axiomatization, we have elected to violate the *no-function-in-structure* principle often followed by the model-based reasoning community (e.g., [2]). Nevertheless, nothing in our proposed representation scheme precludes us from following this principle. For a more extensive axiomatization, see [25].

In order to describe the syntactic form of these axioms, we need the following definition of a simple formula, following [15].

Definition 1 (*Simple formula*). 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.

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}. The differentiation of these constraints into different subsets reflects the role that they play in the context of a theory of actions.

T_{ram}, the set of ramification constraints, constrains the indirect effects of actions. For each fluent F in our language, we may have both positive and negative ramification constraints of the following syntactic form.

$$\upsilon_F^+(\vec{x}, s) \supset F(\vec{x}, s), \tag{2}$$

$$\upsilon_F^-(\vec{x}, s) \supset \neg F(\vec{x}, s), \tag{3}$$

where $\upsilon_F^+(\vec{x}, s)$ and $\upsilon_F^-(\vec{x}, s)$ are simple formulae whose variables are among $\vec{x}, a, s$. These ramification constraints are intended to be interpreted as causal if–then rules. I.e., if $\upsilon_F^+(\vec{x}, s)$ is true, then it causes $F(\vec{x}, s)$ to be true, and similarly, if $\upsilon_F^-(\vec{x}, s)$ is true, then it causes $\neg F(\vec{x}, s)$ to be true. Hence, we say that these ramification constraints *causally* *influence* fluent F. This *directional* or *causal* interpretation of the material implication connective is stronger than the classical interpretation of $\supset$. In the sections to follow, we shall see how to enforce this intended interpretation. The set of ramification constraints for our feedwater example is as follows.

$$Ok(Power, s) \wedge Ok(Pump, s) \wedge On(Pump, s) \supset Wtr\_entering\_hdr(s), \tag{4}$$

$$Mnl\_filling(s) \supset Wtr\_entering\_hdr(s), \tag{5}$$

$$\begin{aligned} &Wtr\_entering\_hdr(s) \wedge Ok(Power, s) \wedge Ok(Boiler, s) \\ &\qquad \wedge On(Boiler, s) \supset Steam(s), \end{aligned} \tag{6}$$

$$\begin{aligned} &\neg(Wtr\_entering\_hdr(s) \wedge Ok(Power, s) \wedge Ok(Boiler, s) \\ &\qquad \wedge On(Boiler, s)) \supset \neg Steam(s). \end{aligned} \tag{7}$$

Axiom (4) states that if the power and pump are operating normally and if the pump is on, then it causes water to be entering the header. If the pump were off and we performed an action to turn it on, this axiom is intended to dictate that the indirect effect of turning on the pump is that water will be entering the header.

T_{qual}, the set of qualification constraints further constrains when actions are possible to perform. The set of qualification constraints for our feedwater example is as follows.

$$\neg(On(Pump, s) \wedge Mnl\_filling(s)). \tag{8}$$

This axiom states that it is impossible for the pump to be on and the header to be manually filling in the same situation. Thus, if the system were manually filling and we wanted to turn on the pump, this axiom would preclude us from doing so, because the resulting situation would violate this state constraint.

T_{domain}, the set of domain constraints, plays no additional role in the context of a theory of action. They simply serve to constrain the state of the system. The set of domain constraints for our feedwater example is as follows.

$$Power \neq Pump \neq Boiler. \tag{9}$$

This completes the axiomatization of our state constraints. Actions are axiomatized as a set of effect axioms T_{ef}, necessary conditions for actions T_{nec}, and unique names axioms for actions T_{UNA}, following the notation originally introduced in [34].

T_{ef}, the set of effect axioms, describes the changes in the truth values of fluents as a direct 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.

$$Poss(a, s) \wedge \gamma_F^+(\vec{x}, a, s) \supset F(\vec{x}, do(a, s)), \tag{10}$$

$$Poss(a, s) \wedge \gamma_F^-(\vec{x}, a, s) \supset \neg F(\vec{x}, do(a, s)), \tag{11}$$

where $\gamma_F^+(\vec{x}, a, s)$ and $\gamma_F^-(\vec{x}, a, s)$ are simple formulae whose variables are among $\vec{x}, a, s$. As with the ramification constraints, we say that these effect axioms *causally influence* fluent F.

The following axioms compose T_{ef} for our feedwater example.

$$Poss(a,s) \wedge a = Turn\_on(x) \supset On(x, do(a,s)), \tag{12}$$
$$Poss(a,s) \wedge a = Turn\_off(x) \supset \neg On(x, do(a,s)), \tag{13}$$
$$Poss(a,s) \wedge a = Start\_mnl\_fill \supset Mnl\_filling(do(a,s)), \tag{14}$$
$$Poss(a,s) \wedge a = Stop\_mnl\_fill \supset \neg Mnl\_filling(do(a,s)), \tag{15}$$
$$Poss(a,s) \wedge a = Disrupt\_siphon \supset \neg Wtr\_entering\_hdr(do(a,s)), \tag{16}$$
$$Poss(a,s) \wedge a = Blow(x) \supset \neg Ok(x, do(a,s)), \tag{17}$$
$$Poss(a,s) \wedge a = Burn\_out(x) \supset \neg Ok(x, do(a,s)), \tag{18}$$
$$Poss(a,s) \wedge a = Fail(x) \supset \neg Ok(x, do(a,s)), \tag{19}$$
$$Poss(a,s) \wedge a = Fix(x) \supset Ok(x, do(a,s)), \tag{20}$$
$$Poss(a,s) \wedge a = Aux\_pwr \supset Ok(Power, do(a,s)). \tag{21}$$

Axiom (12) states that if action a is possible in situation s, and a is the $Turn\_on(x)$ action, then x 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 individual actions to be performed. For each action prototype A in our language, necessary conditions are of the following form.

$$Poss(A(\vec{x}), s) \supset \pi_A^i, \tag{22}$$

where π_A^i is a simple formula with respect to s, whose free variables are among $\vec{x}, s$.

The following axioms compose T_{nec} for our feedwater example.

$$Poss(Turn\_on(x), s) \supset x = Pump \vee x = Boiler, \tag{23}$$
$$Poss(Turn\_on(x), s) \supset \neg On(x, s), \tag{24}$$
$$Poss(Turn\_off(x), s) \supset x = Pump \vee x = Boiler, \tag{25}$$
$$Poss(Turn\_off(x), s) \supset On(x, s), \tag{26}$$
$$Poss(Start\_mnl\_fill, s), \tag{27}$$
$$Poss(Stop\_mnl\_fill, s), \tag{28}$$
$$Poss(Disrupt\_siphon, s) \supset \neg Mnl\_filling(s), \tag{29}$$
$$Poss(Disrupt\_siphon, s) \supset \neg On(Pump, s), \tag{30}$$
$$Poss(Blow(x), s) \supset x = Boiler, \tag{31}$$
$$Poss(Blow(x), s) \supset On(x, s), \tag{32}$$
$$Poss(Blow(Boiler), s) \supset \neg Wtr\_entering\_hdr(s), \tag{33}$$
$$Poss(Burn\_out(x), s) \supset x = Pump, \tag{34}$$
$$Poss(Burn\_out(x), s) \supset On(x, s), \tag{35}$$
$$Poss(Fail(x), s) \supset x = Power, \tag{36}$$
$$Poss(Fail(x), s) \supset Ok(x, s), \tag{37}$$

$$Poss(Fix(x), s) \supset \neg On(x, s), \tag{38}$$

$$Poss(Fix(x), s) \supset x = Power \vee x = Pump \vee x = Boiler, \tag{39}$$

$$Poss(Aux\_power, s). \tag{40}$$

Axioms (29) and (30) state that if it is possible to disrupt the siphon, then the header must not be manual filling and the pump must not be on. Note that many of the actions have no necessary conditions and thus are always possible to perform.

T_{UNA} is the set of unique names axioms for actions. They are of the form of (41) and (42). For different action prototypes A and A':

$$A(x_1, \ldots, x_n) \neq A'(y_1, \ldots, y_m), \tag{41}$$

$$A(x_1, \ldots, x_n) = A(y_1, \ldots, y_n) \supset x_1 = y_1 \wedge \cdots \wedge x_n = y_n. \tag{42}$$

Axiom (41) states that every action name refers to a distinct action. Axiom (42) states that identical actions have identical arguments.

The following axioms compose T_{UNA} for our feedwater example.

$$\begin{aligned} Turn\_on(x_1) &\neq Turn\_off(x_2) \neq Start\_mnl\_fill \neq Stop\_mnl\_fill \\ &\neq Disrupt\_siphon \neq Blow(x_3) \neq Burn\_out(x_4) \\ &\neq Fail(x_5) \neq Fix(x_6) \neq Aux\_power, \end{aligned} \tag{43}$$

$$Turn\_on(x) = Turn\_on(y) \supset x = y, \tag{44}$$

$$Turn\_off(x) = Turn\_off(y) \supset x = y, \tag{45}$$

$$Burn\_out(x) = Burn\_out(y) \supset x = y, \tag{46}$$

$$Blow(x) = Blow(y) \supset x = y, \tag{47}$$

$$Fail(x) = Fail(y) \supset x = y, \tag{48}$$

$$Fix(x) = Fix(y) \supset x = y. \tag{49}$$

This completes the axiomatization of actions.

T_{S_0} is the initial database. It captures what is known of the initial state of the world. T_{S_0} need not be complete, and usually isn't. The following axioms might compose T_{S_0} for our feedwater example.

$$\begin{aligned} &Ok(Power, S_0) \wedge Ok(Pump, S_0) \wedge Ok(Boiler, S_0) \wedge \neg On(Boiler, S_0) \\ &\quad \wedge \neg On(Pump, S_0) \wedge \neg Wtr\_entering\_hdr(S_0) \wedge \neg Mnl\_filling(S_0). \end{aligned} \tag{50}$$

4. The frame and ramification problems

In the previous section, we illustrated a domain axiomatization in the situation calculus that combined state constraints and a representation of action. Once again, this domain axiomatization comprises the set of axioms in (1). The job of the axiomatizer is done, however we observe that these axioms do not provide a solution to the frame, ramification and qualification problems. In this section, we explain the problems presented by the existing axiomatization and propose a solution to the frame and ramification problems for what we argue to be a common class of state constraints. The qualification problem is discussed in a subsequent section.

We adopt the view of Reiter and others (e.g., [28,33,39]) that successor state axioms and action precondition axioms provide an effective solution to the frame and ramification problems, and the qualification problem, respectively, because they are axiomatic, monotonic and generally parsimonious. Indeed, Lin and Reiter [15] provided a semantic definition of a solution to the frame and ramification problems for our situation calculus language. The definition was based on minimal model semantics and a correspondence was identified to successor state axioms. Unfortunately, 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 constraints 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 ramification constraint (4), i.e.,

$$Ok(Power, s) \wedge Ok(Pump, s) \wedge On(Pump, s) \supset Wtr\_entering\_hdr(s).$$

Assume the effect axioms are as defined in the previous section and assume for the sake of simplicity that $\forall a, s.Poss(a, s)$, i.e., that all actions are possible in all situations. Further assume that everything is off, and everything is operating normally in the initial situation. In particular,

$$Ok(Power, S_0) \wedge Ok(Pump, S_0) \wedge \neg On(Pump, S_0) \wedge \neg Wtr\_entering\_hdr(S_0).$$

Now 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 (12), we infer that $On(Pump, S_1)$. What does ramification constraint (4) tell us about the indirect effects of this action? Recall that ramification constraint (4) is logically equivalent to the following axiom.

$$\neg Ok(Power, s) \vee \neg Ok(Pump, s) \vee \neg On(Pump, s) \vee Wtr\_entering\_hdr(s) \tag{51}$$

which holds for situation S_0, but does not hold in situation S_1, if we persist the truth status of all fluents, except $\neg On(Pump, S_0)$, which becomes $On(Pump, S_1)$. We must restore the satisfiability by changing the truth value of other fluents.

The intuition behind solutions to the frame problem is often to maximize the persistence of the truth values of fluents between situations. I.e., don't change the truth value of a fluent unless you are forced to do so to maintain satisfiability. Lin and Reiter's minimization policy [15] is no different. If we maximize the persistence of fluents while maintaining the satisfiability of (51), we produce three minimal models. The relevant portions of the models are as follows.

Ok(Power, S1)	Ok(Pump, S1)	On(Pump, S1)	Wtr-entering-hdr(S1)	(52)
¬Ok(Power, S1)	Ok(Pump, S1)	On(Pump, S1)	¬Wtr-entering-hdr(S1)	(53)
Ok(Power, S1)	¬Ok(Pump, S1)	On(Pump, S1)	¬Wtr-entering-hdr(S1)	(54)

Clearly, the intended model is (52). We intend that turning on the pump results in water entering the header. It does not result in a power supply that is not Ok, nor in a pump that is not Ok. 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 explained above, we intend for positive and negative ramification constraints of the form of (2) and (3) to be interpreted as *if* $\upsilon_F^{+/-}(\vec{x}, s)$ *is true then it causes* $[\neg]F(\vec{x}, s)$*, respectively, to be true*. This directional interpretation of the material implication connective, combined with the notion of maximizing persistence eliminates models (53) and (54), while designating (52), as the unique minimal model. Unfortunately, such an interpretation of the material implication connective, $\supset$, is stronger than the classical interpretation. In solving the ramification problem we must enforce this intended interpretation, eliminating unintended models.

It is interesting to note that the situation calculus ontology and in particular the foundational axioms of the situation calculus already enforce such an interpretation of the implication connective within the situation calculus effect axioms. I.e., given a positive or negative effect axiom of the form of (10) or (11), if $Poss(a, s) \wedge \gamma_F^{[+/-]}(\vec{x}, s)$ is true, then $[\neg]F(\vec{x}, do(a, s))$ is caused to be true. Since the situation tree cannot evolve from $do(a, s)$ to s (as defined by the foundational axioms) there are no other interpretations.

The idea of imposing a stronger, directional interpretation on the material implication connective is not unique to this paper. The logic programming community has done this for some time. In logic programming terminology [11], a literal is *defined* in a rule or set of rules if it appears in the head of a rule or set of rules. Hence, in logic programming terminology, ramification constraints (4) and (5) and effect axiom (16) serve to define the fluent [¬]*Wtr_entering_hdr*. Logic programs realize this directional interpretation by proceduralizing a type of minimization which causes literals contained in the body of a rule to be minimized at a higher priority than those that appear in the head. Some of the literature on reasoning about action (e.g., [14,18,42]) also captures the intuition of exploiting directional influence by providing axiomatizations of state constraints with an explicit non-classical causal connective or a distinguished *Causes* predicate. We will discuss the relationship of this work to ours, at the end of this paper.

In the section to follow, we show how to transform our situation calculus domain axiomatization (1), into another situation calculus domain axiomatization that captures all and only its intended interpretation.

4.2. The solution

In this section we provide a transformation procedure that leads us to a closed-form solution to the frame and ramification problems for axiomatizations whose syntactic representation of ramification constraints and effect axioms collectively form what we refer to as a *solitary stratified theory*.

4.2.1. Preliminaries

Intuitively, a set of effect axioms and ramification constraints forms a solitary stratified theory if the directed graph representing the causal influence between fluents is acyclic. Fig. 2 illustrates the causal influence graph for our feedwater example. A solitary stratified theory separates fluents into a partition $\mathcal{L} = (\mathcal{L}_1, \mathcal{L}_2, \ldots, \mathcal{L}_n)$, and decomposes the axioms T into strata $(T_1, T_2, \ldots, T_n)$, such that the axioms that causally influence a fluent $F_i \in \mathcal{L}_i$ are placed in stratum T_i. Further, the fluents that participate in causally influencing F_i, i.e., the fluents in the antecedents of the ramification constraints and effect axioms for F_i, are all drawn from $\mathcal{L}_j$, $j < i$. The stratification of a solitary stratified theory is not unique and its determination is easily automated.

The notion of a "solitary stratified theory" is derivative of both solitary theories [12] and stratified logic programs (e.g., [11]). For those familiar with stratified logic programs, a solitary stratified theory is a stratified logic program that allows negation in the head of rules. It further differs from a stratified logic program in that the criterion that defines a stratum applies a strictly $<$ relation, rather than $\leqslant$. The intuitive description above should be sufficient for the reader to understand the closed-form solution presented in this section. Nevertheless, we provide the following more formal definition of a solitary stratified theory. The terminology and notation is derivative of definitions in [12] in order to facilitate our semantic justification in Section 7.

Definition 2 (*Solitary stratified theory*). Suppose T is a theory in the language of the situation calculus with domain fluents, $\mathcal{L}$. T is a solitary stratified theory with stratification $(T_1, T_2, \ldots, T_n)$, and partition $(\mathcal{L}_1, \mathcal{L}_2, \ldots, \mathcal{L}_n)$, where $\mathcal{L}_1 \cup \mathcal{L}_2 \cup \cdots \cup \mathcal{L}_n = \mathcal{L}$, if T is the union

$$T_1 \cup T_2 \cup \cdots \cup T_n$$

of sets of axioms T_i where for each stratum, T_i is solitary with respect to $\mathcal{L}_i$; that is, each T_i can be written as the union

$$(\mathcal{D}_i \leqslant \neg\mathcal{L}_i) \cup (\mathcal{E}_i \leqslant \mathcal{L}_i),$$

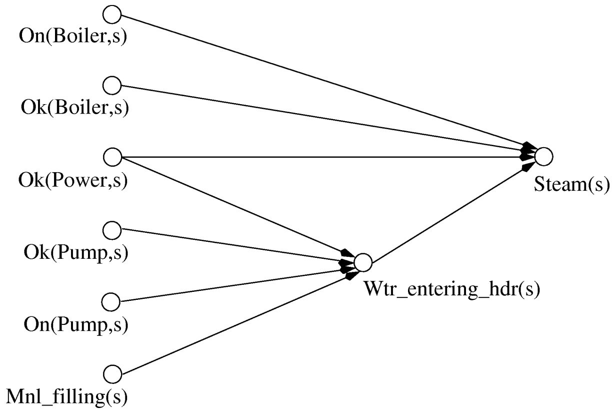

Fig. 2. Causal influence in feedwater example.

where

(1) $\mathcal{L}_i$ is the set of fluents, F_i, such that F_i is only causally influenced by axioms in T_i;

(2) $\mathcal{D}_i \leqslant \neg\mathcal{L}_i$ is an abbreviation for a set of formulae of the form

$$(D_i \supset \neg F_i),$$

one for each fluent $F_i \in \mathcal{L}_i$, where each D_i is a formula containing no fluents drawn from $\mathcal{L}_i \cup \cdots \cup \mathcal{L}_n$;

(3) $\mathcal{E}_i \leqslant \mathcal{L}_i$ is an abbreviation for a set of formulae of the form

$$(E_i \supset F_i),$$

one for each fluent $F_i \in \mathcal{L}_i$, where each E_i is a formula containing no fluents drawn from $\mathcal{L}_i \cup \cdots \cup \mathcal{L}_n$.

Example 1. In our feedwater example, $T = T_{ram} \cup T_{ef}$ is a solitary stratified theory with partition $(\mathcal{L}_1, \mathcal{L}_2, \mathcal{L}_3)$ and stratification (T_1, T_2, T_3), where

- $\mathcal{L}_1 = \{On(Pump, s), On(Boiler, s), Ok(Pump, s), Ok(Boiler, s), Ok(Power, s), Mnl\_filling(s)\}$.
  T_1 comprises all the effect axioms except (16). In particular, $\mathcal{D}_1 \leqslant \neg\mathcal{L}_1$ comprises the set of negative effect axioms, i.e., (13), (15), (17), (18) and (19), and $\mathcal{E}_1 \leqslant \mathcal{L}_1$ comprises the set of positive effect axioms, i.e., (12), (14), (20) and (21).
- $\mathcal{L}_2 = \{Wtr\_entering\_hdr(s)\}$.
  T_2 comprises ramification constraints (4) and (5) and effect axiom (16).
- $\mathcal{L}_3 = \{Steam(s)\}$.
  T_3 comprises ramification constraints (6) and (7).

With our definition of solitary stratified theory in hand, we are now prepared to present a solution to the frame and ramification problems.

4.2.2. A closed-form solution

In what follows, we present a syntactic manipulation procedure that results in a closed-form solution to the frame and ramification problems for solitary stratified theory $T = T_{ef} \cup T_{ram}$. The procedure takes effect axioms, ramification constraints and the (causal) partition of their fluents as input, and under a completeness assumption, transforms them into a set of successor state axioms. The ideas presented in this section draw some intuition from Reiter's solution to the frame problem *without* state constraints [33]. Our solution is predicated on our notion of causal influence and on an appeal to a completeness assumption that enables us to generate explanation closure axioms (e.g., [28,39]).

Transformation Procedure. Let $T = T_{ram} \cup T_{ef}$ be a solitary stratified theory, with partition $(\mathcal{L}_1, \mathcal{L}_2, \ldots, \mathcal{L}_n)$ and stratification $(T_1, T_2, \ldots, T_n)$. Let T_{ef} be comprised of positive and negative effect axioms of the form of (10) and (11). Let T_{ram} be comprised of positive and negative ramification constraints of the form of (2) and (3). Note that henceforth, action and predicate arguments, $\vec{x}$, will not be explicitly represented in canonical formulae.

Step 1. *From the effect axioms, T_{ef}, and the ramification constraints, T_{ram}, generate positive and negative general causal influence axioms, of the following form.*

General Causal Influence Axioms. For every fluent $F_i \in \mathcal{L}_i$,

$$[Poss(a,s) \wedge](\gamma^{+}_{F_i}(a,s) \vee \upsilon^{+}_{F_i}(do(a,s))) \supset F_i(do(a,s)), \tag{55}$$

$$[Poss(a,s) \wedge](\gamma^{-}_{F_i}(a,s) \vee \upsilon^{-}_{F_i}(do(a,s))) \supset \neg F_i(do(a,s)), \tag{56}$$

where $[Poss(a,s) \wedge]$ indicates that $Poss(a,s)$ may or may not occur.

Example 2. Positive and negative general causal influence axioms for the fluent $On(x,s) \in \mathcal{L}_1$ are

$$Poss(a,s) \wedge a = Turn\_on(x) \supset On(x, do(a,s)), \tag{57}$$

$$Poss(a,s) \wedge a = Turn\_off(x) \supset \neg On(x, do(a,s)). \tag{58}$$

Positive and negative general causal influence axioms for $Wtr\_entering\_hdr(s) \in \mathcal{L}_2$ are

$$\begin{aligned}&(Ok(Power, do(a,s)) \wedge Ok(Pump, do(a,s)) \wedge On(Pump, do(a,s)))\\ &\qquad \vee Mnl\_filling(do(a,s)) \supset Wtr\_entering\_hdr(do(a,s)),\end{aligned} \tag{59}$$

$$Poss(a,s) \wedge a = Disrupt\_siphon \supset \neg Wtr\_entering\_hdr(do(a,s)). \tag{60}$$

Step 2. *Make the following causal completeness assumption.*

Causal Completeness Assumption. 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 positive and negative general causal influence axioms for fluent F.

Step 3. *From the causal 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.

Explanation Closure Axioms. For every fluent $F_i \in \mathcal{L}_i$,

$$Poss(a,s) \wedge F_i(s) \wedge \neg F_i(do(a,s)) \supset \gamma^{-}_{F_i}(a,s) \vee \upsilon^{-}_{F_i}(do(a,s)), \tag{61}$$

$$Poss(a,s) \wedge \neg F_i(s) \wedge F_i(do(a,s)) \supset \gamma^{+}_{F_i}(a,s) \vee \upsilon^{+}_{F_i}(do(a,s)). \tag{62}$$

Step 4. *From the positive and negative general causal influence axioms and the explanation closure axioms, define intermediate successor state axioms for each fluent F_i.*

The successor state axioms are distinguished as *intermediate* because in the next step, we simplify them through a further syntactic transformation.

Intermediate Successor State Axioms. For every fluent $F_i \in \mathcal{L}_i$,

$$Poss(a,s) \supset [F_i(do(a,s)) \equiv \Phi^*_{F_i}], \tag{63}$$

where

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

The set of intermediate successor state axioms is defined as the set,

$$T_{ISS} = \bigcup_{i=1,\dots,n} T_{ISS_i},$$

where T_{ISS_i} is the set of intermediate successor state axioms for fluents $F_i \in \mathcal{L}_i$.

The formulae (63) and (66) below may be understood as follows,

$$\begin{aligned} Poss(a,s) \supset [F_i(do(a,s)) \equiv \\ & \textit{an action made it true} \\ & \vee\, \textit{a ramification made it true} \\ & \vee\, F_i \textit{ was already true in } s \\ & \quad \wedge\, \textit{neither an action nor a ramification made it false}]. \end{aligned}$$

Example 3. Intermediate successor state axioms for the fluent $On(x,s) \in \mathcal{L}_1$ and for the fluent $Wtr\_entering\_hdr(s) \in \mathcal{L}_2$ are as follows.

$$\begin{aligned} Poss(a,s) \supset [On(x, do(a,s)) \equiv a = Turn\_on(x) \\ \vee\, (On(x,s) \wedge a \neq Turn\_off(x))], \end{aligned} \tag{64}$$

$$\begin{aligned} & Poss(a,s) \supset [Wtr\_entering\_hdr(do(a,s)) \equiv \\ & \quad Mnl\_filling(do(a,s)) \\ & \quad \vee\, (Ok(Power, do(a,s)) \wedge Ok(Pump, do(a,s)) \wedge On(Pump, do(a,s))) \\ & \quad \vee\, Wtr\_entering\_hdr(s) \wedge a \neq Disrupt\_siphon]. \end{aligned} \tag{65}$$

At this point we could consider ourselves done. Indeed, the intermediate successor state axioms provide a solution to the frame and ramification problems. They capture only the intended interpretation of our effect axioms and ramification constraints. In many instances, we may actually stop with this representation, which is relatively compact. Note however that the intermediate successor state axioms can be further compiled. In particular, observe the intermediate successor state axiom defining the conditions underwhich $F_i(do(a,s))$ will be true is itself defined in terms of other fluents relativized to situation $do(a,s)$. For example, the successor state axiom for *Wtr_entering_hdr* is defined in terms of $Mnl\_filling(do(a,s))$, $Ok(Power, do(a,s))$, etc. Each of these fluents is itself defined in other intermediate successor state axioms. In the final step of our transformation procedure, we use regression rewriting to rewrite these intermediate successor state axioms so that our final successor state axioms are defined in terms of simple formulae, and hence contain no reference to fluents relativized to $do(a,s)$.

Definition 3 (*Regression* (*e.g.*, [33,43]). Regression is a recursive rewriting procedure used here to reduce the nesting of the do function in situation terms. If F is a fluent with (intermediate) successor state axiom

$$Poss(a,s) \supset F(\vec{x}, do(a,s)) \equiv \Phi_F(\vec{x}, a, s)$$

in T_{SS} then the regression of $F(t_1, \ldots, t_n, do(a,s))$,

$$\mathcal{R}_{SS}[F(t_1, \ldots, t_n, do(\alpha, \sigma))] = \Phi_F \mid^{x_1,\ldots,x_n,a,s}_{t_1,\ldots,t_n,\alpha,\sigma} .$$

Regression generalizes over formulae as one would expect. See Appendix A or [25] for a detailed description.

The challenge is that we want this rewriting to terminate in a set of simple formulae. For example, regressing the following two intermediate successor state axioms

$$Poss(a,s) \supset [F(do(a,s)) \equiv G(do(a,s))],$$

$$Poss(a,s) \supset [G(do(a,s)) \equiv F(do(a,s))]$$

will never terminate. The merit of our solution is that for solitary stratified theories, regression rewriting will terminate and it will lead to final successor state axioms defined in terms of simple formulae. In Step 5, we describe the final form of our successor state axioms. In Theorem 1 we prove that regression is guaranteed to terminate and to be defined in terms of simple formulae.

Step 5. *By regressing the intermediate successor state axioms, generate* (*final*) *successor state axioms.*

Successor State Axioms. For every fluent $F_i \in \mathcal{L}_i$,

$$Poss(a,s) \supset [F_i(do(a,s)) \equiv \Phi_{F_i}], \tag{66}$$

where unlike $\Phi^*_{F_i}$, Φ_{F_i} is a simple formula of the following form,

$$\begin{aligned}
\Phi_{F_i} &= \mathcal{R}^{i-1}_{SS}[\Phi^*_{F_i}] \\
&\equiv \mathcal{R}^{i-1}_{SS}[\gamma^+_{F_i}(a,s) \vee \upsilon^+_{F_i}(do(a,s)) \vee (F_i(s) \wedge \neg(\gamma^-_{F_i}(a,s) \vee \upsilon^-_{F_i}(do(a,s))))] \\
&\equiv \gamma^+_{F_i}(a,s) \vee \mathcal{R}^{i-1}_{SS}[\upsilon^+_{F_i}(do(a,s))] \\
&\qquad\qquad \vee \big(F(s) \wedge \neg(\gamma^-_{F_i}(a,s) \vee \mathcal{R}^{i-1}_{SS}[\upsilon^-_{F_i}(do(a,s))])\big).
\end{aligned} \tag{67}$$

$\mathcal{R}^{i-1}_{SS}[\phi]$ is the repeated 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 4. Axiom (64) is both the intermediate and the final successor state axiom for fluent $On(Pump, s)$. The intermediate successor state axiom (65) transforms into the following successor state axiom under regression.

$$\begin{aligned}
&Poss(a,s) \supset [Wtr_entering_hdr(do(a,s)) \equiv \\
&\quad a = Start_mnl_fill \\
&\quad \vee (Mnl_filling(s) \wedge a \neq Stop_mnl_fill) \\
&\quad \vee [(a \neq Fail(Power) \wedge (Ok(Power,s) \vee a = Aux_power \vee a = Fix(Power))) \\
&\qquad \wedge (a \neq Burn_out(Pump) \wedge (Ok(Pump,s) \vee a = Fix(Pump))) \\
&\qquad \wedge (a = Turn_on(Pump) \vee (On(Pump,s) \wedge a \neq Turn_off(Pump)))] \\
&\quad \vee (Wtr_entering_hdr(s) \wedge a \neq Disrupt_siphon)].
\end{aligned} \tag{68}$$

Proposition 1. *Suppose $T = T_{ef} \cup T_{ram}$ is a solitary stratified theory with intermediate successor state axioms T_{ISS} and (final) successor state axioms T_{SS} as defined above. Further suppose,*

- *$\mathcal{R}_{ISS}[\phi]$ denotes the repeated regression of ϕ under T_{ISS},*
- *$\mathcal{R}_{ISS_i}[\phi]$ denotes the repeated regression of ϕ under T_{ISS_i},*
- *$\mathcal{R}^i_{ISS}[\phi]$ denotes the repeated regression of ϕ under $T_{ISS_1}, T_{ISS_2}, \ldots, T_{ISS_i}$.*

Corresponding terminology holds for successor state axioms T_{SS}.

Then for every fluent $F_1 \in \mathcal{L}_1$, the successor state axiom for F_1 is identical to its intermediate successor state axioms, and is of the following general form.

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

More generally, $T_{ISS_1} \equiv T_{SS_1}$.

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

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

Since T is a solitary stratified theory, if $\Phi^*_{F_i}$ mentions fluents relativized to situation $do(a,s)$, then those fluents are drawn from $\{\mathcal{L}_1, \ldots, \mathcal{L}_{i-1}\}$. Also observe that Φ_{F_1} is a simple formula. Hence it follows that if $\mathcal{R}_{SS_{i-1}}[\Phi_{F_i}]$ mentions fluents relativized to situation $do(a,s)$, then those fluents are drawn from $\{\mathcal{L}_1, \ldots, \mathcal{L}_{i-2}\}$. More generally, it follows that $\mathcal{R}^{i-1}_{SS}[\Phi_{F_i}]$ is a simple formula.

Theorem 1. *Suppose $T = T_{ef} \cup T_{ram}$, is a solitary stratified theory with intermediate successor state axioms T_{ISS} and (final) successor state axioms T_{SS} as described above. Then for any fluent $F_i \in \mathcal{L}$, $\mathcal{R}^{i-1}_{SS}[F_i(do(a,s)]$ is a simple formula. More generally, for any successor state axiom of the form of* (66), *$\mathcal{R}^{i-1}_{SS}[\Phi_{F_i}]$ is a simple formula.*

The successor state axioms and the intermediate successor state axioms provide alternate closed-form solutions to the frame and ramification problems. In our axiomatization (1), we may replace T_{ram} and T_{ef} by $T^{S_0}_{ram}$, the ramification constraints relativized to situation S_0, and either T_{SS} or T_{ISS} (henceforth denoted $T_{[I]SS}$), the intermediate or final successor state axioms. In Section 7 we also show that this closed-form solution is conditioned on a consistency assumption that ensures that either an action is impossible to perform, or that the action cannot directly or indirectly make a fluent both true and false in the same situation.

5. The qualification problem

Our new domain axiomatization,

$$T_{UNA} \cup T_{[I]SS} \cup T_{S_0} \cup T_{ram}^{S_0} \cup T_{qual} \cup T_{domain} \cup T_{nec} \tag{69}$$

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 [15], 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}, under an assumption of domain closure on actions. Following their results, we add one more step to our procedure.

Step 6. *Define one action precondition axiom for each action prototype $A(\vec{x})$ as follows.*

Action Peconditon Axioms.

$$Poss(A(\vec{x}), s) \equiv \Pi_A \wedge \bigwedge_{C \in T_{qual}} \Pi_C, \tag{70}$$

where

$$\Pi_C \equiv \mathcal{R}_{SS}[C(do(A(\vec{x}), s))]. \tag{71}$$

$\Pi_A \equiv \pi_A^1 \vee \cdots \vee \pi_A^n$ for each π_A^i of (22) in T_{nec}. $\mathcal{R}_{SS}$ is the repeated regression operator under the successor state axioms, T_{SS}. Recall that following Proposition 1, this regression is equivalent to regression with the intermediate successor state axioms.

Example 5. Consider the qualification constraint (8),

$$\neg(On(Pump, s) \wedge Mnl\_filling(s)),$$

effect axioms (12) and (14),

$$Poss(a, s) \wedge a = Turn\_on(x) \supset On(x, do(a, s)),$$

$$Poss(a, s) \wedge a = Start\_mnl\_fill \supset Mnl\_filling(do(a, s)),$$

and necessary conditions for actions, (23), (24) and (27),

$$Poss(Turn\_on(x), s) \supset x = Pump \vee x = Boiler,$$

$$Poss(Turn\_on(x), s) \supset \neg On(x, s),$$

$$Poss(Start\_mnl\_fill, s).$$

The qualification constraint dictates that it is not possible to perform the action $Turn\_on(Pump)$ when the fluent $Mnl\_filling(s)$ holds, and similarly that it is not possible to perform the action $Start\_mnl\_fill$ when the fluent $On(Pump, s)$ holds.

The action precondition axioms for $Start\_mnl\_fill$ and $Turn\_on(x)$ following Step 6 of our procedure are:

$$
\begin{aligned}
Poss(Turn\_on(x), s) \equiv{} & (x = Pump \vee x = Boiler) \\
& \wedge \neg On(x, s) \\
& \wedge (x = Pump \supset \neg MnL\_filling(s)),
\end{aligned} \tag{72}
$$

$$
Poss(Start\_mnl\_fill, s) \equiv \neg On(Pump, s). \tag{73}
$$

The action precondition axioms provide a closed-form solution to the qualification problem under the assumption of domain closure on actions. 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. We also add 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 axiomatization which integrates our syntactically restricted state constraints and a representation of action, while solving the frame, ramification and qualification problems:

$$
T_{UNA} \cup T_{DCA} \cup T_{[I]SS} \cup T_{AP} \cup T_{S_0} \cup T_{SC}^{S_0} \cup T_{domain}, \tag{74}
$$

where

- T_{UNA} is a set of unique name axioms for actions.
- T_{DCA} is a domain closure axiom for actions.
- $T_{[I]SS}$ is either a set of intermediate or final successor state axioms, derived from T_{ef} and T_{ram} under a causal completeness assumption.
- T_{AP} is a set of action precondition axioms, derived from T_{nec}, T_{qual} and $T_{[I]SS}$ under a causal completeness assumption.
- T_{S_0} is a set of axioms describing what is known of the initial state of the world.
- $T_{SC}^{S_0}$ is the set of state constraints, T_{SC}, relativized to situation S_0.
- T_{SC} is a set of state constraints. It is comprised of T_{ram}, T_{qual}, and T_{domain}.
- T_{ram} is a set of ramification constraints.
- T_{qual} is a set of qualification constraints.
- T_{domain} is a set of domain constraints. These are the state constraints which are neither qualification constraints, nor ramification constraints.

Recall that in Section 1 we presented three criteria for evaluating solutions to the frame, ramification and qualification problems, following Shanahan [40]. We claim that our closed-form solution to the frame and ramification problems adheres to these criteria, as does Lin and Reiter's solution to the qualification problem, which we have adopted to complete our representation. We discuss the criteria with respect to our solution to the frame and ramification problems.

On the subject of representational parsimony, it was previously shown that successor state axioms provide a representationally parsimonious solution to the frame problem in the absence of the ramification problem [32]. In particular, the brute-force approach to addressing the frame problem requires the addition of $2 \times \mathcal{F} \times \mathcal{A}$ frame axioms (where $\mathcal{F}$

is the number of fluents in the language and $\mathcal{A}$ is the number of actions in the language). In contrast, a successor state axiom solution to the frame problem requires only $\mathcal{F}$ successor state axioms. When we use successor state axioms to solve the ramification problem as well, there will still only be $\mathcal{F}$ successor state axioms, however the length of each successor state axiom will be increased.

The most compact solution to the frame and ramification problems is provided by the *intermediate* successor state axioms. In this case, the length of each axiom is roughly proportional to the number of actions, and the number of ramification constraints that directly affect the truth value of the fluent, i.e., the number of effect axioms and ramification constraints for a fluent. Under the assumption that few actions and ramifications directly affect any individual fluent, the intermediate successor state axioms remain short, and the representation is compact. The intermediate successor state axioms also have the virtue of preserving the structure and compositionality of the representation—something that is important for model-based reasoning applications as well as for elaboration tolerance. In contrast, the (final) successor state axioms represent a compiled, and thus less compact, version of the intermediate successor state axioms. In particular, we still have $\mathcal{F}$ successor state axioms, but the length of each axiom will be proportional to the number of actions and ramification constraints that indirectly, as well as directly, affect a fluent. Arguably this is still quite small, however it will grow with the size of the stratification, and has the potential to explode. So, on the basis of parsimony, the intermediate successor state axioms are a superior representation. In practice, there may be instances where the advantages of the precompilation into final successor state axioms, outweigh those of parsimony. Such a trade-off between parsimony and the potential runtime computational advantages of precompilation is an issue best addressed with respect to the specific domain and application.

On the subject of the expressive flexibility criterion, we believe our successor state axiom solution will score well. While there is little in this paper to demonstrate this point, solutions to the frame problem based on successor state axioms have been extended to knowledge producing actions [38], to complex actions [10], and to aspects of continuous domains [29,35]. Preliminary investigation indicates that our solution to the ramification problem extends to these problems in the same straightforward manner.

Finally, on the subject of elaboration tolerance, our successor state axiom solution again scores well. As noted above, intermediate successor state axioms are more amenable to elaboration than final successor state axioms, but elaboration is straightforward and automatable in both cases. If new actions, effect axioms or ramification constraints are added to the theory, they can be incorporated by a simple rewrite of the successor state axioms for the affected fluents, following the syntactic form provided in (63) or (66).

One potential criticism of our closed-form solution is that we are relying on the syntactic form of our axiomatization. In particular, we are relying on the fact that the axiomatizer has written the ramification constraints so that the implication sign may be correctly interpreted as *causal influence*. This need not be the case. Note that for any such axiomatization, we may instead be given the causal relationship between fluents, i.e., the causal influence graph, or a separate data structure describing the causal influence, and use this to generate the stratification and hence the successor state axioms.

A second potential criticism is that our closed-form solution is predicated on a loose appeal to a completeness assumption. We address this in the section to follow by providing independent semantic justification for our closed-form solution.

A third potential criticism of our closed-form solution is that it is restricted to the class of solitary stratified theories. While we can make no definitive claims about the frequency of occurrence of solitary stratified theories in general, they appear to be common in the representations of engineering artifacts. In these systems, the causal influence between fluents often reflects physical connectivity of components and subcomponents.

To close our discussion, we'd like to make the important observation that 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 $T_{[I]SS}$ and T_{AP}, by implications [32]. The Lloyd–Topor transformation [17] 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 [31], thus the logic program produced from translation of our successor state axioms also has a perfect model semantics. This is discussed more thoroughly in [25].

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, and on a causal interpretation of the material implication connective. 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 causal influence ordering among fluents that induces a solitary stratified theory, 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 (66) 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 [15]. In Section 7.2, we observe that our minimization policy is equivalent to a particular instance of prioritized circumscription, where the prioritization is equivalent to the causal influence ordering. Through simple syntactic renaming and by exploiting results from Lifschitz on computing circumscription (e.g., [12]), 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 [33].

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 $ab_{F_i}(a,s)$, one for each fluent F_i in our domain axiomatization. The predicate $ab_{F_i}(a,s)$ is an abbreviation for $\neg[F_i(s) \equiv F_i(do(a,s))]$, i.e., it is *ab*-normal if F_i changes truth value from one situation to the next.

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 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 [16] and Lin and Reiter [15], 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 derived from the partitioning of fluents in our domain axiomatization into $\mathcal{L} = (\mathcal{L}_1, \mathcal{L}_2, \ldots, \mathcal{L}_n)$, according to causal influence. Under this partition, fluents in $\mathcal{L}_i$ are only causally influenced by fluents in $\mathcal{L}_j$, $j < i$. This dictates the following priority ordering for our minimization policy

$$Ab_1 > Ab_2 > \cdots > Ab_n,$$

where Ab_i is a tuple containing the abnormality predicate $ab_{F_i}(a,s)$ for each fluent $F_i \in \mathcal{L}_i$. $Ab_1(a,s)$ is assigned the highest priority for minimization, and $Ab_n(a,s)$ is assigned the lowest priority. Hence, the priority ordering corresponds to our causal influence ordering. This causal influence ordering can be articulated independently, or it can be communicated by an axiomatizer through the strategic placement of material implication connectives in the axioms of our theory, $T = T_{ef} \cup T_{ram}$, as illustrated earlier in this paper. The use of material implication to reflect causal influence results in a solitary stratified theory, with a stratification corresponding to our priority ordering.

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 the truth value of fluents in the resulting situation, $F_k(do(a,s))$, $k = 1, \ldots, n$. As we will see, this prioritized minimization policy captures our intended solution to the frame and ramification problems for solitary stratified theories.

To get a better grasp of the intuition behind this minimization policy, we may think of causal change as temporal change that we have abstracted away in our knowledge representation scheme. Hence, the causal ordering dictated by our causal influence graph, captured in the grouping of fluents in our solitary stratified theory, and exploited in the prioritization of our minimization policy, can be though of as an abstraction of a temporal ordering on the necessary propagation of change. With this intuition in mind, we can think of our minimization policy as *minimizing causal change* according to this implicit temporal ordering, or explicit causal ordering. Since changes in fluents in $\mathcal{L}_1$ cause changes in fluents in $\mathcal{L}_2, \ldots, \mathcal{L}_N$, and changes in fluents in $\mathcal{L}_1$ and $\mathcal{L}_2$ cause changes in fluents in $\mathcal{L}_3, \ldots, \mathcal{L}_n$, and so on, we can minimize overall change by minimizing change following this causal ordering, and hence minimize the propagation of change.

The definition of the minimization policy follows. At first glance, it may look complex and daunting, rather than simple and intuitive. We elected to express it this way to be mathematically precise, and to relate it to previous work on the frame and ramification problems. As will be shown in the section to follow, our nonmonotonic specification reduces to a simple prioritized circumscription, minimizing causal change following our causal influence ordering.

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

Definition 4 (*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 $ab_{F_i}(\vec{x}, a, s)$ is an abbreviation for $\neg[F_i(\vec{x}, s) \equiv F_i(\vec{x}, do(a, s))]$ 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 $F_i(\vec{x}, s)$, $i = 1, \ldots, n$,

$$\mathcal{M}', \sigma_s, \sigma_d \models F_i(\vec{x}, s) \quad \text{iff} \quad \mathcal{M}, \sigma_s, \sigma_d \models F_i(\vec{x}, s).$$

(b) For some i, $1 \leqslant i \leqslant n$,
$\mathcal{M}$ 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

$$\mathcal{M}', \sigma_s, \sigma_a, \sigma_d \models \neg ab_{F_i}(\vec{x}, a, s).$$

Definition 5 (*Minimal model*). $\mathcal{M}$ 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 [15].

Definition 6 (*Semantic specification*). Suppose

$$\Sigma = \Sigma_{found} \cup T_{UNA} \cup T_{ef} \cup T_{ram}, \tag{75}$$

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 [15]. 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* [15].

Remark 1. If $\mathcal{M}$ captures a solution to the frame and ramification problem for Σ as specified in Definition 6, then it also meets Lin and Reiter's general specification for a solution to the frame and ramification problem, as outlined in [15].

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 state constraints, 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 a closed-form solution, whereas Lin and Reiter's does not.

Next we demonstrate the relationship between our semantically specified solution to the frame and ramification problems and the successor state axioms we defined in (66). This relationship is predicated on a consistency assumption.

Assumption 1 (*Consistency Assumption*). For each fluent $F_i \in \mathcal{L}_i$ assume that:

$$\begin{aligned} T_{UNA} \cup T_{nec} \models (\forall a, s).Poss(a, s) \supset \\ \neg[(\gamma^{+}_{F_i}(a, s) \vee \mathcal{R}[\upsilon^{+}_{F_i}(do(a, s))]) \\ \wedge (\gamma^{-}_{F_i}(a, s) \vee \mathcal{R}[\upsilon^{-}_{F_i}(do(a, s))])], \end{aligned} \tag{76}$$

where $\mathcal{R}[\phi]$ is the regression of ϕ under successor state axioms T_{SS}.

This consistency assumption may be understood as follows. The unique names assumptions for actions and the necessary conditions for action, T_{UNA} and T_{nec} enforce that if an action is possible in a situation, i.e., $Poss(a, s)$, then

$$\begin{aligned} Poss(a, s) \supset \textit{It is not the case that both} \\ (\textit{an action or a ramification makes } F_i(do(a, s)) \textit{ true}) \\ \wedge (\textit{an action or a ramification makes } F_i(do(a, s)) \textit{ false}). \end{aligned}$$

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.

The following theorem states that under Consistency Assumption 1, if we replace the effect axioms, T_{ef} and ramification constraints, T_{ram} by the successor state axioms of (66), T_{SS} and the ramification constraints relativized to the initial situation, $T^{S_0}_{ram}$, 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 $S_0 \leqslant s$. (Recall that $\leqslant$ over situations is defined in Σ_{found}.) This enables us to exclude T_{ram} and T_{ef} from our theory, provided T_{SS} and $T^{S_0}_{ram}$ are present.

Theorem 2. *Suppose Σ is the theory defined in Definition* 6 *and T_{SS} is the set of successor state axioms derived from T_{ef} and T_{ram} of Σ as per* (66). *Further, assume Consistency Assumption* 1 *holds.*

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

$$\Sigma_{found} \cup T_{UNA} \cup T_{SS} \cup T^{S_0}_{ram} \models (\forall s).S_0 \leqslant s \supset C(s),$$

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

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

The following theorem proves that, under Consistency Assumption 1, 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 5.

Theorem 3. *Suppose Σ is the theory defined in Definition* 6 *and T_{SS} is the set of successor state axioms derived from T_{ef} and T_{ram} of Σ as per* (66). *Finally assume that Consistency Assumption* 1 *holds.*

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 6.

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 (66) 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 (66), and our nonmonotonic specification of a solution to the frame and ramification problems. In what follows we provide a variety of intermediate results that culminate in the main results, stated in Proposition 3 and Theorem 5.

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 into a simpler theory, Σ^*. The circumscription is then computed with respect to Σ^* by exploiting results of Lifschitz on computing circumscription (e.g., [12,13]). 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 $Wtr\_entering\_hdr(s)$, as originally defined in our feedwater example.

$$Poss(a,s) \wedge a = Disrupt\_siphon \supset \neg Wtr\_entering\_hdr(do(a,s)),$$

$$Ok(Power,s) \wedge Ok(Pump,s) \wedge On(Pump,s) \supset Wtr\_entering\_hdr(s),$$

$$Mnl\_filling(s) \supset Wtr\_entering\_hdr(s).$$

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.

$$Poss(a,s) \wedge Wtr\_entering\_hdr(s) \wedge \neg ab^-_{Wtr\_entering\_hdr}(a,s) \\ \supset Wtr\_entering\_hdr(do(a,s)),$$

$$Poss(a,s) \wedge \neg Wtr\_entering\_hdr(s) \wedge \neg ab^+_{Wtr\_entering\_hdr}(a,s) \\ \supset \neg Wtr\_entering\_hdr(do(a,s)).$$

Next, we rewrite our theory $\Sigma \cup T_{frame}$ as a new theory, Σ^*. 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 $F_i \in \mathcal{L}$. Then, for every axiom in $\Sigma \cup T_{frame}$, 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.

$$Poss^*(a) \wedge a = Disrupt\_siphon \supset \neg Wtr\_entering\_hdr^{**}(a),$$

$$Ok^*(Power) \wedge Ok^*(Pump) \wedge On^*(Pump) \supset Wtr\_entering\_hdr^*,$$

$$Mnl\_filling^* \supset Wtr\_entering\_hdr^*,$$

$$Ok^{**}(Power,a) \wedge Ok^{**}(Pump,a) \wedge On^{**}(Pump,a) \supset Wtr\_entering\_hdr^{**}(a),$$

$$Mnl\_filling^{**}(a) \supset Wtr\_entering\_hdr^{**}(a),$$

$$Poss^*(a) \wedge Wtr\_entering\_hdr^* \wedge \neg ab^{*-}_{Wtr\_entering\_hdr}(a) \supset Wtr\_entering\_hdr^{**}(a),$$

$$Poss^*(a) \wedge \neg Wtr\_entering\_hdr^* \wedge \neg ab^{*+}_{Wtr\_entering\_hdr}(a) \\ \supset \neg Wtr\_entering\_hdr^{**}(a).$$

Using analogous notation to that employed in Σ, we refer to $a = Disrupt\_siphon$ as $\gamma^{*-}_{Wtr\_entering\_hdr}(a)$ and

$$(Ok^*(Power) \wedge Ok^*(Pump) \wedge On^*(Pump)) \vee Mnl\_filling^*,$$

and

$$(Ok^{**}(Power, a) \wedge Ok^{**}(Pump, a) \wedge On^{**}(Pump, a)) \vee Mnl\_filling^{**}(a)$$

as $\upsilon^{*+}_{Wtr\_entering\_hdr}$ and $\upsilon^{**+}_{Wtr\_entering\_hdr}(a)$ respectively. There is no $\gamma^{*+}_{Wtr\_entering\_hdr}(a)$, no $\upsilon^{*-}_{Wtr\_entering\_hdr}$, and no $\upsilon^{**-}_{Wtr\_entering\_hdr}(a)$.

Generalizing this example, Σ^* is produced from $\Sigma \cup T_{frame}$ as follows.

Definition 7 (Σ^*). Suppose Σ is the theory defined in Definition 6. Define Σ^* to be the theory

$$\Sigma_{found} \cup T_{UNA} \cup T^*_{ef} \cup T^*_{ram} \cup T^*_{frame}, \tag{77}$$

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

$$Poss(a, s) \wedge F_i(\vec{x}, s) \wedge \neg ab^-_{F_i}(\vec{x}, a, s) \supset F_i(\vec{x}, do(a, s)), \tag{78}$$

$$Poss(a, s) \wedge \neg F_i(\vec{x}, s) \wedge \neg ab^+_{F_i}(\vec{x}, a, s) \supset \neg F_i(\vec{x}, do(a, s)), \tag{79}$$

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 $Poss^*(a)$.

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

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., [12]), and the definition of our prioritized model preference.

Lemma 1. *Suppose Σ is the theory defined in Definition* 6 *and Σ^* is the theory defined in Definition* 7. *Then $\mathcal{M}$ is a minimal model of Σ with respect to the prioritized model preference of Definition* 4 *iff $\mathcal{M}'$ is a model of*

$$\forall s.CIRC^+(\Sigma^*; Ab_1 > \cdots > Ab_n; F^{**}_1, \ldots, F^{**}_n),$$

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 $CIRC(\Sigma^*; Ab_1 > \cdots > Ab_n; F^{**}_1, \ldots, F^{**}_n)$ 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., [12,13]). 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 constraints, are equivalent to successor state axioms.

The consistency assumption upon which we predicate our theorem is the transformation of Consistency Assumption 1. 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-occur. Since we have added generic frame axioms to Σ^*, we must reflect this addition in the consistency assumption.

Assumption 2 (*Consistency Assumption*). For each fluent $F_i \in \mathcal{L}_i$, assume that:

$$\begin{aligned} T_{UNA} \cup T^*_{nec} \models Poss^*(a) \supset & \\ \neg \big[& (\gamma^{*+}_{F_i}(a) \vee \upsilon^{**+}_{F_i}(a) \vee (F^*_i \wedge \neg ab^{*-}_{F_i}(a))) \\ & \wedge (\gamma^{*-}_{F_i}(a) \vee \upsilon^{**-}_{F_i}(a) \vee (\neg F^*_i \wedge \neg ab^{*+}_{F_i}(a)))\big]. \end{aligned} \tag{80}$$

Theorem 4. *Suppose Σ^* is as defined in Definition* 7 *and Consistency Assumption* 2 *holds. Then*

$$\begin{aligned} & CIRC(\Sigma^*; Ab_1 > \cdots > Ab_n; F^{**}_1(a), \ldots, F^{**}_n(a)) \\ & \quad \equiv \Sigma_{found} \cup T_{UNA} \cup T^*_{ef} \cup T^*_{ram} \cup T^*_{EC} \cup T^*_{Ab\text{-}equivs} \\ & \quad \equiv \Sigma_{found} \cup T_{UNA} \cup T^*_{SS} \cup T^*_{Ab\text{-}equivs}, \end{aligned}$$

where

- $T^*_{EC} = \bigcup_{i=1}^n T^*_{EC_i}$ *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 $F_i \in \mathcal{L}_i$.*

$$\begin{aligned} & Poss^*(a) \wedge F^*_i \wedge \neg F^{**}_i(a) \supset \gamma^{*-}_{F_i}(a) \vee \upsilon^{**-}_{F_i}(a), \\ & Poss^*(a) \wedge \neg F^*_i \wedge F^{**}_i(a) \supset \gamma^{*+}_{F_i}(a) \vee \upsilon^{**+}_{F_i}(a). \end{aligned}$$

- $T^*_{SS} = \bigcup_{i=1}^n T^*_{SS_i}$ *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 $F_i \in \mathcal{L}_i$.*

$$\begin{aligned} Poss^*(a) \supset \big[F^{**}_i(a) \equiv {} & \gamma^{*+}_{F_i}(a) \vee \mathcal{R}^{i-1}[\upsilon^{**+}_{F_i}(a)] \\ & \vee (F^*_i \wedge \neg(\gamma^{*-}_{F_i}(a) \vee \mathcal{R}^{i-1}[\upsilon^{**-}_{F_i}(a)]))\big], \end{aligned}$$

  where $\mathcal{R}^{i-1}$ is the regression operator under the successor state axioms, $T^*_{SS_1} \cup \cdots \cup T^*_{SS_{i-1}}$.
- $T^*_{Ab\text{-}equivs} = \bigcup_{i=1}^n T^*_{Ab\text{-}equivs_i}$ *is the set of circumscribed definition of $ab^{*+}_{F_i}(a)$ and $ab^{*-}_{F_i}(a)$. $T^*_{Ab\text{-}equivs_i}$ is a set of formulae of the following form, one formula for every $F_i \in \mathcal{L}_i$.*

$$\begin{aligned} & ab^{*+}_{F_i}(a) \equiv Poss^*(a) \wedge \neg F^*_i \wedge (\gamma^{*+}_{F_i}(a) \vee \upsilon^{**+}_{F_i}(a)), \\ & ab^{*-}_{F_i}(a) \equiv Poss^*(a) \wedge F^*_i \wedge (\gamma^{*-}_{F_i}(a) \vee \upsilon^{**-}_{F_i}(a)). \end{aligned}$$

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 4 back to the successor state axioms of our original language.

Proposition 3. *Suppose Σ is the theory defined in Definition* 6 *and Σ^* is the theory defined in Definition* 7 *and assume that Consistency Assumption* 2 *holds. Then*

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

where

- *$CIRC^+$ is as defined in Lemma* 1.
- *T_{SS} is the set of successor state axioms for fluents $F_i \in \mathcal{L}_i$ of Σ. They are of the form of* (66).
- *$T_{Ab\text{-}equivs}$ is $T^*_{Ab\text{-}equivs}$ of Theorem* 4*, 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, such that $S_0 \leqslant s$, using notation from Σ_{found}), then 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 5. *Suppose Σ is the theory defined in Definition* 6 *and assume that Consistency Assumption* 1 *holds. Further, suppose $\mathcal{M}$ is a model of Σ.*

Then for variable assignment σ_s to s such that, $S_0 \leqslant s$, $\mathcal{M}$ is a minimal model of Σ with respect to the prioritized model preference of Definition 4 *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 [15] 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 6. *Suppose Σ is the theory defined in Definition* 6 *and that $T_{ram} = \{\}$. Further, assume the following consistency condition holds for every $F_i \in \mathcal{L}_i$,*

$$T_{UNA} \models \neg(\gamma_{F_i}^+(a,s) \wedge \gamma_{F_i}^-(a,s)). \tag{81}$$

Suppose $\mathcal{M}$ is a model of Σ. Then for variable assignment σ_s to s such that, $S_0 \leqslant s$, $\mathcal{M}$ is a minimal model of Σ with respect to the prioritized model preference of Definition 4 *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.*

$$Poss(a,s) \supset [F_i(do(a,s)) \equiv \gamma_{F_i}^+(a,s) \vee (F_i(s) \wedge \neg\gamma_{F_i}^-(a,s))]. \tag{82}$$

Condition (81) 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 (82) are the successor state axioms Reiter identified as his solution to the frame problem [33].

This concludes the independent semantic justification for our closed-form solution. Proofs of theorems can be found at the publisher's web site, as noted at the end of this paper. Detailed proofs and further results can be found in [25].

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. The intuition behind our solution to the frame and ramification problems—the notion of interpreting our ramification constraints as directional or *definitional* in nature was originally inspired by research on the semantics of normal logic programs and deductive databases (e.g., [31]), and by the preliminary work of Pinto [29]. Our compilation approach to solving the ramification problem, and more specifically our appeal to a completeness assumption to generate explanation closure axioms was inspired by Reiter's [33], Schubert's [39] and Pednault's [28] approaches to solving the frame problem.

Our closed-form solution relies on a notion of causal influence, which can either be expressed by a causal interpretation of the implication connective in the axiomatization, or by provision of a separate causal influence graph, describing the causal relationships between fluents. Lin [14] and McCain and Turner [18] were among the first to describe the relationship between fluents in a ramification constraint as *causal*. In contrast to our approach, Lin introduced an explicit *Caused* predicate into his axiomatization, while McCain and Turner express ramification constraints as *causal laws* with a special $\Rightarrow$ connective.

The basic minimization policy we employed in our semantic justification is derivative of Lin and Shoham [16] and Lin and Reiter [15], with the important addition of making the minimization prioritized with respect to the causal influence ordering of fluents. As a result, we were able to draw a correspondence between our nonmonotonic specification and circumscription, to show that the corresponding prioritized circumscription was first-order definable, and that when the circumscription was computed under a consistency assumption, it produced our closed-form solution. Several other researchers have proposed nonmonotonic solutions to the frame and ramification problems that are based on circumscription. For example, Lin [14], Kartha and Lifschitz [9] and Giunchiglia [7] provide circumscription-based solutions in terms of the situation calculus, while Gustafsson and Doherty [8] present a solution in the action language PMON. Each of these characterizations differs in how it axiomatizes the domain, and hence which predicates it minimizes in its circumscription. A distinguishing feature of our characterization is that unlike [8,9,14], our circumscription is prioritized and further, it is applied to the whole theory, not just to some part of the theory. Circumscribing only part of a theory is a technique referred to by Sandewall as *filtered preferential entailment* [36]. Shanahan [40] provides an extended discussion of the merits of and objections to this technique, which he refers to as *forced separation*. That said, the spirit of many of these solutions is similar, whether it is a predicate called *Caused* [14], *Occluded* [8] or *Ab* (e.g., [9]) that is being minimized. Indeed the author suspects that for the syntactically restricted case studied here, all our different proposed solutions may produce equivalent intended interpretations, just as many of the independent solutions to the frame problem proved to be identical under certain conditions [1]. A final distinguishing feature of our work is that we prove that our nonmonotonic solution leads to an articulated closed-form solution, under a consistency assumption. Causality can be expressed as a separate structure and need not be included in the original axiomatization of state constraints. Hence, we claim it is more amenable to addressing the motivating problem we introduced in Section 1.

In addition to circumscription-based solutions to the frame and ramification problem, Ternovskaia [41] and Denecker et al. [3] independently proposed to characterize effect axioms and ramification constraints as inductive definitions of fluents. They used these inductive definitions as an alternative to a circumscription-based definition of solutions to the frame and ramification problems. Ternovskaia [41] went on to show that both Reiter's closed-form solution to the frame problem and our closed-form solution to the frame and ramification problems could be semantically justified by appealing to this notion of inductive definitions, providing an alternative to our justification based on prioritized circumscription.

Finally, several researchers have addressed the ramification problem by exploiting some form of postprocessing or propagation of indirect effects, that follows computation of direct effects. Thielscher [42] suggests computing ramifications by an additional post-processing step over ramification constraints defined as causal laws. This is somewhat akin to Pinto's notion of computing ramifications via prime implicate generation [30]. Both share intuitions with Sandewall's transition cascade semantics [37].

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 a 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 included compilation of 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. Further, the closed-form solution can be viewed as an executable specification, and is easily realized in Prolog.

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 over the causal influence ordering of fluents, we observed that this circumscription was first-order definable and showed that computing the prioritized circumscription produced 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. Not only did these results provide semantic justification for our closed-form solution to the frame and ramification problem, and as a side effect, Reiter's closed-form solution to the frame problem, but they also provided a nonmonotonic characterization of the solution, which relates this work to other research on this subject.

In closing, the research presented in this paper was originally motivated by the specific problem of integrating a representation of action with the representation of the behaviour of physical systems, for the purpose of diagnostic problem solving. McIlraith [24–26] provides an account of this specific problem and the use of our solution in various diagnostic problem solving tasks.

Acknowledgements

A subset of the material presented in this paper appeared in "Representing actions and state constraints in model-based diagnosis", in: Proceedings of Fourteenth National Conference on Artificial Intelligence (AAAI-97), AAAI Press, Menlo Park, CA, 1997.

Many of the basic ideas presented in this paper first appeared in my doctoral dissertation [25] and were presented in [23] and [24]. I would like to thank Ray Reiter, Fangzhen Lin, Yves Lespérance, Hector Levesque and Allan Jepson for insightful comments on the results contained in this paper. I would particularly like to acknowledge Fangzhen Lin for providing helpful comments on the circumscription results, and for helping mathematically articulate my intuitions regarding the existence of a lemma that was essential to the proof of Theorem 4. Thanks to Danny Bobrow for his support and for interesting and helpful discussions relating to this work. Finally, thanks to the anonymous referees for their thorough reviews.

Appendix A. Proofs

Proofs for the results presented in this paper are available electronically from the publisher's web site at http://www.elsevier.nl/locate/artint.

References

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

[2] J. de Kleer, J.S. Brown, A qualitative physics based on confluences, Artificial Intelligence 24 (1984) 7–83.

[3] M. Denecker, D. Theseider Dupré, K. Van Belleghem, An inductive definition approach to ramifications, Electron. Trans. Artificial Intelligence 3 (7) (1998).

[4] C. Elkan, Reasoning about action in first-order logic, in: Proceedings of the Ninth Biennial Conference of the Canadian Society for Computational Studies of Intelligence (CSCSI-92), Vancouver, BC, 1992, pp. 221–227.
[5] J. Finger, Exploiting constraints in design synthesis, Ph.D. Thesis, Department of Computer Science, Stanford University, 1986.
[6] M.L. Ginsberg, D.E. Smith, Reasoning about actions I: A possible worlds approach, Artificial Intelligence 35 (2) (1988) 165–195.
[7] E. Giunchiglia, Determining ramifications in the situation calculus, in: Proceedings of the Fifth International Conference on Principles of Knowledge Representation and Reasoning (KR-96), Cambridge, MA, 1996, pp. 76–86.
[8] J. Gustafsson, P. Doherty, Embracing occlusion in specifying the indirect effects of actions, in: Proceedings of the Fifth International Conference on Principles of Knowledge Representation and Reasoning (KR-96), Cambridge, MA, 1996, pp. 87–98.
[9] G.N. Kartha, V. Lifschitz, A simple formalization of actions using circumscription, in: Proc. IJCAI-95, Montreal, Quebec, 1995, pp. 1970–1975.
[10] H. Levesque, R. Reiter, Y. Lespérance, F. Lin, R. Scherl, GOLOG: A logic programming language for dynamic domains, J. Logic Programming 31 (1997) 59–84.
[11] G. Levi, Advances in Logic Programming Theory, Oxford Science Publications, Oxford, 1994.
[12] V. Lifschitz, Computing circumscription, in: Proc. IJCAI-85, Los Angeles, CA, 1985, pp. 121–127.
[13] V. Lifschitz, Circumscription, in: D.M. Gabbay, C.J. Hogger, J.A. Robinson (Eds.), The Handbook of Logic in AI and Logic Programming, Vol. 3, Oxford University Press, Oxford, 1994, pp. 298–352.
[14] F. Lin, Embracing causality in specifying the indirect effects of actions, in: Proc. IJCAI-95, Montreal, Quebec, 1995, pp. 1985–1991.
[15] F. Lin, R. Reiter, State constraints revisited, J. Logic Comput. (Special Issue on Action and Processes) 4 (5) (1994) 655–678.
[16] F. Lin, Y. Shoham, Provably correct theories of action, J. ACM 42 (2) (1995) 293–320.
[17] J.W. Lloyd, Foundations of Logic Programming, 2nd edn., Springer, Berlin, 1987.
[18] N. McCain, H. Turner, A causal theory of ramifications and qualifications, in: Proc. IJCAI-95, Montreal, Quebec, 1995, pp. 1978–1984.
[19] J. McCarthy, Programs with common sense, in: M. Minsky (Ed.), Semantic Information Processing, MIT Press, Cambridge, MA, 1968, Chapter 7, pp. 403–418.
[20] J. McCarthy, Epistemological problems of artificial intelligence, in: Proc. IJCAI-77, Cambridge, MA, 1977, pp. 1038–1044.
[21] J. McCarthy, Mathematical logic in artificial intelligence, in: Daedalus, Winter, 1988, pp. 297–311.
[22] J. McCarthy, P. Hayes, Some philosophical problems from the standpoint of artificial intelligence, in: B. Meltzer, D. Michie (Eds.), Machine Intelligence 4, Edinburgh University Press, Edinburgh, 1969, pp. 463–502.
[23] S. McIlraith, A closed-form solution to the ramification problem (sometimes), in: Proceedings of the Workshop on Nonmonotonic Reasoning, Action and Change, IJCAI-97, Nagoya, Japan, 1997, pp. 103–126.
[24] S. McIlraith, Representing actions and state constraints in model-based diagnosis, in: Proc. AAAI-97, Providence, RI, 1997, pp. 43–49.
[25] S. McIlraith, Towards a formal account of diagnostic problem solving, Ph.D. Thesis, Department of Computer Science, University of Toronto, Toronto, Ontario, 1997.
[26] S. McIlraith, Explanatory diagnosis: Conjecturing actions to explain observations, in: Proceedings of the Sixth International Conference on Principles of Knowledge Representation and Reasoning (KR-98), Trento, Italy, 1998, pp. 167–177.
[27] J. Mylopolous, B. Kramer et al., Developing an expert system technology for industrial process control: An experience report, in: Proceedings of the 11th Biennial Conference of the Canadian Society for Computational Studies of Intelligence (CSCSI-96), Toronto, Ont., 1996, pp. 172–186.
[28] E. Pednault, Synthesizing plans that contain actions with context-dependent effects, Comput. Intelligence 4 (1988) 356–372.
[29] J. Pinto, Temporal reasoning in the situation calculus, Ph.D. Thesis, Department of Computer Science, University of Toronto, Toronto, Ontario, 1994. Also published as Technical Report, Department of Computer Science, University of Toronto (KRR-TR-94-1), February 1994.

[30] J. Pinto, Compiling ramification constraints into effect axioms, Comput. Intelligence 15 (3) (1999) 280–307.
[31] T.C. Przymusinski, On the declarative and procedural semantics of logic programs, J. Automat. Reason. 5 (1989) 167–205.
[32] R. Reiter, Knowledge in action: logical foundations for describing and implementing dynamical systems, In preparation. Draft available at http://www.cs.toronto.edu/~cogrobo/.
[33] R. Reiter, The frame problem in the situation calculus: A simple solution (sometimes) and a completeness result for goal regression, in: V. Lifschitz (Ed.), Artificial Intelligence and Mathematical Theory of Computation: Papers in Honor of J. McCarthy, Academic Press, San Diego, CA, 1991, pp. 359–380.
[34] R. Reiter, The frame problem in the situation calculus: A soundness and completeness result, with an application to database updates, in: Proceedings First International Conference on AI Planning Systems, College Park, MA, 1992.
[35] R. Reiter, Natural actions, concurrency and continuous time in the situation calculus, in: Proceedings of the Fifth International Conference on Principles of Knowledge Representation and Reasoning (KR-96), Cambridge, MA, 1996, pp. 2–13.
[36] E. Sandewall, Filter preferential entailment for the logic of action in almost continuous worlds, in: Proc. IJCAI-89, Detroit, MI, 1989, pp. 894–899.
[37] E. Sandewall, Assessment of ramification methods that use static domain constraints, in: Proceedings of the Fifth International Conference on Principles of Knowledge Representation and Reasoning (KR-96), Cambridge, MA, 1996, pp. 99–110.
[38] R. Scherl, H. Levesque, The frame problem and knowledge producing actions, in: Proc. AAAI-93, Washington, DC, 1993, pp. 689–695.
[39] 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, G.N. Carlson (Eds.), Knowledge Representation and Defeasible Reasoning, Kluwer Academic Press, Dordrecht, 1990, pp. 23–67.
[40] M. Shanahan, Solving the Frame Problem, MIT Press, Cambridge, MA, 1997.
[41] E. Ternovskaia, Inductive definability and the situation calculus, in: B. Freitag, H. Decker, M. Kifer, A. Voronkov (Eds.), Transactions and Change in Logic Databases, Lecture Notes in Computer Science, Vol. 1472, Springer, Berlin, 1998.
[42] M. Thielscher, Qualified ramifications, in: Proc. AAAI-97, Providence, RI, 1997, pp. 466–471.
[43] R. Waldinger, Achieving several goals simultaneously, in: E. Elcock, D. Michie (Eds.), Machine Intelligence 8, Ellis Horwood, Edinburgh, 1977, pp. 94–136.