A General First-Order Solution to the Ramification Problem

Hannes Strass
Department of Computer Science,
Dresden University of Technology
hannes.strass@inf.tu-dresden.de

Michael Thielscher
School of Computer Science and Engineering,
The University of New South Wales
mit@cse.unsw.edu.au

Abstract

We present a combined solution to the frame and ramification problems that is independent of the underlying time structure. Indirect effects are expressed through ramification rules that are compiled into first-order effect axioms. To cope with the notorious problem of self-justifying cycles, we use techniques known from translations of normal logic programs to logical theories: cyclic fluent dependencies in the ramification rules are identified, and for each such loop, a loop formula is built into the effect axiom to guarantee proper treatment of circular causal dependencies.

Motivation

The ramification problem of reasoning about actions has received considerable attention since its discovery in (Ginsberg and Smith 1987). Yet, current research into the ramification problem is not in the best of states: there exists a great quantity of approaches, most of which are formulated in fundamentally different formalisms. They often rely on non-standard logics and -semantics and employ isolated example scenarios to illustrate each other's weaknesses.

In this paper, we provide a solution to the ramification problem that attempts to remedy this situation. Our approach integrates findings of different approaches to ramification from the last ten to fifteen years. For the first time, we present a solution that: (1) is independent of a particular calculus, (2) is formulated in classical first-order logic, and (3) treats cycles properly. This is achieved as follows: We compile *causal relationships* (Thielscher 1997) into effect axioms that then provide a combined solution to the frame and ramification problems. These resulting effect axioms are expressed in a recently proposed general action formalism, the unifying action calculus (UAC) (Thielscher 2010). The UAC is based on first-order logic and has been shown to encompass a wide range of existing action formalisms. Most notably, it abstracts from the particular time structures underlying specific calculi—it therefore provides an ideal basis for a general treatment of ramifications. Furthermore, staying within first-order logic, we can resort to standard reasoning mechanisms as opposed to non-standard semantics.

Early treatments of ramifications (Ginsberg and Smith 1987; Winslett 1988; Lin and Reiter 1994) all assumed the behavior of the world to be specified by state constraints—static laws that must hold in all states of the world—and relied on the principle of minimal change. Then (Lin 1995), (McCain and Turner 1995), and (Thielscher 1995) independently made the somewhat fundamental observation that mere state constraints are insufficient for a proper treatment of ramifications; the authors concluded that an explicit notion of causation is needed for this purpose. (Thielscher 1997) then showed how information on causal influence among fluents can be used to create a set of *causal relationships* from a set of domain constraints. A state-transition semantics is given there that applies causal relationships to intermediate, inconsistent states and ensures a consistent successor state is reached (if such a state exists). Although our approach is based on (a more general version of) causal relationships, we are not concerned with their creation from domain constraints. We assume that the user, when axiomatizing an action domain, additionally specifies a set of causal relationships that describe the propagation of effects in the domain.

Should these causal relationships now exhibit implicit cyclic fluent dependencies, our approach will deal with that appropriately, both for positive cyclic fluent dependencies (i.e. self-supporting effects) and negative cyclic fluent dependencies (i.e. self-canceling effects). The latter case manifests itself in the resulting effect axioms being inconsistent, which is an indication for a modeling error. (Compilation of causal relationships into effect axioms therefore presents an easy way of checking consistency of a given set of ramification rules.) The former case is accounted for in a preprocessing phase: using techniques from logic programming, we identify positive loops in the set of ramification rules and build their corresponding loop formulas (Lin and Zhao 2004) into the effect axiom.

The rest of the paper is organized as follows. In the next section, we give the necessary background on the unifying action calculus. The section thereafter introduces the effect axiom used here; the subsequent section is devoted to incorporating ramifications into this effect axiom and giving a formal assessment of this incorporation. Related and future work are discussed in the concluding section.

Background: The Unifying Action Calculus

The unifying action calculus was proposed in (Thielscher 2010) to allow for a treatment of problems in reasoning about actions that abstracts from a particular calculus. It is based on a finite, sorted logic language with equality (the *domain signature*) which includes the sorts FLUENT, ACTION, and TIME along with the predicates $<$: TIME $\times$ TIME, that denotes a (possibly partial) ordering on time points; $Holds$: FLUENT $\times$ TIME, that is used to state that a fluent is true at a given time point; and $Poss$: ACTION $\times$ TIME $\times$ TIME, that expresses the applicability of an action for given starting and ending time points.

The following definition introduces the fundamental types of formulas of the UAC: they allow to express properties of action domains at given time points and executability conditions as well as effects of actions.

Definition 1. Let $\vec{t}$ be a sequence of variables of sort TIME.

- A *state formula* $\Phi[\vec{t}]$ *in* $\vec{t}$ is a first-order formula with $\vec{t}$ among its free variables and where
  - for each occurrence of $Holds(f,t)$ in $\Phi[\vec{t}]$ we have $t \in \vec{t}$ and
  - predicate $Poss$ does not occur.

Let s, t be variables of sort TIME and A be a function from any sort into sort ACTION.

- A *precondition axiom for* $A(\vec{x})$ is of the form

$$Poss(A(\vec{x}), s, t) \equiv \pi_A[s] \qquad (1)$$

  where $\pi_A[s]$ is a state formula in s with free variables among $s, t, \vec{x}$.
- An *effect axiom for* $A(\vec{x})$ is of the form

$$\begin{aligned}&Poss(A(\vec{x}), s, t) \supset\\ &\qquad (\exists \vec{y})((\forall f)(\Upsilon^{+}[s,t] \supset Holds(f,t)) \wedge\\ &\qquad\qquad (\forall f)(\Upsilon^{-}[s,t] \supset \neg Holds(f,t))) \quad (2)\end{aligned}$$

  in which both $\Upsilon^{+}[s,t]$ and $\Upsilon^{-}[s,t]$ are state formulas in s, t with free variables among $f, s, t, \vec{x}, \vec{y}$.[1]

We next formalize how action domains are axiomatized in the unifying action calculus.

Definition 2. A *(UAC) domain axiomatization* consists of a finite set of foundational axioms Ω (by which the UAC is instantiated by a concrete time structure, e.g. the branching situations along with the usual ordering from Situation Calculus), a set Π of precondition axioms (1), and a set Υ of effect axioms (2); the latter two for all functions into sort ACTION.

A domain axiomatization is *progressing* if

- $\Omega \models (\exists s : \text{TIME})(\forall t : \text{TIME}) s \leq t$ and
- $\Omega \cup \Pi \models Poss(a, s, t) \supset s < t$.

In this paper, we are only concerned with progressing domain axiomatizations. To be able to reference the unique initial time point, we use the macro $Init(t) \stackrel{\text{def}}{=} \neg(\exists s) s < t$. We will then equip our domain axiomatizations with a set Σ_0 of *initial state axioms* describing the state of the world at the initial time point. These initial state axioms are state formulas of the form $Init(t) \supset \Phi[t]$ where $\Phi[t]$ is a state formula in t.

For presentation purposes, we will make use of the concept of *fluent formulas*: these are standard first-order formulas but where terms of sort FLUENT play the role of atomic formulas. We will denote by $\Phi[s]$ the state formula that is obtained by replacing all fluent literals $[\neg]f$ in a fluent formula Φ by $[\neg]Holds(f, s)$.

The Effect Axiom

This section presents the general, first-order effect axiom that will be employed and elaborated throughout the paper. This axiom is inspired by a specific example scenario from (Thielscher 2010), which in turn, as mentioned there, was inspired by the work of (Giunchiglia et al. 2004). It formalizes the idea of truth by causation: everything that is true must be caused, and vice versa. In the most simple form of the effect axiom, we allow two causes to determine a fluent's truth value: persistence and direct effects. Before introducing the axiom itself, we define two pairs of macros that formalize the individual causes. The first pair expresses persistence.

$$FrameT(f,s,t) \stackrel{\text{def}}{=} Holds(f,s) \wedge Holds(f,t) \qquad (3)$$

$$FrameF(f,s,t) \stackrel{\text{def}}{=} \neg Holds(f,s) \wedge \neg Holds(f,t) \qquad (4)$$

Assume the direct effects of an action are given as a set of condition-effect pairs Φ/ψ (with Φ a fluent formula and ψ a fluent literal) meaning that the action brings about ψ if Φ holds at the starting time point. These expressions can easily be translated into "causes" for the purpose of designing an effect axiom for that action.

Definition 3. Let A be a function into sort ACTION and Γ_A be a set of expressions Φ/ψ (where Φ is a fluent formula and ψ a fluent literal) with free variables among $\vec{x}, \vec{y}$ that denote the direct conditional (local and non-local) effects of $A(\vec{x})$.

$$DirT(f, A(\vec{x}), s, t) \stackrel{\text{def}}{=} \bigvee_{\Phi/F(\vec{x},\vec{y}) \in \Gamma_A} (\exists \vec{y})(f = F(\vec{x},\vec{y}) \wedge \Phi[s]) \qquad (5)$$

$$DirF(f, A(\vec{x}), s, t) \stackrel{\text{def}}{=} \bigvee_{\Phi/\neg F(\vec{x},\vec{y}) \in \Gamma_A} (\exists \vec{y})(f = F(\vec{x},\vec{y}) \wedge \Phi[s]) \qquad (6)$$

Definition 4. Let A be a function into sort ACTION. An *effect axiom with conditional effects and the frame assumption* is of the form[2]

$$\begin{aligned}&Poss(A(\vec{x}), s, t) \supset\\ &\qquad (\forall f)(Holds(f,t) \equiv CausedT(f, A(\vec{x}), s, t)) \wedge\\ &\qquad (\forall f)(\neg Holds(f,t) \equiv CausedF(f, A(\vec{x}), s, t)) \quad (7)\end{aligned}$$

[1]The original definition of UAC effect axioms is much more general; in this paper, we restrict their syntax for the sake of clarity. Variables $\vec{x}$ and $\vec{y}$ can be of any sort.

[2]The attentive reader will have noticed that the syntax of axiom (7) does not quite correspond to Definition 1. Simple syntactical manipulations can however be conducted to transform the effect axiom into a form that matches the structure of (2).

$$\mathit{CausedT}(f, A(\vec{x}), s, t) \stackrel{\text{def}}{=} \mathit{FrameT}(f, s, t) \vee \mathit{DirT}(f, A(\vec{x}), s, t) \quad (8)$$

$$\mathit{CausedF}(f, A(\vec{x}), s, t) \stackrel{\text{def}}{=} \mathit{FrameF}(f, s, t) \vee \mathit{DirF}(f, A(\vec{x}), s, t) \quad (9)$$

The predicates $\mathit{CausedT}$, $\mathit{CausedF}$ will be refined in a subsequent section. When speaking about effect axiom (7), we will understand it retrofitted with their "latest version". For the sake of brevity, we will from now on only define the positive versions of predicate macros (for example $\mathit{CausedT}$) since their negative versions (in the example: $\mathit{CausedF}$) are absolutely symmetric.

The design principle underlying the axiomatization technique of this effect axiom is that of causation: a fluent holds at a time point that is the end point of an action if and only if there is a cause for that; similarly, a fluent does not hold if and only if there is a cause for that, too.

From now on, when speaking about domain axiomatizations, we will understand all effect axioms to be of the form (7) and have the domain axiomatizations include uniqueness-of-names axioms for all finitely many function symbols into sorts FLUENT and ACTION.

Incorporating Indirect Effects

We now present the basic underlying idea of our solution to the ramification problem. It relies on an extension of our effect axiom that preserves the solution of the frame problem and additionally accounts for ramifications by incorporating a set of user-defined *causal relationships*. These relationships specify certain conditions under which a change of truth value of one fluent causes a change of truth value of another fluent.

Definition 5. A *causal relationship* (or *causal rule*) is of the form

$$\Phi : \chi \Rrightarrow \psi \quad (10)$$

Φ, the *context*, is a fluent formula, while χ, the *trigger*, and ψ, the *effect*, are fluent literals. A causal relationship is called *open* if it contains free variables, otherwise it is *closed*.

The notion of a causal relationship was introduced in their closed form in (Thielscher 1997), albeit with a different syntax and for the purpose of using them in a specific action formalism (Fluent Calculus). We now show how to integrate these causal relationships into the general effect axiom. The idea is to express them as implications and take care that inferences in the contrapositive (i.e. non-causal) direction are not possible. The macros $\mathit{IndT}(f, s, t), \mathit{IndF}(f, s, t)$ express that fluent f is an indirect (positive or negative, respectively) effect of an action occurring from s to t. For a causal relationship $\Phi : \chi \Rrightarrow \psi$, the indirect effect ψ is established whenever the rule has been *triggered*, that is, whenever the context Φ holds at the starting time point s and χ, the trigger, has changed from untrue to true from s to t.

Definition 6. Let $\Phi : \chi \Rrightarrow \psi$ be a causal relationship and s, t : TIME be variables.

$$\mathit{Triggered}_{\Phi:\chi\Rrightarrow\psi}(s, t) \stackrel{\text{def}}{=} \Phi[s] \wedge \neg\chi[s] \wedge \chi[t] \quad (11)$$

Let $\mathcal{R}$ be a set of causal relationships and f be a variable of sort FLUENT.

$$\mathit{IndT}(f, s, t) \stackrel{\text{def}}{=} \bigvee_{\Phi(\vec{y}):\chi(\vec{y})\Rrightarrow F(\vec{y})\in\mathcal{R}} (\exists\vec{y}) \Big(f = F(\vec{y}) \wedge \mathit{Triggered}_{\Phi(\vec{y}):\chi(\vec{y})\Rrightarrow F(\vec{y})}(s, t) \Big) \quad (12)$$

(The corresponding definition of $\mathit{IndF}(f, s, t)$ takes all causal rules with negative effect and is symmetric.)

According to these macros, a fluent f is an indirect effect from s to t if there is a corresponding causal relationship with effect f that triggered from s to t. The macros are straightforwardly integrated into the effect axiom as follows.

Definition 7. Let A be a function into sort ACTION. An *effect axiom with conditional effects, the frame assumption, and ramifications* is of the form (7), where

$$\mathit{CausedT}(f, A(\vec{x}), s, t) \stackrel{\text{def}}{=} \mathit{FrameT}(f, s, t) \vee \mathit{DirT}(f, A(\vec{x}), s, t) \vee \mathit{IndT}(f, s, t) \quad (13)$$

We use a slight modification of a simple, well-known ramification domain (Baker 1991) to illustrate how our effect axioms work.

Example 1 (Walking Turkeys)**.** Consider the fluents $\mathsf{Alive}(y)$ and $\mathsf{Walking}(y)$ along with the action $\mathsf{Shoot}(x)$ of shooting x that has the effects $\Gamma_{\mathsf{Shoot}(x)} = \{\top/\neg\mathsf{Alive}(x)\}$ (meaning the object shot at is not alive any more) and the precondition axiom $\mathit{Poss}(\mathsf{Shoot}(x), s, t) \equiv s < t$. The causal relationship below says that for any object, not to be alive is sufficient cause for it not to be walking:

$$\top : \neg\mathsf{Alive}(y) \Rrightarrow \neg\mathsf{Walking}(y)$$

Applying Definition 6 results in

$$\mathit{IndF}(f, s, t) = (\exists y)(f = \mathsf{Walking}(y) \wedge \mathit{Holds}(\mathsf{Alive}(y), s) \wedge \neg\mathit{Holds}(\mathsf{Alive}(y), t))$$

Since $\mathsf{Shoot}(x)$ has no positive effect and there is also no causal relationship with positive effect, the effect axiom thus constructed for our action is now given by

$$\begin{aligned} \mathit{CausedT}(f, \mathsf{Shoot}(x), s, t) &= \mathit{FrameT}(f, s, t) \\ \mathit{CausedF}(f, \mathsf{Shoot}(x), s, t) &= \mathit{FrameF}(f, s, t) \vee \\ & f = \mathsf{Alive}(x) \vee \mathit{IndF}(f, s, t) \end{aligned}$$

We add an initial state axiom stating that two turkeys, Fred and Harry, are initially both alive and walking.

$$\begin{aligned} &\mathit{Init}(t) \supset \\ &(\mathit{Holds}(\mathsf{Alive}(\mathsf{Fred}), t) \wedge \mathit{Holds}(\mathsf{Walking}(\mathsf{Fred}), t) \wedge \\ &\mathit{Holds}(\mathsf{Alive}(\mathsf{Harry}), t) \wedge \mathit{Holds}(\mathsf{Walking}(\mathsf{Harry}), t)) \end{aligned}$$

Now taking Σ to be the domain axiomatization comprised of the above-mentioned effect, precondition, and initial state axioms, we can conclude that shooting at Fred yields the desired result that he immediately dies and stops walking all the while Harry stays alive and walking.

$$\Sigma \models (\mathit{Init}(t_0) \wedge \mathit{Poss}(\mathsf{Shoot}(\mathsf{Fred}), t_0, t_1)) \supset$$
$$(\neg \mathit{Holds}(\mathsf{Alive}(\mathsf{Fred}), t_1) \wedge \neg \mathit{Holds}(\mathsf{Walking}(\mathsf{Fred}), t_1) \wedge$$
$$\mathit{Holds}(\mathsf{Alive}(\mathsf{Harry}), t_1) \wedge \mathit{Holds}(\mathsf{Walking}(\mathsf{Harry}), t_1))$$

While the compilation of causal relationships into effect axioms presented so far works well for simple ramification domains and easily copes with instantaneous effect propagation, it still harbors a serious flaw: it cannot handle cyclic fluent dependencies. This is shown using the following example from (Van Belleghem, Denecker, and Theseider-Dupré 1998).

Example 2 (Gear Wheel Domain)**.** There are two interlocked gear wheels, that can be separately turned and stopped. Let the fluents $\mathsf{W}_1, \mathsf{W}_2$ express that the first (resp. second) gear wheel is turning. The actions to initiate/end this are $\mathsf{Turn}_i, \mathsf{Stop}_i$ with effects $\Gamma_{\mathsf{Turn}_i} = \{\top/\mathsf{W}_i\}$, $\Gamma_{\mathsf{Stop}_i} = \{\top/\neg\mathsf{W}_i\}$, $i = 1, 2$; there also exists a trivial action Wait without any direct effects, $\Gamma_{\mathsf{Wait}} = \emptyset$. The causality relating the interlocked gear wheels is described as follows: whenever the first wheel is turned, it causes the second one to turn, and vice versa; whenever the first wheel is stopped, it causes the second one to stop as well, and vice versa. The respective causal relationships are:

$$\top : \mathsf{W}_1 \Rrightarrow \mathsf{W}_2, \top : \neg\mathsf{W}_1 \Rrightarrow \neg\mathsf{W}_2$$
$$\top : \mathsf{W}_2 \Rrightarrow \mathsf{W}_1, \top : \neg\mathsf{W}_2 \Rrightarrow \neg\mathsf{W}_1$$

Let us compile the (positive half of the) effect axiom for the Wait action. Since there are no direct effects, Definitions 6 and 7 yield

$$\mathit{CausedT}(f, \mathsf{Wait}, s, t) \equiv \mathit{FrameT}(f, s, t) \vee \mathit{IndT}(f, s, t)$$

with

$$\mathit{IndT}(f, s, t) =$$
$$(f = \mathsf{W}_2 \wedge \neg\mathit{Holds}(\mathsf{W}_1, s) \wedge \mathit{Holds}(\mathsf{W}_1, t)) \vee$$
$$(f = \mathsf{W}_1 \wedge \neg\mathit{Holds}(\mathsf{W}_2, s) \wedge \mathit{Holds}(\mathsf{W}_2, t))$$

Assume both wheels initially stand still:

$$\mathit{Init}(t) \supset (\neg\mathit{Holds}(\mathsf{W}_1, t) \wedge \neg\mathit{Holds}(\mathsf{W}_2, t))$$

and consider the interpretation I with

$$\mathrm{TIME}^I = \{T_0, T_1\}, T_0 <^I T_1, \mathit{Poss}^I = \{(\mathsf{Wait}, T_0, T_1)\}$$
$$\mathit{Holds}^I = \{(\mathsf{W}_1, T_1), (\mathsf{W}_2, T_1)\}$$

It is a model for effect axiom (7) for Wait where both wheels magically start turning—one being the cause for the other and vice versa. This is undesired as Wait is intended to have no effect at all.

Loops and Loop Formulas

Much as in the case of Clark's completion of normal logic programs, our compilation of causal relationships into effect axioms allows too many models for predicates/fluents that cyclicly depend on each other. We propose a solution to our problem that is in the spirit of loop formulas (Lin and Zhao 2004) for normal logic programs. In order for the approach to stay practical, we however have to restrict the syntax of the causal relationships $\mathcal{R}$. Firstly, we require that for any $r_1 \neq r_2 \in \mathcal{R}$, we have $\mathit{Var}(r_1) \cap \mathit{Var}(r_2) = \emptyset$. This is not an actual constraint but merely for technical reasons. The second restriction stipulates that for each rule $\Phi : \chi \Rrightarrow \psi \in \mathcal{R}$, we have $\mathit{Var}(\chi) \subseteq \mathit{Var}(\psi)$, that is, there may not be local variables in rule triggers. Thirdly, we do not use proper function symbols as arguments of sort FLUENT. The latter two constraints guarantee the existence of a finite, complete set of loops (Chen et al. 2006).

Throughout the following definitions, we will make explicit use of substitutions, unifiers, and most general unifiers (mgus). Their domains and ranges are understood to be built from the domain signature used for specifying the causal relationships. For unification, negation is treated as a unary function symbol.

Definition 8. Let $\mathcal{R}$ be a set of open causal rules. The *influence graph $G_\mathcal{R}$ of $\mathcal{R}$* is the infinite directed graph $G_\mathcal{R} \overset{\text{def}}{=} (V, E)$, where V is the set of all terms $\mu\theta$ with μ a fluent literal mentioned in $\mathcal{R}$ and θ a substitution; a pair $(\mu, \nu) \in E$ if there exists a rule $\Phi : \chi \Rrightarrow \psi \in \mathcal{R}$ and a substitution θ with $\mu = \chi\theta$ and $\nu = \psi\theta$. A finite, nonempty set L of literals constitutes a *loop in $\mathcal{R}$* if for all $\mu, \nu \in L$ there is a directed path from μ to ν in $G_\mathcal{R}$.

A rule $r = \Phi : \chi \Rrightarrow \psi \in \mathcal{R}$ *leads into the loop L* iff

- there exists a $\mu \in L$ and a substitution θ_r^- with $\theta_r^- = mgu(\psi, \mu)$ and
- for all substitutions θ' with $(\exists\theta'')\theta' = \theta_r^-\theta''$ we have $\{\chi\}\theta' \cap L\theta' = \emptyset$.

Then $\mathcal{R}_L^- \overset{\text{def}}{=} \{r\theta_r^- \mid r \in \mathcal{R} \text{ and } r \text{ leads into the loop } L\}$.

Note that, in contrast to the notions from normal logic programs, the nodes of our dependency graphs are literals instead of only positive atoms. Also, the head of the rule (the effect) does not depend on the whole body (trigger plus context) but only on the trigger.

Example 2 (Continued)**.** The causal relationships of the gear wheel domain give rise to two loops:

$$L_1 = \{\mathsf{W}_1, \mathsf{W}_2\} \text{ and } L_2 = \{\neg\mathsf{W}_1, \neg\mathsf{W}_2\}$$

Having defined the loops for a given set of causal relationships, we can now proceed to define the corresponding loop formulas. The idea of loop formulas is to eliminate the models that arise due to "spontaneous" activation of loops for which no *external support* exists. In the case of logic programs, the external support that counts as "legal" cause for loop activation is a program rule leading into the loop. In our case, the direct effects of an action have to be taken into account as potential reasons for loop activation, too. A loop can also be activated by another loop—but then the union of the two is again a loop, so this case is implicitly catered for.

When translating a logic program into a logical theory, the loops are added to the predicate completion of the program. In case of general effect axioms with their standard first-order semantics, loop formulas are "built into" the axioms. This is done as follows: we enforce the frame assumption for all fluent literals that could possibly change their truth value due to spontaneous loop activation. To achieve this for a given literal μ, we specify non-activation of all loops that are relevant for μ (namely those that contain either μ or $\neg\mu$) as a sufficient cause for persistence of μ's truth value.

We use the notation $L(\vec{y})$ to explicitly refer to the free variables $\vec{y}$ mentioned in the loop L. As far as loop activation through an indirect effect (macro *IndActivated*) is concerned, we have to check whether the corresponding ground instance of a rule leading into the loop has fired. This is done in the last line of (14) in the construction of loop formulas:

Definition 9. Let $\mathcal{R}$ be a set of causal relationships, $\mathit{Loops}(\mathcal{R})$ be the set of all loops of $\mathcal{R}$, $L \in \mathit{Loops}(\mathcal{R})$, s, t be variables of sort TIME.

$$\mathit{IndActivated}_L(\vec{y}, s, t) \stackrel{\text{def}}{=} \bigvee_{\Phi(\vec{z}):\chi(\vec{z})\Rightarrow\psi(\vec{z})\in\mathcal{R}_L^-} (\exists \vec{z}) \Big(\mathit{Triggered}_{\Phi(\vec{z}):\chi(\vec{z})\Rightarrow\psi(\vec{z})}(s,t) \wedge \bigvee_{\mu(\vec{y})\in L(\vec{y})} \psi(\vec{z}) = \mu(\vec{y}) \Big) \tag{14}$$

Let A be a function into sort ACTION.

$$\mathit{Activated}_L(\vec{y}, A(\vec{x}), s, t) \stackrel{\text{def}}{=} \mathit{IndActivated}_L(\vec{y}, s, t) \vee \bigvee_{F(\vec{y})\in L(\vec{y})} \mathit{DirT}(F(\vec{y}), A(\vec{x}), s, t) \vee \bigvee_{\neg F(\vec{y})\in L(\vec{y})} \mathit{DirF}(F(\vec{y}), A(\vec{x}), s, t) \tag{15}$$

Let f : FLUENT be a variable.

$$\mathit{LoopFrameT}(f, A(\vec{x}), s, t) \stackrel{\text{def}}{=} \mathit{Holds}(f, s) \wedge \bigvee_{\substack{L\in \mathit{Loops}(\mathcal{R}),\\ \neg F(\vec{y})\in L(\vec{y})}} (\exists \vec{y}) \Big(f = F(\vec{y}) \wedge \neg \mathit{Activated}_L(\vec{y}, A(\vec{x}), s, t) \Big) \tag{16}$$

Equality for fluent literals is just an abbreviation—if both are positive or both are negative, we compare their affirmative components; if they have different signs, they cannot be equal.

$$\varphi = \psi \stackrel{\text{def}}{=} \begin{cases} |\varphi| = |\psi| & \text{if } \varphi = |\varphi| \text{ iff } \psi = |\psi| \\ \bot & \text{otherwise} \end{cases}$$

Note that if the specified causal relationships do not give rise to any loops, both (16) and its negative version are equivalent to $\bot$. The new causes are added to the effect axiom in the usual way.

Definition 10. Let A be a function into sort ACTION. An *effect axiom with conditional effects, the frame assumption, and ramifications* is of the form (7), where

$$\mathit{CausedT}(f, A(\vec{x}), s, t) \stackrel{\text{def}}{=} \mathit{FrameT}(f, s, t) \vee \mathit{DirT}(f, A(\vec{x}), s, t) \vee \mathit{IndT}(f, s, t) \vee \mathit{LoopFrameT}(f, A(\vec{x}), s, t) \tag{17}$$

Effect axioms taking special care in loop activation are now able to treat the gear wheel domain correctly.

Example 2 (Continued)**.** The Wait action has no direct effects and for neither of the loops L_1, L_2 exists a rule leading into the loop, hence

$$\mathit{Activated}_{L_1}(\mathsf{Wait}, s, t) = \mathit{Activated}_{L_2}(\mathsf{Wait}, s, t) = \bot$$

Accordingly, the new causes added to the effect axiom state that through performing Wait, the truth value of the loop literals must persist: $\mathit{LoopFrameT}(f, \mathsf{Wait}, s, t)$ is equivalent to $\mathit{Holds}(f, s) \wedge (f = \mathsf{W}_1 \vee f = \mathsf{W}_2)$; $\mathit{LoopFrameF}(f, \mathsf{Wait}, s, t)$, in turn, is equivalent to $\neg \mathit{Holds}(f, s) \wedge (f = \mathsf{W}_1 \vee f = \mathsf{W}_2)$.

For the undesired interpretation I seen earlier we now have $I \models \mathit{LoopFrameF}(\mathsf{W}_1, \mathsf{Wait}, T_0, T_1)$ but $I \not\models \neg \mathit{Holds}(\mathsf{W}_1, T_1)$; hence I is no model for the effect axiom any more, just as desired.

Formal Assessment

In order to provide an assessment of our solution to the ramification problem, we now develop a formal correspondence between the solution of (Van Belleghem, Denecker, and Theseider-Dupré 1998) and ours for a particular class of action domains. Their approach is well-suited for this purpose because it is a general, time-independent solution that can cope with cyclic dependencies. Yet, due to its non-standard semantics, it is (until now) not clear how existing action calculi can benefit from it. Our correspondence result thus takes an important step towards integrating that solution into specific formalisms that can be obtained as instances of the unifying action calculus.

To begin with, we introduce a class of action domains that both formalisms can treat, but which is still independent of a specific notion of time. The restriction to propositional actions stems from (Van Belleghem, Denecker, and Theseider-Dupré 1998), where parametric actions play no role. They however consider more general specifications of indirect effects where the trigger is a fluent formula instead of a fluent literal. (They also consider simultaneous actions, which we could emulate by defining appropriate new actions.)

Definition 11. Let $\mathcal{F}$ be a finite set of function symbols, $\mathcal{A}$ be a finite set of action constants, $\mathcal{E}$ be a family of sets Γ_A of expressions Φ/ψ (one for each $A \in \mathcal{A}$), and $\mathcal{R}$ be a set of causal relationships.

The *UAC axiomatization of* $(\mathcal{F}, \mathcal{A}, \mathcal{E}, \mathcal{R})$ consists of a domain signature where the sort FLUENT coincides with $\mathcal{F}$ and the sort ACTION coincides with $\mathcal{A}$, and a set Υ of effect axioms incorporating $\mathcal{R}$ according to Definitions 9 and 10, one for each $A \in \mathcal{A}$.

The *domain definition* Ξ *for* $(\mathcal{F}, \mathcal{A}, \mathcal{E}, \mathcal{R})$ *according to (Van Belleghem, Denecker, and Theseider-Dupré 1998)* consists of: $\mathcal{F}$ and $\mathcal{A}$ as they are; for all actions $A \in \mathcal{A}$ and $\Phi/\psi \in \Gamma_A$, a statement A `causes` ψ `if` Φ; for each causal relationship $\Phi : \chi \Rightarrow \psi \in \mathcal{R}$, a statement `initiating` χ `causes` ψ `if` Φ.

For a domain definition Ξ, an action $A \in \mathcal{A}$, and a state[3] S over $\mathcal{F}$, (Van Belleghem, Denecker, and Theseider-Dupré

[3]A state is a maximally consistent set of fluent literals.

1998) define the *successor state* $Succ_\Xi(A, S)$ *of* S *after* A as $(S \setminus \{\mu \mid \mathbf{Init}(A, S, \neg\mu)\}) \cup \{\mu \mid \mathbf{Init}(A, S, \mu)\}$.[4] There, $\mathbf{Init}(A, S, \mu)$ means "action A in state S initiates truth of fluent literal μ" and is roughly defined via the following inductive definition rules:[5]

- for each direct effect expression A `causes` ψ `if` Φ, the rule $\mathbf{Caus}(A, S, \psi) \leftarrow \mathrm{Ho}(\Phi, S)$;
- for each `initiating` χ `causes` ψ `if` Φ, the rule $\mathbf{Caus}(A, S, \psi) \leftarrow \mathbf{Init}(A, S, \chi), \neg\mathrm{Ho}(\chi, S), \mathrm{Ho}(\Phi, S)$;
- the rules $\mathbf{Init}(A, S, \mu) \leftarrow \mathbf{Caus}(A, S, \mu), \neg\mathrm{Ho}(\mu, S)$ for all fluent literals μ.

For the details we must refer the reader to the original work (Van Belleghem, Denecker, and Theseider-Dupré 1998) due to a lack of space. We remark however that calculating **Init** really only provides the indirect effects, the frame assumption is subsequently implemented by set arithmetics.

The main assessment result now states that our automatically created effect axioms correctly capture the state-transition semantics of (Van Belleghem, Denecker, and Theseider-Dupré 1998) for the class of domains defined above.

Theorem 1. *Let* $\mathcal{F}$ *be a finite set of function symbols,* $\mathcal{A}$ *be a finite set of action constants,* $\mathcal{E}$ *be a family of sets* Γ_A *of expressions* Φ/ψ *(one for each* $A \in \mathcal{A}$*),* $\mathcal{R}$ *be a set of causal relationships,* Υ *the set of effect axioms (7) from the UAC axiomatization of* $(\mathcal{F}, \mathcal{A}, \mathcal{E}, \mathcal{R})$*,* Ξ *be the domain definition for* $(\mathcal{F}, \mathcal{A}, \mathcal{E}, \mathcal{R})$ *according to (Van Belleghem, Denecker, and Theseider-Dupré 1998), and* S_1, S_2 *be states over* $\mathcal{F}$.

Now for some $A \in \mathcal{A}$*, let* $\Upsilon_A[s, t] \in \Upsilon$ *be A's effect axiom. Define an interpretation* I *for* $\Upsilon_A[\tau_1, \tau_2]$ *as follows:* $\mathrm{TIME}^I \overset{def}{=} \{\tau_1, \tau_2\}$, $\tau_1 <^I \tau_2$, $Poss^I \overset{def}{=} \{(A, \tau_1, \tau_2)\}$, $Holds^I \overset{def}{=} \{(F, \tau_i) \mid F \in S_i\}, i = 1, 2$. *We now have:*

$$I \models \Upsilon_A[\tau_1, \tau_2] \text{ iff } Succ_\Xi(A, S_1) = S_2$$

Proof. See the appendix for a sketch. □

Discussion

We proposed a treatment of ramifications that is neither tied to a particular action calculus nor to a particular notion of time. Indirect effects are specified through causal relationships, that express effect propagation in a concrete action domain. These rules are then compiled into effect axioms that provide a combined solution to the frame and ramification problems. We assessed the solution by providing a formal correspondence with a previously established general, calculus-independent solution.

Some notes on related work are in order. (Thielscher 1997) introduced the concept of (closed) causal relationships and provided a solution to the ramification problem, albeit restricted to a specific formalism. (Van Belleghem, Denecker, and Theseider-Dupré 1998) provide a way of modeling indirect effects that is independent of a specific calculus; they however use a three-valued semantics in form of a state transition function. (Shanahan 1999) shows how to handle a particular class of ramifications in the event calculus (Kowalski and Sergot 1986), limited however in that it admittedly cannot treat self-justifying cycles. This is also true for (McCain and Turner 1995) and (Giunchiglia et al. 2004), who addressed the ramification problem with the help of a special causal logic; our general first-order axiom for indirect effects (but without loop formulas) was in fact inspired by their concept of causality. In an approach basically similar to ours, (Pinto 1999) aims at compiling ramification constraints into effect axioms by some preprocessing, yet it does not integrate causality. (McIlraith 2000) uses an implicit notion of causation, but the approach resorts to a minimal-model policy and is not able to deal with cyclic fluent dependencies, since it requires stratification of the ramification constraints. More recently, (Herzig and Varzinczak 2007) dealt with indirect effects of actions through static laws in a special formalism based on modal logic. (Forth and Miller 2007) proposed a treatment of indirect effects in the event calculus based on nested circumscription—the general solution there is hence not always expressible in first-order logic. Additionally, being formulated in the event calculus, the approach is bound to linear time and a narrative-based semantics. Most recently, (Baumann et al. 2010) provide a treatment of ramifications in a general-purpose formalism which is independent of the underlying time structure. They translate causal relationships into Reiter defaults (Reiter 1980) and then use reasoning methods from default logic to infer indirect effects. The approach treats cycles properly (due to the groundedness of extensions in default logic), but it is again outside of the scope of classical logic.

Summing up the novel aspects of our approach, it is: (1) independent of the underlying time structure, (2) formulated in pure first-order logic, and (3) it can cope with the important and challenging problem of cyclic fluent dependencies.

[4]The successor state is assumed to be undefined when the resulting set of literals is not a valid state or when computation of **Init** revealed an inconsistency in the (direct and indirect) effect specifications.

[5]$\mathbf{Caus}(A, S, \mu)$ means "A in S causes μ"; $\mathrm{Ho}(\Phi, S)$ denotes the truth value of Φ in S.

References

Baker, A. B. 1991. Nonmonotonic Reasoning in the Framework of Situation Calculus. *Artificial Intelligence* 49:5–23.

Baumann, R.; Brewka, G.; Strass, H.; Thielscher, M.; and Zaslawski, V. 2010. State Defaults and Ramifications in the Unifying Action Calculus. In *Proceedings of the Twelfth International Conference on the Principles of Knowledge Representation and Reasoning*. To appear.

Chen, Y.; Lin, F.; Wang, Y.; and Zhang, M. 2006. First-Order Loop Formulas for Normal Logic Programs. In *Proceedings of the Tenth International Conference on Principles of Knowledge Representation and Reasoning (KR 2006)*, 298–307. AAAI Press.

Clark, K. L. 1978. Negation as Failure. In Gallaire, H., and Minker, J., eds., *Logic and Data Bases*, 293–322. Plenum Press.

Forth, J., and Miller, R. 2007. Ramifications: An Extension and Correspondence Result for the Event Calculus. *Journal of Logic and Computation* 17(4):639–685.

Gelfond, M., and Lifschitz, V. 1991. Classical Negation in Logic Programs and Disjunctive Databases. *New Generation Computing* 9:365–385.

Ginsberg, M. L., and Smith, D. E. 1987. Reasoning about Action I: A Possible Worlds Approach. *Artificial Intelligence* 35:233–258.

Giunchiglia, E.; Lee, J.; Lifschitz, V.; McCain, N.; and Turner, H. 2004. Nonmonotonic Causal Theories. *Artificial Intelligence* 153(1-2):49–104.

Herzig, A., and Varzinczak, I. J. 2007. Metatheory of actions: Beyond consistency. *Artificial Intelligence* 171(16–17):951–984.

Kowalski, R. A., and Sergot, M. J. 1986. A Logic-based Calculus of Events. *New Generation Computing* 4(1):67–95.

Lin, F., and Reiter, R. 1994. State Constraints Revisited. *Journal of Logic and Computation* 4(5):655–677.

Lin, F., and Zhao, Y. 2004. ASSAT: Computing Answer Sets of a Logic Program by SAT Solvers. *Artificial Intelligence* 157(1-2):115–137.

Lin, F. 1995. Embracing Causality in Specifying the Indirect Effects of Actions. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI-95)*, 1985–1993. Morgan Kaufmann.

McCain, N., and Turner, H. 1995. A Causal Theory of Ramifications and Qualifications. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI-95)*, 1978–1984. Montréal, Québec, Canada: Morgan Kaufmann.

McIlraith, S. 2000. Integrating Actions and State Constraints: A Closed-Form Solution to the Ramification Problem (Sometimes). *Artificial Intelligence* 116(1–2):87–121.

Pinto, J. 1999. Compiling Ramification Constraints into Effect Axioms. *Computational Intelligence* 15:280–307.

Reiter, R. 1980. A Logic for Default Reasoning. *Artificial Intelligence* 13:81–132.

Shanahan, M. 1999. The Ramification Problem in the Event Calculus. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI-99)*, 140–146. Morgan Kaufmann.

Thielscher, M. 1995. Computing Ramifications by Postprocessing. In Mellish, C. S., ed., *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI-95)*, 1994–2000. Montreal, Canada: Morgan Kaufmann.

Thielscher, M. 1997. Ramification and Causality. *Artificial Intelligence* 89(1–2):317–364.

Thielscher, M. 2010. A Unifying Action Calculus. *Artificial Intelligence*. To appear.

Van Belleghem, K.; Denecker, M.; and Theseider-Dupré, D. 1998. A Constructive Approach to the Ramification Problem. In *Reasoning about Actions; Foundations and Applications*, 1–17.

Winslett, M. 1988. Reasoning about Action Using a Possible Models Approach. In *Proceedings of the Seventh National Conference on Artificial Intelligence (AAAI-88)*, 89–93.

Appendix

Proof of Theorem 1 (Sketch). Throughout the proof, we use fluent literals and the literal versions of the cause-macros Caused, Frame, Dir, Ind, $\mathit{LoopFrame}$; for example, for the fluent literal $\mu = \neg f$, $\mathit{Caused}(\mu, a, s, t) = \mathit{CausedF}(f, a, s, t)$ and $\mathit{Caused}(\neg\mu, a, s, t) = \mathit{CausedT}(f, a, s, t)$. An important observations concerning the proof is that by definition of I, for all fluent literals, we have $\mu \in S_i$ iff $I \models \mu[\tau_i]$, $i = 1, 2$. This can be generalized to fluent formulas Φ: $\mathrm{Ho}(\Phi, S_1)$ iff $I \models \Phi[\tau_1]$. Atoms of the form $\mathrm{Ho}(\Phi, S_1)$ form the leaves of *proof trees*, that define the semantics of **Init** in (Van Belleghem, Denecker, and Theseider-Dupré 1998). The proof itself is much simplified by the below lemmata, all of which use the presumptions of Theorem 1. The first one can be proved by induction on the height of the (true) proof tree in the "only if" direction and induction on the length of the causal chain leading to μ in the converse.

Lemma 2. *For $I \models \Upsilon_A[\tau_1, \tau_2]$ and a fluent literal μ, we have that $\mathbf{Init}(A, S_1, \mu)$ is true if and only if $I \models \neg\mu[\tau_1] \wedge \mu[\tau_2]$.*

The proof of the second lemma is fairly straightforward since the only definition rules with head **Caus** are the ones created through grounding of direct and indirect effect rules.

Lemma 3. *Let $\mathit{Succ}_\Xi(A, S_1) = S_2$ and μ be a fluent literal. $\mathbf{Caus}(A, S_1, \mu)$ is true if and only if $I \models \mathit{Dir}(\mu, A, \tau_1, \tau_2) \vee \mathit{Ind}(\mu, \tau_1, \tau_2)$.*

The last lemma is an easy corollary of an observation of (Van Belleghem, Denecker, and Theseider-Dupré 1998).

Lemma 4. *Let $\mathit{Succ}_\Xi(A, S_1)$ be a state and μ be a fluent literal. $\mathbf{Init}(A, S_1, \mu)$ is true if and only if $\mathbf{Caus}(A, S_1, \mu)$ is true and $\mu \notin S_1$.*

The claim of Theorem 1 is now straightforwardly shown. In the "if" direction, we can use the presumption $\mathit{Succ}_\Xi(A, S_1) = S_2$ and Lemmata 4 and 3 to show that for all fluent literals μ, we have $I \models \mu[\tau_2]$ iff $I \models \mathit{Caused}(\mu, A, \tau_1, \tau_2)$ and $I \models \neg\mathit{Caused}(\neg\mu, A, \tau_1, \tau_2)$. This is done via a case distinction on why $\mu \in \mathit{Succ}_\Xi(A, S_1)$— either $\mathbf{Init}(A, S_1, \mu)$ is true or $\mu \in S_1$ and $\mathbf{Init}(A, S_1, \neg\mu)$ is not true. In the "only if"-direction, the presumption $I \models \Upsilon_A[\tau_1, \tau_2]$ enables the use of Lemma 2. To show $\mathit{Succ}_\Xi(A, S_1) \subseteq S_2$, we again make a case distinction on why $\mu \in \mathit{Succ}_\Xi(A, S_1)$ and in each case employ Lemma 2; $\mathit{Succ}_\Xi(A, S_1) \supseteq S_2$ can be shown by a case distinction whether $\mu \in S_1$ and (in each case) use of Lemma 2. □