Available online at www.sciencedirect.com

ScienceDirect

Artificial Intelligence 171 (2007) 951–984

Artificial Intelligence

www.elsevier.com/locate/artint

Metatheory of actions: Beyond consistency

Andreas Herzig, Ivan Varzinczak *

IRIT – Université Paul Sabatier, 118 route de Narbonne, 31062, Toulouse Cedex 9, France

Received 25 April 2006; received in revised form 15 March 2007; accepted 23 April 2007

Available online 29 April 2007

Abstract

Traditionally, consistency is the only criterion for the quality of a theory in logic-based approaches to reasoning about actions. This work goes beyond that and contributes to the metatheory of actions by investigating what other properties a good domain description should have. We state some metatheoretical postulates concerning this sore spot. When all postulates are satisfied we call the action theory modular. Besides being easier to understand and more elaboration tolerant in McCarthy's sense, modular theories have interesting properties. We point out the problems that arise when the postulates about modularity are violated, and propose algorithmic checks that can help the designer of an action theory to overcome them.

Keywords: Reasoning about actions; Action theory; Modularity; Ramifications

1. Introduction

In logic-based approaches to knowledge representation, a given domain is described by a set of logical formulas $\mathcal{T}$, which we call a (non-logical) *theory*. That is also the case for reasoning about actions, where we are interested in theories describing particular actions (or, more precisely, action types). We call such theories *action theories*.

A priori consistency is the only criterion that formal logic provides to check the quality of such descriptions. In the present work we go beyond that, and argue that we should require more than the mere existence of a model for a given theory.

Our starting point is the fact that in reasoning about actions one usually distinguishes several kinds of logical formulas. Among these are effect axioms, precondition axioms, and boolean axioms. In order to distinguish such non-logical axioms from logical axioms, we prefer to speak of effect laws, executability laws, and static laws, respectively. Moreover we single out those effect laws whose effect is $\bot$, and call them inexecutability laws.

Given these types of laws, suppose the language is powerful enough to state conditional effects of actions. For example, suppose that action a is inexecutable in contexts where φ_1 holds, and executable in contexts where φ_2 holds. It follows that there can be no context where $\varphi_1 \wedge \varphi_2$ holds. Now $\neg(\varphi_1 \wedge \varphi_2)$ is a static law that does not mention a. It is natural to expect that $\neg(\varphi_1 \wedge \varphi_2)$ follows from the static laws alone. By means of examples we show that when

* Corresponding author.
E-mail addresses: herzig@irit.fr (A. Herzig), ivan@irit.fr (I. Varzinczak).

0004-3702/$ – see front matter
doi:10.1016/j.artint.2007.04.013

this is not the case, then unexpected conclusions might follow from the theory $\mathcal{T}$, even in the case $\mathcal{T}$ is logically consistent.

This motivates postulates requiring that the different laws of an action theory should be arranged modularly, i.e., in separated components, and in such a way that interactions between them are limited and controlled. In essence, we argue that static laws may entail new effects of actions (that cannot be inferred from the effect laws alone), while effect laws and executability laws should never entail new static laws that do not follow from the set of static laws alone. We here formulate postulates that make these requirements precise. It will turn out that in all existing accounts that allow for these four kinds of laws [1–6], consistent action theories can be written that violate these postulates. In this work we give algorithms that allow one to check whether an action theory satisfies the postulates or not. With such algorithms, the task of correcting flawed action theories can be made easier.

Although we here use the syntax of propositional dynamic logic (PDL) [7], all we shall say applies as well to first-order formalisms, in particular to the Situation Calculus [8]. All postulates we are going to present can be stated as well for other frameworks, in particular for action languages such as $\mathcal{A}$, $\mathcal{AR}$ [9–11] and others, and for Situation Calculus based approaches. In [12] we have given a Situation Calculus version of our analysis, while in [13] we presented a similar notion for ontologies in Description Logics [14]. The present work is the complete version of the one first appeared in [15].

This text is organized as follows: after some background definitions (Section 2) we state some postulates concerning action theories (Section 3). In Sections 4 and 5, we study the two most important of these postulates, giving algorithmic methods to check whether an action theory satisfies them or not. We then generalize our postulates (Section 6) and discuss possible strengthening of them (Section 7). In Section 8 we show interesting features of modular action theories. Before concluding, we assess related work found in the literature on metatheory of actions (Section 9).

2. Preliminaries

2.1. Dynamic logic

Here we establish an ontology of dynamic domains. As our base formalism we use $*$-free PDL, i.e., PDL without the iteration operator $*$. For more details on PDL, see [7,16].

Let $\mathfrak{Act} = \{a_1, a_2, \ldots\}$ be the set of all *atomic action constants* of a given domain. Our running example is in terms of the Walking Turkey Scenario [4]. There, the atomic actions are *load*, *shoot* and *tease*. We use a as a variable for atomic actions. To each atomic action a there is an associated modal operator $[a]$. Here we suppose that the underlying multimodal logic is independently axiomatized (i.e., the logic is a fusion and there is no interaction between the modal operators [17,18]).

$\mathfrak{Prop} = \{p_1, p_2, \ldots\}$ denotes the set of all *propositional constants*, also called *fluents* or *atoms*. Examples of those are *loaded*, *alive* and *walking*. We use p as a variable for propositional constants.

We here suppose that both $\mathfrak{Act}$ and $\mathfrak{Prop}$ are nonempty and finite.

We use small Greek letters $\varphi, \psi, \ldots$ to denote *classical formulas*, also called *boolean* formulas. They are recursively defined in the following way:

$$\varphi ::= p \mid \top \mid \bot \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \varphi \rightarrow \varphi \mid \varphi \leftrightarrow \varphi.$$

$\mathfrak{Fml}$ is the set of all classical formulas.

Examples of classical formulas are *walking* $\rightarrow$ *alive* and $\neg$(*bachelor* $\wedge$ *married*).

A classical formula is *classically consistent* if there is at least one valuation in classical propositional logic that makes it true. Given $\varphi \in \mathfrak{Fml}$, *valuations*($\varphi$) denotes the set of all valuations of φ. We note $\models_{\mathsf{CPL}}$ the logical consequence in classical propositional logic.

The set of all literals is $\mathfrak{Lit} = \mathfrak{Prop} \cup \{\neg p\colon\ p \in \mathfrak{Prop}\}$. Examples of literals are *alive* and $\neg$*walking*. ℓ will be used as a variable for literals. If $\ell = \neg p$, then we identify $\neg\ell$ with p.

A *clause* χ is a disjunction of literals. We say that a literal ℓ *appears* in a clause χ, written $\ell \in \chi$, if ℓ is a disjunct of χ.

We denote complex formulas (possibly with modal operators) by capital Greek letters $\Phi_1, \Phi_2, \ldots$ They are recursively defined in the following way:

$$\Phi ::= \varphi \mid [a]\Phi \mid \langle a\rangle\Phi \mid \neg\Phi \mid \Phi \wedge \Phi \mid \Phi \vee \Phi \mid \Phi \rightarrow \Phi \mid \Phi \leftrightarrow \Phi$$

where Φ denotes a complex formula. $\langle a\rangle$ is the dual operator of $[a]$, defined by: $\langle a\rangle\Phi =_{\text{def}} \neg[a]\neg\Phi$. Sequential composition of actions is defined by the abbreviation $[a_1; a_2]\Phi =_{\text{def}} [a_1][a_2]\Phi$. Examples of complex formulas are $loaded \to [shoot]\neg alive$ and $hasGun \to \langle load; shoot\rangle(\neg alive \wedge \neg loaded)$.

If $\mathcal{T}$ is a set of formulas (modal or classical), $atm(\mathcal{T})$ returns the set of all atoms occurring in $\mathcal{T}$. For instance, $atm(\{\neg\neg\neg p_1, [a]p_2\}) = \{p_1, p_2\}$.

For parsimony's sake, whenever there is no confusion we identify a set of formulas with the conjunction of its elements. The semantics is that for multimodal K [19,20].

Definition 1. A PDL-*model* is a tuple $\mathcal{M} = \langle W, R\rangle$ where W is a set of valuations (alias possible worlds), and $R : \mathfrak{Act} \longrightarrow 2^{W\times W}$ a function mapping action constants a to accessibility relations $R_a \subseteq W \times W$.

As an example, for $\mathfrak{Act} = \{a_1, a_2\}$ and $\mathfrak{Prop} = \{p_1, p_2\}$, we have the PDL-model $\mathcal{M} = \langle W, R\rangle$, where

$$W = \big\{\{p_1, p_2\}, \{p_1, \neg p_2\}, \{\neg p_1, p_2\}\big\},$$

$$R(a_1) = \left\{\begin{array}{l}(\{p_1, p_2\}, \{p_1, \neg p_2\}), (\{p_1, p_2\}, \{\neg p_1, p_2\}),\\ (\{\neg p_1, p_2\}, \{\neg p_1, p_2\}), (\{\neg p_1, p_2\}, \{p_1, \neg p_2\})\end{array}\right\},$$

$$R(a_2) = \big\{\big(\{p_1, p_2\}, \{p_1, \neg p_2\}\big), \big(\{p_1, \neg p_2\}, \{p_1, \neg p_2\}\big)\big\}.$$

Fig. 1 gives a graphical representation of $\mathcal{M}$.

Given $\mathcal{M} = \langle W, R\rangle$, $a \in \mathfrak{Act}$, and $w, w' \in W$, we write R_a instead of $R(a)$, and wR_aw' instead of $w' \in R_a(w)$.

Definition 2. Given a PDL-model $\mathcal{M} = \langle W, R\rangle$, the satisfaction relation is defined as the smallest relation satisfying:

- $\models^{\mathcal{M}}_w p$ (p is true at world w of model $\mathcal{M}$) if $p \in w$;
- $\models^{\mathcal{M}}_w [a]\Phi$ if for every w' such that wR_aw', $\models^{\mathcal{M}}_{w'} \Phi$; and
- the usual truth conditions for the other connectives.

Definition 3. A PDL-model $\mathcal{M}$ is a model of Φ (noted $\models^{\mathcal{M}} \Phi$) if and only if for all $w \in W$, $\models^{\mathcal{M}}_w \Phi$. $\mathcal{M}$ is a model of a set of formulas $\mathcal{T}$ (noted $\models^{\mathcal{M}} \mathcal{T}$) if and only if $\models^{\mathcal{M}} \Phi$ for every $\Phi \in \mathcal{T}$.

In the model depicted in Fig. 1, we have $\models^{\mathcal{M}} p_1 \to [a_2]\neg p_2$ and $\models^{\mathcal{M}} p_1 \vee p_2$.

Definition 4. A formula Φ is a *consequence of the set of global axioms* $\mathcal{T}$ in the class of all PDL-models (noted $\mathcal{T} \models_{\text{PDL}} \Phi$) if and only if for every PDL-model $\mathcal{M}$, if $\models^{\mathcal{M}} \mathcal{T}$, then $\models^{\mathcal{M}} \Phi$.[1]

We here suppose that the logic under consideration is *compact* [21].

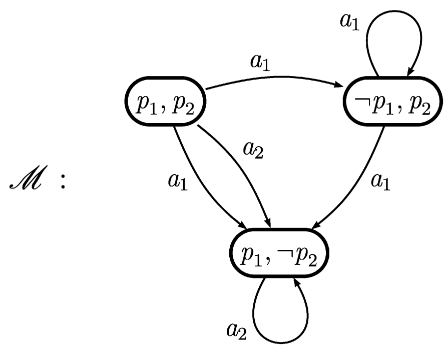

Fig. 1. Example of a PDL-model for $\mathfrak{Act} = \{a_1, a_2\}$, and $\mathfrak{Prop} = \{p_1, p_2\}$.

[1] Instead of global consequence, in [5] local consequence is considered. For that reason, a further modal operator $\Box$ had to be introduced, giving a logic that is multimodal K plus monomodal S4 for $\Box$, and where axiom schema $\Box\Phi \to [a]\Phi$ holds.

Having established the formal substratum our presentation will rely on, we present in the next section the different types of formulas we use to describe dynamic domains.

2.2. Describing action theories in PDL

Before elaborating a theory, we need to specify what we are about to describe, i.e., what the formulas talk about. Following the tradition in the literature, we identify a domain (alias scenario) with the actions we take into account and the fluents they can change. More formally, we have:

Definition 5. A *domain signature* is a tuple $\langle \mathfrak{Act}, \mathfrak{Prop} \rangle$.

An example of a domain is the well-known Yale Shooting Scenario (YSS) [22], whose signature comprises the actions *load*, *wait* and *shoot*, and fluents *loaded* and *alive*.

Given a domain $\langle \mathfrak{Act}, \mathfrak{Prop} \rangle$, we are interested in theories whose statements describe the behavior of actions of $\mathfrak{Act}$ on the fluents of $\mathfrak{Prop}$. PDL allows for the representation of such statements, that we here call *action laws*. We distinguish several types of them. We call *effect laws* formulas relating an action to its effects. Statements of conditions under which an action cannot be executed are called *inexecutability laws*. *Executability laws* in turn stipulate the context where an action is guaranteed to be executable. Finally, *static laws* are formulas that do not mention actions. They express constraints that must hold in every possible state. These four types of laws are our fundamental entities and we introduce them more formally in the sequel.

2.2.1. Static laws

Frameworks which allow for indirect effects of actions make use of logical formulas that state invariant propositions about the world. Such formulas delimit the set of possible states. They do not refer to actions, and we suppose here that they are expressed as formulas of classical propositional logic.

Definition 6. A *static law* is a formula $\varphi \in \mathfrak{Fml}$.

In our running example, the static law $walking \to alive$ says that if a turkey is walking, then it must be alive. Another one is $saved \leftrightarrow (mbox1 \vee mbox2)$, which states that an e-mail message is saved if and only if it is in mailbox 1 or in mailbox 2 or both [23].

In some action languages, such as $\mathcal{AR}$ for example, we would write the statement always $alive \to walking$, and in the Situation Calculus it would be the first-order formula

$$\forall s.\big(Holds(walking, s) \to Holds(alive, s)\big).$$

The set of all static laws of a given domain is denoted by $\mathcal{S}$. At first glance, no requirement concerning consistency of $\mathcal{S}$ is made. Of course, we want $\mathcal{S}$ to be consistent, otherwise the whole theory is inconsistent. As we are going to see in the sequel, however, consistency of $\mathcal{S}$ alone is not enough to guarantee the consistency of a theory.

2.2.2. Effect laws

Logical frameworks for reasoning about actions contain expressions linking actions and their effects. We suppose that such effects might be conditional, and thus get a third component of such laws.

In PDL, the formula $[a]\Phi$ states that formula Φ is true after every possible execution of action a.

Definition 7. An *effect law*[2] *for action a* is of the form $\varphi \to [a]\psi$, where $\varphi, \psi \in \mathfrak{Fml}$, with ψ classically consistent.

The consequent ψ is the effect which obtains when action a is executed in a state where the antecedent φ holds. An example of an effect law is $loaded \to [shoot]\neg alive$, saying that whenever the gun is loaded, after shooting the

[2] Effect laws are often called *action laws*, but we prefer not to use that term here because it would also apply to executability laws that are to be introduced in the sequel.

turkey is dead. Another one is $\top \to [tease]walking$: in every circumstance, the result of teasing is that the turkey starts walking. For parsimony's sake, the latter effect law will be written $[tease]walking$.

Note that the consistency requirement for ψ makes sense: if ψ is inconsistent then we have an inexecutability law, that we consider as a separate entity and which we are about to introduce formally in the sequel. On the other hand, if φ is inconsistent then the effect law is obviously superfluous.

For the first example above, in action languages one would write the statement

$$shoot \text{ causes } \neg alive \text{ if } loaded,$$

and in the Situation Calculus formalism one would write the first-order formula

$$\forall s.\big(Holds(loaded, s) \to \neg Holds\big(alive, do(shoot, s)\big)\big).$$

2.2.3. Inexecutability laws

We consider effect laws with inconsistent consequents as a particular kind of law which we call inexecutability laws. (Such laws are sometimes called qualifications [24].) This allows us to avoid mixing things that are conceptually different: for an action a, an effect law mainly associates it with a consequent ψ, while an inexecutability law only associates it with an antecedent φ, viz. the context which precludes the execution of a.

Definition 8. An *inexecutability law for action a* is of the form $\varphi \to [a]\bot$, with $\varphi \in \mathfrak{Fml}$.

For example $\neg hasGun \to [shoot]\bot$ expresses that *shoot* cannot be executed if the agent has no gun. Another example is $dead \to [tease]\bot$: a dead turkey cannot be teased.

In $\mathcal{AR}$ we would write the statement impossible if $\neg hasGun$, and in the Situation Calculus our example would be

$$\forall s.\big(\neg Holds(hasGun, s) \to \neg Poss(shoot, s)\big).$$

2.2.4. Executability laws

With only static and effect laws one cannot guarantee that the action *shoot* can be executed whenever the agent has a gun. We need thus a way to state such conditions.

In dynamic logic the dual $\langle a\rangle\varphi$, defined as $\neg[a]\neg\varphi$, can be used to express executability. $\langle a\rangle\top$ thus reads "execution of action a is possible".

Definition 9. An *executability law for action a* is of the form $\varphi \to \langle a\rangle\top$, where $\varphi \in \mathfrak{Fml}$.

For instance $hasGun \to \langle shoot\rangle\top$ says that shooting can be executed whenever the agent has a gun, and $\top \to \langle tease\rangle\top$, also written $\langle tease\rangle\top$, establishes that the turkey can always be teased.

Some approaches (most prominently Reiter's) use biconditionals $\varphi \leftrightarrow \langle a\rangle\top$, called precondition axioms. This is equivalent to $\neg\varphi \leftrightarrow [a]\bot$, which highlights that they merge information about inexecutability with information about executability. Here we consider these entities to be different and keep them separate.

In action languages such laws are not represented, they are rather implicitly inferred from inexecutability statements (cf. Section 7). In Situation Calculus our example would be stated as

$$\forall s.\big(Holds(hasGun, s) \to Poss(shoot, s)\big).$$

Whereas all the extant approaches in the literature that allow for indirect effects of actions contain static and effect laws, and provide a way for representing inexecutabilities (in the form of implicit qualifications [2,4,25]), the status of executability laws is less consensual. Some authors [3,4,26,27] more or less tacitly consider that executability laws should not be made explicit but rather inferred by the reasoning mechanism. Others [1,2,5,6] have executability laws as first class objects one can reason about.

It seems a matter of debate whether one can always do without executabilities. In principle it seems to be strange to just state information about necessary conditions for action execution (inexecutabilities) without saying anything about its sufficient conditions. The justification is that given an action we have three possible situations: it is known to be executable, known to be inexecutable, and unknown whether executable. This is the reason why we think that we need executability laws. Indeed, in several domains one wants to explicitly state under which conditions a given action is guaranteed to be executable, e.g. that a robot never gets stuck and is always able to execute a move action. And if we have a plan such as *load*; *shoot* (*load* followed by *shoot*) of which we know that it achieves the goal $\neg alive$, then we would like to be sure that it is executable in the first place![3] In any case, allowing for executability laws gives us more flexibility and expressive power.

2.2.5. Action theories

Given a domain $\langle \mathfrak{Act}, \mathfrak{Prop} \rangle$, for an action $a \in \mathfrak{Act}$, we define $\mathcal{E}^a$ as the set of its effect laws, $\mathcal{X}^a$ the set of its executability laws, and $\mathcal{I}^a$ that of its inexecutability laws.

Definition 10. An *action theory for a* is a tuple $\langle \mathcal{S}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a \rangle$.

In our running scenario example, a theory for the action *shoot* would be

$$\mathcal{S} = \{walking \to alive\}, \qquad \mathcal{E}^{shoot} = \{loaded \to [shoot]\neg alive\},$$
$$\mathcal{X}^{shoot} = \{hasGun \to \langle shoot \rangle \top\}, \qquad \mathcal{I}^{shoot} = \{\neg hasGun \to [shoot]\bot\}.$$

Given a dynamic domain we define

$$\mathcal{E} = \bigcup_{a \in \mathfrak{Act}} \mathcal{E}^a, \quad \mathcal{X} = \bigcup_{a \in \mathfrak{Act}} \mathcal{X}^a, \quad \text{and} \quad \mathcal{I} = \bigcup_{a \in \mathfrak{Act}} \mathcal{I}^a.$$

All these sets are finite, because $\mathfrak{Act}$ is finite and each of the $\mathcal{E}^a$, $\mathcal{X}^a$, $\mathcal{I}^a$ is finite.

Definition 11. An *action theory* is a tuple of the form $\langle \mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \rangle$.

Given an action theory $\langle \mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \rangle$ and a formula Φ, we write $\mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \models_{\mathsf{PDL}} \Phi$ instead of $\mathcal{S} \cup \mathcal{E} \cup \mathcal{X} \cup \mathcal{I} \models_{\mathsf{PDL}} \Phi$.

When formalizing dynamic domains, we face the *frame problem* [8] and the *ramification problem* [28]. In what follows we formally present the logical framework in which action theories will henceforth be described.

2.3. Dynamic logic and the frame problem

As the reader might have already expected, the logical formalism of PDL alone does not solve the frame problem. For instance, if $\langle \mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \rangle$ describes our shooting domain, then

$$\mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \not\models_{\mathsf{PDL}} hasGun \to [load]hasGun.$$

The same can be said about the ramification problem in what concerns the derivation of indirect effects not properly caused by the action under consideration. For example,

$$\mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \models_{\mathsf{PDL}} \neg alive \to [tease]alive.$$

Thus, given an action theory $\langle \mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \rangle$, we need a consequence relation powerful enough to deal with the frame and ramification problems. This means that the deductive power of PDL has to be augmented in order to ensure that the only non-effects of actions that follow from the theory are those that are really relevant. The presence of static laws makes that this is a delicate task, and starting with [2,3], several authors have argued that some notion of causality is needed. In this work we opt for the dependence-based approach presented in [5], which has been shown [29] to subsume Reiter's solution to the frame problem [30], and moreover at least partially accounts for the ramification problem [31].

In the logical framework developed in [5], metalogical information, given in the form of a dependence relation, is added to PDL.

Definition 12 *(Dependence relation [5]).* A *dependence relation* is a binary relation $\leadsto \subseteq \mathfrak{Act} \times \mathfrak{Lit}$.

[3] Of course this would require a solution to the qualification problem [24].

The expression $a \rightsquigarrow \ell$ denotes that the execution of action *a may* make the literal ℓ true. In our example we have

$$\rightsquigarrow = \left\{ \begin{array}{l} \langle \mathit{shoot}, \neg \mathit{loaded}\rangle, \langle \mathit{shoot}, \neg \mathit{alive}\rangle, \\ \langle \mathit{shoot}, \neg \mathit{walking}\rangle, \langle \mathit{tease}, \mathit{walking}\rangle \end{array} \right\},$$

which means that action *shoot* may make the literals ¬*loaded*, ¬*alive* and ¬*walking* true, and action *tease* may make *walking* true.

Semantically, the dependence-based approach relies on the explanation closure assumption [26], and its solution to the frame problem consists in a kind of negation as failure: because $\langle \mathit{load}, \neg \mathit{hasGun}\rangle \notin \rightsquigarrow$, we have $\mathit{load} \not\rightsquigarrow \neg \mathit{hasGun}$, i.e., ¬*hasGun* is never caused by *load*. Thus, in a context where *hasGun* is true, after every execution of *load*, *hasGun* still remains true. We also have $\mathit{tease} \not\rightsquigarrow \mathit{alive}$ and $\mathit{tease} \not\rightsquigarrow \neg \mathit{alive}$. The meaning of all these independences is that the frame axioms $\mathit{hasGun} \rightarrow [\mathit{load}]\mathit{hasGun}$, $\neg \mathit{alive} \rightarrow [\mathit{tease}]\neg \mathit{alive}$ and $\mathit{alive} \rightarrow [\mathit{tease}]\mathit{alive}$ hold.

We assume that $\rightsquigarrow$ is finite.

A dependence relation $\rightsquigarrow$ defines a class of possible worlds models:

Definition 13. A PDL-model $\mathcal{M} = \langle W, R\rangle$ is a $\rightsquigarrow$*-model* if and only if whenever $wR_a w'$ then:

- if $a \not\rightsquigarrow p$, then $\not\models_w^{\mathcal{M}} p$ implies $\not\models_{w'}^{\mathcal{M}} p$; and
- if $a \not\rightsquigarrow \neg p$, then $\models_w^{\mathcal{M}} p$ implies $\models_{w'}^{\mathcal{M}} p$.

Fig. 2 depicts the dependence-based condition on models.

Given a $\rightsquigarrow$-model $\mathcal{M}$, Φ and $\mathcal{T}$, $\models^{\mathcal{M}} \Phi$ and $\models^{\mathcal{M}} \mathcal{T}$ are defined as in Definition 3.

Definition 14. A formula Φ is a $\rightsquigarrow$*-based consequence* of the set of global axioms $\mathcal{T}$ in the class of all $\rightsquigarrow$-models (noted $\mathcal{T} \models_{\rightsquigarrow} \Phi$) if and only if for every $\rightsquigarrow$-model $\mathcal{M}$, if $\models^{\mathcal{M}} \mathcal{T}$, then $\models^{\mathcal{M}} \Phi$.

In our example it thus holds

$$\mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \models_{\rightsquigarrow} \mathit{hasGun} \rightarrow [\mathit{load}]\mathit{hasGun}$$

and

$$\mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \models_{\rightsquigarrow} \neg \mathit{alive} \rightarrow [\mathit{tease}]\neg \mathit{alive}.$$

In this way, the dependence-based approach solves the frame problem. However, it does not entirely solve the ramification problem: while indirect effects such as $\mathit{loaded} \rightarrow [\mathit{shoot}]\neg \mathit{walking}$ can be deduced with $\models_{\rightsquigarrow}$ without explicitly stating that in the set of effect laws for *shoot*, we nevertheless still have to state *indirect dependences* such as $\mathit{shoot} \rightsquigarrow \neg \mathit{walking}$. However, according to Reiter's view:

> "what counts as a solution to the frame problem [...] is a systematic procedure for generating, from the effect laws, [...] a parsimonious representation for [all] the frame axioms" [32].

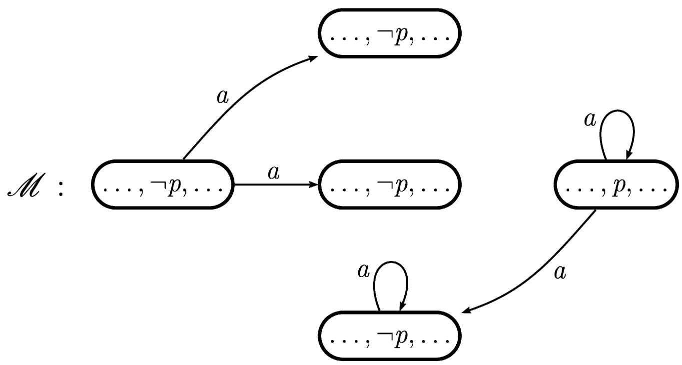

Fig. 2. Dependence-based condition: preservation of literal $\neg p$ under hypothesis $a \not\rightsquigarrow p$.

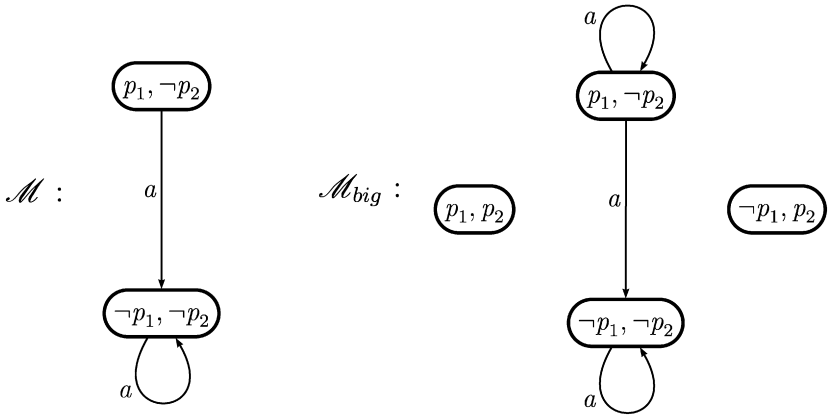

Fig. 3. A model of an action theory and its big model $\mathcal{M}_{big}$.

We comply with that as we can define a semi-automatic procedure for generating the dependence relation from the set of effect laws [33]. Moreover, as it has been argued in [23,31], our approach is in line with the state of the art because none of the existing solutions to the frame and the ramification problems can handle domains with both indeterminate and indirect effects.

In the next section we turn to a metatheoretical analysis of action theories and make a step toward formal criteria for theory evaluation. Before that, we need a definition.

Definition 15. Let $\langle \mathcal{S}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a \rangle$ be an action theory for a, and $\leadsto$ a dependence relation. Then $\mathcal{M} = \langle W, R \rangle$ is the *big* (alias *maximal/standard*) *model* for $\langle \mathcal{S}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a \rangle$ and $\leadsto$ if and only if:

- $\mathcal{M}$ is a $\leadsto$-model;
- $W = valuations(\mathcal{S})$ (all valuations of $\mathcal{S}$); and
- $R_a = \{(w, w')\colon$ for all $\varphi \to [a]\psi \in \mathcal{E}^a \cup \mathcal{I}^a$, if $\models^{\mathcal{M}}_{w} \varphi$, then $\models^{\mathcal{M}}_{w'} \psi\}$.

For an example, consider an action theory whose components are given by

$$\mathcal{S} = \emptyset, \quad \mathcal{E}^a = \{p_1 \to [a]\neg p_2\}, \quad \mathcal{X}^a = \{\langle a\rangle\top\},$$
$$\mathcal{I}^a = \{p_2 \to [a]\bot\}, \quad \text{and} \quad \leadsto = \{\langle a, \neg p_1\rangle, \langle a, \neg p_2\rangle\}.$$

Fig. 3 depicts one of its models and its associated big model.

Big models contain all valuations consistent with $\mathcal{S}$. Clearly, for a big model $\mathcal{M}$ we have $\models^{\mathcal{M}} \mathcal{S} \wedge \mathcal{E}^a \wedge \mathcal{I}^a$. Because $\mathcal{M}$ maximizes executability, it is only $\mathcal{X}^a$ which might not be true in $\mathcal{M}$.

In the rest of the paper we characterize when an action theory with a dependence relation has a big model.

3. The postulates

"When does a given action theory have a model?", and, more importantly, "is that model intended?" are questions that naturally arise when we talk about action theories. Here we claim that all the approaches that are put forward in the literature are too liberal in the sense that we can have satisfiable action theories that are intuitively incorrect. We argue that something beyond the consistency notion is required in order to help us in answering these questions.

Our central thesis is that the different types of laws defined in Section 2.2 should be neatly separated in modules. Besides that, we want such laws to interfere only in one sense: static laws together with action laws for a may have consequences that do not follow from the action laws for a alone. The other way round, action laws should not allow to infer new static laws; effect laws should not allow to infer inexecutability laws; action laws for a should not allow to infer action laws for a', etc. This means that our logical modules should be designed in such a way that they are as specialized and as little dependent on others as possible.

A first step in this direction has been the proposed division of our entities into the sets $\mathcal{S}$, $\mathcal{E}$, $\mathcal{X}$ and $\mathcal{I}$. In order to accomplish our goal, we have to diminish interaction among such modules, rendering them the least interwoven we can. The rest of the section contains postulates expressing this. We here only state the postulates, and defer explanations and discussions to Sections 4–6.

PC (Logical consistency): $\mathcal{S}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a \not\models_{\leadsto} \bot$.

The theory of a given action should be logically consistent.

PS (No implicit static laws):

$$\text{if } \mathcal{S}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a \models_{\leadsto} \varphi, \text{ then } \mathcal{S} \models_{\mathsf{CPL}} \varphi.$$

If a classical formula can be inferred from the action theory, then it should be inferable from the set of static laws alone. (Note that on the left we use the $\leadsto$-based consequence, while on the right we use consequence in classical logic: as both $\mathcal{S}$ and φ are classical, φ should be inferable from $\mathcal{S}$ in classical logic.)

PI (No implicit inexecutability laws):

$$\text{if } \mathcal{S}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a \models_{\leadsto} \varphi \rightarrow [a]\bot, \text{ then } \mathcal{S}, \mathcal{I}^a \models_{\mathsf{PDL}} \varphi \rightarrow [a]\bot.$$

If an inexecutability law for an action a can be inferred from its action theory, then it should be inferable in PDL from the static laws and the inexecutability laws for a alone. Note that we used $\models_{\mathsf{PDL}}$ instead of $\models_{\leadsto}$ because we also suppose that neither frame axioms nor indirect effects should be relevant to derive inexecutability laws. The same remark holds for the postulate that follows:

PX (No implicit executability laws):

$$\text{if } \mathcal{S}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a \models_{\leadsto} \varphi \rightarrow \langle a\rangle\top, \text{ then } \mathcal{S}, \mathcal{X}^a \models_{\mathsf{PDL}} \varphi \rightarrow \langle a\rangle\top.$$

If an executability law for a can be inferred from its action theory, then it should already "be" in $\mathcal{X}^a$, in the sense that it should also be inferable in PDL from the set of static and executability laws for a alone.

Postulate **PC** is obvious, for we are interested in consistent theories. It can be shown that **PX** is a consequence of **PS** (see Corollary 41).

Thus, while **PC** is obvious and **PX** can be ensured by **PS**, things are less obvious for Postulates **PS** and **PI**: it turns out that for all approaches in the literature they are easily violated by action theories that allow to express the four kinds of laws. We therefore study each of these postulates in the subsequent sections by means of examples, give algorithms to decide whether they are satisfied, and discuss about what to do in the case the answer is 'no'.

4. No implicit static laws

While executability laws increase expressive power, they might conflict with inexecutability laws. Consider, for example, the following action theory:

$$\mathcal{S}_1 = \{walking \rightarrow alive\}, \qquad \mathcal{E}_1 = \left\{ \begin{array}{l} [tease]walking, \\ loaded \rightarrow [shoot]\neg alive \end{array} \right\},$$
$$\mathcal{X}_1 = \{\langle tease\rangle\top\}, \qquad \mathcal{I}_1 = \{\neg alive \rightarrow [tease]\bot\}$$

and the dependence relation:

$$\leadsto = \left\{ \begin{array}{l} \langle shoot, \neg loaded\rangle, \langle shoot, \neg alive\rangle, \\ \langle shoot, \neg walking\rangle, \langle tease, walking\rangle \end{array} \right\}.$$

From this description we have the unintuitive inference $\mathcal{X}_1^{tease}, \mathcal{I}_1^{tease} \models_{\mathsf{PDL}} alive$: the turkey is immortal! This is an *implicit static law* because *alive* does not follow from $\mathcal{S}_1$ alone: $\langle \mathcal{S}_1, \mathcal{E}_1^{tease}, \mathcal{X}_1^{tease}, \mathcal{I}_1^{tease}\rangle$ violates Postulate **PS**.

Implicit static laws are not a drawback of our underlying logical formalism. They also appear in Situation Calculus-based approaches and in causal laws theories. To witness,[4] suppose in Lin's framework we have

$$Holds(p_1, s) \to Caused(p_2, true, s) \quad (1)$$

and

$$Caused(p_2, false, s). \quad (2)$$

Then from (2) we get

$$\neg Holds(p_2, s) \quad (3)$$

and then conclude

$$\neg Caused(p_2, true, s). \quad (4)$$

Finally, from (1) and (4) we get

$$\neg Holds(p_1, s)$$

which is an implicit static law.

To see how implicit static laws show up in McCain and Turner's causal laws approach [3], let the causal law $\varphi \Rightarrow \psi$ and $\mathcal{T} = \{\neg\psi\}$. Then $\neg\varphi$ is an implicit static law in such a description.

How can we find out whether an action theory for a with a dependence relation $\leadsto$ satisfies Postulate **PS**?

Theorem 16. *$\langle \mathcal{S}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a \rangle$ and $\leadsto$ satisfy Postulate* **PS** *if and only if the big model for $\langle \mathcal{S}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a \rangle$ and $\leadsto$ is a model of $\langle \mathcal{S}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a \rangle$ and $\leadsto$.*

Proof. Let $\mathcal{M} = \langle W, R \rangle$ be the big model of $\langle \mathcal{S}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a \rangle$ and $\leadsto$.

($\Rightarrow$): As $\mathcal{M}$ is a big model of $\langle \mathcal{S}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a \rangle$ and $\leadsto$, we have $\models^{\mathcal{M}} \mathcal{S} \wedge \mathcal{E}^a \wedge \mathcal{I}^a$. It remains to show that $\models^{\mathcal{M}} \mathcal{X}^a$. Let $\varphi_i \to \langle a \rangle \top \in \mathcal{X}^a$, and let $w \in W$ be such that $\models^{\mathcal{M}}_w \varphi_i$. Therefore, for all $\varphi_j \in \mathfrak{Fml}$ such that $\mathcal{S}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a \models_{\leadsto} \varphi_j \to [a]\bot$, we must have $\not\models^{\mathcal{M}}_w \varphi_j$, because $\mathcal{S}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a \models_{\leadsto} \neg(\varphi_i \wedge \varphi_j)$, and as $\langle \mathcal{S}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a \rangle$ and $\leadsto$ satisfy Postulate **PS**, $\mathcal{S} \models_{\mathsf{CPL}} \neg(\varphi_i \wedge \varphi_j)$, and hence $\models^{\mathcal{M}} \neg(\varphi_i \wedge \varphi_j)$. Then, by the construction of $\mathcal{M}$, there is some $w' \in W$ such that $\models^{\mathcal{M}}_{w'} \psi$, for all $\varphi \to [a]\psi$ such that $\mathcal{S}, \mathcal{E}^a, \mathcal{I}^a \models_{\leadsto} \varphi \to [a]\psi$ and $\models^{\mathcal{M}}_w \varphi$, and $w R_a w'$. Hence, $\models^{\mathcal{M}}_w \varphi_i \to \langle a \rangle \top$, and thus $\mathcal{M}$ is a model of $\langle \mathcal{S}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a \rangle$ and $\leadsto$.

($\Leftarrow$): Suppose $\langle \mathcal{S}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a \rangle$ and $\leadsto$ do not satisfy Postulate **PS**. Then there must be $\varphi \in \mathfrak{Fml}$ such that $\mathcal{S}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a \models_{\leadsto} \varphi$ and $\mathcal{S} \not\models_{\mathsf{CPL}} \varphi$. This means that there is a valuation *val* of $\mathcal{S}$ that falsifies φ. As $val \in W$ (because $\mathcal{M}$ contains all possible valuations of $\mathcal{S}$), $\mathcal{M}$ is not a model of $\langle \mathcal{S}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a \rangle$ and $\leadsto$. □

We shall give an algorithm to find a finite characterization of all[5] implicit static laws of a given action theory $\langle \mathcal{S}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a \rangle$. The idea is as follows: for each executability law $\varphi \to \langle a \rangle \top$ in the theory, construct from $\mathcal{E}^a$, $\mathcal{I}^a$ and $\leadsto$ a set of inexecutabilities $\{\varphi_1 \to [a]\bot, \ldots, \varphi_n \to [a]\bot\}$ that potentially conflict with $\varphi \to \langle a \rangle \top$. For each i, $1 \leqslant i \leqslant n$, if $\varphi \wedge \varphi_i$ is satisfiable w.r.t. $\mathcal{S}$, mark $\neg(\varphi \wedge \varphi_i)$ as an implicit static law. Incrementally repeat this procedure (adding all the implicit $\neg(\varphi \wedge \varphi_i)$ to $\mathcal{S}$) until no more implicit static law is obtained.

For an example of the execution of the algorithm, consider the action theory $\langle \mathcal{S}_1, \mathcal{E}_1^{tease}, \mathcal{X}_1^{tease}, \mathcal{I}_1^{tease} \rangle$ with $\leadsto$ as above. For the action *tease*, we have the executability $\langle tease \rangle \top$. Now, from $\mathcal{E}_1^{tease}$, $\mathcal{I}_1^{tease}$ and $\leadsto$ we try to build an inexecutability for *tease*. We take $[tease]walking$ and compute then all indirect effects of *tease* w.r.t. $\mathcal{S}_1$. From $walking \to alive$, we get that *alive* is an indirect effect of *tease*, giving us $[tease]alive$. But $\langle tease, alive \rangle \notin \leadsto$, which means the frame axiom $\neg alive \to [tease]\neg alive$ holds. Together with $[tease]alive$, this gives us the inexecutability $\neg alive \to [tease]\bot$. As $\mathcal{S}_1 \cup \{\top, \neg alive\}$ is satisfiable ($\top$ is the antecedent of the executability $\langle tease \rangle \top$), we get $\neg alive \to \bot$, i.e., the implicit static law *alive*. For this example no other inexecutability for *tease* can be derived, so the computation stops.

Before presenting the pseudo-code of the algorithm we need some definitions.

[4] The examples are from [34].

[5] Actually what the algorithm does is to find an interpolant of all implicit static laws of the theory.

Definition 17. Let $\varphi \in \mathfrak{Fml}$ and χ a clause. χ is an *implicate of* φ if and only if $\varphi \models_{\mathsf{CPL}} \chi$.

In our running example, $walking \vee alive$ and $\neg walking \vee alive$ are implicates of the set of formulas $\{walking \rightarrow alive, walking\}$.

Definition 18. Let $\varphi \in \mathfrak{Fml}$ and χ a clause. χ is a *prime implicate of* φ if and only if

- χ is an implicate of φ, and
- for every implicate χ' of φ, $\chi' \models_{\mathsf{CPL}} \chi$ implies $\chi \models_{\mathsf{CPL}} \chi'$.

The set of all prime implicates of a formula φ is denoted $PI(\varphi)$.

For example, the set of prime implicates of p_1 is just $\{p_1\}$, and that of $p_1 \wedge (\neg p_1 \vee p_2) \wedge (\neg p_1 \vee p_3 \vee p_4)$ is $\{p_1, p_2, p_3 \vee p_4\}$. In our shooting domain, *alive* is a prime implicate of $\{walking \rightarrow alive, walking\}$. For more on prime implicates, their properties and how to compute them see [35].

Definition 19. Let $\varphi, \psi \in \mathfrak{Fml}$. Then $NewCons(\psi, \varphi) = PI(\varphi \wedge \psi) \setminus PI(\varphi)$.

The function $NewCons(\psi, \varphi)$ computes the *new consequences* of φ w.r.t. ψ: the set of strongest clauses that follow from $\varphi \wedge \psi$, but do not follow from φ alone (cf. e.g. [36]). It is computed by subtracting the prime implicates of φ from those of $\varphi \wedge \psi$. For example, $NewCons((\neg p_1 \vee p_2) \wedge (\neg p_1 \vee p_3 \vee p_4), p_1) = \{p_2, p_3 \vee p_4\}$. And for our scenario, $NewCons(walking, walking \rightarrow alive) = \{alive, walking\}$.

The algorithm below improves the one in [37] by integrating a solution to the frame problem (via the dependence relation $\rightsquigarrow$). For convenience, we define $\mathcal{C}^a = \mathcal{E}^a \cup \mathcal{I}^a$ as the set of all formulas expressing the direct consequences of an action a, whether they are consistent or not.

Algorithm 1 *(Finding all implicit static laws induced by a).*
input: $\langle \mathcal{S}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a \rangle$ and $\rightsquigarrow$
output: $\mathcal{S}_{imp^*}$, the set of all implicit static laws of $\langle \mathcal{S}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a \rangle$

```
S_imp* := ∅
C^a := E^a ∪ I^a
repeat
   S_imp := ∅
   for all φ → ⟨a⟩⊤ ∈ X^a do
      for all Ĉ^a ⊆ Ĉ^a such that Ĉ^a ≠ ∅ do
         φ_Ĉa := ⋀{φ_i: φ_i → [a]ψ_i ∈ Ĉ^a}
         ψ_Ĉa := ⋀{ψ_i: φ_i → [a]ψ_i ∈ Ĉ^a}
         for all χ ∈ NewCons(φ_Ĉa, S) do
            if S ∪ S_imp* ∪ {φ, φ_Ĉa, ¬χ} ⊭_CPL ⊥ and ∀ℓ_i ∈ χ, a ↛ ℓ_i then
                  S_imp := S_imp ∪ {¬(φ ∧ φ_Ĉa ∧ ¬χ)}
   S_imp* := S_imp* ∪ S_imp
until S_imp = ∅
```

This is the key algorithm of the paper. In each step of the algorithm, $\mathcal{S} \cup \mathcal{S}_{imp^*}$ is the updated set of static laws (the original ones fed with the implicit laws caught up to that point). At the end, $\mathcal{S}_{imp^*}$ collects all the implicit static laws.

Theorem 20. *Algorithm* 1 *terminates.*

Proof. Let $\mathcal{C}^a = \mathcal{E}^a \cup \mathcal{I}^a$. First, the set of candidates to be an implicit static law that might be due to a and that are examined in the **repeat**-loop is

$$\left\{\neg(\varphi \wedge \varphi_{\hat{\mathcal{C}}^a} \wedge \neg\chi):\ \hat{\mathcal{C}}^a \subseteq \mathcal{C}^a, \varphi \rightarrow \langle a\rangle\top \in \mathcal{X}^a \text{ and } \chi \in \chi \in NewCons(\varphi_{\hat{\mathcal{C}}^a}, \mathcal{S})\right\}.$$

As $\mathcal{E}^a$, $\mathcal{I}^a$ and $\mathcal{X}^a$ are finite, this set is finite.

In each step, either the algorithm stops because $\mathcal{S}_{imp} = \emptyset$, or at least one of the candidates is put into $\mathcal{S}_{imp}$ in the outermost **for**-loop. (This one terminates, because $\mathcal{X}^a$, $\mathcal{C}^a$ and *NewCons*(.) are finite.) Such a candidate is not going to be put into $\mathcal{S}_{imp}$ in future steps of the algorithm, because once added to $\mathcal{S} \cup \mathcal{S}_{imp^*}$, it will be in the set of laws $\mathcal{S} \cup \mathcal{S}_{imp^*}$ of all subsequent executions of the outermost **for**-loop, falsifying its respective **if**-test for such a candidate. Hence the **repeat**-loop is bounded by the number of candidates, and therefore Algorithm 1 terminates. □

While terminating, our algorithm comes with considerable computational costs: first, the number of formulas $\varphi_{\hat{\mathcal{C}}^a}$ and $\psi_{\hat{\mathcal{C}}^a}$ is exponential in the size of $\mathcal{C}^a$, and second, the computation of $NewCons(\psi_{\hat{\mathcal{C}}^a}, \mathcal{S})$ might result in exponential growth. While we might expect $\mathcal{C}^a$ to be reasonably small in practice (because $\mathcal{E}^a$ and $\mathcal{I}^a$ are in general small), the size of $NewCons(\psi_{\hat{\mathcal{C}}^a}, \mathcal{S})$ is more difficult to control.

Example 21. For $\langle \mathcal{S}_1, \mathcal{E}_1^{tease}, \mathcal{X}_1^{tease}, \mathcal{I}_1^{tease} \rangle$, Algorithm 1 returns $\mathcal{S}_{imp^*} = \{alive\}$.

Theorem 22. *Let $\mathcal{S}_{imp^*}$ be the output of Algorithm 1 on input $\langle \mathcal{S}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a \rangle$ and $\leadsto$. Then $\langle \mathcal{S}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a \rangle$ with $\leadsto$ satisfies Postulate* **PS** *if and only if $\mathcal{S}_{imp^*} = \emptyset$.*

Proof. See Appendix A. □

Theorem 23. *Let $\mathcal{S}_{imp^*}$ be the output of Algorithm 1 on input $\langle \mathcal{S}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a \rangle$ and $\leadsto$. Then*

(1) $\langle \mathcal{S} \cup \mathcal{S}_{imp^*}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a \rangle$ *with $\leadsto$ satisfies* **PS** (*has no implicit static law*).
(2) $\mathcal{S}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a \models_{\leadsto} \bigwedge \mathcal{S}_{imp^*}$.

Proof. Item (1) is straightforward from the termination of Algorithm 1 and Theorem 22. Item (2) follows from the fact that by the **if**-test in Algorithm 1, the only formulas that are put in $\mathcal{S}_{imp^*}$ at each execution of the **repeat**-loop are exactly those that are implicit static laws of the current theory, and therefore of the original theory, too. □

Corollary 24. *For all $\varphi \in \mathfrak{Fml}$, $\mathcal{S}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a \models_{\leadsto} \varphi$ if and only if $\mathcal{S} \cup \mathcal{S}_{imp^*} \models_{\mathsf{CPL}} \varphi$.*

Proof. For the left-to-right direction, let $\mathcal{S}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a \models_{\leadsto} \varphi$, for given $\varphi \in \mathfrak{Fml}$. Then $\mathcal{S} \cup \mathcal{S}_{imp^*}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a \models_{\leadsto} \varphi$, by monotonicity. By Theorem 23(1), $\langle \mathcal{S} \cup \mathcal{S}_{imp^*}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a \rangle$ has no implicit static law, hence $\mathcal{S} \cup \mathcal{S}_{imp^*} \models_{\mathsf{CPL}} \varphi$.

The right-to-left direction is straightforward by Theorem 23(2). □

What shall we do once we have discovered an implicit static law?

The existence of implicit static laws may indicate too strong executability laws: in Example 21, we wrongly assumed that *tease* is always executable. Thus one way of 'repairing' our theory would be to consider the weaker executability $alive \rightarrow \langle tease \rangle \top$ instead of $\langle tease \rangle \top$ in $\mathcal{X}^{tease}$.

On the other hand, implicit static laws may also indicate that the inexecutability laws are too strong:

Example 25. Consider $\mathcal{S} = \emptyset$, $\mathcal{E}^{shoot} = \{loaded \rightarrow [shoot]\neg alive\}$, $\mathcal{X}^{shoot} = \{hasGun \rightarrow \langle shoot \rangle \top\}$ and $\mathcal{I}^{shoot} = \{[shoot]\bot\}$, with the dependence relation $\leadsto = \{\langle shoot, \neg alive \rangle, \langle shoot, \neg walking \rangle$. For this theory Algorithm 1 returns $\mathcal{S}_{imp^*} = \{\neg hasGun\}$.

In Example 25 we discovered that the agent never has a gun. The problem here can be overcome by weakening $[shoot]\bot$ in $\mathcal{I}^{shoot}$ with $\neg hasGun \rightarrow [shoot]\bot$.[6]

[6] Regarding Examples 21 and 25, one might argue that in practice such silly errors will never be made. Nevertheless, the examples here given are quite simplistic, and for applications of real interest, whose complexity will be much higher, we simply cannot rely on the designer's knowledge about all side effects the stated formulas can have.

We can go further on this reasoning and also argue that the problem may be due to a too strong set of effect laws, or even to too strong frame axioms (i.e., a too weak dependence relation). To witness, for Example 21, if we replace the law $[tease]walking$ by the weaker $alive \rightarrow [tease]walking$, the resulting action theory would satisfy Postulate **PS**. In the same way, stating the (unintuitive) dependence $tease \rightsquigarrow alive$ (which means the frame axiom $\neg alive \rightarrow [tease]\neg alive$ is no longer valid) guarantees satisfaction of **PS**. (Note, however, that this solution becomes intuitive when *alive* is replaced by *awake*.)

To finish, implicit static laws of course may also indicate that the static laws are too weak:

Example 26. Suppose a computer representation of the line of integers, in which we can be at a strictly positive number, *pos*, or at a negative one or zero, $\neg pos$. Let *maxInt* and *minInt*, respectively, be the largest and the smallest representable integer number. Action *goLeft* is the action of moving to the biggest integer strictly smaller than the one at which we are. Consider the following action theory for this scenario (at_i means we are at number i):

$$\mathcal{S} = \{at_i \rightarrow pos\colon\ 0 < i \leqslant maxInt\} \cup \{at_i \rightarrow \neg pos\colon\ minInt \leqslant i \leqslant 0\},$$
$$\mathcal{E} = \big\{at_{minInt} \rightarrow [goLeft]underflow\big\} \cup \big\{at_i \rightarrow [goLeft]at_{i-1}\colon\ i > minInt\big\},$$
$$\mathcal{X} = \big\{\langle goLeft\rangle\top\big\}, \qquad \mathcal{I} = \emptyset$$

with the dependence relation ($minInt \leqslant i \leqslant maxInt$):

$$\rightsquigarrow = \left\{ \begin{array}{l} \langle goLeft, at_i\rangle, \langle goLeft, pos\rangle, \\ \langle goLeft, \neg pos\rangle, \langle goLeft, underflow\rangle \end{array} \right\}.$$

Applying Algorithm 1 to this action theory gives us the implicit static law $\neg(at_1 \wedge at_2)$, i.e., we cannot be at numbers 1 and 2 at the same time.

To summarize, in order to satisfy Postulate **PS**, an action theory should contain a complete set of static laws or, alternatively, should not contain too strong action laws.

Remark 27. $\mathcal{S} \cup \mathcal{S}_{imp^*}$ in general is not intuitive.

Whereas in the latter example the implicit static laws should be added to $\mathcal{S}$, in the others the implicit static laws are unintuitive and due to an (in)executability law that is too strong and should be weakened. Of course, how intuitive the modified action theory will be depends mainly on the knowledge engineer's choice.

To sum it up, eliminating implicit static laws may require revision of $\mathcal{S}$, $\mathcal{E}^a$ or $\rightsquigarrow$, or completion of $\mathcal{X}^a$ and $\mathcal{I}^a$. Completing $\mathcal{I}^a$ is the topic we address in the next section.

5. No implicit inexecutability laws

Let $\mathcal{S}_2 = \mathcal{S}_1$, $\mathcal{E}_2 = \mathcal{E}_1$ and $\mathcal{X}_2 = \mathcal{I}_2 = \emptyset$, and let $\rightsquigarrow$ be that for $\langle \mathcal{S}_1, \mathcal{E}_1, \mathcal{X}_1, \mathcal{I}_1\rangle$. Note that $\langle \mathcal{S}_2, \mathcal{E}_2, \mathcal{X}_2, \mathcal{I}_2\rangle$ and $\rightsquigarrow$ satisfy Postulate **PS**. From $[tease]walking$ it follows with $\mathcal{S}_2$ that $[tease]alive$, i.e., in every situation, after teasing the turkey, it is alive: $\mathcal{S}_2, \mathcal{E}_2^{tease} \models_{\mathsf{PDL}} [tease]alive$. Now as $tease \not\rightsquigarrow alive$, the status of *alive* is not modified by *tease*, and we have $\mathcal{S}_2, \mathcal{E}_2^{tease} \models_{\rightsquigarrow} \neg alive \rightarrow [tease]\neg alive$. From the above, it follows

$$\mathcal{S}_2, \mathcal{E}_2^{tease}, \mathcal{X}_2^{tease}, \mathcal{I}_2^{tease} \models_{\rightsquigarrow} \neg alive \rightarrow [tease]\bot,$$

i.e., an inexecutability law stating that a dead turkey cannot be teased. But

$$\mathcal{S}_2, \mathcal{I}_2^{tease} \not\models_{\mathsf{PDL}} \neg alive \rightarrow [tease]\bot,$$

hence Postulate **PI** is violated. Here the formula $\neg alive \rightarrow [tease]\bot$ is an example of what we call an *implicit inexecutability law*.

In the literature, such laws are also known as *implicit qualifications* [25], and it has been often supposed, in a more or less tacit way, that it is a positive feature of frameworks to leave them implicit and provide mechanisms for inferring them [2,38,39]. The other way round, one might argue as well that implicit qualifications indicate that the domain has not been described in an adequate manner: the form of inexecutability laws is simpler than that of effect laws, and it might be reasonably expected that it is easier to exhaustively describe them.[7] Thus, all inexecutabilities of a given action should be explicitly stated, and this is what Postulate **PI** says.

How can we check whether **PI** is violated? We can conceive an algorithm to find implicit inexecutability laws of a given action a. The basic idea is as follows: for every combination of effect laws of the form $(\varphi_1 \wedge \cdots \wedge \varphi_n) \rightarrow [a](\psi_1 \wedge \cdots \wedge \psi_n)$, with each $\varphi_i \rightarrow [a]\psi_i \in \mathcal{E}^a$, if $\varphi_1 \wedge \cdots \wedge \varphi_n$ is consistent w.r.t. to $\mathcal{S}$, $\psi_1 \wedge \cdots \wedge \psi_n$ inconsistent w.r.t. $\mathcal{S}$, and $\mathcal{S}, \mathcal{I}^a \not\models_{\mathsf{PDL}} (\varphi_1 \wedge \cdots \wedge \varphi_n) \rightarrow [a]\bot$, then output $(\varphi_1 \wedge \cdots \wedge \varphi_n) \rightarrow [a]\bot$ as an implicit inexecutability law. Our algorithm basically does this, and moreover takes into account dependence information.

For an example of the execution of the algorithm, take $\langle \mathcal{S}_2, \mathcal{E}_2^{tease}, \mathcal{X}_2^{tease}, \mathcal{I}_2^{tease} \rangle$ with $\leadsto$ as given above. From $\mathcal{E}_2^{tease}$ we get the law $\top \rightarrow [tease]walking$, whose antecedent is consistent with $\mathcal{S}$. As long as $\models_{\leadsto} \neg alive \rightarrow [tease]\neg alive$ and $\mathcal{S} \cup \{walking\} \models_{\mathsf{CPL}} alive$, and because $\mathcal{S}, \mathcal{I}_2^{tease} \not\models_{\mathsf{PDL}} (\top \wedge \neg alive) \rightarrow [tease]\bot$, we caught an implicit inexecutability. As there is no other combination of effect laws for *tease*, we end the simulation here.

Below is the pseudo-code of the algorithm for that (the reason $\mathcal{X}^a$ is not used in the computation will be made clear in the sequel):

Algorithm 2 *(Finding implicit inexecutability laws for a).*
input: $\langle \mathcal{S}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a \rangle$ and $\leadsto$
output: $\mathcal{I}^a_{imp}$, the set of implicit inexecutability laws for a
  $\mathcal{I}^a_{imp} := \emptyset$
  for all $\hat{\mathcal{E}}^a \subseteq \mathcal{E}^a$ **do**
    $\varphi_{\hat{\mathcal{E}}^a} := \bigwedge\{\varphi_i\colon \varphi_i \rightarrow [a]\psi_i \in \hat{\mathcal{E}}^a\}$
    $\psi_{\hat{\mathcal{E}}^a} := \bigwedge\{\psi_i\colon \varphi_i \rightarrow [a]\psi_i \in \hat{\mathcal{E}}^a\}$
    for all $\chi \in NewCons(\psi_{\hat{\mathcal{E}}^a}, \mathcal{S})$ **do**
      if $\forall \ell_i \in \chi, a \not\leadsto \ell_i$ **and** $\mathcal{S}, \mathcal{I}^a \not\models_{\mathsf{PDL}} (\varphi_{\hat{\mathcal{E}}^a} \wedge \neg\chi) \rightarrow [a]\bot$ **then**
        $\mathcal{I}^a_{imp} := \mathcal{I}^a_{imp} \cup \{(\varphi_{\hat{\mathcal{E}}^a} \wedge \neg\chi) \rightarrow [a]\bot\}$

Theorem 28. *Algorithm* 2 *terminates.*

Proof. Straightforward, as we have assumed $\mathcal{S}, \mathcal{E}, \mathcal{I}$ and $\leadsto$ finite, and *NewCons*(.) is finite (because $\mathcal{S}$ and $\psi_{\hat{\mathcal{E}}^a}$ are finite). □

Example 29. Consider $\mathcal{S}_2, \mathcal{E}_2^{tease}, \mathcal{X}_2^{tease}, \mathcal{I}_2^{tease}$ and $\leadsto$ as given above. Then Algorithm 2 returns $\mathcal{I}^{tease}_{imp} = \{\neg alive \rightarrow [tease]\bot\}$.

Nevertheless, applying Algorithm 2 is not enough to guarantee Postulate **PI**, as illustrated by the following example:

Example 30 *(Incompleteness of Algorithm 2 without **PS**).* Let $\mathcal{S} = \emptyset$, $\mathcal{E}^a = \{p_1 \rightarrow [a]p_2\}$, $\mathcal{X}^a = \{\langle a\rangle\top\}$, $\mathcal{I}^a = \{p_2 \rightarrow [a]\bot\}$, and $\leadsto = \emptyset$. Then we have $\mathcal{S}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a \models_{\leadsto} p_1 \rightarrow [a]\bot$, but running Algorithm 2 on $\langle \mathcal{S}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a \rangle$ we get $\mathcal{S}, \mathcal{I}^a_{imp} \not\models_{\mathsf{PDL}} p_1 \rightarrow [a]\bot$.

Example 30 shows that the presence of implicit static laws (induced by executabilities) implies the existence of implicit inexecutabilities that are not caught by Algorithm 2. One way of getting rid of this is requiring $\langle \mathcal{S}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a \rangle$ and $\leadsto$ to satisfy Postulate **PS** prior to running Algorithm 2. This gives us the following result:

Theorem 31. *Let $\mathcal{I}^a_{imp}$ be the output of Algorithm* 2 *on input $\langle \mathcal{S}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a \rangle$ and $\leadsto$. If $\langle \mathcal{S}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a \rangle$ with $\leadsto$ satisfies Postulate* **PS**, *then it satisfies Postulate* **PI** *if and only if $\mathcal{I}^a_{imp} = \emptyset$.*

[7] Note that this concerns the necessary conditions for executability, and thus it is not related to the qualification problem, which basically says that it is difficult to state all the sufficient conditions for executability.

Proof. See Appendix B. □

With Algorithm 2, not only do we decide whether Postulate **PI** is satisfied, but we also get information on how to "repair" the action theory. The set of implicit inexecutabilities so obtained provides logical and metalogical information concerning the correction that must be carried out: in the first case, elements of $\mathcal{I}^a_{imp}$ can be added to $\mathcal{I}^a$; in the second one, $\mathcal{I}^a_{imp}$ helps in properly changing $\mathcal{E}^a$ or $\leadsto$. For instance, to correct the action theory of our example, the knowledge engineer would have the following options:

(1) Add the qualification $\neg alive \to [tease]\bot$ to $\mathcal{I}^{tease}_2$; or
(2) Add the (unintuitive) dependence $\langle tease, alive\rangle$ to $\leadsto$; or
(3) Weaken the effect law $[tease]walking$ to $alive \to [tease]walking$ in $\mathcal{E}^{tease}_2$.

It is easy to see that whatever she opts for, the resulting action theory for *tease* will satisfy Postulate **PI** (while still satisfying **PS**).

Example 32 *(Drinking coffee [12]).* Suppose a situation in which we reason about the effects of drinking a cup of coffee:

$$\mathcal{S}=\emptyset,\qquad \mathcal{E}^{drink}=\left\{\begin{array}{l} sugar\to[drink]happy,\\ salt\to[drink]\neg happy\end{array}\right\},\qquad \mathcal{X}^{drink}=\mathcal{I}^{drink}=\emptyset$$

and the dependence relation

$$\leadsto=\{\langle drink, happy\rangle, \langle drink, \neg happy\rangle\}.$$

Observe that $\langle\mathcal{S},\mathcal{E}^{drink},\mathcal{X}^{drink},\mathcal{I}^{drink}\rangle$ with $\leadsto$ satisfies **PS**. Then, running Algorithm 2 on this action theory will give us $\mathcal{I}^{drink}_{imp}=\{(sugar\wedge salt)\to[drink]\bot\}$.

Remark 33. $\mathcal{I}^a\cup\mathcal{I}^a_{imp}$ is not always intuitive.

Whereas in Example 29 we have got an inexecutability that could be safely added to $\mathcal{I}^{tease}_2$, in Example 32 we got an inexecutability that is unintuitive (just the presence of sugar and salt in the coffee precludes drinking it). In that case, revision of other parts of the theory should be considered in order to make it intuitive. Anyway, the problem pointed out in the depicted scenario just illustrates that intuition is beyond syntax. The scope of this work relies on the syntactical level. Only the knowledge engineer can judge about how intuitive a formula is.

In what follows we revisit our postulates in order to strengthen them to the case where more than one action is under concern and thus get results that can be applied to whole action theories.

6. Generalizing the postulates

We have seen the importance that satisfaction of Postulates **PC**, **PS** and **PI** may have in describing the action theory of a particular action a. However, in applications of real interest more than one action is involved, and thus a natural question that could be raised is "can we have similar metatheoretical results for multiple action theories"?

In this section we generalize our set of postulates to action theories as a whole, i.e., considering all actions of a domain, and prove some interesting results that follow from that. As we are going to see, some of these results are straightforward, while others must rely on some additional assumptions in order to hold.

A generalization of Postulate **PC** is quite easy and has no need for justification:

PC* (Logical consistency): $\mathcal{S},\mathcal{E},\mathcal{X},\mathcal{I}\not\models_{\leadsto}\bot$.

The whole action theory should be logically consistent.

Generalizing Postulate **PS** will give us the following:

PS* (No implicit static laws):

if $\mathcal{S},\mathcal{E},\mathcal{X},\mathcal{I}\models_{\leadsto}\varphi$, then $\mathcal{S}\models_{\mathsf{CPL}}\varphi$.

If a classical formula can be inferred from the whole action theory, then it should be inferable from the set of static laws alone. We have the following results:

Theorem 34. *$\langle \mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \rangle$ and $\leadsto$ satisfy* $\mathbf{PS}^*$ *if and only if $\langle \mathcal{S}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a \rangle$ and $\leadsto$ satisfies* **PS** *for all $a \in \mathfrak{Act}$.*

Proof. (⇒): Straightforward: Suppose that for some $a \in \mathfrak{Act}$ $\langle \mathcal{S}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a \rangle$ does not satisfy **PS**. Then there is $\varphi \in \mathfrak{Fml}$ such that $\mathcal{S}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a \models_{\leadsto} \varphi$ and $\mathcal{S} \not\models_{\mathsf{CPL}} \varphi$. Of course $\mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \models_{\leadsto} \varphi$, by monotonicity, but still $\mathcal{S} \not\models_{\mathsf{CPL}} \varphi$. Hence $\langle \mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \rangle$ does not satisfy $\mathbf{PS}^*$.

(⇐): Suppose $\langle \mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \rangle$ with $\leadsto$ does not satisfy $\mathbf{PS}^*$. Then there is $\varphi \in \mathfrak{Fml}$ such that $\mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \models_{\leadsto} \varphi$ and $\mathcal{S} \not\models_{\mathsf{CPL}} \varphi$. φ is equivalent to $\varphi_1 \wedge \cdots \wedge \varphi_n$, with $\varphi_1, \ldots, \varphi_n \in \mathfrak{Fml}$ and such that there is at least one φ_i such that $\mathcal{S} \not\models_{\mathsf{CPL}} \varphi_i$ (otherwise $\mathcal{S} \models_{\mathsf{CPL}} \varphi$). Because the logic is independently axiomatized, there must be some $a \in \mathfrak{Act}$ such that $\mathcal{S}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a \models_{\leadsto} \varphi_i$. From this and $\mathcal{S} \not\models_{\mathsf{CPL}} \varphi_i$ it follows that $\langle \mathcal{S}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a \rangle$ and with $\leadsto$ do not satisfy **PS**. □

Corollary 35. *$\langle \mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \rangle$ and $\leadsto$ satisfy Postulate* $\mathbf{PS}^*$ *if and only if the big model for $\langle \mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \rangle$ and $\leadsto$ is a model of $\langle \mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \rangle$.*

Proof. The proof follows from Theorems 16 and 34. □

Theorem 36. *If $\langle \mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \rangle$ with $\leadsto$ satisfies* $\mathbf{PS}^*$*, then $\langle \mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \rangle$ with $\leadsto$ satisfies* $\mathbf{PC}^*$ *if and only if $\langle \mathcal{S}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a \rangle$ and $\leadsto$ satisfy* **PC** *for all $a \in \mathfrak{Act}$.*

Proof. Let $\langle \mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \rangle$ and $\leadsto$ satisfy $\mathbf{PS}^*$.

(⇒): Suppose that $\langle \mathcal{S}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a \rangle$ with $\leadsto$ does not satisfy **PC**, for some $a \in \mathfrak{Act}$. Because $\langle \mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \rangle$ and $\leadsto$ satisfy $\mathbf{PS}^*$, $\langle \mathcal{S}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a \rangle$ and $\leadsto$ satisfy Postulate **PS** (Theorem 34), and then $\mathcal{S} \models_{\mathsf{CPL}} \bot$. From this it follows that $\langle \mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \rangle$ does not satisfy Postulate $\mathbf{PC}^*$.

(⇐): Suppose $\langle \mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \rangle$ and $\leadsto$ do not satisfy $\mathbf{PC}^*$. Then $\mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \models_{\leadsto} \bot$. Because $\langle \mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \rangle$ with $\leadsto$ satisfies Postulate $\mathbf{PS}^*$, $\mathcal{S} \models_{\mathsf{CPL}} \bot$. Since $\mathfrak{Act} \neq \emptyset$, there is some $a \in \mathfrak{Act}$ such that $\mathcal{S}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a \models_{\leadsto} \bot$. □

A more general form of Postulate **PI** can also be stated:

PI* (No implicit inexecutability laws):

$$\text{if } \mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \models_{\leadsto} \varphi \to [a]\bot, \text{ then } \mathcal{S}, \mathcal{I} \models_{\mathsf{PDL}} \varphi \to [a]\bot.$$

If an inexecutability law can be inferred from the whole action theory, then it should be inferable in PDL from the static and inexecutability laws alone.

Note that having that $\langle \mathcal{S}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a \rangle$ with $\leadsto$ satisfies **PI** for all $a \in \mathfrak{Act}$ is not enough to $\langle \mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \rangle$ with $\leadsto$ satisfy $\mathbf{PI}^*$ if there are implicit static laws. To witness, let $\mathcal{S} = \mathcal{E}^{a_1} = \emptyset$, and $\mathcal{X}^{a_1} = \{\langle a_1 \rangle \top\}$, $\mathcal{I}^{a_1} = \{\varphi \to [a_1]\bot\}$. Let also $\mathcal{E}^{a_2} = \mathcal{X}^{a_2} = \mathcal{I}^{a_2} = \emptyset$. Observe that both $\langle \mathcal{S}, \mathcal{E}^{a_1}, \mathcal{X}^{a_1}, \mathcal{I}^{a_1} \rangle$ and $\langle \mathcal{S}, \mathcal{E}^{a_2}, \mathcal{X}^{a_2}, \mathcal{I}^{a_2} \rangle$ with $\leadsto$ satisfy **PI**, but $\mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \models_{\leadsto} \varphi \to [a_2]\bot$ and $\mathcal{S}, \mathcal{I} \not\models_{\mathsf{PDL}} \varphi \to [a_2]\bot$.

Nevertheless, under $\mathbf{PS}^*$ the result follows:

Theorem 37. *Let $\langle \mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \rangle$ with $\leadsto$ satisfy* $\mathbf{PS}^*$. *$\langle \mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \rangle$ with $\leadsto$ satisfies* $\mathbf{PI}^*$ *if and only if $\langle \mathcal{S}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a \rangle$ and $\leadsto$ satisfy* **PI** *for all $a \in \mathfrak{Act}$.*

Proof. See Appendix C. □

In the next section we make a step toward an attempt of amending our modularity criteria by investigating possible extensions of our set of postulates.

7. Can we ask for more?

Can we augment our set of postulates to take into account other modules of action theories, or even other metatheoretical issues in reasoning about actions? That is the topic we discuss in what follows.

7.1. Postulates about action effects

It seems to be in line with our postulates to require action theories not to allow for the deduction of new effect laws: if an effect law can be inferred from an action theory (and no inexecutability for the same action in the same context can be derived), then it should be inferable from the set of static and effect laws alone. This means we should have:

PE (No implicit effect laws):

$$\text{if } \mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \models_{\leadsto} \varphi \rightarrow [a]\psi \text{ and } \mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \not\models_{\leadsto} \varphi \rightarrow [a]\bot, \quad \text{then } \mathcal{S}, \mathcal{E} \models_{\leadsto} \varphi \rightarrow [a]\psi.$$

But consider the following intuitively correct action theory:

$$\mathcal{S}_3 = \emptyset, \qquad \mathcal{E}_3 = \left\{ \begin{array}{l} \mathit{loaded} \rightarrow [\mathit{shoot}]\neg \mathit{alive}, \\ (\neg \mathit{loaded} \wedge \mathit{alive}) \rightarrow [\mathit{shoot}]\mathit{alive} \end{array} \right\},$$
$$\mathcal{X}_3 = \{\mathit{hasGun} \rightarrow \langle \mathit{shoot} \rangle \top\}, \qquad \mathcal{I}_3 = \{\neg \mathit{hasGun} \rightarrow [\mathit{shoot}]\bot\}$$

together with the dependence $\mathit{shoot} \leadsto \neg \mathit{alive}$. It satisfies Postulates **PS*** and **PI***, but does not satisfy **PE**. Indeed:

$$\mathcal{S}_3, \mathcal{E}_3, \mathcal{X}_3, \mathcal{I}_3 \models_{\leadsto} \neg \mathit{hasGun} \vee \mathit{loaded} \rightarrow [\mathit{shoot}]\neg \mathit{alive}$$

and

$$\mathcal{S}_3, \mathcal{E}_3, \mathcal{X}_3, \mathcal{I}_3 \not\models_{\leadsto} \neg \mathit{hasGun} \vee \mathit{loaded} \rightarrow [\mathit{shoot}]\bot,$$

but

$$\mathcal{S}_3, \mathcal{E}_3 \not\models_{\leadsto} \neg \mathit{hasGun} \vee \mathit{loaded} \rightarrow [\mathit{shoot}]\neg \mathit{alive}.$$

So, Postulate **PE** would not help us to deliver the goods.

Another possibility of improving our modularity criteria could be:

P⊥ (No unattainable effects):

$$\text{if } \varphi \rightarrow [a]\psi \in \mathcal{E}, \text{ then } \mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \not\models_{\leadsto} \varphi \rightarrow [a]\bot.$$

This expresses that if we have explicitly stated an effect law for a in some context, then there should be no inexecutability law for the same action in the same context. It is straightforward to design an algorithm which checks whether this postulate is satisfied. We do not investigate this further here, but just observe that the slightly stronger version below leads to unintuitive consequences:

P⊥' (No unattainable effects—strong version):

$$\text{if } \mathcal{S}, \mathcal{E} \models_{\leadsto} \varphi \rightarrow [a]\psi, \text{ then } \mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \not\models_{\leadsto} \varphi \rightarrow [a]\bot.$$

Indeed, for the above action theory we have

$$\mathcal{E}_3 \models_{\leadsto} (\neg \mathit{hasGun} \wedge \mathit{loaded}) \rightarrow [\mathit{shoot}]\neg \mathit{alive},$$

but

$$\mathcal{S}_3, \mathcal{E}_3, \mathcal{X}_3, \mathcal{I}_3 \models_{\leadsto} (\neg \mathit{hasGun} \wedge \mathit{loaded}) \rightarrow [\mathit{shoot}]\bot.$$

This is certainly too strong. Our example also illustrates that it is sometimes natural to have 'redundancies' or 'overlaps' between $\mathcal{E}$ and $\mathcal{I}$. Indeed, as we have pointed out, inexecutability laws are a particular kind of effect laws, and the distinction here made is conventional. The decision of considering them as strictly different entities or not depends mainly on the context. At a representational level we prefer to keep them separated, while in Algorithm 1 we have mixed them together in order to compute the consequences of an action.

In what follows we address the problem of completing the set of executability laws of an action theory.

7.2. Maximizing executability

As we have seen, implicit static laws only show up when there are executability laws. So, a question that naturally raises is "which executability laws can be consistently added to a given action theory?".

A hypothesis usually made in the literature is that of maximization of executabilities: in the absence of a proof that an action is inexecutable in a given context, assume its executability for that context. Such a hypothesis is captured by the following postulate that we investigate in this section:

PX$^+$ (Maximal executability laws):

if $\mathcal{S}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a \not\models_{\leadsto} \varphi \rightarrow [a]\bot$, then $\mathcal{S}, \mathcal{X}^a \models_{\mathsf{PDL}} \varphi \rightarrow \langle a \rangle \top$.

Such a postulate expresses that if in context φ no inexecutability for a can be inferred, then the respective executability should follow in PDL from the executability and static laws.

Postulate **PX$^+$** generally holds in non-monotonic frameworks, and can be enforced in monotonic approaches such as ours by maximizing $\mathcal{X}^a$. We nevertheless would like to point out that maximizing executability is not always intuitive. To witness, suppose we know that if we have the ignition key, the tank is full, ..., and the battery tension is beyond 10 V, then the car (necessarily) will start. Suppose we also know that if the tension is below 8 V, then the car will not start. What should we conclude in situations where we know that the tension is 9 V? Maximizing executabilities makes us infer that it will start, but such reasoning is not what we want if we would like to be sure that all possible executions lead to the goal.

8. Exploiting modularity

In this section we present other properties related to consistency and modularity of action theories, emphasizing the main results that we obtain when Postulate **PS*** is satisfied.

Theorem 38. *If $\langle \mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \rangle$ with $\leadsto$ satisfies* **PS****, then $\mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \models_{\leadsto} \bot$ if and only if $\mathcal{S} \models_{\mathsf{CPL}} \bot$.*

This theorem says that if there are no implicit static laws, then consistency of an action theory can be checked by just checking consistency of $\mathcal{S}$.

Theorem 39. *If $\langle \mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \rangle$ and $\leadsto$ satisfy Postulate* **PS****, then $\mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \models_{\leadsto} \varphi \rightarrow [a]\psi$ if and only if $\mathcal{S}, \mathcal{E}^a, \mathcal{I}^a \models_{\leadsto} \varphi \rightarrow [a]\psi$.*

Proof. See Appendix D. □

This means that under **PS*** we have modularity inside $\mathcal{E}$, too: when deducing the effects of a we need not consider the action laws for other actions. Versions for executability and inexecutability can be stated as well:

Theorem 40. *If $\langle \mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \rangle$ and $\leadsto$ satisfy Postulate* **PS****, then $\mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \models_{\leadsto} \varphi \rightarrow \langle a \rangle \top$ if and only if $\mathcal{S}, \mathcal{X}^a \models_{\leadsto} \varphi \rightarrow \langle a \rangle \top$.*

Proof. See Appendix E. □

Corollary 41. **PX** *is a consequence of* **PS**.

Proof. Straightforward. □

Theorem 42. *If $\langle \mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \rangle$ and $\leadsto$ satisfy Postulates* **PS*** *and* **PI****, then $\mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \models_{\leadsto} \varphi \rightarrow [a]\bot$ if and only if $\mathcal{S}, \mathcal{I}^a \models_{\mathsf{PDL}} \varphi \rightarrow [a]\bot$.*

Proof. (⇒): If $\mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \models_{\leadsto} \varphi \rightarrow [a]\bot$, then from **PS*** and Theorem 39 we have $\mathcal{S}, \mathcal{E}^a, \mathcal{I}^a \models_{\leadsto} \varphi \rightarrow [a]\bot$. From this and **PI*** we get $\mathcal{S}, \mathcal{I}^a \models_{\mathsf{PDL}} \varphi \rightarrow [a]\bot$.

(⇐): Suppose $\mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \not\models_{\leadsto} \varphi \rightarrow [a]\bot$. Then there is a $\leadsto$-model $\mathscr{M}$ such that $\models^{\mathscr{M}} \mathcal{S} \wedge \mathcal{E} \wedge \mathcal{X} \wedge \mathcal{I}$ and $\not\models^{\mathscr{M}} \varphi \rightarrow [a]\bot$. Then, given a, we have $\models^{\mathscr{M}} \mathcal{S} \wedge \mathcal{E}^a \wedge \mathcal{X}^a \wedge \mathcal{I}^a$, and then $\models^{\mathscr{M}} \mathcal{S} \wedge \mathcal{I}^a$. Moreover, by definition, $\mathscr{M}$ is a PDL-model. Hence $\mathcal{S}, \mathcal{I}^a \not\models_{\mathsf{PDL}} \varphi \rightarrow [a]\bot$. □

In Theorems 40 and 42, modularity guarantees moreover that no dependence is needed to derive, respectively, executabilities and inexecutabilities.

Let $\mathcal{E}^{a_1,\dots,a_n} = \bigcup_{1 \leqslant i \leqslant n} \mathcal{E}^{a_i}$, $\mathcal{X}^{a_1,\dots,a_n} = \bigcup_{1 \leqslant i \leqslant n} \mathcal{X}^{a_i}$, and $\mathcal{I}^{a_1,\dots,a_n} = \bigcup_{1 \leqslant i \leqslant n} \mathcal{I}^{a_i}$. Under Postulate **PS***, deduction of an effect of a sequence of actions $a_1; \dots; a_n$ (prediction) needs neither the effect and inexecutability laws for actions other than $a_1, \dots, a_n$, nor the executability laws of the domain:

Theorem 43. *If $\langle \mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \rangle$ and $\leadsto$ satisfy Postulate* **PS****, then $\mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \models_{\leadsto} \varphi \rightarrow [a_1; \dots; a_n]\psi$ if and only if $\mathcal{S}, \mathcal{E}^{a_1,\dots,a_n}, \mathcal{I}^{a_1,\dots,a_n} \models_{\leadsto} \varphi \rightarrow [a_1; \dots; a_n]\psi$.*

Proof. See Appendix F. □

The same result holds for testing inexecutability of a sequence of actions:

Corollary 44. *If $\langle \mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \rangle$ and $\leadsto$ satisfy Postulate* **PS****, then $\mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \models_{\leadsto} \varphi \rightarrow [a_1; \dots; a_n]\bot$ if and only if $\mathcal{S}, \mathcal{E}^{a_1,\dots,a_n}, \mathcal{I}^{a_1,\dots,a_n} \models_{\leadsto} \varphi \rightarrow [a_1; \dots; a_n]\bot$.*

Proof. Straightforward, as a special case of Theorem 43. □

The next theorem shows that our notion of modularity is also fruitful in plan validation:

Theorem 45. *Let $\langle \mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \rangle$ and $\leadsto$ satisfy Postulate* **PS****. Then we have $\mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \models_{\leadsto} \varphi \rightarrow \langle a_1; \dots; a_n \rangle \psi$ if and only if $\mathcal{S}, \mathcal{E}^{a_1,\dots,a_n}, \mathcal{X}^{a_1,\dots,a_n}, \mathcal{I}^{a_1,\dots,a_n} \models_{\leadsto} \varphi \rightarrow \langle a_1; \dots; a_n \rangle \psi$.*

Proof. See Appendix G.

And as a consequence, we also optimize testing executability of a plan:

Corollary 46. *Let $\langle \mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \rangle$ and $\leadsto$ satisfy Postulate* **PS****. Then we have $\mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \models_{\leadsto} \varphi \rightarrow \langle a_1; \dots; a_n \rangle \top$ if and only if $\mathcal{S}, \mathcal{E}^{a_1,\dots,a_n}, \mathcal{X}^{a_1,\dots,a_n}, \mathcal{I}^{a_1,\dots,a_n} \models_{\leadsto} \varphi \rightarrow \langle a_1; \dots; a_n \rangle \top$.*

Proof. Straightforward, as a special case of Theorem 45.

Theorems 43 and 45 together with Corollaries 44 and 46 suggest that we can simulate modularization by sub-domains [40]: If $\langle \{a_1, \dots, a_n\}, \mathfrak{Prop}' \rangle$ is a sub-domain for some $\mathfrak{Prop}' \subseteq \mathfrak{Prop}$, then $\langle \mathcal{S}, \mathcal{E}^{a_1,\dots,a_n}, \mathcal{X}^{a_1,\dots,a_n}, \mathcal{I}^{a_1,\dots,a_n} \rangle$ with $\leadsto$ corresponds to the module for $\langle \{a_1, \dots, a_n\}, \mathfrak{Prop}' \rangle$ in Lifschitz and Ren's sense (see the next section).

9. Related work

Pirri and Reiter have investigated the metatheory of the Situation Calculus [41]. In a spirit similar to ours, they use executability laws and effect laws. Contrarily to us, their executability laws are equivalences and are thus at the same time inexecutability laws. As they restrict themselves to domains without ramifications, there are no static laws, i.e., $\mathcal{S} = \emptyset$. For this setting they give a syntactical condition on effect laws guaranteeing that they do not interact with the executability laws in the sense that they do not entail implicit static laws. Basically, the condition says that when there are effect laws $\varphi_1 \rightarrow [a]\psi$ and $\varphi_2 \rightarrow [a]\neg\psi$, then φ_1 and φ_2 are inconsistent (which essentially amounts to having in their theories a kind of "implicit static law schema" of the form $\neg(\varphi_1 \wedge \varphi_2)$).

This then allows them to show that such theories are always consistent. Moreover they thus simplify the entailment problem for this calculus, and show for several problems such as consistency or regression that only some of the modules of an action theory are necessary.

Amir [42] focuses on design and maintainability of action descriptions applying many of the concepts of the object-oriented paradigm in the Situation Calculus. In that work, guidelines for a partitioned representation of a given theory are presented, with which the inference task can also be optimized, as it is restricted to the part of the theory that is really relevant to a given query. This is observed specially when different agents are involved: the design of an agent's theory can be done with no regard to others', and after the integration of multiple agents, queries about an agent's beliefs do not take into account the belief state of other agents.

In the referred work, executabilities are as in [41] and the same condition on effect laws is assumed, which syntactically precludes the existence of implicit static laws. The frame problem is solved using Reiter's solution [32] and then is also restricted to domains without static laws. Ramifications are dealt with by compiling them away *à la* Reiter and Lin [43] based on the method given in [44], which takes into account only some restricted state constraints.

In spite of using many of the object-oriented paradigm tools and techniques, no mention is made to the concepts of cohesion and coupling [45,46], which are closely related to modularity [12]. In the approach presented in [42], even if modules are highly cohesive, they are not necessarily lowly coupled, due to the dependence between objects in the reasoning phase. We do not investigate this further here, but conjecture that this could be done there by, during the reasoning process defined for that approach, avoiding passing to a module a formula of a type different from those it contains.

The present work generalizes and extends Pirri and Reiter's result to the case where $\mathcal{S} \neq \emptyset$ and both these works where the syntactical restriction on effect laws is not made. This gives us more expressive power, as we can reason about inexecutabilities, and a better modularity in the sense that we do not combine formulas that are conceptually different (viz. executabilities and inexecutabilities). It also constitutes a better approach for domains with ramifications as we do not impose any restriction on the domain laws we can deal with.

Zhang et al. [47] have also proposed an assessment of what a good action theory should look like. They develop the ideas in the framework of EPDL [6], an extended version of PDL which allows for propositions as modalities to represent a causal connection between literals. We do not present the details of that, but concentrate on the main meta-theoretical results.

Zhang et al. propose a normal form for describing action theories,[8] and investigate three levels of consistency. Roughly speaking, a set of laws $\mathcal{T}$ is *uniformly consistent* if it is globally consistent (i.e., $\mathcal{T} \not\models_{\mathsf{EPDL}} \bot$); a formula Φ is $\mathcal{T}$*-consistent* if $\mathcal{T} \not\models_{\mathsf{EPDL}} \neg\Phi$, for $\mathcal{T}$ a uniformly consistent theory; $\mathcal{T}$ is *universally consistent* if (in our terms) every logically possible world is accessible.

Furthermore, two assumptions are made to preclude the existence of implicit qualifications. Satisfaction of such assumptions means the theory under consideration is *safe*, i.e., it is uniformly consistent. Such a normal form justifies the two assumptions made and on which their notion of good theories relies.

Given this, they propose algorithms to test the different versions of consistency for a theory $\mathcal{T}$ that is in normal form. This test essentially amounts to checking whether $\mathcal{T}$ is *safe*, i.e., whether $\mathcal{T} \models_{\mathsf{EPDL}} \langle a \rangle \top$, for every action a. Success of this check should mean that the theory under analysis satisfies the consistency requirements.

Although they are concerned with the same kind of problems that have been discussed in this work, they take an overall view of the subject, in the sense that all problems are dealt with together. This means that in their approach no special attention (in our sense) is given to the different components of the theory, and then every time something is wrong with it this is taken as a global problem inherent to the theory as a whole. Whereas such a "systemic" view of action theories is not necessarily a drawback (we have just seen the strong interaction that exists between the different sets of laws composing an action theory), being modular in our sense allows us to better identify the "problematic" laws and take care of them. Moreover, the advantage of allowing to find the set of laws which must be modified in order to achieve the desired consistency is made evident by the algorithms we have proposed (while their results only allow to decide whether a given theory satisfies some consistency requirement).

[8] But not as expressive as one might think: For instance, in modeling the nondeterministic action of dropping a coin on a chessboard, we are not able to state $[drop](black \vee white)$. Instead, we should write something like $[drop_{black}]black$, $[drop_{white}]white$, $[drop_{black,white}]black$ and $[drop_{black,white}]white$, where $drop_{black}$ is the action of dropping the coin on a black square (analogously for the others) and $drop = drop_{black} \cup drop_{white} \cup drop_{black,white}$, with "$\cup$" the nondeterministic composition of actions.

Lang et al. [48] address consistency of action theories in a version of the causal laws approach [3], focusing on the computational aspects.

To solve the frame problem, they suppose an abstract notion of completion. Given a theory $\mathcal{T}^a$ containing logical information about a's direct effects as well as the indirect effects that may follow (expressed in the form of causal laws), the completion of $\mathcal{T}^a$, roughly speaking, is the original theory $\mathcal{T}^a$ amended of some axioms stating the persistence of all non-affected (directly nor indirectly) literals. (Note that such a notion of completion is close to the underlying semantics of the dependence relation used throughout the present work, which essentially amounts to the explanation closure assumption [26].)

Their EXECUTABILITY problem is to check whether action a is executable in all possible initial states (Zhang et al.'s safety property). This amounts to testing whether every possible state w has a successor w' reachable by a such that w and w' both satisfy the completion of $\mathcal{T}^a$. For the Walking Turkey Scenario, the formalization of action *tease* with causal laws is given by:

$$\mathcal{T}^{tease} = \left\{ \begin{array}{l} \top \overset{tease}{\Rightarrow} walking, \\ \neg alive \Rightarrow \neg walking \end{array} \right\}$$

where the first formula is a conditional effect law for *tease*, and the latter a causal law in McCain and Turner's sense. We will not dive in the technical details, and just note that the executability check will return "no" for this example as *tease* cannot be executed in a state satisfying $\neg alive$.

In the mentioned work, the authors are more concerned with the complexity analysis of the problem of doing such a consistency test and no algorithm for performing it is given, however. Despite the fact their motivation is the same as ours, again what is presented is a kind of "yes-no tool" which can help in doing a meta-theoretical analysis of a given action theory, and many of the comments concerning Zhang et al.'s approach could be repeated here.

Another criticism that could be made about both these approaches concerns the assumption of full executability they rely on. We find it too strong to require all actions to be always executable (cf. Section 7), and to reject as bad an action theory admitting situations where some action cannot be executed at all. As an example, consider a very simple action theory $\langle \mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \rangle$, where $\mathcal{S} = \{walking \rightarrow alive\}$, $\mathcal{E} = \{[tease]walking\}$, $\mathcal{X} = \{\langle tease \rangle \top\}$, and $\mathcal{I} = \emptyset$, with a dependence relation given by $\leadsto = \{\langle tease, walking \rangle\}$. Observe that, with our approach, it suffices to derive the implicit inexecutability law $\neg alive \rightarrow [tease]\bot$, change $\mathcal{I}$, and the system will properly run in situations where $\neg alive$ is the case.

On the other hand, if we consider the equivalent representation of such an action theory in the approach of Lang et al., after computing the completion of $\mathcal{T}^{tease}$, if we test its executability, we will get the answer "no", the reason being that *tease* is not executable in the possible state where $\neg alive$ holds. Such an answer is correct, but note that with only this as guideline we have no idea about where a possible modification in the action theory should be carried out in order to achieve full executability for *tease*. The same observation holds for Zhang et al.'s proposal.

Just to see how things can be even worse, let the same action theory as above, but with $\mathcal{X} = \{alive \rightarrow \langle tease \rangle \top\}$, obtained by its correction with the algorithms we proposed. Observe that the resulting theory satisfies all our postulates. It is not hard to see, however, that the representation of such an action theory in the above frameworks, when checked by their respective consistency tests, is still considered to have a problem.

This problem arises because Lang et al.'s proposal do not allow for executability laws, thus one cannot make the distinction between $\mathcal{X} = \{\langle tease \rangle \top\}$, $\mathcal{X} = \{alive \rightarrow \langle tease \rangle \top\}$ and $\mathcal{X} = \emptyset$. By their turn, Zhang et al. allows for specifying executabilities, however their consistency definitions do not distinguish the cases $alive \rightarrow \langle tease \rangle \top$ and $\langle tease \rangle \top$.

A concept similar to that of implicit static laws was firstly addressed, as far as we are concerned, in the realm of regulation consistency with deontic logic [49]. Indeed, the notions of regulation consistency given in the mentioned work and that of modularity presented in [37] and refined here can be proved to be equivalent. The main difference between the mentioned work and the approach in [37] relies on the fact that in [49] some syntactical restrictions on the formulas have to be made in order to make the algorithm to work.

Lifschitz and Ren [40] propose an action description language derived from $\mathcal{C}+$ [50] in which domain descriptions can also be decomposed in modules. Contrarily to our setting, in theirs a module is not a set of formulas for given action a, but rather a description of a subsystem of the theory, i.e., each module describes a set of interrelated fluents and actions. As an example, a module describing Lin's suitcase [2] should contain all causal laws in the sense of $\mathcal{C}+$ that are relevant to the scenario. Actions or fluents having nothing to do, neither directly nor indirectly, with the suitcase should be described in different modules. This feature makes such a decomposition somewhat domain-dependent, while here we have proposed a type-oriented modularization of the formulas, which does not depend on the domain.

In the referred work, modules can be defined in order to specialize other modules. This is done by making the new module to inherit and then specialize other modules' components. This is an important feature when elaborations are involved. In the suitcase example, adding a new action relevant to the suitcase description can be achieved by defining a new module inheriting all properties of the old one and containing the causal laws needed for the new action. Such ideas are interesting from the standpoint of software and knowledge engineer: reusability is an intrinsic property of the framework, and easy scalability promotes elaboration tolerance.

Consistency of a given theory and how to prevent conflicts between modules (independent or inherited) however is not addressed.

In this work we have illustrated by some examples what we can do in order to make a theory intuitive. This involves theory modification. Action theory change has been addressed in the recent literature on revision and update [51–53]. In [54] we have investigated this issue and shown the importance that modularity has in such a task.

10. Conclusion

Our contribution is twofold: general, as we presented postulates that apply to all reasoning about actions formalisms; and specific, as we proposed algorithms for a dependence-based solution to the frame problem.

We have defined here the concept of modularity of an action theory and pointed out some of the problems that arise if it is not satisfied. In particular we have argued that the non-dynamic part of action theories could influence but should not be influenced by the dynamic one.[9]

We have put forward some postulates, and in particular tried to demonstrate that when there are implicit static and inexecutability laws then one has slipped up in designing the action theory in question. As shown, a possible solution comes into its own with Algorithms 1 and 2, which can give us some guidelines in correcting an action theory if needed. By means of examples we have seen that there are several alternatives of correction, and choosing the right module to be modified as well as providing the intuitive information that must be supplied is up to the knowledge engineer.

Given the difficulty of exhaustively enumerating all the preconditions under which a given action is executable (and also those under which such an action cannot be executed), it is reasonable to expect that there is always going to be some executability precondition φ_1 and some inexecutability precondition φ_2 that together lead to a contradiction, forcing, thus, an implicit static law $\neg(\varphi_1 \wedge \varphi_2)$. This is the reason we propose to state some information about both executabilities and inexecutabilities, and then run the algorithms in order to improve the description.

It could be argued that unintuitive consequences in action theories are mainly due to badly written axioms and not to the lack of modularity. True enough, but what we have presented here is the case that making a domain description modular gives us a tool to detect at least some of such problems and correct it. (But note that we do not claim to correct badly written axioms automatically and once for all.) Besides this, having separate entities in the ontology and controlling their interaction help us to localize where the problems are, which can be crucial for real world applications.

In this work we have illustrated by some examples what we can do in order to make a theory intuitive. This involves theory modification. In [31] we have presented a general method for changing a domain description given a formula we want to contract. Their we have defined a semantics for theory contraction and also presented its syntactical counterpart through contraction operators. Soundness and completeness of such operators with respect to the semantics have been established. In that work we have also shown that modularity is a sufficient condition for contraction to be successful. This gives further evidence that the notion of modularity is fruitful.

Our postulates can be formulated in any reasoning about actions framework, but the algorithms require a decidable logic (in particular Algorithm 2). PDL is one candidate for that, as we have seen along the paper. For first-order-based frameworks the consistency checks of Algorithms 1 and 2 are undecidable. We can get rid of this by assuming that there is no function symbol in the language. In this way, the result of *NewCons*(.) is finite and the algorithm terminates.

[9] It might be objected that it is only by doing experiments that one learns the static laws that govern the universe. But note that this involves *learning*, whereas here—as always done in the reasoning about actions field—the static laws are known once forever, and do not evolve.

Hence another candidate for our ideas would have been the Situation Calculus fragment with only propositional fluents.

The present paper is also a step toward a solution to the problem of indirect dependences: indeed, if the implicit dependence $shoot \leadsto \neg walking$ is not in $\leadsto$, then after running Algorithm 2 we get an indirect inexecutability $(loaded \wedge walking) \rightarrow [shoot]\bot$, i.e., *shoot* cannot be executed if $loaded \wedge walking$ holds. Such an unintuitive inexecutability is not in $\mathcal{I}$ and thus indicates the missing indirect dependence.

The general case is nevertheless more complex, and it seems that such indirect dependences cannot be computed automatically in the case of indeterminate effects (cf. the example in [23]). We are currently investigating this issue.

The first work on formalizing modularity in logical systems in general is due to Garson [55]. Modularity of theories in reasoning about actions was originally defined in [15]. Modularization of ontologies in description logics is addressed in [56]. A different viewpoint of the work we presented here can be found in [12], where modularity of action theories is assessed from a software engineering perspective. A modularity-based approach for narrative reasoning about actions is given in [57]. In [13] we show that our modularity notion can also be extended to ontologies in the description logic $\mathcal{ALC}$.

Our postulates do not take into account causality statements linking propositions such as those defined in [2,3,39,58]. This could be a topic for further investigation.

Acknowledgements

The authors are grateful to the anonymous referees for useful comments on an earlier version of this paper.

Ivan Varzinczak has been supported by a fellowship from the government of the FEDERATIVE REPUBLIC OF BRAZIL. Grant: CAPES BEX 1389/01-7.

Appendix A. Proof of Theorem 22

Let $\mathcal{S}_{imp^*}$ be the output of Algorithm 1 on input $\langle \mathcal{S}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a \rangle$ and $\leadsto$. Then $\langle \mathcal{S}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a \rangle$ with $\leadsto$ satisfies Postulate **PS** *if and only if $\mathcal{S}_{imp^*} = \emptyset$.*

We recall that $\models_{\mathsf{CPL}}$ is logical consequence in classical propositional logic, and $PI(A)$ is the set of prime implicates of a set A of classical formulas.

Before giving the proof of the theorems, we recall some properties of prime implicates [35,59] and of the function *NewCons*(.) [36] (see Section 4). Let $\varphi \in \mathfrak{Fml}$, $A \subseteq \mathfrak{Fml}$ finite (identified with the conjunction of its formulas), and χ be a clause. Then

(1) $\models_{\mathsf{CPL}} \varphi \leftrightarrow \bigwedge PI(\varphi)$ [35, Corollary 3.2].
(2) $PI(A) \cup NewCons(\varphi, A) = PI(A \wedge \varphi)$ (by definition of *NewCons*(,)).
(3) $\models_{\mathsf{CPL}} (A \wedge \varphi) \leftrightarrow (A \wedge NewCons(\varphi, A))$ (from (1) and (2))
(4) If $PI(\varphi) \models_{\mathsf{CPL}} \chi$, then there is $\chi' \in PI(\varphi)$ such that $\chi' \models_{\mathsf{CPL}} \chi$ [35, Proposition 3.4].

Let $\langle \mathcal{S}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a \rangle$ with $\leadsto$ be an action theory for a, and let $\varphi \rightarrow \langle a \rangle \top \in \mathcal{X}^a$, $\mathcal{C}^a = \mathcal{E}^a \cup \mathcal{I}^a$, and $\hat{\mathcal{C}}^a \subseteq \mathcal{C}^a$. We define:

$$\varphi_{\hat{\mathcal{C}}^a} = \bigwedge \{\varphi_i \colon \varphi_i \rightarrow [a]\psi_i \in \hat{\mathcal{C}}^a\},$$

$$\psi_{\hat{\mathcal{C}}^a} = \bigwedge \{\psi_i \colon \varphi_i \rightarrow [a]\psi_i \in \hat{\mathcal{C}}^a\}.$$

Moreover, let $indep_a = \{\neg\ell \colon a \not\leadsto \ell\}$.

Lemma 47. *Let $indep'_a \subseteq indep_a$. $\mathcal{S} \cup \{\psi_{\hat{\mathcal{C}}^a}\} \cup indep'_a \models_{\mathsf{CPL}} \bot$ if and only if $\mathcal{S} \cup NewCons(\psi_{\hat{\mathcal{C}}^a}, \mathcal{S}) \cup indep'_a \models_{\mathsf{CPL}} \bot$.*

Proof.

$$\mathcal{S} \cup \{\psi_{\hat{\mathcal{C}}^a}\} \cup indep'_a \models_{\mathsf{CPL}} \bot$$

if and only if

$$PI\big(\mathcal{S} \cup \{\psi_{\hat{\mathcal{C}}_a}\}\big) \cup indep'_a \models_{\mathsf{CPL}} \bot \quad \text{(by property (1))}$$

if and only if

$$PI\big(\mathcal{S} \cup NewCons(\psi_{\hat{\mathcal{C}}_a}, \mathcal{S})\big) \cup indep'_a \models_{\mathsf{CPL}} \bot \quad \text{(by property (2))}$$

if and only if

$$\mathcal{S} \cup NewCons(\psi_{\hat{\mathcal{C}}_a}, \mathcal{S}) \cup indep'_a \models_{\mathsf{CPL}} \bot \quad \text{(by property (1))}. \qquad \square$$

Lemma 48. *Let $indep'_a \subseteq indep_a$. If $\mathcal{S} \cup NewCons(\psi_{\hat{\mathcal{C}}_a}, \mathcal{S}) \cup indep'_a \models_{\mathsf{CPL}} \bot$, then there exists $\chi \in NewCons(\psi_{\hat{\mathcal{C}}_a}, \mathcal{S})$ such that $\mathcal{S} \cup \{\chi\} \cup indep'_a \models_{\mathsf{CPL}} \bot$.*

Proof.

$$\mathcal{S} \cup NewCons(\psi_{\hat{\mathcal{C}}_a}, \mathcal{S}) \cup indep'_a \models_{\mathsf{CPL}} \bot$$

if and only if

$$PI(\mathcal{S}) \cup NewCons(\psi_{\hat{\mathcal{C}}_a}, \mathcal{S}) \cup indep'_a \models_{\mathsf{CPL}} \bot \quad \text{(by property (1))}$$

if and only if

$$PI\big(\mathcal{S} \cup \{\psi_{\hat{\mathcal{C}}_a}\}\big) \cup indep'_a \models_{\mathsf{CPL}} \bot \quad \text{(by property (2))}$$

if and only if

$$PI\big(\mathcal{S} \cup \{\psi_{\hat{\mathcal{C}}_a}\}\big) \models_{\mathsf{CPL}} \neg \bigwedge\{\neg\ell_i\colon \neg\ell_i \in indep'_a\}$$

if and only if

$$PI\big(\mathcal{S} \cup \{\psi_{\hat{\mathcal{C}}_a}\}\big) \models_{\mathsf{CPL}} \bigvee\{\ell_i\colon \neg\ell_i \in indep'_a\}$$

if and only if there exists $\chi \in PI(\mathcal{S} \cup \{\psi_{\hat{\mathcal{C}}_a}\})$ such that

$$\chi \models_{\mathsf{CPL}} \bigvee\{\ell_i\colon \neg\ell_i \in indep'_a\} \quad \text{(by property (4))}$$

if and only if

$$\{\chi\} \cup indep'_a \models_{\mathsf{CPL}} \bot$$

if and only if

$$\mathcal{S} \cup \{\chi\} \cup indep'_a \models_{\mathsf{CPL}} \bot. \qquad \square$$

Lemma 49. *Let $indep'_a \subseteq indep_a$. If we have $\mathcal{S} \cup \{\varphi, \varphi_{\hat{\mathcal{C}}_a}\} \cup indep'_a \not\models_{\mathsf{CPL}} \bot$ and $\mathcal{S} \cup NewCons(\psi_{\hat{\mathcal{C}}_a}, \mathcal{S}) \cup indep'_a \models_{\mathsf{CPL}} \bot$, then there exists $\chi \in NewCons(\psi_{\hat{\mathcal{C}}_a}, \mathcal{S})$ such that $\mathcal{S} \cup \{\chi\} \cup indep'_a \models_{\mathsf{CPL}} \bot$.*

Proof. By Lemma 48 and classical logic. □

Lemma 50. *Let $indep'_a \subseteq indep_a$. If we have $\mathcal{S} \cup \{\varphi, \varphi_{\hat{\mathcal{C}}_a}\} \cup indep'_a \not\models_{\mathsf{CPL}} \bot$ and $\mathcal{S} \cup NewCons(\psi_{\hat{\mathcal{C}}_a}, \mathcal{S}) \cup indep'_a \models_{\mathsf{CPL}} \bot$, then there exists $\chi \in NewCons(\psi_{\hat{\mathcal{C}}_a}, \mathcal{S})$ such that both $\mathcal{S} \cup \{\varphi, \varphi_{\hat{\mathcal{C}}_a}\} \cup indep'_a \not\models_{\mathsf{CPL}} \bot$ and $\mathcal{S} \cup \{\chi\} \cup indep'_a \models_{\mathsf{CPL}} \bot$.*

Proof. Trivially, by Lemma 49. □

Lemma 51. *Let* $indep'_a \subseteq indep_a$. *If* $\chi \in NewCons(\psi_{\hat{\mathcal{C}}^a}, \mathcal{S})$ *is such that* $\mathcal{S} \cup \{\varphi, \varphi_{\hat{\mathcal{C}}^a}\} \cup indep'_a \not\models_{\mathsf{CPL}} \bot$ *and* $\mathcal{S} \cup \{\chi\} \cup indep'_a \models_{\mathsf{CPL}} \bot$, *then both* $\mathcal{S} \cup \{\varphi, \varphi_{\hat{\mathcal{C}}^a}\} \cup \{\neg\ell_i\colon \ell_i \in \chi \text{ and } a \not\rightsquigarrow \ell_i\} \not\models_{\mathsf{CPL}} \bot$ *and* $\mathcal{S} \cup \{\chi\} \cup \{\neg\ell_i\colon \ell_i \in \chi \text{ and } a \not\rightsquigarrow \ell_i\} \models_{\mathsf{CPL}} \bot$.

Proof. Let $\mathcal{S} \cup \{\varphi, \varphi_{\hat{\mathcal{C}}^a}\} \cup indep'_a \not\models_{\mathsf{CPL}} \bot$ and $\chi \in NewCons(\psi_{\hat{\mathcal{C}}^a}, \mathcal{S})$ be such that $\mathcal{S} \cup \{\chi\} \cup indep'_a \models_{\mathsf{CPL}} \bot$.

If $\chi = \bot$, the result is trivial. Otherwise, we have the following cases:

- If $atm(\chi) \not\subset atm(indep'_a)$, then the premise is false (and the lemma trivially holds).
- If $atm(\chi) = atm(indep'_a)$, the lemma holds.
- Let $atm(\chi) \subset atm(indep'_a)$. Then, from

$$\mathcal{S} \cup \{\varphi, \varphi_{\hat{\mathcal{C}}^a}\} \cup indep'_a \not\models_{\mathsf{CPL}} \bot \quad \text{(the hypothesis)}$$

it follows

$$\mathcal{S} \cup \{\varphi, \varphi_{\hat{\mathcal{C}}^a}\} \cup \{\neg\ell_i\colon \ell_i \in \chi \text{ and } a \not\rightsquigarrow \ell_i\} \not\models_{\mathsf{CPL}} \bot.$$

From

$$\mathcal{S} \cup \{\chi\} \cup indep'_a \models_{\mathsf{CPL}} \bot \quad \text{(hypothesis)}$$

and because

$$\mathcal{S} \cup indep'_a \not\models_{\mathsf{CPL}} \bot,$$

it follows

$$\mathcal{S} \cup \{\chi\} \cup \{\neg\ell_i\colon \ell_i \in \chi \text{ and } a \not\rightsquigarrow \ell_i\} \models_{\mathsf{CPL}} \bot. \qquad \Box$$

Lemma 52. *If* $\chi \in NewCons(\psi_{\hat{\mathcal{C}}^a}, \mathcal{S})$ *is such that both* $\mathcal{S} \cup \{\varphi, \varphi_{\hat{\mathcal{C}}^a}\} \cup \{\neg\ell_i\colon \ell_i \in \chi \text{ and } a \not\rightsquigarrow \ell_i\} \not\models_{\mathsf{CPL}} \bot$ *and* $\mathcal{S} \cup \{\chi\} \cup \{\neg\ell_i\colon \ell_i \in \chi \text{ and } a \not\rightsquigarrow \ell_i\} \models_{\mathsf{CPL}} \bot$, *then* $\mathcal{S} \cup \{\varphi, \varphi_{\hat{\mathcal{C}}^a}, \neg\chi\} \not\models_{\mathsf{CPL}} \bot$ *and for all* $\ell_i \in \chi$, $a \not\rightsquigarrow \ell_i$.

Proof. From

$$\mathcal{S} \cup \{\varphi, \varphi_{\hat{\mathcal{C}}^a}\} \cup \{\neg\ell_i\colon \ell_i \in \chi \text{ and } a \not\rightsquigarrow \ell_i\} \not\models_{\mathsf{CPL}} \bot$$

we conclude

$$\mathcal{S} \cup \{\neg\ell_i\colon \ell_i \in \chi \text{ and } a \not\rightsquigarrow \ell_i\} \not\models_{\mathsf{CPL}} \bot.$$

From this and the hypothesis

$$\mathcal{S} \cup \{\chi\} \cup \{\neg\ell_i\colon \ell_i \in \chi \text{ and } a \not\rightsquigarrow \ell_i\} \models_{\mathsf{CPL}} \bot,$$

it follows

$$\mathcal{S} \cup \{\neg\ell_i\colon \ell_i \in \chi \text{ and } a \not\rightsquigarrow \ell_i\} \models_{\mathsf{CPL}} \neg\chi.$$

If $\mathcal{S} \models_{\mathsf{CPL}} \neg\chi$, then $\mathcal{S} \cup \{\psi_{\hat{\mathcal{C}}^a}\} \models_{\mathsf{CPL}} \neg\chi$, and because $\chi \in NewCons(\psi_{\hat{\mathcal{C}}^a}, \mathcal{S})$, we have $\chi \models_{\mathsf{CPL}} \neg\chi$, a contradiction. Hence $\mathcal{S} \cup \{\chi\} \not\models_{\mathsf{CPL}} \bot$.

Suppose now there is a literal $\ell \in \chi$ such that $\neg\ell \notin \{\neg\ell_i\colon \ell_i \in \chi \text{ and } a \not\rightsquigarrow \ell_i\}$. Then, the propositional valuation in which $\chi_{\ell \leftarrow true}$ satisfies

$$\mathcal{S} \cup \{\chi\} \cup \{\neg\ell_i\colon \ell_i \in \chi \text{ and } a \not\rightsquigarrow \ell_i\},$$

and then

$$\mathcal{S} \cup \{\chi\} \cup \{\neg\ell_i\colon \ell_i \in \chi \text{ and } a \not\rightsquigarrow \ell_i\} \not\models_{\mathsf{CPL}} \bot.$$

Hence there cannot be such a literal, and then for all $\ell_i \in \chi$, $a \not\rightsquigarrow \ell_i$.

Now, from $a \not\rightsquigarrow \ell_i$ for all $\ell_i \in \chi$, we have $\models_{\mathsf{CPL}} \bigwedge\{\neg\ell_i\colon\ \ell_i \in \chi \text{ and } a \not\rightsquigarrow \ell_i\} \leftrightarrow \neg\chi$. From this and the hypothesis

$$\mathcal{S} \cup \{\varphi, \varphi_{\hat{\mathcal{C}}^a}\} \cup \{\neg\ell_i\colon\ \ell_i \in \chi \text{ and } a \not\rightsquigarrow \ell_i\} \not\models_{\mathsf{CPL}} \bot$$

it follows $\mathcal{S} \cup \{\varphi, \varphi_{\hat{\mathcal{C}}^a}, \neg\chi\} \not\models_{\mathsf{CPL}} \bot$. □

Proof of Theorem 22. We are about to prove that $\langle\mathcal{S}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a\rangle$ with $\rightsquigarrow$ satisfies Postulate **PS** if and only if $\mathcal{S}_{imp^*} = \emptyset$.

($\Rightarrow$): Suppose $\mathcal{S}_{imp^*} \neq \emptyset$. Then at the first step of the algorithm there has been some $\varphi \to \langle a\rangle\top \in \mathcal{X}^a$ and some $\hat{\mathcal{C}}^a \subseteq \mathcal{C}^a$ such that for some $\chi \in \mathit{NewCons}(\psi_{\hat{\mathcal{C}}^a}, \mathcal{S})$, $\mathcal{S}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a \models_{\rightsquigarrow} \neg(\varphi \wedge \varphi_{\hat{\mathcal{C}}^a} \wedge \neg\chi)$ and $\mathcal{S} \not\models_{\mathsf{CPL}} \neg(\varphi \wedge \varphi_{\hat{\mathcal{C}}^a} \wedge \neg\chi)$. Hence $\langle\mathcal{S}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a\rangle$ does not satisfy Postulate **PS**.

($\Leftarrow$): Suppose that $\mathcal{S}_{imp^*} = \emptyset$. Therefore for all $\varphi' \to \langle a\rangle\top \in \mathcal{X}^a$ and for all subsets $\hat{\mathcal{C}}^a \subseteq \mathcal{C}^a$, we have that

$$\begin{aligned}&\text{for all } \chi \in \mathit{NewCons}(\psi_{\hat{\mathcal{C}}^a}, \mathcal{S}), \text{ if}\\ &\mathcal{S} \cup \{\varphi', \varphi_{\hat{\mathcal{C}}^a}, \neg\chi\} \not\models_{\mathsf{CPL}} \bot,\\ &\text{then there exists } \ell_i \in \chi \text{ such that } a \rightsquigarrow \ell_i.\end{aligned} \tag{A.1}$$

From (A.1) and Lemma 52, we get

$$\begin{aligned}&\text{for all } \chi \in \mathit{NewCons}(\psi_{\hat{\mathcal{C}}^a}, \mathcal{S}), \text{ if}\\ &\mathcal{S} \cup \{\varphi, \varphi_{\hat{\mathcal{C}}^a}\} \cup \{\neg\ell_i\colon\ \ell_i \in \chi \text{ and } a \not\rightsquigarrow \ell_i\} \not\models_{\mathsf{CPL}} \bot,\\ &\text{then } \mathcal{S} \cup \{\chi\} \cup \{\neg\ell_i\colon\ \ell_i \in \chi \text{ and } a \not\rightsquigarrow \ell_i\} \not\models_{\mathsf{CPL}} \bot.\end{aligned}$$

From this and Lemma 51, it follows that

$$\begin{aligned}&\text{for all } \chi \in \mathit{NewCons}(\psi_{\hat{\mathcal{C}}^a}, \mathcal{S}),\ \text{if } \mathcal{S} \cup \{\varphi', \varphi_{\hat{\mathcal{C}}^a}\} \cup \mathit{indep}'_a \not\models_{\mathsf{CPL}} \bot,\\ &\text{then } \mathcal{S} \cup \{\chi\} \cup \mathit{indep}'_a \not\models_{\mathsf{CPL}} \bot.\end{aligned}$$

This and Lemma 50 give us

$$\begin{aligned}&\text{if } \mathcal{S} \cup \{\varphi', \varphi_{\hat{\mathcal{C}}^a}\} \cup \mathit{indep}'_a \not\models_{\mathsf{CPL}} \bot, \text{ then}\\ &\mathcal{S} \cup \mathit{NewCons}(\psi_{\hat{\mathcal{C}}^a}, \mathcal{S}) \cup \mathit{indep}'_a \not\models_{\mathsf{CPL}} \bot\end{aligned}$$

From this and Lemma 47, it follows that for all $\mathit{indep}'_a \subseteq \mathit{indep}_a$, for every $\varphi' \to \langle a\rangle\top \in \mathcal{X}^a$ and all $\hat{\mathcal{C}}^a \subseteq \mathcal{C}^a$,

$$\text{if } \mathcal{S} \cup \{\varphi', \varphi_{\hat{\mathcal{C}}^a}\} \cup \mathit{indep}'_a \not\models_{\mathsf{CPL}} \bot,\ \text{then } \mathcal{S} \cup \{\psi_{\hat{\mathcal{C}}^a}\} \cup \mathit{indep}'_a \not\models_{\mathsf{CPL}} \bot. \tag{A.2}$$

Now, suppose $\mathcal{S} \not\models_{\mathsf{CPL}} \varphi$ for some propositional φ. We will build a model $\mathcal{M}$ such that $\mathcal{M}$ is a $\rightsquigarrow$-model for $\langle\mathcal{S}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a\rangle$ that does not satisfy φ.

Let $\mathcal{M} = \langle W, R_a\rangle$ be such that $W = \mathit{valuations}(\mathcal{S})$, and R_a be such that for all $w, w' \in W$, wR_aw' if and only if

- $\models^{\mathcal{M}}_{w'} \psi_i$ for every $\varphi_i \to [a]\psi_i \in \mathcal{C}^a$ such that $\models^{\mathcal{M}}_{w} \varphi_i$; and
- $\models^{\mathcal{M}}_{w'} \neg\ell$ for all ℓ such that $a \not\rightsquigarrow \ell$ and $\models^{\mathcal{M}}_{w} \neg\ell$.

We have that $\mathcal{M}$ is a $\rightsquigarrow$-model, by the definition of R_a. By the definition of W, $\mathcal{M}$ is a model of $\mathcal{S}$. We have that $\mathcal{M}$ is a model of $\mathcal{E}^a$ and $\mathcal{I}^a$, too: for every $\varphi_i \to [a]\psi_i \in \mathcal{C}^a$ and every world $w \in W$, if $\models^{\mathcal{M}}_{w} \varphi_i$, then, by the definition of R_a, $\models^{\mathcal{M}}_{w'} \psi_i$ for all $w' \in W$ such that wR_aw'. Moreover, $\mathcal{M}$ is also a model of $\mathcal{X}^a$: for every $\varphi_i \to \langle a\rangle\top \in \mathcal{X}^a$ and every world $w \in W$, if $\models^{\mathcal{M}}_{w} \varphi_i$, then

$$\mathcal{E}^a(w) = \{\varphi_i \to [a]\psi_i \in \mathcal{E}^a\colon\ \models^{\mathcal{M}}_{w} \varphi_i\},$$

and

$$\mathit{indep}_a(w) = \{\neg\ell\colon\ a \not\rightsquigarrow \ell \text{ and } \models^{\mathcal{M}}_{w} \neg\ell\}$$

are such that $\mathcal{S} \cup \{\varphi_i, \varphi_{\mathcal{E}^a(w)}\} \cup indep_a(w) \not\models_{\mathsf{CPL}} \bot$, where

$$\varphi_{\mathcal{E}^a(w)} = \bigwedge \{\varphi_i \colon \varphi_i \to [a]\psi_i \in \mathcal{E}^a(w)\}.$$

From this and (A.2), we have $\mathcal{S} \cup \{\psi_{\mathcal{E}^a(w)}\} \cup indep_a(w) \not\models_{\mathsf{CPL}} \bot$, where

$$\psi_{\mathcal{E}^a(w)} = \bigwedge \{\psi_i \colon \varphi_i \to [a]\psi_i \in \mathcal{E}^a(w)\}.$$

As W is maximal, there exists w' such that $\models_{w'}^{\mathcal{M}} \psi_{\mathcal{E}^a(w)} \wedge indep_a(w)$. As R_a is maximal by definition, we have wR_aw'. Hence there exists at least one w' such that wR_aw', and $\models_w^{\mathcal{M}} \langle a\rangle\top$.

Hence, $\mathcal{M}$ is a model of $\langle \mathcal{S}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a\rangle$ and $\leadsto$. Clearly $\not\models^{\mathcal{M}} \varphi$, by the definition of W. Hence $\mathcal{S}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a \not\models_{\leadsto} \varphi$. Therefore $\langle \mathcal{S}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a\rangle$ with $\leadsto$ satisfies Postulate **PS**. □

Appendix B. Proof of Theorem 31

Let $\mathcal{I}^a_{imp}$ be the output of Algorithm 2 on input $\langle \mathcal{S}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a\rangle$ and $\leadsto$. If $\langle \mathcal{S}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a\rangle$ with $\leadsto$ satisfies Postulate **PS**, *then it satisfies Postulate **PI** if and only if $\mathcal{I}^a_{imp} = \emptyset$.*

Let $\langle \mathcal{S}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a\rangle$ be an action theory for a with a dependence relation $\leadsto$. For every $\hat{\mathcal{E}}^a \subseteq \mathcal{E}^a$ we define:

$$\varphi_{\hat{\mathcal{E}}^a} = \bigwedge \{\varphi_i \colon \varphi_i \to [a]\psi_i \in \hat{\mathcal{E}}^a\},$$

$$\psi_{\hat{\mathcal{E}}^a} = \bigwedge \{\psi_i \colon \varphi_i \to [a]\psi_i \in \hat{\mathcal{E}}^a\}.$$

Moreover, let $indep_a = \{\neg\ell \colon a \not\leadsto \ell\}$.

Lemma 53. *If $\mathcal{S}, \mathcal{I}^a \not\models_{\mathsf{PDL}} (\varphi_{\hat{\mathcal{E}}^a} \wedge indep'_a) \to [a]\bot$ and $\mathcal{S} \cup \{\psi_{\hat{\mathcal{E}}^a}\} \cup indep'_a \models_{\mathsf{CPL}} \bot$, then there is $\chi \in NewCons(\psi_{\hat{\mathcal{E}}^a}, \mathcal{S})$ such that $\mathcal{S}, \mathcal{I}^a \not\models_{\mathsf{PDL}} (\varphi_{\hat{\mathcal{E}}^a} \wedge \neg\chi) \to [a]\bot$ and $a \not\leadsto \ell_i$ for all $\ell_i \in \chi$.*

Proof. Let $\mathcal{S}, \mathcal{I}^a \not\models_{\mathsf{PDL}} (\varphi_{\hat{\mathcal{E}}^a} \wedge indep'_a) \to [a]\bot$. Then there is a PDL-model $\mathcal{M} = \langle W, R_a\rangle$ such that $\models^{\mathcal{M}} \mathcal{S} \wedge \mathcal{I}^a$ and $\not\models^{\mathcal{M}} (\varphi_{\hat{\mathcal{E}}^a} \wedge indep'_a) \to [a]\bot$. This means that there is a possible world $v \in W$ such that $\models_v^{\mathcal{M}} \varphi_{\hat{\mathcal{E}}^a} \wedge indep'_a$ and $\not\models_v^{\mathcal{M}} [a]\bot$. From $\models_v^{\mathcal{M}} \varphi_{\hat{\mathcal{E}}^a} \wedge indep'_a$, it follows

$$\mathcal{S} \cup \{\varphi_{\hat{\mathcal{E}}^a}\} \cup indep'_a \not\models_{\mathsf{CPL}} \bot. \tag{B.1}$$

From hypothesis $\mathcal{S} \cup \{\psi_{\hat{\mathcal{E}}^a}\} \cup indep'_a \models_{\mathsf{CPL}} \bot$ and Lemma 47, we get

$$\mathcal{S} \cup NewCons(\psi_{\hat{\mathcal{E}}^a}, \mathcal{S}) \cup indep'_a \models_{\mathsf{CPL}} \bot$$

and from this and Lemma 48 we have that there is $\chi \in NewCons(\psi_{\hat{\mathcal{E}}^a}, \mathcal{S})$ such that

$$\mathcal{S} \cup \{\chi\} \cup indep'_a \models_{\mathsf{CPL}} \bot. \tag{B.2}$$

From (B.1), (B.2) and classical logic, there is $\chi \in NewCons(\psi_{\hat{\mathcal{E}}^a}, \mathcal{S})$ such that

$$\mathcal{S} \cup \{\varphi_{\hat{\mathcal{E}}^a}\} \cup indep'_a \not\models_{\mathsf{CPL}} \bot \text{ and } \mathcal{S} \cup \{\chi\} \cup indep'_a \models_{\mathsf{CPL}} \bot.$$

From this and Lemma 51 it follows that there is $\chi \in NewCons(\psi_{\hat{\mathcal{E}}^a}, \mathcal{S})$ such that

$$\mathcal{S} \cup \{\varphi, \varphi_{\hat{\mathcal{E}}^a}\} \cup \{\neg\ell_i \colon \ell_i \in \chi \text{ and } a \not\leadsto \ell_i\} \not\models_{\mathsf{CPL}} \bot$$

and

$$\mathcal{S} \cup \{\chi\} \cup \{\neg\ell_i \colon \ell_i \in \chi \text{ and } a \not\leadsto \ell_i\} \models_{\mathsf{CPL}} \bot.$$

This and Lemma 52 gives us that for all $\ell_i \in \chi$, $a \not\leadsto \ell_i$.

Now, because $\mathcal{M}$ above is such that $\models_v^{\mathcal{M}} \varphi_{\hat{\mathcal{E}}^a} \wedge indep'_a$, from this and $\mathcal{S} \cup \{\chi\} \cup indep'_a \models_{\mathsf{CPL}} \bot$, we have that $\models_v^{\mathcal{M}} \varphi_{\hat{\mathcal{E}}^a} \wedge \neg\chi$. Because $\not\models_v^{\mathcal{M}} [a]\bot$, we therefore have $\mathcal{S}, \mathcal{I}^a \not\models_{\mathsf{PDL}} (\varphi_{\hat{\mathcal{E}}^a} \wedge \neg\chi) \to [a]\bot$. □

Proof of Theorem 31. We are about to prove that if $\langle \mathcal{S}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a \rangle$ with $\leadsto$ satisfies Postulate **PS**, then it satisfies Postulate **PI** if and only if $\mathcal{I}^a_{imp} = \emptyset$.

($\Rightarrow$): Straightforward, as every time $\mathcal{S}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a \models_{\leadsto} \varphi \to [a]\bot$, we have $\mathcal{S}, \mathcal{I}^a \models_{\mathsf{PDL}} \varphi \to [a]\bot$, and then $\mathcal{I}^a_{imp}$ never changes.

($\Leftarrow$): Suppose that $\mathcal{I}^a_{imp} = \emptyset$. Therefore for all subsets $\hat{\mathcal{E}}^a \subseteq \mathcal{E}^a$, we have that

$$\begin{aligned} &\text{for all } \chi \in \mathit{NewCons}(\psi_{\hat{\mathcal{E}}^a}, \mathcal{S}),\ \text{if } \mathcal{S}, \mathcal{I}^a \not\models_{\mathsf{PDL}} (\varphi_{\hat{\mathcal{E}}^a} \wedge \neg\chi) \to [a]\bot, \\ &\text{then there exists } \ell_i \in \chi \text{ such that } a \leadsto \ell_i. \end{aligned} \tag{B.3}$$

From (B.3) and Lemma 53, it follows that for all $\hat{\mathcal{E}}^a \subseteq \mathcal{E}^a$,

$$\begin{aligned} &\text{if } \mathcal{S}, \mathcal{I}^a \not\models_{\mathsf{PDL}} (\varphi_{\hat{\mathcal{E}}^a} \wedge \mathit{indep}'_a) \to [a]\bot, \\ &\text{then } \mathcal{S} \cup \{\psi_{\hat{\mathcal{E}}^a}\} \cup \mathit{indep}'_a \not\models_{\mathsf{CPL}} \bot. \end{aligned} \tag{B.4}$$

Suppose $\mathcal{S}, \mathcal{I}^a \not\models_{\mathsf{PDL}} \varphi \to [a]\bot$ for some $\varphi \in \mathfrak{Fml}$. Then there exists a PDL-model $\mathcal{M} = \langle W, R_a \rangle$ such that $\models^{\mathcal{M}} \mathcal{S} \wedge \mathcal{I}^a$ and $\not\models^{\mathcal{M}} \varphi \to [a]\bot$. This means that there is a possible world $v \in W$ such that $\models^{\mathcal{M}}_v \varphi$ and $\not\models^{\mathcal{M}}_v [a]\bot$.

(We are going to build a $\leadsto$-model of $\langle \mathcal{S}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a \rangle$, and hence conclude that $\mathcal{S}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a \not\models_{\leadsto} \varphi \to [a]\bot$.)

For given $w \in W$, we define:

$$\mathcal{I}^a(w) = \{\varphi_i \to [a]\bot \in \mathcal{I}^a\colon \models^{\mathcal{M}}_w \varphi_i\}.$$

Because $\langle \mathcal{S}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a \rangle$ with $\leadsto$ satisfies Postulate **PS**, we can extend $\mathcal{M}$ to a big model $\mathcal{M}' = \langle W', R'_a \rangle$ such that $W = \mathit{valuations}(\mathcal{S})$, and R'_a is defined such that for all $w, w' \in W'$, $wR'_a w'$ if and only if

- $\models^{\mathcal{M}'}_{w'} \neg\ell$ for all ℓ such that $a \not\leadsto \ell$ and $\models^{\mathcal{M}'}_{w} \neg\ell$;
- $\models^{\mathcal{M}'}_{w'} \psi_i$ for every $\varphi_i \to [a]\psi_i \in \mathcal{E}^a$ such that $\models^{\mathcal{M}'}_{w} \varphi_i$; and
- $\mathcal{I}^a(w) = \emptyset$.

By definition, $\mathcal{M}'$ is a $\leadsto$-model. We also have $\models^{\mathcal{M}'} \mathcal{S}$, by the definition of W'. $\mathcal{M}'$ is a model of $\mathcal{E}^a$, too: for every $\varphi_i \to [a]\psi_i \in \mathcal{E}^a$ and every $w \in W'$, if $\models^{\mathcal{M}'}_{w} \varphi_i$, then $\models^{\mathcal{M}'}_{w'} \psi_i$ for all $w' \in W'$ such that $wR'_a w'$. Clearly $\mathcal{M}'$ is also a model of $\mathcal{I}^a$: for every $\varphi_i \to [a]\bot \in \mathcal{I}^a$ and every $w \in W'$, if $\models^{\mathcal{M}'}_{w} \varphi_i$, then $\mathcal{I}^a(w) \neq \emptyset$ and $R'_a(w) = \emptyset$. $\mathcal{M}'$ is a model of $\mathcal{X}^a$, too: for every $\varphi_i \to \langle a\rangle\top \in \mathcal{X}^a$ and every $w \in W'$, if $\models^{\mathcal{M}'}_{w} \varphi_i$, then

$$\mathcal{E}^a(w) = \{\varphi_i \to [a]\psi_i \in \mathcal{E}^a\colon \models^{\mathcal{M}}_w \varphi_i\}$$

and

$$\mathit{indep}_a(w) = \{\neg\ell\colon a \not\leadsto \ell \text{ and } \models^{\mathcal{M}}_w \neg\ell\}$$

are such that $\mathcal{S}, \mathcal{I}^a \not\models_{\mathsf{PDL}} (\varphi_{\mathcal{E}^a(w)} \wedge \mathit{indep}_a(w)) \to [a]\bot$, where

$$\varphi_{\mathcal{E}^a(w)} = \bigwedge \{\varphi_i\colon \varphi_i \to [a]\psi_i \in \mathcal{E}^a(w)\}.$$

The justification is that $\mathcal{S}, \mathcal{I}^a \models_{\mathsf{PDL}} (\varphi_{\mathcal{E}^a(w)} \wedge \mathit{indep}_a(w)) \to [a]\bot$ would imply $\mathcal{S}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a \models_{\leadsto} (\varphi_{\mathcal{E}^a(w)} \wedge \mathit{indep}_a(w)) \to [a]\bot$, and as long as $\varphi_i \to \langle a\rangle\top \in \mathcal{X}^a$, $\mathcal{S}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a \models_{\leadsto} \neg(\varphi_i \wedge \varphi_{\mathcal{E}^a(w)} \wedge \mathit{indep}_a(w))$. As by hypothesis $\langle \mathcal{S}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a \rangle$ satisfies **PS**, $\mathcal{S} \models_{\mathsf{CPL}} \neg(\varphi_i \wedge \varphi_{\mathcal{E}^a(w)} \wedge \mathit{indep}_a(w))$, and then $w \notin W'$.

Hence, from $\mathcal{S}, \mathcal{I}^a \not\models_{\mathsf{PDL}} (\varphi_{\mathcal{E}^a(w)} \wedge \mathit{indep}_a(w)) \to [a]\bot$ and (B.4), it follows that $\mathcal{S} \cup \{\psi_{\mathcal{E}^a(w)}\} \cup \mathit{indep}_a(w) \not\models_{\mathsf{CPL}} \bot$, where

$$\psi_{\mathcal{E}^a(w)} = \bigwedge \{\psi_i\colon \varphi_i \to [a]\psi_i \in \mathcal{E}^a(w)\}.$$

As W' is maximal, there exists w' such that $\models^{\mathcal{M}'}_{w'} \psi_{\mathcal{E}^a(w)} \wedge \mathit{indep}_a(w)$. As R'_a is maximal by definition, we have $wR'_a w'$. Hence there exists at least one w' such that $wR'_a w'$, and then $\models^{\mathcal{M}'}_{w'} \langle a\rangle\top$.

Therefore, $\mathcal{M}'$ is a model of $\langle \mathcal{S}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a \rangle$ and $\leadsto$.

Looking at $v \in W'$, we must have $\mathcal{S}, \mathcal{I}^a \not\models_{\mathsf{PDL}} (\varphi_{\mathcal{E}^a(v)} \wedge \mathit{indep}_a(v)) \to [a]\bot$, because otherwise $R_a(v) = \emptyset$, against the hypothesis that $\not\models^{\mathcal{M}}_v [a]\bot$. Hence, from (B.4) it follows that $\mathcal{S} \cup \{\psi_{\mathcal{E}^a(v)}\} \cup \mathit{indep}_a(v) \not\models_{\mathsf{CPL}} \bot$, and then there exists at least one v' such that $vR'_a v'$, and then $\models_{v'}^{\mathcal{M}'} \langle a\rangle\top$. From this it follows that $\mathcal{S}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a \not\models_{\leadsto} \varphi \to [a]\bot$. Therefore $\langle \mathcal{S}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a\rangle$ and $\leadsto$ satisfy Postulate **PI**. □

Appendix C. Proof of Theorem 37

Let $\langle \mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I}\rangle$ with $\leadsto$ satisfy **PS***. *$\langle \mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I}\rangle$ with $\leadsto$ satisfies* **PI*** *if and only if $\langle \mathcal{S}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a\rangle$ and $\leadsto$ satisfy* **PI** *for all $a \in \mathfrak{Act}$.*

($\Rightarrow$): Suppose that $\mathcal{S}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a \models_{\leadsto} \varphi \to [a]\bot$. By monotonicity, $\mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \models_{\leadsto} \varphi \to [a]\bot$, too. Now suppose that $\mathcal{S}, \mathcal{I}^a \not\models_{\mathsf{PDL}} \varphi \to [a]\bot$. Then there exists a possible worlds model $\mathcal{M} = \langle W, R_a\rangle$ such that $\models^{\mathcal{M}} \mathcal{S} \wedge \mathcal{I}^a$ and there is a possible world $v \in W$ such that $\models_v^{\mathcal{M}} \varphi$ and $\not\models_v^{\mathcal{M}} [a]\bot$. Let $\mathcal{M}' = \langle W', R'\rangle$ be such that $W' = W$, and $R'_{a'} = \emptyset$, for all $a' \neq a$, and $R'_a = R_a$. Then $\models^{\mathcal{M}'} \mathcal{S} \wedge \mathcal{I}$, and then $\mathcal{S}, \mathcal{I} \not\models_{\mathsf{PDL}} \varphi \to [a]\bot$. Hence $\langle \mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I}\rangle$ with $\leadsto$ does not satisfy **PI***.

($\Leftarrow$): Suppose $\langle \mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I}\rangle$ and $\leadsto$ do not satisfy Postulate **PI***. Then there exists $\varphi \in \mathfrak{Fml}$ such that $\mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \models_{\leadsto} \varphi \to [a]\bot$ and $\mathcal{S}, \mathcal{I} \not\models_{\mathsf{PDL}} \varphi \to [a]\bot$.

Claim. $\mathcal{S}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a \models_{\leadsto} \varphi \to [a]\bot$.

Proof of the claim. Suppose $\mathcal{S}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a \not\models_{\leadsto} \varphi \to [a]\bot$. Then there exists a $\leadsto$-model $\mathcal{M} = \langle W, R_a\rangle$ such that $\models^{\mathcal{M}} \mathcal{S} \wedge \mathcal{E}^a \wedge \mathcal{X}^a \wedge \mathcal{I}^a$ and $\not\models^{\mathcal{M}} \varphi \to [a]\bot$. This means that there is a possible world $v \in W$ such that $\models_v^{\mathcal{M}} \varphi$ and $\not\models_v^{\mathcal{M}} [a]\bot$, i.e., there is $v' \in W$ such that $R_a(v) = v'$.

(We extend $\mathcal{M}$ to all other actions $\langle \mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I}\rangle$ speaks of and obtain a $\leadsto$-model of $\langle \mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I}\rangle$.)

Given $w \in W$, for each $a_i \in \mathfrak{Act}$ we define:

$$\mathcal{I}^{a_i}(w) = \{\varphi_j \to [a_i]\bot \in \mathcal{I}^{a_i}\colon \models_w^{\mathcal{M}} \varphi_j\},$$
$$\mathcal{X}^{a_i}(w) = \{\varphi_j \to \langle a_i\rangle\top \in \mathcal{X}^{a_i}\colon \models_w^{\mathcal{M}} \varphi_j\}.$$

Let $\mathcal{M}' = \langle W', R'\rangle$ be such that $W' = W$, and $R' = R_a \cup \bigcup_{a' \neq a} R_{a'}$, where for each $a' \neq a$ and every $w, w' \in W'$, $wR_{a'}w'$ if and only if

- $\models_{w'}^{\mathcal{M}'} \neg\ell$ for all ℓ such that $a' \not\leadsto \ell$ and $\models_w^{\mathcal{M}'} \neg\ell$;
- $\models_{w'}^{\mathcal{M}'} \psi_i$ for every $\varphi_i \to [a']\psi_i \in \mathcal{E}^{a'}$ such that $\models_w^{\mathcal{M}'} \varphi_i$; and
- $\mathcal{I}^{a'}(w) = \emptyset$.

By definition, $\mathcal{M}'$ is a model of the dependence relation $\leadsto$. Because, by hypothesis, $\langle \mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I}\rangle$ with $\leadsto$ satisfies **PS***, there is no implicit static law, i.e., for every $a_i \in \mathfrak{Act}$ and every $w \in W'$, if $\mathcal{I}^{a_i}(w) \neq \emptyset$, then $\mathcal{X}^{a_i}(w) = \emptyset$. Then, as $W' = \mathit{valuations}(\mathcal{S})$, $\mathcal{M}'$ is a model of $\mathcal{S}$. We have that $\mathcal{M}'$ is a model of $\mathcal{E}$, too: it is a model of $\mathcal{E}^a$, and given $a' \neq a$, for every $\varphi_i \to [a']\psi_i \in \mathcal{E}$ and every $w \in W'$, if $\models_w^{\mathcal{M}'} \varphi_i$, then $\models_{w'}^{\mathcal{M}'} \psi_i$ for all $w' \in W'$ such that $wR_{a'}w'$. Clearly $\mathcal{M}'$ is also a model of $\mathcal{I}$: it is a model of $\mathcal{I}^a$, and given $a' \neq a$, for every $\varphi_i \to [a']\bot \in \mathcal{I}$ and every $w \in W'$, if $\models_w^{\mathcal{M}'} \varphi_i$, then $\mathcal{I}^{a'}(w) \neq \emptyset$ and $R_{a'}(w) = \emptyset$. $\mathcal{M}'$ is a model of $\mathcal{X}$, too: besides being a model of $\mathcal{X}^a$, for every $a' \neq a$ and all worlds $w \in W'$ such that $\mathcal{X}^{a'}(w) \neq \emptyset$ there is a world accessible by $R_{a'}$, because $R_{a'}(w) = \emptyset$ in this case would preclude $\mathcal{X}^{a'}(w) \neq \emptyset$, and otherwise $w \notin W'$, which is impossible as long as **PS*** is satisfied. Thus $\models^{\mathcal{M}'} \mathcal{S} \wedge \mathcal{E} \wedge \mathcal{X} \wedge \mathcal{I}$, but if this is the case, $\mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \not\models_{\leadsto} \varphi \to [a]\bot$, hence we must have $\mathcal{S}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a \models_{\leadsto} \varphi \to [a]\bot$. (End of the proof of the claim.)

From $\mathcal{S}, \mathcal{I} \not\models_{\mathsf{PDL}} \varphi \to [a]\bot$ it follows $\mathcal{S}, \mathcal{I}^a \not\models_{\mathsf{PDL}} \varphi \to [a]\bot$. Putting all the results together, we have that $\langle \mathcal{S}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a\rangle$ with $\leadsto$ does not satisfy Postulate **PI**. □

Appendix D. Proof of Theorem 39

If $\langle \mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I}\rangle$ and $\leadsto$ satisfy Postulate **PS***, *then $\mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \models_{\leadsto} \varphi \to [a]\psi$ if and only if $\mathcal{S}, \mathcal{E}^a, \mathcal{I}^a \models_{\leadsto} \varphi \to [a]\psi$.*

(⇒): Let $\langle \mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \rangle$ and $\leadsto$ satisfy Postulate $\mathbf{PS}^*$, and also suppose that $\mathcal{S}, \mathcal{E}^a, \mathcal{I}^a \not\models_{\leadsto} \varphi \to [a]\psi$. Then there exists a $\leadsto$-model $\mathcal{M} = \langle W, R_a \rangle$, such that $\models^{\mathcal{M}} \mathcal{S} \wedge \mathcal{E}^a \wedge \mathcal{I}^a$ and $\not\models^{\mathcal{M}} \varphi \to [a]\psi$. This means that there is a possible world $v \in W$ such that $\models^{\mathcal{M}}_v \varphi$ and $\not\models^{\mathcal{M}}_v [a]\psi$, i.e., there is $v' \in W$ such that $R_a(v) = v'$ and $\not\models^{\mathcal{M}}_{v'} \psi$.

(We will extend $\mathcal{M}$ to obtain a $\leadsto$-model of $\langle \mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \rangle$ and thus show that $\mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \not\models_{\leadsto} \varphi \to [a]\psi$.)

Given $w \in W$, for each $a_i \in \mathfrak{Act}$ we define:

$$\mathcal{I}^{a_i}(w) = \{\varphi_j \to [a_i]\bot \in \mathcal{I}^{a_i} \colon \models^{\mathcal{M}}_w \varphi_j\},$$
$$\mathcal{X}^{a_i}(w) = \{\varphi_j \to \langle a_i \rangle\top \in \mathcal{X}^{a_i} \colon \models^{\mathcal{M}}_w \varphi_j\}.$$

Let $\mathcal{M}' = \langle W', R' \rangle$ be such that $W' = W$, and $R' = R_a \cup \bigcup_{a' \neq a} R_{a'}$, where for each $a' \neq a$ and every $w, w' \in W'$, $w R_{a'} w'$ if and only if

- $\models^{\mathcal{M}'}_{w'} \neg\ell$ for all ℓ such that $a' \not\leadsto \ell$ and $\models^{\mathcal{M}'}_w \neg\ell$;
- $\models^{\mathcal{M}'}_{w'} \psi_i$ for every $\varphi_i \to [a']\psi_i \in \mathcal{E}^{a'}$ such that $\models^{\mathcal{M}'}_w \varphi_i$; and
- $\mathcal{I}^{a'}(w) = \emptyset$.

By definition, $\mathcal{M}'$ is a model of the dependence relation $\leadsto$. Because, by hypothesis, $\langle \mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \rangle$ satisfies $\mathbf{PS}^*$, there is no implicit static law, i.e., for every $a_i \in \mathfrak{Act}$ and every $w \in W'$, if $\mathcal{I}^{a_i}(w) \neq \emptyset$, then $\mathcal{X}^{a_i}(w) = \emptyset$. Then, as $W' = \textit{valuations}(\mathcal{S})$, $\mathcal{M}'$ is a model of $\mathcal{S}$. We have that $\mathcal{M}'$ is a model of $\mathcal{E}$, too: it is a model of $\mathcal{E}^a$, and given $a' \neq a$, for every $\varphi_i \to [a']\psi_i \in \mathcal{E}$ and every $w \in W'$, if $\models^{\mathcal{M}'}_w \varphi_i$, then $\models^{\mathcal{M}'}_{w'} \psi_i$ for all $w' \in W'$ such that $w R_{a'} w'$. Clearly $\mathcal{M}'$ is also a model of $\mathcal{I}$: besides being a model of $\mathcal{I}^a$, given $a' \neq a$, for every $\varphi_i \to [a']\bot \in \mathcal{I}$ and every $w \in W'$, if $\models^{\mathcal{M}'}_w \varphi_i$, then $\mathcal{I}^{a'}(w) \neq \emptyset$ and $R_{a'}(w) = \emptyset$. $\mathcal{M}'$ is a model of $\mathcal{X}$, too: it is a model of $\mathcal{X}^a$, and for every $a' \neq a$ and all worlds $w \in W'$ such that $\mathcal{X}^{a'}(w) \neq \emptyset$ there is a world accessible by $R_{a'}$, because $R_{a'}(w) = \emptyset$ in this case would preclude $\mathcal{X}^{a'}(w) \neq \emptyset$, and otherwise $w \notin W'$, which is impossible as long as $\mathbf{PS}^*$ is satisfied. Thus $\models^{\mathcal{M}'} \mathcal{S} \wedge \mathcal{E} \wedge \mathcal{X} \wedge \mathcal{I}$. Because there are $v, v' \in W'$ such that $\models^{\mathcal{M}'}_v \varphi$, $v R_a v'$ and $\not\models^{\mathcal{M}'}_{v'} \psi$, we have $\mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \not\models_{\leadsto} \varphi \to [a]\psi$.

(⇐): Suppose $\mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \not\models_{\leadsto} \varphi \to [a]\psi$. Then there is a $\leadsto$-model $\mathcal{M}$ such that $\models^{\mathcal{M}} \mathcal{S} \wedge \mathcal{E} \wedge \mathcal{X} \wedge \mathcal{I}$ and $\not\models^{\mathcal{M}} \varphi \to [a]\psi$. Then, given a, we have $\models^{\mathcal{M}} \mathcal{S} \wedge \mathcal{E}^a \wedge \mathcal{X}^a \wedge \mathcal{I}^a$, and then $\models^{\mathcal{M}} \mathcal{S} \wedge \mathcal{E}^a \wedge \mathcal{I}^a$. Hence $\mathcal{S}, \mathcal{E}^a, \mathcal{I}^a \not\models_{\leadsto} \varphi \to [a]\psi$. □

Appendix E. Proof of Theorem 40

If $\langle \mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \rangle$ *and* $\leadsto$ *satisfy Postulate* $\mathbf{PS}^*$, *then* $\mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \models_{\leadsto} \varphi \to \langle a \rangle\top$ *if and only if* $\mathcal{S}, \mathcal{X}^a \models_{\leadsto} \varphi \to \langle a \rangle\top$.

(⇒): Let $\langle \mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \rangle$ and $\leadsto$ satisfy Postulate $\mathbf{PS}^*$, and suppose $\mathcal{S}, \mathcal{X}^a \not\models_{\mathsf{PDL}} \varphi \to \langle a \rangle\top$. Then there exists a PDL-model $\mathcal{M} = \langle W, R_a \rangle$, such that $\models^{\mathcal{M}} \mathcal{S} \wedge \mathcal{X}^a$ and $\not\models^{\mathcal{M}} \varphi \to \langle a \rangle\top$. This means that there is a possible world $v \in W$ such that $\models^{\mathcal{M}}_v \varphi$ and $\not\models^{\mathcal{M}}_v \langle a \rangle\top$.

(We extend $\mathcal{M}$ to build a $\leadsto$-model of $\langle \mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \rangle$ and then conclude that $\mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \not\models_{\leadsto} \varphi \to \langle a \rangle\top$.)

Given $w \in W$, for each $a_i \in \mathfrak{Act}$ we define:

$$\mathcal{I}^{a_i}(w) = \{\varphi_j \to [a_i]\bot \in \mathcal{I}^{a_i} \colon \models^{\mathcal{M}}_w \varphi_j\},$$
$$\mathcal{X}^{a_i}(w) = \{\varphi_j \to \langle a_i \rangle\top \in \mathcal{X}^{a_i} \colon \models^{\mathcal{M}}_w \varphi_j\}.$$

Let $\mathcal{M}' = \langle W', R' \rangle$ be such that $W' = W$, and $R' = R_a \cup \bigcup_{a' \neq a} R_{a'}$, where for each $a' \neq a$ and every $w, w' \in W'$, $w R_{a'} w'$ if and only if

- $\models^{\mathcal{M}'}_{w'} \neg\ell$ for all ℓ such that $a' \not\leadsto \ell$ and $\models^{\mathcal{M}'}_w \neg\ell$;
- $\models^{\mathcal{M}'}_{w'} \psi_i$ for every $\varphi_i \to [a']\psi_i \in \mathcal{E}^{a'}$ such that $\models^{\mathcal{M}'}_w \varphi_i$; and
- $\mathcal{I}^{a'}(w) = \emptyset$.

By definition, $\mathcal{M}'$ is a model of the dependence relation $\leadsto$. Because, by hypothesis, $\langle \mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \rangle$ satisfies $\mathbf{PS}^*$, there is no implicit static law, i.e., for every $a_i \in \mathfrak{Act}$ and every $w \in W'$, if $\mathcal{X}^{a_i}(w) \neq \emptyset$, then $\mathcal{I}^{a_i}(w) = \emptyset$. Then, as $W' = \textit{valuations}(\mathcal{S})$, $\mathcal{M}'$ is a model of $\mathcal{S}$. We have that $\mathcal{M}'$ is a model of $\mathcal{E}$, too: it is a model of $\mathcal{E}^a$, and given $a' \neq a$, for every $\varphi_i \to [a']\psi_i \in \mathcal{E}$ and every $w \in W'$, if $\models^{\mathcal{M}'}_w \varphi_i$, then $\models^{\mathcal{M}'}_{w'} \psi_i$ for all $w' \in W'$ such that $w R_{a'} w'$. Clearly $\mathcal{M}'$ is also a model of $\mathcal{I}$: it is a model of $\mathcal{I}^a$, and given $a' \neq a$, for every $\varphi_i \to [a']\bot \in \mathcal{I}$ and every $w \in W'$, if $\models_w^{\mathcal{M}'} \varphi_i$, then $\mathcal{I}^{a'}(w) \neq \emptyset$ and $R_{a'}(w) = \emptyset$. $\mathcal{M}'$ is a model of $\mathcal{X}$, too: besides being a model of $\mathcal{X}^a$, for every $a' \neq a$ and all worlds $w \in W'$ such that $\mathcal{X}^{a'}(w) \neq \emptyset$ there is a world accessible by $R_{a'}$, because $R_{a'}(w) = \emptyset$ in this case would preclude $\mathcal{X}^{a'}(w) \neq \emptyset$, and otherwise $w \notin W'$, which is impossible as long as **PS*** is satisfied. Hence $\models^{\mathcal{M}'} \mathcal{S} \wedge \mathcal{E} \wedge \mathcal{X} \wedge \mathcal{I}$. Because there is $v \in W'$ such that $\models_v^{\mathcal{M}'} \varphi$ and $\not\models_v^{\mathcal{M}'} \langle a \rangle \top$, we have $\mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \not\models_{\leadsto} \varphi \to \langle a \rangle \top$.

($\Leftarrow$): Suppose $\mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \not\models_{\leadsto} \varphi \to \langle a \rangle \top$. Then there is a $\leadsto$-model $\mathcal{M}$ such that $\models^{\mathcal{M}} \mathcal{S} \wedge \mathcal{E} \wedge \mathcal{X} \wedge \mathcal{I}$ and $\not\models^{\mathcal{M}} \varphi \to \langle a \rangle \top$. Then, given a, we have $\models^{\mathcal{M}} \mathcal{S} \wedge \mathcal{E}^a \wedge \mathcal{X}^a \wedge \mathcal{I}^a$, and then $\models^{\mathcal{M}} \mathcal{S} \wedge \mathcal{X}^a$. Moreover, by definition, $\mathcal{M}$ is a PDL-model. Hence $\mathcal{S}, \mathcal{X}^a \not\models_{\mathsf{PDL}} \varphi \to \langle a \rangle \top$. □

Appendix F. Proof of Theorem 43

If $\langle \mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \rangle$ and $\leadsto$ satisfy Postulate **PS****, then $\mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \models_{\leadsto} \varphi \to [a_1; \ldots; a_n]\psi$ if and only if $\mathcal{S}, \mathcal{E}^{a_1,\ldots,a_n}$, $\mathcal{I}^{a_1,\ldots,a_n} \models_{\leadsto} \varphi \to [a_1; \ldots; a_n]\psi$.*

Lemma 54. *If $\mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \models_{\leadsto} \varphi \to [a_1; \ldots; a_n]\psi$, then there is $\varphi' \in \mathfrak{Fml}$ such that $\mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \models_{\leadsto} \varphi \to [a_1; \ldots; a_{n-1}]\varphi'$ and $\mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \models_{\leadsto} \varphi' \to [a_n]\psi$.*

Proof. Let $\mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \models_{\leadsto} \varphi \to [a_1; \ldots; a_n]\psi$. In the case we have $\mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \models_{\leadsto} \varphi \to [a_1; \ldots; a_n]\bot$, the result immediately follows. Then, given a model $\mathcal{M} = \langle W, R \rangle$ of $\langle \mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \rangle$ such that $\models_w^{\mathcal{M}} \varphi$ for some $w \in W$, if $\models_w^{\mathcal{M}} \langle a_1; \ldots; a_n \rangle \top$, there must be at least one w'_{n-1} such that $\models_{w'_{n-1}}^{\mathcal{M}} [a_n]\psi$. Take all such w'_{n-1} and let φ' be

$$\bigvee_{\models_{w'_{n-1}}^{\mathcal{M}} [a_n]\psi} w'_{n-1}.$$

Then we have both $\mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \models_{\leadsto} \varphi \to [a_1; \ldots; a_{n-1}]\varphi'$, and $\mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \models_{\leadsto} \varphi' \to [a_n]\psi$. □

Proof of Theorem 43. ($\Rightarrow$): The proof is by induction on the number of action operators.

Base: $n = 1$. As $\langle \mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \rangle$ satisfies Postulate **PS***, the result follows from Theorem 39.

Induction hypothesis: for any $k < n$, if $\mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \models_{\leadsto} \varphi \to [a_1; \ldots; a_k]\psi$, then $\mathcal{S}, \mathcal{E}^{a_1,\ldots,a_k}, \mathcal{I}^{a_1,\ldots,a_k} \models_{\leadsto} \varphi \to [a_1; \ldots; a_k]\psi$.

Step: let $\mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \models_{\leadsto} \varphi \to [a_1; \ldots; a_n]\psi$. By Lemma 54, there is a classical formula φ' such that $\mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \models_{\leadsto} \varphi \to [a_1; \ldots; a_{n-1}]\varphi'$ and $\mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \models_{\leadsto} \varphi' \to [a_n]\psi$. From the induction hypothesis, we have $\mathcal{S}, \mathcal{E}^{a_1,\ldots,a_{n-1}}, \mathcal{I}^{a_1,\ldots,a_{n-1}} \models_{\leadsto} \varphi \to [a_1; \ldots; a_{n-1}]\varphi'$ and $\mathcal{S}, \mathcal{E}^{a_n}, \mathcal{I}^{a_n} \models_{\leadsto} \varphi' \to [a_n]\psi$. Thus $\mathcal{S}, \mathcal{E}^{a_1,\ldots,a_n}, \mathcal{I}^{a_1,\ldots,a_n} \models_{\leadsto} \varphi \to [a_1; \ldots; a_n]\psi$.

($\Leftarrow$): Suppose $\mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \not\models_{\leadsto} \varphi \to [a_1; \ldots; a_n]\psi$. Then there is a $\leadsto$-model $\mathcal{M}$ such that $\models^{\mathcal{M}} \mathcal{S} \wedge \mathcal{E} \wedge \mathcal{X} \wedge \mathcal{I}$ and $\not\models^{\mathcal{M}} \varphi \to [a_1; \ldots; a_n]\psi$. Then, given $a_1, \ldots, a_n$, we have $\models^{\mathcal{M}} \mathcal{S} \wedge \mathcal{E}^{a_1,\ldots,a_n} \wedge \mathcal{X}^{a_1,\ldots,a_n} \wedge \mathcal{I}^{a_1,\ldots,a_n}$, and then $\models^{\mathcal{M}} \mathcal{S} \wedge \mathcal{E}^{a_1,\ldots,a_n} \wedge \mathcal{I}^{a_1,\ldots,a_n}$. Hence $\mathcal{S}, \mathcal{E}^{a_1,\ldots,a_n}, \mathcal{I}^{a_1,\ldots,a_n} \not\models_{\leadsto} \varphi \to [a_1; \ldots; a_n]\psi$. □

Appendix G. Proof of Theorem 45

Let $\langle \mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \rangle$ and $\leadsto$ satisfy Postulate **PS****. Then we have $\mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \models_{\leadsto} \varphi \to \langle a_1; \ldots; a_n \rangle\psi$ if and only if $\mathcal{S}, \mathcal{E}^{a_1,\ldots,a_n}, \mathcal{X}^{a_1,\ldots,a_n}, \mathcal{I}^{a_1,\ldots,a_n} \models_{\leadsto} \varphi \to \langle a_1; \ldots; a_n \rangle\psi$.*

Lemma 55. *Let $\langle \mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \rangle$ and $\leadsto$ satisfy Postulate* **PS****. If $\mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \models_{\leadsto} \varphi \to \langle a \rangle\psi$, then $\mathcal{S}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a \models_{\leadsto} \varphi \to \langle a \rangle\psi$.*

Proof. Let $\langle \mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \rangle$ satisfy Postulate **PS*** and suppose $\mathcal{S}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a \not\models_{\leadsto} \varphi \to \langle a \rangle\psi$. Then there exists a $\leadsto$-model $\mathcal{M} = \langle W, R_a \rangle$, such that both $\models^{\mathcal{M}} \mathcal{S} \wedge \mathcal{E}^a \wedge \mathcal{X}^a \wedge \mathcal{I}^a$ and $\not\models^{\mathcal{M}} \varphi \to \langle a \rangle\psi$. This means that there is a possible world $v \in W$ such that $\models_v^{\mathcal{M}} \varphi$ and $\not\models_v^{\mathcal{M}} \langle a \rangle\psi$.

(We extend $\mathcal{M}$ to build a model of $\langle \mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \rangle$ and then conclude that $\mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \not\models_{\leadsto} \varphi \to \langle a \rangle\psi$.)

Given $w \in W$, for each $a_i \in \mathfrak{Act}$ we define:

$$\mathcal{I}^{a_i}(w) = \{\varphi_j \to [a_i]\bot \in \mathcal{I}^{a_i} \colon \models_w^{\mathcal{M}} \varphi_j\},$$

$$\mathcal{X}^{a_i}(w) = \{\varphi_j \to \langle a_i\rangle\top \in \mathcal{X}^{a_i} \colon \models_w^{\mathcal{M}} \varphi_j\}.$$

Let $\mathcal{M}' = \langle W', R'\rangle$ be such that $W' = W$, and $R' = R_a \cup \bigcup_{a' \neq a} R_{a'}$ (we extend $\mathcal{M}$ to all other actions $\langle \mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I}\rangle$ speaks of), where for each $a' \neq a$ and every $w, w' \in W'$, $wR_{a'}w'$ if and only if

- $\models_{w'}^{\mathcal{M}'} \neg\ell$ for all ℓ such that $a' \not\leadsto \ell$ and $\models_w^{\mathcal{M}'} \neg\ell$;
- $\models_{w'}^{\mathcal{M}'} \psi_i$ for every $\varphi_i \to [a']\psi_i \in \mathcal{E}^{a'}$ such that $\models_w^{\mathcal{M}'} \varphi_i$; and
- $\mathcal{I}^{a'}(w) = \emptyset$.

By definition, $\mathcal{M}'$ is a model of the dependence relation $\leadsto$. Because, by hypothesis, $\langle \mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I}\rangle$ satisfies **PS***, there is no implicit static law, i.e., for every $a_i \in \mathfrak{Act}$ and every $w \in W'$, if $\mathcal{X}^{a_i}(w) \neq \emptyset$, then $\mathcal{I}^{a_i}(w) = \emptyset$. Then, as $W' = valuations(\mathcal{S})$, $\mathcal{M}'$ is a model of $\mathcal{S}$. We have that $\mathcal{M}'$ is a model of $\mathcal{E}$, too: it is a model of $\mathcal{E}^a$, and given $a' \neq a$, for every $\varphi_i \to [a']\psi_i \in \mathcal{E}$ and every $w \in W'$, if $\models_w^{\mathcal{M}'} \varphi_i$, then $\models_{w'}^{\mathcal{M}'} \psi_i$ for all $w' \in W'$ such that $wR_{a'}w'$. Clearly $\mathcal{M}'$ is also a model of $\mathcal{I}$: it is a model of $\mathcal{I}^a$, and given $a' \neq a$, for every $\varphi_i \to [a']\bot \in \mathcal{I}$ and every $w \in W'$, if $\models_w^{\mathcal{M}'} \varphi_i$, then $\mathcal{I}^{a'}(w) \neq \emptyset$ and $R_{a'}(w) = \emptyset$. $\mathcal{M}'$ is a model of $\mathcal{X}$, too: besides being a model of $\mathcal{X}^a$, for every $a' \neq a$ and all worlds $w \in W'$ such that $\mathcal{X}^{a'}(w) \neq \emptyset$ there is a world accessible by $R_{a'}$, because $R_{a'}(w) = \emptyset$ in this case would preclude $\mathcal{X}^{a'}(w) \neq \emptyset$, and otherwise $w \notin W'$, which is impossible as long as **PS*** is satisfied. Hence $\models^{\mathcal{M}'} \mathcal{S} \wedge \mathcal{E} \wedge \mathcal{X} \wedge \mathcal{I}$. Because there is $v \in W'$ such that $\models_v^{\mathcal{M}'} \varphi$ and $\not\models_v^{\mathcal{M}'} \langle a\rangle\psi$, we have $\mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \not\models_{\leadsto} \varphi \to \langle a\rangle\psi$. □

Lemma 56. *If $\mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \models_{\leadsto} \varphi \to \langle a_1; \ldots; a_n\rangle\psi$, then there is $\varphi' \in \mathfrak{Fml}$ such that $\mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \models_{\leadsto} \varphi \to \langle a_1; \ldots; a_{n-1}\rangle\varphi'$ and $\mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \models_{\leadsto} \varphi' \to \langle a_n\rangle\psi$.*

Proof. The proof is by induction on the number of action operators.

Base: $n = 2$. Suppose $\mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \models_{\leadsto} \varphi \to \langle a_1; a_2\rangle\psi$. Then $\mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \models_{\leadsto} \varphi \to \langle a_1\rangle\langle a_2\rangle\psi$. For every model $\mathcal{M} = \langle W, R\rangle$ of $\langle \mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I}\rangle$ and for every $w \in W$ such that $\models_w^{\mathcal{M}} \varphi$, there is $w' \in W$ such that $wR_{a_1}w'$ and $\models_{w'}^{\mathcal{M}} \langle a_2\rangle\psi$. Let φ' be $\bigwedge\{\ell : \ell \in w'\}$ and the result follows.

Induction hypothesis: for any $k < n$, if $\mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \models_{\leadsto} \varphi \to \langle a_1; \ldots; a_k\rangle\psi$, then there is $\varphi' \in \mathfrak{Fml}$ such that $\mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \models_{\leadsto} \varphi \to \langle a_1; \ldots; a_{k-1}\rangle\varphi'$ and $\mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \models_{\leadsto} \varphi' \to \langle a_k\rangle\psi$.

Step: let $\mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \models_{\leadsto} \varphi \to \langle a_1; \ldots; a_n\rangle\psi$. Then we have that $\mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \models_{\leadsto} \varphi \to \langle a_1; \ldots; a_{n-1}\rangle\top$. By the induction hypothesis, there is $\varphi' \in \mathfrak{Fml}$ such that $\mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \models_{\leadsto} \varphi \to \langle a_1; \ldots; a_{n-2}\rangle\varphi'$ and $\mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \models_{\leadsto} \varphi' \to \langle a_{n-1}\rangle\top$. Because $\mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \models_{\leadsto} \varphi \to \langle a_1; \ldots; a_n\rangle\psi$, given a model $\mathcal{M} = \langle W, R\rangle$ of $\langle \mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I}\rangle$ such that $\models_w^{\mathcal{M}} \varphi$ for some $w \in W$, there must be $w'_{n-2} \in W$ such that $\models_{w'_{n-2}}^{\mathcal{M}} \langle a_{n-1}\rangle\langle a_n\rangle\psi$. Then we can safely take φ' as $\bigwedge\{\ell\colon \ell \in w'_{n-2}\}$. Now, $\mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \models_{\leadsto} \varphi' \to \langle a_{n-1}\rangle\langle a_n\rangle\psi$. By the base step, there is $\varphi'' \in \mathfrak{Fml}$ such that $\mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \models_{\leadsto} \varphi' \to \langle a_{n-1}\rangle\varphi''$ and $\mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \models_{\leadsto} \varphi'' \to \langle a_n\rangle\psi$. Putting all the results together, we get $\mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \models_{\leadsto} \varphi \to \langle a_1; \ldots; a_{n-1}\rangle\varphi''$ and $\mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \models_{\leadsto} \varphi'' \to \langle a_n\rangle\psi$, for some $\varphi'' \in \mathfrak{Fml}$. □

Proof of Theorem 45. ($\Rightarrow$): The proof is by induction on the number of action operators.

Base: $n = 1$. As $\langle \mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I}\rangle$ satisfies Postulate **PS***, the result follows from Lemma 55.

Induction hypothesis: for any $k < n$, if $\mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \models_{\leadsto} \varphi \to \langle a_1; \ldots; a_k\rangle\psi$, then $\mathcal{S}, \mathcal{E}^{a_1,\ldots,a_k}, \mathcal{X}^{a_1,\ldots,a_k}, \mathcal{I}^{a_1,\ldots,a_k} \models_{\leadsto} \varphi \to \langle a_1; \ldots; a_k\rangle\psi$.

Step: let $\mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \models_{\leadsto} \varphi \to \langle a_1; \ldots; a_n\rangle\psi$. By Lemma 56, there is $\varphi' \in \mathfrak{Fml}$ such that $\mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \models_{\leadsto} \varphi \to \langle a_1; \ldots; a_{n-1}\rangle\varphi'$ and $\mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \models_{\leadsto} \varphi' \to \langle a_n\rangle\psi$. By the induction hypothesis, we have $\mathcal{S}, \mathcal{E}^{a_1,\ldots,a_{n-1}}, \mathcal{X}^{a_1,\ldots,a_{n-1}}, \mathcal{I}^{a_1,\ldots,a_{n-1}} \models_{\leadsto} \varphi \to \langle a_1; \ldots; a_{n-1}\rangle\varphi'$ and also $\mathcal{S}, \mathcal{E}^{a_n}, \mathcal{X}^{a_n}, \mathcal{I}^{a_n} \models_{\leadsto} \varphi' \to \langle a_n\rangle\psi$. Then, this gives us $\mathcal{S}, \mathcal{E}^{a_1,\ldots,a_n}, \mathcal{X}^{a_1,\ldots,a_n}, \mathcal{I}^{a_1,\ldots,a_n} \models_{\leadsto} \varphi \to \langle a_1; \ldots; a_n\rangle\psi$.

($\Leftarrow$): Suppose $\mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \not\models_{\leadsto} \varphi \to \langle a_1; \ldots; a_n\rangle\psi$. Then there is a $\leadsto$-model $\mathcal{M}$ such that $\models^{\mathcal{M}} \mathcal{S} \wedge \mathcal{E} \wedge \mathcal{X} \wedge \mathcal{I}$ and $\not\models^{\mathcal{M}} \varphi \to \langle a_1; \ldots; a_n\rangle\psi$. Then, given $a_1, \ldots, a_n$, $\models^{\mathcal{M}} \mathcal{S} \wedge \mathcal{E}^{a_1,\ldots,a_n} \wedge \mathcal{X}^{a_1,\ldots,a_n} \wedge \mathcal{I}^{a_1,\ldots,a_n}$, and hence $\mathcal{S}, \mathcal{E}^{a_1,\ldots,a_n}, \mathcal{X}^{a_1,\ldots,a_n}, \mathcal{I}^{a_1,\ldots,a_n} \not\models_{\leadsto} \varphi \to \langle a_1; \ldots; a_n\rangle\psi$. □

References

[1] G. De Giacomo, M. Lenzerini, PDL-based framework for reasoning about actions, in: M. Gori, G. Soda (Eds.), Proc. 4th Congress of the Italian Association for Artificial Intelligence (IA*AI'95), in: Lecture Notes Artif. Intell., vol. 992, Springer-Verlag, 1995, pp. 103–114.

[2] F. Lin, Embracing causality in specifying the indirect effects of actions, in: Mellish [62], pp. 1985–1991.

[3] N. McCain, H. Turner, A causal theory of ramifications and qualifications, in: Mellish [62], pp. 1978–1984.

[4] M. Thielscher, Computing ramifications by postprocessing, in: Mellish [62], pp. 1994–2000.

[5] M. Castilho, O. Gasquet, A. Herzig, Formalizing action and change in modal logic I: The frame problem, Journal of Logic and Computation 9 (5) (1999) 701–735.

[6] D. Zhang, N. Foo, EPDL: A logic for causal reasoning, in: B. Nebel (Ed.), Proc. 17th Intl. Joint Conf. on Artificial Intelligence (IJCAI'01), Seattle, Morgan Kaufmann Publishers, 2001, pp. 131–138.

[7] D. Harel, Dynamic logic, in: D. Gabbay, F. Günthner (Eds.), Handbook of Philosophical Logic, vol. II, D. Reidel, Dordrecht, 1984, pp. 497–604.

[8] J. McCarthy, P. Hayes, Some philosophical problems from the standpoint of artificial intelligence, in: B. Meltzer, D. Mitchie (Eds.), Machine Intelligence, vol. 4, Edinburgh University Press, 1969, pp. 463–502.

[9] M. Gelfond, V. Lifschitz, Representing action and change by logic programs, Journal of Logic Programming 17 (2/3–4) (1993) 301–321.

[10] N. Kartha, V. Lifschitz, Actions with indirect effects (preliminary report), in: J. Doyle, E. Sandewall, P. Torasso (Eds.), Proc. 4th Intl. Conf. on Knowledge Representation and Reasoning (KR'94), Bonn, Morgan Kaufmann Publishers, 1994, pp. 341–350.

[11] E. Giunchiglia, G. Kartha, V. Lifschitz, Representing action: Indeterminacy and ramifications, Artificial Intelligence 95 (2) (1997) 409–438.

[12] A. Herzig, I. Varzinczak, Cohesion, coupling and the meta-theory of actions, in: Kaelbling and Saffiotti [60], pp. 442–447.

[13] A. Herzig, I. Varzinczak, A modularity approach for a fragment of $\mathcal{ALC}$, in: M. Fisher, W. van der Hoek, B. Konev, A. Lisitsa (Eds.), Proc. 10th Eur. Conf. on Logics in Artificial Intelligence (JELIA'2006), in: Lecture Notes Artif. Intell., vol. 4160, Springer-Verlag, 2006, pp. 216–228.

[14] F. Baader, D. Calvanese, D. McGuinness, D. Nardi, P. Patel-Schneider (Eds.), Description Logic Handbook, Cambridge University Press, 2003.

[15] A. Herzig, I. Varzinczak, Domain descriptions should be modular, in: R. López de Mántaras, L. Saitta (Eds.), Proc. 16th Eur. Conf. on Artificial Intelligence (ECAI'04), Valencia, IOS Press, 2004, pp. 348–352.

[16] D. Harel, J. Tiuryn, D. Kozen, Dynamic Logic, MIT Press, Cambridge, MA, 2000.

[17] M. Kracht, F. Wolter, Properties of independently axiomatizable bimodal logics, Journal of Symbolic Logic 56 (4) (1991) 1469–1485.

[18] M. Kracht, F. Wolter, Simulation and transfer results in modal logic: A survey, Studia Logica 59 (1997) 149–177.

[19] S. Popkorn, First Steps in Modal Logic, Cambridge University Press, 1994.

[20] P. Blackburn, M. de Rijke, Y. Venema, Modal Logic, Cambridge Tracts in Theoretical Computer Science, Cambridge University Press, 2001.

[21] M. Fitting, Proof Methods for Modal and Intuitionistic Logics, D. Reidel, Dordrecht, 1983.

[22] S. Hanks, D. McDermott, Default reasoning, nonmonotonic logics, and the frame problem, in: T. Kehler, S. Rosenschein (Eds.), Proc. 5th Natl. Conf. on Artificial Intelligence (AAAI'86), Philadelphia, Morgan Kaufmann Publishers, 1986, pp. 328–333.

[23] M. Castilho, A. Herzig, I. Varzinczak, It depends on the context! a decidable logic of actions and plans based on a ternary dependence relation, in: S. Benferhat, E. Giunchiglia (Eds.), Workshop on Nonmonotonic Reasoning (NMR'02), Toulouse, 2002, pp. 343–348.

[24] J. McCarthy, Epistemological problems of artificial intelligence, in: N. Sridharan (Ed.), Proc. 5th Intl. Joint Conf. on Artificial Intelligence (IJCAI'77), Cambridge, MA, Morgan Kaufmann Publishers, 1977, pp. 1038–1044.

[25] M. Ginsberg, D. Smith, Reasoning about actions II: The qualification problem, Artificial Intelligence 35 (3) (1988) 311–342.

[26] L. Schubert, Monotonic solution of the frame problem in the situation calculus: An efficient method for worlds with fully specified actions, in: H. Kyberg, R. Loui, G. Carlson (Eds.), Knowledge Representation and Defeasible Reasoning, Kluwer Academic Publishers, 1990, pp. 23–67.

[27] P. Doherty, W. Łukaszewicz, A. Szałas, Explaining explanation closure, in: Proc. 9th Intl. Symposium on Methodologies for Intelligent Systems, Zakopane, Poland, in: Lecture Notes Comput. Sci., vol. 1079, Springer-Verlag, 1996.

[28] J. Finger, Exploiting constraints in design synthesis, Ph.D. thesis, Stanford University, Stanford, 1987.

[29] R. Demolombe, A. Herzig, I. Varzinczak, Regression in modal logic, Journal of Applied Non-Classical Logics (JANCL) 13 (2) (2003) 165–185.

[30] 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 John McCarthy, Academic Press, San Diego, 1991, pp. 359–380.

[31] I. Varzinczak, What is a good domain description? Evaluating and revising action theories in dynamic logic, Ph.D. thesis, Université Paul Sabatier, Toulouse, 2006.

[32] R. Reiter, Knowledge in Action: Logical Foundations for Specifying and Implementing Dynamical Systems, MIT Press, Cambridge, MA, 2001.

[33] M. Castilho, O. Gasquet, A. Herzig, A dependence-based framework for actions with indeterminate and indirect effects, Tech. Rep. RT-98-04-R, Institut de recherche en informatique de Toulouse (IRIT), Université Paul Sabatier, http://www.irit.fr/LILaC/, February 1998.

[34] C. Schwind, Causality in action theories, Linköping Electronic Articles in Computer and Information Science 4 (4).

[35] P. Marquis, Consequence finding algorithms, in: D. Gabbay, P. Smets (Eds.), Algorithms for Defensible and Uncertain Reasoning, in: S. Moral, J. Kohlas (Eds.), Handbook of Defeasible Reasoning and Uncertainty Management Systems, vol. 5, Kluwer Academic Publishers, 2000, pp. 41–145, Chapter 2.

[36] K. Inoue, Linear resolution for consequence finding, Artificial Intelligence 56 (2–3) (1992) 301–353.

[37] A. Herzig, I. Varzinczak, On the modularity of theories, in: R. Schmidt, I. Pratt-Hartmann, M. Reynolds, H. Wansing (Eds.), Advances in Modal Logic, vol. 5, King's College Publications, 2005, pp. 93–109, selected papers of AiML 2004, also available at http://www.aiml.net/volumes/volume5.
[38] F. Lin, Embracing causality in specifying the indeterminate effects of actions, in: Shrobe and Senator [61], pp. 670–676.
[39] M. Thielscher, Ramification and causality, Artificial Intelligence 89 (1–2) (1997) 317–364.
[40] V. Lifschitz, W. Ren, Towards a modular action description language, in: Proc. 21st Natl. Conf. on Artificial Intelligence (AAAI'2006), Boston, AAAI Press/MIT Press, 2006.
[41] F. Pirri, R. Reiter, Some contributions to the metatheory of the situation calculus, Journal of the ACM 46 (3) (1999) 325–361.
[42] E. Amir, (De)composition of situation calculus theories, in: Proc. 17th Natl. Conf. on Artificial Intelligence (AAAI'2000), Austin, AAAI Press/MIT Press, 2000, pp. 456–463.
[43] F. Lin, R. Reiter, State constraints revisited, Journal of Logic and Computation 4 (5) (1994) 655–678.
[44] S. McIlraith, Integrating actions and state constraints: A closed-form solution to the ramification problem (sometimes), Artificial Intelligence 116 (1–2) (2000) 87–121.
[45] I. Sommerville, Software Engineering, Addison Wesley, 1985.
[46] R. Pressman, Software Engineering: A Practitioner's Approach, McGraw-Hill, 1992.
[47] D. Zhang, S. Chopra, N. Foo, Consistency of action descriptions, in: M. Ishizuka, A. Sattar (Eds.), Proc. 7th Pacific Rim Intl. Conf. on Artificial Intelligence: Trends in Artificial Intelligence, in: Lecture Notes Comput. Sci., vol. 2417, Springer-Verlag, 2002, pp. 70–79.
[48] J. Lang, F. Lin, P. Marquis, Causal theories of action—A computational core, in: V. Sorge, S. Colton, M. Fisher, J. Gow (Eds.), Proc. 18th Intl. Joint Conf. on Artificial Intelligence (IJCAI'03), Acapulco, Morgan Kaufmann Publishers, 2003, pp. 1073–1078.
[49] L. Cholvy, Checking regulation consistency by using SOL-resolution, in: Proc. 7th Intl. Conf. on AI and Law, Oslo, 1999, pp. 73–79.
[50] E. Giunchiglia, J. Lee, V. Lifschitz, N. McCain, H. Turner, Nonmonotonic causal theories, Artificial Intelligence 153 (1–2) (2004) 49–104.
[51] R. Li, L. Pereira, What is believed is what is explained, in: Shrobe and Senator [61], pp. 550–555.
[52] P. Liberatore, A framework for belief update, in: Proc. 7th Eur. Conf. on Logics in Artificial Intelligence (JELIA'2000), 2000, pp. 361–375.
[53] T. Eiter, E. Erdem, M. Fink, J. Senko, Updating action domain descriptions, in: Kaelbling and Saffiotti [60], pp. 418–423.
[54] A. Herzig, L. Perrussel, I. Varzinczak, Elaborating domain descriptions, in: G. Brewka, S. Coradeschi, A. Perini, P. Traverso (Eds.), Proc. 17th Eur. Conf. on Artificial Intelligence (ECAI'06), Riva del Garda, IOS Press, 2006, pp. 397–401.
[55] J. Garson, Modularity and relevant logic, Notre Dame Journal of Formal Logic 30 (2) (1989) 207–223.
[56] B. Cuenca Grau, B. Parsia, E. Sirin, A. Kalyanpur, Modularity and web ontologies, in: P. Doherty, J. Mylopoulos, C. Welty (Eds.), Proc. 10th Intl. Conf. on Knowledge Representation and Reasoning (KR'2006), Lake District, Morgan Kaufmann Publishers, 2006, pp. 198–208.
[57] A. Kakas, L. Michael, R. Miller, $\mathcal{M}$odular-$\mathcal{E}$: An elaboration tolerant approach to the ramification and qualification problems, in: C. Baral, G. Greco, N. Leone, G. Terracina (Eds.), Proc. 8th Intl. Conf. Logic Programming and Nonmonotonic Reasoning, Diamante, Springer-Verlag, 2005, pp. 211–226.
[58] J. Gustafsson, P. Doherty, Embracing occlusion in specifying the indirect effects of actions, in: L. Aiello, J. Doyle, S. Shapiro (Eds.), Proc. 5th Intl. Conf. on Knowledge Representation and Reasoning (KR'96), Cambridge, MA, Morgan Kaufmann Publishers, 1996, pp. 87–98.
[59] P. Marquis, Knowledge compilation using theory prime implicates, in: Mellish [62], pp. 837–843.
[60] L. Kaelbling, A. Saffiotti (Eds.), Proc. 19th Intl. Joint Conf. on Artificial Intelligence (IJCAI'05), Edinburgh, Morgan Kaufmann Publishers, 2005.
[61] H. Shrobe, T. Senator (Eds.), Proc. 13th Natl. Conf. on Artificial Intelligence (AAAI'96), Portland, AAAI Press/MIT Press, 1996.
[62] C. Mellish (Ed.), Proc. 14th Intl. Joint Conf. on Artificial Intelligence (IJCAI'95), Montreal, Morgan Kaufmann Publishers, 1995.