

Metatheory of actions: beyond consistency

Andreas Herzig Ivan Varzinczak

IRIT – 118 route de Narbonne

31062 Toulouse Cedex – France

e-mail: {herzig, ivan}@irit.fr

<http://www.irit.fr/recherches/LILAC>

June 28, 2018

Abstract

Consistency check has been the only criterion for theory evaluation 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 in reasoning about actions should have. We state some metatheoretical postulates concerning this sore spot. When all postulates are satisfied together we have a modular action theory. 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.

Contents

1	Introduction	3
2	Preliminaries	4
2.1	Dynamic logic	4
2.2	Describing action theories in PDL	6
2.2.1	Static laws	6
2.2.2	Effect laws	7
2.2.3	Inexecutability laws	8
2.2.4	Executability laws	8
2.2.5	Action theories	9
2.3	Dynamic logic and the frame problem	10
3	Postulates	12
4	No implicit static laws	14
5	No implicit inexecutability laws	19
6	Generalizing the postulates	23
7	Disturbing modularity	26
7.1	Postulates about effects of actions	26
7.2	Maximizing executabilities	27
8	Exploiting modularity	28
9	Related work	30
10	Conclusion	35
A	Proof of Theorem 4.3	41
B	Proof of Theorem 5.2	44

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. We call such theories *action theories*.

A priori satisfiability is the only criterion that formal logic provides to check the quality of such descriptions. In this 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 that in reasoning about actions one usually distinguishes several kinds of logical formulas. Among these are effect axioms, precondition axioms, and domain constraints. 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 \perp , and call them inexecutability laws.

Given these types of laws, suppose the language is powerful enough to state 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 this is not the case, then unexpected conclusions might follow from the theory \mathcal{T} , even in the case \mathcal{T} is 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 influence the laws for actions, but the dynamic part of a theory should not influence the non-dynamic one. It will turn out that in all existing accounts allowing for these four kinds of laws [31, 34, 44, 3, 47], consistent action theories can be written that violate this requirement. We here give algorithms that allow one to check whether an action theory satisfies the postulates we state. 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) [15], all we shall say applies as well to first-order formalisms, in particular to the Situation Calculus [36]. 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} [10, 24, 12] and others, and for Situation Calculus based approaches. In [19] we have given a Situation Calculus version of our analysis.

This work is organized as follows: after some background definitions (Sec-

tion 2) we state (Section 3) some postulates concerning action descriptions. 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 (Section 6) and discuss (Section 7) possible strengthenings of our set of postulates, and show interesting results that their satisfaction gives us (Section 8). Finally, 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 the ontology of dynamic domains. As our base formalism we use PDL. For more details, see [15, 16].

Let $\mathfrak{Act} = \{a_1, a_2, \dots\}$ be the set of all *atomic action constants* of a given domain. Examples of atomic actions are *load* and *shoot*. We use a as a variable standing for a particular atomic action. 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 [25, 26]).

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

We suppose both \mathfrak{Act} and \mathfrak{Prop} are finite.

We use small Greek letters φ, ψ, \dots to denote *classical formulas*. They are recursively defined in the following way:

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

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

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

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

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

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

We denote complex formulas (with modal operators) by capital Greek letters Φ_1, Φ_2, \dots . 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 as $\langle a \rangle \Phi =_{\text{Def}} \neg[a]\neg\Phi$. 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* \rightarrow *[shoot]alive* and *[load]loaded*.

For parsimony's sake, whenever there is no confusion we identify a set of formulas with the conjunction of the formulas it is made of. The semantics we take into account here is that for multimodal K [39, 2].

Definition 2.1 A PDL-model is a triple $\mathcal{M} = \langle W, R, V \rangle$ where W is a nonempty set of possible worlds (alias possible states), $R: \mathbf{Act} \rightarrow 2^{W \times W}$ maps action constants a to accessibility relations $R_a \subseteq W \times W$, and $V: \mathbf{Prop} \rightarrow 2^W$ maps propositional constants to subsets of W .

Definition 2.2 Given a PDL-model $\mathcal{M} = \langle W, R, V \rangle$, the satisfaction relation is defined as the smallest relation satisfying:

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

Definition 2.3 A PDL-model \mathcal{M} is a model of Φ (noted $\models^{\mathcal{M}} \Phi$) if and only if for all $w \in W$, $\models_w^{\mathcal{M}} \Phi$. \mathcal{M} is a model of a set of formulas \mathcal{T} (noted $\models^{\mathcal{M}} \mathcal{T}$) if and only if $\models_w^{\mathcal{M}} \Phi$ for every $\Phi \in \mathcal{T}$.

Definition 2.4 A formula Φ is a consequence of the set of global axioms $\{\Phi_1, \dots, \Phi_n\}$ in the class of all PDL-models (noted $\{\Phi_1, \dots, \Phi_n\} \models_{\text{PDL}} \Phi$) if and only if for every PDL-model \mathcal{M} , if $\models^{\mathcal{M}} \Phi_i$ for every Φ_i , then $\models^{\mathcal{M}} \Phi$.¹

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.

¹In [3] local consequence is considered. For that reason a further modal operator \square had to be introduced, resulting in a logic which is multimodal K plus monomodal S4 for \square , and where axiom schema $\square\Phi \rightarrow [a]\Phi$ holds.

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 we state 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 2.5 A *domain* is a tuple $\langle \mathfrak{A}ct, \mathfrak{P}rop \rangle$.

An example of a domain is the well-known Yale Shooting Scenario [14], whose actions are *load*, *wait* and *shoot*, and whose fluents are *loaded* and *alive*.

Given a domain, we are interested in theories whose statements describe the behavior of actions on the considered fluents. 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 2.6 A *static law*² is a formula $\varphi \in \mathfrak{Fml}$ that is classically consistent.

An example of a static law is $walking \rightarrow alive$, saying that if a turkey is walking, then it must be alive [44]. 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 [4].

²Static laws are often called *domain constraints* or *integrity constraints*. Because the different laws for actions that we shall introduce in the sequel could in principle also be called like that, we avoid these terms.

In action languages such as \mathcal{A} and \mathcal{AR} we would write the statement *alive* if *walking*, and in the Situation Calculus it would be the first-order formula

$$\forall s(Holds(walking, s) \rightarrow Holds(alive, s)).$$

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 $[a]\varphi$ expresses that φ is true after every possible execution of a .

Definition 2.7 An *effect law*³ for action a is of the form $\varphi \rightarrow [a]\psi$, where $\varphi, \psi \in \mathfrak{Fml}$, with φ and ψ both 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* \rightarrow *[shoot]¬alive*, saying that whenever the gun is loaded, after shooting the turkey is dead. Another one is $\top \rightarrow$ *[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 requirements for φ and ψ make sense: if φ is inconsistent then the effect law is superfluous; 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.

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

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

³Effect 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.

2.2.3 Inexecutability laws

We consider effect laws whose consequent ψ is inconsistent as a particular kind of law which we call inexecutability laws. (Such laws are sometimes called qualifications [35].) 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 2.8 An *inexecutability law for action a* is of the form $\varphi \rightarrow [a]\perp$, where $\varphi \in \mathfrak{Fml}$ is classically consistent.

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

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

$$\forall s(\neg\text{Holds}(\text{hasGun}, s) \rightarrow \neg\text{Poss}(\text{shoot}, s)).$$

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 the conditions under which an action is guaranteed to be executable.

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 “the execution of action a is possible”.

Definition 2.9 An *executability law⁴ for action a* is of the form $\varphi \rightarrow \langle a \rangle \top$, where $\varphi \in \mathfrak{Fml}$ is classically consistent.

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

In action languages such laws are not represented. In Situation Calculus our example would be stated as

$$\forall s(\text{Holds}(\text{hasGun}, s) \rightarrow \text{Poss}(\text{shoot}, s)).$$

⁴Some approaches (most prominently Reiter’s) use biconditionals $\varphi \leftrightarrow \langle a \rangle \top$, called precondition axioms. This is equivalent to $\neg\varphi \leftrightarrow [a]\perp$, highlighting that they merge information about inexecutability with information about executability. In this work we consider these entities different and keep them separated.

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 [11, 31, 44]), the status of executability laws is less consensual. Some authors [43, 7, 34, 44] more or less tacitly consider that executability laws should not be made explicit but rather inferred by the reasoning mechanism. Others [31, 47] 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. 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!⁵ 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 2.10 An *action theory* for a is a tuple $\mathcal{T}^a = \langle \mathcal{S}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a \rangle$.

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

$$\mathcal{S} = \{walking \rightarrow alive\}, \quad \mathcal{E}^{shoot} = \{loaded \rightarrow [shoot]\neg alive\},$$

$$\mathcal{X}^{shoot} = \{hasGun \rightarrow \langle shoot \rangle \top\}, \quad \mathcal{I}^{shoot} = \{\neg hasGun \rightarrow [shoot] \perp\}$$

Given a dynamic domain we define $\mathcal{E} = \bigcup_{a \in \mathfrak{Act}} \mathcal{E}^a$, $\mathcal{X} = \bigcup_{a \in \mathfrak{Act}} \mathcal{X}^a$, and $\mathcal{I} = \bigcup_{a \in \mathfrak{Act}} \mathcal{I}^a$. 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 2.11 An *action theory* \mathcal{T} is a tuple of the form $\langle \mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \rangle$.

For parsimony's sake, whenever there is no confusion we write $\mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \models_{\text{PDL}} \Phi$ instead of $\mathcal{S} \cup \mathcal{E} \cup \mathcal{X} \cup \mathcal{I} \models_{\text{PDL}} \Phi$.

⁵Of course this would require a solution to the qualification problem [35].

When performing the task of formalizing dynamic domains, we face the *frame problem* [36] and the *ramification problem* [9]. In what follows we formally present the logic of actions in which action theories will henceforth be described.

2.3 Dynamic logic and the frame problem

As it was 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

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

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,

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

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 constraints makes that this is a delicate task, and starting with [31, 34], several authors have argued that some notion of causality is needed. We here opt for the dependence based approach presented in [3], which has been shown in [6] to subsume Reiter's solution to the frame problem [41], and moreover at least partially accounts for the ramification problem.

In the logical framework developed in [3], metalogical information, given in the form of a dependence relation, is added to PDL.

Definition 2.12 (Dependence relation [3]) A *dependence relation* is a binary relation $\rightsquigarrow \subseteq \mathfrak{Act} \times \mathfrak{Lit}$.

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

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

which means that action *shoot* may make the literals $\neg\text{loaded}$, $\neg\text{alive}$ and $\neg\text{walking}$ true, and action *tease* may make *walking* true.

Semantically, the dependence-based approach relies on the explanation closure assumption [43]. The reasoning behind its solution to the frame problem consists in a kind of negation as failure: Because $\langle \text{load}, \neg \text{hasGun} \rangle \notin \rightsquigarrow$, we have $\text{load} \not\rightsquigarrow \neg \text{hasGun}$, i.e., $\neg \text{hasGun}$ 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 $\text{tease} \not\rightsquigarrow \text{alive}$ and $\text{tease} \not\rightsquigarrow \neg \text{alive}$. The meaning of all these independences is that the frame axioms $\text{hasGun} \rightarrow [\text{load}]\text{hasGun}$, $\neg \text{alive} \rightarrow [\text{tease}]\neg \text{alive}$ and $\text{alive} \rightarrow [\text{tease}]\text{alive}$ hold.

We assume \rightsquigarrow is finite.

A dependence relation \rightsquigarrow defines a class of possible worlds models $\mathcal{M}_{\rightsquigarrow}$.

Definition 2.13 Given a \rightsquigarrow -model $\mathcal{M} = \langle W, R, V \rangle$, the satisfaction relation is defined as the smallest relation satisfying:

- all the truth conditions of Definition 2.2.
- whenever $wR_a w'$ then:
 - $\not\models_w^{\mathcal{M}} p$ implies $\not\models_{w'}^{\mathcal{M}} p$, if $a \not\rightsquigarrow p$;
 - $\models_w^{\mathcal{M}} p$ implies $\models_{w'}^{\mathcal{M}} p$, if $a \not\rightsquigarrow \neg p$.

Given $\mathcal{M} \in \mathcal{M}_{\rightsquigarrow}$, Φ and \mathcal{T} , $\models^{\mathcal{M}} \Phi$ and $\models^{\mathcal{M}} \mathcal{T}$ are defined as in Definition 2.3.

Definition 2.14 A formula Φ is a \rightsquigarrow -based consequence of $\{\Phi_1, \dots, \Phi_n\}$ in the class of all \rightsquigarrow -models (noted $\{\Phi_1, \dots, \Phi_n\} \models_{\rightsquigarrow} \Phi$) if and only if for every \rightsquigarrow -model \mathcal{M} , if $\models^{\mathcal{M}} \Phi_i$ for every Φ_i , then $\models^{\mathcal{M}} \Phi$.

In our example it thus holds

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

and

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

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 $\text{loaded} \rightarrow [\text{shoot}]\neg \text{walking}$ can be deduced with $\models_{\rightsquigarrow}$ without explicitly stating that *in the set of effect laws for shoot*, we still have to state *indirect dependences* such as $\text{shoot} \rightsquigarrow \neg \text{walking}$. Nevertheless, 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” [42].

We comply with that as we can define a semi-automatic procedure for generating the dependence relation from the set of effect laws. Moreover, as it has been argued in [4, 18], 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 2.15 Let $\langle \mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \rangle$ be an action theory and \rightsquigarrow its associated dependence relation. Then $\mathcal{M} = \langle W, R, V \rangle$ is the *big* (alias *maximal/standard*) model for $\langle \mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \rangle$ and \rightsquigarrow if and only if:

- \mathcal{M} is a \rightsquigarrow -model;
- $W = val(\mathcal{S})$ (all valuations of \mathcal{S});
- $R_a = \{(w, w') : \forall \varphi \rightarrow [a]\psi \in \mathcal{E} \cup \mathcal{I}, \text{ if } \models_w^{\mathcal{M}} \varphi, \text{ then } \models_{w'}^{\mathcal{M}} \psi\}$.

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

3 Postulates

“When does a given action theory have a model?”, and, more importantly, “is that model what we really expect from it?” 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 those questions.

We do not attempt here to provide a ‘magical’ method for making an action theory intuitive. Instead, what we are going to do in what follows is to provide some guidelines that help detecting unintuitive consequences of a theory and identifying its problematic part(s).

Our central thesis is that the different types of laws define in Section 2.2 should be neatly separated in different 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.

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

The theory of a given action should be logically consistent.

PS (No implicit static laws):

$$\text{if } \mathcal{S}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a \models_{\sim} \varphi, \text{ then } \mathcal{S} \models \varphi$$

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 consequence in \mathcal{M}_{\sim} , 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):

$$\text{if } \mathcal{S}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a \models_{\sim} \varphi \rightarrow [a]\perp, \text{ then } \mathcal{S}, \mathcal{I}^a \models_{\text{PDL}} \varphi \rightarrow [a]\perp$$

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_{PDL} instead of \models_{\sim} because we also suppose that neither frame axioms nor indirect effects should be relevant to derive inexecutability laws. The same remark holds for the postulates that follow.

PX (No implicit executability laws):

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

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 8.1).

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

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

and the dependence relation:

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

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

How can we find out whether an action theory for a satisfies Postulate **PS**?

Theorem 4.1 $\langle \mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \rangle$ and \rightsquigarrow satisfy Postulate **PS** if and only if the big model for $\langle \mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \rangle$ and \rightsquigarrow is a model of $\langle \mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \rangle$ and \rightsquigarrow .

Proof:

(\Rightarrow): Let $\mathcal{M} = \langle W, R, V \rangle$ be a big model of $\langle \mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \rangle$ and \rightsquigarrow , and suppose $\models^{\mathcal{M}} \mathcal{S} \wedge \mathcal{E} \wedge \mathcal{X} \wedge \mathcal{I}$ (\mathcal{M} is a model of $\mathcal{S} \cup \mathcal{E} \cup \mathcal{X} \cup \mathcal{I}$). Then $W = \text{val}(\mathcal{S})$, i.e., for all $\varphi \in \mathfrak{Fml}$ and all $w \in W$, if $\models_w^{\mathcal{M}} \varphi$, then there is a valuation v of \mathcal{S}

such that v makes φ true. From this it follows that if $\models_w^{\mathcal{M}} \varphi$ for all $w \in W$, then φ is true in all valuations of \mathcal{S} . Hence $\mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \models_{\sim} \varphi$ implies $\mathcal{S} \models \varphi$, and then $\langle \mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \rangle$ and \sim satisfy Postulate **PS**.

(\Leftarrow): Let $\mathcal{M} = \langle W, R, V \rangle$ be a big model of $\langle \mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \rangle$ and \sim . Suppose $\langle \mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \rangle$ and \sim do not satisfy Postulate **PS**. Then there must be $\varphi \in \mathfrak{Fml}$ such that $\mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \models_{\sim} \varphi$ and $\mathcal{S} \not\models \varphi$. This means that there is a valuation v of \mathcal{S} that falsifies φ . As $v \in W$ (because \mathcal{M} is a big model) then \mathcal{M} is not a model of $\langle \mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \rangle$ and \sim . ■

We shall give an algorithm to find a finite characterization of all⁶ 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 $\varphi \rightarrow \langle a \rangle \top$ in the theory, construct from $\mathcal{E}^a, \mathcal{I}^a$ and \sim a set of inexecutabilities $\{\varphi_1 \rightarrow [a] \perp, \dots, \varphi_n \rightarrow [a] \perp\}$ that potentially conflict with $\varphi \rightarrow \langle a \rangle \top$. For each i , $1 \leq i \leq n$, 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 $\neg(\varphi \wedge \varphi_i)$ that were caught to \mathcal{S}) until no implicit static law is obtained.

For an example of the execution of the algorithm, consider $\langle \mathcal{S}_1, \mathcal{E}_1^{tease}, \mathcal{X}_1^{tease}, \mathcal{I}_1^{tease} \rangle$ with \sim as above. For the action *tease*, we have the executability $\langle \text{tease} \rangle \top$. Now, from $\mathcal{E}_1^{tease}, \mathcal{I}_1^{tease}$ and \sim we try to build an inexecutability for *tease*. We take $[\text{tease}] \text{walking}$ and compute then all indirect effects of *tease* w.r.t. \mathcal{S}_1 . From *walking* $\rightarrow \text{alive}$, we get that *alive* is an indirect effect of *tease*, giving us $[\text{tease}] \text{alive}$. But $\langle \text{tease}, \text{alive} \rangle \notin \sim$, which means the frame axiom $\neg \text{alive} \rightarrow [\text{tease}] \neg \text{alive}$ holds. Together with $[\text{tease}] \text{alive}$, this gives us the inexecutability $\neg \text{alive} \rightarrow [\text{tease}] \perp$. As $\mathcal{S}_1 \cup \{\top, \neg \text{alive}\}$ is satisfiable (\top is the antecedent of the executability $\langle \text{tease} \rangle \top$), we get $\neg \text{alive} \rightarrow \perp$, 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.

Definition 4.1 Let $\varphi \in \mathfrak{Fml}$ and χ a clause. χ is an *implicate* of φ if and only if $\varphi \models \chi$.

In our running example, *alive* is an implicate of the set of formulas $\{\text{walking} \rightarrow \text{alive}, \text{walking}\}$.

Definition 4.2 Let $\varphi \in \mathfrak{Fml}$ and χ a clause. χ is a *prime implicate* of φ if and only if

⁶Actually what the algorithm does is to find an interpolant of all implicit static laws of the theory.

- χ is an implicate of φ , and
- for every implicate χ' of φ , $\chi' \models \chi$ implies $\chi \models \chi'$.

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

For example, the set of prime implicants 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 $\{\text{walking} \rightarrow \text{alive}, \text{walking}\}$. For more on prime implicants and their properties, see [33].

Definition 4.3 Let $\varphi, \psi \in \mathfrak{Fml}$. Then $NewCons_\varphi(\psi) = PI(\varphi \wedge \psi) \setminus PI(\varphi)$.

The function $NewCons_\varphi(\psi)$ 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. [21]). It is computed by subtracting the prime implicants of φ from those of $\varphi \wedge \psi$. For example, $NewCons_{p_1}((\neg p_1 \vee p_2) \wedge (\neg p_1 \vee p_3 \vee p_4)) = \{p_2, p_3 \vee p_4\}$. And for our scenario, $NewCons_{\text{walking} \rightarrow \text{alive}}(\text{walking}) = \{\text{alive}, \text{walking}\}$.

The algorithm below improves the one in [20] by integrating a solution to the frame problem (via the dependence relation \rightsquigarrow). As a matter of notation, 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 4.1 (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  $\langle \mathcal{S}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a \rangle$ 
 $\mathcal{S}_{imp^*} := \emptyset$ 
 $\mathcal{C}^a := \mathcal{E}^a \cup \mathcal{I}^a$ 
repeat
   $\mathcal{S}_{imp} := \emptyset$ 
  for all  $\varphi \rightarrow \langle a \rangle \top \in \mathcal{X}^a$  do
    for all  $\hat{\mathcal{C}}^a \subseteq \mathcal{C}^a$  such that  $\hat{\mathcal{C}}^a \neq \emptyset$  do
       $\varphi_{\hat{\mathcal{C}}^a} := \bigwedge \{\varphi_i : \varphi_i \rightarrow [a]\psi_i \in \hat{\mathcal{C}}^a\}$ 
       $\psi_{\hat{\mathcal{C}}^a} := \bigwedge \{\psi_i : \varphi_i \rightarrow [a]\psi_i \in \hat{\mathcal{C}}^a\}$ 
      for all  $\chi \in NewCons_{\mathcal{S}}(\psi_{\hat{\mathcal{C}}^a})$  do
        if  $\mathcal{S} \cup \mathcal{S}_{imp^*} \cup \{\varphi, \varphi_{\hat{\mathcal{C}}^a}, \neg \chi\} \not\models \perp$  and  $\forall l_i \in \chi, a \not\rightsquigarrow l_i$  then
           $\mathcal{S}_{imp} := \mathcal{S}_{imp} \cup \{\neg(\varphi \wedge \varphi_{\hat{\mathcal{C}}^a} \wedge \neg \chi)\}$ 
     $\mathcal{S}_{imp^*} := \mathcal{S}_{imp^*} \cup \mathcal{S}_{imp}$ 
  until  $\mathcal{S}_{imp} = \emptyset$ 

```

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 4.2 Algorithm 4.1 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

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

As \mathcal{X}^a and \mathcal{I}^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, 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 4.1 terminates. ■

This is the key algorithm of the paper. We are aware that it 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 $NewCons_{\mathcal{S}}(\psi_{\hat{\mathcal{C}}^a})$ 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 $NewCons_{\mathcal{S}}(\psi_{\hat{\mathcal{C}}^a})$ is more difficult to control.

Example 4.1 For $\langle \mathcal{S}_1, \mathcal{E}_1^{tease}, \mathcal{X}_1^{tease}, \mathcal{I}_1^{tease} \rangle$, Algorithm 4.1 returns $\mathcal{S}_{imp^*} = \{\text{alive}\}$.

Theorem 4.3 An action theory $\langle \mathcal{S}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a \rangle$ with \sim satisfies Postulate **PS** if and only if $\mathcal{S}_{imp^*} = \emptyset$.

Proof: See Appendix A. ■

Theorem 4.4 Let \mathcal{S}_{imp^*} be the output of Algorithm 4.1 on input $\langle \mathcal{S}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a \rangle$ and \sim . Then

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

Proof: Item 1. is straightforward from the termination of Algorithm 4.1 and Theorem 4.3. Item 2. follows from the fact that by the **if**-test in Algorithm 4.1, 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 original theory. ■

Corollary 4.1 For all $\varphi \in \mathfrak{Fml}$, $\mathcal{S}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a \models_{\sim} \varphi$ if and only if $\mathcal{S} \cup \mathcal{S}_{imp^*} \models \varphi$.

Proof: For the left-to-right direction, let $\varphi \in \mathfrak{Fml}$ be such that $\mathcal{S}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a \models_{\sim} \varphi$. Then $\mathcal{S} \cup \mathcal{S}_{imp^*}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a \models_{\sim} \varphi$, by monotonicity. By Theorem 4.4-1., $\langle \mathcal{S} \cup \mathcal{S}_{imp^*}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a \rangle$ has no implicit static law, hence $\mathcal{S} \cup \mathcal{S}_{imp^*} \models \varphi$. The right-to-left direction is straightforward by Theorem 4.4-2. ■

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 4.1, we wrongly assumed that *tease* is always executable. Thus one way of ‘repairing’ our theory would be to consider the weaker executability *alive* $\rightarrow \langle \text{tease} \rangle \top$ instead of $\langle \text{tease} \rangle \top$ in $\mathcal{X}^{\text{tease}}$.

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

Example 4.2 Consider $\mathcal{S} = \emptyset$, $\mathcal{E}^{shoot} = \{[\text{loaded}] \rightarrow [\text{shoot}] \neg \text{alive}\}$, $\mathcal{X}^{shoot} = \{[\text{hasGun}] \rightarrow \langle \text{shoot} \rangle \top\}$ and $\mathcal{I}^{shoot} = \{[\text{shoot}] \perp\}$, with \rightsquigarrow still as above. For this theory Algorithm 4.1 returns $\mathcal{S}_{imp^*} = \{\neg \text{hasGun}\}$.

In Example 4.2 we discovered that the agent never has a gun. The problem here can be overcome by weakening $[\text{shoot}] \perp$ in \mathcal{I}^{shoot} with $\neg \text{hasGun} \rightarrow [\text{shoot}] \perp$.⁷

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 4.1, if we replace the law $[\text{tease}] \text{walking}$ by the weaker $\text{alive} \rightarrow [\text{tease}] \text{walking}$, the resulting action theory would satisfy Postulate **PS**. In the same way, stating the (unintuitive) dependence $\text{tease} \rightsquigarrow \text{alive}$ (which means the frame axiom $\neg \text{alive} \rightarrow [\text{tease}] \neg \text{alive}$ is no longer valid) guarantees satisfaction of **PS**. (Note, however, that this solution becomes intuitive when *alive* is replaced by *awake*.)

⁷Regarding Examples 4.1 and 4.2, 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.

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

Example 4.3 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. $goLeft$ 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):

$$\mathcal{S} = \{at_i \rightarrow pos : 0 < i \leq maxInt\} \cup \{at_i \rightarrow \neg pos : minInt \leq i \leq 0\}$$

$$\mathcal{E} = \begin{cases} \{at_{minInt} \rightarrow [goLeft] underflow\} \cup \\ \{at_i \rightarrow [goLeft] at_{i-1} : i > minInt\}, \end{cases} \quad \mathcal{X} = \{\langle goLeft \rangle \top\}, \quad \mathcal{I} = \emptyset$$

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

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

Applying Algorithm 4.1 to this action theory gives us all the implicit static laws of the form $\neg(at_i \wedge at_j)$, $i \neq j$, i.e., we cannot be at two different numbers 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 4.1 $\mathcal{S} \cup \mathcal{S}_{imp^*}$ 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 $\mathcal{I}_2 = \emptyset$ (executabilities do not matter here), and let \rightsquigarrow be that for $\langle \mathcal{S}_1, \mathcal{E}_1, \mathcal{X}_1, \mathcal{I}_1 \rangle$. Note that $\langle \mathcal{S}_2, \mathcal{E}_2, \mathcal{X}_2, \mathcal{I}_2 \rangle$ satisfies 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: $\mathcal{S}_2, \mathcal{E}_2^{\text{tease}} \models_{\text{PDL}} [\text{tease}]\text{alive}$. Now as $\text{tease} \not\rightarrow \text{alive}$, the status of *alive* is not modified by *tease*, and we have $\mathcal{S}_2, \mathcal{E}_2^{\text{tease}} \models_{\sim} \neg\text{alive} \rightarrow [\text{tease}]\neg\text{alive}$. From the above, it follows

$$\mathcal{S}_2, \mathcal{E}_2^{\text{tease}}, \mathcal{X}_2^{\text{tease}}, \mathcal{I}_2^{\text{tease}} \models_{\sim} \neg\text{alive} \rightarrow [\text{tease}]\perp,$$

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

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

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

In the literature, such laws are also known as *implicit qualifications* [11], 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 [31, 45]. 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.⁸ 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 $(\varphi_1 \wedge \dots \wedge \varphi_n) \rightarrow [a](\psi_1 \wedge \dots \wedge \psi_n)$, with each $\varphi_i \rightarrow [a]\psi_i \in \mathcal{E}^a$, if $\varphi_1 \wedge \dots \wedge \varphi_n$ is consistent w.r.t. to \mathcal{S} , $\psi_1 \wedge \dots \wedge \psi_n$ inconsistent w.r.t. \mathcal{S} , and $\mathcal{S}, \mathcal{I}^a \not\models_{\text{PDL}} (\varphi_1 \wedge \dots \wedge \varphi_n) \rightarrow [a]\perp$, then output $(\varphi_1 \wedge \dots \wedge \varphi_n) \rightarrow [a]\perp$ 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 $\langle \mathcal{S}_2, \mathcal{E}_2^{\text{tease}}, \mathcal{X}_2^{\text{tease}}, \mathcal{I}_2^{\text{tease}} \rangle$ with \sim as given above. From $\mathcal{E}_2^{\text{tease}}$ we get $\top \rightarrow [\text{tease}]\text{walking}$, whose antecedent is consistent with \mathcal{S} . As $\models_{\sim} \neg\text{alive} \rightarrow [\text{tease}]\neg\text{alive}$ and $\mathcal{S} \cup \{\text{walking}\} \models \text{alive}$, and because $\mathcal{S}, \mathcal{I}_2^{\text{tease}} \not\models_{\text{PDL}} (\top \wedge \neg\text{alive}) \rightarrow [\text{tease}]\perp$, 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 needed in the input will be made clear in the sequel):

Algorithm 5.1 (Finding implicit inexecutability laws for a)

⁸Note 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.

input: $\langle \mathcal{S}, \mathcal{E}^a, \mathcal{I}^a \rangle$ and \sim
output: \mathcal{I}_{imp}^a , the set of implicit inexecutability laws for a

$$\mathcal{I}_{imp}^a := \emptyset$$

for all $\hat{\mathcal{E}}^a \subseteq \mathcal{E}^a$ **do**

$$\varphi_{\mathcal{E}^a} := \bigwedge \{\varphi_i : \varphi_i \rightarrow [a] \psi_i \in \hat{\mathcal{E}}^a\}$$

$$\psi_{\mathcal{E}^a} := \bigwedge \{\psi_i : \varphi_i \rightarrow [a] \psi_i \in \hat{\mathcal{E}}^a\}$$

for all $\chi \in NewCons_{\mathcal{S}}(\psi_{\mathcal{E}^a})$ **do**

if $\forall l_i \in \chi, a \not\sim l_i$ **and** $\mathcal{S}, \mathcal{I}^a \not\models (\varphi_{\mathcal{E}^a} \wedge \neg\chi) \rightarrow [a]\perp$ **then**

$$\mathcal{I}_{imp}^a := \mathcal{I}_{imp}^a \cup \{(\varphi_{\mathcal{E}^a} \wedge \neg\chi) \rightarrow [a]\perp\}$$

Theorem 5.1 Algorithm 5.1 terminates.

Proof: Straightforward, as we have assumed \mathcal{S} , \mathcal{E} , \mathcal{I} and \sim finite, and $NewCons$ is finite (because \mathcal{S} and $\psi_{\mathcal{E}^a}$ are finite). \blacksquare

Example 5.1 Consider $\mathcal{S}_2, \mathcal{E}_2^{tease}, \mathcal{I}_2^{tease}$ and \sim as given above. Then Algorithm 5.1 returns $\mathcal{I}_{imp}^{tease} = \{\neg alive \rightarrow [tease]\perp\}$.

Nevertheless, to apply Algorithm 5.1 is not enough to guarantee Postulate **PI**, as illustrated by the following example:

Example 5.2 (Incompleteness of Algorithm 5.1 without PS) Let $\mathcal{S} = \emptyset$, $\mathcal{E}^a = \{p_1 \rightarrow [a]p_2\}$, $\mathcal{X}^a = \{\langle a \rangle \top\}$, $\mathcal{I}^a = \{p_2 \rightarrow [a]\perp\}$, and $\sim = \emptyset$. Then we have $\mathcal{S}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a \models_{\sim} p_1 \rightarrow [a]\perp$, but after running Algorithm 5.1 on $\langle \mathcal{S}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a \rangle$ we have $\mathcal{S}, \mathcal{I}_{imp}^a \not\models_{PDL} p_1 \rightarrow [a]\perp$.

Example 5.2 shows that the presence of implicit static laws (induced by executabilities) implies the existence of implicit inexecutabilities that are not caught by Algorithm 5.1. One possibility of getting rid of this is by considering the weaker version of **PI**:

PI' (No implicit inexecutability laws – weak version):

if $\mathcal{S}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a \models_{\sim} \varphi \rightarrow [a]\perp$, and $\mathcal{S}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a \not\models_{\sim} \neg\varphi$,

then $\mathcal{S}, \mathcal{I}^a \models_{PDL} \varphi \rightarrow [a]\perp$

If a non-trivial inexecutability law for a given action a can be inferred from its respective theory, then it should be inferable in **PDL** from the static and inexecutability laws for it alone.

With an adaptation of Algorithm 5.1 to take \mathcal{X}^a in its input and support a test for satisfiability of an inexecutability's antecedent, we could guarantee completeness with respect to Postulate **PI'**. However such a test has the same

complexity of checking whether Postulate **PS** is satisfied. That is the reason we keep abide on **PI** and require $\langle \mathcal{S}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a \rangle$ to satisfy Postulate **PS** prior to running Algorithm 5.1. This gives us the following result:

Theorem 5.2 If $\langle \mathcal{S}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a \rangle$ with \rightsquigarrow satisfies Postulate **PS**, then it satisfies Postulate **PI** if and only if $\mathcal{I}_{imp}^a = \emptyset$.

Proof: See Appendix B. ■

With Algorithm 5.1, 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}_{imp}^a can be added to \mathcal{I}^a ; in the second one, \mathcal{I}_{imp}^a helps in properly changing \mathcal{E}^a or \rightsquigarrow . For instance, to correct the action theory of our example, the knowledge engineer would have the following options:

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

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 5.3 (Drinking coffee [19]) Suppose, for instance, a hypothetical situation in which we reason about the effects of drinking a cup of coffee:

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

and the dependence relation

$$\rightsquigarrow = \{\langle drink, happy \rangle, \langle drink, \neg happy \rangle\}$$

Observe that $\langle \mathcal{S}, \mathcal{E}^{drink}, \mathcal{X}^{drink}, \mathcal{I}^{drink} \rangle$ satisfies **PS**. Then, running Algorithm 5.1 on this action theory will give us $\mathcal{I}_{imp}^{drink} = \{(sugar \wedge salt) \rightarrow [drink]\perp\}$.

Remark 5.1 $\mathcal{I}^a \cup \mathcal{I}_{imp}^a$ is not always intuitive.

Whereas in Example 5.1 we have got an inexecutability that could be safely added to \mathcal{I}_2^{tease} , in Example 5.3 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 complex 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**): $\mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \not\models_{\sim} \perp$

The whole action theory should be logically consistent.

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

PS* (**No implicit static laws**):

if $\mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \models_{\sim} \varphi$, then $\mathcal{S} \models_{\text{PDL}} \varphi$

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 6.1 $\langle \mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \rangle$ satisfies **PS*** if and only if $\langle \mathcal{S}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a \rangle$ satisfies **PS** for all $a \in \mathfrak{Act}$.

Proof:

(\Rightarrow): Straightforward.

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

Theorem 6.2 If $\langle \mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \rangle$ satisfies **PS***, then $\langle \mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \rangle$ satisfies **PC*** if and only if $\langle \mathcal{S}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a \rangle$ satisfies **PC** for all $a \in \mathfrak{Act}$.

Proof: Straightforward as the underlying logic is independently axiomatized. ■

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

PI* (**No implicit inexexecutability laws**):

$$\text{if } \mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \models_{\sim} \varphi \rightarrow [a]\perp, \text{ then } \mathcal{S}, \mathcal{I} \models_{\text{PDL}} \varphi \rightarrow [a]\perp$$

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

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

Nevertheless, under **PS*** the result follows:

Theorem 6.3 Let $\langle \mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \rangle$ satisfy **PS***. $\langle \mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \rangle$ satisfies **PI*** if and only if $\langle \mathcal{S}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a \rangle$ satisfies **PI** for all $a \in \mathfrak{Act}$.

Proof:

(\Rightarrow): Suppose that $\mathcal{S}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a \models_{\sim} \varphi \rightarrow [a]\perp$. By monotonicity of \models_{\sim} , $\mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \models_{\sim} \varphi \rightarrow [a]\perp$, too. As $\langle \mathcal{S}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a \rangle$ satisfies **PI***, $\mathcal{S}, \mathcal{I} \models_{\text{PDL}} \varphi \rightarrow [a]\perp$.

Now suppose that $\mathcal{S}, \mathcal{I}^a \not\models_{\text{PDL}} \varphi \rightarrow [a]\perp$. Then there exists a possible worlds model $\mathcal{M} = \langle W, R_a, V \rangle$ 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 $\not\models_v^{\mathcal{M}} [a]\perp$. Let $\mathcal{M}' = \langle W', R', V' \rangle$ be such that $W' = W$, $V' = V$, $R'_{a'} = \emptyset$, for $a' \neq a$, and $R'_a = R_a$. Then $\models^{\mathcal{M}'} \mathcal{S} \wedge \mathcal{I}$, and as $\mathcal{S}, \mathcal{I} \models_{\text{PDL}} \varphi \rightarrow [a]\perp$, we get a contradiction.

(\Leftarrow): Suppose that $\langle \mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \rangle$ does not satisfy **PI***. Then there exists $\varphi \in \mathfrak{Fml}$ such that $\mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \models_{\sim} \varphi \rightarrow [a]\perp$ and $\mathcal{S}, \mathcal{I} \not\models_{\text{PDL}} \varphi \rightarrow [a]\perp$.

Claim: $\mathcal{S}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a \models_{\sim} \varphi \rightarrow [a]\perp$.

To witness, suppose $\mathcal{S}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a \not\models_{\sim} \varphi \rightarrow [a]\perp$. Then there exists a possible worlds model $\mathcal{M} = \langle W, R_a, V \rangle$ such that $\models^{\mathcal{M}} \mathcal{S} \wedge \mathcal{E}^a \wedge \mathcal{X}^a \wedge \mathcal{I}^a$ and there is a possible world $v \in W$ such that $\models_v^{\mathcal{M}} \varphi$ and $\not\models_v^{\mathcal{M}} [a]\perp$, i.e., there is $v' \in W$ such that $R_a(v) = v'$. (We are going to extend \mathcal{M} to be a model of $\langle \mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \rangle$.)

For each $a' \in \mathfrak{Act}$, $a' \neq a$, we define:

$$\begin{aligned}\mathcal{E}_{\psi}^{a'}(w) &= \{\psi : \varphi \rightarrow [a']\psi \in \mathcal{E}^{a'} \text{ and } \models_w^{\mathcal{M}} \psi\} \\ \mathcal{I}^{a'}(w) &= \{\varphi : \varphi \rightarrow [a']\perp \in \mathcal{I}^{a'} \text{ and } \models_w^{\mathcal{M}} \varphi\} \\ \mathcal{X}^{a'}(w) &= \{\varphi : \varphi \rightarrow \langle a' \rangle \top \in \mathcal{X}^{a'} \text{ and } \models_w^{\mathcal{M}} \varphi\}\end{aligned}$$

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

- $R'(w) = \emptyset$, if $\mathcal{I}^{a'}(w) \neq \emptyset$;
- $R'(w) = w'$, if $\mathcal{E}_{\psi}^{a'}(w') \neq \emptyset$.

Because, by hypothesis, $\langle \mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \rangle$ satisfies **PS***, there is no implicit static law, i.e., \mathcal{S} is complete in our sense. Then, \mathcal{M}' is a model of \mathcal{S} . We have that \mathcal{M}' is a model of \mathcal{E} , too: for every $\varphi \rightarrow [a]\psi \in \mathcal{E}$ and every $w \in W'$, if $\models_w^{\mathcal{M}'} \varphi$, then $\models_{w'}^{\mathcal{M}'} \psi$ for all $w' \in W'$ such that $wR'w'$. Clearly \mathcal{M}' is also a model of \mathcal{I} . \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 those worlds $w \in W'$ such that $\mathcal{X}^{a'}(w) \neq \emptyset$ there is a world accessible by R' , viz. some w' such that $\mathcal{E}_{\psi}^{a'}(w') \neq \emptyset$ (because $R'(w) = \emptyset$ in this case would preclude $\mathcal{X}^{a'}(w) \neq \emptyset$, 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, $\mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \not\models_{\sim} \varphi \rightarrow [a]\perp$, hence we must have $\mathcal{S}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a \models_{\sim} \varphi \rightarrow [a]\perp$. (End of the proof of the claim.)

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

■

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 Disturbing modularity

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 effects of actions

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):

$$\text{if } \mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \models_{\sim} \varphi \rightarrow [a]\psi \text{ and } \mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \not\models_{\sim} \varphi \rightarrow [a]\perp, \\ \text{then } \mathcal{S}, \mathcal{E} \models_{\sim} \varphi \rightarrow [a]\psi$$

But consider the following intuitively correct action theory:

$$\mathcal{S}_4 = \emptyset, \quad \mathcal{E}_4 = \left\{ \begin{array}{l} \text{loaded} \rightarrow [\text{shoot}] \neg \text{alive}, \\ (\neg \text{loaded} \wedge \text{alive}) \rightarrow [\text{shoot}] \text{alive} \end{array} \right\}$$

$$\mathcal{X}_4 = \{\text{hasGun} \rightarrow \langle \text{shoot} \rangle \top\}, \quad \mathcal{I}_4 = \{\neg \text{hasGun} \rightarrow [\text{shoot}] \perp\}$$

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

$$\mathcal{S}_4, \mathcal{E}_4, \mathcal{X}_4, \mathcal{I}_4 \models_{\sim} \neg \text{hasGun} \vee \text{loaded} \rightarrow [\text{shoot}] \neg \text{alive}$$

and

$$\mathcal{S}_4, \mathcal{E}_4, \mathcal{X}_4, \mathcal{I}_4 \not\models_{\sim} \neg \text{hasGun} \vee \text{loaded} \rightarrow [\text{shoot}] \perp,$$

but

$$\mathcal{S}_4, \mathcal{E}_4 \not\models_{\sim} \neg \text{hasGun} \vee \text{loaded} \rightarrow [\text{shoot}] \neg \text{alive}$$

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

Another possibility of improving our modularity criteria could be:

P⊥ (No unattainable effects):

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

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 \perp' (No unattainable effects – strong version):

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

Indeed, for the above action theory we have

$$\mathcal{E}_4 \models_{\sim} (\neg \text{hasGun} \wedge \text{loaded}) \rightarrow [\text{shoot}] \neg \text{alive},$$

but

$$\mathcal{S}_4, \mathcal{E}_4, \mathcal{X}_4, \mathcal{I}_4 \models_{\sim} (\neg \text{hasGun} \wedge \text{loaded}) \rightarrow [\text{shoot}] \perp.$$

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 4.1 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 executabilities

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):

$$\text{if } \mathcal{S}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a \not\models_{\sim} \varphi \rightarrow [a]\perp, \text{ then } \mathcal{S}, \mathcal{X}^a \models_{\text{PDL}} \varphi \rightarrow \langle a \rangle \top$$

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 nonmonotonic 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 10V, then the car (necessarily) will start. Suppose we also know that if the tension is below 8V, then the car will not start. What should we conclude in situations where we know that the tension is 9V? 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 8.1 If $\langle \mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \rangle$ satisfies Postulate **PS***, then $\mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \models_{\sim} \perp$ if and only if $\mathcal{S} \models \perp$.

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

Proof:

(\Leftarrow): Straightforward, by monotonicity.

(\Rightarrow): Suppose that $\mathcal{S}, \mathcal{E}^a, \mathcal{I}^a \not\models_{\sim} \varphi \rightarrow [a]\psi$. Then there exists a possible worlds model $\mathcal{M} \in \mathcal{M}_{\sim}$, $\mathcal{M} = \langle W, R_a, V \rangle$, such that $\models^{\mathcal{M}} \mathcal{S} \wedge \mathcal{E}^a \wedge \mathcal{I}^a$ and there is a possible world $v \in W$ such that $\models_v^{\mathcal{M}} \varphi$ and $\not\models_v^{\mathcal{M}} [a]\psi$, i.e., there is $v' \in W$ such that $R_a(v) = v'$ and $\not\models_{v'}^{\mathcal{M}} \psi$. (We are going to extend \mathcal{M} to obtain a model of $\langle \mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \rangle$ and thus show that $\mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \not\models_{\sim} \varphi \rightarrow [a]\psi$.)

For each $a' \in \mathfrak{Act}$, $a' \neq a$, we define:

$$\begin{aligned}\mathcal{E}_{\psi}^{a'}(w) &= \{\psi : \varphi \rightarrow [a']\psi \in \mathcal{E}^{a'} \text{ and } \models_w^{\mathcal{M}} \psi\} \\ \mathcal{I}^{a'}(w) &= \{\varphi : \varphi \rightarrow [a']\perp \in \mathcal{I}^{a'} \text{ and } \models_w^{\mathcal{M}} \varphi\}\end{aligned}$$

$$\mathcal{X}^{a'}(w) = \{\varphi : \varphi \rightarrow \langle a' \rangle \top \in \mathcal{X}^{a'} \text{ and } \models_w^{\mathcal{M}} \varphi\}$$

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

- $R'(w) = \emptyset$, if $\mathcal{I}^{a'}(w) \neq \emptyset$;
- $R'(w) = w'$, if $\mathcal{E}_\psi^{a'}(w') \neq \emptyset$.

Because, by hypothesis, $\langle \mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \rangle$ satisfies **PS***, there is no implicit static law, i.e., \mathcal{S} is complete in our sense. Then, \mathcal{M}' is a model of \mathcal{S} . We have that \mathcal{M}' is a model of \mathcal{E} , too: for every $\varphi \rightarrow [a]\psi \in \mathcal{E}$ and every $w \in W'$, if $\models_w^{\mathcal{M}'} \varphi$, then $\models_{w'}^{\mathcal{M}'} \psi$ for all $w' \in W'$ such that $wR'w'$. Clearly \mathcal{M}' is also a model of \mathcal{I} . \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 those worlds $w \in W'$ such that $\mathcal{X}^{a'}(w) \neq \emptyset$ there is a world accessible by R' , viz. some w' such that $\mathcal{E}_\psi^{a'}(w') \neq \emptyset$ (because $R'(w) = \emptyset$ in this case would preclude $\mathcal{X}^{a'}(w) \neq \emptyset$, as long as **PS*** is satisfied). Hence $\models_w^{\mathcal{M}'} \mathcal{S} \wedge \mathcal{E} \wedge \mathcal{X} \wedge \mathcal{I}$. Because there are $v, v' \in W'$ such that $\models_v^{\mathcal{M}'} \varphi$, $R'(v) = v'$ and $\not\models_{v'}^{\mathcal{M}'} \psi$, we have $\mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \not\models_{v'} \varphi \rightarrow [a]\psi$. ■

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 8.3 If $\langle \mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \rangle$ satisfies Postulate **PS***, then $\mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \models_{\sim} \varphi \rightarrow \langle a \rangle \top$ if and only if $\mathcal{S}, \mathcal{X}^a \models_{\sim} \varphi \rightarrow \langle a \rangle \top$.

Proof:

(\Leftarrow): Straightforward, by monotonicity.

(\Rightarrow): Suppose that $\mathcal{S}, \mathcal{X}^a \not\models_{\sim} \varphi \rightarrow \langle a \rangle \top$. Then there exists a possible worlds model $\mathcal{M} \in \mathcal{M}_{\sim}$, $\mathcal{M} = \langle W, R_a, V \rangle$, such that $\models_w^{\mathcal{M}} \mathcal{S} \wedge \mathcal{X}^a$ and there is a possible world $v \in W$ such that $\models_v^{\mathcal{M}} \varphi$ and $\not\models_v^{\mathcal{M}} \langle a \rangle \top$. (We are going to extend \mathcal{M} to build a model of $\langle \mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \rangle$ and thus conclude that $\mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \not\models_{\sim} \varphi \rightarrow \langle a \rangle \top$.)

For each $a' \in \mathbf{Act}$, $a' \neq a$, we define:

$$\begin{aligned}\mathcal{E}_\psi^{a'}(w) &= \{\psi : \varphi \rightarrow [a']\psi \in \mathcal{E}^{a'} \text{ and } \models_w^{\mathcal{M}} \psi\} \\ \mathcal{I}^{a'}(w) &= \{\varphi : \varphi \rightarrow [a']\perp \in \mathcal{I}^{a'} \text{ and } \models_w^{\mathcal{M}} \varphi\} \\ \mathcal{X}^{a'}(w) &= \{\varphi : \varphi \rightarrow \langle a' \rangle \top \in \mathcal{X}^{a'} \text{ and } \models_w^{\mathcal{M}} \varphi\}\end{aligned}$$

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

- $R'(w) = \emptyset$, if $\mathcal{I}^{a'}(w) \neq \emptyset$;
- $R'(w) = w'$, if $\mathcal{E}_\psi^{a'}(w') \neq \emptyset$.

Because, by hypothesis, $\langle \mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \rangle$ satisfies **PS***, there is no implicit static law, i.e., \mathcal{S} is complete in our sense. Then, \mathcal{M}' is a model of \mathcal{S} . We have that \mathcal{M}' is a model of \mathcal{E} , too: for every $\varphi \rightarrow [a]\psi \in \mathcal{E}$ and every $w \in W'$, if $\models_w^{\mathcal{M}'} \varphi$, then $\models_{w'}^{\mathcal{M}'} \psi$ for all $w' \in W'$ such that $wR'w'$. Clearly \mathcal{M}' is also a model of \mathcal{I} . \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 those worlds $w \in W'$ such that $\mathcal{X}^{a'}(w) \neq \emptyset$ there is a world accessible by R' , viz. some w' such that $\mathcal{E}_\psi^{a'}(w') \neq \emptyset$ (because $R'(w) = \emptyset$ in this case would preclude $\mathcal{X}^{a'}(w) \neq \emptyset$, 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 $\mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \not\models_{\sim} \varphi \rightarrow \langle a \rangle \top$. ■

Corollary 8.1 **PX** is a consequence of **PS**.

Proof: Straightforward. ■

Theorem 8.4 If $\langle \mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \rangle$ satisfies Postulates **PS*** and **PI***, then $\mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \models_{\sim} \varphi \rightarrow [a]\perp$ if and only if $\mathcal{S}, \mathcal{I}^a \models_{\sim} \varphi \rightarrow [a]\perp$.

Proof:

(\Leftarrow): Straightforward, by monotonicity.

(\Rightarrow): If $\mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \models_{\sim} \varphi \rightarrow [a]\perp$, then from **PS*** and Theorem 8.2 we have $\mathcal{S}, \mathcal{E}^a, \mathcal{I}^a \models_{\sim} \varphi \rightarrow [a]\perp$. From this and **PI*** it follows $\mathcal{S}, \mathcal{I}^a \models_{\sim} \varphi \rightarrow [a]\perp$. ■

9 Related work

Pirri and Reiter have investigated the metatheory of the Situation Calculus [38]. 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 [1] 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 [38] and the same condition on effect laws is assumed, which syntactically precludes the existence of implicit static laws.

Despite of using many of the object-oriented paradigm tools and techniques, no mention is made to the concepts of cohesion and coupling [40], which are closely related to modularity [19]. In the approach presented in [1], 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).

Zhang *et al.* [46] have also proposed an assessment of what a good action theory should look like. They develop the ideas in the framework of EPDL [47], an extended version of PDL which allows for propositions as modalities to represent causal connection between literals. We do not present the details of that, but concentrate on the main metatheoretical results.

Zhang *et al.* propose a normal form for describing action theories,⁹ and investigate three levels of consistency. Roughly speaking, an action theory \mathcal{T}

⁹But 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 $[drop_{black}]black$, $[drop_{white}]white$,

is *uniformly consistent* if it is globally consistent (i.e., $\mathcal{T} \not\models_{\text{EPDL}} \perp$); a formula φ is \mathcal{T} -*consistent* if $\mathcal{T} \models_{\text{EPDL}} \neg\varphi$, for \mathcal{T} a uniformly consistent theory; \mathcal{T} is *universally consistent* if (in our terms) every logically possible world is accessible. $\mathcal{T} \models_{\text{EPDL}} \varphi$ implies $\models_{\text{EPDL}} \varphi$.

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

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

Nevertheless, this is only a necessary condition: it is not hard to imagine action theories that are uniformly consistent but in which we can still have implicit inexecutabilities that are not caught by their algorithm. Consider for instance a scenario with a lamp that can be turned on and off by a toggle action, and its EPDL representation given by:

$$\mathcal{T} = \left\{ \begin{array}{l} \text{on} \rightarrow [\text{toggle}] \neg \text{on}, \\ \text{off} \rightarrow [\text{toggle}] \text{on}, \\ [\text{on}] \neg \text{off}, \\ [\neg \text{on}] \text{off} \end{array} \right\}$$

The causal statement $[\text{on}] \neg \text{off}$ means that *on* causes $\neg \text{off}$. Such an action theory satisfies each of the consistency requirements (in particular it is uniformly consistent, as $\mathcal{T} \not\models_{\text{EPDL}} \perp$). Nevertheless, \mathcal{T} is not safe for the static law $\neg(\text{on} \wedge \text{off})$ cannot be proved.¹⁰

Although they are concerned with the same kind of problems that have been discussed in this paper, 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 action theory, and then every time something is wrong

$[\text{drop}_{\text{black,white}}] \text{black}$ and $[\text{drop}_{\text{black,white}}] \text{white}$, where $\text{drop}_{\text{black}}$ is the action of dropping the coin on a black square (analogously for the others) and $\text{drop} = \text{drop}_{\text{black}} \cup \text{drop}_{\text{white}} \cup \text{drop}_{\text{black,white}}$, with “ \cup ” the nondeterministic composition of actions.

¹⁰A possible solution could be to consider the set of static constraints explicitly in the action theory (viz. in the deductive system). For the running example, taking into account the constraint $\text{on} \leftrightarrow \neg \text{off}$ (derived from the causal statements and the EPDL global axioms), we can conclude that \mathcal{T} is safe. On the other hand, all the side effects such a modification could have on the whole theory has yet to be analyzed.

with it this is taken as a global problem inherent to the action 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).

Lang *et al.* [27] address consistency in the causal laws approach [34], focusing on the computational aspects. They suppose an abstract notion of completion of an action theory solving the frame problem. Given an action 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 logical 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 paper, which essentially amounts to the explanation closure assumption [43].)

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 instance, still considering the lamp scenario, the representation of the action theory for *toggle* is:

$$\mathcal{T}^{toggle} = \left\{ \begin{array}{l} on \xrightarrow{toggle} off, \\ off \xrightarrow{toggle} on, \\ off \longrightarrow \neg on, \\ on \longrightarrow \neg off \end{array} \right\}$$

where the first two formulas are conditional effect laws for *toggle*, and the latter two causal laws 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 *toggle* cannot be executed in a state satisfying $on \wedge off$.

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. In spite of 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 metatheoretical 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, and to reject as bad an action theory admitting situations where some action cannot be executed at all. As an example, consider the very simple action theory given by $\mathcal{S}_5 = \mathcal{S}_1$, $\mathcal{E}_5 = \{[\text{tease}]\text{walking}\}$, $\mathcal{X}_5 = \mathcal{X}_1$ and $\mathcal{I}_5 = \mathcal{I}_1$, and consider $\rightsquigarrow = \{\langle \text{tease}, \text{walking} \rangle\}$. Observe that, with our approach, it suffices to derive the implicit inexecutability law $\neg \text{alive} \rightarrow [\text{tease}] \perp$, change \mathcal{I} , and the system will properly run in situations where $\neg \text{alive}$ 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 $\mathcal{T}^{\text{tease}}$, if we test its executability, we will get the answer “no”, the reason being that *tease* is not executable in the possible state where $\neg \text{alive}$ 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 on 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, consider the action theory $\langle \mathcal{S}'_5, \mathcal{E}'_5, \mathcal{X}'_5, \mathcal{I}'_5 \rangle$, with $\mathcal{S}'_5 = \mathcal{S}_5$, $\mathcal{E}'_5 = \mathcal{E}_5$, $\mathcal{X}'_5 = \{\text{alive} \rightarrow \langle \text{tease} \rangle \top\}$ and $\mathcal{I}'_5 = \{\neg \text{alive} \rightarrow [\text{tease}] \perp\}$, with the same \rightsquigarrow , obtained by the correction of $\langle \mathcal{S}_5, \mathcal{E}_5, \mathcal{X}_5, \mathcal{I}_5 \rangle$ above with the algorithms we propose. Observe that $\langle \mathcal{S}'_5, \mathcal{E}'_5, \mathcal{X}'_5, \mathcal{I}'_5 \rangle$ 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 $\mathcal{X} = \{\langle \text{tease} \rangle \top\}$, $\mathcal{X} = \{\text{alive} \rightarrow \langle \text{tease} \rangle \top\}$ and $\mathcal{X} = \emptyset$. By their turn, Zhang *et al.*’s allows for specifying executabilities, but their consistency definitions do not distinguish the cases $\text{alive} \rightarrow \langle \text{tease} \rangle \top$ and $\langle \text{tease} \rangle \top$.

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 [5]. Indeed, the notions of regulation consistency given in the mentioned work and that of modularity presented in [20] and refined here can be proved to be equivalent. The main difference between the mentioned work and the approach in [20] relies on the fact that in [5] some syntactical restrictions on the formulas have to be made in order to make the algorithm to work.

Lifschitz and Ren [30] propose an action description language derived from $\mathcal{C}+$ [13] 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 [31] 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 [28, 29, 8]. In [17] 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.¹¹

We have put forward some postulates, and in particular tried to demonstrate that when there are implicit static and inexecutability laws then one

¹¹It 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.

has slipped up in designing the action theory in question. As shown, a possible solution comes into its own with Algorithms 4.1 and 5.1, 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 used a version of PDL, but our notions and results can be applied to other frameworks as well. It is worth noting however that for first-order based frameworks the consistency checks of Algorithms 4.1 and 5.1 are undecidable. We can get rid of this by assuming that $\langle \mathcal{S}, \mathcal{E}, \mathcal{X}, \mathcal{I} \rangle$ is finite and there is no function symbol in the language. In this way, the result of *NewCons* is finite and the algorithm terminates.

The present paper is also a step toward a solution to the problem of indirect dependences: indeed, if the indirect dependence $shoot \rightsquigarrow \neg walking$ is not in \rightsquigarrow , then after running Algorithm 5.1 we get an indirect inexecutability $(loaded \wedge walking) \rightarrow [shoot] \perp$, 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 [4]). We are currently investigating this issue.

A different viewpoint of the work we presented here can be found in [19], where modularity of action theories is assessed from a software engineering

perspective. A modularity-based approach for narrative reasoning about actions is given in [23].

Our postulates do not take into account causality statements linking propositions such as those defined in [31, 34]. This could be a topic for further investigation.

Acknowledgments

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

References

- [1] E. Amir. (De)composition of situation calculus theories. In *Proc. 17th Nat. Conf. on Artificial Intelligence (AAAI'2000)*, pages 456–463, Austin, 2000. AAAI Press/MIT Press.
- [2] P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic*. Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, 2001.
- [3] M. A. Castilho, O. Gasquet, and A. Herzig. Formalizing action and change in modal logic I: the frame problem. *J. of Logic and Computation*, 9(5):701–735, 1999.
- [4] M. A. Castilho, A. Herzig, and I. J. Varzinczak. It depends on the context! a decidable logic of actions and plans based on a ternary dependence relation. In S. Benferhat and E. Giunchiglia, editors, *Workshop on Non-Monotonic Reasoning (NMR'02)*, pages 343–348, Toulouse, 2002.
- [5] L. Cholvy. Checking regulation consistency by using SOL-resolution. In *Proc. 7th Int. Conf. on AI and Law*, pages 73–79, Oslo, 1999.
- [6] R. Demolombe, A. Herzig, and I. Varzinczak. Regression in modal logic. *J. of Applied Non-Classical Logics (JANCL)*, 13(2):165–185, 2003.
- [7] P. Doherty, W. Lukaszewicz, and A. Szałas. Explaining explanation closure. In *Proc. 9th Int. Symposium on Methodologies for Intelligent Systems*, number 1079 in LNCS, Zakopane, Poland, 1996. Springer-Verlag.
- [8] T. Eiter, E. Erdem, M. Fink, and J. Senko. Updating action domain descriptions. In Kaelbling and Saffiotti [22], pages 418–423.

- [9] J. J. Finger. *Exploiting constraints in design synthesis*. PhD thesis, Stanford University, Stanford, 1987.
- [10] M. Gelfond and V. Lifschitz. Representing action and change by logic programs. *Journal of Logic Programming*, 17(2/3&4):301–321, 1993.
- [11] M. L. Ginsberg and D. E. Smith. Reasoning about actions II: The qualification problem. *Artificial Intelligence*, 35(3):311–342, 1988.
- [12] E. Giunchiglia, G. N. Kartha, and V. Lifschitz. Representing action: indeterminacy and ramifications. *Artificial Intelligence*, 95(2):409–438, 1997.
- [13] E. Giunchiglia, J. Lee, V. Lifschitz, N. McCain, and H. Turner. Non-monotonic causal theories. *Artificial Intelligence*, 153(1–2):49–104, 2004.
- [14] S. Hanks and D. McDermott. Default reasoning, nonmonotonic logics, and the frame problem. In T. Kehler and S. Rosenschein, editors, *Proc. 5th Nat. Conf. on Artificial Intelligence (AAAI'86)*, pages 328–333, Philadelphia, 1986. Morgan Kaufmann Publishers.
- [15] D. Harel. Dynamic logic. In D. M. Gabbay and F. Günthner, editors, *Handbook of Philosophical Logic*, volume II, pages 497–604. D. Reidel, Dordrecht, 1984.
- [16] D. Harel, J. Tiuryn, and D. Kozen. *Dynamic Logic*. MIT Press, 2000.
- [17] A. Herzig, L. Perrussel, and I. Varzinczak. Elaborating domain descriptions (preliminary report). Technical Report RT-2006-01-FR, Institut de recherche en informatique de Toulouse (IRIT), Université Paul Sabatier, January 2006. <http://www.irit.fr/ACTIVITES/LILaC/>.
- [18] A. Herzig and I. Varzinczak. An assessment of actions with indeterminate and indirect effects in some causal approaches. Technical Report 2004-08-R, Institut de recherche en informatique de Toulouse (IRIT), Université Paul Sabatier, May 2004. <http://www.irit.fr/ACTIVITES/LILaC/>.
- [19] A. Herzig and I. Varzinczak. Cohesion, coupling and the meta-theory of actions. In Kaelbling and Saffiotti [22], pages 442–447.
- [20] A. Herzig and I. Varzinczak. On the modularity of theories. In R. Schmidt, I. Pratt-Hartmann, M. Reynolds, and H. Wansing, editors, *Advances in Modal Logic*, volume 5, pages 93–109. King’s College

Publications, 2005. Selected papers of AiML 2004 (also available at <http://www.aiml.net/volumes/volume5>).

- [21] K. Inoue. Linear resolution for consequence finding. *Artificial Intelligence*, 56(2–3):301–353, 1992.
- [22] L. Kaelbling and A. Saffiotti, editors. *Proc. 19th Int. Joint Conf. on Artificial Intelligence (IJCAI'05)*, Edinburgh, 2005. Morgan Kaufmann Publishers.
- [23] A. Kakas, L. Michael, and R. Miller. *Modular- \mathcal{E}* : an elaboration tolerant approach to the ramification and qualification problems - preliminary report. Proc. of the 7th Int. Symp. on Logical Formalizations of Commonsense Reasoning. Corfu, Greece, 2005.
- [24] N. Kartha and V. Lifschitz. Actions with indirect effects (preliminary report). In J. Doyle, E. Sandewall, and P. Torasso, editors, *Proc. 4th Int. Conf. on Knowledge Representation and Reasoning (KR'94)*, pages 341–350, Bonn, 1994. Morgan Kaufmann Publishers.
- [25] M. Kracht and F. Wolter. Properties of independently axiomatizable bimodal logics. *J. of Symbolic Logic*, 56(4):1469–1485, 1991.
- [26] M. Kracht and F. Wolter. Simulation and transfer results in modal logic: A survey. *Studia Logica*, 59:149–177, 1997.
- [27] J. Lang, F. Lin, and P. Marquis. Causal theories of action – a computational core. In V. Sorge, S. Colton, M. Fisher, and J. Gow, editors, *Proc. 18th Int. Joint Conf. on Artificial Intelligence (IJCAI'03)*, pages 1073–1078, Acapulco, 2003. Morgan Kaufmann Publishers.
- [28] R. Li and L.M. Pereira. What is believed is what is explained. In H. Shrobe and T. Senator, editors, *Proc. 13th Nat. Conf. on Artificial Intelligence (AAAI'96)*, pages 550–555, Portland, 1996. AAAI Press/MIT Press.
- [29] P. Liberatore. A framework for belief update. In *Proc. 7th Eur. Workshop on Logics in AI (JELIA 2000)*, pages 361–375, 2000.
- [30] V. Lifschitz and W. Ren. Towards a modular action description language. To appear in the Working Notes of the AAAI Symposium on Formalizing Background Knowledge.
- [31] F. Lin. Embracing causality in specifying the indirect effects of actions. In Mellish [37], pages 1985–1991.

- [32] P. Marquis. Knowledge compilation using theory prime implicants. In Mellish [37], pages 837–843.
- [33] P. Marquis. Consequence finding algorithms. In D. M. Gabbay and Ph. Smets, editors, *Algorithms for Defensible and Uncertain Reasoning, in S. Moral, J. Kohlas (Eds), Handbook of Defeasible Reasoning and Uncertainty Management Systems*, volume 5, chapter 2, pages 41–145. Kluwer Academic Publishers, 2000.
- [34] N. McCain and H. Turner. A causal theory of ramifications and qualifications. In Mellish [37], pages 1978–1984.
- [35] J. McCarthy. Epistemological problems of artificial intelligence. In N. S. Sridharan, editor, *Proc. 5th Int. Joint Conf. on Artificial Intelligence (IJCAI'77)*, pages 1038–1044, Cambridge, MA, 1977. Morgan Kaufmann Publishers.
- [36] J. McCarthy and P. J. Hayes. Some philosophical problems from the standpoint of artificial intelligence. In B. Meltzer and D. Mitchie, editors, *Machine Intelligence*, volume 4, pages 463–502. Edinburgh University Press, 1969.
- [37] C. Mellish, editor. *Proc. 14th Int. Joint Conf. on Artificial Intelligence (IJCAI'95)*, Montreal, 1995. Morgan Kaufmann Publishers.
- [38] F. Pirri and R. Reiter. Some contributions to the metatheory of the situation calculus. *Journal of the ACM*, 46(3):325–361, 1999.
- [39] S. Popkorn. *First Steps in Modal Logic*. Cambridge University Press, 1994.
- [40] R. S. Pressman. *Software Engineering: A Practitioner's Approach*. McGraw-Hill, 1992.
- [41] R. Reiter. The frame problem in the situation calculus: A simple solution (sometimes) and a completeness result for goal regression. In V. Lifschitz, editor, *Artificial Intelligence and Mathematical Theory of Computation: Papers in Honor of John McCarthy*, pages 359–380. Academic Press, San Diego, 1991.
- [42] R. Reiter. *Knowledge in Action: Logical Foundations for Specifying and Implementing Dynamical Systems*. MIT Press, Cambridge, MA, 2001.

- [43] L. K. Schubert. Monotonic solution of the frame problem in the situation calculus: an efficient method for worlds with fully specified actions. In H. E. Kyberg, R. P. Loui, and G. N. Carlson, editors, *Knowledge Representation and Defeasible Reasoning*, pages 23–67. Kluwer Academic Publishers, 1990.
- [44] M. Thielscher. Computing ramifications by postprocessing. In Mellish [37], pages 1994–2000.
- [45] M. Thielscher. Ramification and causality. *Artificial Intelligence*, 89(1–2):317–364, 1997.
- [46] D. Zhang, S. Chopra, and N. Y. Foo. Consistency of action descriptions. In *PRICAI'02, Topics in Artificial Intelligence*. Springer-Verlag, 2002.
- [47] D. Zhang and N. Y. Foo. EPDL: A logic for causal reasoning. In B. Nebel, editor, *Proc. 17th Int. Joint Conf. on Artificial Intelligence (IJCAI'01)*, pages 131–138, Seattle, 2001. Morgan Kaufmann Publishers.

A Proof of Theorem 4.3

An action theory $\langle \mathcal{S}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a \rangle$ with \rightsquigarrow satisfies Postulate **PS** if and only if $\mathcal{S}_{imp^*} = \emptyset$.

We recall that \models is logical consequence in Classical Propositional Logic, and $PI(A)$ is the set of prime implicants of the set A of classical formulas.

Before giving the proof of the theorem, we recall some properties of prime implicants [32, 33] and the function *NewCons* [21]. Let $\varphi \in \mathfrak{Fml}$, $A \subseteq \mathfrak{Fml}$, and χ be a clause. Then

1. $\models \varphi \leftrightarrow \bigwedge PI(\varphi)$ [33, Corollary 3.2].
2. $PI(A) \cup NewCons_A(\varphi) = PI(A \wedge \varphi)$ (from the definition of *NewCons*).
3. $\models A \wedge \varphi \leftrightarrow A \wedge NewCons_A(\varphi)$ (from 1 and 2)
4. If $PI(\varphi) \models \chi$, then there exists $\chi' \in PI(\varphi)$ such that $\chi' \models \chi$ [33, Proposition 3.4].

Let $\rightsquigarrow \subseteq \mathfrak{Act} \times \mathfrak{Lit}$, $\varphi \rightarrow \langle a \rangle \top \in \mathcal{X}^a$, $\mathcal{C}^a = \mathcal{E}^a \cup \mathcal{I}^a$, and $\hat{\mathcal{C}}^a \subseteq \mathcal{C}^a$. We define:

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

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

Lemma A.1 $\mathcal{S} \cup \{\psi_{\hat{\mathcal{C}}^a}\} \cup \{\bigwedge_{\models_{\sim} \neg l_j \rightarrow [a] \neg l_j} \neg l_j\} \models \perp$ if and only if $\mathcal{S} \cup \text{NewCons}_{\mathcal{S}}(\psi_{\hat{\mathcal{C}}^a}) \cup \{\bigwedge_{\models_{\sim} \neg l_j \rightarrow [a] \neg l_j} \neg l_j\} \models \perp$.

Proof: Consequence of Property 3. ■

Lemma A.2 If $\mathcal{S} \cup \text{NewCons}_{\mathcal{S}}(\psi_{\hat{\mathcal{C}}^a}) \cup \{\bigwedge_{\models_{\sim} \neg l_j \rightarrow [a] \neg l_j} \neg l_j\} \models \perp$, then $\exists \chi \in \text{NewCons}_{\mathcal{S}}(\psi_{\hat{\mathcal{C}}^a})$ such that $\mathcal{S} \cup \{\chi\} \cup \{\bigwedge_{\models_{\sim} \neg l_j \rightarrow [a] \neg l_j} \neg l_j\} \models \perp$.

Proof: Consequence of Properties 1, 2 and 4. ■

Lemma A.3 If $\mathcal{S} \cup \{\varphi, \varphi_{\hat{\mathcal{C}}^a}\} \cup \{\bigwedge_{\models_{\sim} \neg l_j \rightarrow [a] \neg l_j} \neg l_j\} \not\models \perp$ and $\mathcal{S} \cup \text{NewCons}_{\mathcal{S}}(\psi_{\hat{\mathcal{C}}^a}) \cup \{\bigwedge_{\models_{\sim} \neg l_j \rightarrow [a] \neg l_j} \neg l_j\} \models \perp$, then $\exists \chi \in \text{NewCons}_{\mathcal{S}}(\psi_{\hat{\mathcal{C}}^a})$ such that $\mathcal{S} \cup \{\chi\} \cup \{\bigwedge_{\models_{\sim} \neg l_j \rightarrow [a] \neg l_j} \neg l_j\} \models \perp$.

Proof: By Lemma A.2 and Classical Logic. ■

Lemma A.4 If $\mathcal{S} \cup \{\varphi, \varphi_{\hat{\mathcal{C}}^a}\} \cup \{\bigwedge_{\models_{\sim} \neg l_j \rightarrow [a] \neg l_j} \neg l_j\} \not\models \perp$ and $\mathcal{S} \cup \text{NewCons}_{\mathcal{S}}(\psi_{\hat{\mathcal{C}}^a}) \cup \{\bigwedge_{\models_{\sim} \neg l_j \rightarrow [a] \neg l_j} \neg l_j\} \models \perp$, then $\exists \chi \in \text{NewCons}_{\mathcal{S}}(\psi_{\hat{\mathcal{C}}^a})$ such that both $\mathcal{S} \cup \{\varphi, \varphi_{\hat{\mathcal{C}}^a}\} \cup \{\bigwedge_{\models_{\sim} \neg l_j \rightarrow [a] \neg l_j} \neg l_j\} \not\models \perp$ and $\mathcal{S} \cup \{\chi\} \cup \{\bigwedge_{\models_{\sim} \neg l_j \rightarrow [a] \neg l_j} \neg l_j\} \models \perp$.

Proof: Trivially, by Lemma A.3. ■

Lemma A.5 If $\chi \in \text{NewCons}_{\mathcal{S}}(\psi_{\hat{\mathcal{C}}^a})$ is such that $\mathcal{S} \cup \{\varphi, \varphi_{\hat{\mathcal{C}}^a}\} \cup \{\bigwedge_{\models_{\sim} \neg l_j \rightarrow [a] \neg l_j} \neg l_j\} \not\models \perp$ and $\mathcal{S} \cup \{\chi\} \cup \{\bigwedge_{\models_{\sim} \neg l_j \rightarrow [a] \neg l_j} \neg l_j\} \models \perp$, then $\mathcal{S} \cup \{\varphi, \varphi_{\hat{\mathcal{C}}^a}\} \cup \{\bigwedge_{\substack{i \in \chi \\ a \not\sim i}} \neg l_i\} \not\models \perp$ and $\mathcal{S} \cup \{\chi\} \cup \{\bigwedge_{\substack{i \in \chi \\ a \not\sim i}} \neg l_i\} \models \perp$.

Proof: Let $\chi \in \text{NewCons}_{\mathcal{S}}(\psi_{\hat{\mathcal{C}}^a})$ be such that $\mathcal{S} \cup \{\varphi, \varphi_{\hat{\mathcal{C}}^a}\} \cup \{\bigwedge_{\models_{\sim} \neg l_j \rightarrow [a] \neg l_j} \neg l_j\} \not\models \perp$ and $\mathcal{S} \cup \{\chi\} \cup \{\bigwedge_{\models_{\sim} \neg l_j \rightarrow [a] \neg l_j} \neg l_j\} \models \perp$.

If $\chi = \perp$, the result is trivial.

Let $\text{atm}(\varphi)$ denote the set of atoms occurring in a classical formula φ .

- If $\text{atm}(\chi) \not\subset \text{atm}(\bigwedge_{\models_{\sim} \neg l_j \rightarrow [a] \neg l_j} \neg l_j)$, then the premise is false (and the lemma trivially holds).
- If $\text{atm}(\chi) = \text{atm}(\bigwedge_{\models_{\sim} \neg l_j \rightarrow [a] \neg l_j} \neg l_j)$, the lemma holds.

- Let $atm(\chi) \subset atm(\bigwedge_{\substack{\models \neg l_j \rightarrow [a] \neg l_j}} \neg l_j)$. From $\mathcal{S} \cup \{\varphi, \varphi_{\hat{\mathcal{C}}^a}\} \cup \{\bigwedge_{\substack{\models \neg l_j \rightarrow [a] \neg l_j}} \neg l_j\} \not\models \perp$ it follows $\mathcal{S} \cup \{\varphi, \varphi_{\hat{\mathcal{C}}^a}\} \cup \{\bigwedge_{\substack{l_i \in \chi \\ a \not\sim l_i}} \neg l_i\} \not\models \perp$. From $\mathcal{S} \cup \{\chi\} \cup \{\bigwedge_{\substack{\models \neg l_j \rightarrow [a] \neg l_j}} \neg l_j\} \models \perp$ and because $\mathcal{S} \cup \{\bigwedge_{\substack{\models \neg l_j \rightarrow [a] \neg l_j}} \neg l_j\} \not\models \perp$, it follows $\mathcal{S} \cup \{\chi\} \cup \{\bigwedge_{\substack{l_i \in \chi \\ a \not\sim l_i}} \neg l_i\} \models \perp$.

■

Lemma A.6 If $\chi \in NewCons_{\mathcal{S}}(\psi_{\hat{\mathcal{C}}^a})$ is such that $\mathcal{S} \cup \{\varphi, \varphi_{\hat{\mathcal{C}}^a}\} \cup \{\bigwedge_{\substack{l_i \in \chi \\ a \not\sim l_i}} \neg l_i\} \not\models \perp$ and $\mathcal{S} \cup \{\chi\} \cup \{\bigwedge_{\substack{l_i \in \chi \\ a \not\sim l_i}} \neg l_i\} \models \perp$, then $\mathcal{S} \cup \{\varphi, \varphi_{\hat{\mathcal{C}}^a}\} \cup \{\bigwedge_{\substack{l_i \in \chi \\ a \not\sim l_i}} \neg l_i\} \not\models \perp$ and $\forall l_i \in \chi, a \not\sim l_i$.

Proof: From $\mathcal{S} \cup \{\varphi, \varphi_{\hat{\mathcal{C}}^a}\} \cup \{\bigwedge_{\substack{l_i \in \chi \\ a \not\sim l_i}} \neg l_i\} \not\models \perp$ we conclude $\mathcal{S} \cup \{\bigwedge_{\substack{l_i \in \chi \\ a \not\sim l_i}} \neg l_i\} \not\models \perp$. From this and the hypothesis $\mathcal{S} \cup \{\chi\} \cup \{\bigwedge_{\substack{l_i \in \chi \\ a \not\sim l_i}} \neg l_i\} \models \perp$, it follows $\mathcal{S} \cup \{\bigwedge_{\substack{l_i \in \chi \\ a \not\sim l_i}} \neg l_i\} \models \neg \chi$. If $\mathcal{S} \models \neg \chi$, then $\mathcal{S}, \psi_{\hat{\mathcal{C}}^a} \models \neg \chi$, and because $\chi \in NewCons_{\mathcal{S}}(\psi_{\hat{\mathcal{C}}^a})$, we have $\chi \models \neg \chi$, a contradiction. Hence $\mathcal{S} \cup \{\chi\} \not\models \perp$. Suppose now that there is at least one literal $l \in \chi$ such that $\neg l$ does not appear in $\bigwedge_{\substack{l_i \in \chi \\ a \not\sim l_i}} \neg l_i$. Then, the propositional valuation in which $\chi \leftarrow \text{true}$ satisfies $\mathcal{S} \cup \{\chi\} \cup \bigwedge_{\substack{l_i \in \chi \\ a \not\sim l_i}} \neg l_i$, and then $\mathcal{S}, \{\chi\}, \bigwedge_{\substack{l_i \in \chi \\ a \not\sim l_i}} \neg l_i \not\models \perp$. Hence there cannot be such a literal, and then $\forall l_i \in \chi, a \not\sim l_i$. ■

Proof of Theorem 4.3

(\Rightarrow): Suppose $\mathcal{S}_{imp^*} \neq \emptyset$. Then at the first step of the algorithm there has been some $\varphi \rightarrow \langle a \rangle T \in \mathcal{X}^a$ and some $\hat{\mathcal{C}}^a \subseteq \mathcal{C}^a$ such that $\mathcal{S}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a \models \neg(\varphi \wedge \varphi_{\hat{\mathcal{C}}^a})$ and $\mathcal{S} \not\models \neg(\varphi \wedge \varphi_{\hat{\mathcal{C}}^a})$. Hence $\langle \mathcal{S}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a \rangle$ with \rightsquigarrow does not satisfy Postulate **PS**.

(\Leftarrow): Suppose that $\mathcal{S}_{imp^*} = \emptyset$. Therefore for all $\varphi' \rightarrow \langle a \rangle T \in \mathcal{X}^a$ and for all subsets $\hat{\mathcal{C}}^a \subseteq \mathcal{C}^a$, we have that

$$\forall \chi \in NewCons_{\mathcal{S}}(\psi_{\hat{\mathcal{C}}^a}) \text{ if } \mathcal{S} \cup \{\varphi', \varphi_{\hat{\mathcal{C}}^a}, \neg \chi\} \not\models \perp, \text{ then } \exists l_i \in \chi, a \rightsquigarrow l_i \quad (1)$$

From (1), the contraposition of Lemmas A.6–A.3, and Lemma A.1, it follows that for all $\varphi' \rightarrow \langle a \rangle T \in \mathcal{X}^a$ and $\hat{\mathcal{C}}^a \subseteq \mathcal{C}^a$,

$$\text{if } \mathcal{S} \cup \{\varphi', \varphi_{\hat{\mathcal{C}}^a}\} \cup \{\bigwedge_{\substack{\models \neg l_j \rightarrow [a] \neg l_j}} \neg l_j\} \not\models \perp, \text{ then } \mathcal{S} \cup \{\psi_{\hat{\mathcal{C}}^a}\} \cup \{\bigwedge_{\substack{\models \neg l_j \rightarrow [a] \neg l_j}} \neg l_j\} \not\models \perp. \quad (2)$$

Now, suppose $\mathcal{S} \not\models \varphi$ for some propositional φ . We will build a model \mathcal{M} such that \mathcal{M} is a \sim -model for $\langle \mathcal{S}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a \rangle$ that does not satisfy φ . Let W be the set of all propositional valuations satisfying \mathcal{S} that falsify φ . As $\mathcal{S} \not\models \varphi$, $\mathcal{S} \cup \{\neg\varphi\}$ is satisfiable, hence W must be nonempty. We define the binary relation R_a on W such that $wR_a w'$ if and only if for every $\varphi \rightarrow [a]\psi \in \mathcal{C}^a$ such that $\models_w^\mathcal{M} \varphi$:

- $\models_{w'}^\mathcal{M} \psi$; and
- $\models_{w'}^\mathcal{M} \neg l_j$ for all l_j such that $a \not\sim l_j$ and $\models_w^\mathcal{M} \neg l_j$.

Taking the obvious definition of V we obtain a model $\mathcal{M} = \langle W, R, V \rangle$. We have that \mathcal{M} is a \sim -model, by the definition of R_a , and $\models^\mathcal{M} \mathcal{S} \wedge \mathcal{E}^a \wedge \mathcal{X}^a \wedge \mathcal{I}^a$, because:

- $\models^\mathcal{M} \mathcal{S}$: by definition of W ;
- $\models^\mathcal{M} \mathcal{C}^a$: for every world w and every $\varphi \rightarrow [a]\psi \in \mathcal{C}^a$, if $\models_w^\mathcal{M} \varphi$, then, by the definition of R_a , $\models_{w'}^\mathcal{M} \psi$ for all $w' \in W$ such that $wR_a w'$;
- $\models^\mathcal{M} \mathcal{X}^a$: let $\mathcal{E}^a(w) = \{\varphi \rightarrow [a]\psi \in \mathcal{E}^a : \models_w^\mathcal{M} \varphi\}$, and $\text{indep}_a(w) = \{\neg l : a \not\sim l \text{ and } \models_w^\mathcal{M} \neg l\}$. Then, for every world w and every $\varphi' \rightarrow \langle a \rangle \top \in \mathcal{X}^a$, if $\models_w^\mathcal{M} \varphi' \wedge \varphi_{\mathcal{E}^a(w)} \wedge \text{indep}_a(w)$, then from (2), $\psi_{\mathcal{E}^a(w)} \wedge \text{indep}_a(w) \not\models \perp$. As W is maximal, there exists at least one w' such that $\models_{w'}^\mathcal{M} \psi_{\mathcal{E}^a(w)} \wedge \text{indep}_a(w)$. As R_a is maximal by definition, we have $wR_a w'$. and the definition of R_a , there exists at least one w' such that $wR_a w'$.

Clearly $\not\models^\mathcal{M} \varphi$, by the definition of W . Hence $\mathcal{S}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a \not\models_\sim \varphi$. Therefore $\langle \mathcal{S}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a \rangle$ and \sim violate Postulate **PS**. ■

B Proof of Theorem 5.2

If $\langle \mathcal{S}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a \rangle$ with \sim satisfies Postulate **PS**, then it satisfies Postulate **PI** if and only if $\mathcal{I}_{imp}^a = \emptyset$.

Let $\sim \subseteq \mathfrak{Act} \times \mathfrak{Lit}$ and $\varphi \rightarrow \langle a \rangle \top \in \mathcal{X}^a$. For every $\hat{\mathcal{E}}^a \subseteq \mathcal{E}^a$ we define:

$$\varphi_{\hat{\mathcal{E}}^a} = \bigwedge \{\varphi_i : \varphi_i \rightarrow [a]\psi_i \in \hat{\mathcal{E}}^a\}$$

$$\psi_{\hat{\mathcal{E}}^a} = \bigwedge \{\psi_i : \varphi_i \rightarrow [a]\psi_i \in \hat{\mathcal{E}}^a\}$$

Moreover, we define

$$\mathcal{I}_{imp}^a = \{(\varphi_{\mathcal{E}^a} \wedge \neg \chi) \rightarrow [a]\perp : \hat{\mathcal{E}}^a \subseteq \mathcal{E}^a, \mathcal{S} \cup \{\varphi_{\mathcal{E}^a}, \neg \chi\} \not\models \perp, \chi \in \text{NewCons}_{\mathcal{S}}(\psi_{\hat{\mathcal{E}}^a}), a \not\sim l_i \forall l_i \in \chi\}$$

Lemma B.1 If $\langle \mathcal{S}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a \rangle$ satisfies Postulate **PS**, then $\mathcal{S}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a \models_{\sim} \varphi \rightarrow [a]\perp$ implies $\mathcal{S}, \mathcal{E}^a, \mathcal{I}^a \models_{\sim} \varphi \rightarrow [a]\perp$.

Proof: Straightforward as a special case of Theorem 8.2. ■

Lemma B.2 If for each $\hat{\mathcal{E}}^a$ we have $\mathcal{S} \cup \{\varphi_{\hat{\mathcal{E}}^a}\} \not\models \perp$ implies $\mathcal{S} \cup \{\psi_{\hat{\mathcal{E}}^a}\} \not\models \perp$, then $\mathcal{S}, \mathcal{E}^a, \mathcal{I}^a \models_{PDL} \varphi \rightarrow [a]\perp$ implies $\mathcal{S}, \mathcal{I}^a \models_{PDL} \varphi \rightarrow [a]\perp$.

Proof: If $\mathcal{S}, \mathcal{I}^a \not\models_{PDL} \varphi \rightarrow [a]\perp$, then there exists a PDL-model $\mathcal{M} = \langle W, R, V \rangle$ such that $\models^{\mathcal{M}} \mathcal{S} \wedge \mathcal{I}^a$, and there is a possible world $w \in W$ such that $\models_w^{\mathcal{M}} \varphi$ and $\models_w^{\mathcal{M}} \langle a \rangle \top$.

(We are going to construct a counter-model for $\mathcal{S}, \mathcal{E}^a, \mathcal{I}^a \models_{PDL} \varphi \rightarrow [a]\perp$.)

Let $\mathcal{E}^a(w) = \{\varphi \rightarrow [a]\psi \in \mathcal{E}^a : \models_w^{\mathcal{M}} \varphi\}$. Then $\varphi_{\mathcal{E}^a(w)} = \bigwedge \{\varphi_i : \varphi_i \rightarrow [a]\psi_i \in \mathcal{E}^a(w)\}$ is such that $\models_w^{\mathcal{M}} \varphi_{\mathcal{E}^a(w)}$. As $\mathcal{S} \wedge \varphi_{\mathcal{E}^a(w)}$ is thus satisfiable, $\mathcal{S} \wedge \psi_{\mathcal{E}^a(w)}$, with $\psi_{\mathcal{E}^a(w)} = \bigwedge \{\psi_i : \varphi_i \rightarrow [a]\psi_i \in \mathcal{E}^a(w)\}$, must be satisfiable, too (by hypothesis, because $\mathcal{E}^a(w) \subseteq \mathcal{E}^a$). Hence, there exists a propositional valuation val such that $val(\mathcal{S} \wedge \psi_{\mathcal{E}^a(w)}) = 1$.

Consider, thus, v such that $v \notin W$, and extend V such that $V(v) = val$. Let $\mathcal{M}' = \langle W', R', V' \rangle$ be such that $W' = W \cup \{v\}$, $R'_a(u) = \{v\}$ for all u such that $\models_u^{\mathcal{M}} \varphi$ and $R'_a(u) = \emptyset$ otherwise, and $V' = V \cup \{(v, V(v))\}$.

Then:

- $\models^{\mathcal{M}'} \mathcal{S}$ because $\models^{\mathcal{M}} \mathcal{S}$ and $val(\mathcal{S}) = 1$.
- $\models_v^{\mathcal{M}'} \mathcal{I}^a$ because $\models_v^{\mathcal{M}'} [a]\perp$, by definition of $R'_a(v)$, and $\models_w^{\mathcal{M}'} \mathcal{I}^a$ because $\not\models_w^{\mathcal{M}'} \varphi$ for all $\varphi \rightarrow [a]\perp \in \mathcal{I}^a$, as $\not\models_w^{\mathcal{M}'} \varphi$ (otherwise, as $\models^{\mathcal{M}} \mathcal{I}^a$, we would not have $\models_w^{\mathcal{M}} \langle a \rangle \top$).
- $\models_v^{\mathcal{M}'} \mathcal{E}^a$ because $\models_v^{\mathcal{M}'} [a]\perp$, and $\models_w^{\mathcal{M}'} \mathcal{E}^a$ by construction of \mathcal{M}' : $R'_a(w) = \{v\}$ and $\models_v^{\mathcal{M}'} \psi_{\mathcal{E}^a(w)}$.
- $\models_w^{\mathcal{M}'} \varphi \wedge \langle a \rangle \top$.

Hence \mathcal{M}' is still a model of \mathcal{S} , \mathcal{I}^a and \mathcal{E}^a . Of course, \mathcal{M}' is a counter-model for $\varphi \rightarrow [a]\perp$. ■

Lemma B.3 Let $\mathcal{F}^a = \{\neg l \rightarrow [a]\neg l : a \not\sim l\}$. Then if $\mathcal{S}, \mathcal{E}^a, \mathcal{I}^a, \mathcal{I}_{imp}^a, \mathcal{F}^a \models_{\text{PDL}} \varphi \rightarrow [a]\perp$, then $\mathcal{S}, \mathcal{E}^a, \mathcal{I}^a, \mathcal{I}_{imp}^a \models_{\text{PDL}} \varphi \rightarrow [a]\perp$.

Proof: If $\mathcal{S}, \mathcal{E}^a, \mathcal{I}^a, \mathcal{I}_{imp}^a \not\models_{\text{PDL}} \varphi \rightarrow [a]\perp$, then there exists a PDL-model $\mathcal{M} = \langle W, R, V \rangle$ such that $\models_{\mathcal{M}} \mathcal{S} \wedge \mathcal{E}^a \wedge \mathcal{I}^a \wedge \mathcal{I}_{imp}^a$, and there is a possible world $w \in W$ such that $\models_w^{\mathcal{M}} \varphi$ and $\models_w^{\mathcal{M}} \langle a \rangle \top$.

We are going to construct a counter-model for $\mathcal{S}, \mathcal{E}^a, \mathcal{I}^a, \mathcal{I}_{imp}^a, \mathcal{F}^a \models_{\text{PDL}} \varphi \rightarrow [a]\perp$. Let $\mathcal{M}' = \langle W', R', V' \rangle$ be such that $W' = W$, and $R'_a(w') = \emptyset$ for every $w' \neq w$, and $V' = V$.

Of course, \mathcal{M}' is still a model of $\mathcal{S}, \mathcal{E}^a, \mathcal{I}^a$ and \mathcal{I}_{imp}^a .

For every $\hat{\mathcal{E}}^a \subseteq \mathcal{E}^a$, the case where $\models_w^{\mathcal{M}'} \varphi_{\hat{\mathcal{E}}^a} \wedge \neg \chi$, with $\chi \in \text{NewCons}_{\mathcal{S}}(\psi_{\hat{\mathcal{E}}^a})$, $a \not\sim l_i, \forall l_i \in \chi$, is impossible, because $\models_{\mathcal{M}'} \mathcal{I}_{imp}^a$ and hence we would have $R_a(w) = \emptyset$, contradicting the hypothesis that $\models_w^{\mathcal{M}'} \langle a \rangle \top$.

Thus, we have to consider only the following cases:

- if $\models_w^{\mathcal{M}'} l_i$, for every l_i such that $a \not\sim l_i$, then \mathcal{M}' is also a model of \mathcal{F}^a , and then we have a counter-model for $\mathcal{S}, \mathcal{E}^a, \mathcal{I}^a, \mathcal{I}_{imp}^a, \mathcal{F}^a \models_{\text{PDL}} \varphi \rightarrow [a]\perp$.
- if $\models_w^{\mathcal{M}'} \varphi_{\hat{\mathcal{E}}^a} \wedge \bigwedge \neg l_i$, where $a \not\sim l_i$ and there is no clause $\chi \in \text{NewCons}_{\mathcal{S}}(\psi_{\hat{\mathcal{E}}^a})$ such that $l_i \in \chi$, for some $\hat{\mathcal{E}}^a \subseteq \mathcal{E}^a$, then of course $\psi_{\hat{\mathcal{E}}^a} \wedge \bigwedge \neg l_i$ is satisfiable, i.e., there is a valuation where $\psi_{\hat{\mathcal{E}}^a} \wedge \bigwedge \neg l_i$ holds. Let $\text{val}_{\hat{\mathcal{E}}^a}$ be such a valuation.

Consider, thus, v such that $v \notin W$, and extend V' such that $V'(v) = \text{val}_{\hat{\mathcal{E}}^a}$. Now let $\mathcal{M}'' = \langle W'', R'', V'' \rangle$ be such that $W'' = W' \cup \{v\}$, and $R''_a(w) = \text{val}_{\hat{\mathcal{E}}^a}$, and $V'' = V' \cup \{(v, V'(v))\}$.

Again, it can easily be checked that \mathcal{M}'' is a model of $\mathcal{S}, \mathcal{E}^a, \mathcal{I}^a$ and \mathcal{I}_{imp}^a . Moreover, it is a model of \mathcal{F}^a , and hence a model for $\mathcal{S}, \mathcal{E}^a, \mathcal{I}^a, \mathcal{I}_{imp}^a, \mathcal{F}^a$ and $\varphi \wedge \langle a \rangle \top$.

■

Lemma B.4 If $\mathcal{S}, \mathcal{E}^a, \mathcal{I}^a \models_{\sim} \varphi \rightarrow [a]\perp$, then $\mathcal{S}, \mathcal{I}^a, \mathcal{I}_{imp}^a \models_{\text{PDL}} \varphi \rightarrow [a]\perp$.

Proof: Let

$$\mathcal{E}^{a+} = \{\varphi_{\hat{\mathcal{E}}^a} \rightarrow [a]\psi_{\hat{\mathcal{E}}^a} : \hat{\mathcal{E}}^a \subseteq \mathcal{E}^a, \mathcal{S} \cup \{\varphi_{\hat{\mathcal{E}}^a}\} \not\models \perp, \mathcal{S} \cup \{\psi_{\hat{\mathcal{E}}^a}\} \not\models \perp\}$$

$$\mathcal{E}^{a-} = \{\varphi_{\hat{\mathcal{E}}^a} \rightarrow [a]\psi_{\hat{\mathcal{E}}^a} : \hat{\mathcal{E}}^a \subseteq \mathcal{E}^a, \mathcal{S} \cup \{\varphi_{\hat{\mathcal{E}}^a}\} \not\models \perp, \mathcal{S} \models \psi_{\hat{\mathcal{E}}^a} \rightarrow \perp\}$$

The following steps establish the result.

1. $\mathcal{S}, \mathcal{E}^a, \mathcal{I}^a \models_{\sim} \varphi \rightarrow [a]\perp$, by hypothesis
 2. $\mathcal{S}, \mathcal{E}^a, \mathcal{I}^a, \mathcal{I}_{imp}^a \models_{\sim} \varphi \rightarrow [a]\perp$, from 1. and monotonicity
 3. $\mathcal{S} \models_{\text{PDL}} \bigwedge \mathcal{E}^a \leftrightarrow (\bigwedge \mathcal{E}^{a+} \wedge \bigwedge \mathcal{E}^{a-})$, by definition of \mathcal{E}^{a+} and \mathcal{E}^{a-} , and PDL
 4. $\mathcal{S}, \mathcal{E}^{a+} \cup \mathcal{E}^{a-}, \mathcal{I}^a, \mathcal{I}_{imp}^a \models_{\sim} \varphi \rightarrow [a]\perp$, from 2. and 3.
 5. $\mathcal{E}^{a-} \subseteq \mathcal{I}_{imp}^a$, as if $\varphi_{\mathcal{E}^a} \rightarrow [a]\psi_{\mathcal{E}^a} \in \mathcal{E}^{a-}$, then $\mathcal{S} \models \psi_{\mathcal{E}^a} \rightarrow \perp$, and then $\perp \in \text{NewCons}_{\mathcal{S}}(\psi_{\mathcal{E}^a})$, from which it follows that $(\varphi_{\mathcal{E}^a} \wedge \top) \rightarrow [a]\perp \in \mathcal{I}_{imp}^a$, and then $\varphi_{\mathcal{E}^a} \rightarrow [a]\perp \in \mathcal{I}_{imp}^a$
 6. $\mathcal{S}, \mathcal{E}^{a+}, \mathcal{I}^a, \mathcal{I}_{imp}^a \models_{\sim} \varphi \rightarrow [a]\perp$, from 4. and 5.
 7. $\mathcal{S}, \mathcal{E}^{a+}, \mathcal{I}^a, \mathcal{I}_{imp}^a, \mathcal{F}^a \models_{\text{PDL}} \varphi \rightarrow [a]\perp$, from 6. and definition of \sim , where $\mathcal{F}^a = \{\neg l \rightarrow [a]\neg l : a \not\sim l\}$.
 8. $\mathcal{S}, \mathcal{E}^{a+}, \mathcal{I}^a, \mathcal{I}_{imp}^a \models_{\text{PDL}} \varphi \rightarrow [a]\perp$, from 7. and Lemma B.3.
 9. $\mathcal{S}, \mathcal{I}^a, \mathcal{I}_{imp}^a \models_{\text{PDL}} \varphi \rightarrow [a]\perp$, from 8. and Lemma B.2, whose hypothesis is satisfied by the definition of \mathcal{E}^{a+} .
-

Proof of Theorem 5.2

(\Rightarrow) : Straightforward, as every time $\mathcal{S}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a \models_{\sim} \varphi \rightarrow [a]\perp$, we have $\mathcal{S}, \mathcal{I}^a \models_{\text{PDL}} \varphi \rightarrow [a]\perp$, and then \mathcal{I}_{imp}^a never changes.

(\Leftarrow) : We are going to show that if $\mathcal{S}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a \models_{\sim} \varphi \rightarrow [a]\perp$ and $\mathcal{I}_{imp}^a = \emptyset$, then $\mathcal{S}, \mathcal{I}^a \models_{\text{PDL}} \varphi \rightarrow [a]\perp$.

1. $\mathcal{S}, \mathcal{E}^a, \mathcal{X}^a, \mathcal{I}^a \models_{\sim} \varphi \rightarrow [a]\perp$, by hypothesis
 2. $\mathcal{S}, \mathcal{E}^a, \mathcal{I}^a \models_{\sim} \varphi \rightarrow [a]\perp$, from 1. and Lemma B.1
 3. $\mathcal{S}, \mathcal{I}^a, \mathcal{I}_{imp}^a \models_{\text{PDL}} \varphi \rightarrow [a]\perp$, from 2. and Lemma B.4
 4. $\mathcal{S}, \mathcal{I}^a \models_{\text{PDL}} \varphi \rightarrow [a]\perp$, from 3. and hypothesis $\mathcal{I}_{imp}^a = \emptyset$.
-