Encoding solutions of the frame problem in dynamic logic

Norman Foo, Dongmo Zhang
Yan Zhang, Samir Chopra and Bao Quoc Vo

Knowledge Systems Group
School of Computer Science and Engineering
University of New South Wales
Australia

Abstract. We investigate the relationship amongst some solutions to the frame problem. We encode Pednault's syntax-based solution [20], Baker's state-minimization policy [1], and Gelfond & Lifchitz's Action Language $\mathcal{A}$ [7] in the propositional dynamic logic (PDL). The formal relationships among these solutions are given. The results of the paper show that dynamic logic, as one of the formalisms for reasoning about dynamic domains, can be used as a formal tool for comparing and unifying logics of action.
Keywords: relationships between formalisms, frame problem, dynamic logic.

1 Introduction

Among the established formalisms for specifying and reasoning about actions are the situation calculus [19, 23] , STRIPS [3], the event calculus [17], action languages [7] and some other monotonic or nonmonotonic logics such as in [9]. Fundamental problems in this area, such as the frame problem, ramification problem, and qualification problem, have been widely investigated with varying degrees of success. Clearly, the time has come to analyze, compare and systematize these formalisms and solutions in order to obtain a more complete and unified (if possible) theory of action.

This paper focuses on solutions to the frame problem. We compare and analyze the main solutions to the frame problem in the literature by encoding them in the propositional dynamic logic (PDL). The reasons for choosing PDL as the medium are twofold. First, the language of dynamic logic is expressive. It provides built-in expression of compound actions (i.e., generated from primitive actions by the program connectives $;, \cup, ?, *$), non-deterministic effects and qualifications of actions. It has also been extended to represent concurrent actions [10], non-execution of actions [8], indirect effects of actions [11, 26]. Second, dynamic logic features a sound and complete axiomatic deductive system and a well-developed Kripkean semantics. Its proof and model theory have reached a high degree of sophistication through the development of theoretical computer science. Some features, such as decidability and the finite model property of PDL, and techniques such as bisimulation and filtration, are well understood.

In contrast to other formalisms, such as the situation calculus [23] and action languages [7], PDL does not have a built-in solution to the frame problem (PDL-based solutions to the frame problem have been proposed via extensions [2, 8, 22]). In this paper, however, we show that three solutions to the frame problem (Pednault's syntax-based approach [20], Baker's circumscription [1] and the action language $\mathcal{A}$ [7]) can be encoded in PDL. The relationship amongst these solutions is clarified and we prove that in the case that action descriptions are in normal form and queries are simple, these solutions to the frame problem are essentially equivalent. In contrast to the work in [14], our results show that the equivalence of the solutions heavily depends on the syntactical restrictions of action descriptions and queries.

Due to the limitation of space, we omit all the proofs of theorems[1].

2 Reasoning about actions in PDL

In dynamic logic, a causal relation between an action α and its effect A is expressed by a modal formula: $[\alpha]A$, read as α *always causes* A. For instance, $[Shoot]\neg alive$ represents "shooting at a turkey kills the turkey". The formula $\langle\alpha\rangle A$ reads as α *is executable and possibly causes* A *to be true*, where $\langle\alpha\rangle$ is the dual operator of $[\alpha]$. In particular, $\langle\alpha\rangle\top$ represents α is executable, where $\top$ represents the logical constant **true**. $\prec a \succ A$ denotes "$\langle a\rangle\top \to \langle a\rangle A$", meaning "*if* a *is executable, then* a *may cause* A." A language of PDL consists of a set Flu of fluent symbols (propositional variables) and a set Act_P of primitive action symbols. We will use f, f_1, f_2, *etc.*, to denote fluents, and use a, a_1,a_2, *etc.*, for primitive actions. The formulas ($A \in Fma$) and actions ($\alpha \in Act$) can be defined as usual [18]. A formula which does not include modal operators is referred to as a *propositional formula* ($\varphi \in Fma_P$). The semantics and deductive system of PDL can be found in any standard introductory text e.g [18].

2.1 Action description

PDL provides a formal language to describe behaviors and internal relations of a dynamic system. Those sentences which describe the generic effects of actions, domain constraints and causal ramifications are generally called *action description*. In this paper, an action description of a dynamic system is a any (finite or infinite) set of PDL formulas.

Example 1 Consider the Yale Shooting Problem [12] described by the following action description:

$$\Sigma = \left\{ \begin{array}{l} \neg loaded \to [Load]loaded \\ loaded \to [Shoot]\neg alive \\ loaded \to [Shoot]\neg loaded \\ \langle Load\rangle\top, \langle Wait\rangle\top, \langle Shoot\rangle\top \end{array} \right\}$$

[1] They are available at http://www.cse.unsw.edu.au/~ksg/Pubs/ksgworking.html.

The first three sentences state the effects of action *Load* and *Shoot* on fluent *loaded* and *alive* (effect axioms). The last three represent the executability of actions (qualification axioms).

An action description Σ is *normal* if each formula in Σ is of the form:

- $\varphi \to [a]l$ (*deterministic action law*)
- $\varphi \to \prec a \succ l$ (*non-deterministic action law*)
- $\varphi \to \langle a \rangle \top$ (*qualification law*)

  where $\varphi \in Fma_P$, $a \in Act_P$ and l is a fluent literal.[2]

It is easy to see that the action description in Example 1 is normal.

2.2 Reasoning with action description

A formula in an action description is different from an ordinary formula. The sentence "$loaded \to [Shoot]\neg alive$" states that whenever *loaded* is true, *Shoot* must cause $\neg alive$. In the situation calculus this is written as $\forall s(loaded(s) \to \neg alive(do(Shoot, s)))$ instead of $loaded(s) \to \neg alive(do(Shoot, s))$ for some particular situation s. A simple approach to the problem in dynamic logic, which was introduced in [26], is to treat an action description as a set of extra axioms of PDL.

Definition 1 [26] An *action description* Σ *is a set of formulas.* A formula A is Σ*-provable,* written $\vdash^{\Sigma} A$, if it belongs to the smallest set of formulas which contains all theorems of PDL all elements of Σ, and is closed under *modus ponens and modal generalization* [18].

Consider the action description Σ in Example 1. We can prove that $\vdash^{\Sigma} \neg loaded \to [Load; Shoot]\neg alive$.[3]

2.3 Consistency of action description

An action description Σ is *consistent*[4] if $\vdash^{\Sigma} \bot$, where $\bot$ represents logical constant "**false**". Let Σ be a normal action description. For any fluent f and any primitive action a, if we merge the action laws about a and $f(\neg f)$ in each form together, there are at most five laws about a and f in Σ:

$$\varphi_0 \to \langle a \rangle \top$$
$$\varphi_{1,1} \to [a]f,\ \varphi_{1,2} \to [a]\neg f$$
$$\varphi_{2,1} \to \prec a \succ \neg f,\ \varphi_{2,2} \to \prec a \succ f$$

[2] In [25], the normal form of action descriptions is defined in a more general version to express indirect effects of actions based on the extended PDL [26].

[3] By using PDL axioms and Definition 1.

[4] In [26], it is called *uniformly consistent*, distinguishing from the consistency of normal set of formulas.

It is easy to see that if φ_0, $\varphi_{1,1}$and $\varphi_{1,2}$ are true simultaneously, then the action description will contain a contradiction. Similarly for φ_0, $\varphi_{1,j}$ and $\varphi_{2,j}$ ($j = 1$ or $j = 2$). We call a normal action description Σ is *safe* if it is satisfies the following assumption:

$\vdash \neg\varphi_0 \vee \neg\varphi_{1,1} \vee\neg\varphi_{1,2}$ and $\vdash \neg\varphi_0 \vee \neg\varphi_{1,j} \vee\neg\varphi_{2,j}$ *($j = 1, 2$)*

The following theorem shows that the safety is a sufficient condition of the consistency of normal action descriptions.

Theorem 1 [25]*Let Σ be a normal action description. If Σ is safe, then it is consistent.*

Since the action description in Example 1 is safe, it is consistent.

We remark that although not every action description can be expressed by normal form, the normal form is quite expressive. Any action description written in the form of pre-condition axioms and successor state axioms in the *propositional* situation calculus language (that is, there are no sort *object* and function symbols in the language [23]) can be translated into normal form (see appendix) (the resultant action descriptions are safe). Action descriptions written in $\mathcal{A}$ or in STRIPS and the determinism of actions (i.e., for any initial state there exists one and only one next state) can also be expressed by normal form.

3 Properties of PDL models

We now present some special properties of PDL models which are not included the standard discourse of dynamic logic but are useful for the purpose of the paper.

3.1 PDL models

A model for a PDL language is a structure of the form $M = (W, \{R_a : a \in Act_P\}, V)$, with R_a a binary relation on W for each primitive action. Note that we only consider the accessibility relations of primitive actions. Those for compound actions can be defined by using the standard model conditions [18]. The satisfiability relation is defined as usual. A model M *satisfying* a formula A in world w is denoted $M \models_w A$. A is *valid* in M, denoted by $M \models A$, if $M \models_w A$ for all $w \in W$. Let Σ be an action description. A model M is a *Σ-model* if $M \models A$ for any $A \in \Sigma$. Intuitively, a model is a Σ-model if Σ is true in every world of the model; $Mod(\Sigma)$ denotes the set of all Σ-models. In [26], it is shown that *for any action description Σ, $\vdash^{\Sigma} A$ iff A is valid in all Σ-models.* We now investigate models which are relevant to the models of action language and situation calculus.

Definition 2 A model $M = (W, \mathcal{R}, V)$ is *saturated* if for each interpretation I of Flu, there exists $w \in W$ such that $M \models_w I$. We use $Mod_S(\Sigma)$ to denote the set of all saturated Σ-models.

Proposition 1 *If Σ is normal and safe, then $\vdash^{\Sigma} A$ iff $M \models A$ for any $M \in Mod_S(\Sigma)$.*

We will show in section 6 that the saturation of PDL models corresponds to the Existence of Situation Axioms (ESA) [1]. Note that Proposition 1 depends on the definition of the normal form of action description. If we allow a normal action description to describe domain constraints or indirect effects, this proposition will cease to hold.

Definition 3 A model $M = (W, \mathcal{R}, V)$ is *natural* if
1. W is the set of all interpretations of Flu,
2. for any $f \in Flu$, $w \in V(f)$ iff $f \in w$.
We denote the set of all natural Σ-models by $Mod_N(\Sigma)$.

It is easy to see that any natural model is saturated.

Proposition 2 *If Σ be normal and safe, then $\vdash^\Sigma \varphi \to [a_1; \cdots; a_n]l$ iff $M \models \varphi \to [a_1; \cdots; a_n]l$ for any $M \in Mod_N(\Sigma)$.*

A formula in the form $\varphi \to [a_1; \cdots; a_n]l$ is referred to as *a simple query*, where φ is a propositional formula, l a literal. Notice that Proposition 2 is only true for simple queries. For instance, let $\Sigma = \emptyset$, $Flu = \{f\}$ and $Act_P = \{a\}$. Let $A = f \vee \prec a \succ \neg f \vee \prec a \succ \prec a \succ f \vee \prec a \succ \prec a \succ \prec a \succ f$. Then A is valid in all the natural models but $\not\vdash^\Sigma A$. This proposition is a key lemma of Theorem 2.

Definition 4 A model $M = (W, \mathcal{R}, V)$ is *functional* if for any $a \in Act_P$, R_a is a function on W. We denote the set of all natural functional Σ-models by $Mod_{NF}(\Sigma)$.

The syntactical condition with respect to functional models is so-called *determinism*, which means that each state can have and only have one next state after an action.

Definition 5 *Let $\Xi = \{\langle a \rangle f \to [a]f : a \in Act_P$ and $f \in Flu\} \cup \{\langle a \rangle \top : a \in Act_P\}$. An action description is deterministic*[5] *if $\Sigma \vdash \Xi$.*

Note that $\langle a \rangle f \to [a]f$ can be expressed in normal form in the following way: $f' \to [a]f$, $\neg f' \to [a]\neg f$, where f' is a new fluent symbol (in most cases, we can put the descriptions of determinism and effects of actions together without introducing new fluent symbols).

Proposition 3 *Let Σ be normal and safe. If Σ is deterministic, then $\vdash^\Sigma A$ iff $M \models A$ for any $M \in Mod_{NF}(\Sigma)$.*

Note the difference between Proposition 2 and 3. We can relax the restriction of simple query at the price of allowing only deterministic action descriptions.

[5] Here a deterministic action must be always executable, a condition that can be relaxed at the price of a more complex formalization.

3.2 Minimizing PDL models

Let $M = (W, \mathcal{R}, V)$ be a PDL model. For any $w \in W$, let $||w|| = \{f \in Flu : M \models_w f\} \cup \{\neg f : f \in Flu \ \&\ M \models_w f\}$. We denote $Chg(M) = \{(a, f, w) : \exists w'(wR_a w' \ \&\ f \in (||w|| \backslash ||w'||) \cup (||w'|| \backslash ||w||)\}$. In words, $(a, f, w) \in Chg(M)$ iff there exists an accessible world w' to w on action a such that the truth value of f is different at w and w'.

Definition 6 For any M_1, $M_2 \in Mod(\Sigma)$, $M_1 \sqsubset M_2$ iff

1. $W_1 = W_2$,

2. $V_1(f) = V_2(f)$,

3. $Chg(M_1) \subset Chg(M_2)$.

We denote the set of $\sqsubset$-minimal models in $Mod(\Sigma)$ as $\min(Mod(\Sigma))$. Intuitively, $M_1 \sqsubset M_2$ *means* M_1 *has lesser state change than* M_2.

4 Pednault's solution to the frame problem

We first encode Pednault's syntax-based solution [20] to the frame problem in PDL. Before doing this, let's recall the meaning of the frame problem.

To formalize the effects of actions in a dynamic system, it is necessary to provide all the effect axioms of actions (which specify what is affected by actions). Often this is easy because most actions affect only a few of the relevant fluents. In contrast, listing all the frame axioms (which specify what is not affected by actions) is tedious. Moreover, they are much more numerous than effect axioms. For instance, in Example 1, only effect axioms were listed. There are nine frame axioms, such as $alive \rightarrow [Load]alive$, $loaded \rightarrow [Wait]loaded$ etc., that were not listed. Without these axioms, the action description is incomplete. We cannot even establish the intuitive assertion $\vdash^{\Sigma} alive \rightarrow [Load]alive$. The frame problem is how to invent an inference mechanism for reasoning about effect of action with incomplete action descriptions.

Pednault [20] introduced an approach to the frame problem with which frame axioms can be automatically generated from effect axioms and qualification axioms. Consider an normal action description Σ without non-deterministic action laws. Suppose that the positive and negative effect axioms and qualification axioms about action a and fluent f in an action description Σ are:

$$\varphi_0 \rightarrow \langle a \rangle \top,\ \varphi_1 \rightarrow [a]f,\ \varphi_2 \rightarrow [a]\neg f.$$

According to the *Completeness Assumption* [23], we have the following frame axioms:

$FA^+_{a,f} : (\neg\varphi_0 \vee \neg\varphi_2) \wedge f \rightarrow [a]f$

$FA^-_{a,f} : (\neg\varphi_0 \vee \neg\varphi_1) \wedge \neg f \rightarrow [a]\neg f$

All frame axioms generated by this procedure are referred to as *the frame axioms with respect to* Σ. For instance, $\neg loaded \wedge alive \rightarrow [Shoot]alive$ is a frame axiom about $Shoot$ and $alive$ with respect to the action description in Example 1. Suppose that Δ is the set of all the generated frame axioms with respect to Σ. Then we are able to prove that $\{\neg loaded\} \vdash^{\Sigma \cup \Delta} [Load; Wait; Shoot]\neg alive$.

In general, given a set Σ of effect axioms, we generate all the frame axioms with the above procedure. Let Δ be all the generated frame axioms. Then $\Sigma \cup \Delta$ will be the complete action description with respect to Σ. Therefore, to answer a query A, we only have to make the inference $\vdash^{\Sigma \cup \Delta} A$.

The following theorem establishes the semantic condition for Pednault's solution. It also gives the relationship between syntax-based and minimization-based approaches.

Observation 1 *Let Σ be a normal action description without non-deterministic action laws. If Σ is safe, then $\vdash^{\Sigma \cup \Delta} A$ iff $M \models A$ for any $M \in min(Mod_S(\Sigma))$ where Δ is the set of frame axioms with respect to Σ.*

It is not hard to extend Pednault's solution to non-deterministic case.

5 Encoding the action language $\mathcal{A}$ in PDL

The action languages [6] offer a simple and elegant solution to the frame problem. In this section, we show that the action language $\mathcal{A}$ can be embedded into PDL. Our approach can also be extended the action language $\mathcal{B}$ and $\mathcal{C}$ if we base on the extended propositional logic (EPDL) [26].

An action description Σ in the language $\mathcal{A}$ [6] is a set of expressions of the form: a **causes** l **if** φ, where a is a primitive action, l is a fluent literal, and φ is a conjunction of literals. The state of a dynamic domain is expressed by a set of *axioms* of the form: **now** l. A query in action language $\mathcal{A}$ is an expression of the form: **necessarily** φ **after** $a_1, \cdots, a_n$, where φ is a propositional formula and $a_1, \cdots, a_n$ are primitive actions.

A structure $T = (W, \{R_a \subseteq W \times W : a \in Act_P\}, V)$ is a *transition system* of an action description Σ if

1. W is the set of all interpretations of Flu,
2. V is a function from Flu to 2^W such that $f \in V(w)$ iff $f \in w$.
3. $(w, w') \in R_a$ iff $E(a, w) \subseteq w' \subseteq E(a, w) \cup w$, where $E(a, w)$ is the set of the head l of all expression "a **causes** l **if** φ" in Σ such that w satisfies φ.

Let Γ be a set of expressions in the form: **now** l. A query "**necessarily** φ **after** $a_1, \cdots, a_n$" is a consequence of Γ in T if, for any chain $(w_0, w_1) \in R_{a_1}$, $\cdots$, $(w_{n-1}, w_n) \in R_{a_n}$, whenever w_0 satisfies l for each **now** $l \in \Gamma$, w_n satisfies φ. A query "**necessarily** φ **after** $a_1, \cdots, a_n$ is a consequence of Γ with respect to an action description Σ if it is a consequence of Γ in any transition system of Σ.

According to the translation between $\mathcal{A}$ and PDL shown in the Appendix, we can easily transform an action description and a state description between two languages. Since such a translation is one-to-one, we will only use PDL language to describe action descriptions, initial states and queries. They are easily recognized with context. It is easy to see that an action description in language $\mathcal{A}$ is always normal and safe. There is an important difference between the semantics of action language and PDL. In $\mathcal{A}$, there is no explicit expression for qualification of actions. An underlying assumption, called *Qualification Completeness,* in the semantics is that *an action is always executable unless the action description implies that it is not.* In PDL, there is no such assumption. Thus qualification of actions must be explicitly specified.

Let Σ be a finite action description in $\mathcal{A}$. Suppose the action laws about an action a and fluent f are $\varphi_1 \to [a]f$ and $\varphi_2 \to [a]\neg f$. This implies that a is not executable when $\varphi_1 \wedge \varphi_2$. Collecting all the conditions of non-executability: $\varphi_1^1 \wedge \varphi_2^1, \cdots, \varphi_1^n \wedge \varphi_2^n$, we know that a is not executable if $(\varphi_1^1 \wedge \varphi_2^1) \vee \cdots \vee (\varphi_1^n \wedge \varphi_2^n)$. By *Qualification Completeness*, we assume that $(\neg(\varphi_1^1 \wedge \varphi_2^1) \wedge \cdots \wedge \neg(\varphi_1^n \wedge \varphi_2^n)) \to \langle a \rangle \top$; such a condition is an *induced qualification law* . Let Λ be the set of all such laws from Σ. Then we have

Observation 2 *Let Σ be a finite action description and Γ a finite set of axioms, both in $\mathcal{A}$. A query "***necessarily** φ **after** $a_1, \cdots, a_n$*" is a consequence of Γ with respect to Σ iff $\vdash^{\Sigma \cup \Lambda \cup \Delta} (\bigwedge \Gamma) \to [a_1; \cdots; a_n]l$, where Δ is the frame axioms with respect to $\Sigma \cup \Lambda$.*

Clearly, the expressive power of $\mathcal{A}$ is quite restricted. Action descriptions can only be normal and queries can only be simple in our terminology.

6 Encoding Baker's solution in PDL models

Finally we consider Baker's solution. First, we have to recall the basic assumption of the approach.

6.1 Models of situation calculus

A model of the situation calculus [19, 1], (an SC-model), consists of the various domains: the domain of situations $|\mathcal{M}|_s$, the domain of actions $|\mathcal{M}|_a$ and the domain of fluents $|\mathcal{M}|_f$; as well as interpretations for the constants:

1. Interpretations for the relations $Holds$ and $Results$:
   $Holds^{\mathcal{M}} \subseteq |\mathcal{M}|_f \times |M|_s, \quad Ab^{\mathcal{M}} \subseteq |\mathcal{M}|_a \times |\mathcal{M}|_f \times |\mathcal{M}|_s,$
2. Interpretation for the $Result$ function:
   $Result^{\mathcal{M}} \in (|\mathcal{M}|_a \times |\mathcal{M}|_s \to |\mathcal{M}|_s).$

The following axioms were used in Baker's circumscriptive solution to the frame problem:

1. Unique names axioms:
   - Unique Name Axioms for fluents (UNAF): for any $f_1, f_2 \in Flu$, $f_1 \neq f_2$.
   - Unique Name Axioms for Actions(UNAA): for any $a_1, a_2 \in Flu$, $a_1 \neq a_2$.
2. Commonsense Law of Inertia (CLI):
   $\neg Ab(a, f, s) \leftrightarrow (Holds(f, Result(a, s)) \leftrightarrow Holds(f, s))$
3. Domain Closure Axioms:
   - Domain Closure Axiom for Fluents (DCAF):
     $f = f_1 \vee f = f_2 \vee \cdots \vee f = f_n \vee \cdots$
   - Domain Closure Axiom for Actions (DCAA):
     $a = a_1 \vee a = a_2 \vee \cdots \vee a = a_n \vee \cdots$

4. Existence of Situation Axioms (ESA):
$\exists s(Holds(f_1, s) \wedge Holds(f_2, s) \wedge \cdots \wedge Holds(f_n, s) \wedge \cdots)$
$\exists s(Holds(f_1, s) \wedge \neg Holds(f_2, s) \wedge \cdots \wedge Holds(f_n, s) \wedge \cdots)$
$\cdots$
$\exists s(\neg Holds(f_1, s) \wedge \neg Holds(f_2, s) \wedge \cdots \wedge \neg Holds(f_n, s) \wedge \cdots)$

For the sake of simplicity, we omit the formal presentation of domain closure and existence of situation axioms and ignore language differences in the representation of an action description based on the translation in the Appendix. Therefore, Σ is an action description in the situation calculus if it is a translation from an action description in PDL. Furthermore, an SC model $\mathcal{M}$ is a *Σ-model* if $\mathcal{M}$ satisfies all the formulas in Σ.

6.2 Relations between SC models and PDL models

First, we translate an SC model to a PDL model.

Definition 7 Let $\mathcal{M}$ be an SC model. A PDL model $M = (W, \mathcal{R}, V)$ is the *corresponding model* of $\mathcal{M}$ if

1. $W = |\mathcal{M}|_s$,
2. $(s_1^{\mathcal{M}}, s_2^{\mathcal{M}}) \in R_a$ iff $s_2^{\mathcal{M}} = Result^{\mathcal{M}}(a^{\mathcal{M}}, s_1^{\mathcal{M}})$,
3. $s^{\mathcal{M}} \in V(f)$ iff $Holds^{\mathcal{M}}(f^{\mathcal{M}}, s^{\mathcal{M}})$.

Lemma 1 *Let $M = (W, \mathcal{R}, V)$ be a PDL model and $\mathcal{M}$ the corresponding model of M. Then*

1. $M \models_{s^{\mathcal{M}}} \varphi$ iff $\mathcal{M} \models Holds(\varphi, s)$.
2. M is functional.
3. If $\mathcal{M}$ satisfies the common sense law of inertia, then $(a, f, s^{\mathcal{M}}) \in Chg(M)$ iff $(a^{\mathcal{M}}, f^{\mathcal{M}}, s^{\mathcal{M}}) \in Ab^{\mathcal{M}}$.
4. If Σ is a normal action description, then M is a Σ-model iff $\mathcal{M}$ is a Σ-model.
5. If $\mathcal{M}$ satisfies Existence of Situation Axioms, then M is saturated.

Next, we consider the transformation of PDL models to SC models.

Definition 8 Let $M = (W, \mathcal{R}, V)$ be a functional PDL model. An SC model $\mathcal{M}$ is the *corresponding model* of M if

1. $|\mathcal{M}|_f = Flu$, $|\mathcal{M}|_a = Act_P$, $|\mathcal{M}|_s = W$.
2. $s' = Result^{\mathcal{M}}(a, s)$ iff $(s, s') \in R_a$.
3. $(f, s) \in Holds^{\mathcal{M}}$ iff $f \in V(s)$.
4. $(a, f, s) \in Ab^{\mathcal{M}}$ iff $(a, f, s) \in Chg(M)$.

Lemma 2 *Let $\mathcal{M}$ be the corresponding model of a functional PDL model $M = (W, \mathcal{R}, V)$. Then*

1. $\mathcal{M} \models Holds(\varphi, s)$ iff $M \models_{s^{\mathcal{M}}} \varphi$
2. $\mathcal{M}$ satisfies the Commonsense Law of Inertia.
3. $\mathcal{M}$ satisfies Domain Closure Axioms for Fluents and Actions.
4. $\mathcal{M}$ satisfies Unique Names Axioms for Fluents and Actions.

5. If Σ is a normal action description, then $\mathcal{M}$ is a Σ-model iff M is a Σ-model.

6. If M is saturated, then $\mathcal{M}$ satisfies Existence-of-Situation Axiom.

The following shows the relationship between SC models and PDL models.

Lemma 3 *Let M be a functional PDL model. If $\mathcal{M}$ is the corresponding model of M, then M is the corresponding model of $\mathcal{M}$. Conversely, suppose that $\mathcal{M}$ is an SC model and M the corresponding model of $\mathcal{M}$. If $\mathcal{M}$ satisfies:*

1. Domain Closure Axioms for Fluents and Actions,

2. Unique Names Axioms for Fluents and Actions,

then $\mathcal{M}$ is the corresponding model of M.

6.3 Relationship between Baker's circumscription policy and PDL-model-based minimization

Since the $Holds$ function cannot be nested, not every formula in PDL can be translated into the situation calculus language. We call an action description is SC-expressible if it can be translated into situation calculus language.

Observation 3 *Let Σ be a deterministic and SC-expressible action description.*

1. $M \in min(Mod_F(\Sigma))$ iff its corresponding SC model is a model of $CIRCUM(\Sigma \cup \Psi; Ab; Result)$.

2. $\mathcal{M}$ is a model of $CIRCUM(\Sigma \cup \Psi; Ab; Result)$ if and only if its corresponding SC model in $min(Mod_F(\Sigma))$.

where Ψ is the set of UNAF, UNAA, DCAF and DCAA.

Note that the action description in the observation is not necessarily normal. However, if we impose syntactical restrictions on action description and queries, we can prove that all the solutions to the frame problem we considered thus far are equivalent. This result corresponds to the one in [14].

Corollary 1 *Let Σ be a finite normal action description, Γ a finite set of literals. If Σ is deterministic and safe, then the following statements are equivalent:*

1. $\vdash^{\Sigma \cup \Delta} (\bigwedge \Gamma) \to [a_1; \cdots; a_n]l$, where Δ is the set of frame axioms with respect to Σ.

2. For any model $M \in min(Mod_{NF}(\Sigma))$, $M \models (\bigwedge \Gamma) \to [a_1; \cdots; a_n]l$.

*3. "**necessarily** l **after** $a_1, \cdots, a_n$" is a consequence of Γ with respect to Σ.*

4. $CIRCUM(\Sigma \cup \Psi; Ab; Result) \models \forall s(Holds((\bigwedge \Gamma), s) \to Holds(l, Result(a_1, \cdots, a_n, s)))$.

where Ψ is the set of UNAF, UNAA, DCAF, DCAA and ESA.

7 Conclusion and discussion

We have encoded three typical solutions to the frame problem: Pednault's syntax-based solution, Baker's circumscription and Gelfond & Lifchitz's action language A, in PDL in either syntax or semantics. Three observations have been given which show the formal relationships among these solutions, which are helpful for a fuller and deeper understanding of the frame problem and the associated solutions. As a corollary of these observations, we know that for *finite normal and safe* action descriptions and *simple* queries, all the solutions to the frame problem are equivalent. This corresponds to the result in [14], where Pednault's, Reiter's and Baker's solutions to the frame problem were compared based on action language $\mathcal{A}$. A crucial difference between Kartha's result and ours is the following. Action language $\mathcal{A}$ is the least expressive language among the formalisms of action. Under its restrictions we cannot see the difference among the solutions (Corollary 1). In contrast, dynamic logic is the highest with regard to a certain level (propositional or first-order). This makes a systematic comparison of formalisms on action possible. Additionally, the soundness and completeness of dynamic logic bridge the syntax and semantics, which makes the unification of different approaches possible.

With help of the formal results in the paper, we would like to make the following remarks:

Syntactical restrictions: The equivalence among the solutions to the frame problem relies heavily on the syntactical restrictions on action description and queries. For instance, if Σ is not normal, the validity of a formula A in all the natural saturated Σ-models does not guarantee $\vdash^{\Sigma} A$. Thus the link between the Σ-provability in PDL and provability from transition systems of action language $\mathcal{A}$ will not exist. Additionally, the form of queries is also crucial to the equivalence. Fortunately, the link between minimizing PDL models and minimizing SC models does not depend on the normality of action description.

Extensibility of action formalisms: Each formalism of action has been or is intended to be extended to accommodate *non-deterministic and indirect effects* of actions and *compound and concurrent* actions. Compatible extensions of these formalisms will approximate dynamic logic in expressiveness. For instance, to extend $\mathcal{A}$ to express general queries requires transition systems to allow "non-natural" models according to Proposition 2. Currently, to express programs or compound actions, dynamic logic might be the best formalism among the existent ones.

Epistemic minimization and physical minimization: We know that Baker's circumscriptive policy (varying $Result$) corresponds exactly to the minimization of PDL models. We may remember that we took a detour, varying $Holds$, before we reached the "right solution": state-minimization[24]. Such a detour does not seem necessary in PDL models or transition systems. There is a subtle difference between circumscriptive first-order models and minimizing PDL models. With circumscription we minimize abnormality whereas in PDL we minimize change of worlds. We refer to the former kind of minimization as to be *epistemic* and the latter as to be *physical*.

Action-oriented frame problem: We have considered Pednault's syntax-based solution and Baker's model-based solution to the frame problem. However, these solutions only work for the so-called *fluent-oriented frame problem* (see [13]). A remaining challenge is to encode the solutions to the action-oriented frame problem, i.e, how to make it the default in PDL that only actions mentioned in the action description have effects. A typical approach to the action-oriented frame problem is using action variables to range over all actions which have effects. A compact representation of frame axioms can then be offered by using the *Explanation Closure Assumption* and quantifying over action variables [23, 13]. Such an approach cannot be encoded in PDL because there are no action variables (even in first-order versions). In [5], it was shown that given a normal and safe action description Σ, if $\vdash^{\Sigma \cup \Delta} A$, where Δ is the set of all the frame axioms with respect to Σ, then there is a subset Δ' of Δ such that all the action symbols occur in Δ' occur in A. This means that if we postpone listing frame axioms till a query arises, frame axioms in which the actions are irrelevant to the query are not needed for answering the query. Therefore, the action-oriented frame problem is not a problem in this sense.

Appendix: Translations between languages

We now provide an intertranslation between dynamic logic, situation calculus and action languages. This intertranslation is not formal. For instance, a fluent symbol stands for a proposition in PDL but is an individual in situation calculus. Again, $Holds(\varphi, S_0)$ make sense only in the extended predicate of $Holds$. Additionally, all these translations depend on the semantics of the associated action logics.

1. Expressions for describing initial state:

Dynamic Logic	*Situation Calculus*	*Action Language* $\mathcal{A}$
f	$Holds(f, S_0)$	**now** f
$\neg f$	$\neg Holds(f, S_0)$	**now** $\neg f$
φ	$Holds(\varphi, S_0)$	

2. Expressions for describing queries

Dynamic Logic	*Situation Calculus*	*Action Language* $\mathcal{A}$
$[a_1, \cdots, a_n]\varphi$	$Holds(\varphi, Result([a_1, \cdots, a_n], s))$	φ **after** $a_1, \cdots, a_n$

3. Expressions for describing causation between propositions

Dynamic Logic	*Situation Calculus*	*Action Language* $\mathcal{B}$
$[\varphi]\psi$	$Holds(\varphi, s) \rightarrow Caused(\psi, true, s)$	ψ **if** φ

4. Expressions for describing domain axioms:

Dynamic Logic	*Situation Calculus*	*Action Language* $\mathcal{A}$*or* $\mathcal{C}$
$\varphi \rightarrow [a]l$	$\forall s(Holds(\varphi, s) \rightarrow Holds(L, Result(a, s)))$	a **causes** l **if** φ
$\varphi \rightarrow \prec a \succ l$		a **may cause** l **if** φ
$\varphi \rightarrow \langle a \rangle \top$	$\forall s(Holds(\varphi, s) \rightarrow Poss(a, s))$	**executable** a **if** φ

References

1. A. Baker, Nonmonotonic reasoning in the framework of situation calculus, *Artificial Intelligence*, 49:5-23, 1991.
2. M.A. Castilho, O. Gasquet, and A. Herzig, Formalizing action and change in modal logic I: the frame problem, *J. of Logic and Computations*, 5(9):701-735, 1999.
3. R. Fikes and N. Nilsson, STRIPS: a new approach to the application of theorem proving to problem solving, In: P*roceedings of the 2nd International Joint Conference on Artificial Intelligence*, William Kaufmann, 608-620, 1971.

4. N. Foo and D. Zhang, Dealing with the ramification problem in extended propositional dynamic logic, To appear in F. Wolter, H. Wansing, M. de Rijke, and M. Zakharyaschev eds, *Advances in Modal Logic, Volume 3*, CSLI Publications, 2001.
5. N. Foo and D. Zhang, Lazy-formalization to the frame problem, manuscript. (available at http://www.cse.unsw.edu.au/~ksg/Pubs/ksgworking.html).
6. M. Gelfond and V. Lifschitz, Representing actions and change by logic programs, *Journal of Logic Programming*, vol. 17, no. 2,3,4, 301–323, 1993.
7. M. Gelfond and V. Lifschitz, Action languages, *Electronic Transactions on AI*, 16(3), 1998.
8. G. Giacomo and M. Lenzerini, PDL-based framework for reasoning about actions, In M. Gori and G. Soda (Eds.), *Topics in Artificial Intelligence*, LNAI 992,103-114, 1995.
9. M. L. Ginsberg, *Nonmonotonic Reasoning* , Morgan Kaufmann, 1988.
10. L. Giordano, A. Martelli, and C. Schwind, Dealing with concurrent actions in modal action logics, *ECAI'98*, 537-541, 1998.
11. L. Giordano, A. Martelli and C. Schwind, Ramification and causality in a modal action logic, *J. Logic Computat.* 5(10), 615-662, 2000.
12. S. Hanks and D. McDermott, Nonmonotonic logic and temporal projection, *Artificial Intelligence*, 33(3):379-412, 1987.
13. F. Lin and Y. Shoham, Provably correct theories of action, *AAAI-91*, 349-354, 1991.
14. G. Kartha, Soundness and completeness theorems for three formalization of action, *IJCAI-93*, 724-729, 1993.
15. G. Kartha, V. Lifschitz: Actions with indirect effects (Preliminary Report), *KR'94*, 341-350, 1994
16. G. Kartha: On the range of applicability of Baker's approach to the frame problem. *AAAI-96* ,664-669, 1996.
17. R A. Kowalski and M. J. Sergot, A logic-based calculus of events, *New Generation Computing*, Vol. 4, 67-95, 1986.
18. D. Kozen and J. Tiuryn, Logics of programs, In J. van Leeuwen ed. *Handbook of Theoretical Computer Science*, Elsevier, 789-840, 1990.
19. J. McCarthy and P. Hayes, Some philosophical problems from the standpoint of artificial intelligence, In B. Meltzer and D. Michie edits, *Machine Intelligence* 4, Edinburgh University Press, 463-502, 1969.
20. E. Pednault, ADL: exploring the middle ground between STRIPS and the situation calculus, *KR-89*, 324-332.
21. Peleg, D. Concurrent dynamic logic, *J. ACM*, 34(2):450-479,1987.
22. H. Prendinger and G. Schurz, Reasoning about action and change: A dynamic logic approach, *J. of Logic, Language, and Information*, 5:209-245, 1996.
23. 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*, Academic Press, 1991, 359-380.
24. M. Shanahan, *Solving the frame problem: a mathematical investigation of the common sense law of inertia*, The MIT Press, 1997.
25. D. Zhang and S. Chopra, Consistency analysis of action descriptions, in *Proceedings of Australian Workshop on Computational Logic 2001, 107-116.*
26. D. Zhang and N. Foo, EPDL: a logic for causal reasoning, to appear in *Proceedings of IJCAI-01.*