Minimal Semantics for action Specifications in PDL

Jan Broersen* Remco Feenstra*† Roel Wieringa*

Abstract

In this paper we investigate minimal semantics for Propositional Dynamic Logic formulas. The goal is to be able to write action specifications in a declarative pre/post-condition style. The declarative specification of actions comes with some well known problems: the frame problem, the qualification problem and the ramification problem. We incorporate the assumptions that are inherent to both the frame and qualification problem into the semantics of Dynamic Logic by defining preferences over Dynamic logic models. This gives us an intended semantics that, for each declarative action specification, selects a unique meaning for each action.

1 Introduction

Our main interest concerns the declarative specification of transactions in a pre-/post-condition style. a transaction can be a database transaction or any other system transaction, as long as it is taken to be atomic. The only requirement we impose on transactions is that they are terminating processes of which the intermediary states are not observable. Transactions can be nondeterministic. That is, a transaction may result in one of several possible next states.

A convenient starting point is dynamic logic [7] [8]. It is designed to reason about arbitrary programs/actions (in the propositional case) or programs built up from specific actions such as assignments (in the first order case). Dynamic Logic is a multi modal logic in which each action is accompanied by its own modal operators.

If we try to make specifications in PDL we run into three well known problems: the frame problem, the qualification problem and the ramification problem. The frame problem concerns the fact that we want to leave as an assumption that anything that is not specified to change does not change. The qualification problem can also be seen to deal with an assumption; the assumption that actions actually occur unless it is explicitly specified that they do not.

In this paper we define preferential semantics for PDL formulas that incorporate these assumptions. The preference relation (ordering) that is concerned with the frame assumption is based on a notion of 'difference' between states. The preference relation guarantees that an action is interpreted always to lead to a minimally differing state, thus implementing the assumption that no state elements change that are not specified to change. The qualification assumption is implemented by defining a second ordering over PDL-models.

[1] Faculty of Mathematics and Computer Science, Vrije Universiteit, De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands, email: broersen|rbfeens|roelw@cs.vu.nl.

[2] This research is partially supported by the Netherlands Organization for Scientific Research (NWO), SION project 612–317–408.

This ordering is based on a notion of 'maximal reachability' between states, meaning that actions take place unless the specification contains necessary preconditions (guards) or static constraints that contradict this. Taking the maximal elements (models) over this ordering guarantees that actions are interpreted under the qualification assumption. Both orderings over PDL-models can be applied successively, provided this is done in the right order (minimization first). The orderings are also applicable in the presence of static constraints, thus implementing that both assumptions also apply to ramifications.

The structure of the paper is as follows. In Section 2 we introduce Propositional Dynamic Logic (PDL) and define a semantics for specifications in PDL. In Section 3, we discuss the frame, qualification and ramification problems in more detail and indicate what their interconnections are. Section 4 then proposes several preferential semantics for PDL specifications. Section 5 compares our approach with other approaches. Section 6 concludes the paper and lists some topics for further research.

2 Propositional Dynamic Logic

Propositional dynamic logic (PDL) is a logic to reason about terminating programs. It relates assertions about composite programs to assertions about its parts and vice versa. The modal formulas $\langle a\rangle\phi$, where a is an action, mean that there is a possible occurrence of a after which ϕ is true. The standard PDL-semantics of actions as relations over states is given in the next Section. We also consider compound actions; $a;\beta$, $a \cup \beta$, $a?$ and a^* represent sequential composition, choice, test and iteration, respectively. The semantics of these constructs is of importance when proving properties of specifications. These constructs are however not used for the writing of specifications.

2.1 The PDL language

Definition 1 *The language is built with the following elements:*

Punctuation brackets:
the brackets ')', '(' and the comma ','

Proposition symbols:
a countable set P of proposition symbols

Propositional connectives:
the set of symbols $\{\neg, \vee\}$

Atomic actions
a countable set T of action symbols

Action connectives:
the set of symbols $\{\cup, \ ^*, ;\}$

Combined connectives:
the set of operation symbols $\{\langle\rangle, ?\}$

The intended use of the logic is the writing of system specifications. A specific specification uses only a finite subset of the language. A specifier has to give this subset in a signature.

Definition 2 *A **signature** Σ is thus defined as a specific combination of finite subsets of the proposition and atomic action symbols: $\Sigma = (\mathcal{P}, \mathcal{T})$.*

The proposition symbols are referred to by A, B, ... Atomic action symbols have associated meta-variables a, b, ... A signature contains the symbols that are to be given an interpretation relative to a specification. All other symbols are given logical interpretations.

Definition 3 *An **action** (α) over a signature Σ is defined as:*

$$\alpha \quad ::= \quad \textit{atomic action symbol} \mid \alpha \cup \beta \mid \alpha ; \beta \mid \alpha^* \mid \phi ?$$

*A **well formed formula** (ϕ) over a signature Σ is defined as:*

$$\phi \quad ::= \quad \textit{proposition symbol} \mid \textit{true} \mid \textit{false} \mid \neg\phi \mid \phi \vee \psi \mid \langle\alpha\rangle\phi$$

We use the term "compound action" for actions that contain action connectives. As usual, we abbreviate $\neg(\neg\phi \vee \neg\psi)$ to $\phi \wedge \psi$, $\neg\phi \vee \psi$ to $\phi \rightarrow \psi$, $(\phi \rightarrow \psi) \wedge (\psi \rightarrow \phi)$ to $\phi \leftrightarrow \psi$ and $\neg\langle\alpha\rangle\neg\phi$ to $[\alpha]\phi$.

A **specification** $\mathit{Spec} = (\Sigma, \Phi)$ is a pair consisting of a signature Σ and a finite set Φ of formulas over Σ.

2.2 The semantics of PDL

PDL formulas are interpreted over a special kind of Kripke structures.

Definition 4 *Given a signature $\Sigma = (\mathcal{P}, \mathcal{A})$, a structure $(W, \mathcal{I}_A)$ is defined as follows:*

- $W \subseteq 2^{\mathcal{P}}$
- *$\mathcal{I}_A$ is a total function $\mathcal{A} \rightarrow 2^{W \times W}$, equivalently $\mathcal{I}_A(a) \subseteq W \times W$*

We identify worlds with possible interpretations of the atomic formulas. So we do not discriminate between worlds and interpretations of the set of atomic formulas. Therefore we do not need an interpretation function to interpret atomic formulas in worlds. The distinction between worlds and interpretations is only useful if we want to consider different worlds with equal interpretations of atomic formulas or different interpretations of atomic formulas in the same world. For our purposes this restricted notion of a structure suffices.

$\mathcal{I}_A$ says how actions are interpreted as transitions between possible worlds by mapping atomic action symbols to binary relations over W.

Elements of $2^{\mathcal{P}}$ are referred to as w, w', ...

The following definition shows how PDL formulas are interpreted over these structures.

First we define validity of a formula in a world w of a structure $S = (W, \mathcal{I}_A)$.

Definition 5 *Validity of a formula ϕ in a world w of a structure $S = (W, \mathcal{I}_A)$, denoted by $S, w \models \phi$ is defined as:*

$$\begin{array}{lll}
S,w \models true & & \\
S,w \models P & \textit{iff} & P \in w \\
S,w \models \phi \vee \psi & \textit{iff} & S,w \models \phi \textit{ or } S,w \models \psi \\
S,w \models \neg\phi & \textit{iff} & S,w \not\models \phi \\
S,w \models \langle a \rangle \phi & \textit{iff} & \textit{for some } w' \in W \textit{ holds } (w,w') \in \mathcal{I}_A(a) \textit{ and } S,w' \models \phi \\
S,w \not\models false & &
\end{array}$$

Compound actions are not used in specifications. However, compound actions may be used in formulas that express properties we want to be able to prove of a specification. Therefor we also give the semantics of compound actions. ($\langle\alpha\rangle^n\phi$ is defined as $\langle\alpha\rangle\langle\alpha\rangle^{n-1}\phi$ for $n > 1$, $\langle\alpha\rangle^1\phi$ as $\langle\alpha\rangle\phi$ and $\langle\alpha\rangle^0\phi$ as ϕ)

Definition 6

$$\begin{array}{lll}
S,w \models \langle \psi ? \rangle \phi & \textit{iff} & S,w \models \psi \wedge \phi \\
S,w \models \langle \alpha ; \beta \rangle \phi & \textit{iff} & S,w \models \langle \alpha \rangle \langle \beta \rangle \phi \\
S,w \models \langle \alpha \cup \beta \rangle \phi & \textit{iff} & S,w \models \langle \alpha \rangle \phi \vee \langle \beta \rangle \phi \\
S,w \models \langle \alpha^* \rangle \phi & \textit{iff} & \textit{for some } n \in \{0,1,2\ldots\} \textit{ holds } S,w \models \langle \alpha \rangle^n \phi
\end{array}$$

Note that test, sequentiation and choice could actually be introduced as syntactic abbreviations. This means that iteration is responsible for the the real gain in expresivity when introducing action connectives.

A formula is S-valid if it is true in all worlds of a structure S. In that case, we say that the structure satisfies the formula and that the structure is a model of the formula. A formula is valid if it is S-valid for every structure S. Note that this is not entirely conventional. In the literature, the word "structure" is reserved for frames, consisting only of worlds and an accessibility relation, without a valuation. In our definitions worlds are identical with interpretations. What we call a structure is in literature on modal logic usually called a model. We prefer to use the word "model" for structures that satisfy a set of formulas.

3 Action specification

We investigate the use of PDL for the writing of specifications for atomic actions. We focus on two aspects: the effect and the possible occurrence of actions. For the specification of the effect of an action a we can use formulas called "conditional postcondition formulas".

- **Conditional postcondition formulas:** $\phi_i \rightarrow [a]\psi_i$ We say that ϕ_i is a sufficient precondition of a with respect to the postcondition ψ_i.

The possible occurrence of actions can be controlled with "guard formulas".

- **Guard formulas:** $\langle a \rangle true \rightarrow \chi_j$ We call χ_j a guard of a, equivalently, a necessary precondition for the possible occurrence of a.

Guard formulas actually can be seen as a special case of conditional postcondition formulas. This is however of no importance. What is important is that we can express guards (necessary preconditions for actions) in PDL. Furthermore it is important to investigate guards independently because they are strongly interconnected with the qualification problem.

We also want to investigate the influence of "static constraints" (formulas without modalities) on the effect and possible occurrence of actions.

- **Static constraints:** θ_k These are assertions that must be obeyed under any circumstance.

The formulas ϕ_i, ψ_i, θ_k and χ_j contain no modal constructs; they are just propositional logic formulas. To simplify the setting we assume that these are the only formulas a specifier uses when stating a specification. This follows the syntactic restrictions of LCM [5]. It is motivated by the fact that we simply don't need intricately nested modalities to specify actions declaratively.

Definition 7 *An action specification axiom is either a postcondition formula, a guard formula or a static constraint. We use Φ_{spec} to indicate an arbitrary set of action specification axioms.*

As it appears each of the above three (groups of) formulas is associated with a (well known) problem; the postcondition formulas with the frame problem, the guard formulas with the qualification problem and the constraints with the ramification problem.

The standard PDL-interpretation of postcondition formulas, is that when ϕ_i is true, action a, if present, leads to a situation where ψ_i is true. The formulas describe the effect of an action. However, defining the effect of an action by means of a postcondition always causes **the frame problem** [3]. This problem states that when specifying a postcondition we only want to specify conditions that have changed. We do not want, and often are not able, to specify all the conditions that do not change as the result of an action. So we want to make the frame assumption that everything that has not been specified to change has not changed. However, this is not reflected by the semantics, as defined in 2.2.

The interpretation of guard formulas is that action a can only take place when χ_j is true. Thus χ_j is a necessary precondition for the possible occurrence of a. So guard formulas control the possible occurrence of actions. The problem associated with the possible occurrence of actions is **the qualification problem**. This problem states that it is not possible to foresee all necessary preconditions for the success of an action [6]. This means that a specifier actually is never able to give a sufficient precondition for an action. Of course he may want to give a sufficient precondition anyway, but he then must face the fact that he may end up with an inconsistent specification because the action he gave a sufficient precondition for may violate constraints or may have contradictory postcondition axioms associated to it. A possible solution to this problem is to weaken sufficient preconditions by specifying them as defaults [2] [10]. We take the perspective that a specifier should only be allowed to give necessary preconditions (guards) and that the assumption that actions occur unless this contradicts guards (or static constraints, or conflicting postcondition axioms) is somehow implemented in the semantics. The standard PDL-semantics does not provide this. It is easily seen that there are interpretations of action specification axioms in which actions do not occur even if the guards are true. In particular the structure where the accessibility relation between possible worlds is completely empty always satisfies a set of action specification axioms.

Static constraints influence both effect and possible occurrence of actions. The possible occurrence of actions is influenced because static constraints may 'cut out' all the worlds a transition would have gone to in case constraints where not given. In [9] this particular situation is referred to by the statement that constraints can play the role of 'implicit axioms about action preconditions'. In the context of the updating of databases this is sometimes seen as the only way to interpret static constraints: if a database transaction violates a constraint then 'role back'. Under this view constraints are interpreted as mere limitations on the state space. If we adhere a more 'logical' interpretation to constraints, they influence the effects of actions. Under a standard logical interpretation constraints give rise to ramifications (derived effects). On the level of worlds this is explained as follows: if a constraint cuts out worlds, then transitions may go to other worlds that do comply with the constraints, thus giving rise to a ramification. So constraints may give rise to derived effects. But then for these derived effects also the qualification assumption should hold. How to deal with derived effects is referred to in the literature as **the ramification problem**.

In the following we will define semantics that incorporate all the desired properties mentioned above.

4 Minimal change/maximal reachability semantics

In this section we implement the frame assumption by formalizing the notion of minimal change between possible worlds and we implement the qualification assumption by formalizing the notion of maximal reachability. To achieve this we define orderings over PDL-models to sort out the intended ones. The difference with circumscription [14] is that we do not define this ordering with respect to particular predicate(s) (proposition(s)).

4.1 Minimal change

Because we want to compare models on the change that actions accomplish, we need a notion of difference (distance) between states.

Definition 8 *Given a signature $\Sigma = (\mathcal{P}, \mathcal{A})$ and a structure $S = (W, \mathcal{I}_A)$ and two worlds w and w' in W. The difference $Diff(w, w')$ between them is defined as the set of atoms in which the worlds differ:*

$$Diff(w, w') \equiv (w \setminus w') \cup (w' \setminus w)$$

Using this definition we can define orderings over structures that capture the notion of minimal change between possible worlds. We will now first give two slightly different orderings, before we explain what they are really about. (The domain of a binary relation, $dom(R)$ is defined as $\{x \mid (x, y) \in R\}$)

Definition 9 *Given a signature $\Sigma = (\mathcal{P}, \mathcal{A})$ and two structures $S = (W, \mathcal{I}_A)$ and $S' = (W', \mathcal{I}'_A)$,*

$S' \sqsubseteq_{mc} S$ *iff*
$W' = W$
and
$\forall a \in \mathcal{A},\ dom(\mathcal{I}'_A(a)) = dom(\mathcal{I}_A(a))$

and

$$\forall a \in \mathcal{A},\ \forall w \in W,\ \forall w' \in W,\ ((w,w') \in \mathcal{I}'_A(a) \Rightarrow \exists w'' \in W,\ ((w,w'') \in \mathcal{I}_A(a) \wedge N(\mathit{Diff}(w,w')) \leq N(\mathit{Diff}(w,w''))))$$

$\sqsubseteq_{mc}$ is a pre-order on structures, because $\sqsubseteq_{mc}$ can easily seen to be transitive and reflexive. MC stands for minimal cardinality. A structure is called an MC-model of Φ if it is a $\sqsubseteq_{mc}$-minimal model of Φ. MC-models determine the minimal cardinality interpretation (semantics) of a specification (Σ, Φ).

Definition 10 *Given a signature* $\Sigma = (\mathcal{P}, \mathcal{A})$ *and two structures* $S = (W, \mathcal{I}_A)$ *and* $S' = (W', \mathcal{I}'_A)$,

$S' \sqsubseteq_{ms} S$ *iff*

$W' = W$

and

$\forall a \in \mathcal{A},\ \mathit{dom}(\mathcal{I}'_A(a)) = \mathit{dom}(\mathcal{I}_A(a))$

and

$$\forall a \in \mathcal{A},\ \forall w \in W,\ \forall w' \in W,\ ((w,w') \in \mathcal{I}'_A(a) \Rightarrow \exists w'' \in W,\ ((w,w'') \in \mathcal{I}_A(a) \wedge \mathit{Diff}(w,w') \subseteq \mathit{Diff}(w,w''))))$$

$\sqsubseteq_{ms}$ is a pre-order on structures, because $\sqsubseteq_{ms}$ can easily seen to be transitive and reflexive. MS stands for minimal subset. A structure is called an MS-model of Φ if it is a $\sqsubseteq_{ms}$-minimal model of Φ. MS-models determine the minimal subset interpretation (semantics) of a specification (Σ, Φ).

The first requirement in both $\sqsubseteq_{mc}$ and $\sqsubseteq_{ms}$ ordering is that structures can only be compared if they are based on the same set of worlds. In figure 1, where three models are compared on minimal change, this is reflected by the fact that all models contain the same set of black dots. This is actually not a necessary condition for the definition of the semantics, but it helps to make the definition more understandable. Models with exactly the same state transitions but differing in worlds that 'stand alone', can not be compared. If we would drop this condition such worlds would be comparable, but one could never be prefered above the other. The question what worlds are actually in the intended model of a specification is dealt with by a second ordering, to be defined in 4.2.

The second requirement in both orderings forces that structures can only be compared if the same transitions (actions) occur in the same worlds, as is also the case in figure 1. (There can be difference in to which worlds transitions lead. This is actually where we want to compare structures on; we want to prefer structures where transitions lead to closer worlds.) This is an intuitive criteria because we don't want this ordering to deal with the possible occurrence of transitions (actions); this ordering should compare structures purely on the 'length' of transitions. (This observation is also made in [2] [10].)

The last requirement deals with this 'length' of transitions. In words it says: if $S' \sqsubseteq_{mc} S$ then for all transitions (w, w') in S' there is a transition (w, w'') in S that is 'longer'. In yet other words: if $S' \sqsubseteq_{mc} S$ then for corresponding transitions in corresponding worlds in the compared structures, the 'longest' transition in S' is always less or equal to the 'longest' transition in S. (To see why the ordering is not a partial order: when the 'longest' transitions from the same worlds in both structures are equal, still both structures can differ in transitions that are 'shorter' then these 'longest' transitions.) In figure 1 the distance between dots represents the difference between worlds. Clearly the minimal one

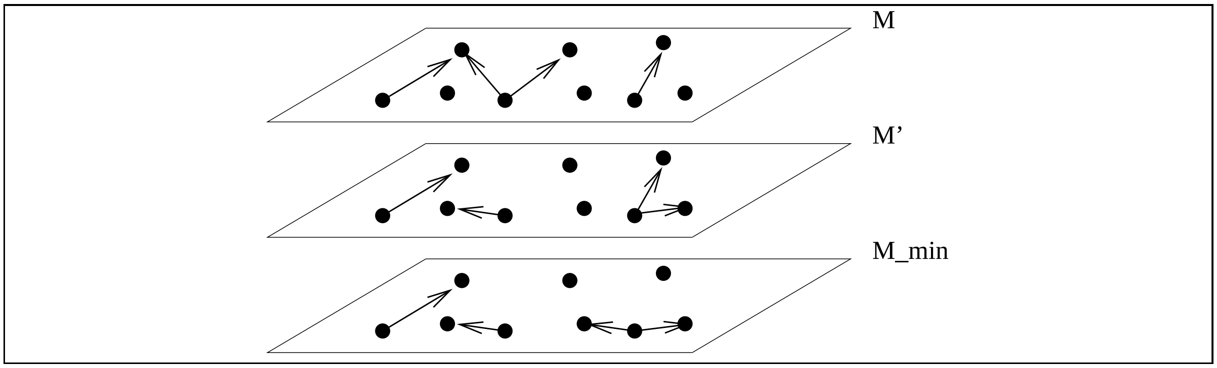


Figure 1: Comparing models on minimal change

is the one for which the property holds. The reason why this looks rather complicated is that we allow non-deterministic actions; we have to compare structures in where actions from one world can lead to several other worlds. In 4.3 we will see that for deterministic models orderings can be defined much simpler.

It is important that minimal models exist if 'ordinary' models do. Therefore we prove:

Proposition 1 *Let Φ be a set of PDL-formulas that has a model $S = (W, \mathcal{I}_A)$ for which $\exists a$, $\mathcal{I}_A(a) \neq \emptyset$. Then there is an MS-model and an MC-model of Φ for which $\exists a$, $\mathcal{I}_A(a) \neq \emptyset$.*

Proof Let S be a model of Φ for which $\exists a$, $\mathcal{I}_A(a) \neq \emptyset$. It follows from the finiteness of the number of propositions and actions that there are only finitely many models. Then either S is already minimal, or there is a model S' that is minimal and $S' \sqsubseteq_{ms} S$ ($S' \sqsubseteq_{mc} S$) and not $S \sqsubseteq_{ms} S'$ ($S \sqsubseteq_{mc} S'$). In the latter case S' can not be a model that 'contains no transitions' because it follows from the definitions of the orderings that in that case S' can not be compared with S.

■

The difference between the semantics generated by both orderings will become more clear when we look at an example in 4.5. But we can say already that the minimal subset semantics is weaker than the minimal cardinality semantics, as is expressed by the following proposition.

Proposition 2 *An MC-model of Φ is also an MS-model of Φ.*

Proof Follows directly from the fact that from $\mathit{Diff}(w, w'') \subseteq \mathit{Diff}(w, w'')$ it follows that $N(\mathit{Diff}(w, w'')) \leq N(\mathit{Diff}(w, w''))$.

■

The correspondence between both semantics is best shown by focusing on a special class of formulas for which both semantics coincide. This is also a class of formulas for which both MS-models and MC-models are deterministic. This is actually an important class. Because the minimal (intended) models are deterministic for these formulas, the only intention a specifier can have when providing formulas of this form, is to specify a deterministic system. Interpreting the formulas under a non-minimal semantics would allow for non-deterministic interpretation of the formulas, which is not what is intended.

Definition 11 *A structure $S = (W, \mathcal{I}_A)$ is* deterministic *if for each world $w \in W$ and for each a there is maximally one world w' such that $(w, w') \in \mathcal{I}_A(a)$.*

Definition 12 *Description of the class Φ_{det} of determinate action specification formulas.*

$$\begin{array}{ll} postconditionformulas: & \phi \rightarrow [a](L_1 \wedge L_2 \wedge \ldots \wedge L_k) \\ guards: & \langle a \rangle true \rightarrow \chi \\ constraints: & (M_1 \wedge M_2 \wedge \ldots \wedge M_l) \vee (N_1 \wedge N_2 \wedge \ldots \wedge N_m) \end{array}$$

with the L_h, M_i and N_j positive or negated atomic formulas (literals).

First we prove the property that for formulas of the class Φ_{det}, minus the constraints, both MS-models and MC-models are deterministic. We leave out the constraints here because otherwise the proof would become to lengthy.

Definition 13 *A structure $S = (W, \mathcal{I}_A)$ is* deterministic *if for each world $w \in W$ and for each a there is maximally one world w' such that $(w, w') \in \mathcal{I}_A(a)$.*

Proposition 3 *For a specification (Σ, Φ_{det}) (minus the constraints) MS-models are deterministic.*

Proof

Assume that there is an MS-model S of Φ_{det} that is not deterministic. Then we have that $\exists a \in \mathcal{A}$, $\exists w \in W$, $\exists w' \in W$, $\exists w'' \in W$, $(w, w') \in \mathcal{I}_A(a) \wedge (w, w'') \in \mathcal{I}_A(a)$

Now let $\{L_{i1}, \ldots, L_{in}\}$ be the set of literals that appear in the determinate postcondition formulas $\phi_i \rightarrow [a](L_{i1} \wedge L_{i2} \wedge \ldots \wedge L_{in})$ in Φ_{det} for which $S, w, \mathcal{I}_V \models \phi_i$. Because S is a model, the atoms in $\{L_{i1}, L_{i2}, \ldots, L_{in}\}$ have the same valuation in both w'' and all w'. This means that w'' differs from w in the same atoms among $\{L_{i1}, \ldots, L_{in}\}$, as w' does. This means that w'' and w' differ in at least one atom, say A_d not among the atoms in $\{L_{i1}, L_{i2}, \ldots, L_{in}\}$. Without loss of generality we assume that w' is the world where A_d has an interpretation different from its interpretation in w and w'' is the world where A_d has an interpretation equal to its interpretation in w. Now we can construct the model S' that is equal to S' except for that the transition (w, w') is left out. Obviously S' is below S in the $\sqsubseteq_{ms}$-ordering, because the transition w, w' that changes more then w, w'' is left out. But this contradicts that S is an MS-model.

■

Proposition 4 *For a specification (Σ, Φ_{det}) MC-models are deterministic.*

Proof The proof is analogous to the former one.

■

Proposition 5 *For a specification (Σ, Φ_{det}) the minimal subset and minimal cardinality semantics coincide.*

Proof We already proved that an MC-model is always an MS-model. We also proved that for determinate specification formulas MS-models and MC-models are deterministic. It's not difficult to see that this implies that for determinate specification formulas, MS-models are also MC-models. In both MS-models and MC-models we can have only one transition from each state in the model and each MC-model is an MS-model, then each

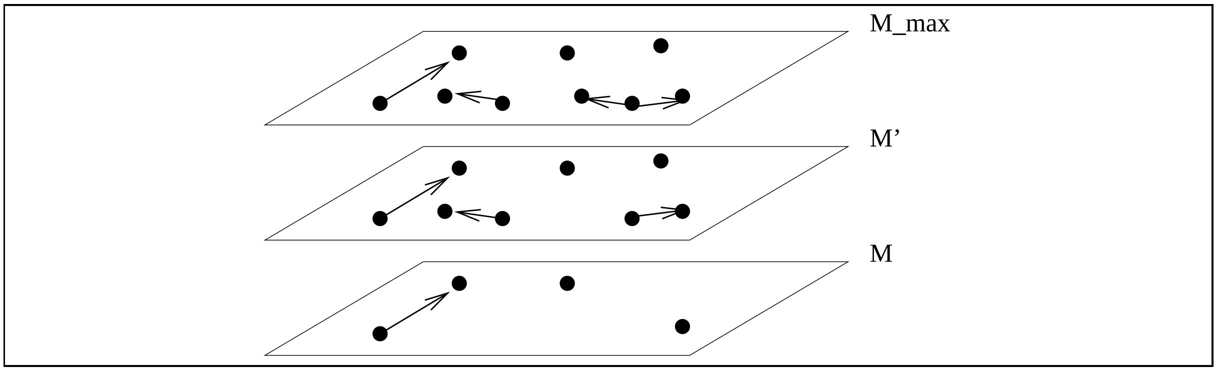


Figure 2: Comparing models on maximal reachability

MS-model is also an MC-model.

■

The minimal interpretation of sets of formulas results in a non-monotonic notion of entailment. As an example of this take the set of formulas $\Phi = \{A \to [a]B\}$. All structures that MS-satisfy (MC-satisfy) Φ also satisfy the formula $[a]A \wedge B$. So, under these interpretations, Φ preferentially entails $[a]A \wedge B$. But the entailment no longer holds for $\Phi \cup \{\langle a \rangle \neg A\}$.

4.2 Maximal reachability

Guards are necessary preconditions, so they do not 'force' transitions. In this section we look for interpretations that tend to interpret the guards specified for an action a as sufficient, provided that the possible occurrence of a is compatible with the constraints and postconditions in the postcondition axioms. We accomplish this by formalizing the qualification assumption in the notion of maximal reachability.

Definition 14 *Given a signature $\Sigma = (\mathcal{P}, \mathcal{A})$ and two structures $S' = (W', \mathcal{I}'_A)$ and $S = (W, \mathcal{I}_A)$*

$$\begin{aligned} &S' \sqsubseteq_{mr} S \textit{ iff} \\ &\quad W' \subseteq W \\ &\quad \textit{and} \\ &\quad \textit{for all } a \in \mathcal{A}, \mathcal{I}'_A(a) \subseteq \mathcal{I}_A(a) \end{aligned}$$

$\sqsubseteq_{mr}$ is a partial order on structures, because $\sqsubseteq_{mr}$ can easily seen to be transitive, reflexive and anti-symmetric. The ordering just 'prefers' as much worlds and transitions over them as possible, thus implementing the notion of 'maximal reachability'. In Figure 2 this is reflected by the fact that all transitions and worlds in lower models are also in the model at the top.

Proposition 6 *Let Φ be a set of formulas for which there is an model. Then there is a $\sqsubseteq_{mr}$-maximal, model.*

Proof follows from the fact that there are only finitely many different MS-models of a set of formulas Φ and the fact that $\sqsubseteq_{mr}$ is a partial order on them.

■

Preferring $\sqsubseteq_{mr}$-maximal structures leads to yet another interpretation of formulas. Again, this interpretation results in non-monotonic entailment. An example of this is provided by the set $\Phi = \{[a]A\}$. Under interpretation over $\sqsubseteq_{mr}$-maximal structures, $\langle a\rangle A$ is entailed by Φ. However, this is no longer true for $\Phi \cup \{[a]\neg A\}$.

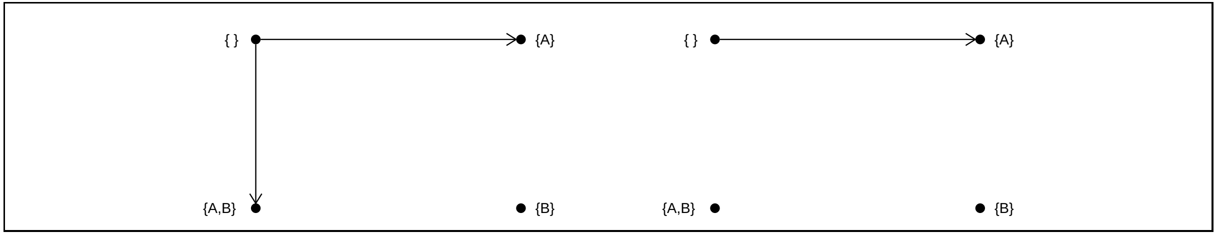
4.3 Min-Max-models

For the interpretation of specifications under both frame and qualification assumption we have to combine orderings. We do this by applying them one after the other. What is the right (intuitive) order in which to do this? The answer can be found by looking at what the orderings are supposed to represent. The minimal change orderings concern the effect of actions independent from whether they occur or not. Maximal reachability deals with possible occurrence. It is intuitive then to apply minimal change first. The minimal change ordering determines what action we actually mean by 'determining' their effects. After that we can 'talk' about the possible occurrence of actions. This motivates the following definition.

Definition 15 *Given a specification* (Σ, Φ)*, a Min-Max-model is defined as a* $\sqsubseteq_{mr}$*-maximal element of the set of MS-models of* Φ*.*

(Of course we can make a similar definition with MC-models. Actually, in the rest of this paper "MS-model" can at any time be replaced by MC-model, resulting in a somewhat different semantics.) The next example shows that defining this in reverse order does not provide the semantics we are looking for.

We take a signature with $\mathcal{P} = \{A, B\}$, and $\mathcal{A} = \{a\}$, and the following set of formulas: $\Phi = \{[a]A, \langle a\rangle true \rightarrow \neg A \wedge \neg B\}$.



The left picture shows the $\sqsubseteq_{mr}$-maximal structure that satisfies Φ. This structure is not MS-satisfying. This means it is not useful to apply maximality and minimality in this order. There are only two MS-models of the example formulas, the structure with no access to other worlds at all and the one shown in the right picture. Clearly the one in the picture is $\sqsubseteq_{mr}$-maximal. We show in refexample that this construction also works in case there are ramifications.

The MS-models of a set of formulas Φ (we mean general formulas here) form a partially ordered set. This set is not a lattice, as is shown by the following example. Take the formula $\neg(\langle a\rangle A \wedge \langle a\rangle B)$ and the two MS-satisfying structures S with $\mathcal{I}_A$ is $\{a \rightarrow (\{\}, \{A\})\}$ and S' with $\mathcal{I}'_A$ is $\{a \rightarrow (\{\}, \{B\})\}$. There is no MS-satisfying structure that is an upper bound for both structures S and S'. The structure S'' with $\mathcal{I}''_A$ is $\{a \rightarrow (\{\}, \{A\}), a \rightarrow (\{\}, \{B\})\}$ is an upper bound under the $\sqsubseteq_{mr}$-ordering, but is not MS-satisfying (not even satisfying). So in general there can be more Min-Max-models of a set of formulas Φ. However, for specification formulas as defined in 3, we can prove that the MS-models do form a complete lattice. Therefore, the following proposition holds.

Proposition 7 *Each specification* Φ_{spec} *has a unique Min-Max-model if it has a model.*

Proof We must prove that the MS-satisfying structures of a specification Φ_{spec} form a complete lattice under the $\sqsubseteq_{mr}$-ordering. For this it is sufficient to prove that they form a lattice, because the set is finite. To prove that the set forms a lattice we define the *lub* of two MS-satisfying structures $S = (W, \mathcal{I}_A)$ and $S' = (W', \mathcal{I}'_A)$ as $S'' = (W'', \mathcal{I}''_A)$ with $W'' = W \cup W'$ and $\mathcal{I}''_A$ such that $\forall a \in A,\ \mathcal{I}''_A(a) = \mathcal{I}'_A(a) \cup \mathcal{I}_A(a)$ and the *glb* as $S''' = (W, \mathcal{I}'''_A)$ with $W'' = W \cap W'$ and $\mathcal{I}'''_A$ such that $\forall a \in A,\ \mathcal{I}'''_A(a) = \mathcal{I}'_A(a) \cap \mathcal{I}_A(a)$. It is not difficult to see that both *lub* and *glb* are MS-satisfying structures of Φ_{spec}.

■

For deterministic structures (models) a Min-Max model can be defined with the help of one, much simpler ordering $\sqsubseteq_{mm}$.

Definition 16 *Given a signature $\Sigma = (\mathcal{P}, \mathcal{A})$ and two structures $S = (W, \mathcal{I}_A)$ and $S' = (W', \mathcal{I}'_A)$,*

$$
\begin{aligned}
&S' \sqsubseteq_{mm} S \textit{ iff}\\
&\quad W' \subseteq W\\
&\quad \textit{and}\\
&\quad \forall a \in \mathcal{A},\ \forall w \in W,\ (\exists w'' \in W,\ (w, w'') \in \mathcal{I}_A(a) \Rightarrow \exists w' \in W',\ ((w, w') \in \mathcal{I}'_A(a)\\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \wedge \mathit{Diff}(w, w') \subseteq \mathit{Diff}(w, w'')))
\end{aligned}
$$

It is not difficult to see that the above (partial) ordering combines the notion of maximal reachability and minimal change. It can be proven that for specifications with determinate postcondition formulas, guards and static constraints, the minimal models determined by the above ordering coincide with Min-Max-models.

4.4 Static constraint interpretation

There are two alternatives for the semantics of the static constraints. Given a specification (Σ, Φ_{spec}) with $\Phi_{spec} = \Phi_{post} \cup \Phi_{guard} \cup \Phi_{cons}$, we define two models that interpret it.

- **Constraint semantics**: The $\sqsubseteq_{mr}$-maximal model of the set MSIC, with MSIC the set of MS-models of $\Phi_{post} \cup \Phi_{guard}$ that in addition satisfy Φ_{cons}
- **Ramification semantics**: The Min-Max-model of Φ_{spec}.

The ramification semantics corresponds to the standard logic interpretation of constraints and gives rise to 'derived effects'.

The constraint semantics does not give rise to derived effects (worlds), it merely 'cuts out' access to worlds in which the constraints are not satisfied. We accomplish this by first looking at minimal models of specifications without taking the constraints into account. Transitions in these minimal models possibly lead to worlds that do not comply with constraints. By applying the criterion that models also have to comply with static constraints in a second step, we lose these transitions all together.

In section 4.5 we show how the Ramification semantics extends the frame assumption and the qualification assumption to ramifications. Constraint and ramification semantics can of course be combined by splitting the sets of non-modal axioms into two set: one set to be interpreted by the constraint semantics and one by the ramification semantics. The formulas to be interpreted by the ramification semantics are called **derivation rules**. Spruit et al. [12] works this out for Propositional Dynamic Database Logic (PDDL) with Horn Clauses as derivation rules.

4.5 Example

As an example specification we take:

$$
\begin{aligned}
&Gun\_loaded\_sharp \rightarrow [fire\_gun]Gun\_blown\_up \vee Bullet\_emitted\\
&Gun\_loaded\_blank \rightarrow [fire\_gun]Gun\_blown\_up \vee Air\_and\_dust\_emitted\\
&\rightarrow [fire\_gun]Big\_noise\\
&\langle fire\_gun\rangle true \rightarrow \neg Gun\_blown\_up\\
&\langle fire\_gun\rangle true \rightarrow Gun\_loaded\_sharp \vee Gun\_loaded\_blank\\
&Bullet\_emitted \rightarrow Somebody\_is\_hit\\
&\neg(Gun\_loaded\_sharp \wedge Gun\_loaded\_blank)
\end{aligned}
$$

For the interpretation of this specification we have to choose between the several semantics we defined. First we look at the interpretation of constraints. There is only one action, $fire\_gun$. It is a nondeterministic action because some postcondition formulas contain disjunctions. One of the alternatives represented by the disjunction is $Bullet\_emitted$. This atom is also present in one of the constraints. If we interpret this constraint under the ”constraint semantics”, the condition ”Somebody_is_hit” can actually never change to true as a result of the action $fire\_gun$, it can only be true after the action $fire\_gun$ if it was already true in the world before $fire\_gun$ took place. This means the most intuitive interpretation for this constraint is the ramification semantics. We think the ramification semantics is usually the best interpretation for constraints.

Now we look at which minimality criterion to apply. We have to choose between the minimal subset and the minimal cardinality criterion. The choice is not too difficult if we look at the postcondition $Gun\_blown\_up \vee Bullet\_emitted$. Given the choice that we interpret constraints under the ramification semantics, the truth of $Bullet\_emitted$ will imply the truth of $Somebody\_is\_hit$ in resulting states. This means that the action $fire\_gun$ usually 'has the choice' between making one atom true (Gun_blown_up) or two ($Bullet\_emitted$ and $Somebody\_is\_hit$). The minimal cardinality semantics will choose the first alternative which is really counterintuitive. The minimal subset semantics just makes no choice; both successor worlds are possible. This means that the minimal subset semantics is a more non-deterministic interpretation. We think the minimal subset semantics is always the more intuitive one.

We now show how the action $fire\_gun$ can be qualified. The maximality criterion assures that in the Min-max-model of the specification, $fire\_gun$-transitions are actually there. But of course we can constrain the occurrence of transitions by adding extra formulas to the specification. Adding $\{Gun\_blown\_up\}$ or $\{\neg Big\_noise\}$ would leave us with no transitions at all. Adding $\{\neg Gun\_loaded\_sharp\}$ would result in transitions that never make $Somebody\_is\_hit$ true.

Finally we name some properties that are true in the Min-max-model (with the minimal subset / ramification criteria) of the specification and that are formulated using action connectives. The first one is:

$Gun\_loaded\_blank \rightarrow [fire\_gun^*]Gun\_loaded\_blank$

It says that if the gun was loaded with a blank it will stay loaded with blanks after a possibly infinite amount times of shooting. A second one is:

$[?Gun\_loaded\_sharp\,; fire\_gun](\neg Gun\_blown\_up \rightarrow Somebody\_is\_hit)$.

It says that if the gun is fired in a situation where it is loaded with a bullet, and as the result of it the gun will not blow up, then somebody is hit.

5 Discussion

The frame problem and the associated problems of qualification and ramification are common themes in the AI literature on knowledge representation and reasoning about action and change [3].

Reiter [11] discusses dealing with the frame problem from a database point of view by using precondition axioms and successor state axioms to specify actions in the situation calculus. Lin and Reiter [9] study the interrelationship between the ramification and qualification problems caused by interacting state constraints and action effects.

Borgida et al. [1] take the perspective of the designer of specification languages and discuss ways to state that "nothing else changes" by syntactic as well as semantic means. Our work can be regarded as introducing a richer semantics for the specification language to capture this.

On the other side of the spectrum, Winslett's work on database update semantics [13] focuses on a model-oriented approach to updates, de-emphasizing the relation between the specification and the models. Instead, we base our semantics on the declarative semantics of a specification in PDL, which allows us to reason about updates in the same language.

Brass and Lipeck [2] [10] study action specification with the help of defaults. They also define orderings over modal interpretations. Frame and other assumptions are represented by formulas interpreted as defaults. This still puts the responsibility on the specifier to provide such formulas, which does not always seem desirable. Furthermore their models represent 'action traces' and do not allow for non-determinism.

When restricting ourselves to finite sets of atomic actions and atomic formulas, some of the semantics presented here for explicit effect axioms are a convenient starting point for operationalization. In this case, the state space of the specified systems is finite, and we are able to construct it explicitly. This opens the door to the application of model checking techniques (like in [4]) on this state space to verify system properties.

6 Conclusions and future work

In this paper we defined several alternative semantics for action specification formulas under the frame assumption and the qualification assumption. Choosing one of them removes the ambiguity introduced by implicit frame assumptions and implicit qualification assumptions. Procedures such as adding frame axioms or applying completion that are usually necessary to reveal the intended meaning of a specification, can be checked against the semantics we defined. We plan to define similar semantics for the first order case. Furthermore, we plan to check existing procedures for scenario generation and reachability analysis in the first order case against our semantics.

References

[1] A. Borgida, J. Mylopoulos, and R. Reiter. On the frame problem in procedure specifications. *IEEE Transactions on Software Engineering*, 21:785–798, 1995.

[2] S. Brass, U. W. Lipeck, and P. Resende. Specification of object behaviour with defaults. In G. Koschorreck U. W. Lipeck, editor, *Proceedings of the International Workshop on Information Systems – Correctness and Reusability (IS-CORE'93)*, pages 155–177. Informatik-Berichte 01/93, Universit t Hannover, Januari 1993.

[3] Frank M. Brown, editor. *The frame problem in artificial intelligence: proceedings of the 1987 workshop, April 12-15, 1987, Lawrence, Kansas.* Kaufmann, 1987.

[4] E.M. Clarke, E.A. Emerson, and A.P. Sistla. Automatic verification of finite-state concurrent systems using temporal logic specifications. *ACM Transactions on Programming Languages and Systems*, 8(2), April 1986.

[5] R.B. Feenstra and R.J. Wieringa. LCM 3.0: a language for describing conceptual models. Technical Report IR-344, Faculty of Mathematics and Computer Science, Vrije Universiteit, Amsterdam, December 1993.

[6] Matthew L. Ginsberg and David E. Smith. Reasoning about action 2: the qualification problem. *Artificial Intelligence*, 35:311–342, 1988.

[7] D. Harel. *First Order Dynamic Logic.* Springer, 1979. Lecture Notes in Computer Science 68.

[8] D. Kozen and J. Tiuryn. Logics of programs. In Jan van Leeuwen, editor, *Handbook of Theoretical Computer Science*, pages 789–840. Elsevier Science Publishers, 1990.

[9] F. Lin and R. Reiter. State Constraints Revisited. *Journal of Logic and Computation*, 4(5):655–678, 1994. Special Issue on Action and Processes.

[10] U. W. Lipeck and S. Brass. Object-oriented system specification using defaults. In H. Marburger K. von Luck, editor, *Management and Processing of Complex Data Structures - Third Workshop on Information Systems and Artificial Intelligence 1994*, Lecture Notes in Computer Science 777, pages 22–43. Springer-Verlag, 1994.

[11] R. Reiter. On specifying databse updates. *Journal of Logic Programming*, 25(1):53–91, 1995.

[12] P.A. Spruit, R.J. Wieringa, and J.-J.Ch. Meyer. Axiomatization, declarative semantics and operational semantics of passive and active updates in logic databases. *Journal of Logic and Computation*, 5(1):27–50, 1995.

[13] M. Winslett. *Updating Logical Databases.* Cambridge University Press, 1990.

[14] M. Winslett. Cicrumscriptive semantics for updating knowledge bases. *Annals of Mathematics and Artificial Intelligence*, 3:429–450, 1991.