Minimal Semantics for Transaction Specifications in a Multi-modal Logic

Jan Broersen* Remco Feenstra*† Roel Wieringa*

July 18, 1996

Abstract

This paper presents an extension of propositional dynamic database logic in which the specification writer can specify arbitrary database transactions declaratively. Since declaratively specified transactions are under-specified, the specification must be supplemented with a frame assumption about what does *not* change as the result of a transaction. In addition, static database constraints imply preconditions, also called qualifications , that must be satisfied for a transaction to occur and static constraints may imply derived updates for a transaction (also called ramifications). We solve these problems by defining preferences over Kripke structures and showing how these can be used to define an intended semantics that, for each declarative transaction specification, selects a unique meaning for each transaction. In particular, the intended semantics formalizes a frame assumption and selects qualifications and ramifications for each specification. The notion of preferential entailment based on the semantics provides a natural way of reasoning about declaratively specified transactions.

1 Introduction

The role of Integrity Constraints (ICs) in databases can be described as to characterize the set of possible observable database system states. Transactions define the set of possible observable transitions between these states. We define a database transaction here as an atomic interaction between the database system and its environment, where an interaction is defined to be atomic if it has no (observable) intermediary states. Recent research by Paul Spruit [7, 6] has shown that dynamic logic can be used to define elementary transactions of logic databases that, roughly, have the form $\&X(IP(T)$ **where** $\phi)$, to be read as "for all X for which ϕ is true, insert T into P". Similar updates have been defined for deletion of tuples from a predicate and for updates of predicates. In this paper, we generalize this approach by taking arbitrary updates into account. This means that we allow the writer of a database schema to specify arbitrary database transactions by means of dynamic logic.

We propose a declarative style of transaction specification by means of pre– and postconditions. It is well-known that declarative specifications are incomplete in the sense that they must be supplemented with *frame assumptions* about what does *not* change as the result of a transaction. In addition, the ICs imply certain preconditions called *qualifications* that must be satisfied for a transaction to occur and they may imply derived updates for a transaction, called *ramifications*. The solution for these problems requires fixation of an intended semantics of the specification, with respect to which the transactions are interpreted. In this intended semantics, decisions have been made concerning frame assumptions and transaction qualifications and ramifications. In this paper, we define minimal semantics for declarative specifications of

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

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

logic database transactions. In future work, we will extend this with operational transaction semantics that corresponds to the intended minimal specification semantics.

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 their interconnections. Section 4 then proposes two minimal semantics for PDL specifications and shows that these have non-monotonic properties. Section 5 compares our approach with other approaches and discusses an application of the theory developed so far. 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. In this paper, we use PDL as a formalism for writing specifications of database transactions.

We indicate transactions with the meta-variables α, β, ... and arbitrary actions with the meta-variables a, b, ... Arbitrary actions are compounds of the form $a;b$, $a \cup b$ or $a*$, representing sequential composition, choice and iteration, respectively. The modal formulas $\langle\alpha\rangle\phi$, where α is a transaction, mean that there is a possible occurrence of α after which ϕ is true. The standard PDL-semantics of transactions as relations over (database) states is given in the next Section. We also define how compound actions are interpreted over Kripke structures.

2.1 The PDL language

A **signature** consists of two non-empty finite sets: $\Sigma = (\mathcal{P}, \mathcal{A})$. $\mathcal{P}$ is a set of atomic formulas, referred to by the meta-variables A, B, ... $\mathcal{A}$ is a set of atomic actions (transactions) with associated meta-variables α, β, ...

The set of well-formed formulas $L(\Sigma)$ over a signature Σ is the smallest set that satisfies the following rules:

1. All elements of $\mathcal{P}$ are well-formed formulas.
2. For all well-formed formulas p and q, the formulas $(p \vee q)$ and $\neg p$ are well-formed formulas.
3. For all α in $\mathcal{A}$ and each well-formed formula p, $\langle\alpha\rangle p$ is a well-formed formula.
4. For all well-formed formulas $\langle a\rangle p$ and $\langle b\rangle p$, the formulas $\langle a;b\rangle p$, $\langle a \cup b\rangle p$ and $\langle a*\rangle p$ are well-formed formulas.

As usual, we abbreviate $\neg(\neg p \vee \neg q)$ to $p \wedge q$, $\neg p \vee q$ to $p \rightarrow q$, $(p \rightarrow q) \wedge (q \rightarrow p)$ to $p \equiv q$ and $\neg\langle a\rangle\neg p$ to $[a]p$.

A **specification** $Spec = (\Sigma, \Phi)$ is a pair consisting of a finite set Φ of formulas over signature Σ. In the rest of the paper we assume a fixed specification $Spec$.

2.2 The semantics of PDL

PDL formulas are interpreted over a special kind of Kripke structures. A **structure** S is defined to be a tuple (W, Acc) where

- $W \subseteq 2^{\mathcal{P}}$, and $W \neq \emptyset$
- Acc is a total function $W \times W \rightarrow 2^{\mathcal{A}}$

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 the restricted notion of a structure suffices.

Acc says for each pair of worlds how atomic programs α are interpreted over them. Elements of $2^{\mathcal{P}}$ and thus of W are referred to as w, w', ... PDL formulas are interpreted over these structures. First we define truth of a formula in a world w of a structure $S = (W, Acc)$. ($\langle a \rangle^n$ is an abbreviation for $\langle a \rangle_1 \langle a \rangle_2 \ldots \langle a \rangle_n$)

$$
\begin{array}{lll}
S, w \models P & \text{iff} & P \in w \\
S, w \models p \vee q & \text{iff} & S, w \models p \text{ or } S, w \models q \\
S, w \models \neg p & \text{iff} & S, w \not\models p \\
S, w \models \langle \alpha \rangle p & \text{iff} & \exists w' \in W \text{ for which } \alpha \in Acc(w, w') \text{ and } S, w' \models p \\
S, w \models \langle a; b \rangle p & \text{iff} & S, w \models \langle a \rangle \langle b \rangle p \\
S, w \models \langle a \cup b \rangle p & \text{iff} & S, w \models \langle a \rangle p \text{ or } S, w \models \langle b \rangle p \\
S, w \models \langle a* \rangle p & \text{iff} & \exists n \in N.\ S, w \models \langle a \rangle^n p
\end{array}
$$

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 formulas

We use PDL as a language to specify atomic actions and their effects. For that purpose we turn our attention to PDL-formulas of the following forms:

- **Postcondition formulas** $\phi_i \rightarrow [\alpha]\psi_i$ We say that ϕ_i is a sufficient precondition of α with respect to the postcondition ψ_i.
- **Guard formulas** $\langle \alpha \rangle true \rightarrow \chi_j$ We call χ_j a guard of α, equivalently, a necessary precondition for the possible occurrence of α.
- **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 are elements of $L(\Sigma)$ that contain no modal constructs. So they are just propositional logic formulas. We assume that these are the only PDL-formulas we are allowed to use when stating a specification. This follows the syntactic restrictions of LCM [3]. These restrictions are motivated by the fact that in the specification of database transactions, we only need precondition and guard formulas and in the specification of static integrity constraints, we only need static constraint formulas.

So we allow a specification writer to use only a very restricted part of the PDL-language. The reason we also presented the semantics of general PDL-formulas (including ones with compound actions) is that properties whose validity we want to be able to check may be stated in formulas of any form.

Definition 1 *A transaction axiom is either a postcondition formula, a guard formula or a static constraint. We use Φ_{trans} to indicate an arbitrary set of transaction axioms.*

The task we pose ourselves is to define exactly which models formalize what we want to say with our formulas. 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 PDL-interpretation of postcondition formulas, as formally defined in Section 2.2, is that when ϕ_i is true, transaction α leads to a situation where ψ_i is true. We use these formulas to describe the effect of an transaction. However, defining the effect of an transaction by means of a postcondition always causes **the frame problem.** 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 a transaction. So we want to make the frame assumption that everything that has not been specified to change has not changed. However, this is not part of the PDL-semantics, as defined in 2.2. In Section 4.1, we will formalize an alternative meaning for PDL formulas and thus for postcondition formulas that incorporates this frame assumption.

The PDL-interpretation of guard formulas is that transaction α can only take place when χ_j is true. Thus χ_j is a necessary precondition for the occurrence of α. Guard formulas only state necessary preconditions, we cannot (and do not want to) express sufficient preconditions with the transaction axioms we defined. This relates to the **qualification problem**. This problem states that it is not possible to foresee all necessary preconditions for the success of a transaction. This is equivalent to saying that it is impossible to check exhaustively for every state whether a transaction is not allowed to occur in it. So the guard axioms of α can be seen to divide possible worlds into two sets: a set of worlds in which α is definitely not allowed to occur and a set of worlds in which α may occur. The reluctance about the occurrence of transactions in this second set of worlds stems from the fact that we cannot be sure to have foreseen all circumstances under which transactions are not allowed to occur. It may be necessary to make this set of worlds smaller by adding new guard axioms to prevent actions from taking place in situations that we did not think of earlier-on. In particular, we cannot foresee whether the occurrence of a transaction violates constraints or gives rise to contradictory effects caused by contradictory postcondition axioms. This shows why we deliberately omitted the possibility to specify sufficient precondition axioms; in that case a specification can become inconsistent because of conflicts with constraints or contradictory postcondition axioms. But we do want to take into account that the conjunction of guard axioms is taken to be sufficient. This means that we want to make what we call 'the qualification assumption'. The PDL-semantics does not provide this. It is easily seen that there are PDL-interpretations of transaction axioms in which transactions do not occur even if the guards are true. In particular the structure where the accessibility relation between possible worlds is completely empty always PDL-satisfies a set of transaction axioms. In Section 4.2 we solve this problem by formalizing the qualification assumption.

A third class of problems is related to the static integrity constraints. One problem, which we already touched upon in the foregoing, is that in general, a specification writer cannot foresee whether a transaction will violate a constraint or not. Under the PDL-semantics, a specification writer will probably express sufficient preconditions by means of completion. So under the PDL-semantics it is the responsibility of the specification writer to ensure that the conjunction of the guards of an transaction always guarantees the preservation of the static constraints. Since the specification writer cannot possibly do this in general, we must adopt the qualification assumption to incorporate the presence of static ICs.

If we interpret constraints not only as restrictions on possible worlds but as rules that can give rise to derivations, we run into **the ramification problem**. This problem is caused by the inability of the specification writer to foresee what all the derived effects will be, so he cannot specify the derived effects explicitly. This leaves the responsibility to state what the derived effects are to the semantics of the derivation rules. But then the same problems concerning the

frame assumption and the qualification assumption also apply to derivations; derivations should only change conditions that are explicitly derived to change and derivations that end up in the violation of a constraint should be excluded.

In Section 4.3 we will formalize the two possible interpretations of constraints in a way that extends the frame assumption and qualification assumption to apply to ramifications too.

4 Minimal change/maximal reachability semantics

In this Section we formalize the frame assumption in the notion of minimal change between possible worlds and the qualification assumption in the notion of maximum reachability.

4.1 Minimal change

Definition 2 *Given a signature $\Sigma = (\mathcal{P}, \mathcal{A})$ and a structure $S = (W, Acc)$ 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:*

$$Diff(w, w') \equiv$$
$$\{P \in \mathcal{P} \mid S, w \models P \text{ and } S, w' \not\models P\} \cup \{P \in \mathcal{P} \mid S, w \not\models p \text{ and } S, w' \models P\}$$

Using this definition we can define alternative semantics for PDL that try to capture the notion of minimal change between possible worlds. The minimal semantics defined below follow Winslett's [8] classification of update semantics. To implement the notion of minimal change, we define partial orderings over structures.

Definition 3 *Given two structures $S = (W, Acc)$ and $S' = (W', Acc')$*

$$S' \sqsubseteq_{mc} S \text{ iff}$$
$$\text{for all } (w, w') \in W \times W, \text{ for all } \alpha \in Acc(w, w')$$
$$\text{there is a } (w, w'') \in W' \times W' \text{ with } \alpha \in Acc'(w, w'')$$
$$\text{and}$$
$$\text{for all } (w, w'') \in W' \times W', \text{ for all } \alpha \in Acc'(w, w'')$$
$$\text{there is a } (w, w') \in W \times W \text{ with } \alpha \in Acc(w, w')$$
$$\text{such that } N(Diff(w, w'')) \leq N(Diff(w, w'))$$

Proposition 1 *$\sqsubseteq_{mc}$ is a partial ordering over structures.*

Proof $\sqsubseteq_{mr}$ can easily seen to be transitive, reflexive and anti-symmetric. ■

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 4 *Given two structures $S = (W, Acc)$ and $S' = (W', Acc')$*

$$S' \sqsubseteq_{mc} S \text{ iff}$$
$$\text{for all } (w, w') \in W \times W, \text{ for all } \alpha \in Acc(w, w')$$
$$\text{there is a } (w, w'') \in W' \times W' \text{ with } \alpha \in Acc'(w, w'')$$
$$\text{and}$$
$$\text{for all } (w, w'') \in W' \times W', \text{ for all } \alpha \in Acc'(w, w'')$$
$$\text{there is a } (w, w') \in W \times W \text{ with } \alpha \in Acc(w, w')$$
$$\text{such that } Diff(w, w'') \subseteq Diff(w, w')$$

Proposition 2 *$\sqsubseteq_{ms}$ is a partial ordering over structures.*

Proof $\sqsubseteq_{mr}$ can easily seen to be transitive, reflexive and anti-symmetric.

■

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).

Proposition 3 *Let Φ be a set of PDL-formulas that has a model with a non-empty access function. Then there is a MS(MC)-model of Φ with non-empty access function*

Proof Let S be a model of Φ with a non-empty access function. It follows from the finiteness of the number of propositions and transactions that there are only finite many models. Then either S is already minimal, or there is a model S' that is minimal and $S' \sqsubseteq_{ms} S$ ($S' \sqsubseteq_{mc} S$). In the latter case S' can not be a model with empty access function because it follows from the definitions of the orderings that in that case S' can not be compared with S.

■

The minimal subset semantics is weaker than the minimal cardinality semantics, as is expressed by the following proposition.

Proposition 4 *an MC-model of Φ is also an MS-model of Φ.*

Proof Follows directly form 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 minimal subset semantics is the more appropriate and intuitive one for our purposes. In 4.4 we will use this semantics in an example. Furthermore, both semantics coincide for an important class of postcondition formulas: the determinate postcondition formulas.

Definition 5 *A **determinate postcondition formula** has the form $\phi \rightarrow [\alpha]L_1 \wedge L_2 \wedge \ldots \wedge L_n$, with L_i a literal, that is, a positive or negated atomic formula.*

We take Φ_{det} to stand for an arbitrary set of determinate postcondition axioms and guard axioms.

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

Proof

We concentrate on one transaction α. The validity in the context of more transactions follows trivially from this. We prove that if S is an MS-model of Φ_{det}, it also is an MC-model of Φ_{det}. The reverse already has been proven.

Assume that $S = (W, Acc)$ is a model of Φ_{det} that is $\sqsubseteq_{ms}$-minimal but not $\sqsubseteq_{mc}$-minimal. Then Acc cannot be empty because a model with an empty accessibility relation always is an MC-model and an MS-model at the same time. For the same reason accessibility relations in S cannot all be reflexive. This means there is a $S' = (W', Acc')$ such that $S' \sqsubseteq_{mc} S$ and not $S \sqsubseteq_{mc} S'$. Then from the definition of $\sqsubseteq_{mc}$-minimality we have that $\exists (w, w') \in W \times W, \exists \alpha \in Acc(w, w')$ such that $\forall (w, w'') \in W' \times W'$ with $\alpha \in Acc'(w, w''), N(\mathit{Diff}(w, w'')) < N(\mathit{Diff}(w, w'))$

Now from the $\sqsubseteq_{ms}$-minimality of S it follows that $\mathit{Diff}(w, w') \subseteq \{atoms(L_{i1}, L_{i2}, \ldots, L_{in})\}$, whith $\{L_{i1}, L_{i2}, \ldots, L_{in}\}$ the set of literals that appear in the determinate postcondition formulas $\phi_i \rightarrow [\alpha]L_{i1} \wedge L_{i2} \wedge \ldots \wedge L_{in}$ in Φ_{det} for which $S, w \models \phi_i$. It is not difficult to see that if

$Diff(w, w') \subseteq \{atoms(L_{i1}, L_{i2}, \ldots, L_{in})\}$ would not be true, we could easily construct a structure that is more MS-minimal then S.

Because both S and S' are models, in both w' and w'' the atoms in $\{L_{i1}, L_{i2}, \ldots, L_{in}\}$ have the same valuation. This means that w' differs from w in the same atoms, and thus in the same number of atoms among $\{L_{i1}, L_{i2}, \ldots, L_{in}\}$ as w'' does. Then from $N(Diff(w, w'')) < N(Diff(w, w'))$ it follows that there is an atom not among the atoms in $\{L_{i1}, L_{i2}, \ldots, L_{in}\}$ whose valuation in w' differs from its valuation in w, while in w'' the valuation is equal to the one in w. But this is in contradiction with $Diff(w, w') \subseteq \{atoms(L_{i1}, L_{i2}, \ldots, L_{in})\}$.

■

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

4.2 Maximum reachability

The guarding axioms of α can be seen to divide possible worlds into two sets: a set of worlds in which α is definitely not allowed to occur and a set of worlds in which α may occur. The reluctance about the occurrence of transactions in this second set of worlds stems from the fact that we cannot be sure to have foreseen all circumstances under which transactions are not allowed to occur. In particular, we cannot foresee whether the occurrence of a transaction violates constraints or gives rise to contradictory effects caused by contradictory postcondition axioms. We want that our formal semantics takes all of this into account. So we want interpretations that tend to interpret the guard axioms specified for a transaction α as sufficient, provided that the occurrence of α is compatible with the constraints and postconditions in the postcondition axioms. We accomplish this by formalizing the qualification assumption in the notion of maximum reachability.

Definition 6 *Given two structures $S = (W, Acc)$ and $S' = (W', Acc')$*

$$
\begin{aligned}
&S \sqsubseteq_{mr} S' \text{ iff} \\
&\quad W \subseteq W' \text{ and} \\
&\quad \text{for all } (w, w') \in W \times W,\ Acc(w, w') \subseteq Acc'(w, w')
\end{aligned}
$$

Proposition 6 *$\sqsubseteq_{mr}$ is a partial ordering over structures.*

Proof $\sqsubseteq_{mr}$ can easily seen to be transitive, reflexive and anti-symmetric.

■

Proposition 7 *Let Φ be a set of formulas for which there is an MS(MC)-model. Then there is a $\sqsubseteq_{mr}$-maximal, MS(MC)-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 ordering over them.

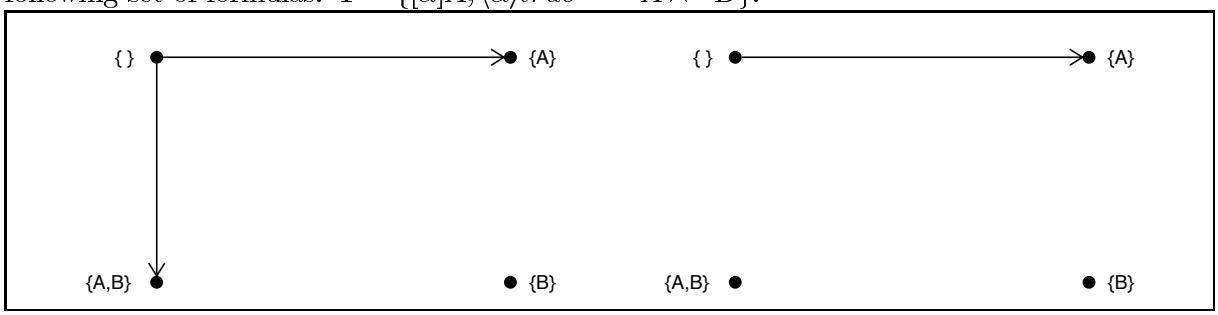
■

We can now identify the structures that interpret a specification both under the frame assumption and the qualification assumption.

Definition 7 *Given a specification (Σ, Φ), a Min-Max-model is defined as a $\sqsubseteq_{mr}$-maximal element of the set of MS(MC)-models of Φ.*

The above definition states that a Min-Max-model is a $\sqsubseteq_{mr}$-maximal element among the $\sqsubseteq_{ms}$-minimal ($\sqsubseteq_{mc}$-minimal) models of Φ. It is tempting to think we could also have defined this in reverse order, thus: a Min-Max-model is a $\sqsubseteq_{ms}$-minimal ($\sqsubseteq_{mc}$-minimal) element among the $\sqsubseteq_{mr}$-maximal structures that satisfy Φ. However, this does not provide the semantics we are looking for. The next example shows this.

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



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 4.4 that this construction also works in case there are ramifications.

The MS(MC)-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 $\neg(\langle\alpha\rangle A \wedge \langle\alpha\rangle B)$ and the two MS-satisfying structures S with Acc is $\{(\{\}, \{A\}) \rightarrow \alpha\}$ and S' with Acc' is $\{(\{\}, \{B\}) \rightarrow \alpha\}$. There is no MS-satisfying structure that is an upper bound for both structures S and S'. The structure S'' with Acc'' is $\{(\{\}, \{A\}) \rightarrow \alpha, (\{\}, \{B\}) \rightarrow \alpha\}$ 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 of the restricted form we defined, we can prove that the MS-models do form a complete lattice. Therefore, the following proposition holds.

Proposition 8 *Each specification Φ_{trans} has a unique Min-Max-model if it has a model.*

Proof We must prove that the MS-satisfying structures of a specification Φ_{trans} form a complete lattice. 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 $S = (W, Acc)$ and $S' = (W, Acc')$ as $S'' = (W, Acc \cup Acc')$ and the *glb* as $S'' = (W, Acc \cap Acc')$. It is not difficult to see that both *lub* and *glb* are MS-satisfying.

■

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

4.3 Static constraint interpretation

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

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

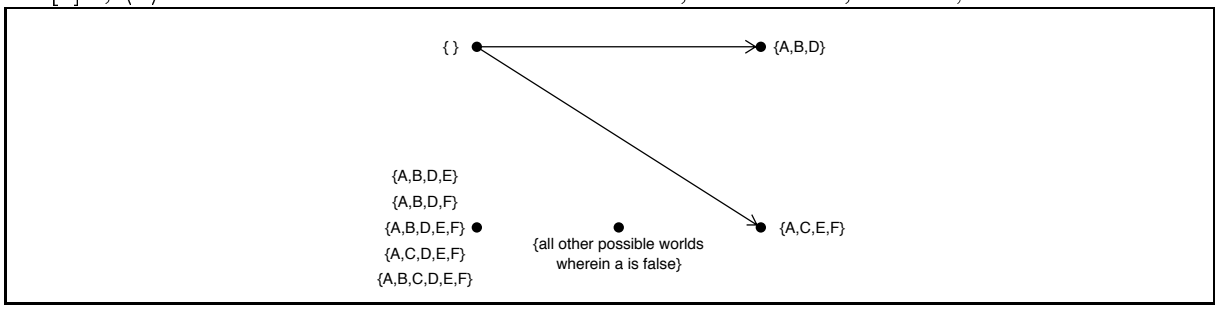
The constraint semantics does not give rise to derived updates (worlds), it merely 'cuts out' access to worlds in which the constraints are not satisfied. The trick here is that we first look to minimal models without taking the constraints into account. In these models there possibly is access to worlds that do not comply with the constraints. By selecting the models that do comply with the constraints, we implicitly 'throw away' all access to worlds that do not comply with the constraints. Under the ramification semantics these worlds would not be there from the beginning, causing the minimal access to be directed to alternative worlds (the 'derived worlds'). Finally we select the models with as many worlds and access between them as possible and thus we apply the $\sqsubseteq_{mr}$-maximality criterion.

In 4.4 we show how the Ramification semantics extends the frame assumption and the qualification assumption to ramifications. It is possible to combine constraint and ramification semantics by splitting the set 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[7] works this out for Propositional Dynamic Database Logic (PDDL).

4.4 Example

As example specification we take a signature with $\mathcal{P} = \{A, B, C, D, E, F\}$, and $\mathcal{A} = \{\alpha\}$, and the following set of formulas:

$[a]A$, $\langle\alpha\rangle true \rightarrow \neg A \wedge \neg B \wedge \neg C \wedge \neg D \wedge \neg E \wedge \neg F$, $A \rightarrow B \vee C$, $B \rightarrow D$, $C \rightarrow E \wedge F$.



We use the minimal subset / ramification semantics to interpret the example formulas. The picture shows the unique Min-Max-model of the example specification. If we had chosen the minimal cardinality semantics, the world $\{A, C, E, F\}$ would not be reachable. This shows that the minimal cardinality semantics leads to asymmetry in the interpretation of disjunctions, which does not seem desirable. We see that the frame assumption also applies to ramifications, because worlds like $\{A, B, C, D, E, F\}$ are not reachable.

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 [2]. The restricted context of specifying database updates allows us to avoid some complications because we usually deal with transactions with deterministic effects, and constraints can be restricted to simpler forms.

Reiter[5] discusses dealing with the frame problem from a database point of view by using precondition axioms and successor state axioms to specify transactions in the situation calculus. Lin and Reiter[4] 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 [8] 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.

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 [**?**]) on this state space to verify system properties.

6 Conclusions and future work

In this paper we defined several alternative semantics for transaction specification formulas under the frame assumption and the sufficient precondition assumption. Choosing one of them removes the ambiguity introduced by implicit frame assumptions and implicit sufficient precondition 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. "$\cdots$ and nothing else changes": The frame problem in procedure specifications. published as Borgida95a.

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

[3] 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.

[4] F. Lin and R. Reiter. State Constraints Revisited. Technical report, Department of Computer Science, University of Toronto, 1994. To appear in the Journal of Logic and Computation, Special Issue on Action and Processes.

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

[6] P.A. Spruit. *Logics of Database Updates*. PhD thesis, Faculty of Mathematics and Computer Science, Vrije Universiteit, Amsterdam, 1994.

[7] 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.

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