1980

A Note on the Semantics of Looping Programs in Propositional Dynamic Logic

Francine Berman

Report Number:
80-346

A Note on the Semantics of Looping Programs in Propositional Dynamic Logic

By

Francine Berman

Purdue University

CSD-TR 346

A Note on the Semantics of Looping Programs in Propositional Dynamic Logic

by

Francine Berman

Purdue University

The research reported here was supported in part by NSF Grant MCS77-02474. Most of the results in this paper are from the dissertation of F. Berman written at the University of Washington under the direction of R. W. Ritchie. The model in Figure 4 was included in the paper "A Completeness Technique for D-axiomatizable Semantics" presented at the 11th Annual ACM Symposium on the Theory of Computing in May, 1979.

July 14, 1980

Abstract

We discuss the representation of looping programs in Propositional Dynamic Logic (PDL). We show that PDL is not expressive enough to distinguish between models in which loops are interpreted as the set of all finite sequences of iterations and models in which loops are interpreted as a set of computations which preserve loop invariants. We note that for distinguishable models of finite domain, both interpretations of loops coincide.

Introduction

Propositional Dynamic Logic (PDL) is a formal language for reasoning about programs. As with flowchart schemes, programs in PDL are represented by regular expressions with tests. Such programs, when combined with a simple assertion language can be used to describe such properties as termination, correctness, loop invariance and failure conditions.

In this paper, we describe two sets of models which interpret looping programs in PDL: the class of Standard models and the class of Loop Invariant models. The class of Standard models interprets the looping program a* as the set of all finite iterations of program a. The class of Loop Invariant models interprets a* as a set of computations which preserve the invariant assertions of program a. In Section 1, we define these classes of models and review pre-

vious results. Although both classes satisfy different semantic constraints, we note that PDL is too weak to distinguish between the two differing interpretations of iteration. In Section 2 we note that the class of Standard models is contained within the class of Loop Invariant models and show that this containment is proper. In addition, we note that distinguishable Loop Invariant models of finite domain are also Standard models.

Section 1: Semantics

Propositional Dynamic Logic was introduced by M. Fischer and R. Ladner in [F&L]. Programs are represented in the language by regular expressions with tests and formulae are represented by boolean combinations of propositional expressions and formulae with modalities. For an introduction to the syntax of PDL, see [Bel], [F&L] or [Ha].

definition: A model of PDL is a triple (W, Π, p) where

- W is a set of states,
- Π is a formula valuation function which assigns to each basic formula a set of states (at which that formula is true),

ρ is a program valuation function which assigns to each program b a set of pairs (w,v) of states (where w corresponds to the initial state and v corresponds to a final state in a computation of program b).

We extend Π to evaluate all formulae as follows:

$$\Pi(p \vee q) = \Pi(p) \cup \Pi(q)$$

$$\Pi(\neg p) = W - \Pi(p)$$

$$\Pi(\langle a \rangle p) = \{w \mid \exists v((w,v) \in \rho(a) \wedge v \in \Pi(p))\}.$$

Note that a model preserves the intended interpretation of PDL formulae through restrictions on the interpretation of the boolean connectives ¬ (not) and ∨ (or) and on the modal operator <>. However, a model may arbitrarily assign an interpretation to PDL programs. In particular, component subprograms may be interpreted with no relation to the interpretation of their larger program.

definition: Let $M = (W, \Pi, \rho)$ be a model. We say that states w and w' in W are indistinguishable in M if for all PDL formulae p w is in $\Pi(p)$ iff w' is in $\Pi(p)$. When the model referred to is clear, we say that w and w' are indistinguishable. A model in which no two states are indistinguishable is called distinguishable.

ρ is a program valuation function which assigns to each program b a set of pairs (w,v) of states (where w corresponds to the initial state and v corresponds to a final state in a computation of program b).

We extend Π to evaluate all formulae as follows:

$$\Pi(p \vee q) = \Pi(p) \cup \Pi(q)$$
$$\Pi(\neg p) = W - \Pi(p)$$
$$\Pi(\langle a \rangle p) = \{w \mid \exists v (\ (w,v) \in \rho(a) \wedge v \in \Pi(p)\ )\}.$$

Note that a model preserves the intended interpretation of PDL formulae through restrictions on the interpretation of the boolean connectives $\neg$ (not) and $\vee$ (or) and on the modal operator $\langle\rangle$. However, a model may arbitrarily assign an interpretation to PDL programs. In particular, component subprograms may be interpreted with no relation to the interpretation of their larger program.

definition: Let $M = (W, \Pi, \rho)$ be a model. We say that states w and w' in W are indistinguishable in M if for all PDL formulae p w is in $\Pi(p)$ iff w' is in $\Pi(p)$. When the model referred to is clear, we say that w and w' are indistinguishable. A model in which no two states are indistinguishable is called distinguishable.

We use the notation $M,w \models p$ to denote the statement "w is in $\Pi(p)$ for $M = (W, \Pi, p)$". If $M,w \models p$ for all w in W, we say $M \models p$.

We now define two classes of models which differ in their interpretation of the iteration operator *.

defintion: A Standard model is a model in which the program valuation function is constrained as follows:

$$p(aUb) = p(a) \cup p(b)$$

$$p(a;b) = p(a) \circ p(b)$$

$$p(p?) = \{(w,w) \mid w \in \Pi(p)\}$$

$$p(a^*) = \bigcup_{n \geq 0} p(a^n) \quad \text{(where } a^0 \text{ is } (pv\text{-}p)?\text{)}.$$

Standard models were first introduced by M. Fischer and R. Ladner in [F&L]. Note that the program a* is interpreted as the reflexive and transitive closure of the interpretation of program a, i.e. the set of finite sequences of iterations of a. Note also that the interpretation of a program c in a Standard model corresponds to the language defined by c as a regular expression where tests p? are interpreted as single symbols.

definition: A Loop Invariant (LI) model is a model in which the program valuation function is constrained as follows:

$p(aUb) = p(a) \cup p(b)$

$p(a;b) = p(a) \circ p(b)$

$p(p?) = \{(w,w) \mid w \in \Pi(p)\}$

$p(a^0) \subseteq p(a^*)$

$p(a) \subseteq p(a^*)$

$p(a^*) \circ p(a^*) \subseteq p(a^*)$

$\Pi(\langle a^* \rangle p \rightarrow (p \vee \langle a^* \rangle (\neg p \wedge \langle a \rangle p))) = W.$

Loop Invariant models were introduced in [Par] (as Nonstandard models) and developed in [Be2]. Note that the interpretation of the program a* may properly contain the set of interpretations of all finite sequences of iterations of program a.

What do the other state-pairs in this set represent? The requirement that the induction schema I $(\langle a^* \rangle p \ (p \vee \langle a^* \rangle (\neg p \ \langle a \rangle p)))$ is valid forces all state-pairs to preserve the invariant assertions of program a. The extra state-pairs correspond to infinite halting computations. In addition, the induction schema constrains these computations to have tails composed of finite sequences of iterations.

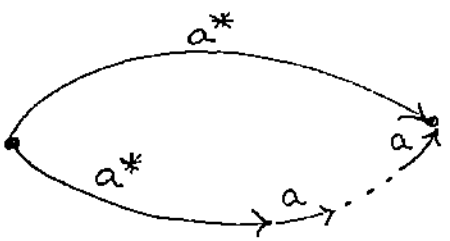

Figure 1

The propositional simplicity which permitted a

straightforward decision procedure for satisfiability in PDL ([F&L]) renders the language too weak to express the difference between the infinite halting computation of Figure 1 and the set of finite halting iterations. Consequently, the theories of the class of Loop Invariant models and the class of Standard models are precisely the same set of formulae ([Par], [Be2]). Not surprisingly, in distinguishable models of finite domain, the Loop Invariant interpretation given in Figure 1 reduces to the Standard interpretation in Figure 2.

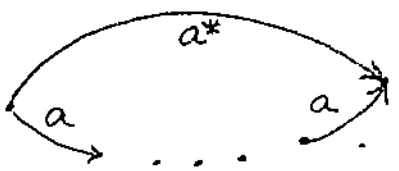

Figure 2

Section 2: Not all Loop Invariant models are Standard

Since any finite sequence of iterations of a program b preserves the invariants of b, one would expect the class of Standard models to be contained in the class of Loop Invariant models. This containment is in fact proper, i.e. there are Loop Invariant models which are not Standard. This is trivial to see if we consider models with indistinguishable states. In particular, the model described by Figure 3 is a Loop Invariant model but not a Standard model if w and w' are indistinguishable.

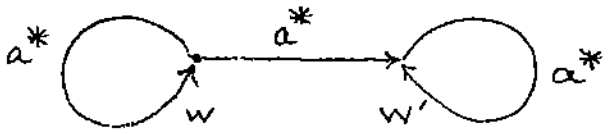

Figure 3

If we consider distinguishable models, it becomes non-trivial to find an LI model which is not Standard. In Theorem 2, we show that for distinguishable models of finite domain, every Loop Invariant model is a Standard model, hence we are forced to seek our counterexample among LI models of infinite domain. Such a counterexample is given in Figure 4. The description of the model follows.

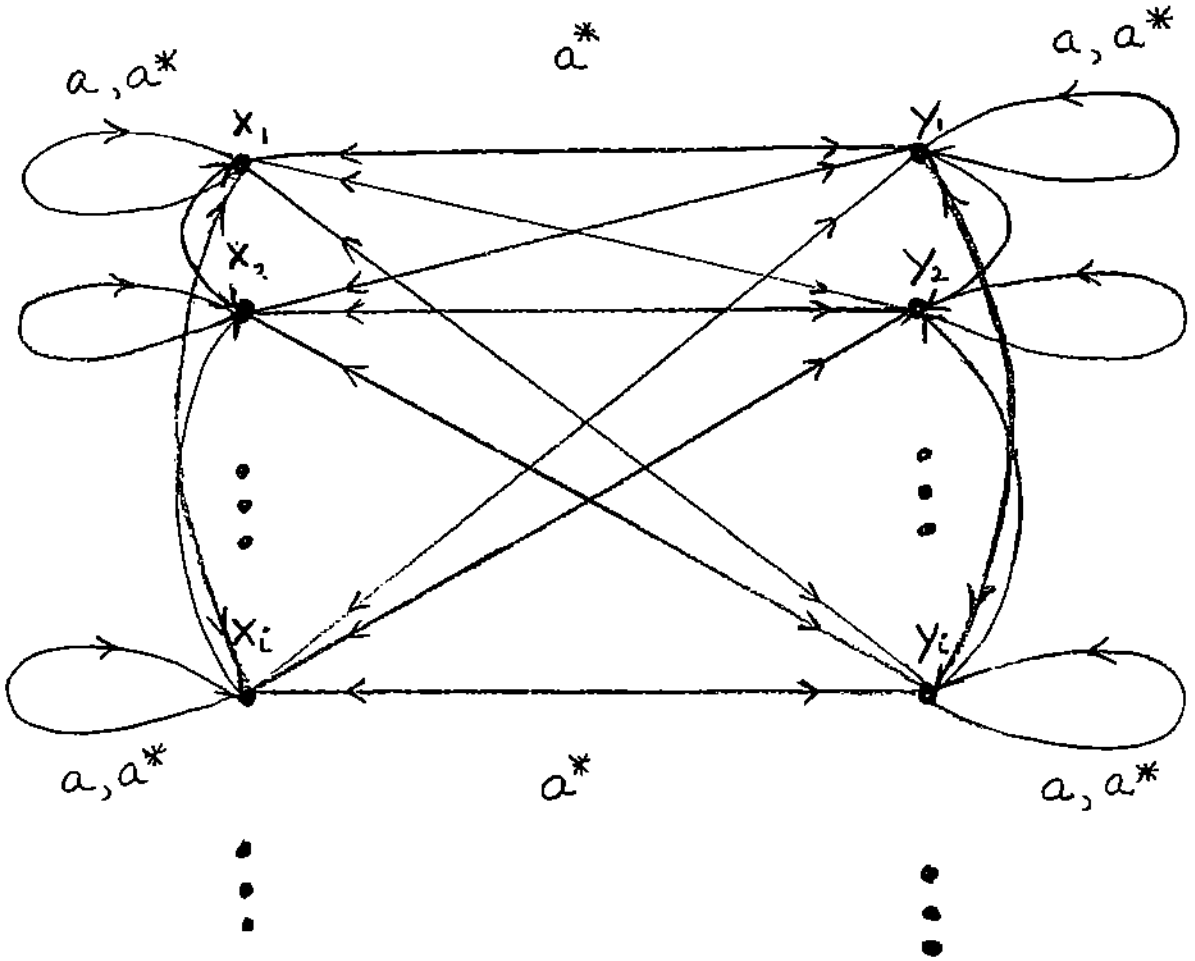

Figure 4

We define the model M given by Figure 4. Let $\{Q_n\}_{n>0}$ be the set of finite subsets of positive integers according to some fixed enumeration. Let $\{p_i\}_{i>0}$ be the set of basic assertions of PDL and let a be a basic program. Let M = (W, Π, p) where

$W = \{x_i\}_{i>0} \cup \{y_i\}_{i>0}$

x_i is in $\Pi(p_j)$ iff j is in Q_i

y_i is in $\Pi(p_j)$ iff j is not in Q_i

and

$p(a) = \{(x_i, x_j) \mid \text{for all } i,j\} \cup \{(y_i, y_j) \mid \text{for all } i,j\}$

$p(c) = \emptyset$ for c a basic program $\neq$ a

Extend Π, p according to the definition given in Section 1 and let

$p(c^*) = \bigcup_{n\geq 0} p(c^n)$ for $c \neq a$

$p(a^*) = W \times W$.

Note that the model is basically two Standard models (the x side and the y side of the figure) strung together by a* edges. To show that the induction schema holds (and consequently that the model is Loop Invariant), we will show in the proof of Theorem 1 that no PDL formula can distinguish the x side of the figure from the y side of the figure. Note that the x side and the y side can be distinguished by

modal formulae in $L_{\omega_1,\omega'}$ i.e. each y_i but no x_j satisfies $\langle a\rangle(p_1 \wedge p_2 \wedge \ldots \wedge p_n \wedge \ldots)$.

Theorem 1

M is a distinguishable Loop Invariant model which is not Standard.

Proof

By construction, M is not a Standard model. To see this, we need only notice that for all i and j, (x_i, y_j) is in $p(a^*) - \bigcup_{n\geq 0} p(a^n)$. It is also clear that M is distinguishable since each state differs from each other state on at least one basic assertion. To show that M is a Loop Invariant model, we must show that for all programs c and d and for all formulae p,

$$p(c \cup d) = p(c) \cup p(d)$$

$$p(c;d) = p(c)\ p(d)$$

$$p(p?) = \{(w,w) \mid w \text{ in } \Pi(p)\}$$

$$p(c^0) \subseteq p(c^*)$$

$$p(c) \subseteq p(c^*)$$

$$p(c^*) \circ p(c^*) \subseteq p(c^*)$$

$$\Pi(\langle c^*\rangle p \rightarrow (p \vee \langle c^*\rangle(\neg p \wedge \langle c\rangle p))) = \Pi(I) = W.$$

It is straightforward from the definition to show that all semantic constraints except the last hold. Hence to show that M is Loop Invariant, it is enough to show that the

induction schema I is valid in M. Note that the only non-trivial instance of I is when c is the basic program a. (Otherwise, by definition $p(c^*) = \bigcup_{n\geq 0} p(c^n)$). We show that for any formula p, $M \models \langle a^* \rangle p \rightarrow (p \vee \langle a^* \rangle (\neg p \wedge \langle a \rangle p))$ with the aid of the following construction.

Let p be a formula. Let Φ_p be the set of all basic assertions occuring in p. Define a relation $\sim_p$ on W as follows: For w, v in W, let $w \sim_p v$ iff for all formulae q in Φ_p, $M,w \models q$ iff $M,v \models q$. It is straightforward to show that $\sim_p$ is an equivalence relation on W of finite index. Note that each equivalence class contains members from both $\{x_i\}_{i>0}$ and $\{y_i\}_{i>0}$. Denote the equivalence class of a state w by w_p. Note that $\{(w_p,v_p) \mid (w,v) \text{ is in } p(a)\} = \{(w_p,v_p) \mid (w,v) \text{ is in } p(a^*)\} = W_p \times W_p$ for all formulae p.

Note the following properties of the model M:

Let w be a state in w_p and v be a state in v_p. Let c be a test-free program. Then for each w' in w_p, there is a state v' in v_p such that

1) (w,v) is in p(c) iff (w',v') is in p(c).

Let w and w' be states in w_p. Let q be a formula all of whose basic assertions occur in Φ_p. Then

2) $M,w \models q$ iff $M,w' \models q$.

Assume for a moment that we have shown properties 1) and 2). Let p be a formula in PDL. Assume that $M,x_i \models \langle a^* \rangle p$. Then there is a state v with $M,v \models p$. If $M,x_i \models p$ then $M,x_i \models I$. Otherwise, $M,x_i \models \neg p$. By definition, (x_i,x_i) is in $p(a^*)$. By property 2), $M,v \models p$ iff $M,v' \models p$ for all v' in v_p. Let v' be a state in $v_p \cap \{x_k\}_{k>0}$. Then (x_i,v') is in $p(a)$. Hence $M,x_i \models I$ for all i. By symmetry, $M,y_i \models I$ for all i.

To show property 1), note that precisely the same set of states is accessible via c from each of the x_i and from each of the y_i. Hence if w and w' are both on the x side or both on the y side, 1) holds with v = v'. Assume without loss of generality that w is x_i and w' is y_j. Let V = $\{v \mid (x_i,v)$ is in $p(c)\}$ and V' = $\{v' \mid (y_j,v')$ is in $p(c)\}$. Then depending on c, V is {w} iff V' is {w'}, $\{x_k \mid k>0\}$ iff V' is $\{y_k \mid k>0\}$ and W iff V' is W. Let v' be a state in $V' \cap v_p$. This set is nonempty and there is a path labelled with c from w' to v'. Hence

(w,v) in p(c) iff (w',v') is in p(c).

By symmetry, 1) holds for $w = y_i$.

To show property 2), we proceed by induction on q:

If q is a basic assertion then clearly, $M,w \models q$ iff $M,w' \models q$ for all w, w' in w_p.

In addition, if $q = r \vee s$ or $q = -r$ then by induction, property 2) holds.

Let $q = \langle c \rangle r$. Then $M,w \models \langle c \rangle r$ iff there is a state v with (w,v) in $p(c)$ and $M,v \models r$. By induction, for all v' in v_p, $M,v' \models r$. Let c be test-free. Then by property 1), there is a state v' in v_p with (w',v') in $p(c)$. Hence $M,w' \models \langle c \rangle r$.

Assume that c contains one or more tests. Then the path from w to v is a path in $p(t_1?;c_1;\ldots t_n?;c_n;t_{n+1}?) \subseteq p(c)$ where each c_i is test-free and either $t_1?$ or $t_{n+1}?$ may be equivalent to $(pv\text{-}p)?$. Then there exist states $w_1,\ldots,w_{n-1}$ with (w,w) in $p(t_1?)$, (w,w_1) in $p(c_1)$, (w_1,w_1) in $p(t_2?),\ldots$,(w_{n-1},w_{n-1}) in $p(t_n?)$, (w_{n-1},v) in $p(c_n)$ and (v,v) in $p(t_{n+1}?)$. Note that for all j, $|t_j| < |q|$ hence by induction, $M,w_i \models t_j$ iff $M,w_i' \models t_j$ for all w_i' in $(w_i)_p$. Also by induction and property 1) (recall that the c_i are test-free), there are states $w_1',\ldots,w_n',v'$ with w_1' in $(w_1)_p,\ldots$,w_n' in $(w_n)_p$, v' in v_p such that (w',w') is in $p(t_1?)$, (w', w_1') is in $p(c_1)$, (w_1',w_1') is in $p(t_2?),\ldots$, (w_{n-1}',w_{n-1}') is in $p(t_n?)$, (w_{n-1}',v') is in $p(c_n)$ and (v',v') is in $p(t_{n+1}?)$. Hence $M,w' \models \langle c \rangle r$. The other direction is analogous.

We have shown that $M \models I$. Since M satisfies the other semantic constraints, M is a Loop Invariant model. ⊠

As it turns out, no finite distinguishable model will serve as a counterexample for Theorem 1. Distinguishable Loop Invariant models of finite domain are in fact Standard models. This is straightforward to show by a slight modification of Lemma 5 in [Par]. The proof is essentially an application of the pigeon-hole principle.

Theorem 2 (After Parikh)

Let M be a distinguishable Loop Invariant model of finite domain. Then M is a Standard model.

Proof

We present a sketch of the proof. For further details see Lemma 5 in [Par] or Proposition 6 in [Bel].

Let $M = (W, \Pi, p)$ be a distinguishable Loop Invariant model of finite domain and let (w,v) be a state-pair in $p(a^*) - \bigcup_{n \geq 0} p(a^n)$. Then by repeated applications of the induction schema, for any positive integer n there is a path between w and v with a tail composed of a sequence of n iterations of program a (see Figure 1). Since M has only a finite number of distinguishable states, for each state x, there is a formula which distinguishes x from any other state in M. We can use such distinguishing formulae and the induction schema to show the existence of a loop-free path between w and v with a tail of $|W|$ iterations of program a.

This provides a contradiction since some state must be repeated.

Acknowledgements

We would like to thank Mike Fischer and Bob Ritchie for their support. We would also like to thank Mike O'Donnell for his constructive comments on and careful reading of this paper.

Bibliography

[Be1] Berman, F., "Syntactic and Semantic Structure in Propositional Dynamic Logic," Ph.D. Dissertation, University of Washington, Seattle, Washington, 1979.

[Be2] Berman, F., "Models for Verifiers," T.R. 343, Purdue University, July, 1980.

[F&L] Fischer, M.J. and R.E. Ladner, "Propositional Modal Logic of Programs," 9th Symposium on the Theory of Computing, 1977.

[Ha] Harel, D., First-Order Dynamic Logic, Lecture Notes in Computer Science No. 68, Springer-Verlag, New York/Berlin, 1979.

[Par] Parikh, R., "A Completeness Result for Propositional Dynamic Logic," Symposium on the Mathematical Foundations of Computer Science, Zakopane, Poland, 1978.