PROPOSITIONAL DYNAMIC LOGIC OF CONTEXT-FREE PROGRAMS

David Harel,[*+] Amir Pnueli* and Jonathan Stavi[+]

Abstract: The borderline between decidable and undecidable Propositional Dynamic Logic (PDL) is sought when iterative programs represented by regular expressions are augmented with increasingly more complex recursive programs represented by context-free languages. The results in this paper and its companion [HPS] indicate that this line is extremely close to the original regular PDL.

The main result of the present paper is: The validity problem for PDL with additional programs $\alpha^{\Delta}(\beta)\gamma^{\Delta}$ for regular α, β and γ, defined as $\bigcup_i \alpha^i ; \beta ; \gamma^i$, is Π^1_1-complete. One of the results of [HPS] shows that the single program $A^{\Delta}(B)A^{\Delta}$ for atomic A and B is actually sufficient for obtaining Π^1_1-completeness. However, the proofs of this paper use different techniques which seem to be worthwhile in their own right.

*Department of Applied Mathematics, The Weizmann Institute of Science, 76 100 Rehovot, Israel

+Department of Mathematics and Computer Science, Bar-Ilan University, Ramat-Gan, Israel

1. Introduction

Propositional Dynamic Logic, henceforth PDL, is a formal logic for reasoning on a propositional level about programs. PDL was defined by Fischer and Ladner [FL], based upon work of Pratt [Pr1], as a direct extension of the propositional calculus, in which assertions concerning the in/out (i.e., before/after) behavior of programs can be made.

Given an alphabet Σ of atomic programs and tests, the class of programs allowed in formulas of PDL is taken to be the set RG of regular expressions over Σ. The justification of this choice is rooted in the well-known correspondence between iterative programs over Σ, as modelled, say, by flowcharts, and regular sets of strings over Σ. See, e.g. [dBM]. The set of strings defined by a program $\alpha \in RG$ is thought of as the set of possible computation sequences constituting α. In the sequel this fixed version of PDL is denoted by PDL_{RG}.

In [FL] it was shown that the validity problem for PDL_{RG} is decidable. In fact, it is decidable in deterministic exponential time [Pr2], and to within a polynomial this upper bound is the best possible [FL].

Consider the set CF of context-free grammars over Σ. There is an analogous correspondence (see [dBM]) between recursive programs over Σ and context-free sets of strings over Σ, justifying the study of PDL_{CF}.

Unfortunately, the equivalence and inclusion problems for context-free grammars, which are undecidable, can easily be reduced to the validity problem for PDL_{CF}, rendering the latter undecidable too. This was pointed out in 1977 by R. Ladner.

One question arising here concerns the degree of undecidability of PDL_{CF}. Since the equivalence problem for CF is co-r.e., the aforementioned observation cannot be used to show that PDL_{CF} is any harder than Π_1^0. However, of even greater interest is the problem of locating the precise point between RG and CF at which PDL becomes undecidable. This question gains some momentum upon observing that there are interesting classes of context-free grammars for which inclusion and equivalence are known to be decidable, and others for which some of these, and similar problems, are open. See, e.g., [H,L,Y,GF]. In many of these cases, the restrictions which admit a context-free grammer into the class in question correspond to reasonable syntactic restrictions on the correspoinding recursive program.

In this paper and its companion [HPS] it is shown that the borderline between decidable and undecidable PDL is extremely close to RG, and, furthermore, that the transition is most striking: from decidable in exponential time for PDL_{RG} to Π_1^1 - completeness for most of our extensions.

Specifically, the general class K of programs which we consider contains RG and all programs of the form $\alpha^{\Delta}(\beta)\gamma^{\Delta}$ for $\alpha,\beta,\gamma \in$ RG. The new program is defined to contain all computations of $\alpha^i;\beta;\gamma^i$, for all $i \geq 0$.

In Section 2 we define PDL_K and show that the inclusion and equivalence problems for K are decidable so that PDL_K cannot be shown <u>undecidable</u> by Ladner's observation. We also show that PDL_K lacks the finite model property, so that it cannot be shown <u>decidable</u> by the finite model method of [FL].

In Section 3 we use a reduction of the Post correspondence problem to show the undecidability of PDL_K. This result, although subsumed by the main result of the paper, is presented by virtue of its relative simplicity.

In Section 4 we prove that PDL_K is Π_1^1-complete by reducing to it the truth of formulas of the form $\forall f \exists x P$, where P is a diophantine relation. That these formulas are universal Π_1^1 (see [R]) follows from Matijasevic's Theorem [M]. We also show how to improve this proof method obtaining a stronger version of the result. Our strongest version of this result, namely, that PDL with the additional single program $A^{\Delta}(B)A^{\Delta}$ is Π_1^1-complete, is proved in [HPS] using a different technique consisting of encoding certain Turing machine computations.

In [HPS] several results concerning other nonregular programs, notably programs over one-letter alphabets, are also presented.

These results constitute a full answer to the first question posed, and a partial answer to the second. First, since PDL_{CF} is easily seen to be in Π_1^1, our results establish its Π_1^1-completeness. Second, the results show that some extremely conservative additions to RG result in a highly undecidable PDL, to be contrasted with exponential time decidability in their absence. A comprehensive characterization of the classes of programs for which PDL is decidable remains an intriguing topic for future research. In particular, it is open at the time of writing as to whether there is any nonregular program whose addition to RG does not destroy the decidability of PDL. (See note at end of paper.)

2. Definitions and Preliminary Observations

Let Π be a set of atomic programs, with $\Theta \in \Pi$, (the empty program), and let Φ be a set of atomic propositions.

Let $\Sigma = \Pi \cup \{P? | P \in \Phi\} \cup \{\sim P? | P \in \Phi\}$. Let PROG be a given set of expressions, called programs, each associated with some subset of Σ^*. For $\alpha \in PROG$ this subset is denoted $L_{PROG}(\alpha)$, or just $L(\alpha)$ when the context is clear. Throughout we assume $L(\Theta) = \emptyset$.

The formulas of the propositional dynamic logic of PROG, denoted PDL_{PROG}, are defined as follows:

1) $\Phi \subseteq PDL_{PROG}$,

2) if $p,q, \in PDL_{PROG}$ then $\sim p, p \vee q \in PDL_{PROG}$,

3) if $p \in PDL_{PROG}$ and $\alpha \in PROG$ then $\langle\alpha\rangle p \in PDL_{PROG}$.

We use true, false, $\wedge$, $\supset$ and $\equiv$ as abbreviations in the standard way. In addition, we abbreviate $\sim\langle\alpha\rangle\sim p$ to $[\alpha]p$.

A structure (or model) is a triple $S = (W^S, \pi^S, \rho^S)$, where W^S is a nonempty set, the elements of which are called states, π^S is a satisfiability relation on Φ, i.e., $\pi^S: \Phi \to 2^W$, and $\rho^S: \Pi \to 2^{W \times W}$ provides a binary relation on W as the meaning of each atomic program in Π. Most often we will omit the superscript of the components of S.

We extend ρ to words over Σ as follows:

1) $\rho(\lambda) = \{(u,u) | u \in W\}$, ($\lambda$ is the empty string),

2) $\rho(P?) = \{(u,u) | u \in \pi(P)\}$ $p \in \Phi$,

3) $\rho(\sim P?) = (W \times W) - \rho(P?)$,

4) $\rho(x;y) = \rho(x) \circ \rho(y)$. $x,y \in \Sigma^*$, (o is the composition operator on binary relations)

Given a structure S, the satisfiability relation is defined for all formulas of PDL_{PROG} as follows:

1) $u \models P$ iff $u \in \pi(P)$, for $p \in \Phi$,

2) $u \models \sim p$ iff not $u \models p$,

3) $u \models p \vee q$ iff either $u \models p$ or $u \models q$

4) $u \models \langle\alpha\rangle p$ iff $\exists x \in L(\alpha).\ \exists v \in W.\ (u,v) \in \rho(x)$ and $v \models P$.

Although we allow only atomic tests and their negations in PDL_{PROG}, since our results are all negative, they hold also for the more general case of tests p? for any formula $p \in PDL_{PROG}$.

Let RG be the set of regular expressions over Σ. The reader can easily check that PDL_{RG} coincides with PDL, as defined, say, in [FL], with the above restriction on tests.

In particular, since $L(\alpha^*) = (L(\alpha))^* = \bigcup_i L(\alpha^i)$, with $\alpha^0 = \lambda$ and $\alpha^{i+1} = \alpha;\alpha^i$, we have $u \models \langle\alpha\rangle^* p$ iff $\exists i,\ u \models \langle\alpha^i\rangle p$ -

A formula $p \in PDL_{PROG}$ is valid, denoted $\models p$, if for every structure S and for every $u \in W^S$, $u \models p$; it is satisfiable if $\sim p$ is not valid. Hence p is satisfiable if there is a structure S and state $u \in W^S$ such that $u \models p$. The latter is sometimes written $S,u \models p$.

The inclusion (respectively, equivalence) problem for PROG is the problem of deciding, given $\alpha,\beta \in PROG$, whether or not $L(\alpha) \subseteq L(\beta)$ (resp. $L(\alpha) = L(\beta)$). The validity problem for PDL_{PROG} is the problem of deciding, given $p \in PDL_{PROG}$, whether or not $\models p$.

Fischer and Ladner [FL] have shown that every satisfiable formula p of PDL_{RG} is satisfied in a structure in which the number of states is finite and exponential in the size of p. This fact, termed the small model property, is used in [FL] to show that the validity problem for PDL_{RG} is decidable.

Let CF_o (respectively, CF) be the set of context, free grammars over terminals Π (respectively Σ) and some fixed set of nonterminals. It is well known that the equivalence (and hence also the inclusion) problem for CF_o is undecidable [BPS]. This fact can be used to show that the validity problem for PDL_{CF_o}, and hence also for PDL_{CF}, is undecidable.

<u>Proposition 2.1</u> (due to R. Ladner): For any $\alpha,\beta \in CF_o$, $P\in\Phi$, $\models(\langle\alpha\rangle P \supset \langle\beta\rangle P)$ iff $L(\alpha) \subseteq L(\beta)$.

<u>Proof</u>: (if) Immediate from the definition of $\langle\alpha\rangle P$.

(only if) Let $x\in L(\alpha)$, where $x = A_1,\ldots,A_k$, and the A_i are (not necessarily distinct) elements of Π. Define the structure $S_x = (\{u_o,\ldots,u_k\},\pi,\rho)$ such that $\pi(P) = \{u_k\}$, and such that for any $A \in \Pi$,

$$(u_i,u_j) \in \rho(A) \text{ iff } j = i+1 \text{ and } A = A_i.$$

S_x is illustrated in Fig.1. Clearly $S_x, u_o \models\langle\alpha\rangle P$ and hence by assumption also $S_x, u_o \models\langle\beta\rangle P$. But this implies that $x \in L(\beta)$. □

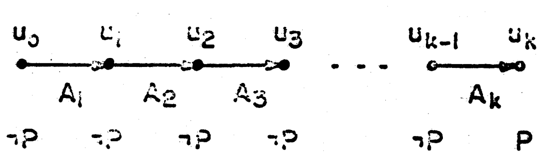

Figure 1

<u>Corollary 2.2</u>: The validity problems for PDL_{CF_o} and PDL_{CF} are undecidable.

We now define our set of programs K. It will become clear that $RG < K < CF$, where $PROG1 < PROG2$ whenever $\{L_{PROG1}(\alpha) \mid \alpha\in PROG1\} \subsetneq \{L_{PROG2}(\alpha) \mid \alpha\in PROG2\}$.

$$K = RG \cup \{(\alpha^{\Delta}(\beta)\gamma^{\Delta}) \mid \alpha,\beta,\gamma \in RG\}.$$

When there is no ambiguity we will drop the additional parentheses.

Sets of strings over Σ^* are associated with programs in K as follows:

1) $L_K(x) = \{x\}$, for $x\in\Sigma - \{\theta\}$, $L_K(\theta) = \emptyset$.

2) $L_K(\alpha\cup\beta) = L_K(\alpha) \cup L_K(\beta)$,

3) $L_K(\alpha;\beta) = L_K(\alpha)\cdot L_K(\beta) = \{xy \mid x\in L_K(\alpha) , y\in L_K(\beta)\}$,

4) $L_K(\alpha^*) = (L_K(\alpha))^* = \bigcup_{i\geq 0} L_K(\alpha^i)$,

5) $L_K(\alpha^{\Delta}(\beta)\gamma^{\Delta}) = \bigcup_{i\geq 0} L_K(\alpha^i;\beta;\gamma^i)$.

We shall abbreviate $(\alpha^{\Delta}(\theta^*)\gamma^{\Delta})$ to $(\alpha^{\Delta}\gamma^{\Delta})$

<u>Proposition 2.3</u>: The inclusion and equivalence problems for K are decidable.

<u>Idea of proof</u>: Each $\alpha\in K$ can be written as a grammar in CF which is <u>simple-deterministic</u> <u>stack uniform</u> [L]. The result then follows from [L]. We omit the details. □

It follows that PDL_K cannot be shown to be undecidable by Proposition 2.1. We prove now that it cannot be shown decidable by the Fischer-Ladner method, since it lacks the small model property. Let <u>force</u> be the following formula of PDL_K :

$$(P \wedge [A^*]\langle A;B^*\rangle P) \quad \wedge \quad [(A\cup B)^* ; B; A]\,\underline{\text{false}}$$
$$\wedge\ [A^*; A; A^{\Delta}B^{\Delta}]\sim P \quad \wedge \quad [A^{\Delta}B^{\Delta}; B]\,\underline{\text{false}} .$$

<u>Proposition 2.4</u>: <u>Force</u> is satisfiable but has no finite model.

<u>Proof</u>: Let S_o be the structure illustrated in Fig. 2, in which the only states satisfying P are those marked ●. It is easy to see that $S, u \models$ <u>force</u> .

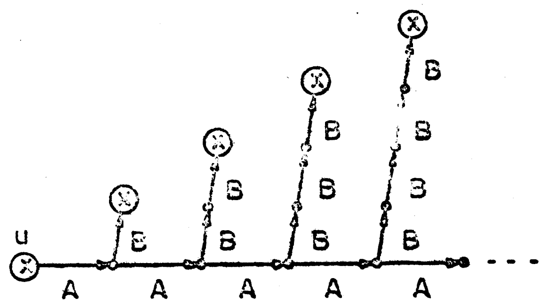

Figure 2

Assume now that $S, u \models$ force where $|W^S| < \infty$, $u \in W^S$. S can be thought of as a finite directed graph with atomic programs labeling edges and sets of atomic propositions labeling nodes. An (A,B)-path is one in which each edge is labelled A or B. Associating paths in S with the sequences of labels along their edges, Let $U \subseteq \{A,B\}^*$ be the set of words labelling (A,B)-paths connecting u with states satisfying P. Since S is finite, this is exactly the definition of a set of words recognized by a finite transition graph, hence u is regular. On the other hand, the second conjunct of force eliminates from U paths which contain B followed by A, forcing U to be contained in A^*B^*. Moreover, the third and fourth conjuncts force U to be a subset of $\{A^iB^i | i \geq 0\}$. Finally, the first conjunct of force states that for each $i \geq 0$, A^iB^i is in U.

Hence $U = \{A^iB^i | i \geq 0\}$, and so cannot be regular, contradicting the assumed finiteness of S. □

3. PDL_K is Undecidable

In this section we reduce the solvability of Post Correspondence Problems (PCP's) to the satisfiability of formulas of PDL_K. Since the former is undecidable, in fact r.e., so is the latter, rendering the dual validity problem Π_1^0-hard.

Specifically, let $H = \{(x_1,y_1),\ldots,(x_n,y_n)\}$ be a PCP, where $x_i, y_i \in \{a,b\}^*$, for $1 \leq i \leq n$. A solution to H is a sequence $(i_1,\ldots,i_k)$, where $1 \leq i_j \leq n$ for $1 \leq j \leq k$, such that, denoting the reverse of a word $x \in \{a,b\}^*$ by x^R, we have $x_{i_1},\ldots,x_{i_k} = y_{i_1}^R,\ldots,y_{i_k}^R$. Note that if $w = x_{i_1},\ldots,x_{i_k}$ then $w^R = y_{i_k},\ldots,y_{i_1}$. It is easy to relate the classical formulation of PCP to our slightly modified version. We shall construct a formula $\text{reduce}_H \in PDL_K$ such that reduce_H is satisfiable iff H has a solution.

Let H be given. The formula reduce_H involves the two atomic programs A and B and atomic propositions $P,Q,R_1,\ldots,R_n$. The letters a and b will be encoded as the programs A;~Q? and A;Q?, respectively, or similarly with B replacing A, so that words over $\{a,b\}^*$ can be identified with sequences of truth values of Q along paths of A's or B's. $R_1,\ldots,R_n$ will be used to encode indices between 1 and n. (Actually, log n atomic propositions suffice here.)

The idea is to force models of reduce_H to contain a block of A's followed by a block of B's of equal length, encoding, respectively, w and w^R for some word $w \in \{a,b\}^*$, and such that w consists of a sequence of words from among the x's, w^R of a sequence of the same length of words from among the y's, and such that indices of words in both blocks correspond.

For each $1 \leq i \leq n$ define $R^{(i)}$ to be the program $\sim R_1?; \sim R_2?;\ldots;\sim R_n?$ with $\sim R_i?$ replaced by $R_i?$. For any $z \in \{a,b\}^*$ define the program $C^A(z)$ inductively as follows:

$$C^A(a) = A;Q? \quad , \qquad C^A(b) = A;\sim Q?$$

$$C^A(z_1z_2) = C^A(z_1)C^A(z_2) \ .$$

$C^B(z)$ is defined in the same way with B replacing A throughout.

Define $L_x = \bigcup_{1 \le i \le n} (R^{(i)}; C^A(x_i))$

$L_y = \bigcup_{1 \le i \le n} (C^B(y_i); R^{(i)})$

Now, let $\underline{reduce_H}$ be the conjunction of the following formulas:

$\underline{\text{exist-path}}$: $\sim P \wedge \langle L_x{}^{\Delta} L_y{}^{\Delta} \rangle P$

$\underline{\text{indices-correspond}}$: $[L_x{}^*; R^{(i)}?; L_x{}^{\Delta} L_y{}^{\Delta}] R^{(i)}$,

$\underline{\text{same-length}}$: $[A^{\Delta} B^{\Delta}] P \wedge [A^*; A; A^{\Delta} B^{\Delta}] \sim P$

$\wedge\ [(A \cup B)^*; P?; (A \cup B)]\underline{false}$,

$\underline{\text{same-word}}$: $[A^*; A; Q?; A^{\Delta} B^{\Delta}; B] Q \ \wedge$

$[A^*; A; \sim Q?; A^{\Delta} B^{\Delta}; B] \sim Q$.

<u>Lemma 3.1</u>: For any $H = \{(x_1,y_1),\ldots,(x_n,y_n)\}$, H has a solution iff $\underline{reduce_H}$ is satisfiable.

<u>Proof</u>: (if) Assume $S, u \models \underline{reduce_H}$. By <u>exist-path</u> there is a nonempty path p in S, starting at u, which encodes in order the words $x_{i_1},\ldots,x_{i_k}$ for some $k > 0$ and some $i_1,\ldots,i_k$, using A, followed by $y_{j_k},\ldots,y_{j_1}$ for some $j_1,\ldots,j_k$, encoded using B. Furthermore, by <u>same-length</u> we know (respectively, in the order of its conjuncts) that any path of the form $A^{\Delta}B^{\Delta}$ ends with P holding, that P holds at the end of no path A^iB^j with $j < i$, and that P holds at most once along any $\{A,B\}$ path. Consequently, p consists precisely of two blocks of A's and B's of equal lengths. In other words $|x_{i_1},\ldots,x_{i_k}| = |y_{j_k},\ldots,y_{j_1}|$. By <u>indices-correspond</u> considered along path p, we have $i_\ell = j_\ell$. Finally, by <u>same-word</u> considered along p we conclude that $x_{i_1},\ldots,x_{i_k} = (y_{i_k},\ldots,y_{i_1})^R = y_{i_1}^R,\ldots,y_{i_k}^R$.

(only if) Let $(i_1,\ldots,i_k)$ be a solution to H. Construct the structure S of Fig.3, where the words x_{i_ℓ} and y_{i_ℓ} are encoded using Q as described above. The reader can easily verify that $S, u \models \underline{reduce_H}$. ◻

<u>Corollary 3.1</u>: The validity problem for PDL_K is undecidable.

4. $\underline{PDL_K \text{ is } \Pi_1^1\text{-complete}}$.

In this section we reduce to PDL_K the truth of formulas $F(m)$ of the form $\forall f(f(0)=1 \supset \exists xP)$ where $P(m,f(x),f(x+1))$ is a diophantine relation involving m and the two values of f, $f(x)$ and $f(x+1)$. It can be shown using Matijasevic's Theorem [M,DMR], that associated with each Π_1^1-complete set X of natural numbers there is such a formula F_X, with $m \in X$ iff $F_X(m)$ is true. Moreover, the equation P can be transformed into a conjunction φ of equalities of the form $t_i=0$, $t_i=1$, $t_i+t_j=t_k$ and $t_i \cdot t_j=t_k$, where the t's are from among $m, f(x), f(x+1)$ and new variables $y_1,\ldots,y_\ell$ which are existentialy quantified, i.e. $P \equiv \exists \bar{y}\varphi$. Here ℓ depends on the equation P.

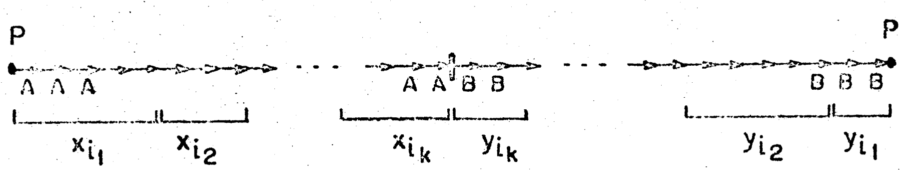

Figure 3.

In the sequel $\varphi(x_o,\ldots,x_{\ell+2})$ will denote a conjunction of such equalities over $x_o,\ldots,x_{\ell+2}$. Consequently, in order to show that the validity problem for PDL_K is Π^1_1-hard, or equivalently that the satisfiability problem is Σ^1_1-hard, it suffices to find, for each such φ a formula $\underline{reduce}^m_\varphi$ of PDL_K, effectively depending on m, which is satisfiable iff $\exists f(f(o) = 1 \wedge \forall x \exists y_1,\ldots,\exists y_\ell \varphi(m,y_1,\ldots,y_\ell,f(x),f(x+1))$ is true.

First we show how to simulate the conjunction $\varphi(x_o,\ldots,x_{\ell+2})$ by a PDL_K formula on particularly well behaved structures.

Let $\bar{n} = (n_o,\ldots,n_{\ell+2})$ be an arbitrary tuple of natural numbers. A nice structure for $\bar{n}$ is any structure $S = (W, \pi, \rho)$ such that $\{u_o,\ldots,u_p\} \subseteq W$, $\{(u_i,u_{i+1}) | 0 \leq i < p\} \subseteq \rho(A)$, $u_i \in \pi(P_j)$ iff $i = n_j$, and $u_i \in \pi(S_j)$ iff $i = a \cdot n_j$ for some $a \geq 0$. Moreover, $p \geq \max_i(n_i^2)$. In other words, the "A-part" of S (termed the A cut of S from U_o in [MSM]) contains an initial segment of the natural numbers large enough to contain all squares of the n_i. P_j encodes n_j by being true precisely at distance n_j from the start, u_o, and S_j encodes similarly all multiples of n_j which fall within the segment. Given φ, define the formula $\underline{simulate}_\varphi$ inductively on the structure of φ as follows:

$\underline{simulate}_{\varphi\wedge\varphi'} = \underline{simulate}_\varphi \wedge \underline{simulate}_{\varphi'}$,

$\underline{simulate}_{x_i=o} = P_i$,

$\underline{simulate}_{x_i=1} = [A]\, P_i$,

$\underline{simulate}_{x_i+x_j=x_k} = [A^\Delta(P_i?;\ A^*;\ P_j?)\ A^\Delta]\ P_k \wedge [A^\Delta(P_j?;\ A^*;\ P_i?)\ A^\Delta] P_k$,

$\underline{simulate}_{x_i \cdot x_j=x_k} = ((P_i \vee P_j) \supset P_k)$

$\wedge [A;A^\Delta(P_i?;A^*;P_j?)((A;\sim S_j?)^*\ ;\ A;S_j?)^\Delta]\ P_k$

$\wedge [A;A^\Delta(P_j?;A^*;P_i?)((A;\sim S_i?)^*;\ A;S_i?)^\Delta]\ P_k$.

Lemma 4.1: For any $\bar{n} = (n_o,\ldots,n_{\ell+2})$, $S, u_o \models \underline{simulate}_\varphi$ for some nice structure S for $\bar{n}$, iff $\varphi(\bar{n})$ is true.

Proof: (only if) Let S be nice for $\bar{n}$, and let $S, u_o \models \underline{simulate}_\varphi$. We show that $\varphi(\bar{n})$ is true by induction on the structure of φ. The cases $\varphi \wedge \varphi'$ and $x_i = 0$ are trivial. For the case $x_i = 1$, we have $S, u_o \models [A]P_i$, which implies $S, u_1 \models P_i$, or $u_1 \in \pi(P_i)$, which in turn, implies $n_i = 1$.

For the case where φ is of the form $x_i + x_j = x_k$, the formula $\underline{simulate}_{x_i+x_j=x_k}$ can be seen to state that when $n_i \leq n_j$ (i.e., P_i becomes true before P_j when traversing the u branch of the structure S starting from u_o) we have in fact $n_i+(n_j-n_i)+n_i=n_k$, and that when $n_j \leq n_i$, $n_j + (n_i-n_j) + n_j = n_k$. In either case $n_i + n_j = n_k$. Fig. 4 illustrates this case.

For the case where φ is of the form $x_i \cdot x_j=x_k$, the formula $\underline{simulate}_{x_i \cdot x_j=x_k}$ states that if one of

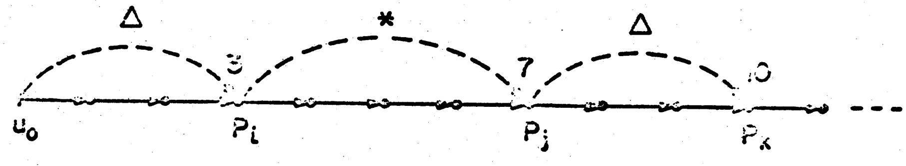

Figure 4.

n_i or n_j is 0 then so is n_k, and if $0 < n_i \leq n_j$ then $1 + (n_i-1) + (n_j-n_i) + (n_i-1)\cdot n_j = n_k$, and if $0 < n_j \leq n_i$ then $1 + (n_j-1) + (n_i-n_j) + (n_j-1)\cdot n_i = n_k$. In either case $n_i \cdot n_j = n_k$. Fig. 5 illustrates this case. The structure has to be long enough to encode all multiples of the n_i so that the clauses for + and $\cdot$ should not be vacuously true.

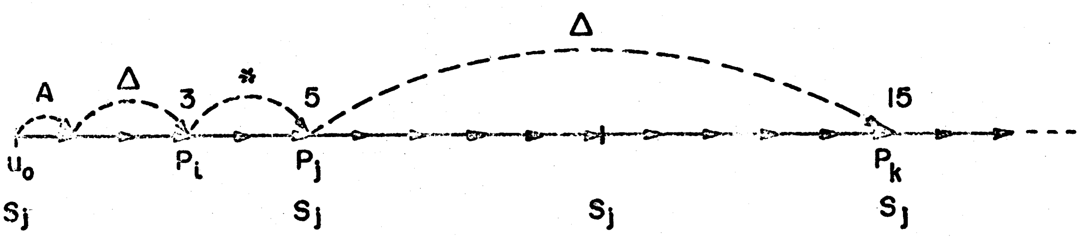

Figure 5.

(if) If $\varphi(\bar{n})$ is true, construct the nice structure $S_{\bar{n}}$ for $\bar{n}$ simply by replacing both $\subseteq$ by $=$ in the definition of nice structures. There is now only one linear A-path in the structure. By induction on the structure of φ one shows that $S_{\bar{n}}, u_o \models \underline{simulate}_\varphi$. We argue the case $x_i + x_j = x_k$ and leave the rest to the reader. If $n_i + n_j = n_k$ and $n_i < n_j$ then the first conjunct of $\underline{simulate}_{x_i+x_j=x_k}$ is true in u_o since it states that $n_i + (n_j-n_i) + n_i = n_k$. The second conjunct is vacuously true by virtue of the structure containing no path upon which P_j becomes true no earlier than P_i. Similarly, if $n_j < n_i$ then the first conjunct is vacuously true and the second follows from $n_i+n_j=n_k$. Finally, if $n_i = n_j$, both conjuncts state that $n_i + n_i = n_j + n_j = n_k$. □

We now turn to the construction of $\underline{reduce}_\varphi^m$. The idea is to force models of $\underline{reduce}_\varphi^m$ to contain an infinite (possibly cyclic) sequence of blocks separated by a single execution of atomic program B. Each block looks basically like a nice structure for some $\bar{n} = (n_o,\ldots n_{\ell+2})$; i.e., it consists of a large enough finite path of executions of atomic program A, upon which the n_i and their multiples are encoded with the aid of the P_i and S_i as above. Furthermore, P_o encodes m on each block, and $P_{\ell+1}$ and $P_{\ell+2}$ are forced to encode the values of f(a) and f(a+1) for some function f, where the block considered is the a'th from the start, beginning with a=0. Finally, $\underline{simulate}_\varphi$ is asserted to hold at the beginning state of each block.

Define the program block in RG as follows:

$$\underline{block}: \bigcup_{(i_o,\ldots,i_{\ell+2})} (A^*;\ P_{i_o}?;A^*;\ P_{i_1}?;\ldots;P_{i_{\ell+2}}?;\ A^*;B),$$

where the union is taken over all permutations $(i_o,\ldots,i_{\ell+2})$ of $\{0,1,\ldots,\ell+2\}$. For each $1 \leq i \leq \ell+2$, define the formulas P_i-behaves and S_i-behaves as follows, where A^+ abbreviates $A^*;A$:

$$P_i\text{-}\underline{behaves} = [A^*;\ P_i?;\ A^+]{\sim}P_i$$

$$S_i\text{-}\underline{behaves} = S_i \wedge ([A^*;\ P_i?]\ S_i \wedge [A^\Delta(P_i?;\ A^*;\ S_i?)\ A^\Delta]S_i) \wedge ([A^+;\ S_i?;\ A^+]{\sim}P_i \wedge [A^\Delta({\sim}S_i?;\ A^*;\ S_i?)A^\Delta]{\sim}S_i)$$

P_i-behaves prevents P_i from holding more than once on any A-path. If n_i is the distance between the start and the single state on some A-path which satisfies P_i, then S_i-behaves forces S_i (respectively by its conjuncts in order) to hold at the start, to hold at all reachable distances $a\cdot n_i$ for $a > 1$, and to hold at no reachable distances $a \cdot n_i + b$, for $a > 0$, $0 < b < n_i$. That is, S_i-behaves forces S_i to encode reachable multiples of n_i.

The formula $\underline{reduce}^m_\varphi$ is now defined to be:

$$[A]\ P_{\ell+1} \wedge [\underline{block}^*]\ (\langle \underline{block} \rangle\ \underline{true}$$

$$\wedge \bigwedge_{i=0}^{\ell+2} [A^*; A^\Delta(P_i?)((A;\sim S_i?)^*; A; S_i)^\Delta; (A;\sim S_i?)^*; B]\ \underline{false}$$

$$\wedge \bigwedge_{i=0}^{\ell+2} (P_i\text{-}\underline{behaves} \wedge S_i\text{-}\underline{behaves})$$

$$\wedge [A^m]\ P_o$$

$$\wedge [A^\Delta(P_{\ell+2}?;\ A^*;\ B)\ A^\Delta]\ P_{\ell+1}$$

$$\wedge \underline{simulate}_\varphi)\ .$$

Lemma 4.2: For any m, $\underline{reduce}^m_\varphi$ is satisfiable iff the formula

$\exists f\ (f(o) = 1 \wedge \forall x\ \exists y_1,\ldots,y_\ell\ \varphi\ (m, y_1,\ldots,y_\ell, f(x), f(x+1))$ is true.

Proof: (if) Let f be a function satisfying $f(o) = 1 \wedge \forall x \exists \bar{y} \varphi$. Construct the model S illustrated in Fig. 6. If we number the blocks of A's BL_o, $BL_1, \ldots$ each P_i, $0 \leq i \leq \ell+2$, is taken to hold at precisely one point on each block BL_a, and thus encodes a distance n^a_i from the beginning of the that block. On each block BL_a we choose $n^a_o = m$, $n^a_{\ell+1} = f(a)$, $n^a_{\ell+2} = f(a+1)$, and for $1 \leq i \leq \ell$ the value of n^a_i will be the value of y_i guaranteed to exist for $x = a$ by the truth of $\forall x \exists \bar{y} \varphi$. Furthermore, $n^o_{\ell+1} = 1$, thus capturing $f(o) = 1$. On each block BL_a, S_i will hold at precisely all distances which are multiples of n^a_i and which are still within the block. It is now easy to see that all but the $\underline{simulate}_\varphi$ conjuncts appearing in the definition of $\underline{reduce}^m_\varphi$ are true in the state u_o of S. In particular, $[A^\Delta(P_{\ell+2}?;\ A^*;\ B)A^\Delta]P_{\ell+1}$ holds at the beginning of each block by virtue of $n^a_{\ell+1} = n^{a+1}_{\ell+2} = f(a+1)$ holding. See Fig. 7. Also, the second conjunct in the parentheses prevents a block from ending before $n_i^{\ 2}$. Now, since $\underline{simulate}_\varphi$ contains no reference to B, and since any A-block in S can be regarded as a nice structure for $\bar{n} = (n^a_o,\ldots,n^a_{\ell+2})$, it follows from the (if) direction of Lemma 4.1 that $\underline{simulate}_\varphi$ also holds at the start state of any such block. Hence $S, u_o \models \underline{reduce}^m_\varphi$.

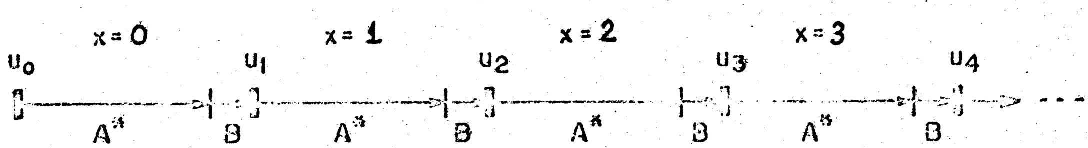

Figure 6.

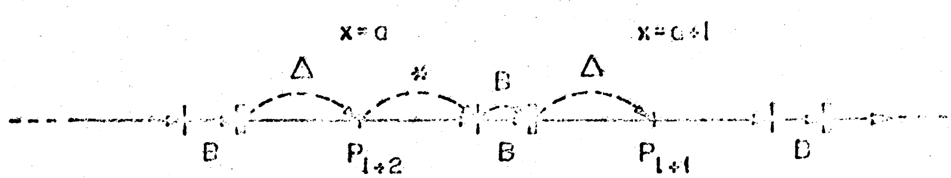

Figure 7.

(only if) Let $S, u_o \models \underline{reduce}^m_\varphi$. By [$\underline{block}$*] <$\underline{block}$> $\underline{true}$ there is an infinite (possibly cyclic) path p in S of the form A*BA*B..., and each P_i is true at least once on any maximal A-block of p. Furthermore, the next clause forces each such block to be at least as long as is required from a nice structure for the appropriate $\bar{n}$. Let u_a denote the start state of the a'th block of A's on the path p. See Fig. 6. By virtue of P_i-$\underline{behaves}$ holding at all states u_a, P_i cannot be true more than once in any block, thus we can denote by n_i^a the distance between u_a and the unique state satisfying P_i on the a'th block of p. By virtue of $[A^m]P_o$ being true at each u_a we know that $n_o^a = m$ for all a, and by $[A^\Delta(P_{\ell+2}?;\ A^*;\ B)A^\Delta]P_{\ell+1}$ we know that $n^a_{\ell+2} = n^{a+1}_{\ell+1}$.

We now define the function f with $f(a) = n^a_{\ell+1}$ for all a, and are guaranteed by the previous remark that $n^a_{\ell+2} = f(a+1)$. The reader can also verify that the truth of S_i-$\underline{behaves}$ at each u_a guarantees that S_i holds precisely at all multiples of n_i^a within the a'th block of A's on p. Thus each such block can be regarded as a nice model for $\bar{n} = (m,\ n_1^a,\dots,n_\ell^a,\ f(a),\ f(a+1))$.

By the (only if) direction of Lemma 4.1, the truth of $\underline{simulate}_\varphi$ at each u_a guarantees the truth of $\varphi(m,\ n_1^a,\dots,n_\ell^a,\ f(a),\ f(a+1))$. Thus, observing that the truth of $[A]P_{\ell+1}$ at u_o implies that $f(0) = 1$, we conclude that $\exists f(f(0) = 1 \wedge \forall x\ \exists y_1,\dots,y_\ell\ \varphi(m,y_1\dots,y_\ell,f(x),\ f(x+1)))$ is true. □

$\underline{\text{Corollary 4.3}}$: The validity problem for PDL_K is Π_1^1-hard. □

It is a standard exercise to verify that the problem is in Π_1^1. (For some details of such an exercise see Proposition 4.3 of [HPS].) We thus obtain

$\underline{\text{Theorem 4.4}}$: The validity problem for PDL_K is Π_1^1-complete. □

It is possible to push this proof technique further. One can simplify the programs of the form $\alpha^\Delta(\beta)\gamma^\Delta$ used in the above proof by suitably refining and complicating the block models constructed and the corresponding formula $\underline{reduce}^m_\varphi$. We breifly indicate how this can be done.

In general α,β and γ in programs of the form $\alpha^\Delta(\beta)\gamma^\Delta$ appearing in $\underline{reduce}^m_\varphi$ are not atomic. Although α is always the atomic A, β is invariably of the form Q?;A*;X, where X is either a test or B, and γ, when not atomic, expresses execution of a maximal block of $A;{\sim}S_i?$. These two complex forms of β and γ can be simplified as follows. For each i define the new atomic formula V_i to hold precisely at the first n_i distances which are multiples of n_i-1 . In this way, if $n_i \cdot n_j = n_k$ and $i \leq j$, V_j will hold at distance n_k-n_i, and S_j will hold (as will P_k) at distance n_k . This construction makes possible the replacement of the appropriate part of $\underline{simulate}_{x_i \cdot x_j = x_k}$ by $[A^\Delta(P_i?;A^*;P_j?;A^*;V_j?)A^\Delta](S_j \supset P_k)$. A similar replacement is possible in the second conjunct under [$\underline{block}$*] .

An additional formula, V_i-$\underline{behaves}$, forcing V_i to behave as described above, can be constructed using only atomic α and γ .

As far as making β atomic is concerned, one introduces, for each i, a new atomic formula Q_i holding at distance $\lfloor n_i/2 \rfloor$. With the aid of Q_i (easily forced to behave properly with an additional formula Q_i-behaves), one replaces, e.g. $[A^\Delta(P_i?;A^*;P_j A^\Delta]P_k$ with $[A^*;P_i?;A^\Delta(Q_k?)A^\Delta]P_j$ or $[A^*;P_i?;A^\Delta(Q_k?) A^\Delta;A]P_j$, depending upon the (easily tested) parity of n_k.

A similar device, involving a new atomic formula Q, true halfway through each block, can be used in conjunction with a clause which "copies" $n_{\ell+1}$ of each block at the end of the previous block with, say, R, to reduce $[A^\Delta(P_{\ell+2}?;\ A^*;\ B)A^\Delta]P_{\ell+1}$ to the form $[A^*;\ P_{\ell+2}?;\ A^\Delta(Q?)\ A^\Delta]R$.

These observations can be formalized to yield:

Proposition 4.5: If K' is the set of programs of K in which $\alpha^\Delta(\beta)\gamma^\Delta$ is allowed only in the form $A^\Delta(X)A^\Delta$, where X is either B or some atomic test $P?$, then the validity problem for $PDL_{K'}$ is Π^1_1-complete.

As remarked in the introduction, this result is actually true if X is always B. See [HPS].

Finally, we should remark that the nondeterminism present in the α^* and $\alpha^\Delta(\beta)\gamma^\Delta$ constructs of K is not essential for obtaining the results. The reader will notice that all uses of the $*$ and Δ constructs involve tests (or an application of B) to determine the number of iterations. It is possible to formalize this observation to yield:

Proposition 4.6: If K' is the set of programs of K in which $*$ is allowed only in the deterministic form $(P?;\alpha)^*;\sim P?$ and Δ only in the deterministic form $(\sim P?;\alpha)^\Delta(P?;\beta)\gamma^\Delta$, then the validity problem for $PDL_{K'}$ is Π^1_1-complete.

We close by remarking that the possible nondeterminism of the atomic programs A and B is of no help in the proofs, and appropriate versions of Theorem 4.1 and Propositions 4.1 and 4.2 where atomic programs are deterministic, trivially follow from the proofs of the original versions.

Acknowledgments:

We are grateful to A. Yehudai for his help concerning the formal-languages part of the paper; namely, Proposition 2.3 and references [L,Y,GF]. Y. Feldman pointed out an error in a previous version of Section 4.

References:

[dBM] deBakker, J.W. and L.G.L.T. Meertens. On the completeness of the inductive assertion method. J. Comp. Sys. Sciences, 11 (1975).pp.323-357.

[BPS] Bar-Hillel, Y., M. Perles and E. Shamir. On formal properties of simple phrase structure grammars. Z. Phonetik, Sprach. Kommunikation., 14 (1961), pp.143-172.

[DMR] Davis, M., Y. Matijasevic and J. Robinson. Hibert's tenth problem. Diophantive equations: positive aspects of a negative solution. Proc. Symp. Pure Math., Springer-Verlag Lecture Notes in Math., 28 (1976) pp.323-378.

[FL] Fischer, M.J. and R.E. Ladner. Propositional dynamic logic of regular programs. J. Comp. Sys. Sciences, 18 (1979), pp.194-211.

[GF] Greibach S. and E. Friedman. Super Deterministic PDA's : A Subcase with a decidable equivalence problem. J. ACM, 27 (1980), pp.675-700.

[HPS] Harel, D., A. Pnueli and J. Stavi. Further results on propositional dynamic logic of nonregular programs. Proc. Workshop on Logics of Programs (D. Kozen, ed.) Springer-Verlag, 1981. To appear.

[H] Harrison, M. Introduction to Formal Language Theory. Addison-Wesley, 1978.

[L] Linna, M. Two decidability results for deterministic pushdown automata. J. Comp. Sys. Sciences, 18 (1979), pp.92-107.

[M] Matijasevic, Y. Enumerable sets are diophantine. Soviet Math. Doklady, 11 (1970), pp.354-357.

[MSM] Meyer, A.R., R.S. Streett and G. Mirkowska. The deducibility problem in propositional dynamic logic. Proc. 8th Int. Colloq. on Autom. Lang. Prog., Springer-Verlag Lecture Notes in Computer Science, (1981).

[P1] Pratt, V.R. Semantical considerations on Floyd-Hoare logic. Proc. 17th IEEE Symp. on Found. of Comp. Sc., (1976), pp.109-121

[P2] Pratt, V.R. A near optimal method for reasoning about action. J. Comp. Sys. Sciences, 20, (1980), pp.231-254.

[R] Rogers, H.,Jr., Theory of Recursive Functions and Effective Computability. McGraw-Hill Co. New York, 1967.

[Y] Yehudai, A. The decidability of equivalence for a family of linear grammars. Inf. and Control, 47;2 (1981), pp.122-136.

Note added in proof: Recently T. Olshansky and the second author have been able to show that $PDL_{RG+\{A^{\Delta}B^{\Delta}\}}$ is decidable. This result will appear elsewhere.