Reprinted from JOURNAL OF COMPUTER AND SYSTEM SCIENCES
All Rights Reserved by Academic Press, New York and London

Vol. 26, No. 2, April 1983
Printed in Belgium

Propositional Dynamic Logic of Nonregular Programs

DAVID HAREL,*,† AMIR PNUELI,* AND JONATHAN STAVI†

Department of Applied Mathematics, The Weizmann Institute of Science, 76100 Rohovot, Israel and
†*Department of Mathematics and Computer Science, Bar-Ilan University, Ramat-Gan, Israel*

Received January 8, 1982; revised July 1982

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 nonregular languages. The results in this paper indicate that this line is extremely close to the original regular PDL. Moreover, the versions of PDL which we show to be beyond this borderline are shown to be actually very highly undecidable. The main results of the paper are: (a) The validity problem for PDL with the single additional context-free program $A^{\Delta}(B)A^{\Delta}$, for atomic programs A, B, defined as $\bigcup_{i>0} A^i; B; A^i$, is Π_1^1-complete. (b) There exists a recursive (but nonregular, and hence noncontext-free) one-letter program $L \subseteq A^*$ such that the validity problem for PDL with the single additional program L is Π_1^1-complete. Undecidability and Π_1^1-completeness of a less restricted version of PDL than the one in (a) are proved separately using different techniques.

1. INTRODUCTION

Propositional dynamic logic (PDL) is a formal logic for reasoning on a propositional level about programs. PDL was defined by Fischer and Ladner [4], based upon the first-order dynamic logic of Pratt [12], as a direct extension of the propositional calculus, in which assertations 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., [1]. The set of strings defined by a program $\alpha \in \mathrm{RG}$ is thought of as the set of possible sequences of atomic programs and tests constituting α. In the sequel this fixed version of PDL is denoted by $\mathrm{PDL}_{\mathrm{RG}}$.

In [4] it was shown that the validity problem for $\mathrm{PDL}_{\mathrm{RG}}$ is decidable. In fact, it is decidable in deterministic exponential time [13], and to within a polynomial this upper bound is the best possible [4].

Consider the set CF of context-free grammars over Σ. There is an analogous correspondence (see [1]) between recursive programs over Σ and context-free sets of strings over Σ, justifying the study of $\mathrm{PDL}_{\mathrm{CF}}$. Unfortunately, the equivalence and

0022-0000/83 $3.00

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., [5, 7, 8, 15]. 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 corresponding recursive program.

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

In Section 2 we define a general class K of programs which contains RG and the additional context-free programs $(\alpha^{\Delta}(\beta)\gamma^{\Delta})$ for $\alpha, \beta, \gamma \in$ RG. The new program is defined to contain all computations of $\alpha^i; \beta; \gamma^i$, for all $i \geqslant 0$. We observe that the inclusion and equivalence problems for the subsets of K used later in the paper to obtain undecidability of certain versions of PDL are decidable, so that these versions cannot be shown *undecidable* by Ladner's observation. We also show that these subsets lack the finite model property, so that they cannot be shown *decidable* by the finite model method of [4].

In Section 3 we use a reduction of the Post correspondence problem to show the undecidability of PDL_K.

In Section 4 we prove that PDL_K is actually Π_1^1-complete by reducing to satisfiability in PDL_K the truth of formulas of the form $\exists f \forall x P$, where P is a diophantine relation. That these formulas are universal Σ_1^1 (see [14]) follows from Matijasevic's theorem [9]. We also show how to improve this proof method obtaining a somewhat stronger version of the result.

The strongest version of this result is obtained in Section 5, where a direct encoding of certain infinite computations of nondeterministic Turing machines is used to yield the Π_1^1-completeness of PDL with the single additional program $A^{\Delta}(B)A^{\Delta}$ for atomic A and B. The proof can be slightly modified to yield Π_1^1-completeness of PDL with either the single additional program $L = \{ww^R \mid w \in \{A, B\}^*\}$, or both of $A^{\Delta}B^{\Delta}$ and $B^{\Delta}A^{\Delta}$. Here, e.g., $A^{\Delta}B^{\Delta}$ abbreviates A^{Δ} (*skip*) B^{Δ}.

In Section 6 we consider one-letter programs $L \subseteq A^*$ (which, in order to be nonregular have to also be noncontext-free). We exhibit a particular such program L and show that the addition to PDL_{RG} of L results in a Π_1^1-complete validity problem. Section 7 contains open problems.

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.

In response to a question in a preliminary version of this paper [6], a proof has been sketched in [11] that PDL with the single additional program $A^{\Delta}B^{\Delta}$ is decidable. Given this background, a comprehensive characterization of the classes of programs for which PDL is decidable remains an intriguing topic for future research.

We remark that the results of Sections 3 and 4 are subsumed by the main result of Section 5. Nevertheless, we present the proofs therein because of the simplicity of the first and the application of [9] in the second. Both might prove useful in obtaining future negative results for similar logics.

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 $\Sigma = \Pi \cup \{P? \mid P \in \Phi\} \cup \{{\sim}P? \mid P \in \Phi\}$. Let C be a given set of expressions, *called programs*, such that each program α is associated with some subset $L_C(\alpha)$ of Σ^*, or just $L(\alpha)$ when the context is clear. Throughout we assume $L(\theta) = \varnothing$.

The formulas of the *propositional dynamic logic of* C, denoted PDL_C, are defined as

(1) $\Phi \subseteq \mathrm{PDL}_C$,

(2) if $p, q \in \mathrm{PDL}_C$, then ${\sim}p, p \vee q \in \mathrm{PDL}_C$, and

(3) if $p \in \mathrm{PDL}_C$ and $\alpha \in C$, then $\langle\alpha\rangle p \in \mathrm{PDL}_C$.

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:

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

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

(3) $\rho({\sim}P?) = \{(u, u) \mid u \notin \pi(P)\}$, and

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

Given a structure S, the satisfiability relation is defined for all formulas of PDL_C as

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

(2) $u \vDash \sim p$ iff not $u \vDash p$,

(3) $u \vDash p \vee q$ iff either $u \vDash p$ or $u \vDash q$, and

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

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

Let RG be the set of regular expressions over Σ. The reader can easily check that $\mathrm{PDL}_{\mathrm{RG}}$ coincides with PDL, as defined, say, in [4], 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 $u \vDash \langle \alpha^* \rangle p$ iff $\exists i$, $u \vDash \langle \alpha^i \rangle p$.

A formula $p \in \mathrm{PDL}_C$ is *valid*, denoted $\vDash p$, if for every structure S and for every $u \in W^S$, $u \vDash p$; 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 $u \vDash p$. The latter is sometimes written $S, u \vDash p$.

The *inclusion* (resp. *equivalence*) *problem for* C is the problem of deciding, given $\alpha, \beta \in C$ whether or not $L(\alpha) \subseteq L(\beta)$ (resp. $L(\alpha) = L(\beta)$). The *validity problem* for PDL_C is the problem of deciding, given $p \in \mathrm{PDL}_C$, whether or not $\vDash p$.

Fischer and Ladner [4] have shown that every satisfiable formula p of $\mathrm{PDL}_{\mathrm{RG}}$ 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 [4] to show that the validity problem for $\mathrm{PDL}_{\mathrm{RG}}$ is decidable.

Let CF_0 (resp. CF) be the set of context-free grammars over terminals Π (resp. Σ) and some fixed set of nonterminals. It is well known that the equivalence (and hence also the inclusion) problem for CF_0 is undecidable [2]. This fact can be used to show that the validity problem for $\mathrm{PDL}_{\mathrm{CF}_0}$, and hence also for $\mathrm{PDL}_{\mathrm{CF}}$, is undecidable.

PROPOSITION 2.1 (due to R. Ladner). *For any* $\alpha, \beta \in \mathrm{CF}_0$, $P \in \Phi$, $\vDash (\langle \alpha \rangle P \supset \langle \beta \rangle P)$ *iff* $L(\alpha) \subseteq L(\beta)$.

Proof. 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 $S_x = (\{u_0, \ldots, u_k\}, \pi, \rho)$ such that $\pi(P) = \{u_k\}$, and such that for any $A \in \Pi$,

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

S_x is illustrated in Fig. 1. Clearly S_x, $u_0 \vDash \langle \alpha \rangle P$ and hence by assumption also S_x, $u_0 \vDash \langle \beta \rangle P$. But this implies that $x \in L(\beta)$. ∎

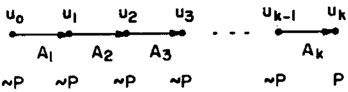

FIGURE 1

COROLLARY 2.2. *The validity problems for* $\mathrm{PDL}_{\mathrm{CF}_0}$ *and* $\mathrm{PDL}_{\mathrm{CF}}$ *are undecidable.*

We now define our set of programs K. It will become clear that $\mathrm{RG} < K < \mathrm{CF}$, where $C1 < C2$ whenever $\{L_{C1}(\alpha) \mid \alpha \in C1\} \subsetneqq \{L_{C2}(\alpha) \mid \alpha \in C2\}$.

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

When there is no ambiguity we will drop the additional parentheses. Sets of strings over Σ^* are associated with programs in K by

(1) $L_K(x) = \{x\}$, for $x \in \Sigma - \{\theta\}$, $L_K(\theta) = \varnothing$,
(2) $L_K(\alpha \cup \beta) = L_K(\alpha) \cup L_K(\beta)$,
(3) $L_K(\alpha; \beta) = L_K(\alpha)\, L_K(\beta) = \{xy \mid x \in L_K(\alpha),\ y \in L_K(\beta)\}$,
(4) $L_K(\alpha^*) = (L_K(\alpha))^* = \bigcup_{i \geq 0} L_K(\alpha^i)$, and
(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 $(\alpha^{\Delta}\gamma^{\Delta})$.

We would have liked to be able to state here that the inclusion and equivalence problems for K are decidable and thus that PDL_K cannot be proved undecidable by Proposition 2.1. However, an attempt to prove this has revealed some subtle problems with applying the appropriate results from, e.g., [5, 7, 8, 15] to K. All we can state here at this point is the following informal observation which can be proved by showing that all languages involved are *simple-deterministic stack uniform*, and then apply the results from [8].

PROPOSITION 2.3. *For all subsets K' of K used in the undecidability proofs in this paper, the inclusion and equivalence problems are decidable.*

It follows that none of our results, not even the mere undecidability of the versions of PDL involved, can be proved by Proposition 2.1.

We prove now that PDL_K cannot be shown decidable by the Fischer–Ladner method, since it lacks the small model property. Let *force* be the following formula of PDL_K:

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

PROPOSITION 2.4. Force *is satisfiable but has no finite model.*

Proof. Let S_0 be the structure illustrated in Fig. 2, in which the only states satisfying P are those marked ⊗. It is easy to see that $S, u \models \mathit{force}$. Assume now that $S, u \models \mathit{force}$, 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 labeled 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 labeling (A, B) paths connecting u with states satisfying P. Since S is finite, this is

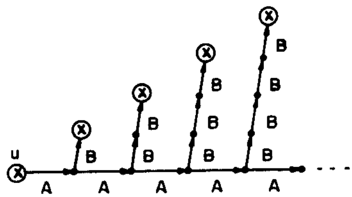

FIGURE 2

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 $\{A^iB^i \mid i \geqslant 0\}$. Finally, the first conjunct of *force* states that for each $i \geqslant 0$, A^iB^i is in U.

Hence $U = \{A^iB^i \mid i \geqslant 0\}$, 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 (PCPs) 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 $H = \{(x_1, y_1), ..., (x_n, y_n)\}$ be a PCP, where $x_i, y_i \in \{a, b\}^*$, for $1 \leqslant i \leqslant n$. A *solution* to H is a sequence $(i_1, ..., i_k)$, where $1 \leqslant i_j \leqslant n$ for $1 \leqslant j \leqslant k$, such that, denoting the reverse of a word $x \in \{a, b\}^*$ by x^R, we have $x_{i_1}, ..., x_{i_k} = y_{i_1}^R, ..., y_{i_k}^R$. Note that if $w = x_{i_1}, ..., x_{1_k}$, then $w^R = y_{i_k}, ..., y_{i_1}$. It is easy to relate the classical formulation of PCP to our slightly modified version. We shall construct a formula $reduce_H \in$ 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 $P, Q, R_1, ..., R_n$. The letters a and b will be encoded as the programs $A; {\sim}Q?$ 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. $R_1, ..., R_n$ 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 words from among the y's, with the same number of words and the same total length and such that indices of words in both blocks correspond.

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

$$C^A(a) = A; Q?, \qquad C^A(b) = A; {\sim}Q?, \qquad C^A(z_1 z_2) = C^A(z_1)\, C^A(z_2).$$

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

FIGURE 3

Define

$$L_x = \bigcup_{1 \leqslant i \leqslant n} (R^{(i)}; C^A(x_i)), \qquad L_y = \bigcup_{1 \leqslant i \leqslant n} (C^B(y_i); R^{(i)})$$

Now, let $reduce_H$ be the conjunction of the formulas

exist-path:	$\sim P \wedge \langle L_x^\Delta L_y^\Delta \rangle P$
indices-correspond:	$[L_x^*; R^{(i)}?; L_x^\Delta L_y^\Delta]\, R^{(i)};$
same-length:	$[A; A^\Delta B^\Delta; B]\, P \wedge [A^*; A; A^\Delta B^\Delta] \sim P$ $\wedge\, [(A \cup B)^*; P?; (A \cup B)]\, false,$
same-word:	$[A^*; A; Q?; A^\Delta B^\Delta; B]\, Q$ $\wedge\, [A^*; A; \sim Q?; A^\Delta B^\Delta; B] \sim Q.$

LEMMA 3.1. *For any $H = \{(x_1, y_1),..., (x_n, y_n)\}$, H has a solution* iff reduce$_H$ *is satisfiable.*

Proof. if. Assume $S, u \models reduce_H$. By *exist-path* there is a nonempty path p in S, starting at u, which encodes in order the words $x_{i_1},..., x_{i_k}$ for some $k > 0$ and some $i_1,..., i_k$, using A, followed by $y_{j_k},..., y_{j_1}$ for some $j_1,..., j_k$, encoded using B. Furthermore, by *same-length* we know (resp. 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 $A^i B^j$ 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, $|x_{i_1},..., x_{i_k}| = |y_{j_k},..., y_{j_1}|$. By *indices-correspond* considered along path p, we have $i_l = j_l$. Finally, by *same-word* considered along p we conclude that $x_{i_1},..., x_{i_k} = (y_{i_k},..., y_{i_1})^R = y_{i_1}^R,..., y_{i_k}^R$.

only if. Let $(i_1,..., i_k)$ 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 $S, u \models reduce_H$. ∎

COROLLARY 3.2. *The validity problem for* PDL$_K$ *is undecidable.*

4. PDL$_K$ IS Π_1^1-COMPLETE

In this section we reduce to satisfiability in PDL$_K$ the truth of formulas $F(m)$ of the form $\exists f(f(0) = 1 \wedge \forall x P)$, where $P(m, f(x), f(x+1))$ is a diophantine relation involving m and the two values of f: $f(x)$ and $f(x+1)$.

In the Appendix it is shown that $\exists f(f(0)=1 \wedge \forall x R)$ is a universal Σ_1^1-formula, where R is a (primitive) recursive relation of $m, f(x)$, and $f(x+1)$. Replacing R by a diophantine relation P follows from Matijasevic's theorem [3, 9]. Moreover, the relation 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 $t_i t_j = t_k$, where the t's are from among $m, f(x), f(x+1)$, and new variables $y_1, \ldots, y_l$ which are existentially quantified, i.e., $P \equiv \exists \bar{y} \varphi$. Here l depends on the equation P.

In the sequel $\varphi(x_0, \ldots, x_{l+2})$ will denote a conjunction of such equalities over $x_0, \ldots, x_{l+2}$. 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 $reduce_\varphi^m$ of PDL_K, effectively depending on m, which is satisfiable iff $\exists f(f(0)=1 \wedge \forall x \exists y_1, \ldots, \exists y_l \varphi(m, y_1, \ldots, y_l, f(x), f(x+1))$ is true.

First we show how to simulate the conjunction $\varphi(x_0, \ldots, x_{l+2})$ by a PDL_K formula on particularly well-behaved structures.

Let $\bar{n}=(n_0, \ldots, n_{l+2})$ be an arbitrary tuple of natural numbers. A *nice structure for* $\bar{n}$ is any structure $S=(W, \pi, \rho)$ such that there exists $p \geqslant \max_i(n_i^2)$ and $\{u_0, \ldots, u_p\} \subseteq W$, $\{(u_i, u_{i+1}) \mid 0 \leqslant i < p\} \subseteq \rho(A)$, $\rho(A)$ is functional (i.e., A is deterministic in S) $u_i \in \pi(P_j)$ iff $i=n_j$, and $u_i \in \pi(S_j)$ iff $i=an_j$ for some $a \geqslant 0$. In other words, the "A-part" of S (termed the *A cut of S from* u_0 in [10]) 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_0, and S_j encodes similarly all multiples of n_j which fall within the segment. Given φ, define the formula $simulate_\varphi$ inductively on the structure of φ as

$$\begin{aligned}
simulate_{\varphi \wedge \varphi'} &= simulate_\varphi \wedge simulate_{\varphi'}, \\
simulate_{x_i=0} &= P_i, \\
simulate_{x_i=1} &= [A]P_i, \\
simulate_{x_i+x_j=x_k} &= [A^\Delta(P_i?; A^*; P_j?)A^\Delta]P_k \wedge [A^\Delta(P_j?; A^*; P_i?)A^\Delta]P_k, \\
simulate_{x_i x_j=x_k} &= ((P_i \vee P_j) \supset P_k) \\
&\quad \wedge [A; A^\Delta(P_i?; A^*; P_j?)((A; {\sim}S_j?)^*; A; S_j?)^\Delta]P_k \\
&\quad \wedge [A; A^\Delta(P_j?; A^*; P_i?)((A; {\sim}S_i?)^*; A; S_i?)^\Delta]P_k.
\end{aligned}$$

LEMMA 4.1. *For any* $\bar{n}=(n_0, \ldots, n_{l+2})$, $S, u_0 \models$ simulate$_\varphi$ *for all nice structures* S *for* $\bar{n}$, iff $\varphi(\bar{n})$ *is true.*

Proof. only if. Let S be nice for $\bar{n}$, and let $S, u_0 \models simulate_\varphi$. 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 $S, u_0 \models [A]P_i$, 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 $simulate_{x_i+x_j=x_k}$ can be seen to state that when $n_i \leqslant n_j$ (i.e., P_i becomes true before P_j when traversing the u branch of the structure S starting from u_0) we have in fact $n_i+(n_j-n_i)+n_i=n_k$, and that when $n_j \leqslant n_i$, $n_j+(n_i-n_j)+n_j=n_k$. In either case $n_i+n_j=n_k$. Figure 4 illustrates this case.

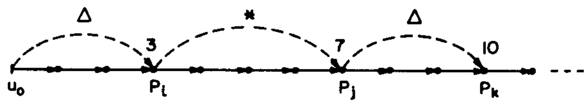

FIGURE 4

For the case where φ is of the form $x_i x_j = x_k$, the formula $simulate_{x_i x_j = x_k}$ states that if one of n_i or n_j is 0, then so is n_k, and if $0 < n_i \leqslant n_j$, then $1 + (n_i - 1) + (n_j - n_i) + (n_i - 1)\, n_j = n_k$, and if $0 < n_j \leqslant n_i$, then $1 + (n_j - 1) + (n_i - n_j) + (n_j - 1)\, n_i = n_k$. In either case $n_i n_j = n_k$. Figure 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 · should not be vacuously true.

if. Assume $\varphi(\bar{n})$ is true. Let $S_{\bar{n}}$ be any nice structure for $\bar{n}$, and consider u_o. By induction on the structure of φ one shows that $S_{\bar{n}}, u_0 \models simulate_\varphi$. 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 conjuct of $simulate_{x_i + x_j = x_k}$ is true in u_0 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 $reduce^m_\varphi$. The idea is to force models of $reduce^m_\varphi$ 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 $\bar{n} = (n_0, ..., n_{l+2})$; 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_0 encodes m on each block, and P_{l+1} and P_{l+2} are forced to encode the values of $f(a)$ and $f(a+1)$ for some function f, where the block considered is the ath from the start, beginning with $a = 0$. Finally, $simulate_\varphi$ is asserted to hold at the beginning state of each block.

Define the program *block* in RG as

$$block\colon \quad \bigcup_{i_0, \ldots, i_{l+2}} (A^*; P_{i_0}?; A^*; P_{i_1}?; \ldots; P_{i_{l+2}}?; A^*; B),$$

where the union is taken over all permutations $(i_0, ..., i_{l+2})$ of $\{0, 1, ..., l+2\}$. For each

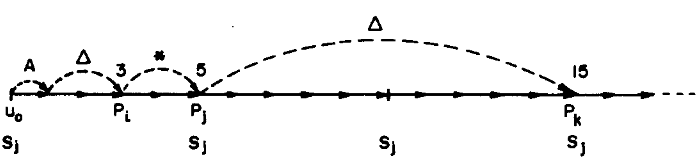

FIGURE 5

$1 \leqslant i \leqslant l+2$, define the formulas P_i-*behaves* and S_i-*behaves* as follows, where A^+ abbreviates $A^*;A$:

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

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 (resp. by its conjuncts in order) to hold at the start, to hold at all reachable distances an_i for $a > 1$, and to hold at no reachable distances $an_i + b$, for $a > 0$, $0 < b < n_i$. That is, S_i-*behaves* forces S_i to encode reachable multiples of n_i.

The formula $reduce^m_\varphi$ is now defined to be

$$[A]\, P_{l+1} \wedge [block^*]((\langle block\rangle\, true$$
$$\wedge \bigwedge_{i=0}^{l+2} [A^*; A^\Delta(P_i?)((A; \sim S_i?)^*; A; S_i)^\Delta; (A; \sim S_i?)^*; B]\, false$$
$$\wedge \bigwedge_{i=0}^{l+2} (P_i\text{-}behaves \wedge S_i\text{-}behaves)$$
$$\wedge [A^m]\, P_0$$
$$\wedge [A^\Delta(P_{l+2}?; A^*; B)A^\Delta]\, P_{l+1}$$
$$\wedge\ simulate_\varphi).$$

LEMMA 4.2. *For any m,* reduce$^m_\varphi$ *is satisfiable* iff *the formula*

$$\exists f(f(0) = 1 \wedge \forall x \exists y_1, \ldots, y_l \varphi(m, y_1, \ldots, y_l, f(x), f(x+1))$$

is true.

Proof. if. Let f be a function satisfying $f(0) = 1 \wedge \forall x \exists \bar{y} \varphi$. Construct the model S partly illustrated in Fig. 6. If we number the blocks of A's $\mathrm{BL}_0, \mathrm{BL}_1, \ldots$, each P_i, $0 \leqslant i \leqslant l+2$, is taken to hold at precisely one point on each block BL_a, and thus encodes a distance n_i^a from the beginning of that block. On each block BL_a we choose $n_0^a = m$, $n_{l+1}^a = f(a)$, $n_{l+2}^a = f(a+1)$, and for $1 \leqslant i \leqslant l$ the value of n_i^a will be the value of y_i guaranteed to exist for $x = a$ by the truth of $\forall x \exists \bar{y} \varphi$. Furthermore, $n_{l+1}^0 = 1$, thus capturing $f(0) = 1$. On each block BL_a, S_i will hold at precisely all

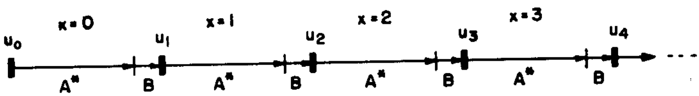

FIGURE 6

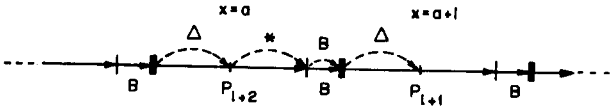

FIGURE 7

distances which are multiples of n_l^a and which are still within the block. It is now easy to see that all but possibly the *simulate*$_\varphi$ conjuncts appearing in the definition of *reduce*$_\varphi^m$ are true in the state u_0 of S. In particular, $[A^\Delta(P_{l+2}?;A^*;B)A^\Delta]\,P_{l+1}$ holds at the beginning of each block by virtue of $n_{l+1}^a = n_{l+2}^{a+1} = f(a+1)$ holding. See Fig. 7. Also, the second conjunct in the parentheses prevents a block from ending before n_l^2. Now, since *simulate*$_\varphi$ contains no reference to B, and since any A block in S can be regarded as a nice structure for $\bar{n} = (n_0^a,..., n_{l+2}^a)$, it follows from the *if* direction of Lemma 4.1 that *simulate*$_\varphi$ also holds at the start state of any such block. Hence $S, u_0 \models$ *reduce*$_\varphi^m$.

only if. Let $S, u_0 \models$ *reduce*$_\varphi^m$. By $[block^*]\langle block\rangle true$ there is an infinite (possibly cyclic) path p in S of the form $A^*BA^*B\cdots$, 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 ath block of A's on the path p. See Fig. 6. By virtue of P_i-*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 ath block of p. By virtue of $[A^m]\,P_0$ being true at each u_a we know that $n_0^a = m$ for all a, and by $[A^\Delta(P_{l+2}?;A^*;B)A^\Delta]\,P_{l+1}$ we know that $n_{l+2}^a = n_{l+1}^{a+1}$.

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

By the *only if* direction of Lemma 4.1, the truth of *simulate*$_\varphi$ at each u_a guarantees the truth of $\varphi(m, n_1^a,..., n_l^a, f(a), f(a+1))$. Thus, observing that the truth of $[A]\,P_{l+1}$ at u_0 implies that $f(0) = 1$, we conclude that $\exists f(f(0) = 1 \wedge \forall x \exists y_1,..., y_l\,\varphi(m, y_1,..., y_l, f(x+1)))$ is true. ∎

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 Lemma 6.3.) We thus obtain

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 $reduce^m_\varphi$. We briefly indicate how this can be done.

In general α, β, and γ in programs of the form $\alpha^\Delta(\beta)\gamma^\Delta$ appearing in $reduce^m_\varphi$ 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 $n_i n_j = n_k$ and $i \leqslant j$, 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 $simulate_{x_i x_j = x_k}$ 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 [block*].

An additional formula, V_i-*behaves*, forcing V_i to behave as decribed 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_{l+1} of each block at the end of the previous block with, say, R, to reduce $[A^\Delta(P_{l+2}?; A^*; B)A^\Delta]P_{l+1}$ to the form $[A^*; P_{l+2}?; A^\Delta(Q?)A^\Delta]R$.

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.*

We see no way of obtaining the stronger version given in Section 5, using the present proof technique.

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* $\mathrm{PDL}_{K'}$ *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 4.4 and Propositions 4.5 and 4.6, where atomic programs are deterministic, trivially follow from the proofs of the original versions.

5. $\mathrm{PDL}_{\mathrm{RG}+\{A^\Delta(B)A^\Delta\}}$ IS Π_1^1-COMPLETE

First we show that the existence of certain infinite computations for nondeterministic Turing machines is a Σ_1^1-complete problem. We then reduce this problem to the satisfiability of formulas in $\mathrm{PDL}_{\mathrm{RG}+\{A^\Delta(B)A^\Delta\}}$. Let $\{T_m\}$, $m \in N$, be an effective enumeration of the (nondeterministic) Turing machines.

PROPOSITION 5.1. *The set $G = \{m \mid T_m$, starting on an empty tape, has an infinite computation which repeats its start state infinitely often$\}$ is Σ_1^1-complete.*

Sketch of Proof (in Σ_1^1). Given m, consider the Σ_1^1-formula φ_m: $\exists f(f(0) = C \wedge \forall x \exists y\, g_m(y, f(x), f(x+1)))$, where C encodes the initial empty-tape configuration of T_m, and $g_m(y, v, w)$ is the (recursive) predicate true if y encodes a legal segment of computation of T_m starting at the configuration encoded by v and ending in that encoded by w, and, moreover, the states in both v and w are the start state of T_m. Clearly, φ_m is true iff $m \in G$.

Complete in Σ_1^1. Consider formulas of the form φ: $\exists f(f(0) = 1 \wedge \forall x\, g(f(x), f(x+1)))$, for recursive g. That these are universal Σ_1^1-formulas follows from Claim 1 in the Appendix.

For any such φ construct a nondeterministic Turing machine which, starting on the empty tape, initially writes down $x = 0$ and $f(x) = 1$, and then keeps indefinitely augmenting x and looking nondeterministically for a new value for $f(x+1)$ satisfying g. Whenever it finds such $f(x+1)$ it signals by reentering its start state. Clearly, φ is true iff $m \in G$, where T_m is the Turing machine just constructed. ∎

Given a nondeterministic Turing machine T we shall now construct a formula $reduce_T$ in $\mathrm{PDL}_{\mathrm{RG}+\{A^\Delta(B)A^\Delta\}}$ and show that T has the property described in Proposition 5.1 iff $reduce_T$ is satisfiable; hence satisfiability (resp. validity) in $\mathrm{PDL}_{\mathrm{RG}+\{A^\Delta(B)A^\Delta\}}$ is Σ_1^1-complete (resp. Π_1^1-complete).

Let the tape alphabet Σ of T include the blank symbol $\not b$, and let V be the set of states, with q_0 the start state. (We hope the use of the symbol Σ in this section will not cause the reader to confuse it with its different use in the definitions of Section 2.) Denote $\Sigma_V = \Sigma \cup V$. A configuration of T can be represented by the nonblank portion of the tape surrounded on either side by at least one $\not b$, and with the current state inserted just prior to the symbol being read. The initial configuration can thus be represented by $\not b\, q_0\, \not b$. The transition table is given by a yield function $\delta: \Sigma \times V \times \Sigma \to 2^{(\Sigma_V)^3}$ such that a configuration $c = x\sigma q\tau z$, for $x, z \in \Sigma^*$, $\sigma, \tau \in \Sigma$ and $q \in V$, can result in a configuration $xy^R z$ for each $y \in \delta(\sigma, q, \tau)$. Let $\bar{\delta}(\sigma, q, \tau) = \Sigma_V^3 - \delta(\sigma, q, \tau)$. Clearly, for every triple σ, q, τ, both $\delta(\sigma, q, \tau)$ and $\bar{\delta}(\sigma, q, \tau)$ are finite.

Our formula $reduce_T$ will involve atomic programs A and B, and atomic propositions P_σ for each $\sigma \in \Sigma$ and P_q for each $q \in V$. We let $C(\sigma)$ stand for the program $A; P_\sigma?$, and similarly for $C(q)$. C is extended to strings over Σ_V, and to sets of such strings by $C(xy) = C(x); C(y)$ and $C(W) = \bigcup_{w \in W} C(w)$.

The idea of the reduction is to force models of $reduce_T$ to contain an encoding of

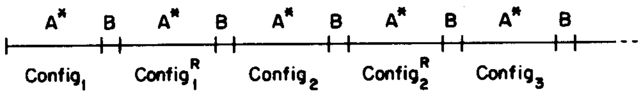

FIGURE 8

the infinite computation of T sought for, in the form of an infinite (possibly cyclic) sequence of executions of A and B of the form $p = A^*BA^*BA \cdots$. The odd numbered blocks of A's in p encode successive configurations of the computation, and the even blocks encode the reflections around B of their respective previous blocks. The new program $A^\Delta(B)A^\Delta$ is used to force p to contain correct transitions between successive configurations, correct reflections between reflected configurations, and also to ensure a length increase in the blocks of A's to make possible extension of the nonblank portion of the tape. See Fig. 8.

Define the program *config* to be

$$C(\not{b});\ C(\Sigma)^*;\ C(V);\ C(\Sigma)^*;\ C(\not{b});\ B.$$

The program *good-config* is defined in the same way but with $C(q_0)$ replacing $C(V)$.

The formula $reduce_T$ is taken to be the conjunction of the following formulas:

∃computation: $[config^*]\langle config^*; good\text{-}config\rangle\, true$

single letter: $[(A \cup B)^*; A]\left(\bigvee_{a \in \Sigma \cup V}\left(P_a \wedge \bigwedge_{\substack{b \in \Sigma \cup V \\ b \neq a}} \sim P_b\right)\right)$

start: $\bigwedge_{\substack{a \in \Sigma \cup V \\ a \notin \{\not{b}, q_0\}}} [A^*; C(a)]\, false$

lengthen: $[config^*]\,([A^*; A; A^\Delta(B)A^\Delta; A; B]\, false$

$\wedge\ [A^\Delta(B)A^\Delta; A; A]\, false$

$\wedge\ [A^\Delta(B)A^\Delta; A]\, P_{\not{b}})$

reflection: $\bigwedge_{a \in \Sigma \cup V} [(config; config)^*; A^*; C(a); A^\Delta(B)A^\Delta; A]\, P_a$

transition: $\bigwedge_{\sigma, \tau, \sigma' \in \Sigma} [(config; config)^*; config; A^*;$

$C(\sigma\tau\sigma'); A^\Delta(B)A^\Delta; A; A]\, P_\tau$

$\wedge\ \bigwedge_{\substack{\sigma, \tau \in \Sigma \\ q \in V}} [(config; config)^*; config; A^*;$

$C(\sigma q \tau); A^\Delta(B)A^\Delta; C(\bar{\delta}(\sigma, q, \tau))]\, false$

LEMMA 5.2. *The formula* reduce$_T$ *is satisfiable* iff *there exists an infinite computation of T, starting on the empty tape, which repeats the start state q_0 infinitely often.*

Proof. *if.* Let $c_1, c_2,...,$ be a representation of the successive configurations of such a computation of T. Without loss of generality assume that for each i, $|c_{i+1}| = |c_i| + 2$, and that the two extra elements in c_{i+1} represent an added $b\!\!/$ on either side of c_i. Let c_i' be $c_i^R b\!\!/$. Then clearly, $c_{i+1} = b\!\!/ c_i'' b\!\!/$, where c_i'' is a direct outcome of c_i by the transition table of T. Construct the model S such that its only executions of A and B are given by an infinite sequence, starting at some state u, of the form $A^{|c_1|}BA^{|c_1|+1}BA^{|c_2|}B \cdots$, upon which c_1, c_1', c_2, $c_2',...,$ are encoded exclusively by the appropriate atomic propositions. For example, if $c_1 = b\!\!/ q_0 b\!\!/$, then we might view the initial part of the model as an execution of A; $P_{b\!\!/}$?; A; P_{q_0}?; A; $P_{b\!\!/}$?; B; A; $P_{b\!\!/}$?; A; P_{q_0}?; A; $P_{b\!\!/}$?; A; $P_{b\!\!/}$; B;.... We leave the reader to check that all conjuncts of *reduce$_T$* are true in S at state u. In particular, since q_0 repeats infinitely often, *good-config* can be executed infinitely often in the model, contributing to the truth at u of $\exists$*computation*. Hence $S, u \models reduce_T$.

only if. Let $S, u \models reduce_T$. By $\exists$*computation* there is an infinite (possibly cyclic) sequence of executions of A and B, starting at u, of the form $p = A^{i_1}BA^{i_2}B \cdots$. By *lengthen* we have $i_{j+1} = i_j + 1$ for all j. By *single-letter* there is an element of $\Sigma \cup V$ associated with each execution of A along p, enabling us to think of p as representing a sequence c_1, c_1', c_2, $c_2',...,$ of words over $\Sigma \cup V$. Consequently, by $\exists$*computation* and the structure of *config*, each such word contains exactly one state in V and hence actually encodes a configuration of T. By *start*, the word c_1 must be of the form $b\!\!/ b\!\!/^* q_0 b\!\!/^* b\!\!/$, which represents a start configuration. By *reflection* we have $c_i' = c_i^R b\!\!/$. Now, the first conjunct of *transition* ensures retainment of those parts of the tape of T untouched by a transition from c_i to c_{i+1}, and the second conjunct ensures that this transition is indeed according to the yield function δ. Finally, $\exists$*computation* ensures the occurrence of "good" configurations infinitely often along p, and hence that q_0 repeats infinitely often during the computation $c_1, c_2,....$ ∎

Following immediately from Proposition 5.1 and Lemma 5.2, observing the obvious containment in Π_1^1, we have

THEOREM 5.3. *The validity problem for* PDL$_{RG+\{A^\Delta(B)A^\Delta\}}$ *is Π_1^1-complete.*

It is quite straightforward to modify the proof of Theorem 5.3 in such a way that rather than a sequence of executions of the form $A^{|c_1|}BA^{|c_1|+1}BA^{|c_2|}B \cdots$, we have a sequence of the form $A^{|c_1|}B^{|c_1|+1}A^{|c_2|}B^{|c_2|+1} \cdots$, with the configurations encoded using the A's and their reflections encoded using the B's. All ocurrences $A^\Delta(B)A^\Delta$ are replaced by the appropriate ones of $A^\Delta B^\Delta$ or $B^\Delta A^\Delta$. Further easy modifications of *lengthen* are required. In this way one obtains

PROPOSITION 5.4. *The validity problem for* PDL$_{RG+\{A^\Delta B^\Delta, B^\Delta A^\Delta\}}$ *is Π_1^1-complete.*

By replacing a single B in the proof of Theorem 5.3 with a double $B; B$, it is possible to obtain the same result for the additional program $L = \{w; w^R \mid w \in \{A, B\}^*\}$. Each $A^\Delta(B)A^\Delta$ is simply replaced by L, and along the path $A \cdots ABBA \cdots ABBA \cdots$ of interest, computations of L coincide with those of $A^\Delta(B; B)A^\Delta$. Various other linear context-free grammars give rise to simple programs whose addition to RG results in Π_1^1-completeness. In particular, one can define infinite classes of such programs each of which has the above Π_1^1 property. For example, $C = \{L \mid L \text{ is of the form } \{A^iBA^{ki} \mid i \geqslant 0, \text{ fixed } k\}\}$. In each case the aforementioned proof goes through slightly modified.

6. Π_1^1-Completeness over One Atomic Program

In this section we consider the decision problem for validity in PDL_C, where the set C of programs consists of $\mathrm{RG}(A)$ (the regular expressions over the single letter A) together with finitely many additional programs denoted by the symbols $\Gamma_1, ..., \Gamma_k$, which are interpreted by (not necessarily regular) subsets of A^*. Thus, the semantics of PDL_C is determined by a list $S_1, ..., S_k$ of subsets of ω ($\omega = \{0, 1, 2, ...\}$) which serve to interpret the programs Γ_i as follows: $L(\Gamma_i) = \{A^n \mid n \in S_i\}$. Satisfaction of formulas by states in a given PDL-structure is now defined as in Section 2. To obtain undecidability results we shall assume that the language of PDL_C has as many atomic propositions as are needed for the proofs presented below. They will be denoted by P, P_0, P_1, $P_2, ...$, etc.

Note that the sets $S_1, ..., S_k$ are only needed for specifying the semantics of PDL_C and do not figure in the syntax. Nevertheless, we shall write A^{S_i} instead of Γ_i to emphasize that the interpretation $L(\Gamma_i) = \{A^n \mid n \in S_i\}$ is being used.

For $S \subseteq \omega$, we denote $\bar{S} = \omega - S$ (the complement of S). For $S_1, S_2 \subseteq \omega$, we write $S_1 \leqslant_m S_2$, and say that S_1 is many-one reducible to S_2, if there is a total recursive function $f\colon \omega \to \omega$ such that

$$\forall n (n \in S_1 \Leftrightarrow f(n) \in S_2).$$

Note that if $S_1 \leqslant_m S_2$, then clearly, S_1 is recursive in S_2, that is, membership in S_1 is decidable using a Turing machine with an oracle from membership in S_2. Sometimes one of the sets S_1, S_2 is a set of strings over some finite alphabet (e.g., formulas of some language) and is identified with the set of Gödel numbers of its members, so that the notation $S_1 \leqslant_m S_2$ still makes sense.

Given $S_1, ..., S_k \subseteq \omega$ we denote by $\mathrm{vld}(S_1, ..., S_k)$ the set of all logically valid PDL_C formulas, where $C = \mathrm{RG}(A) \cup \{A^{S_1}, ..., A^{S_k}\}$, as described above. Similarly $\mathrm{stl}(S_1, ..., S_k)$ is the set of all satisfiable PDL_C formulas. Clearly, a PDL_C formula Q is valid iff $\sim Q$ is unsatisfiable, hence each of the above two sets of formulas is recursive in the other. We shall study the complexity of $\mathrm{vld}(S_1, ..., S_k)$ and especially of $\mathrm{vld}(S)$ (the case $k = 1$) for a given complexity of $S_1, ..., S_k$ or of S.

The main results (some of which are trivial observations) are summarized in Lemmas 6.1–6.3 and Theorem 6.4.

LEMMA 6.1. *For any* $S_1,\ldots, S_k \subseteq \omega$ *and* $1 \leqslant i \leqslant k$, $S_i \leqslant_m \mathrm{vld}(S_i) \leqslant_m \mathrm{vld}(S_1,\ldots, S_k)$. *Hence, if* $\mathrm{vld}(S_1,\ldots, S_k)$ *is decidable, then each set* S_i *is recursive.*

LEMMA 6.2. *Let* $S_1,\ldots,S_k \subseteq \omega$, $k > 1$ *and let* $S = \{kn - i \mid 1 \leqslant i \leqslant k,\ 1 \leqslant n \in S_i\}$. *Then* $\mathrm{vld}(S_1,\ldots, S_k) \leqslant_m \mathrm{vld}(S)$.

LEMMA 6.3. *If* $S_1,\ldots, S_k$ *are recursive (or even merely* Δ_1^1*) subsets of* ω, *then* $\mathrm{vld}(S_1,\ldots, S_k)$ *is a* Π_1^1 *set.*

THEOREM 6.4. *There exists a primitive recursive set* $S \subseteq \omega$ *such that* $\mathrm{vld}(S)$ *is a complete* Π_1^1 *set.*

Note that Theorem 6.4 shows that for recursive S $\mathrm{vld}(S)$ may sometimes be as complex as is allowed for by Lemma 6.3.

Proof of Lemma 6.1. Note that $n \in S_i$ iff the formula

$$[A^{S_i}]\, P \supset [A^n]\, P$$

is valid, hence $S_i \leqslant_m \mathrm{vld}(S_i)$. The rest of the lemma is obvious. ∎

Proof of Lemma 6.2. It will suffice to prove that $\mathrm{stl}(S_1,\ldots, S_k) \leqslant_m \mathrm{stl}(S)$, in view of the connection between validity and satisfiability mentioned earlier. Observe now that if $0 \in S_i$ and we let $S_i' = S_i - \{0\}$, then $[A^{S_i}] \equiv p \wedge [A^{S_i'}]\, p$ and $\langle A^{S_i}\rangle p \equiv p \vee \langle A^{S_i'}\rangle p$ are valid for any formula p, hence $\mathrm{PDL}_{\mathrm{RG}(A)\cup\{A^{S_1},\ldots,A^{S_i},\ldots,A^{S_k}\}}$ is translatable to $\mathrm{PDL}_{\mathrm{RG}(A)\cup\{A^{S_1},\ldots,A^{S_i'},\ldots,A^{S_k}\}}$, and hence $\mathrm{stl}(S_1,\ldots, S_k) \leqslant_m \mathrm{stl}(S_1,\ldots, S_i',\ldots, S_k)$. Thus, by successive applications of this process, we see that $\mathrm{stl}(S_1,\ldots, S_k) \leqslant_m \mathrm{stl}(S_1 - \{0\},\ldots, S_k - \{0\})$ and since the set S in Lemma 6.2 depends only on the nonzero numbers of $S_1,\ldots, S_k$ there is no loss of generality in assuming that $0 \notin S_i$ (for $i = 1,\ldots, k$) from the start.

Suppose now that a formula Q of $\mathrm{PDL}_{\mathrm{RG}(A)\cup\{A^{S_1},\ldots,A^{S_k}\}}$ is given. We want to associate with Q, in an effective way, a formula $\tilde{Q}$ of $\mathrm{PDL}_{\mathrm{RG}(A)\cup\{A^S\}}$ so that Q is satisfiable iff $\tilde{Q}$ is satisfiable. To make $\tilde{Q}$ more intelligible we write it as a formula of $\mathrm{PDL}_{\mathrm{RG}(B)\cup\{B^S\}}$. The idea is that the role of A in Q will be played by B^k in $\tilde{Q}$.

$\tilde{Q}$ is the conjunction of the following formulas, where $P_0,\ldots, P_{k-1}$ are new atomic propositions (not occurring in Q):

(1) P_0,

(2) $[B^*](P_0 \vee \cdots \vee P_{k-1})$,

(3) $[B^*] \bigwedge_{0 \leqslant i < j < k} \sim (P_i \wedge P_j)$,

(4) $[B^*] \bigwedge_{0 \leqslant i < k} (P_i \supset [B]\, P_{i+1})$ (for $i = k - 1$ P_{i+1} is taken to be P_0), and

(5) Q_1.

Here Q_1 is obtained from Q by the following replacements: Substitute B^k for A everywhere in Q. Also, wherever A^{S_i} occurs in Q replace it by $P_0?;\ B^S;\ B^i;\ P_0?$. Thus, $[A^{S_i}]$ is replaced by $[P_0?][B^S][B^i; P_0?]$ and $\langle A^{S_i}\rangle$ is replaced similarly. (The idea is that to perform B^k n times for some $n \in S_i$ we perform B $m+i$ times for some $m \in S$ such that $m+i \equiv 0 \pmod{k}$. And indeed by the definition of S and the assumption that $0 \notin S_i$ we have: $\{kn \mid n \in S_i\} = \{m+i \mid m \in S,\ m+i \equiv 0 \pmod{k}\}$.)

It is easy to see that Q is satisfiable iff $\tilde{Q}$ is satisfiable. For if $\tilde{Q}$ is satisfied by a state u_0 in some PDL-structure we obtain a model of Q by restricting attention to the states satisfying P_0 and interpreting A by $\rho(A) = \rho(B)^k$. Conversely, if Q is satisfied by a state u_0 in some PDL structure we obtain a model of Q_1 by adding new states, replacing each edge $u \to^A v$ in the graph corresponding to the original structure by a chain $u \to^B \circ \to^B \cdots \to \circ \to^B v$ involving $k-1$ new states satisfying $P_1, \ldots, P_{k-1}$, respectively (all the old states shall satisfy P_0). This completes the proof that $\mathrm{stl}(S_1, \ldots, S_k) \leqslant_m \mathrm{stl}(S)$, hence $\mathrm{vld}(S_1, \ldots, S_k) \leqslant_m \mathrm{vld}(S)$. ∎

Proof of Lemma 6.3. Let $S_1, \ldots, S_k$ be Δ_1^1 subsets of w. If suffices to show that $\mathrm{stl}(S_1, \ldots, S_k)$ is a Σ_1^1 set. First note that every satisfiable formula is satisfiable in a countable structure, as is seen by a standard Lowenheim–Skolem argument. Thus, $Q \in \mathrm{stl}(S_1, \ldots, S_k)$ iff there exists a set $X \subseteq \omega$ (the set of states) and certain subsets of X (the interpretations of the atomic propositions) and a relation over S (the interpretation of the program A) which together constitute a PDL structure in which Q is satisfied. It is a routine exercise to write this as a Σ_1^1 predicate about Q (see Rogers [14] for the requisite background on the analytical hierarchy). ∎

Proof of Theorem 6.4. Let $E \subseteq \omega$ be any complete Σ_1^1 set (so that E is Σ_1^1 and if D is any Σ_1^1 set, then $D \leqslant_m E$). We will construct a primitive recursive set $S \subseteq \omega$ such that $E \leqslant_m \mathrm{stl}(S)$. Then $\mathrm{stl}(S)$ will be a complete Σ_1^1 set (it is Σ_1^1 by Lemma 6.3) and hence $\mathrm{vld}(S)$ will be a complete Π_1^1 set.

We shall make use of the following normal form for Σ_1^1 sets: If $E \subseteq \omega$ is Σ_1^1, then there exists a primitive recursive relation $R \subseteq \omega^3$ such that for all $m \in \omega$

$$\text{(A)} \quad m \in E \Leftrightarrow \exists X_1 (\forall x, y \in X_1)[x < y \Rightarrow R(m, x, y)].$$

Here (and throughout this proof) the variables $X_1, X_2, \ldots,$ range over *infinite* subsets of ω only. The existence of this normal form is proved in the Appendix.

Now let $S_1 = \{2^n \mid n \in \omega\}$ and let $S_2 = \{2^y - 2^x - 2^m \mid m < x < y \text{ and } R(m, x, y)\}$, where R is chosen to correspond to the particular set E with which we start. Then we have

$$\text{(B)} \quad m \in E \Leftrightarrow \exists X_2[(\forall n \in X_2)(n \in S_1 \wedge n > 2^m) \wedge (\forall n_1, n_2 \in X_2)(n_1 < n_2 \Rightarrow n_2 - n_1 - 2^m \in S_2)].$$

Indeed, if $m \in E$ and X_1 is a set as in the rhs of (A) let $X_2 = \{2^x \mid x \in X_1 \wedge x > m\}$. Then X_2 will clearly satisfy the rhs of (B). Conversely, if X_2 satisfies the rhs of (B) let $X_1 = \{\log_2 n \mid n \in X_2\}$. Then X_1 is infinite and if $x, y \in X_1$, $x < y$, then $m < x < y$ and 2^x, $2^y \in X_2$ and so $2^y - 2^x - 2^m \in S_2$. But the triple (m, x, y) is *uniquely* determined by the number $2^y - 2^x - 2^m$, given that $m < x < y$. Hence, if $2^y - 2^x - 2^m \in S_2$, then $R(m, x, y)$ holds (by definition of S_2). Thus X_1 satisfies the rhs of (A) and it follows that $m \in E$. This proves (B).

We can now effectively associate with every $m \in \omega$ a formula Q_m of

$\text{PDL}_{\text{RG}(A)\cup\{A^{\bar{S}_1}, A^{\bar{S}_2}\}}$ so that $m \in E$ iff Q_m is satisfiable. Roughly speaking Q_m describes the rhs of (B) with the atomic proposition P being true at those states whose "distance from the origin" is a member of X_2. Q_m is the conjunction of the formulas

(1) $[A^*]\langle A; A^*\rangle P$ ("X_2 is infinite"),

(2) $[A^{\bar{S}_1}] \sim P$ ("$X_2 \subseteq S_1$"),

(3) $\bigwedge_{i=0}^{2m} [A^i] \sim P$ ("$(\forall n \in X_2)\, n > 2^m$"), and

(4) $[A^*; P?; A^{2m}; A^{\bar{S}_2}] \sim P$ ("if $n_1 \in X_2$ and $z \in \bar{S}_2$, then $n_1 + 2^m + z \notin X_2$").

Note that if $m \in E$, then Q_m is satisfied at the root (0) of the "linear" model

$$\underset{0}{\circ} \xrightarrow{A} \underset{1}{\circ} \xrightarrow{A} \underset{2}{\circ} \xrightarrow{A} \underset{3}{\circ} \longrightarrow \cdots,$$

where P is declared true at n iff $n \in X_2$, given a set X_2 as on the rhs of (B). Conversely, given any PDL-structure in which some state u_0 satisfies Q_m the conjunct (1) of Q_m guarantees the existence of a sequence of (not necessarily distinct) states

$$\underset{u_0}{\circ} \xrightarrow{A} \underset{u_1}{\circ} \xrightarrow{A} \underset{u_2}{\circ} \xrightarrow{A} \underset{u_3}{\circ} \longrightarrow \cdots$$

starting from u_0 on which P is true infinitely many times. Letting $X_2 = \{n \mid u_n \models P\}$, conjuncts (2)–(4) imply that X_2 satisfies the rhs of (B), hence $m \in E$.

We have thus established that $E \leqslant_m \text{stl}(\bar{S}_2, \bar{S}_2)$, hence by Lemma 6.2 $E \leqslant_m \text{stl}(S)$, where $S = \{2n-1 \mid 1 \leqslant n \in \bar{S}_1\} \cup \{2n-2 \mid 1 \leqslant n \in \bar{S}_2\}$. A look at the definitions of S_1 and S_2 shows that they are primitive recursive (note that if $n = 2^y - 2^x - 2^m \in S_2$, then $n > \frac{1}{2}2^y$, hence $y < \log_2 2n$ so a bound on m, x, y in terms of n is available) hence so is S. This completes the proof of Theorem 6.4. ∎

7. Open Questions

The overall open direction for research is the classification of nonregular programs in terms of their effect on the validity of PDL. This paper contains negative results only. In [11] a sketch is presented of a proof that $\text{PDL}_{\text{RG}+\{A^\Delta B^\Delta\}}$ is decidable. If correct, this (and its variants) is the only known positive result.

The proof in [11] uses "pushdown models," making heavy use of the fact that $A^\Delta B^\Delta$ and languages obtained from it and elements of RG are context-free. However, we cannot even rule out the possibility that certain noncontext-free languages do not destroy the decidability of PDL. For example, is $\text{PDL}_{\text{RG}+\{A^\Delta B^\Delta C^\Delta\}}$ decidable? None of the methods for showing undecidability introduced in the present paper seem to work, as there are no atomic programs "playing two roles" as in $A^\Delta(B)A^\Delta$ or in $A^\Delta B^\Delta$ combined with $B^\Delta A^\Delta$.

As far as one-letter programs are concerned we have no positive results. Is there some recursive but nonregular $L \subseteq A^*$ such that $\text{PDL}_{\text{RG}+\{L\}}$ is decidable? Some particular languages such as $L=\{A^{n^2} \mid n \geqslant 0\}$ and $\{A^{n^3} \mid n \geqslant 0\}$ are particularly intriguing. We conjecture that their addition ruins the decidability of PDL, but do not have a proof.

APPENDIX: NORMAL FORMS FOR Σ_1^1 SETS

We prove Claims 1 and 2, which have been used in the paper.

CLAIM 1. If E is a Σ_1^1 subset of ω, then there exists a primitive recursive relation $R_1 \subseteq \omega^3$ such that for all $m \in \omega$: $m \in E$ iff $\exists f[f(0)=1 \wedge \forall x\, R_1(m, f(x), f(x+1))]$.

CLAIM 2. If E is a Σ_1^1 subset of ω, then there exists a primitive recursive relation $R_2 \subseteq \omega^3$ such that for all $m \in \omega$: $m \in E$ iff $\exists X_1(\forall x, y \in X_1)[x<y \Rightarrow R_2(m, x, y)]$.

In Claim 1 "f" ranges over functions from ω into ω and Claim 2 "X_1" ranges over *infinite* subsets of ω. It should be clear that the converses of the two claims are also true (even if R_1, R_2 are merely assumed to be Δ_1^1 rather than primitive recursive) so that we actually have here general normal forms for Σ_1^1 sets. We assume elementary knowledge of the analytical hierarchy (Rogers [14, Sect. 16.1] should suffice for this appendix).

To prove both claims we start from the following well-known normal form of a Σ_1^1-set E (cf. [14, Sect. 16.1, Corollary V]):

(1) $m \in E \Leftrightarrow \exists f_1\, \forall x R(m, \bar{f}_1(x))$.

Here R is a primitive-recursive relation (depending on E) and $\bar{f}_1$ is the "history function" of f_1, i.e., for each x, $\bar{f}_1(x)$ is a number coding the finite sequence $(f_1(0), \ldots, f_1(x-1))$. To be definite we choose the following method of coding finite sequences of numbers by numbers, which differs from that of [14]):

$$(x_1, \ldots, x_n) \mapsto \langle x_1, \ldots, x_n \rangle = 2^n p_1^{x_1} \cdots p_n^{x_n},$$

where $(3=) p_1 < p_2 < \cdots$ are the primes >2 in increasing order. In particular, the empty sequence is coded by $\langle\ \rangle = 2^0 = 1$. Let seq$(x)$ mean that x codes some finite sequence and let $x \lessdot y$ mean that seq(x) and seq(y) and the sequence coded by x is a proper initial segment of the one coded by y. Finally let lh(x) be the length of the sequence coded by x if seq(x), lh$(x)=0$, otherwise. Note that seq and $\lessdot$ are prim-rec relations and lh is a prim-rec function. Also note that $x \lessdot y \Rightarrow x < y$.

Proof of Claim 1. Given a Σ_1^1 set E choose a prim-rec $R \subseteq \omega^2$ so that (1) holds for all $m \in \omega$. It clearly follows from (1) that

$$m \in E \Leftrightarrow \exists f[f=\bar{f}_1 \text{ for some } f_1 \text{ and } \forall x R(m, f(x))].$$

But in order for f to be the "history function" of some f_1 it is necessary and sufficient that $\forall x[\text{seq}(f(x) \wedge \text{lh}(f(x))=x]$ and moreover $f(x) \lessdot f(x+1)$ for each x.

Equivalently, the condition is that $f(0)=\langle\ \rangle=1$ and $\forall x[f(x) \lesssim f(x+1) \wedge (\text{lh}(f(x+1))=\text{lh}(f(x))+1)]$. Define $R_1 \subseteq \omega^3$ by $R_1(m,u,v) \Leftrightarrow R(m,u) \wedge u \lesssim v \wedge \text{lh}(v)=\text{lh}(u)+1$. Then R_1 is prim-rec and $m \in E \Leftrightarrow \exists f[f(0)=1 \wedge \forall x R_1(m, f(x), f(x+1))]$, as required. ∎

Proof of Claim 2. Start again from the normal form (1) of E. Define $R_2 \subseteq \omega^3$ by $R_2(m,u,v) \Leftrightarrow u \lesssim v \wedge \forall z(z \lesssim v \Rightarrow R(m,z))$. Thus $R_2(m,u,v)$ says that v codes some sequence $v=\langle v_1,..., v_l\rangle$, u is of the form $\langle v_1,..., v_k\rangle$ for some $k<l$, and $R(m, \langle v_1,..., v_i\rangle)$ holds for every $i<l$. Note that R_2 is prim-rec. We claim that

$$(2) \quad m \in E \Leftrightarrow \exists X_1(\forall x, y \in X_1)[x<y \Rightarrow R_2(m,x,y)].$$

Suppose that $m \in E$ and let f_1 be as on the rhs of (1). Let $X_1=\{\bar{f}_1(n) \mid n \in \omega\}$. Then X_1 is infinite. If $x, y \in X_1$ and $x<y$, then $x=\langle f_1(0),...,f_1(k-1)\rangle$, $y=\langle f_1(0),...,f_1(l-1)\rangle$, where $k<l$ and $R_2(m,x,y)$ clearly holds.

Conversely, suppose that X_1 is an infinite set satisfying the rhs of (2). Then for all $x, y \in X_1$, $x<y \Rightarrow x \lesssim y$, hence there exists a unique function $f_1: \omega \to \omega$ such that $X_1 \subseteq \{\bar{f}_1(n) \mid n \in \omega\}$. For any $k \in \omega$ we can find $n_2>n_1>k$ such that $\bar{f}_1(n_1) \in X_1$ and $\bar{f}_1(n_2) \in X_1$, so that $R_2(m, \bar{f}_1(n_1), \bar{f}_1(n_2))$ holds, so the number $z=\bar{f}_1(k)$ satisfies $z \lesssim \bar{f}_1(n_2)$ and hence $R(m,z)$ (by the definition of R_2). Thus $\forall k\, R(m, \bar{f}_1(k))$ so that f_1 satisfies the rhs of (1), whence $m \in E$.

This proves (2) and thereby proves Claim 2. ∎

Acknowledgments

We are grateful to A. Yehudai for his help concerning the formal-languages part of the paper; namely, Proposition 2.3 and [5, 8, 15]. Y. Feldman and J. Halpern pointed out errors in a previous version.

References

1. J. W. deBakker and L. G. L. T. Meertens, On the completeness of the inductive assertion method, *J. Comput. Sys. Sci.* **11** (1975), 323–357.
2. Y. Bar-Hillel, M. Perles, and E. Shamir, On formal properties of simple phrase structure grammars, *Z. Phonetik, Sprach. Kommunikation.* **14** (1961), 143–172.
3. M. Davis, Y. Matijasevic, and J. Robinson, Hilbert's tenth problem. Diophantine equations: Positive aspects of a negative solution, *in* "Proc. Symp. Pure Math.," Springer-Verlag Lecture Notes in Math., No. 28, pp. 323–378, 1976.
4. M. J. Fischer and R. E. Ladner, Propositional dynamic logic of regular programs, *J. Comput. Sys. Sci.* **18** (1979), 194–211.
5. S. Greibach and E. Friedman, Super deterministic PDA's: A subcase with a decidable equivalence problem, *J. Assoc. Comput. Mach.* **27** (1980), 675–700.
6. D. Harel, A. Pnueli, and J. Stavi, Further results on propositional dynamic logic of nonregular programs, *in* "Proc. Workshop on Logics of Programs" (D. Kozen, Ed.), Springer-Verlag Lecture Notes in Computer Science, Berlin/New York No. 131, 1981.
7. M. Harrison, "Introduction to Formal Language Theory," Addison–Wesley, Reading, Mass., 1978.

8. M. LINNA, Two decidability results for deterministic pushdown automata, *J. Comput. Sys. Sci.* **18** (1979), 92–107.
9. Y. MATIJASEVIC, Enumerable sets are diophantine, *Soviet Math. Dokl.* **11** (1970), 354–357.
10. A. R. MEYER, R. S. STREETT, AND G. MIRKOWSKA, The deducibility problem in propositional dynamic logic, *in* "Proc. 8th Int. Colloq. on Autom. Lang. Prog.," Springer-Verlag Lecture Notes in Computer Science, Berlin/New York, 1981.
11. T. OLSHANSKY AND A. PNUELI, "There Exist Decidable Context-Free Propositional Dynamic Logics," manuscript.
12. V. R. PRATT, Semantical considerations on Floyd–Hoare logic, *in* "Proc. 17th IEEE Symp. on Foundations of Computer Science," pp. 109–112, 1976.
13. V. R. PRATT, A near optimal method for reasoning about action, *J. Comput. Sys. Sci.* **20** (1980), 231–254.
14. H. ROGERS, JR., "Theory of Recursive Functions and Effective Computability," McGraw–Hill, New York, 1967.
15. A. YEHUDAI, The decidability of equivalence for a family of linear grammars, *Inform. and Control* **47** (2) (1981), 122–136.

Printed by the St. Catherine Press Ltd., Tempelhof 41, Bruges, Belgium