INFORMATION AND CONTROL **64**, 119–135 (1985)

Propositional Dynamic Logic of Flowcharts*

D. HAREL AND R. SHERMAN

Department of Applied Mathematics, The Weizmann Institute of Science, Rehovot 76100, Israel

Following a suggestion of Pratt, we consider propositional dynamic logic in which programs are nondeterministic finite automata over atomic programs and tests (i.e., flowcharts), rather than regular expressions. While the resulting version of PDL, call it APDL, is clearly equivalent in expressive power to PDL, it is also (in the worst case) exponentially more succinct. In particular, deciding its validity problem by reducing it to that of PDL leads to a double exponential time procedure, although PDL itself is decidable in exponential time. We present an elementary combined proof of the completeness of a simple axiom system for APDL and decidability of the validity problem in exponential time. The results are thus stronger than those for PDL, since PDL can be encoded in APDL with no additional cost, and the proofs simpler, since induction on the structure of programs is virtually eliminated. Our axiom system for APDL relates to the PDL system just as Floyd's proof method for partial correctness relates to Hoare's.

1. INTRODUCTION

The propositional version of dynamic logic (Fischer and Ladner, 1979; Pratt, 1976) is used to reason about the before–after behavior of programs. In PDL programs are taken to be regular sets of execution sequences represented by regular expressions. An execution sequence is a finite word over an alphabet of atomic programs and tests. The choice of a particular representation for these regular sets clearly has no influence on the expressive power of the language. It is significant, however, in the sense that some representations might be more natural or economical than others. The regular expressions of PDL are natural and often give rise to proofs by induction on their structure. In particular, PDL is known to be decidable in exponential time and to admit a complete axiomatization consisting of a finite set of very natural axiom schemes including one for each of the regular operations on programs (see Fischer and Ladner, 1979; Pratt, 1979; Kozen and Parikh, 1981; Sherman and Harel, 1983; Harel, 1984).

[*] This paper is a revised and expanded version of a paper presented at the International Conference on "Foundations of Computation Theory" held in Borgholm, Sweden, August 21–27, 1983.

0019-9958/85 $3.00

Pratt (1981) raised the question of the behavior of a version of PDL in which programs are represented by flowcharts. A nondeterministic flowchart is simply a finite directed graph with a designated entry node and some exit nodes, whose edges are labelled with atomic programs and tests. Since such a flowchart can clearly be regarded as the transition diagram of a nondeterministic finite automaton, it is immediate that this new version of PDL, call it APDL, is equivalent in expressive power to the standard version. However, if validity in APDL is decided by translating automata into regular expressions and working in PDL, the translation can cost in the worst case an exponential in the size of the automaton (Ehrenfeucht and Zeiger, 1976). Hence formulas of APDL grow exponentially in length when transformed into PDL formulas, resulting in a double-exponential time decision procedure. Moreover, the axioms of PDL are unfit for APDL unless such a translation is carried out as a preliminary step of each proof.

Pratt (1981) sketched a tableau-like algorithm for deciding APDL in single exponential time, and also indicated, using an algebraic approach, how an axiom system for APDL might be constructed, eliminating the need for translating into PDL. In independent work, Abrahamson (1980), with different motivation, gave a $2^{O(n)2^{O(m)}}$ time decision procedure for PDL with m boolean variables. APDL can indeed be expressed succinctly in this PDL, implying a $2^{O(n \log(n))}$ time decision procedure, but, although better than two exponentials, this upper bound is still super-exponential.

In this paper we borrow the motivation and some basic ideas of Pratt (1981) and provide an elementary combined proof of the two fundamental properties of APDL: exponential-time decidability of the validity problem, and completeness of a simple finitary axiom system. The axiom system is in a sense simpler than that of PDL as it deals globally with the automata rather than with each of the regular operators. The axioms are similar to those given by Wolper (1983) for his extended temporal logic. Also, the combined proof itself is a simplification of the similar proof we have given for PDL (Sherman and Harel, 1983), as it replaces the three clauses for regular operators in all inductions on the structure of programs by a single clause for an automaton.

Since regular expressions can be translated easily into automata, with no essential growth in size, APDL is a more fundamental formalism than PDL, and the results are thus stronger than those for PDL.

The reader will observe that since APDL relates to PDL as flowcharts do to structured programs, the axiom system for APDL (and our proof of its completeness) relates to that of PDL (and the proof of its completeness) just as Floyd's (1967) inductive assertion method for partial correctness relates to Hoare's (1969) axiomatic system. This point is also hinted at in Pratt (1981).

We have used the automata approach presented herein to obtain results for some extensions of APDL (and hence of PDL), which are discussed briefly in Section 4 and which will appear separately. In particular it has been used by the second-listed author and A. Pnueli to prove exponential time decidability for PDL with *loop*, previously known to be decidable only in triple-exponential time (Streett, 1982).

Section 2 of the paper contains preliminaries and Section 3 contains the main results.

2. Syntax and Semantics

Definition. A *finite* (*nondeterministic*) *automaton* over an aphabet Σ is a 4-tuple $\mathscr{F} = \langle Q, q_0, \eta, F \rangle$, where:

Q is a finite set of states.

$q_0 \in Q$ is the initial state.

$\eta: Q \times \Sigma \to 2^Q$ is a transition function assigning a set of states to each state and letter from the alphabet.

$F \subseteq Q$ is a set of accepting states.

A word $\sigma \in \Sigma^*$, $\sigma = (\sigma_0 \cdots \sigma_{l-1})$, is *accepted* by $\mathscr{F}$ if there exists a sequence of states $(q_0,..., q_l)$ such that $q_l \in F$ and for every i, $0 \leqslant i < l$ $q_{i+1} \in \eta(q_i, \sigma_i)$.

Every finite automaton over the alphabet Σ can be represented as a union of (possibly nondeterministic) automata of the form (n, i, j, δ) where:

$\bar{n} = \{1, 2,..., n\}$ is the set of states.

$i \in \bar{n}$ is the initial state.

$j \in \bar{n}$ is the final state.

$\delta: \bar{n} \times \bar{n} \to \Sigma$ is a partial labeling (transition) function.

A word $\sigma \in \Sigma^*$, $\sigma = (\sigma_0 \cdots \sigma_{l-1})$, is *accepted* by (n, i, j, δ) if there exists a sequence of states $(i_0,..., i_l)$, $i_0 = i$, $i_l = j$, $i_k \in \bar{n}$, and $\sigma_k = \delta(i_k, i_{k+1})$, $0 \leqslant k < l$.

Note that a nondeterministic finite automaton with m states over a finite alphabet Σ, can be represented by a union of at most m automata of the above form each with $n \leqslant m \cdot |\Sigma|$ states.

APDL is defined over two sets of symbols: Φ_0, the set of *atomic formulas*, and Π_0, the set of *atomic programs*. Φ_0 and Π_0 are, respectively, abstractions of properties of states, and basic instructions such as assignment statements, which transform one state into another. From these basic alphabets we inductively construct the set Φ of expressions for compound formulas, representing assertions about states, and the set Π of programs representing transformations on states by finite automata.

The following clauses define Φ:

$true \in \Phi$; $false \in \Phi$; $\Phi_0 \subseteq \Phi$,

if $p \in \Phi$ and $q \in \Phi$ then $\neg p \in \Phi$ and $(p \vee q) \in \Phi$,

if $p \in \Phi$ and $\alpha \in \Pi$ then $\langle \alpha \rangle p \in \Phi$.

The following clauses define Π:

$\Pi_0 \subseteq \Pi$,

$\Phi? \subseteq \Pi$, where $\Phi? = \{p? \mid p \in \Phi\}$,

if $\alpha = (n, i, j, \delta)$ is an automaton over the alphabet $\Pi_0 \cup \Phi?$ then $\alpha \in \Pi$.

We use $\wedge$, $\equiv$, $\supset$ as abbreviations in the standard way and, in addition, abbreviate $\neg \langle \alpha \rangle \neg p$ as $[\alpha] p$.

The semantics of APDL is defined relative to a given *structure* (or *model*) $\mathscr{A} = (W, \tau, \rho)$ where:

W is a set of elements called *states*, (not to be confused with the sets $\bar{n}$ of automata states),

$\tau: \Phi_0 \rightarrow 2^W$,

$\rho: \Pi_0 \rightarrow 2^{W \times W}$.

Informally, the mapping τ assigns to each atomic formula P the set $\tau(P) \subseteq W$ of states in which it is true and ρ assigns to each atomic program a a binary relation with the intended meaning $(s, t) \in \rho(a)$ iff execution of a can lead from state s to state t. Such an $\mathscr{A}$ is called a structure *over* Φ_0 and Π_0. The mappings τ and ρ are extended to supply meanings for the full sets Φ and Π as follows:

$$\tau(true) = W; \qquad \tau(false) = \varnothing,$$

$$\tau(\neg p) = W - \tau(p),$$

$$\tau(p \vee q) = \tau(p) \cup \tau(q),$$

$$\tau(\langle \alpha \rangle p) = \{s \in W \mid \exists t \in W((s, t) \in \rho(\alpha) \wedge t \in \tau(p))\},$$

$$\rho(p?) = \{(s, s) \mid s \in \tau(p)\},$$

$$\begin{aligned}\rho(n, i, j, \delta) = \{(s, t) \mid \exists k((\exists(i_0, ..., i_k), i_0 = i, i_k = j, \forall l\, i_l \in \bar{n}) \\ \wedge (\exists(s_0, ..., s_k), s_0 = s, s_k = t, \forall l\, s_l \in W) \\ \text{s.t. } (s_l, s_{l+1}) \in \rho(\delta(i_l, i_{l+1}))\ \forall l, 0 \leqslant l < k)\}.\end{aligned}$$

Actually what the last definition says is that $\rho(n, i, j, \delta)$ is the set of transitions in the model corresponding to transitions from state i to state j in the automaton.

We shall write $\mathscr{A}$, $s \models p$ and say that p *is true in* s or that s *satisfies* p if $s \in \tau(p)$, and omit $\mathscr{A}$ when it is clear from the context. We say that p is *valid* if $\mathscr{A}$, $s \models p$ for every structure $\mathscr{A}$ and state s and write $\models p$, and that p is *satisfiable* if there exist $\mathscr{A}$ and s such that $\mathscr{A}$, $s \models p$. Clearly p is valid iff $\neg p$ is not satisfiable.

DEFINITION. The *sizes* of a formula p, and a program α, denoted $|p|$ and $|\alpha|$ respectively, are defined as follows:

$$|a| = |P| = 1 \qquad \text{for } P \in \Phi_0,\ a \in \Pi_0$$

$$|\neg q| = |q| + 1$$

$$|q \vee r| = |q| + |r| + 1$$

$$|\langle \alpha \rangle q| = |\alpha| + |q| + 1$$

$$|q?| = |q| + 1$$

$$|(n, i, j, \delta) = n + \Sigma_{(k,l) \in V} |\delta(k, l)|$$

$$\text{where } V = \{(k, l) \mid k, l \in \bar{n} \text{ and } \delta(k, l) \text{ is defined}\}.$$

It is easy to show that APDL with its special kind of automata is only (in the worst case) quadratically less succinct than a version employing standard nondeterministic automata, by the remark following the definition of automata above. Thus, for our purposes no generality is lost in considering APDL. Also, while we impose a "pre-processing" of sorts, transforming a general automaton into a union of those of the type we use, this transformation can be added, if so desired, to the axiom system of Section 3. Our automata are more elementary, and hence the axiom system of Section 3 is not burdened with automata-theoretic details.

3. DECIDABILITY AND COMPLETENESS

The completeness of a simple axiom system for APDL is established, and from the proof it is concluded that the validity of formulas of APDL is decidable deterministically in time which is on the order of an exponential in the size of the input formula. Specifically, validity of p can be tested in deterministic time $2^{c \cdot |p|}$ for some $c > 0$.

The following definition captures a certain notion of the subformulas of a formula, and is analogous to the Fischer/Ladner closure of (Fischer and Ladner, 1979; Kozen and Parikh, 1981).

DEFINITION. Let p be a formula of APDL; i.e., $p \in \Phi$. The *closure of* p, denoted $\mathrm{CL}(p)$, is the smallest set S of formulas containing p and satisfying the following closure rules for all $a \in \Pi_0$, $(n, i, j, \delta) \in \Pi$, and q, $r \in \Phi$.

$$\neg q \in S \Rightarrow q \in S$$

$$q \vee r \in S \Rightarrow q \in S, r \in S$$

$$\langle a \rangle\, q \in S \Rightarrow q \in S$$

$$\langle q? \rangle\, r \in S \Rightarrow q \in S, r \in S$$

$$\langle n, i, j, \delta \rangle\, q \in S \Rightarrow \text{for every } k \in \bar{n} \text{ such that } \delta(i, k) \text{ is defined,}$$

$$\langle \delta(i, k) \rangle \langle n, k, j, \delta \rangle\, q \in S,$$

$$\text{and in addition if } i = j \text{ then } q \in S.$$

It is easy to see that $|\mathrm{CL}(p)|$ (i.e., the number of formulas in $\mathrm{CL}(p)$) is linear in the length of p; i.e., $|\mathrm{CL}(p)| = O(|p|)$.

Let $\neg\mathrm{CL}(p)$ be defined as $\{\neg q \mid q \in \mathrm{CL}(p)\}$. Denote $\mathrm{CL}(p) \cup \neg\mathrm{CL}(p)$ by Z. We now define certain sets of formulas from Z called *atoms*, which are free of "immediate" inconsistencies. Later we eliminate those which are inconsistent with all others.

Note. In the rest of the section we identify a formula of the form $\neg\neg q$ with q.

DEFINITION. An *atom* for p (or just *atom* when p is assumed) is a subset A of Z satisfying the following, for every $(n, i, j, \delta) \in \Pi$, q, $r \in \Phi$:

if $q \in Z$ then $q \in A \Leftrightarrow \neg q \notin A$

if $q \vee r \in Z$ then $q \vee r \in A \Leftrightarrow q \in A$ or $r \in A$

if $\langle q? \rangle\, r \in Z$ then $\langle q? \rangle\, r \in A \Leftrightarrow q \in A$ and $r \in A$

if $\langle n, i, j, \delta \rangle\, q \in Z$ then $\langle n, i, j, \delta \rangle\, q \in A \Leftrightarrow$ either $i = j$ and $q \in A$ or $\langle \delta(i, k) \rangle \langle n, k, j, \delta \rangle\, q \in A$ for some $k \in \bar{n}$.

Denote the set of atoms for p by $At(p)$; clearly $|At(p)| \leqslant 2^{O(|p|)}$.

Let there be given a fixed formula $p \in \Phi$. Since we will be interested only in formulas connected directly with some such given p, we assume, without loss of generality, that Φ_0 and Π_0 consist solely of the atomic formulas and programs appearing in p. A particular finite structure $\mathscr{A} = (W, \tau, \rho)$ is constructed in steps as follows:

$\mathscr{A}_0 = (W_0, \tau_0, \rho_0)$ is defined by

$W_0 = At(p)$,

$\tau_0 \colon \Phi_0 \rightarrow 2^{W_0}$ where, for each $P \in \Phi_0$, $A \in \tau_0(P)$ iff $P \in A$,

$\rho_0 \colon \Pi_0 \rightarrow 2^{W_0 \times W_0}$ where, for each $a \in \Pi_0$, $(A, B) \in \rho_0(a)$ iff for every $[a]\, q \in A$ we have $q \in B$.

We extend ρ_0 to Π in the usual way. In the following we use a special extension of ρ_0 to $\Pi_0 \cup Z? \cup \{(n, i, j, \delta)\}$ (where (n, i, j, δ) is an automaton over $\Pi_0 \cup Z?$). This extension, denoted ρ_0', is defined in the usual way except for the definition for test programs:

$$\rho_0'(q?) = \{(A, A) \mid A \in W_0, q \in A\} \qquad \text{for } q \in Z$$

(rather than $\{(A, A) \mid A \in W_0, \mathscr{A}_0, A \models q\}$)

For $i \geqslant 0$ let $\mathscr{A}_{i+1} = (W_{i+1}, \tau_{i+1}, \rho_{i+1})$ be given by

$$\begin{aligned} W_{i+1} = \{A \mid\ & A \in W_i \text{ and for every } \langle\alpha\rangle\, q \in A, \text{ where } \alpha \in \Pi, \\ & q \in \Phi, \text{ there is } B \in W_i \text{ such that } (A, B) \in \rho_i'(\alpha) \text{ and} \\ & q \in B\} \\ \tau_{i+1}(P) = \tau_i(P) \cap W_{i+1} & \qquad \text{for } P \in \Phi_0 \\ \rho_{i+1}(a) = \rho_i(a) \cap (W_{i+1} \times W_{i+1}) & \qquad \text{for } a \in \Pi_0, \end{aligned}$$

and ρ_{i+1}' denotes the special extension of ρ_{i+1} defined similarly to ρ_0'.

Clearly, from the finiteness of $At(p)$ there is some i_0 where the construction closes up; i.e., for every $j > i_0$, $\mathscr{A}_j = \mathscr{A}_{i_0}$. Accordingly we set

$$\mathscr{A} = (W, \tau, \rho) = (W_{i_0}, \tau_{i_0}, \rho_{i_0}) = \mathscr{A}_{i_0}.$$

The transition from W_i to W_{i+1} is meant to bring the model one step closer to a final one by deleting states which do not keep "$\langle\alpha\rangle$-promises" for any α. Clearly, the deletion of some states in one such stage can cause new "promises" to be violated, necessitating additioned stages. (The ρ_i' vs. ρ_i part is a technicality needed for dealing with tests that involve programs.)

Remark. Since $|W_0| \leqslant 2^{0(|p|)}$, and the computation of $\mathscr{A}_{i+1}$ is clearly polynomial in the size of W_i, it follows that the structure $\mathscr{A}$ can be computed in time exponential in the length of p.

The following lemma connects the two roles played by an atom in W: that of a set of subformulas of p and that of a state in $\mathscr{A}$.

LEMMA 1. *For every $A \in W$ and $q \in \mathrm{CL}(p)$,*

$$q \in A \qquad \textit{iff} \quad \mathscr{A}, A \models q.$$

Proof. The claim is proved by induction on the structure of q:

$q = Q \in \Phi_0$: $Q \in A \Leftrightarrow A \in \tau_0(Q) \Leftrightarrow A \in \tau_0(Q) \cap W \Leftrightarrow A \in \tau(Q) \Leftrightarrow A \models Q$.

$q = \neg r$: $\neg r \in A \Leftrightarrow r \notin A \Leftrightarrow$ (ind. hyp.) $A \not\models r \Leftrightarrow A \models \neg r$.

$q = r \vee s$: $r \vee s \in A \Leftrightarrow r \in A \vee s \in A \Leftrightarrow$ (ind. hyp.) $A \models r \vee A \models s \Leftrightarrow A \models r \vee s$.

$q = \langle \alpha \rangle r$.

To prove this we prove the following claim:

For every $A \in W$ and $\langle \beta \rangle s \in \mathrm{CL}(p)$, $\langle \beta \rangle s \in A$ iff there is $B \in W$ such that $(A, B) \in \rho(\beta)$ and $s \in B$.

Suppose $\langle \beta \rangle s \in A$, then by the construction of $\mathscr{A}$, $\exists B \in W$, $(A, B) \in \rho'(\beta)$ and $s \in B$. We show that $\exists B \in W$, $(A, B) \in \rho(\beta)$, and $s \in B$, for the possible forms of β:

$\beta = b \in \Pi_0$: by the definition of ρ, $(A, B) \in \rho'(b) \Rightarrow (A, B) \in \rho(b)$.

$\beta = u? \in Z?$: by the definition of atoms $\langle u? \rangle s \in A \Rightarrow u \in A \wedge s \in A \Rightarrow$ (ind. hyp.) $A \models u$ and $s \in A \Rightarrow (A, A) \in \rho(u?)$ and $s \in A$.

$\beta = (n, i, j, \delta)$: $(A, B) \in \rho'(n, i, j, \delta)$ and $s \in B \Rightarrow \exists \sigma \in L(n, i, j, \delta)$, $\sigma = (\sigma_0 \cdots \sigma_{k-1})$, $\sigma_l \in \Pi_0 \cup Z?$ for $0 \leqslant l < k$ and $\exists (A_0, ..., A_k)$, $A_0 = A$, $A_k = B$, $(A_l, A_{l+1}) \in \rho'(\sigma_l)$ for $0 \leqslant l < k$. By the first two cases for β, $(A_l, A_{l+1}) \in \rho'(\sigma_l)$ implies $(A_l, A_{l+1}) \in \rho(\sigma_l)$ for $0 \leqslant l < k$ hence $(A, B) \in \rho(\beta)$ and $s \in B$.

For the "if" part we proceed as follows:

$\beta = b \in \Pi_0$: Assume $\langle b \rangle s \notin A$. By the definition of an atom $\neg \langle b \rangle s \in A$, i.e., $[b] \neg s \in A$. Now if $(A, B) \in \rho(b)$ then by the definition of ρ we certainly have $(A, B) \in \rho_0(b)$, from which, by the definition of ρ_0 and the fact that $[b] \neg s \in A$ we obtain $\neg s \in B$, or $s \notin B$.

$\beta = u?$: $\exists B((A, B) \in \rho(u?) \wedge s \in B) \Rightarrow ((A, A) \in \rho(u?) \wedge s \in A) \Rightarrow (A \models u \wedge s \in A) \Rightarrow$ (main ind. hyp.) $(u \in A \wedge s \in A) \Rightarrow \langle u? \rangle s \in A$ by the definition of atoms.

$\beta = (n, i, j, \delta)$: $\exists B((A, B) \in \rho(n, i, j, \delta) \wedge s \in B) \Rightarrow \exists (i_0, ..., i_k)$, $i_0 = i$, $i_k = j$, $\exists (A_0, ..., A_k)$, $A_0 = A$, $A_k = B$, s.t. $(A_l, A_{l+1}) \in \rho(\delta(i_l, i_{l+1}))$ for every l, $0 \leqslant l < k$. We prove that $\langle n, i, j, \delta \rangle s \in A$ by induction on k.

For $k = 0$: $i = j$, $A = B$, then by the definition of atoms $s \in A$ implies $\langle n, i, i, \delta \rangle s \in A$.

Suppose the claim is true for k. Then for $k + 1$: $\langle n, i_0, i_{k+1}, \delta \rangle s \in \mathrm{CL}(p)$ implies that $\langle \delta(i_0, i_1) \rangle \langle n, i_1, i_{k+1}, \delta \rangle s \in \mathrm{CL}(p)$ and hence $\langle n, i_1, i_{k+1}, \delta \rangle s \in \mathrm{CL}(p)$. Since $(A_1, A_{k+1}) \in \rho(n, i_1, i_{k+1}, \delta)$, it follows from the induction hypothesis on k that $\langle n, i_1, i_{k+1}, \delta \rangle s \in A_1$. Now by $\delta(i_0, i_1) \in \Pi_0 \cup \Phi_A?$ and the first two cases for β we obtain $\langle \delta(i_0, i_1) \rangle \langle n, i_1, i_{k+1}, \delta \rangle s \in A_0$ and this implies by the definition of atoms that $\langle n, i_0, i_{k+1}, \delta \rangle s \in A$. This completes the proof of the claim.

Back to the main proof: clearly a straightforward argument shows that since $\langle \alpha \rangle r \in \mathrm{CL}(p)$ also $r \in \mathrm{CL}(p)$, and so the induction hypothesis for r

can be used, $\langle\alpha\rangle r \in A \Leftrightarrow$ (by the claim) $\exists B \in W((A, B) \in \rho(\alpha) \wedge r \in B) \Leftrightarrow$ (ind. hyp.) $\exists B \in W((A, B) \in \rho(\alpha) \wedge B \models r) \Leftrightarrow A \models \langle\alpha\rangle r$. ▌

We now introduce an axiomatic system for APDL.

Notation. for k, $l \in \bar{n}$ we write "$\delta(k, l)\downarrow$," for "$\delta(k, l)$ is defined". For $k \in \bar{n}$ we denote by $\Theta(k)$ the set $\{l \mid l \in \bar{n}, \delta(k, l)\downarrow\}$.

Our axioms (A4) and (A5) (Table I) are very similar to axioms (G1) and (G2), respectively, of Wolper (1983, p. 82). Axiom (A4) states that the possibility of starting at state i and reaching state j with p true is equivalent to that of starting at i and reaching some immediate successor k of i and then from k reaching j with p true. The induction axiom (A5) says that if one has chosen a set $\{p_l\}$ of assertions, and has shown that (i) p_i is true at state i, and (ii) the truth of p_k at some state k (reachable from i) implies the truth of p_l at any successor l of k, then he has in fact established that p_j is true when j is reached from i. Thus, axiom (A5) formalizes Floyd's inductive assertions method for proving partial correctness; the p_l are the inductive assertions.

It is easy to establish the following two derived rules:

Invariance (I):

$$\frac{\{p_k \supset [\delta(k, l)]\, p_l\}_{k,l \in \bar{n}, \delta(k,l)\downarrow}}{p_i \supset [n, i, j, \delta]\, p_j}$$

(apply (G) with $[n, i, k, \delta]$, then (MP) with (A5)).

TABLE I

Axiom schemes:

(A1) All instances of tautologies of the propositional calculus.

(A2) $\langle\alpha\rangle(p \vee q) \equiv \langle\alpha\rangle p \vee \langle\alpha\rangle q$

(A3) $\langle p?\rangle q \equiv p \wedge q$

(A4) $\langle n, i, j, \delta\rangle p \equiv \bigvee_{k \in \bar{n}, \delta(i,k)\downarrow} \langle\delta(i, k)\rangle\langle n, k, j, \delta\rangle p$, for $i \neq j$

(A4′) $\langle n, i, i, \delta\rangle p \equiv p \vee \bigvee_{k \in \bar{n}, \delta(i,k)\downarrow} \langle\delta(i, k)\rangle\langle n, k, i, \delta\rangle p$

(A5) (Induction axiom)

$$(\textstyle\bigwedge_{k,l \in \bar{n}, \delta(k,l)\downarrow} [n, i, k, \delta](p_k \supset [\delta(k, l)]\, p_l)) \supset (p_i \supset [n, i, j, \delta]\, p_j)$$

(A6) $[\alpha](p \supset q) \supset ([\alpha]\, p \supset [\alpha]\, q)$

Inference rules:

(R1) Modus ponens (MP) $(p, p \supset q)/q$

(R2) Generalization (G) $p/([\alpha]\, p)$

Distribution (D):

$$\frac{p \supset q}{[\alpha]\, p \supset [\alpha]\, q}$$

(apply (G) with $[\alpha]$, then (MP) with (A6)).

Provability of a formula p in the system is denoted $\vdash p$.

THEOREM 2. *The axiom system is sound; i.e., for every $p \in \Phi$, $\vdash p \Rightarrow \models p$.*

Proof. It is immediate from the definition of the semantics of APDL that all instances of axioms of the above system are valid and all rules of inference preserve validity. ▌

DEFINITION. For a finite set $A \subset \Phi$, let $\hat{A}$ denote $\bigwedge_{q \in A} q$.

The following lemma shows that non-atoms are provably inconsistent.

LEMMA 3. *Let $A \subseteq Z$, such that for $q \in Z$ either $q \in A$ or $\neg q \in A$. If $A \notin At(p)$ then $\vdash \neg \hat{A}$.*

Proof. If A does not satisfy the first property of an atom, namely $q \in A \Leftrightarrow \neg q \notin A$ then there will be some $q \in A$ with $\neg q \in A$. One then proves $\neg \hat{A}$ by (A1, MP). Assume, therefore, that $q \in A \Leftrightarrow \neg q \notin A$ for every $q \in Z$. For each of the three remaining properties of an atom it is straightforward to show how a violation causes a provable contradiction. We illustrate this with the $\langle n, i, j, \delta \rangle q$ property for $i \neq j$: Assume $\langle n, i,j, \delta \rangle q \in A$ but $\langle \delta(i, k) \rangle \langle n, k, j, \delta \rangle q \notin A$ for every k s.t. $\delta(i, k)\downarrow$. Hence by our assumption $\neg \langle \delta(i, k) \rangle \langle n, k, j, \delta \rangle q \in A$ for every k s.t. $\delta(i, k)\downarrow$, hence we have by (A1)$\vdash \hat{A} \supset (\bigwedge_{k \in \bar{n}} \neg \langle \delta(i, k) \rangle \langle n, k, j, \delta \rangle q)$; hence also $\vdash \hat{A} \supset \neg (\bigvee_{k \in \bar{n}} \langle \delta(i, k) \rangle \langle n, k, j, \delta \rangle q)$, and with (A4), $\vdash \hat{A} \supset \neg \langle n, i, j, \delta \rangle q$. But $\langle n, i, j, \delta \rangle q \in A$; so $\vdash \hat{A} \supset \langle n, i, j, \delta \rangle q$. Hence $\vdash \neg \hat{A}$. ▌

COROLLARY 4. *For every $q \in Z$, $E \subseteq At(p)$,*

$$\vdash \Bigl(q \equiv \bigvee_{\substack{A \in At(p) \\ q \in A}} \hat{A} \Bigr) \tag{1}$$

$$\vdash \Bigl(\bigvee_{A \in E} \hat{A} \equiv \bigwedge_{B \in At(p) - E} \neg \hat{B} \Bigr). \tag{2}$$

Proof. Let

$$V = \{ A \mid A \subseteq Z, \forall q \in Z (q \in A \vee \neg q \in A) \}.$$

Then clearly

$$\vdash\left(q\equiv \bigvee_{\substack{A\in V\\ q\in A}} \hat{A}\right).$$

But by Lemma 3, if $A\in V-At(p)$ then $\vdash \neg\hat{A}$. Thus

$$\vdash\left(q\equiv \bigvee_{\substack{A\in At(p)\\ q\in A}} \hat{A}\right).$$

Also $\vdash \bigvee_{A\in V}\hat{A}$ and for similar reasons actually, $\vdash \bigvee_{A\in At(p)}\hat{A}$. Now if A, $B\in V$, $A\neq B$, then clearly $\vdash\neg(\hat{A}\wedge\hat{B})$, from which we have

$$\vdash\left(\bigvee_{A\in E}\hat{A}\vee\bigvee_{B\in At(p)-E}\hat{B}\right)\wedge\neg\left(\bigwedge_{A\in E}\hat{A}\wedge\bigwedge_{B\in At(p)-E}\hat{B}\right).$$

From this it is immediate that

$$\vdash\left(\bigvee_{A\in E}\hat{A}\equiv\neg\bigvee_{B\in At(p)-E}\hat{B}\right). \quad \blacksquare$$

The following is the main technical lemma needed in the proof, which says that for every formula $\langle\alpha\rangle q\in \mathrm{CL}(p)$ and atom $A\in At(p)$, $\hat{A}$ implies that after every α execution $\hat{B}$ is true for some B such that $(A, B)\in\rho_0'(\alpha)$. It follows that if for every B, $(A, B)\in\rho_0'(\alpha)$ implies that $q\notin B$ then $\vdash\hat{A}\supset[\alpha]\neg q$. Hence for A with $\langle\alpha\rangle q\in A$ we conclude that $\neg\hat{A}$ is provable, which justifies the rejection of A from the set of states of the constructed model.

LEMMA 5. *Let $A\in At(p)$ and $\langle\alpha\rangle q\in \mathrm{CL}(p)$ then*

$$\vdash\hat{A}\supset[\alpha]\left(\bigvee_{(A,B)\in\rho_0'(\alpha)}\hat{B}\right).$$

Proof. We prove the claim for the three possible forms of α:

$\alpha=a\in\Pi_0$. Clearly by Corollary 4 (2) it suffices to show

$$\vdash\hat{A}\supset[a]\left(\bigwedge_{(A,B)\notin\rho_0'(a)}\neg\hat{B}\right)$$

or, using axiom (A2) and the finiteness of $At(p)$, that $\vdash\hat{A}\supset[a]\neg\hat{B}$ for every B such that $(A, B)\notin\rho_0'(a)$. For such a B, by the definition of ρ_0', it must be the case that there is some $[a]r\in A$ with $\neg r\in B$. Hence $\vdash\hat{A}\supset[a]r$ and $\vdash\hat{B}\supset\neg r$ or $\vdash r\supset\neg\hat{B}$. Using (D) we obtain $\vdash\hat{A}\supset[a]\neg\hat{B}$.

$\alpha = r? \in Z?$. Tautologically, $\vdash \hat{A} \supset \neg (r \wedge \neg \hat{A})$, thus by axiom (A3) $\vdash \hat{A} \supset \neg \langle r? \rangle \neg \hat{A}$, or $\vdash \hat{A} \supset [r?] \hat{A}$. Since $\langle r? \rangle q \in \mathrm{CL}(p)$ we have $r \in \mathrm{CL}(p)$. If $r \in A$ then by the definition of $\rho_0'(A, A) \in \rho_0'(r?)$ and hence A is a special case of the required disjunction. If $r \notin A$ then $\vdash \hat{A} \supset \neg r$ and hence

$$\vdash \hat{A} \supset [r?] \Bigg(\bigvee_{(A,B) \in \rho_0'(r?)} \hat{B} \Bigg).$$

Thus the claim follows for both cases.

$\alpha = (n, i, j, \delta)$. For each $k \in \bar{n}$ denote by p_k the formula:

$$\bigvee_{(A,B) \in \rho_0'(n,i,k,\delta)} \hat{B}.$$

We show first that for each $k \in \bar{n}$ and $l \in \Theta(k)$

$$\vdash p_k \supset [\delta(k, l)]\, p_l.$$

If $\langle n, i, j, \delta \rangle q \in \mathrm{CL}(p)$ and $k \in \bar{n}$ is accessible from i then by induction on the length of the run from i to k, it is easy to prove that $\langle n, k, j, \delta \rangle q \in \mathrm{CL}(p)$ and hence for every $l \in \Theta(k)$, $\langle \delta(k, l) \rangle \langle n, l, j, \delta \rangle q \in \mathrm{CL}(p)$. Hence for every atom B, by the first two cases of this lemma for $\delta(k, l) \in \Pi_0 \cup Z?$ we obtain

$$\vdash \hat{B} \supset [\delta(k, l)] \Bigg(\bigvee_{(B,C) \in \rho_0'(\delta(k,l)} \hat{C} \Bigg). \tag{3}$$

If $(A, B) \in \rho_0'(n, i, k, \delta)$ and $(B, C) \in \rho_0'(\delta(k, l))$ then $(A, C) \in \rho_0'(n, i, l, \delta)$ which together with (3) implies

$$\vdash \hat{B} \supset [\delta(k, l)]\, p_l. \tag{4}$$

Hence, since (4) holds for every $\hat{B}$ in the disjunct defining p_k, we obtain that for every $k \in \bar{n}$ and $l \in \Theta(k)$:

$$\vdash p_k \supset [\delta(k, l)] p_l.$$

By the invariance rule (I) this implies

$$\vdash p_i \supset [n, i, j, \delta] p_j. \tag{5}$$

As $(A, A) \in \rho_0'(n, i, i, \delta)$, $\hat{A}$ is a special case of the disjunct defining p_i, thus

$$\vdash \hat{A} \supset p_i. \tag{6}$$

By (5) and (6) we conclude

$$\vdash \hat{A} \supset [n, i, j, \delta] \left(\bigvee_{(A,B) \in \rho_0'(n,i,j,\delta)} \hat{B} \right). \quad \blacksquare$$

COROLLARY 6. *Let $A \in At(p)$ and $\langle \alpha \rangle q \in \mathrm{CL}(p)$ then*

$$\vdash \hat{A} \supset [\alpha] \left(\neg q \vee \bigvee_{\substack{q \in B \\ (A,B) \in \rho_0'(\alpha)}} \hat{B} \right).$$

Proof. We can rewrite the claim in Lemma 5 as

$$\vdash \hat{A} \supset [\alpha] \left(\bigvee_{\substack{\neg q \in C, \\ (A,C) \in \rho_0'(\alpha)}} \hat{C} \vee \bigvee_{\substack{q \in B \\ (A,B) \in \rho_0'(\alpha)}} \hat{B} \right).$$

For every C in the left disjunct we have $\vdash \hat{C} \supset \neg q$, hence the claim follows. ∎

We now show that not only non-atoms but even atoms are provably inconsistent, if they are rejected from being states in $\mathscr{A}$.

LEMMA 7. *For every $A \in At(p)$, if $A \notin W$ then $\vdash \neg \hat{A}$.*

Proof. The lemma is proved by induction on the order in which atoms are rejected from W. The proof uses the claims in Lemma 5 and Corollary 6 for ρ_i', $i \geqslant 0$. Thus we prove the following for every $i \geqslant 0$:

(1) if $A \notin W_i$ then $\vdash \neg \hat{A}$

(2) if $A \in W_i$ and $\langle \alpha \rangle q \in \mathrm{CL}(p)$ then

$$\vdash \hat{A} \supset [\alpha] \left(\bigvee_{(A,B) \in \rho_i'(\alpha)} \hat{B} \right).$$

For $i = 0$ clause (1) holds since $W_0 = At(p)$, and (2) holds by Lemma 5. Suppose clauses (1) and (2) hold for i. Let $A \notin W_{i+1}$, then there must be some $\langle \alpha \rangle q \in A$ such that for every $B \in W_i$, $(A, B) \in \rho_i'(\alpha)$ so that $q \notin B$. By the induction hypothesis, exactly as in Corollary 6, we have

$$\vdash \hat{A} \supset [\alpha] \left(\neg q \wedge \bigvee_{\substack{q \in B, B \in W_i \\ (A,B) \in \rho_i'(\alpha)}} \hat{B} \right).$$

But by the assumption the right disjunct is empty. We are left with $\vdash \hat{A} \supset [\alpha] \neg q$ or $\vdash \hat{A} \supset \neg \langle \alpha \rangle q$. However, since $\langle \alpha \rangle q \in A$, we have $\vdash \hat{A} \supset \langle \alpha \rangle q$ from which at once we obtain $\vdash \neg \hat{A}$.

To prove clause (2) for $i+1$, assume first that $\alpha \in \Pi_0 \cup Z?$, then by the induction hypothesis we can rewrite clause (2) as

$$\vdash \hat{A} \supset [\alpha] \Bigg(\bigvee_{(A,B) \in \rho'_{i+1}(\alpha)} \hat{B} \wedge \bigvee_{\substack{C \notin W_{i+1} \\ (A,C) \in \rho'_i(\alpha)}} \hat{C} \Bigg).$$

By clause (1) we have for each C in the right disjunct $\vdash \neg \hat{C}$, hence clause (2) holds for $\alpha \in \Pi_0 \cup Z?$. For $\alpha = (n, i, j, \delta)$ the claim now follows exactly as in the proof of Lemma 5. ▌

COROLLARY 8. *p is satisfiable iff $p \in A$ for some $A \in W$.*

Proof. One direction is obvious by Lemma 1. Let it now be the case that for every A such that $p \in A$, $A \notin W$. Then by Lemma 7 $\vdash \bigwedge_{p \in A} \neg \hat{A}$, or $\vdash \neg \bigvee_{p \in A} \hat{A}$, which by Corollary 4 (1) yields $\vdash \neg p$. Hence p cannot be satisfiable without violating Theorem 2, the soundness of the axiom system. ▌

THEOREM 9. *The axiom system is complete; i.e., for every $p \in \Phi$, $\models p \Rightarrow \vdash p$.*

Proof. If $\models p$ then $\neg p$ is not satisfiable, hence for each $A \in W$, $\neg p \notin A$. This means, together with Corollary 4 (1), that $\vdash \neg p \equiv \bigvee_{\neg p \in A, A \notin W} \hat{A}$. But by Lemma 7, $\vdash \neg \bigvee_{A \notin W} \hat{A}$. Hence $\vdash p$. ▌

THEOREM 10. *Validity in* APDL *is decidable in deterministic exponential time.*

Proof. By Corollary 8 p is valid if $\neg p \notin A$ for each $A \in W$. As discussed above, the construction of W can be carried out deterministically in time $2^{O(|p|)}$. ▌

4. EXTENSIONS OF APDL

Some extensions of APDL can be shown to be exponentially decidable and complete by modification of the proofs in Section 3.

(1) Deterministic APDL, DAPDL for short, is syntactically identical to APDL but the structures $\mathscr{A} = (W, \tau, \rho)$ are restricted so that for every $a \in \Pi_0$ if $(s, t) \in \rho(a)$ and $(s, t') \in \rho(a)$ then $t = t'$. To prove that DAPDL is exponentially decidable we change the definition of ρ_0 for $\mathscr{A}_0$ to be: $(A, B) \in \rho_0(a)$ iff for every $\langle a \rangle q \in Z$, $\langle a \rangle q \in A$ iff $q \in B$.

The proof of Lemma 1 follows now as for APDL, except that we then have to show that the final structure $\mathscr{A} = (W, \tau, \rho)$ can be "unwound" into a tree like deterministic structure as in the decidability proof for DPDL in Ben-Ari, Halpern, and Pnueli (1982).

For a complete axiomatic system the following axiom is added to those of Section 3: (A7) $\langle a\rangle p \supset [a]\, p$.

The proof of Lemma 5 for the case $\alpha \in \Pi_0$ is now as follows: for every B such that $q \in B$, $(A, B) \notin \rho_0(a)$ it must be that either there is some $[a]\, r \in A$ with $\neg r \in B$ and hence $\vdash \hat{A} \supset [a] \neg \hat{B}$, as in the proof of Lemma 5, or there is some $\langle a\rangle\, r \in A$ with $\neg r \in B$ and hence $\vdash \hat{A} \supset \langle a\rangle\, r$ and $\vdash r \supset \neg \hat{B}$. By axiom (A7) it follows that $\vdash \hat{A} \supset [a]\, r$ and using (D) $\vdash \hat{A} \supset [a] \neg \hat{B}$.

(2) APDL with converse (or reverse), CAPDL for short. This version of APDL allows converse programs α^- which have the meaning $\rho(\alpha^-) = \{(s, t) \mid (t, s) \in \rho(\alpha)\}$. Formulas of the form $\langle (n, i, j, \delta)^- \rangle\, q$ can be translated to $\langle n, j, i, \delta' \rangle\, q$, where $\delta'(k, l) = (\delta(l, k))^-$ for every $l, k \in \bar{n}$ such that $\delta(l, k)$ is defined. Hence we can translate CAPDL formulas into formulas such that the only programs that appear with converse are atomic programs (note that $p?^- = p?$). The definition of the structure $\mathscr{A}$ is extended as follows: Π_0' now consists of the atomic programs and reverse atomic programs that appear in p, and the definition of ρ_0' is extended for $b \in \Pi_0'$ by: $(A, B) \in \rho_0'(a)$ iff

(a) for every $[a]\, q \in A$ we have $q \in B$

(b) for every $[a^-]\, q \in B$ we have $q \in A$.

In addition, $(A, B) \in \rho_0'(a^-)$ iff

(a)′ for every $[a^-]\, q \in A$ we have $q \in B$

(b)′ for every $[a]\, q \in B$ we have $q \in A$.

For the final structure $\mathscr{A}$, ρ is defined by: For $a \in \Pi_0$, $\rho(a) = \rho'(a) \cup \{(s, t) \mid (t, s) \in \rho'(a^-)\}$. The proof of Lemma 1 follows now as for APDL. To obtain a complete axiom system for CAPDL we add the axioms (A7) $p \supset [a]\langle a^- \rangle\, p$ and (A8) $p \supset [a^-]\langle a \rangle\, p$.

The proof of the first part of Lemma 5 is changed as follows: Let $\langle a \rangle\, q \in A$. For B with $q \in B$, and $(A, B) \notin \rho_0(a)$, either there exists some $[a]\, r \in A$ with $\neg r \in B$, which implies $\vdash \hat{A} \supset [a] \neg \hat{B}$, or there exists some $[a^-]\, r \in B$ with $\neg r \in A$, which implies $\vdash \hat{B} \supset [a^-]\, r$, and hence $\vdash \langle a^- \rangle \neg r \supset \neg \hat{B}$. By (D) it follows that $\vdash [a]\langle a_- \rangle \neg r \supset [a] \neg \hat{B}$. Now by $\vdash \hat{A} \supset \neg r$ and axiom (A7) it follows that $\vdash \hat{A} \supset [a] \neg \hat{B}$. This yields the first part of Lemma 5. Axiom (A8) is used similarly for the case $\langle a^- \rangle\, q \in A$.

(3) APDL with *loop*, LAPDL, is a version of APDL which allows assertions of the form: "there exists an infinite computation of α from a

state s." Formally: for a structure $\mathscr{A} = (W, \tau, \rho)$ formulas of the form *loop* (n, i, j, δ) have the meaning:

$$\begin{aligned}\tau(loop(n, i, j, \delta)) = \{s \mid {} & \exists(i_0, i_1, ...), i_0 = i, \forall k \geqslant 0\ i_k \in \bar{n} \wedge \\ & \exists(s_0, s_1, ...), s_0 = s, \forall k \geqslant 0\ s_k \in W \wedge \\ & (s_k, s_{k+1}) \in \rho(\delta(i_k, i_{k+1})), \forall k \geqslant 0\}.\end{aligned}$$

The corresponding version of PDL, LPDL or PDL$^+$, is discussed in (Harel and Pratt, 1978; Streett, 1982). The best known decision procedure for LPDL is of triple-exponential complexity (Streett, 1982). No completeness result has been obtained for LPDL. By using the representation of programs as automata and extending the ideas used in this paper, the second author together with Pnueli have provided LAPDL with an exponential time decision procedure for validity and a simple complete axiomatic system. Clearly these results, which will appear separately, imply corresponding results for LPDL.

Acknowledgments

We thank B. Herwig for pointing out an error in an early version of our similar completeness proof for PDL, prompting a correction in the present paper too. A. Pnueli suggested an important improvement in the structure of the main proof.

References

Abrahamson, K. R. (1980), "Decidability and Expressiveness of Logics of Processes," Ph.D. thesis, TR 80-08-01, University of Washington, Seattle, Wash.

Ben-Ari, M., Halpern, J. Y., and Pnueli, A. (1982), Deterministic propositional dynamic logic: Finite models, complexity, and completeness, *J. Comput. System. Sci.* **25**, 402–417.

Ehrenfeucht, A., and Zeiger, P. (1976), Complexity measures for regular expressions, *J. Comput. System Sci.* **12**, 134–146.

Floyd, R. W. (1967), Assigning meanings to programs, *in* "19th AMS Sympos. Appl. Math.," pp. 19–31, Amer. Math. Soc., Providence, R. I.

Fischer, M. J., and Ladner, R. E. (1979), Propositional dynamic logic of regular programs, *J. Comput. System. Sci.* **18**, 194–211.

Harel, D. (1984), Dynamic logic, *in* "Handbook of Philosophical Logic," Vol. II, pp. 497–604, Reidel, Dordrecht.

Harel, D., and Pratt, V. R. (1978), Nondeterminism in logics of programs, *in* "5th ACM Sympos. on Principles of Programming Languages," pp. 203–213.

Hoare, C. A. R. (1969), An axiomatic basis for computer programing, *Comm. ACM* **12**, 576–583.

Kozen, D., and Parikh, R. (1981), An elementary proof of the completeness of PDL, *Theoret. Comput. Sci.* **14**, 113–118.

PRATT, V. R. (1976), Semantical considerations on Floyd–Hoare logic, *in* "17th IEEE Sympos. on Found. of Comput. Sci.," pp. 119–121.

PRATT, V. R. (1979), Models of program logics, *in* "20th IEEE Sympos. on Found. of Comput. Sci.," pp. 115–122.

PRATT, V. R. (1981), Using graphs to understand PDL, *in* "Workshop on Logics of Programs" (D. Kozen, Ed.), Lect. Notes in Comput. Sci. Vol. 131, pp. 387–396, Springer-Verlag, New York.

SHERMAN, R., AND HAREL, D. (1983), A combined proof of one exponential decidability and completeness for PDL, *in* "1st Int. Workshop on Found. Theoret. Comput. Sci.," pp. 221–233, GTI, Paderborn, West Germany.

STREETT, R. S. (1982), Propositional dynamic logic of looping and converse is elementarily decidable, *Inform. and Control* **54**, 121–141.

WOLPER, P. (1983), Temporal logic can be more expressive, *Inform. and Control* **56**, 72–99.