Propositional Dynamic Logic of Flowcharts

D. Harel and R. Sherman

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

Abstract

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 [FL,P1] is used to reason about the before-after behaviour 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 [FL,P2,KP,SH,Ha].)

Pratt [P3] 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 [EZ]. 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 [P3] 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 this paper we borrow the motivation and some basic ideas of [P3] 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 [W] for his extended temporal logic. Also, the combined proof itself is a simplification of the similar proof we have given for PDL [SH], 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 [F] inductive assertion method for partial correctness relates to Hoare's [Ho] axiomatic system. This point is also hinted at in [P3].

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 author and A. Pnueli to prove exponential time decidability for PDL with \textit{loop}, previously known to be decidable only in triple-exponential time [S]. Section 2 contains preliminaries and Section 3 contains the main results.

2. Syntax and Semantics

\textbf{Definition :} A \textit{finite (nondeterministic) automaton} over an alphabet $\Sigma$ is a 4-tuple $\mathcal{F} = < Q, q_0, \eta, F >$ where:

- $Q$ - is a finite set of states.
- $q_0 \in Q$ - is the initial state.
- $\eta : Q \times \Sigma \rightarrow 2^Q$ - 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 $\sigma \in \Sigma^*$, $\sigma = (\sigma_0 \ldots \sigma_{\ell-1})$, is \textit{accepted} by $\mathcal{F}$ if there exists a sequence of states $(q_0, \ldots, q_\ell)$ such that $q_\ell \in F$ and for every $i$, $0 \leq i < \ell$, $q_{i+1} \in \eta(q_i, \sigma_i)$. 