INFORMATION AND COMPUTATION **81**, 249–264 (1989)

An Automata Theoretic Decision Procedure for the Propositional Mu-Calculus

ROBERT S. STREETT

Department of Mathematics and Computer Science, Mills College,
5000 Macarthur Boulevard, Oakland, California 94613

AND

E. ALLEN EMERSON*

Computer Science Department, University of Texas,
Austin, Texas 78712

The propositional mu-calculus is a propositional logic of programs which incorporates a least fixpoint operator and subsumes the propositional dynamic logic of Fischer and Ladner, the infinite looping construct of Streett, and the game logic of Parikh. We give an elementary time decision procedure, using a reduction to the emptiness problem for automata on infinite trees. A small model theorem is obtained as a corollary. © 1989 Academic Press, Inc.

1. INTRODUCTION

First-order logic is inadequate for formalizing reasoning about programs; concepts such as termination and totality require logics strictly more powerful than first-order (Kfoury and Park, 1975). The use of a least fixpoint operator as a remedy for these deficiencies has been investigated by Park (1970, 1976), Hitchcock and Park (1973), de Bakker and de Roever (1973), de Roever (1974), Emerson and Clarke (1980), and others. The resulting formal systems are often called mu-calculi and can express such important properties of sequential and parallel programs as termination, liveness, and freedom from deadlock and starvation.

Dynamic logic (Pratt, 1976; Harel, 1979) applies concepts from modal logic to a relational semantics of programs to yield systems for reasoning about the before–after behavior of programs. Analogous to the modal logic assertions $\Diamond\, p$ (possibly p) and $\Box\, p$ (necessarily p) are the dynamic logic constructs $\langle A \rangle\, p$ and $[A]\, p$. If A is a program and p is an assertion about the state of a computation, then $\langle A \rangle\, p$ asserts that after executing A, p can be the case, and $[A]\, p$ asserts that after executing A, p must be the case.

* The work of the second author was supported in part by NSF Grant MCS-8302878.

0890-5401/89 $3.00

Propositional versions of the mu-calculus have been proposed by Pratt (1981) and Kozen (1982). These logics use a least fixpoint construct to increase the expressive power of propositional dynamic logic (PDL) of Fischer and Ladner (1979). Kozen's formulation captures the infinite looping construct of Streett (1982) and subsumes Parikh's game logic (1983a, 1983b), whereas Pratt's logic is designed to express the converse operator of PDL. The filtration-based decision procedure and small model theorem obtained for PDL extend to Pratt's mu-calculus, but the ability to express infinite looping renders the filtration technique inapplicable to Kozen's version.

Kozen (1982) and Vardi and Wolper (1984) have obtained exponential time decision procedures for fragments of Kozen's mu-calculus. Both fragments can express all of PDL, but are not strong enough to capture the infinite looping construct of Streett (1981). Kozen and Parikh (1983) have shown that the satisfiability problem for the full propositional mu-calculus can be reduced to the second-order theory of several successor functions (*SnS*). By results of Rabin (1969) this supplies a decision procedure for the propositional mu-calculus, but one which runs in non-elementary time, i.e., time not bounded by any fixed number of compositions of exponential functions. Meyer (1974) has shown that Rabin's algorithm for *SnS* cannot be substantially improved; *SnS* is inherently nonelementary.

In this paper, we show that the satisfiability problem for sentences of the mu-calculus can be reduced to a certain emptiness problem for finite automata on infinite trees (Rabin, 1969; Hossley and Rackoff, 1972). A result of Streett (1981) shows that this reduction can be used to derive a triple-exponential time decision procedure for the propositional mu-calculus. Vardi (1984) has recently claimed a better upper bound for the automata theoretic emptiness problem, which would lead to an exponential space decision procedure.

2. Syntax and Semantics

Definition 2.1. The formulas of the propositional mu-calculus are:

(1) propositional letters $P, Q, R, \ldots,$

(2) propositional variables $X, Y, Z \ldots,$

(3) $\neg p$, $p \vee q$, and $p \wedge q$, where p and q are any formulas,

(4) $\langle A \rangle p$ and $[A] p$, where A is a member of a set of program letters $A, B, C, \ldots$ and p is any formula,

(5) $\mu X.f(X)$ and $\nu X.f(X)$, where $f(X)$ is any formula syntactically monotone in the propositional variable X, i.e., all occurrences of X in $f(X)$ fall under an even number of negations.

A sentence is a formula containing no free propositional variables, i.e., no variables unbound by a μ or ν operator. Sentences are interpreted in Kripke structures (borrowed from Kripke's semantics for modal logic (Kripke, 1963)), in which propositional letters denote subsets of states and program letters denote binary relations on states.

DEFINITION 2.2. A Kripke structure is a triple $\langle U, \models, \rightarrow \rangle$, where U is a universe of states, $\models$ is a satisfaction relation between states and propositional letters, and $\rightarrow$ gives, for each program letter A, a binary relation $\rightarrow^A$ on states.

DEFINITION 2.3. A model is a Kripke structure with the satisfaction relation $\models$ extended to all sentences by means of the following rules: (In what follows we use informally the notion of a formula being satisfied under an interpretation of its free variables.)

(1) $x \models \neg p$ iff $x \not\models p$,

(2) $x \models p \vee q$ iff $x \models p$ or $x \models q$,

(3) $x \models p \wedge q$ iff $x \models p$ and $x \models q$,

(4) $x \models \langle A \rangle p$ iff for some state y, $x \rightarrow^A y$ and $y \models p$,

(5) $x \models [A]\, p$ iff for every y such that $x \rightarrow^A y$, $y \models p$,

(6) $x \models \mu X.f(X)$ iff $x \in \bigcap\{S \subseteq U \mid S = \{y \mid y \models f(X)$ with X interpreted as $S\}\}$,

(7) $x \models \nu X.f(X)$ iff $x \in \bigcup\{S \subseteq U \mid S = \{y \mid y \models f(X)$ with X interpreted as $S\}\}$.

In a sentence $\mu X.f(X)$, f denotes a monotone function (monotonicity is ensured by the syntactic monotonicity of the formula $f(X)$) on sets of states, and $\mu X.f(X)$ is interpreted as the least fixpoint of this operator, i.e., the smallest set S of states such that $S = f(S)$. The sentence $\nu X.f(X)$ denotes the greatest fixpoint of the function f. The sentences $\mu X.f(X)$ and $\nu X.f(X)$ are dual, i.e., $\nu X.f(X) \equiv \neg\, \mu X. \neg\, f(\neg X)$.

EXAMPLE. Here are some rather trivial fixpoint sentences:

(1) $\mu X.X \equiv \mathit{false}$, $\nu X.X \equiv \mathit{true}$,

(2) $\mu X.P \equiv \nu X.P \equiv P$,

(3) $\mu X.X \vee P \equiv P$, $\nu X.X \vee P \equiv \mathit{true}$,

(4) $\mu X.X \wedge P \equiv \mathit{false}$, $\nu X.X \wedge P \equiv P$,

(5) $\mu X.\langle A \rangle X \equiv \mathit{false}$,

(6) $\nu X.[A] X \equiv \mathit{true}$.

EXAMPLE. The sentence $\nu X. \langle A \rangle X$ is true at x if there is an infinite chain of A edges from x. It is equivalent to the infinite looping construct ΔA of Streett (1982). Its negation, $\neg \nu X. \langle A \rangle X$, can also be written as $\mu X. [A] X$ (∇A in the notation of Streett).

EXAMPLE. The sentence $\mu X. P \vee \langle A \rangle X$ is true at a state x if there is a chain (possibly empty) of A edges leading from x to a state satisfying P. It is equivalent to the sentence $\langle A^* \rangle P$ of PDL.

EXAMPLE. In PDL, if α is a regular expression over the alphabet of program letters, we can form a sentence $\langle \alpha \rangle p$, which is true at a state when there is a chain of edges labelled with a string from the regular set α leading to a state satisfying p. The following transformation rules show how to translate such sentences into the mu-calculus:

(1) $\langle A \rangle p \Rightarrow \langle A \rangle p$,

(2) $\langle \alpha ; \beta \rangle p \Rightarrow \langle \alpha \rangle \langle \beta \rangle p$,

(3) $\langle \alpha \cup \beta \rangle p \Rightarrow \langle \alpha \rangle p \vee \langle \beta \rangle p$,

(4) $\langle \alpha^* \rangle p \Rightarrow \mu X. p \vee \langle \alpha \rangle X$.

For example, the PDL sentence $\langle A^* \cup A ; (B \cup AC)^* \rangle \langle B \rangle P$ is equivalent to the mu-calculus sentence $(\mu X. \langle B \rangle P \vee \langle A \rangle X) \vee \langle A \rangle (\mu X. \langle B \rangle P \vee \langle B \rangle X \vee \langle A \rangle \langle C \rangle X)$. Note that the translation is not succinct; consider a PDL sentence $\langle A \cup B \rangle \cdots \langle A \cup B \rangle P$.

DEFINITION 2.4. A formula is in positive form when all negations apply directly to propositional letters. The following rules can be used to convert a formula to positive form:

(1) $\neg \neg P \Rightarrow P$,

(2) $\neg (p \vee q) \Rightarrow \neg p \wedge \neg q$,

(3) $\neg (p \wedge q) \Rightarrow \neg p \neg q$,

(4) $\neg \langle A \rangle p \Rightarrow [A] \neg p$,

(5) $\neg [A] p \Rightarrow \langle A \rangle \neg p$,

(6) $\neg \mu X. f(X) \Rightarrow \nu X. \neg f(\neg X)$,

(7) $\neg \nu X. f(X) \Rightarrow \mu X. \neg f(\neg X)$.

DEFINITION 2.5. Let positive(p) denote the positive form of a sentence p, and let not(p) denote positive ($\neg p$), i.e., a positive representation of the negation of p.

It will be convenient to deal only with positive sentences. It is straightforward to extend a satisfaction relation from positive sentences to all sentences by means of the rule: $x \models p$ iff $x \models$ positive(p).

3. Ordinal Ranks and Signatures

By the Tarski–Knaster theorem, least and greatest fixpoints of monotonic functions over subsets of a set U can be defined by transfinite induction, i.e., the least fixpoint $\mu(f) = \bigcup_\alpha \mu_\alpha(f)$, where

$\mu_0(f) = \varnothing$,

$\mu_{\alpha+1}(f) = f(\mu_\alpha(f))$,

$\mu_\lambda(f) = \bigcup_{\alpha<\lambda} \mu_\alpha(f)$, for λ a limit ordinal.

Similarly, the greatest fixpoint $\nu(f) = \bigcap_\alpha \nu_\alpha(f)$, where

$\nu_0(f) = U$,

$\nu_{\alpha+1}(f) = f(\nu_\alpha(f))$,

$\nu_\lambda(f) = \bigcap_{\alpha<\lambda} \nu_\alpha(f)$, for λ a limit ordinal.

It will be useful to consider an extension of the propositional mu-calculus which contains, for each ordinal α and formula $f(X)$ syntactically monotone in X, formulas $\mu_\alpha X.f(X)$ and $\nu_\alpha X.f(X)$. A model can be extended to cover these ordinal sentences by means of the following additional rules:

(8) $x \not\models \mu_0 X.f(X)$,

(9) $x \models \mu_{\alpha+1} X.f(X)$ iff $x \models f(\mu_\alpha X.f(X))$,

(10) if λ is a limit ordinal, then $x \models \mu_\lambda X.f(X)$ iff for some $\alpha < \lambda$, $x \models \mu_\alpha X.f(X)$,

(11) $x \models \nu_0 X.f(X)$,

(12) $x \models \nu_{\alpha+1} X.f(X)$ iff $x \models f(\nu_\alpha X.f(X))$,

(13) if λ is a limit ordinal, then $x \models \nu_\lambda X.f(X)$ iff for all $\alpha < \lambda$, $x \models \nu_\alpha X.f(X)$,

It is then possible to recast rules (6) and (7) of Definition 2.3 in the forms

(6′) $x \models \mu X.f(X)$ iff for some ordinal α, $x \models \mu_\alpha X.f(X)$,

(7′) $x \models \nu X.f(X)$ iff for all ordinals α, $x \models \nu_\alpha X.f(X)$.

Definition 3.1. A mu-sentence $\mu X.f(X)$ has rank α at a state x if α is the least ordinal such that $\mu_\alpha X.f(X)$ is true at α.

Example. Consider a model with an infinite backwards chain of A edges ending in a state satisfying P, i.e.,

$$\cdots \xrightarrow{A} x_n \xrightarrow{A} \cdots x_3 \xrightarrow{A} x_2 \xrightarrow{A} x_1 \models P.$$

If $x_n \models \neg P$ for $x > 1$, then the sentence $\mu X.P \vee \langle A \rangle X$ has rank n at x_n, for $n \geqslant 1$.

EXAMPLE. In a model in which there are arbitrarily long finite chains (but no infinite chains) of A edges from the state x, the sentence $\mu X.[A]X$ will have infinite rank $\geqslant \omega$ at x (if every A successor of x has only bounded chains of A edges then $\mu X.[A]X$ has exactly rank ω at x).

Remark. The range of the ordinals used in connection with the fixed points was not specified. We could take it to be the collection of all ordinals, which is a *proper class* rather than a set. It suffices however to take it to be the *set* of all ordinals of cardinality at most that of the state space, since the closure ordinal of a monotone operator will not be greater. This ensures that the lexicographical ordered collection of bounded length sequences of ordinals, as used subsequently, is a well-founded set.

Since a mu-sentence can contain other mu-sentences as subsentences, it is useful to associate a sequence of ordinal ranks to a sentence.

DEFINITION 3.2. A signature is a sequence of ordinals. If s and t are signatures, we will write $s < t$ to mean that s lexicographically precedes t. Over a set of bounded length signatures, the lexicographic ordering is a well-ordering.

DEFINITION 3.3. The mu-height of a sentence is the depth of nesting of mu-subsentences of the sentence.

EXAMPLE. The sentence $\mu X.P \vee \langle A\rangle(\mu Y.X \vee \langle B\rangle Y)$ has mu-height 1, since the subformula $\mu Y.X \vee \langle B\rangle Y$ is not a sentence (it contains a free variable X).

DEFINITION 3.4. Given a sentence p of mu-height n and a signature $s = \alpha_1 \cdots \alpha_n$, we say that p has signature s at x if s is the lexicographically least signature such that the sentence obtained by replacing each mu-subsentence $\mu X.f(X)$ of mu-height i by $\mu_{\alpha_i}\colon X.f(X)$ is true at x.

EXAMPLE. In a model in which the state x has countably many B-successors $y_1, ..., y_n, ...$ such that $\mu X.P \vee \langle A\rangle X$ has rank n at y_n, the sentence $[B]\mu X.P \vee \langle A\rangle X$ has signature ω at x.

EXAMPLE. Consider $\mu Y.(\mu X.P \vee \langle A\rangle(\mu Z.X \vee \langle B\rangle Z)) \vee \langle B\rangle Y$, with mu-height 2 and equivalent to the PDL sentence $\langle B^*\rangle\langle (AB^*)^*\rangle P$. Consider a model in which there is a chain

$$x_9 \xrightarrow{B} x_8 \xrightarrow{A} x_7 \xrightarrow{B} x_6 \xrightarrow{A} x_5 \xrightarrow{B} x_4 \xrightarrow{B} x_3 \xrightarrow{B} x_2 \xrightarrow{B} x_1 \models P.$$

If $x_n \models \neg P$ for $n > 1$ then this sentence has signature 3, 2 at x_9, 3, 1 at x_8, 2, 2 at x_7, 2, 1 at x_6, 1, 5 at x_5, 1, 4 at x_4, 1, 3 at x_3, 1, 2 at x_2, and finally signature 1, 1 at x_1.

LEMMA 3.5. *The following rules hold for signatures:*

(1) *if $p \vee q$ has signature s at x, then either p or q has signature $t \leqslant s$ at x.*

(2) *if $p \wedge q$ has signature s at x, then both p and q have signatures $\leqslant s$ at x.*

(3) *if $\langle A \rangle p$ has signature s at x, then p has signature s at some A-successor of x.*

(4) *if $[A] p$ has signature s at x, then p has signature $\leqslant s$ at all A-successors of x.*

(5) *if $\mu X.f(X)$ has signature s at x, then $f(\mu X.f(X))$ has signature $t < s$ at x.*

(6) *if $\nu X.f(X)$ has signature s at x, then $f(\nu X.f(X))$ has signature t, where s is a prefix of t.*

Proof. We will do case (5) only. Suppose $\mu X.f(X)$ has mu-height n. The mu-height of $f(\mu X.f(X))$ will be $m \geqslant n$. The mu-subsentences of $f(\mu X.f(X))$ can be divided into three classes:

(1) The proper mu-subsentences of $\mu X.f(X)$, with mu-height $< n$.

(2) $\mu X.f(X)$ itself, with mu-height $= n$.

(3) Mu-sentences properly containing $\mu X.f(X)$, with mu-height $> n$.

If $\mu Y. g(Y)$ is in the first class and can be replaced by $\mu_\alpha Y.g(Y)$ within $\mu X.f(X)$ at x, then it can similarly be replaced within $f(\mu X.f(X))$ at x. If $\mu X.f(X)$ has rank α at x, then $\mu X.f(X)$ can be replaced by $\mu_\beta X.f(X)$, for some $\beta < \alpha$, within $f(\mu X.f(X))$ at x. Hence if $\mu X.f(X)$ has signature $s = \alpha_1 \cdots \alpha_n$ at x, then $f(\mu X.f(X))$ will have signature $t = \beta_1 \cdots \beta_{n-1} \beta_n \beta_{n+1} \cdots \beta_m$ at x, where $\beta_i \leqslant \alpha_i$ for $i < n$ and $\beta_n < \alpha_n$, so that $t < s$.

4. CHOICE FUNCTIONS

We can evaluate simple sentences in models by recursively evaluating subsentences. Thus to check whether or not $P \vee \langle A \rangle Q$ is true at a state x we either confirm that P is true at x or we look for an A edge leading to a state satisfying Q. In order to evaluate fixpoint sentences, we will need to confirm the fixpoint property, i.e., that $\mu X.f(X) \equiv f(\mu X.f(X))$ and $\nu X.f(X) \equiv f(\nu X.f(X))$. Thus evaluating a sentence may require recursively evaluating a supersentence and hence subsentences of supersentences and vice versa. The set of sentences whose evaluation is triggered in this way is not too large, however, and can be defined as follows.

DEFINITION 4.1. The Fischer–Ladner closure of a sentence p in positive form, is the smallest set $\mathrm{FL}(p)$ of sentences satisfying the following constraints:

(1) $p \in \mathrm{FL}(p)$,

(2) if $q \in \mathrm{FL}(p)$ then $\mathrm{not}(q) \in \mathrm{FL}(p)$,

(3) if $q \vee r \in \mathrm{FL}(p)$ then $q, r \in \mathrm{FL}(p)$,

(4) if $q \wedge r \in \mathrm{FL}(p)$ then $q, r \in \mathrm{FL}(p)$,

(5) if $\langle A \rangle q \in \mathrm{FL}(p)$ then $q \in \mathrm{FL}(p)$,

(6) if $[A] q \in \mathrm{FL}(p)$ then $q \in \mathrm{FL}(p)$,

(7) if $\mu X.f(X) \in \mathrm{FL}(p)$ then $f(\mu X.f(X)) \in \mathrm{FL}(p)$.

(8) if $\nu X.f(X) \in \mathrm{FL}(p)$ then $f(\nu X.f(X)) \in \mathrm{FL}(p)$.

EXAMPLE. The Fischer–Ladner closure of the sentence $\mu X.[A]X$ contains only four sentences: $\mu X.[A]X$, $\nu X.\langle A \rangle X$, $[A]\mu X.[A]X$, and $\langle A \rangle(\nu X.\langle A \rangle X)$.

EXAMPLE. The Fischer–Ladner closure of the sentence $\mu X.P \vee \langle A \rangle X$ consists of the following eight sentences:

(1) $\mu XP.P \vee \langle A \rangle X$,

(2) $\nu X.\neg P \wedge [A]X$,

(3) $P \vee \langle A \rangle(\mu X.P \vee \langle A \rangle X)$,

(4) $\neg P \wedge [A](\nu X.\neg P \wedge [A]X)$,

(5) P,

(6) $\neg P$,

(7) $\langle A \rangle(\mu X.P \vee \langle A \rangle X)$,

(8) $[A](\nu X.\neg P \wedge [A]X)$,

LEMMA 4.2. *The cardinality of the Fischer–Ladner closure of a sentence p is linear in the length of p, i.e.,* $|\mathrm{FL}(p)| = O(|p|)$.

Proof. A straightforward adaptation of the proof for PDL (Fischer and Ladner, 1979).

The following definition includes exactly those properties of a model which can be easily checked by recursive evaluation of closure sentences.

DEFINITION 4.3. A pre-model is a Kripke structure with a satisfaction relation $\models$ extended to positive sentences under the following constraints:

(1) $x \models p$ iff $x \not\models \text{not}(p)$,

(2) $x \models p \vee q$ iff either $x \models p$ or $x \models q$,

(3) $x \models \langle A \rangle p$ iff there is some edge $x \rightarrow^{A} y$ such that $y \models p$,

(4) $x \models \mu X.f(X)$ iff $x \models f(\mu X.f(X))$.

A pre-model is almost a model, except that rule (4) permits $\mu X \cdot f(X)$ to be interpreted as an arbitrary fixpoint (least, greatest, or intermediate). (Rule (1) ensures the proper complementary behavior for negated propositional letters, conjunctions, universal program sentences, and greatest fixpoint sentences.)

Example. Consider a Kripke structure with a single state x such that $x \rightarrow^{A} x$ and $x \models \neg P$. This structure can be extended to a pre-model in which $x \models \mu X.P \vee \langle A \rangle X$, $x \models P \vee \langle A \rangle(\mu X.P \vee \langle A \rangle X)$, and $x \models \langle A \rangle(\mu X.P \vee \langle A \rangle X)$. This pre-model will not, however, be a model.

Fixpoint sentences can generate nonterminating evaluation sequences. For example, occurrences of $\mu X.X$ and $\nu X.X$ merely trigger re-evaluation of themselves via the fixpoint property, while $\mu X.\langle A \rangle X$ and $\nu X.\langle A \rangle X$ can generate infinite sequences of reoccurrences along a chain of A edges. The presence or absence of nonterminating evaluations distinguishes least from greatest fixpoints (both of which share the fixpoint property). Least fixpoint sentences must have terminating evaluations, while nontermination is consistent with the semantics for greatest fixpoints (this explains why $\mu X.X \equiv \mathit{false}$ and $\nu X.X \equiv \mathit{true}$).

Disjunctions $p \vee q$ and existential program sentences $\langle A \rangle p$ introduce a complication; termination of the evaluation process depends on the choice of disjunct or edge used to satisfy such sentences. For example, the sentence $\mu X.P \vee X$ expands to $P \vee (\mu X.P \vee X)$; the disjunct P leads to termination, the disjunct $\mu X.P \vee X$ to nontermination. Consider the sentence $\mu X.P \vee \langle A \rangle X$, equivalent to the PDL sentence $\langle A^{*} \rangle P$, which is satisfied in a Kripke structure exactly when the sentence P is true somewhere along some path of A's. By the fixpoint property, $\mu X.P \vee \langle A \rangle X$ is equivalent to the disjunction $P \vee \langle A \rangle(\mu X.P \vee \langle A \rangle X)$. A terminating evaluation occurs if the A edges chosen to satisfy $\langle A \rangle(\mu X.P \vee \langle A \rangle X)$ eventually lead to a state where the disjunct P can be chosen. Consistently choosing to evaluate the disjunct $\langle A \rangle(\mu X.P \vee \langle A \rangle X)$ will lead to a nonterminating evaluation along an infinite A chain (since nonterminating evaluations are consistent with greatest fixpoints; this explains why $\nu X.P \vee \langle A \rangle X \equiv (\mu X.P \vee \langle A \rangle X) \vee (\nu X.\langle A \rangle X)$).

We shall consider pre-models supplied with a choice function responsible for guiding the evaluation of least fixpoint sentences towards termination.

DEFINITION 4.4. A choice function for a pre-model is a function which chooses, for every occurrence of a disjunction at a state, an occurrence of one of the disjuncts at that state, and for every occurrence of an existential program sentence $\langle A \rangle q$ at a state, an occurrence of q at an A-successor of that state.

DEFINITION 4.5. Any choice function over a pre-model determines a derivation relation between occurrences of sentences, defined by the following rules:

(1) A disjunction, $q \vee r$, derives the disjunct selected by the choice function.

(2) A conjunction, $q \wedge r$, derives both conjuncts.

(3) A program sentence, $\langle A \rangle q$, occurring at a state x, derives the occurrence of q selected by the choice function.

(4) A program sentence, $[A]q$, occurring at x generates occurrences of q at all A-successors of x,

(5) A mu-sentence, $\mu X.f(X)$, derives $f(\mu X.f(X))$.

(6) A nu-sentence, $\nu X.f(X)$, derives $f(\nu X.f(X))$.

It should be obvious that a sentence can only derive members of its Fischer–Ladner closure.

We would like to say that a pre-model is in fact a model when there is no infinite derivation sequence which rederives a mu-sentence infinitely often. However, this claim is true only when restricted to derivations in which the given mu-sentence appears as a subsentence of every derivation step, hence the following definition.

DEFINITION 4.6. A least fixpoint sentence $\mu X.f(X)$ is regenerated from x to y if $\mu X.f(X)$ at x derives $\mu X.f(X)$ at y in such a way that $\mu X.f(X)$ is a subsentence of every derivation step.

EXAMPLE. The sentence $\mu Y.(\mu X.(P \vee \langle A \rangle(\mu Y.X \vee \langle B \rangle Y)) \vee \langle B \rangle Y)$ can be regenerated across a B-edge, but not across an A-edge. A derivation across an A-edge is possible, but requires $\mu X.P \vee \langle A \rangle(\mu Y.X \vee \langle B \rangle Y)$ as a derivation step.

EXAMPLE. The sentence $p = \mu X.(\nu X.P \wedge \langle A \rangle(\mu Y.X \vee Y)) \vee \langle A \rangle Y)$ is true when there is an infinite chain of A edges along which P is infinitely often true. Any model of this sentence will contain infinite derivation sequences rederiving p infinitely often, but the subsentence $q = \nu X.P \wedge \langle A \rangle(\mu Y.X \vee \langle A \rangle Y)$ must then occur infinitely often as a derivation step.

It is possible to construct a choice function such that any regeneration sequence from p ultimately terminates at the choice q from the derived disjunction $q \vee \langle A \rangle p$.

DEFINITION 4.7. A choice function is well founded when the regeneration relations for least fixpoint sentences are well founded. A pre-model is well founded if it has a well-founded choice function.

THEOREM 4.8. *Every model is a well-founded pre-model.*

Proof. Given a model, construct a choice function which always selects the choice with lexicographically least signature. If $\mu X.f(X)$, of mu-height n, is regenerated from x to y, then $\mu X.f(X)$ must be a subsentence of each derivation step. Hence each sentence in the derivation has mu-height $\geqslant n$, and thus signature of length at least n. We shall show that the signature of $\mu X.f(X)$ decreases (lexicographically) from x to y. The derivation sequence must begin with $\mu X.f(X) \Rightarrow f(\mu X.f(X))$. By Lemma 3.5, the signature of $f(\mu X.f(X))$ lexicographically precedes the signature of $\mu X.f(X)$ at the nth position. We shall show that the remaining derivation steps cannot cancel this initial decrease.

Clearly, derivation steps from conjunctions $p \wedge q$ and universal program sentences $[A]\, p$ cannot increase signature, regardless of the particular choice function involved. The use of a choice function which selects on the basis of least signatures guarantees that derivation steps from disjunctions $p \vee q$ and existential program formulas $\langle A \rangle\, p$ do not increase signature.

A derivation step may involve a fixpoint sentence $\mu Y.g(Y)$ or $\nu Z.h(Z)$ which contains $\mu X.f(X)$ as a subsentence. In the former case, signature does not increase. In the latter case, signature may actually increase, since the signature of $h(\nu Z.h(Z))$ may be an extension of the signature of $\nu Z.h(Z)$. However, the net change in signature from the original sentence $\mu X.f(X)$ at state x will still be decreasing, since extending the signature after the nth position cannot cancel the effect of a decrease at the nth position.

We have therefore shown that regeneration always decrease signature. The signatures occurring in a derivation sequence from a sentence p have bounded length (the upper bound is the maximum mu-height of a sentence in $\mathrm{FL}(p)$), so that the lexicographic ordering is well founded, forcing the regeneration relations to be well founded.

THEOREM 4.9. *Each well-founded pre-model is a model.*

Proof. Suppose M is a pre-model supplied with a choice function so that the regeneration relation for each mu-sentence is well founded. Then each occurrence of a mu-sentence is associated with an ordinal, the well-ordering ordinal of the regeneration relation from that occurrence. It is thus possible to define a signature $\bar{\alpha} = \alpha_1, \alpha_2, ..., \alpha_n$ for each sentence q at state x as follows:

$$\alpha_i = \text{lub}\{\alpha\text{: } q \text{ at } x \text{ generates mu-sentence } r \text{ at } y, r \text{ has mu-depth } i, \text{ and } r \text{ at } y \text{ has regeneration ordinal } \alpha\}.$$

The labelling L of M can be extended so that for each sentence q and state x, if $q \in L(x)$ then $q\bar{\alpha}$ is added to $L(x)$, thereby annotating each sentence with its signature in the labelling. It is now easy to argue by induction on formula structure and signature that

$$q\bar{\alpha} \in L(x) \text{ implies } x \models q\bar{\alpha}.$$

Thus $q \in L(x)$ implies $x \models q$, and M is indeed a model. This completes the proof of Theorem 4.9.

COROLLARY 4.10. *For any sentence p, if p has a model, then p has a model of bounded outdegree $\leqslant |p|$.*

Proof. Consider the subset of $\text{FL}(p)$ containing just the existential program sentences of the form $\langle A \rangle q$. This subset is no larger than $|p|$, since each program letter in p contributes at most one member to this subset. Any model M of p has, by Theorem 4.8, a well-founded choice function and thus defines a well-founded pre-model. Take the underlying Kripke structure of M and prune it to outdegree $\leqslant |p|$ by allowing edges $x \rightarrow^A y$ iff $x \models \langle A \rangle q$, where $\langle A \rangle q \in \text{FL}(p)$ and y is chosen for $\langle A \rangle q$ at x by the choice function of the original model M. The resulting, pruned Kripke structure together with the choice function still defines a well-founded pre-model M', which is of bounded outdegree $\leqslant |p|$. By Theorem 4.9, M' is indeed a model.

5. THE DECISION PROCEDURE

Corollary 4.10 states that every satisfiable mu-calculus sentence p has a model (or equivalently, a well-founded pre-model) with outdegree $\leqslant |p|$. Such structures can be unwound into labelled trees of outdegree (arity) $\leqslant |p|$ which are suitable as input to finite automata on infinite trees (Rabin, 1969; Hossley and Rackoff, 1972). In this section we will sketch how, given a fixed mu-calculus sentence p, to program such an automaton to recognize well-founded pre-models for p.

The input for the automaton for p will be a tree T where each node x has been labelled with a subset of $\text{FL}(p)$. We will assume that each disjunction occurring on a node is marked to indicate a chosen disjunct. We can number the existential program sentences occurring in $\mathrm{FL}(p)$ as $\langle A_1\rangle q_1, ..., \langle A_n\rangle q_n$ and assume that whenever $\langle A_i\rangle q_i$ occurs on a node, the choice function will choose the ith successor of the node. The automaton for p is built from two component automata, which we call the local automaton and the global automaton.

The local automaton is a large but simple deterministic automaton on infinite trees. It performs three tasks. First, it ensures that p is among the sentences labelling the root of the input tree. Second, it guarantees that at every node, the subset $S \subseteq \mathrm{FL}(p)$ on that node is locally consistent, i.e., that

(1) $q \in S$ iff $\mathrm{not}(q) \notin S$,

(2) $q \vee r \in S$ iff $q \in S$ or $r \in S$,

(3) if $q \vee r \in S$ then its chosen disjunct $\in S$,

(4) if $\mu X.f(X) \in S$ iff $f(\mu X.f(X)) \in S$,

(5) $\mu X.f(X)$ cannot regenerate itself within S.

Third, it checks that the input tree is edge consistent, i.e., that

(1) if $\langle A_i\rangle q_i$ occurs on x, then the ith successor of x is labelled with the sentence q_i,

(2) if $[A]q$ occurs on x, then for all i such that $A = A_i$, the ith successor of x is labelled with q.

The local automaton can be built with $O(2^{|p|})$ states; it needs to remember subsets of $\mathrm{FL}(p)$.

The global automaton is a smaller but more sophisticated nondeterministic automata on infinite strings; it will be run down every path of the input tree. Its purpose is to look for an infinite regeneration sequence for some mu-sentence in $\mathrm{FL}(p)$. It nondeterministically selects an occurrence of a mu-sentence and a chain of nodes leading from that occurrence. At each node in this chain it determines whether a regeneration sequence could continue across the node. In order to do this, it must remember the final derivation step from the preceding node, i.e., the existential or universal program sentence which extended the derivation across a program edge. The global automaton accepts if it can find a regeneration sequence which regenerates $\mu X.f(X)$ infinitely often. The global automaton needs only $O(|p|)$ states, since it remembers only single sentences in $\mathrm{FL}(p)$.

Since the global automaton accepts when it finds an infinite regeneration sequence, an input tree will be a well-founded tree model only when it is accepted by the local automaton and every path of the input tree is rejected by the global automaton.

It is possible to take the nondeterministic global automaton and convert it to a deterministic automaton which accepts exactly the paths rejected by the original automaton (such a construction is given by McNaughton, 1966). Unfortunately, the new automaton will have $O(2^{2^{|p|}})$ states, since McNaughton's construct involves a double exponential blowup. This new automaton can be combined with the local automaton to produce a single automaton on infinite trees, with $O(2^{2^{|p|}})$ states, which accepts only well-founded pre-models for p. The sentence p is satisfiable if and only if this final automaton accepts a non-empty set of input trees. Hossley and Rackoff (1972) give a decision procedure for testing whether or not an arbitrary infinite tree automaton accepts an empty or non-empty set of input trees; their decision procedure runs in time doubly exponential in the size of the state space of the automaton. We have thus arrived at a decision procedure for the propositional mu-calculus which runs in time quadruply exponential in the length of the sentence tested.

This decision procedure can be improved by noting that the global and local automata can be combined to yield a single complemented pairs automation with $O(2^{2^{|p|}})$ states but only $O(2^{|p|})$ pairs. The emptiness problem for complemented pairs automata with n states and m pairs is decidable in time $O(2^n \cdot 2^{2^m})$. (Complemented pairs automata and their emptiness problem have been investigated by Streett, 1981.) This yields a triply exponential time decision procedure for the mu-calculus.

Vardi (1984) considers the following automata theoretic problem: given an infinite tree automaton and an infinite string automaton, is there any input tree which is accepted by the infinite tree automaton while having every path rejected by the infinite string automaton. Vardi claims that, if the tree automaton has n states and the string automaton m states, then this emptiness problem is decidable in space polynomial in $n \cdot 2^m$. This result would yield an exponential space decision procedure for the mu-calculus.

An exponential space upper bound would be tantalizingly close to the exponential time lower bound which is currently the best known. This exponential time bound is a trivial extension of the Fischer and Ladner (1979) lower bound result for PDL.

The propositional mu-calculus satisfies a finite model theorem: every satisfiable sentence has a model with finitely many states. This result is an easy corollary of a result about automata on infinite trees: every automaton recognizable set of trees must contain a finitely generable tree, i.e., a tree obtained from unwinding a finite graph. Every satisfiable mu-calculus sentence p thus has a finite graph which unwinds into a model. In fact this finite graph is a finite model.

The results of this paper are easily extended to include multiple fixpoints as described by Vardi and Wolper (1984). Informally, an n-tuple of formulas $f_1(X_1, ..., X_n), ..., f(X_1, ..., X_n)$ (where the X_i's are free variables) denotes a monotonic function on tuples of sets of states. The least or greatest fixpoint of this function will be a tuple of states; selecting a component of this tuple yields a single set of states, i.e., a suitable interpretation for a sentence.

DEFINITION 5.1. The mu-calculus of multiple fixpoints includes the following sentences: If, for $1 \leqslant i \leqslant n$, $f_i(X_1, ..., X_n)$ is a formula syntactically monotone in all the free variables $X_1, ..., X_n$ (which need not be all the free variables in the f_i's), then for $1 \leqslant i \leqslant n$, $\mu X_i(X_1, ..., X_n).(f_1(X_1, ..., X_n), ..., f_n(X_1, ..., X_n))$ and $\nu X_i(X_1, ..., X_n).(f_1(X_1, ..., X_n), ..., f_n(X_1, ..., X_n))$ are formulas (with semantics described informally above).

The fixpoint property for multiple fixpoints is cumbersome to express without abbreviation. So, for $1 \leqslant i \leqslant n$, let p_i abbreviate $\mu X_i(X_1, ..., X_n).(f_1(X_1, ..., X_n), ..., f_n(X_1, ..., X_n))$. Then the fixpoint property for least fixpoints can be written as: $p_i \equiv f_i(p_1, ..., p_n)$.

Multiple fixpoints can be used to give a succinct (i.e., linear) translation of PDL into the propositional mu-calculus. The translation rule $\langle \alpha \cup \beta \rangle p \Rightarrow \langle \alpha \rangle p \vee \langle \beta \rangle p$ (which causes a potential exponential blowup through the duplication of p) can be replaced by the rule $\langle \alpha \cup \beta \rangle p \Rightarrow \mu X(X, Y).(AY \vee BY, P)$, which uses a double fixpoint to avoid duplication of p. Other uses of multiple fixpoints are discussed by Vardi and Wolper (1984).

RECEIVED August 9, 1985; ACCEPTED June 19, 1986

REFERENCES

DE BAKKER J., AND DE ROEVER, W. P. (1973), A calculus for recursive program schemes, *in* "First International Colloquium on Automata, Languages, and Programming," pp. 167–196.

DE ROEVER, W. P. (1974), "Recursive Program Schemes: Semantics and Proof Theory," Ph.D. thesis, Free University, Amsterdam.

EMERSON, E. A., AND CLARKE, E. M. (1980), Characterizing correctness properties of parallel programs using fixpoints, *in* "Seventh International Colloquium on Automata, Languages and Programming," pp. 169–181.

FISCHER, M. J., AND LADNER, R. E. (1979), Propositional dynamic logic of regular programs, *J. Comput. System Sci.* **18**, 194–211.

HAREL, D. (1979), "First-Order Dynamic Logic," Lecture Notes in Computer Science, Vol. 68, Springer-Verlag.

HITCHCOCK, P., AND PARK, D. M. R. (1973), Induction rules and termination proofs, *in* "First International Colloquium on Automata, Languages, and Programming," pp. 225–251.

HOSSLEY, R., AND RACKOFF, C. W. (1972), The emptiness problem for automata on infinite trees, *in* "Thirteenth IEEE Symposium on Switching and Automata Theory," pp. 121–124.

KFOURY, A. J., AND PARK, D. M. R. (1975), On termination of program schemes, *Inform. and Control* **29**, 243–251.

KOZEN, D. (1982), Results on the propositional mu-calculus, *in* "Ninth International Colloquium on Automata, Languages, and Programming," pp. 348–359.

KOZEN, D., AND PARIKH, R. J. (1983), A decision procedure for the propositional mu-calculus, *in* "Second Workshop on Logics of Programs."

KRIPKE, S. A. (1963), Semantical considerations on model logics, *Acta Philos. Fennica.*

MCNAUGHTON, R. (1966), Testing and generating infinite sequences by a finite automaton, *Inform. and Control* **9**, 521–530.

MEYER, A. R. (1974), Weak monadic second order theory of successor is not elementary recursive, *in* "Boston Logic Colloquium," Lecture Notes in Mathematics Vol. 453, Springer-Verlag, New York/Berlin.

PARIKH, R. J. (1979), A decidability result for a second order process logic, *in* "Nineteenth IEEE Symposium on the Foundations of Computer Science," pp. 177–183.

PARIKH, R. J. (1983a), Cake cutting, dynamic logic, games, and fairness, *in* "Second Workshop on Logics of Programs."

PARIKH, R. J. (1983b), Propositional game logic, *in* "Twenty-third IEEE Symposium on the Foundations of Computer Science."

PARK, D. M. R. (1970), "Fixpoint Induction and Proof of Program Semantics," Machine Intelligence Vol. 5, Edinburgh Univ. Press, Edinburgh.

PARK, D. M. R. (1976), Finiteness is mu-ineffable, *Theoret. Comput. Sci.* **3**, 173–181.

PRATT, V. R. (1976), Semantical considerations on Floyd–Hoare logic, *in* "Seventeenth IEEE Symposium on Foundations of Computer Science," pp. 109–121.

PRATT, V. R. (1982), A decidable mu-calculus: Preliminary report, *in* "Twenty-second IEEE Symposium on the Foundations of Computer Science," pp. 421–427.

RABIN, M. O. (1969), Decidability of second order theories and automata on infinite trees, *Trans. Amer. Math. Soc.* **141**, 1–35.

STREETT, R. S. (1981), "Propositional Dynamic Logic of Looping and Converse," Technical Report TR-263, MIT LCS.

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

STREETT, R. S., AND EMERSON, E. A. (1984), The propositional mu-calculus is elementary, *in* "Eleventh International Colloquium on Automato, Languages, and Programming," Lecture Notes in Computer Science Vol. 172, pp. 465–472, Springer-Verlag, New York/Berlin.

VARDI, M. (1984), private communication.

VARDI, M., AND WOLPER, P. (1984), Automata theoretic techniques for modal logics of programs, *in* "Sixteenth ACM Symposium on the Theory of Computing."