Information Processing Letters 24 (1987) 77–79
North-Holland

30 January 1987

UNIFORM INEVITABILITY IS TREE AUTOMATON INEFFABLE *

E. Allen EMERSON

Department of Computer Science, College of Natural Sciences, University of Texas at Austin, Taylor Hall 2.124, Austin, TX 78712-1188, U.S.A.

Communicated by G.R. Andrews
Received 20 March 1986
Revised 17 April 1986

In this short article we consider the limits of the expressive power of finite automata on infinite trees. We exhibit a surprisingly simple correctness property, uniform inevitability, which is not definable by any type of finite automaton on infinite trees.

Keywords: Temporal logic, expressive power, tree automaton, finite automaton on infinite trees

1. Introduction

The fundamental correctness property of concurrent programs known as *inevitability of predicate* P is expressible in the branching time temporal logic CTL* as, simply, AFP, meaning that along every future computation path π there exists a time i along π at which P is true (cf. [3]). The time i in general depends on the particular path π followed. In other words, it has the pattern $\forall$ path $\exists$ time P (cf. [2]). We also can consider the correctness property defined by swapping the positions of the quantifiers to get the—equally simple—pattern $\exists$ time $\forall$ path P, meaning that there exists a time i such that, along every computation path π, P holds at time i. We call this correctness property *uniform inevitability of predicate* P to emphasize that the time i is uniform over all paths π.

Despite the apparently close relationship between uniform inevitability and inevitability, we can show, using a simple 'pumping lemma' type argument, that uniform inevitability is not definable by any finite-state automaton on infinite trees. Hence, this property is not expressible in any of the temporal logics such as CTL, CTL*, PDL, PDL-Δ, YAPL, the propositional μ-calculus, etc. (cf. [3,4,5,7,8]) shown to be decidable in elementary time using finite automata on infinite trees (cf. [9]), since, as noted in [1], for these logics, not only is the problem of testing satisfiability reducible to that of testing nonemptiness of tree automata, but, in fact, all the temporal operators of these logics are directly definable as tree automata.

The remainder of this short article is organized as follows: in Section 2 we give some basic definitions and preliminaries, while the main result is proved in Section 3. Finally, some concluding remarks are made in Section 4.

2. Preliminaries

For simplicity, we only consider automata on infinite binary trees. The set $\{0, 1\}^*$ of all infinite strings over alphabet $\{0, 1\}$ may be viewed as an

* This research was partly supported by the National Science Foundation under Grant DCR8511354.

infinite binary tree where the *root node* is the empty string λ and each *node* $v \in \{0, 1\}^*$ has as its *successors* the nodes v_0 and v_1. A finite (respectively, infinite) *path* through the tree is a finite (respectively, infinite) sequence $\pi = v_0 v_1 v_2$ of nodes such that, for all i, v_{i+1} is a successor of v_i. Let Σ be a finite alphabet of symbols. An *infinite binary Σ-tree* is a labeling T which maps $\{0, 1\}^* \to \Sigma$.

A *finite automaton $\mathscr{A}$ on infinite binary Σ-trees* consists of a tuple (Σ, S, δ, s_0) plus an *acceptance condition* (which is described below), where

— Σ is the *input alphabet* labeling the nodes of the input tree,
— S is the finite set of *states* of the automaton,
— $\delta : S \times \Sigma \to \text{PowerSet}(S \times S)$ is the (nondeterministic) *transition function*, and
— $s_0 \in S$ is the *start state* of the automaton.

A *run* of $\mathscr{A}$ on the input Σ-tree T is a function $\rho : \{0, 1\}^* \to S$ such that, for all $v \in \{0, 1\}^*$,

$$(\rho(v_0), \rho(v_1)) \in \delta(\rho(v), T(v)) \text{ and } \rho(\lambda) = s_0.$$

We say that $\mathscr{A}$ *accepts* input Σ-tree T iff there exists a run ρ of $\mathscr{A}$ on T such that, for all infinite paths π starting at the root of $\{0, 1\}^*$, the acceptance condition (as below) holds along $r = \rho \mid \pi$, the sequence of states obtained by restricting the run ρ to path π.

Given an infinite sequence of states $r = s_0 s_1 s_2 \ldots$ we use In(r) to denote the set of states that appear infinitely often along r. For a *Muller* automaton, acceptance is defined in terms of a family $\mathscr{F} \subseteq \text{PowerSet}(S)$ of designated subsets of S. A sequence r meets the *Muller acceptance condition* iff $\text{In}(r) \in \mathscr{F}$. It is well known that all other types of acceptance, such as Büchi acceptance, pairs acceptance, and complemented pairs acceptance, can be viewed as special cases of the Muller acceptance condition. Therefore, without loss of generality, we restrict our attention to these Muller automata.

Given a set of atomic propositions $AP = \{P_1, \ldots, P_k\}$, let Σ be the set of 2^k strings $P_1^* \ldots P_k^*$ where each P_i^* denotes either P_i or (intuitively, its negation) $\overline{P}_i$. Then, finite automata on infinite binary Σ-trees can be viewed in an obvious way as defining (branching time) temporal logic modalities. For example, (ordinary) inevitability of P, AFP, corresponds to the automaton $\mathscr{A}_0 = (\Sigma, S, \delta, s_0, \mathscr{F})$, where (for simplicity)

$$\Sigma = \{P, \overline{P}\}, \qquad S = \{s_0, s_1\},$$
$$\mathscr{F} = \{\{s_1\}\}, \qquad \delta(s_0, P) = (s_1, s_1),$$
$$\delta(s_0, \overline{P}) = (s_0, s_0), \qquad \delta(s_1, P) = (s_1, s_1),$$

and

$$\delta(s_1, \overline{P}) = (s_1, s_1).$$

3. Main result

Theorem. *Uniform inevitability is not definable by a finite automaton on infinite trees.*

Proof. Suppose there is a Muller automaton $\mathscr{A}$ such that the language it accepts consists of exactly those Σ-trees ($\Sigma = \{P, \overline{P}\}$) which define models of $\exists$ time $\forall$ path P. $\mathscr{A}$ has some number n of states. Without loss of generality we may assume $n \geqslant 2$. Pick $m > n$, and let T_m be the Σ-tree such that exactly those nodes at level m are labeled with P, while all other nodes are labeled with $\overline{P}$; formally,

$$T_m(u) = \begin{cases} P & \text{if } |u| = m, \\ \overline{P} & \text{if } |u| \neq m. \end{cases}$$

Note that T_m is accepted by $\mathscr{A}$. So, let ρ be an accepting run of $\mathscr{A}$ on T_m, and consider the tree T'_m obtained by annotating T_m with (states according to) the run $\rho : T'_m(u) = (T_m(u), \rho(u))$. Let π be an infinite path in T'_m. Since $m > n$, there are two nodes x and xy along π such that ρ labels both x and xy with the same state s, and, moreover, $0 < |x| < |xy| < m$. Now, consider the annotated tree T'' formed by chopping out and replacing the subtree rooted at x by the subtree rooted at xy; formally, $T''(u) = T'_m(u)$ if x is not a prefix of u, and $T''(u) = T''(xw) = T'_m(xyw)$ if x is a prefix of u, with $u = xw$ (see Fig. 1). This tree T'' is still annotated by an accepting run of $\mathscr{A}$ since, along each path π of T'', $\text{In}(\pi) = \text{In}(\pi')$ for some path π' of T'_m, and all such paths π' meet the Muller acceptance condition. Thus, the tree T''' obtained from T'' by 'erasing' the state labels is a Σ-tree accepted by $\mathscr{A}$. On the other hand, T''' is not a

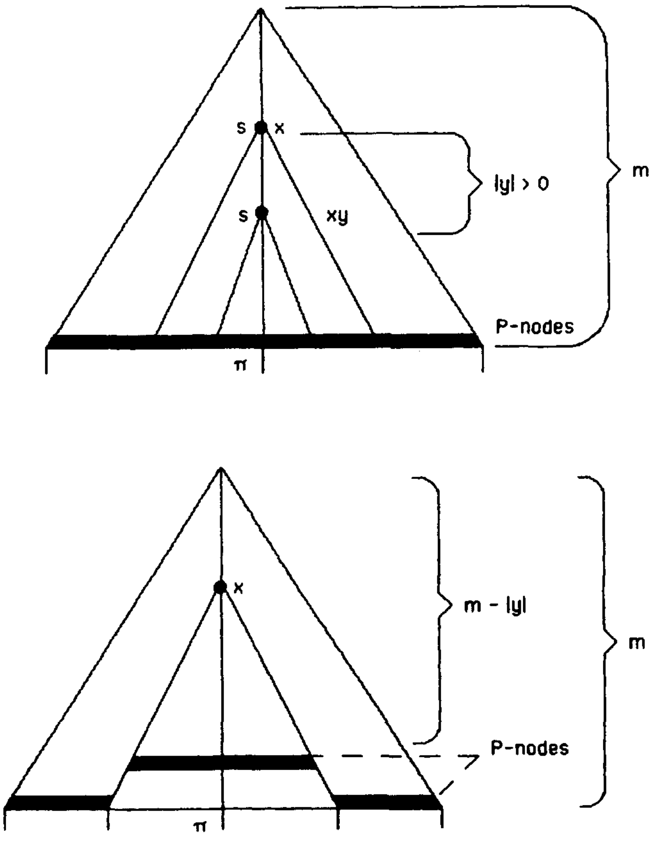

Fig. 1.

model of ∃ time ∀ path P since the P-nodes occur at two different levels. More precisely, along each infinite path π of T''' which does not go through node x there is a unique occurrence of a node z labeled P, where $|z| = m$, while along each path π that does go through x there is a unique node z' labeled P where $|z'| = m - |y|$. Thus, $\mathscr{A}$ accepts a nonmodel of ∃ time ∀ path P, a contradiction. We conclude that no finite-state Muller automaton $\mathscr{A}$, accepting exactly the models of ∃ time ∀ path P, exists. □

4. Conclusion

We have shown that the correctness property, uniform inevitability of P, is not definable by a finite automaton on infinite trees. It is interesting to note, however, that a tree automaton with a pushdown store can be used to define uniform inevitability. (It guesses the uniform time i by pushing i symbols on its stack initially, and then decrements the stack as it advances from level to level.) Thus, such pushdown tree automata and their applications to temporal logics may merit further study.

References

[1] E.A. Emerson, Automata, tableaux, and temporal logic (Extended Abstract), in: Proc. Conf. on Logics of Programs, New York, Lecture Notes in Computer Science, Vol. 193 (Springer, Berlin, 1985) 79–88 (full version in preparation, Spring 1986).

[2] E.A. Emerson and E.M. Clarke, Characterizing correctness properties of parallel programs as fixpoints, in: Proc. 7th Internat. Coll. on Automata, Languages, and Programming, Lecture Notes in Computer Science, Vol. 85 (Springer, Berlin, 1981).

[3] E.A. Emerson and J.Y. Halpern, 'Sometimes' and 'not never' revisited: On branching versus linear time temporal logic (with J. Halpern), J. ACM 33 (1) (1986) 151–178.

[4] E.A. Emerson and A.P. Sistla, Deciding full branching time logic, Inform. and Control 61 (3) (1984) 175–201.

[5] D. Kozen, Results on the propositional μ-calculus, Theoret. Comput. Sci. 27 (3) (1983) 333–354.

[6] R.S. Streett, Propositional dynamic logic of looping and converse, Inform. and Control 54 (1982) 121–141 (full version: Propositional dynamic logic of looping and converse, Ph.D. Thesis, MIT Lab. for Computer Science, 1981).

[7] R.S. Streett and E.A. Emerson, The propositional μ-calculus is elementary, in: Proc. 11th Internat. Coll. on Automata, Languages, and Programming, Antwerpen, Belgium, Lecture Notes in Computer Science, Vol. 172 (Springer, Berlin, 1984) 465–472.

[8] M. Vardi and P. Wolper, Yet another process logic, in: CMU Workshop on Logics of Programs, Lecture Notes in Computer Science, Vol. 164 (Springer, Berlin, 1983) 501–512.

[9] M. Vardi and P. Wolper, Automata theoretic techniques for modal logics of programs, ACM STOC84 (1984) 446–455.