Theoretical Computer Science 15 (1981) 111–114
North-Holland Publishing Company

NOTE

THE FINITE POWER PROPERTY FOR CONTEXT-FREE LANGUAGES

Charles E. HUGHES
Computer Science Department, University of Central Florida, Orlando, FL 32816, U.S.A.

Stanley M. SELKOW
Computer Science Department, Worcester Polytechnic Institute, Worcester, MA 01609, U.S.A.

Communicated by M. Nivat
Received October 1980
Revised December 1980

1. Introduction

A language L is said to have the finite power property if there exists some positive integer n such that $L^* = (\{\lambda\} \cup L)^n$. The problem of determining for which classes of languages the finite power property is decidable was first posed by J.A. Brzozowski in 1966. Simon [4] has developed a decision procedure for the class of regular languages. Our major result is to prove that this problem is undecidable for the class of context-free languages.

As a preliminary, we will first establish the unsolvability of the conceptually easier problem of testing if $L = L^2$ for context-free languages.

Theorem 1. *There is no effective procedure for deciding whether $L = L^2$ for an arbitrary context-free language L.*

Proof. The problem of deciding whether $L = \Sigma^*$ is known to be unsolvable. (See, for instance [3, p. 220].) We will show how to reduce this problem to the problem of deciding whether $L = L^2$.

Given an arbitrary context-free language L, we can certainly decide whether $(\{\lambda\} \cup \Sigma) \subset L$, (because $\{\lambda\} \cup \Sigma$ is a finite set). If $(\{\lambda\} \cup \Sigma) \not\subset L$, then $L \neq \Sigma^*$. But if $(\{\lambda\} \cup \Sigma) \subset L$, then clearly $L = \Sigma^*$ if and only if $L = L^2$.

If $L = L^2$, then L has the finite power property. To show that the converse is not true, consider the context-free language $L = \{\lambda, a, a^3, a^4, a^5, \ldots\}$. It is easily seen that $L \subsetneq L^2 = L^3$.

0304-3975/81/0000–0000/$02.50

In order to establish the undecidability of the finite power property, we will reduce the immortality problem for Turing machines, which is known to be undecidable, to our problem. An intermediate step in this reduction involves the uniform halting problem for Turing machines.

2. Turing machines that uniformly halt

Let T be an arbitrary Turing machine (TM). An *instantaneous description* (ID) of T is a (possibly infinite length) representation of T's current state, tape position and tape contents. Representations involving tapes for which only a finite number of squares are marked are called finite ID's.

A Turing machine T, with ID set $\mathscr{C}$, *uniformly halts* if there exists some n such that, for each finite ID $C \in \mathscr{C}$, T started at C halts execution after at most n steps. T is *immortal* if there is an ID (possibly of infinite length) from which it will never halt; otherwise it is *mortal*. The result of this paper is achieved by reducing the problem of deciding whether or not a Turing machine uniformly halts to the problem of deciding whether or not a context-free language has the finite power property. Unsolvability of the finite power property then proceeds from the following.

Theorem 2. *The set of Turing machines which uniformly halt is recursively enumerable.*

Proof. Any TM which uniformly halts in at most n steps cannot scan a square more than n squares from the initial square scanned. Therefore, there exist only a finite number of initial ID's to check in deciding if a TM halts in at most n steps, and the TM must be simulated for each such ID for at most n steps.

We can use a dovetailing procedure which will simulate the enumerable set of TM's to generate the subsets which uniformly halt in n steps, as n increases to infinity.

Theorem 3. *The set of immortal TM's is not recursive.*

Proof. See [2].

Theorem 4. *The set of mortal TM's is equal to the set of TM's which uniformly halt. Therefore, the set of TM's which uniformly halt is not recursive.*

Proof. Any TM which uniformly halts must clearly be mortal.

Let T be a TM which does not uniformly halt. If any finite ID does not lead to a halt, then clearly T is immortal. Assume then that T does not uniformly halt but all finite ID's cause it to halt. Let $\mathscr{I}$ be the set of all ID's such that for each $I \in \mathscr{I}$, when T starts in I it will eventually scan each square of the tape containing a symbol of I

before it scans a square not containing a symbol of I. Let $\{q_1, \ldots, q_m\}$ be the states of T. We define a forest of m trees, one for each state of T, such that the jth tree has root q_j. If $I_0, I_1 \in \mathcal{I}$, and q_j is a symbol of I_0 and I_1, and $I_1 = \sigma I_0$ or $I_1 = I_0\sigma$ where σ is a tape symbol, then I_0 is a parent of I_1 in the jth tree. Note that when T starts in I_1, the square containing σ is scanned after every other square of I_1 but before any square not in I_1. Since T does not uniformly halt but every finite ID causes it to halt, at least one of the trees of the forest must be infinite. The degree of each vertex in each tree is finite (it is bounded by the number of tape symbols). By König's Infinity Lemma, at least one of the trees must have an infinite branch. Therefore, there exists an infinite ID which causes T to travel an infinite distance on the tape. It follows that T is immortal.

3. Unsolvability of the finite power property

Let T be an arbitrary Turing machine and let $\mathscr{C}$ be the set of finite ID's of T. An ID C_2 is said to be the *immediate successor* of C_1 in T if C_2 is the ID occurring if T performs one step after being started at C_1. A sequence of finite ID's $C_1, C_2, \ldots, C_m$ is called a *trace* (of *length* m) of T if, for each i, $1 \leq i < m$, C_{i+1} is the immediate successor of C_i. T uniformly halts just in case there is some n such that no trace is of length greater than n. It is this property of traces that we use in order to construct a context-free language L having the finite power property if and only if T uniformly halts.

Our language L uses as its alphabet all symbols appearing in ID's of T plus the two special symbols \$ and ¢. In defining L we make use of the notation C^R to denote the reversal of a string C. That is, if $C = a_1a_2 \ldots a_m$ then $C^R = a_m \ldots a_2a_1$. L is defined as follows:

$$L_1 = \{C_1\$C_2^R¢ \mid C_1, C_2 \in \mathscr{C}\},$$

$$L_2 = \{C_1\$C_2^R¢C_3\$C_4^R¢ \ldots C_{2k-1}\$C_{2k}^R¢ \mid k \geq 1 \text{ and, for some } i \geq 1, C_{i+1} \text{ is not the immediate successor of } C_i\},$$

$$L = L_1 \cup L_2.$$

L may easily be shown to be context-free. (See, for example, [1].)

Our analysis of L^* is simplified by the fact that

$$L_2^* = \{\lambda\} \cup L_2 \quad \text{and} \quad L_1L_2 = L_2L_1 = L_2.$$

Thus,

$$\{\lambda \cup L\}^n = \{\lambda\} \cup L_1 \cup L_1^2 \ldots L_1^n \cup L_2.$$

Analyzing the definitions of L_1 and L_2 we see that

$$L_1^n \cap L_2 = \emptyset \quad \text{(the empty set)}$$

just in case there is some word

$$C_1\$C_2^R¢C_3\$C_4^R¢\ldots C_{2n-1}\$C_{2n}^R¢$$

in L_1^n which word is not also in L_2. But then, this is so just in case there is some trace of length $2n$. Clearly then, L has the finite power property if and only if Γ uniformly halts. In summary

Theorem 5. *The finite power property for context-free languages is unsolvable.*

Acknowledgment

The connection between the uniform halting problem and the immortality problem for Turing machines was first suggested to us by J. Opatrny of Concordia University.

References

[1] J. Hartmanis, Context-free languages and Turing machine computations, *Proc. Symposia in Applied Mathematics* **19** (AMS, Providence, RI, 1967).
[2] P.K. Hooper, The undecidability of the Turing machine immortality problem, *J. Symbolic Logic* **31** (1966) 219–234.
[3] J.E. Hopcroft and J.D. Ullman, *Formal Languages and their Relation to Automata* (Addison-Wesley, Reading, MA, 1969).
[4] T. Simon, Limited subsets of a free monoid, *Proc. 19th Foundations of Computer Science Symposia* (IEEE, New York, 1978) 143–150.