# RECOGNITION OF DETERMINISTIC ET0L LANGUAGES IN POLYNOMIAL TIME

by

Neil D. Jones

and

Sven Skyum

DAIMI PB-63

October 1976

Institute of Mathematics University of Aarhus

DEPARTMENT OF COMPUTER SCIENCE

Ny Munkegade - 8000 Aarhus C - Denmark

Phone 06-12 83 55

## ABSTRACT

It is shown that if G is a deterministic ET0L system, there is a nondeterministic log space algorithm to determine membership in L(G). Consequently, every deterministic ET0L language is recognizable in polynomial time. As a corollary, all context-free languages of finite index, and all indian parallel languages are recognizable within the same bounds.

# INTRODUCTION

Recently considerable interest has been shown in questions concerning the complexity of the membership problem for various types of L systems[1]. In van Leeuwen (1975a) it is shown that there is an ET0L system G such that the set L(G) is $\mathfrak{NP}$-complete. Opatrný and Culik (1975) showed that E0L membership could be determined by a DTIME $(n^4)$ algorithm, and Sudborough (1976) gave a DSPACE($(\log n)^2$) algorithm (based on a construction in van Leeuwen (1975b)). Sudborough also gave a DSPACE(log n) algorithm for membership in any deterministic E0L language.

In this paper we show that membership in L(G) for a deterministic ET0L system G may be determined in NSPACE(log n) and hence that L(G) is also recognizable by a deterministic polynomial time algorithm. This indicates that recent attempts to find an $\mathfrak{NP}$-complete deterministic ET0L language are quite unlikely to succeed, since this would imply $\mathfrak{P} = \mathfrak{NP}$ = NSPACE(log n), a major breakthrough in complexity theory.

A consequence of the main theorem is that every indian parallel language, and every context-free language of finite index is in NSPACE(log n) (see Salomaa (1973) for definitions). It is unknown whether every context-free language is in NSPACE(log n), and has been for some time. Thus our results enlarge the known extent of the intersection of NSPACE(log n) with the context-free languages.

---

1. We assume the reader is familiar with the basic concepts concerning L systems and time- or tape-bounded Turing machines (e.g. see Hopcroft & Ullmann (1969) and Herman & Rozenberg (1975)). Also, let DSPACE(S(n)) denote the family of sets accepted by deterministic offline multitape Turing machines which operate in space S(n) on inputs of length n. NSPACE(S(n)) is defined analogously for nondeterministic machines, and DTIME(S(n)) and NTIME(S(n)) are similarly defined for the time measure.

## THE MAIN THEOREM

Let $G = (V, \wp, a_1, \Sigma)$ be a fixed propagating deterministic ET0L system, where $V = \{a_1, a_2, \ldots, a_p\}$. The problem is to determine whether an arbitrary word $x \in \Sigma^*$ is in $L(G)$. Let $n = |x|$, and let $\# \notin V$.

<u>Definition 1</u> An <u>assignment</u> (of subwords of $x$ to symbols in $V$) is any p-tuple $(x_1, \ldots, x_p)$ where each $x_i$ is either $\#$ or a subword of $x$. The notation $\bar{x}$ will be used to abbreviate $(x_1, \ldots, x_p)$.

<u>Definition 2</u> An assignment $\bar{x}$ is said to be <u>consistent</u> if there is a table sequence $T_1, \ldots, T_m \in \wp$ such that for each $i = 1, \ldots, p$ either $x_i = \#$ or $a_i \xRightarrow[T_1 \ldots T_m]{} x_i$.

<u>Remarks</u> Suppose $\alpha \overset{*}{\Rightarrow} x$. Since $G$ is deterministic, every occurrence of a symbol $a_i$ in $\alpha$ must derive the same subword of $x$. Thus we can construct an assignment $\bar{x}$ from $\alpha$ by letting $x_i$ be the subword derived from $a_i$ if $a_i$ occurs in $\alpha$, and letting $x_i = \#$ if $a_i$ does not occur in $\alpha$. Our algorithm will operate by nondeterministically simulating a derivation by $G$, and storing only the assignment corresponding to the current sentential form, rather than the sentential form itself. Clearly it is necessary to show that this procedure is valid – i.e. that any sequence of assignments obtained this way will in fact correspond to some derivation.

Any subword of $x$ can be identified by its starting and ending positions within $x$, and so within $0(\log n)$ bits. Since $p$ is fixed, it follows that an entire assignment can also be stored in $0(\log n)$ space.

<u>Definition 3</u> Let $\bar{x}$ and $\bar{y}$ be assignments. By definition $\bar{x} \rightarrow \bar{y}$ if there is a table $T \in \wp$ such that for $i = 1, \ldots, p$, if $y_i \neq \#$, then

$y_i = x_{i_1} x_{i_2} \ldots x_{i_k}$ , where $a_i \to a_{i_1} \ldots a_{i_k}$ is the unique $a_i$ production in T.

<u>Lemma 1</u> Let $\bar{x}$ and $\bar{y}$ be assignments. If $\bar{x}$ is consistent and $\bar{x} \to \bar{y}$ then $\bar{y}$ is also consistent.

<u>Proof</u> Let $T_1 \ldots T_m$, T and $a_i \to a_{i_1} \ldots a_{i_k}$ be as in Definitions 2 and 3. If $y_i \neq \#$ then $a_i \underset{T}{\Rightarrow} a_{i_1} \ldots a_{i_k} \xRightarrow[T_1 \ldots T_m]{} x_{i_1} \ldots x_{i_k} = y_i$. Thus for each i, $y_i \neq \#$ or $a_i \xRightarrow[TT_1 \ldots T_m]{} y_i$, which means that $\bar{y}$ is consistent. □

Define $\mathrm{Alph}(\alpha)$ to be the set of all letters which occur in $\alpha$, for any $\alpha \in V^*$.

<u>Lemma 2</u> Let $\alpha, \beta \in V^*$ be such that $\alpha \underset{T}{\Rightarrow} \beta \xRightarrow[T_1 \ldots T_m]{} x$, and let $\bar{x}$ be a consistent assignment such that for each i, $a_i \xRightarrow[T_1 \ldots T_m]{} x_i$ if $a_i \in \mathrm{Alph}(\beta)$, and $x_i = \#$ otherwise. Then there exists a consistent assignment $\bar{y}$ such that $\bar{x} \to \bar{y}$ for each i, $a_i \xRightarrow[TT_1 \ldots T_m]{} y_i$ if $a_i \in \mathrm{Alph}(\alpha)$, and $y_i = \#$ otherwise.

<u>Proof</u> Define for all i

$$y_i = \begin{cases} x_{i_1} \ldots x_{i_k} & \text{if } a_i \in \mathrm{Alph}(\alpha) \text{ and } a_i \to a_{i_1} \ldots a_{i_k} \text{ is in T} \\ \# & \text{if } a_i \notin \mathrm{Alph}(\alpha) \end{cases}$$

Note that $a_i \in \mathrm{Alph}(\alpha)$ implies that $a_{i_1}, \ldots, a_{i_k} \in \mathrm{Alph}(\beta)$, and therefore that $x_{i_r} \neq \#$ for $1 \leq r \leq k$. It is now immediate that $\bar{y}$ satisfies the conditions of the statement of the lemma. □

Lemma 3 $x \in L(G)$ if and only if there exists a sequence of assignments $\bar{x}^{(1)} \to \bar{x}^{(2)} \to \ldots \to \bar{x}^{(k)}$ such that $\bar{x}^{(1)} = (b_1, \ldots, b_p)$, where $b_i = a_i$ if $a_i \in \mathrm{Alph}(x)$ and $b_i = \#$ otherwise, and $\bar{x}^{(k)} = (x, \#, \ldots, \#)$.

Proof The "if" part is immediate from Lemma 1. Now suppose $a_1 = \alpha_k \Rightarrow \ldots \Rightarrow \alpha_2 \Rightarrow \alpha_1 = x$. Note that $\bar{x}^{(1)}$ is consistent and satisfies the conditions of Lemma 2 with empty $T_1 \ldots T_m$, $\alpha = \alpha_1$ and $\beta = \alpha_2$. Thus there is a consistent $\bar{x}^{(2)}$ such that $\bar{x}^{(1)} \to \bar{x}^{(2)}$. The remaining $\bar{x}^{(j)}$ are obtained in the same way. □

Theorem For each deterministic ET0L system G, L(G) is in NSPACE(log n).

Proof Without loss of generality G is propagating (see e.g. Nielsen et al. (1974)). By the preceding lemma membership may be determined as follows:

```
for i := 1 to p do
    x_i := if a_i ∈ Alph(x) then a_i else #;
x̄ := (x_1,...,x_p);
while x̄ ≠ (x, #,...,#) do
    begin Pick assignment ȳ at random
        if x̄ → ȳ then x̄ := ȳ
                  else halt
    end;
accept
```

At most two assignments must be stored at any time, which by earlier remarks can be done in O(log n) space. □

## CONSEQUENCES

Corollary 1

Any deterministic ET0L languages is recognizable in polynomial time.

This is immediate, since it is well known that NSPACE(log n) $\subseteq \mathcal{P}$. However, the usual construction for this requires at least time $n^{cp}$, where $c$ is a positive constant and $p$ is the size of the alphabet of $G$. This is a definite contrast to the situation for context-free or E0L languages, which have membership algorithms which operate in time $0(n^{2.81})$ and $0(n^4)$ respectively, regardless of $G$. Also, it may be of interest to note that the non-deterministic algorithm above operates in time bounded by a polynomial whose degree is independent of $p$.

Corollary 2

Every indian parallel language is in NSPACE(log n).

Corollary 3

Every context-free language of finite index is in NSPACE(log n).

Proofs Immediate from Theorem 2 of Skyum (1974) and Theorem 6 of Salomaa (1974). □

Corollary 4

There is a deterministic ET0L language which is NSPACE(log n) complete. This is immediate from Sudborough (1975).

## References


Herman, G.T. and Rozenberg, G. (1974) Developmental Systems and Languages. North-Holland.

Hopcroft, J.E., and Ullmann, J.D. (1969) Formal Languages and their Relation to Automata. Addison-Wesley.

Nielsen, M., Rozenberg, G., Salomaa, A. and Skyum, S. (1974) Nonterminals, Homomorphisms and Codings in Different Variations of 0L systems. I. Deterministic Systems. Acta Informatica 4, 87-106.

Opatrný, J. and Culik, K. II (1975) Time complexity of L-languages, Abstracts of papers, Conference on Formal Languages, Automata and Development, University of Utrecht, Netherlands.

Salomaa, A. (1973) Formal Languages. Academic Press, Inc.

Salomaa, A. (1974) Parallelism in Rewriting Systems, in Loeckx, J. (ed.) Automata, Languages and Programming, Springer-Verlag.

Skyum, S. (1974) Parallel Context-free Languages, Inf. and Control, Vol. 26, No. 3, pp. 280-285.

Sudborough, I.H. (1975) A note on tape-bounded complexity classes and linear context-free languages, JACM 22, pp. 499-500.

Sudborough, I.H. (1976) The Complexity of the Membership Problem for Some Extensions of Context-free Languages. Report. Dept. of Comp. Sciences, Northwestern University, Evanston, Illinois 60201.

van Leeuwen, J. (1975a) The membership question for ET0L languages is polynomially complete. Inf. Proc. Letters 3, 138-143.

van Leeuwen, J. (1975b) The Tape Complexity of Context-independent Developmental Languages, J. Comp. and Syst. Sc. 11, 203-211.