COMPLEXITY OF SOME PROBLEMS CONCERNING L SYSTEMS (Preliminary Report)

by

Neil D. Jones
and
Sven Skyum

DAIMI PB-67
November 1976

Institute of Mathematics University of Aarhus
DEPARTMENT OF COMPUTER SCIENCE
Ny Munkegade - 8000 Aarhus C - Denmark
Phone 06-12 83 55

1. INTRODUCTION

Recently, considerable interest has been shown in questions concerning the complexity of the membership problem for various types of L systems. Van Leeuwen showed in [11] that there are ETOL systems G such that L(G) is complete for $\mathcal{NP}$ (nondeterministic polynomial time). Opatrný and Culik showed in [7] that EOL membership (for fixed grammars) may be decided deterministically in time n^4, and Sudborough gave a $(\log n)^2$ space algorithm for the same problem in [10], based on a construction by van Leeuwen [12]. Sudborough also gave a deterministic $\log n$ space algorithm for EDOL membership in [10], and showed in [9] that some linear languages (and hence some EOL and deterministic ETOL languages) are complete for nondeterministic log space. In a companion paper [4], we have shown that each deterministic ETOL language can be recognized nondeterministically in $\log n$ space, and therefore deterministically in polynomial time.

In this paper we study the complexity of the emptiness and finiteness questions for each of these classes (ETOL, EOL, and their deterministic counterparts), as well as the <u>general</u> membership problem.

Let $\bar{G}$ be a linearly encoded form of an ETOL system over a fixed alphabet independent of G. (E.g. represent symbols $v_1, v_2, \ldots, v_m$ in the form $V\bar{i}$ where $\bar{i}$ is the binary representation of i, $1 \le i \le m$.) The problems we discuss may all be represented in terms of membership in the following sets. C denotes any of the system classes just mentioned.

1. $\text{NONEMPTY}^C = \{\bar{G} \mid G \text{ is in } C \text{ and } L(G) \neq \emptyset\}$
2. $\text{INFINITE}^C = \{\bar{G} \mid G \text{ is in } C \text{ and } L(G) \text{ is infinite}\}$
3. $\text{MEMBER}^C = \{<\bar{G}, \bar{x}> \mid G \text{ is in } C \text{ and } x \text{ is in } L(G)\}$
4. $L(G)$ for a fixed grammar G in C

The work referenced above establishes upper and lower complexity bounds on problems of type 4 (except for a lower bound on deterministic E0L membership). We shall outline a series of constructions which suffice to establish both upper and lower bounds on the remaining problems (in most cases rather tight). As we shall see, the complexity of the general membership problem (in which the input is the system as well as the terminal string) can be much higher than that of determining whether x is in $l(G)$ for some fixed G. In the most extreme case, if C is the class of deterministic ET0L grammars, membership for fixed systems may be determined in $\log n$ space, while the general problem requires essentially linear space (both by nondeterministic algorithms).

2. TERMINOLOGY AND RESULTS

The results may be presented in the form of a table as follows. For the sake of comparison we have included the context-free and context-sensitive classes as well. In the system class names, D indicates "deterministic", and P indicates "propagating" (i.e. the absence of productions with the empty string on the right side).

PROBLEM

GRAMMAR CLASS	MEMBER (FIXED G)	MEMBER (GENERAL)	NONEMPTY	INFINITE	BOUNDS
CONTEXT SENSITIVE	NSPACE(n)	NSPACE (n log n)	UNDECIDABLE	UNDECIDABLE	UPPER
		NSPACE(n)			LOWER
ETOL, EPTOL	$n\mathcal{P}$	NSPACE (n log n)	NSPACE(n)	NSPACE(n)	UPPER
		NSPACE $(n^{1-\epsilon})$	NSPACE $(n^{1-\epsilon})$	NSPACE $(n^{1-\epsilon})$	LOWER
EDTOL, EPDTOL	$n\mathcal{L}$	NSPACE (n log n)	NSPACE(n)	NSPACE(n)	UPPER
		NSPACE $(n^{1-\epsilon})$	NSPACE $(n^{1-\epsilon})$	NSPACE $(n^{1-\epsilon})$	LOWER
EOL, EPOL	DSPACE($\log^2$ n) DTIME(n^4)	$n\mathcal{P}$	DSPACE(n)	NSPACE(n)	UPPER
	$n\mathcal{L}$		$n\mathcal{P}$	$n\mathcal{P}$	LOWER
EDOL, EPDOL	$\mathcal{L}$	$n\mathcal{P}$	$n\mathcal{P}$	$n\mathcal{P}$	UPPER
	$\mathcal{L}$	$\mathcal{L}$			LOWER
CONTEXT FREE	DSPACE($\log^2$ n) DTIME(n^3)	$\mathcal{P}$	$\mathcal{P}$	$\mathcal{P}$	UPPER
	$n\mathcal{L}$				LOWER

TERMINOLOGY

1. $\mathrm{DSPACE}(S(n)) = \{ L \mid L$ is accepted by some <u>deterministic</u> offline Turing machine which operates within <u>space</u> $S(n)$ on all inputs of length $n\}$

   $\mathrm{NSPACE}(S(n))$ is defined analogously for nondeterministic machines, and and $\mathrm{DTIME}(S(n))$, $\mathrm{NTIME}(S(n))$ are defined similarly for the time measure.

2. $\mathcal{L} = \mathrm{DSPACE}(\log n)$, $\quad \mathfrak{n}\mathcal{L} = \mathrm{NSPACE}(\log n)$

   $\mathcal{P} = \bigcup_{k=1}^{\infty} \mathrm{DTIME}(n^k)$, $\quad \mathfrak{n}\mathcal{P} = \bigcup_{k=1}^{\infty} \mathrm{NTIME}(n^k)$

3. A table entry of the form $\boxed{\begin{array}{c} U \\ \hline L \end{array}}$ for problem P indicates that

   a) P is in class U

   b) If L is $\mathfrak{n}\mathcal{L}$, $\mathcal{P}$ or $\mathfrak{n}\mathcal{P}$, then some complete problem (and so <u>any</u> problem) in class L is reducible to P

   c) if L is $\mathrm{NSPACE}(S(n,\epsilon))$, then for any $\epsilon > 0$, P is <u>not</u> in $\mathrm{NSPACE}(S(n,\epsilon))$.

   d) If L is $\mathcal{L}$, then any algorithm which solves P in $\mathrm{DSPACE}(S(n))$ must satisfy $\sup\limits_{n\to\infty} \dfrac{S(n)}{\log n} > 0$.

4. A table entry $\boxed{LU}$ for problem P indicates that P is complete for class LU.

3. OVERVIEW OF PROOF METHODS

Theorem NONEMPTY^{EDOL} is $\mathcal{NP}$ hard.

Proof Method By Stockmeyer & Meyer [8] the following problem is $\mathcal{NP}$-hard:

Given a regular expression R of the form

$$0^{p_1}(0^{q_1})^* + \ldots + 0^{p_r}(0^{q_r})^*$$

to determine whether $L(R) \neq 0^*$.

Construct an EDOL system

$G = (\{Z_i^j \mid 1 \leq i \leq r,\ 0 \leq j \leq p_i + q_i - 1\},\ P,\ Z_1^o \ldots Z_r^o,$
$\{Z_i^j \mid j \neq p_i,\ 1 \leq i \leq r,\ 0 \leq j \leq p_i + q_i - 1\})$ where P consists of the productions $(i = 1, \ldots, r)$:

$Z_i^j \rightarrow Z_i^{j+1}$ for $j = 0, \ldots, p_i + q_i - 2$

$Z_i^{p_i + q_i - 1} \rightarrow Z_i^{p_i}$.

Then $L(G) \neq \emptyset$ iff $L(R) \neq 0^*$; consequently NONEMPTY^{EDOL} is $\mathcal{NP}$ hard.

Theorem NONEMPTY^{EDOL} is in $\mathcal{NP}$.

Proof Method Let $G = (V, P, w, \Sigma)$ be an EDOL grammar. Construct a nondeterministic finite automaton $M = (V, \{0\}, \delta, S_0, \Sigma)$ where $S_0 = \{a \in V \mid a \text{ occurs in } w\}$, and $\delta(a) = \{a_1, a_2, \ldots, a_m\}$ just in case $a \rightarrow a_1 \ldots a_m$ is a production in P. It is easily seen that $L(G) \neq \emptyset$ if and only if $L(M) \neq 0^*$. By Stockmeyer and Meyer, this test can be carried out nondeterministically in polynomial time. □

Corollary

NONEMPTY^{EDOL} is $\mathcal{NP}$ complete.

Theorem MEMBER^{EPOL} is in $\mathcal{NP}$.

Proof Method Given $<\overline{G}, \overline{x}>$, we can determine whether $x \in L(G)$ as follows:

```
α := Axiom of G;
for I := 1 Step 1 until |x| do
begin choose β so that α ⇒* β and |α| = |β|;
      if β = x then accept;
      choose γ so β ⇒ γ and |β| < |γ|;
      α := γ
end
```

This procedure will provide a polynomial time membership algorithm if the step "choose β ..." can be done in polynomial time; however there can be nonrepeating derivations of length greater than any polynomial in $|\overline{G}|$. Let $\alpha = a_1 \ldots a_m$ and $\beta = b_1 \ldots b_m$. Then $\alpha \overset{p}{\Rightarrow} \beta$ iff $a_1 \overset{p}{\Rightarrow} b_1, \ldots,$ and $a_m \overset{p}{\Rightarrow} b_m$; and $\alpha \overset{*}{\Rightarrow} \beta$ iff $\alpha \overset{p}{\Rightarrow} \beta$ for some $p \leq k^m$ where k is the size of the alphabet of G. The test $a_i \overset{p}{\Rightarrow} b_i$ can be done by forming a connection matrix M ($M(a,b) = 1$ iff $a \to b$ is a production), and calculating $M, M^2, M^4, \ldots, M^{2^{\lceil m \cdot \log k \rceil}}$ by repeated squaring. M^p may be obtained as a product of some of these matrices, chosen nondeterministically; and $\alpha \overset{p}{\Rightarrow} \beta$ may be easily determined from M^p. □

Theorem NONEMPTY$^{E0L} \in$ DSPACE(n).

Proof Let $G = (V, P, w, \Sigma)$ be given. Define $A_0 = \Sigma$, $A_{i+1} = \{a \mid a \to \alpha$ is a production in P such that $\alpha \in A_i^* \}$. Then $L(G) \neq \emptyset$ iff $w \in A_i^*$ for some i. The DSPACE(n) algorithm is simply to calculate $A_0, A_1, \ldots$, storing only the most recent one (as a bit vector), and comparing the letters in w against A_i. □

__Theorem__ $\mathrm{INFINITE}^{\mathrm{ET0L}} \in \mathrm{NSPACE}(n)$.

__Proof Method__ $L(G)$ is infinite if and only if there exists a derivation of a word $x \in L(G)$ such that $S \Rightarrow^* v_1 a v_2 \Rightarrow^* w_1 \alpha w_2 \Rightarrow^* x$, where $a \Rightarrow^* \alpha$, $\mathrm{Alph}(v_1 a v_2) = \mathrm{Alph}(w_1 \alpha w_2)$, and α contains the letter a and another occurrence of a letter, say b, yielding a nonempty subword of x.

The algorithm simulates such a derivation by nondeterministically choosing $v_1 a v_2$, a, and b and checking whether the statements above are satisfied. The only information needed for that,is information about the alphabet of the current sentential form and two letters derived from a and b. $\square$

__Theorem__ $\mathrm{MEMBER}^{\mathrm{EDT0L}} \notin \mathrm{NSPACE}(n^{1-\epsilon})$ for any $\epsilon > 0$.

__Proof__ Let $Z = (K, \Sigma, \Gamma, \#, \delta, q_0, \{q_f\})$ be an arbitrary 1 tape Turing machine which operates in space n ($\#$ is an end marker). For any $x = a_1 \ldots a_n$, construct the EDT0L system $G_x = (V_n, \mathfrak{T}_n, w_x, \{0\})$ where

$$V_n = \{g, 0\} \cup \{A^i \mid A \in \Gamma \text{ and } 0 \leq i \leq n+1\} \cup K$$

$$w_x = p\,\#^0\, a_1^1 a_2^2 \ldots a_n^n\, \#^{n+1}$$

for each $(p, a) \in (K - \{q_f\}) \times \Gamma$ there will be a table $T_{p,a}$ in $\mathfrak{T}_n$ defined as follows:

If $\delta(p, a) = (q, b, R)$ then

$$T_{p,a} = \{p \to q,\ a^0 \to b^{n+1}\} \cup \{c^i \to c^{i-1} \mid c \in \Gamma,\ 0 < i \leq n+1\} \cup G_{p,a}$$

where $G_{p,a}$ contains $d \to g$ for every $d \in V_n$ other than p, a^0 or c^i for $c \in \Gamma$, $0 < i \leq n+1$.

If $\delta(p, a) = (q, b, C)$ then

$$T_{p,a} = \{p \to q,\ a^0 \to b^0\} \cup \{c^i \to c^i \mid c \in \Gamma,\ 0 < i \leq n+1\} \cup G_{p,a}$$

If $\delta(p, a) = (q, b, L)$ then

$$T_{p,a} = \{p \to q,\ a^0 \to b^1\} \cup \{c^i \to c^{i+1} \mid c \in \Gamma,\ 0 < i \leq n\} \cup \{c^{n+1} \to c^0 \mid c \in \Gamma\} \cup G_{p,a}.$$

In addition, $\mathfrak{T}_n$ contains the table

$$T_f = \{q_f \to 0\} \cup \{c^i \to 0 \mid c \in \Gamma,\ 0 \leq i \leq n+1\} \cup \{a \to g \mid a \in K \cup \{g, 0\} - \{q_f\}\}.$$

It is easily verified that Z yields an I.D. $\alpha = b_0 \ldots b_{i-1}\, p\, b_i \ldots b_{n+1}$ iff G derives the string $p\, b_0^{n-i+2} \ldots b_{i-1}^{n+1}\, b_i^0 \ldots b_{n+1}^{n-i+1}$. Consequently $L(G) = \{0^{n+3}\}$ if Z accepts x, and $L(G) = \emptyset$ if Z does not accept x. Further, $|\overline{G}| = 0(n \log n)$. In the usual way this implies $\mathrm{MEMBER}^{\mathrm{EDT0L}} \notin \mathrm{NSPACE}(n^{1-\epsilon})$, for any $\epsilon > 0$. □

<u>Corollary</u>

$\mathrm{MEMBER}^{\mathrm{EDT0L}}$ is complete for polynomial space.

<u>Theorem</u> There is a deterministic E0L language L such that if L is in DSPACE(S(n)), then

sp $$\sup_{n \to \infty} \frac{S(n)}{\log n} > 0$$

<u>Proof</u> $L = \{ab^n cd^n \mid n \geq 0\}$ is clearly a deterministic E0L language. By Alt and Mehlhorn [1], L in DSPACE(S(n)) implies that S must satisfy the condition above. □

REFERENCES

[1] Alt, H., Mehlhorn, K. Lower bounds for the space complexity of context-free recognition, Automata, Languages and Programming, 338-354, July 1976, University Press, Edinburgh, Scotland.

[2] Hopcroft, J.E., Ullman, J.D. Formal Languages and their Relation to Automata, Addison-Wesley, 1969, 242 pp.

[3] Jones, N.D., Laaser, W.T., Complete Problems for Deterministic Polynomial Time, J. Theoretical Computer Science, 1976.

[4] Jones. N.D., Skyum, S., EDT0L Membership in Polynomial Time, Technical report, University of Aarhus, Denmark.

[5] Lewis, P.M., Stearns, R.E., Hartmanis, J., Memory Bounds for the Recognition of Context-free and Context-sensitive Languages, IEEE Conference Record on Switching Circuit Theory and Logical Design, Ann Arbor, Michigan, pp. 191-202.

[6] Meyer, A.R., Stockmeyer, L.J. The Equivalence Problem for Regular Expressions with Squaring Requires Exponential Space, 13th IEEE Symposium on Switching and Automata Theory, Oct. 1972, pp. 125-129.

[7] Opatrný, J., Culik, K. II, Time Complexity of L Languages, Abstracts of papers, Conference on formal languages, automata and development, University of Utrecht, Netherlands, 1975.

[8] Stockmeyer, L.J., Meyer, A.R., Word Problems requiring Exponential Time, 5th ACM Symposium on Theory of Computing, May 1973, pp. 1-9.

[9] Sudborough, I.H. A Note on Tape-bounded Complexity Classes and Linear Context-free Languages, J. ACM 22, 1975, pp. 499-500.

[10] Sudborough, I.H. The Complexity of the Membership Problem for some Extensions of Context-free Languages, Technical Report, Northwestern University Computer Science Dept., Evanston, Ill., 1976.

[11] van Leeuwen, J. The Membership Question for ET0L Languages is Polynomially Complete, Information Processing Letters 3, 1975, pp. 138-143.

[12] van Leeuwen, J. The Tape Complexity of Context-independent Developmental Languages, J. Computer and Systems Sciences 15, 1975, pp. 203-211.

[13] Younger, D.H. Recognition and Parsing of Context-free Languages in Time n^3, Information and Control 10:2, pp. 189-208.