ISSN 0105-8517

COMPLEXITY OF SOME PROBLEMS CONCERNING L SYSTEMS

by

Neil D. Jones
and
Sven Skyum

DAIMI PB-85 (Revised version)
February 1979

Computer Science Department
AARHUS UNIVERSITY
Ny Munkegade – DK 8000 Aarhus C – DENMARK
Telephone: 06 – 12 83 55

ABSTRACT

We determine the computational complexity of membership, emptiness and infiniteness for several types of L systems. The L systems we consider are EDOL, EOL, EDTOL, and ETOL, with and without empty productions. For each problem and each type of system we establish both upper and lower bounds on the time or memory required for solution by Turing machines.

1. INTRODUCTION

The theory of computational complexity (see [1]) has made it possible to compare previously studied language families in a new way - by the relative complexity of their decision problems. Recently several authors have examined the complexity of some questions concerning L systems, a family of language-generating devices which are similar to context-free grammars but which interpret the productions as parallel rewriting rules (see [4] for an introduction). In this paper we obtain both upper and lower bounds for the complexity of the general membership, finiteness and emptiness problems for several classes of L systems.

We begin by summarizing previous results in this area. Van Leeuwen [17] showed that there is an ET0L system G such that L(G) is complete for $\mathfrak{NP}$ (the family of languages nondeterministically recognizable in polynomial time). He also showed [16] that E0L membership (for fixed systems) may be decided deterministically in time $n^{3.81}$, and Sudborough [13] and [14] gave a $(\log n)^2$ space algorithm for the same problem, based on a construction by van Leeuwen [18]. Sudborough [14] also gave a deterministic log n space algorithm for ED0L membership, and showed in [13] that some linear languages (and hence some E0L and deterministic ET0L languages) are complete for nondeterministic log space. Harju [3] showed that each deterministic ET0L language can be recognized in polynomial time. Jones and Skyum [7] showed that EDT0L membership is complete for nondeterministic log space, using an independently discovered algorithm similar to that of [3]; and the same result was again independently shown in [14]. Vitányi [19] showed that general membership for PD0L systems and infiniteness for D0L systems can be decided deterministically in polynomial time.

In this paper we establish bounds on the complexity of the emptiness and finiteness questions for each of the classes ET0L, E0L, and their deterministic and propagating versions, as well as bounds on the general membership problem (that is, to determine whether $x \in L(G)$, if given both x and G as data). In each case an upper bound is established by exhibiting an efficient algorithm to solve the problem, and analyzing its time or space requirements. The lower bounds are established by reducibility arguments. In most cases the problems are complete for $\mathcal{NP}$ or PSPACE. Tight bounds are established for space requirements of many problems.

The previously published results concerning L systems, with the exception of [19], establish the complexity of deciding membership in L(G) for fixed G. The general membership problem can be significantly more complex. The most extreme case is the EDT0L systems - each L(G) may be recognized in log n space, but deciding whether $x \in L(G)$ if both x and G are given as inputs requires essentially linear space (both by nondeterministic algorithms).

In general it appears that problems about propagating systems are of the same complexity as those for non-propagating systems, although some upper bound constructions are complicated by the presence of λ-productions, and lower bound constructions are complicated by their absence.

In section 2 we briefly review the relevant terminology about complexity and L systems. In section 3 we introduce some definitions and lemmas which will be used throughout the remainder of the paper. These will be used to efficiently simulate derivations in which large numbers of symbols are generated and then subsequently erased. Most of the complexity bounds for the membership question are established in section 4, the exceptions being several $\mathcal{NP}$ lower bounds which are corollary to results of section 5, where bounds on the (non) emptiness and infiniteness problems are established. Each section begins with bounds for the most general L systems and progresses towards the simpler versions. Finally, section 6 contains a summary of results, in the form of a table. The reader may wish to consult this table while working through sections 4 and 5, for the sake of perspective.

2. NOTATION AND TERMINOLOGY

We recapitulate here the definitions from computational complexity and L systems theory which are relevant to our results. The reader may find more leisurely and motivated descriptions in [1] and [4].

Complexity definitions

The classes of problems solvable within limited time or space bounds are defined as follows:

DSPACE(S(n)) = { L | L is accepted by some deterministic offline Turing machine which operates within space S(n) on all inputs of length n }

NSPACE(S(n)) is defined analogously for nondeterministic Turing machines, and DTIME(S(n)), NTIME(S(n)) are defined similarly for the time measure.

The important classes $\mathcal{P}$, $\mathcal{NP}$ and PSPACE are defined by

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

$$\mathcal{NP} = \bigcup_{k=1}^{\infty} \mathrm{NTIME}(n^k)$$

$$\mathrm{PSPACE} = \bigcup_{k=1}^{\infty} \mathrm{DSPACE}(n^k) = \bigcup_{k=1}^{\infty} \mathrm{NSPACE}(n^k)$$

Let $L, M \subseteq \Sigma^*$. We say that L is reducible to M just in case there is a polynomial-time-computable function f such that for all x, $x \in L$ if and only if $f(x) \in M$. We say that M is $\mathcal{NP}$-hard if any set in $\mathcal{NP}$ is reducible to M. M is complete for $\mathcal{NP}$ if M is $\mathcal{NP}$-hard, and M is in $\mathcal{NP}$. To show that a problem M is $\mathcal{NP}$-hard it suffices to show that some other problem already known to be $\mathcal{NP}$-hard is reducible to M (this follows since reducibility is transitive). Hardness and completeness can also be defined for PSPACE, in the same way.

L system definitions

Definition An ETOL system is a construct $G = (V, \mathcal{P}, w, \Sigma)$ where

a) V is a finite alphabet.

b) $w \in V^+$ is a word called the axiom.

c) $\mathcal{P}$ is a finite set of tables of which each element T is a finite binary relation, $T \subseteq V \times V^*$, such that for every symbol a from V there exists α in V^* such that $\langle a, \alpha \rangle$ is in T. $\langle a, \alpha \rangle \in T$ is usually written $a \rightarrow_T \alpha$ or $a \rightarrow \alpha$ if it is clear from the context which table T is meant.

d) $\Sigma \subseteq V$ is called the target alphabet.

If for every T in $\mathcal{P}$ and for every a in V there exists exactly one α in V^* such that $a \rightarrow \alpha$ then G is called deterministic. If for every T in $\mathcal{P}$ we have that $T \subset V \times V^+$ then G is called propagating. If there is only one table in G then G is called an E0L system and we write $G = (V, P, w, \Sigma)$ instead of $G = (V, \{P\}, w, \Sigma)$.

We will use the letters P and D to denote the deterministic and propagating restrictions respectively. Thus e.g., EPD0L denotes a propagating and deterministic E0L system.

Definition Let $G = (V, \mathcal{P}, w, \Sigma)$ be an ETOL system.

a) Let $x = a_1 a_2 \ldots a_k$, $k \geq 0$, $a_1, a_2, \ldots, a_k \in V$. Let T be a table in $\mathcal{P}$, and let $y \in V^*$. We write $x \Rightarrow_T y$ if there exist $\alpha_1, \alpha_2, \ldots, \alpha_k$

in V^* such that $a_i \longrightarrow_T \alpha_i$ for $1 \leq i \leq k$ and $y = \alpha_1\alpha_2\ldots\alpha_k$. We write $x \underset{G}{\Rightarrow} y$ if $x \Rightarrow_T y$ for a table T in $\mathcal{P}$. G may be omitted if clear from context.

b) $\underset{G}{\Rightarrow}^*$ denotes as usual the transitive and reflexive closure of the binary relation $\underset{G}{\Rightarrow}$ on $V^* \times V^*$. Again G may be omitted.

c) The language of G, denoted L(G) is defined by $L(G) = \{x \in \Sigma^* \mid w \Rightarrow^* x\}$.

Notation

Throughout this paper p will denote the cardinality of V. If $x \in V^*$ then Alph(x) denotes the minimal alphabet $A \subseteq V$ such that $x \in A^*$. A derivation in an ET0L system $G = (V, \mathcal{P}, w, S)$ is a sequence of words $\alpha_1, \alpha_2, \ldots, \alpha_k$ in V^* such that $\alpha_1 = w$ and $\alpha_i \Rightarrow \alpha_{i+1}$ for $1 \leq i < k$. A derivation is written $\alpha_1 \Rightarrow \alpha_2 \Rightarrow \ldots \Rightarrow \alpha_k$. An occurrence of a symbol a in α_i is productive with respect to the derivation if it derives a nonempty subword of α_k.

We call a symbol $a \in V$ dying if $a \Rightarrow^* \lambda$. The set of dying symbols, $\{a \in V \mid a \Rightarrow^* \lambda\}$ will be denoted by $\underline{V_d}$. Note that if $a \Rightarrow^* \lambda$ then $a \Rightarrow^p \lambda$. All nonproductive symbols are dying, but a dying symbol might occur as a productive letter in a derivation. Whenever an L system is an input to an algorithm, it will be encoded as a word in the following manner.

An alphabet $V = \{v_1, v_2, \ldots, v_p\}$ is represented by the word $\overline{V} = [V\overline{1}; V\overline{2}; \ldots; V\overline{p}]$ where $\overline{i}$ is the binary representation of i. This can naturally be extended to words and productions. We will encode an ET0L system $G = (V, \mathcal{P}, w, \Sigma)$ as the word $\overline{G} = [\overline{V}; \overline{\mathcal{P}}; \overline{w}; \overline{\Sigma}]$ over the alphabet $\{V, 0, 1, [, ;,], \longrightarrow\}$. Note that $p \log p = O(|\overline{G}|)$.

The problems we discuss may all be represented as membership questions for the following sets. Let C denote any L system class, and let $\overline{x}$ denote the obvious coding of the word $x \in \Sigma^*$.

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

Note that an upper complexity bound for a problem is automatically an upper bound for a subproblem. Thus, for example, the upper bound on MEMBER^{E0L} also applies to MEMBER^{EP0L} and MEMBER^{ED0L}. Similarly, a lower bound for a subproblem is also a lower bound for the general problem.

3. DERIVATIONS WITHOUT DYING LETTERS

The upper bound constructions are complicated considerably by the need to handle systems containing λ-productions. For example an ED0L system may in n steps derive strings containing more than 2^n symbols, all of which are then erased in one step by applying a single λ-production. This causes straightforward simulation of even short derivations to use excessive amounts of time and space.

The following definitions and lemmas will be used to provide time- or space-efficient simulation of L system derivations.

Let $\alpha_1 \Rightarrow \alpha_2 \Rightarrow \ldots \Rightarrow \alpha_k$ be a derivation in an ET0L system. Such a derivation will be simulated by storing for each α_i a pair (β, B), where B is the set of nonproductive symbols occurring in α_i, and β is α_i with the nonproductive symbols removed. Following are some definitions which will be helpful in explaining just how this can be done.

a) For $A, B \subseteq V$ we define $A \rightsquigarrow_T B$ if and only if there are $u, v \in V^*$ such that $u \Rightarrow_T v$, $A = \mathrm{ALPH}(u)$ and $B = \mathrm{Alph}(v)$. We define $A \rightsquigarrow B$ if $A \rightsquigarrow_T B$ for some T in $\mathcal{P}$. $A \rightsquigarrow^* B$, $A \rightsquigarrow^+ B$ and $A \rightsquigarrow^k B$ are defined in the usual way.

b) For $\alpha, \beta \in V^*$ and $A \subseteq V$ we define $\alpha <^A \beta$ if and only if we can write $\alpha = a_1 a_2 \ldots a_k$ and $\beta = x_0 a_1 x_1 a_2 \ldots a_k x_{k+1}$ where $a_i \in V$ and $x_j \in A^*$ for $1 \leq i \leq k$, $0 \leq j \leq k+1$.
Note that $\alpha <^A \alpha$ for any A, α, and $\alpha <^{\emptyset} \beta$ if and only if $\alpha = \beta$.

c) For $\alpha, \beta \in V^*$, $A, B \subseteq V$ and table T in $\mathcal{P}$ we define $(\alpha, A) \Rightarrow_T (\beta, B)$ if and only if

I) we can write $\alpha = a_1 a_2 \ldots a_k$, $\beta = \beta_1 \beta_2 \ldots \beta_k$ where for each $i = 1, 2, \ldots, k$ there is a production $a_i \to \gamma_i$ in T such that $\lambda \neq \beta_i <^B \gamma_i$

II) $A \overset{}{\underset{T}{\rightsquigarrow}} C$ for some $C \subseteq B$

The relations $\Rightarrow$, $\Rightarrow^+$, $\Rightarrow^*$, $\Rightarrow^k$ are defined in the usual way. Note that I implies $|\alpha| \leq |\beta|$. In the E0L and ED0L cases we omit the T, since there is only one table.

The following lemmas show that the pairs (α, A) may be used to faithfully simulate derivations in an ET0L system. Let the system be $G = (V, \mathcal{P}, w, \Sigma)$. The goal is to show that for each derivation $w = w_0 \Rightarrow w_1 \Rightarrow \ldots \Rightarrow w_k \in \Sigma^*$ of G there is a corresponding derivation $(w_0', A_0) \Rightarrow (w_1', A_1) \Rightarrow \ldots \Rightarrow (w_{k-1}', A_{k-1}) \Rightarrow (w_k, \emptyset)$, and conversely. At each step w_i' will consist of the productive letters in w_i, and A_i will contain all letters in w_i which yield λ in this derivation.

Lemma 1

Let $\alpha \Rightarrow_T \beta$ and $B \subseteq V$ for some $\alpha, \beta \in V^*$ and $T \in \mathcal{P}$. Let $A = \{a \mid a \to \gamma \in T \text{ for some } \gamma \in B^*\}$. Then for all β' with $\beta' <^B \beta$ there exists an $\alpha' \in V^*$ such that

$$\alpha' <^A \alpha \text{ and } (\alpha', A) \Rightarrow_T (\beta', B)$$

Proof

Let $\alpha = a_1 a_2 \ldots a_k$ and $\beta = \beta_1 \beta_2 \ldots \beta_k$ where $a_i \to \beta_i \in T$ for $i = 1, \ldots, k$. Decompose β' into $\beta' = \beta_1' \beta_2' \ldots \beta_k'$ so $\beta_i' <^B \beta_i$ for $i = 1, \ldots, k$. Let α' be the word obtained from α by removing each a_i with $\beta_i' = \lambda$.

Now $\lambda = \beta_i' <^B \beta_i$ implies $\beta_i \in B^*$, so that $a_i \in A$; consequently $\alpha' <^A \alpha$. Further, $(a_i, A) \Rightarrow_T (\beta_i', B)$ for each a_i in α', hence $(\alpha', A) \Rightarrow_T (\beta', B)$.

$\square$

Lemma 2

Let $\alpha' <^A \alpha$ and $(\alpha', A) \Rightarrow_T (\beta', B)$ for some $\alpha, \alpha', \beta' \in V^*$, $T \in \mathcal{P}$ and $A, B \subseteq V$. Then there exists a $\beta \in V^*$ such that $\alpha \Rightarrow_T \beta$ and $\beta' <^B \beta$.

Proof

Let $\alpha' = a_1 \ldots a_k$ and $\alpha = x_0 a_1 x_1 \ldots a_k x_k$ where $a_i \in V$ and $x_j \in A^*$. For each i let $a_i \to \gamma_i$ be a production in T such that $\beta' = \beta_1 \beta_2 \ldots \beta_k$ and $\beta_i <^B \gamma_i$.

Since $A \rightsquigarrow_T C$ for $C \subseteq B$, there must exist strings $v_i \in B^*$ such that $x_i \Rightarrow_T v_i$. We now choose $\beta = v_0 \gamma_1 v_1 \gamma_2 \ldots \gamma_k v_k$. Clearly $\alpha \Rightarrow_T \beta$, and $\beta' = \beta_1 \beta_2 \ldots \beta_k <^B \gamma_1 \gamma_2 \ldots \gamma_k <^B v_0 \gamma_1 v_1 \ldots \gamma_k v_k = \beta$. $\square$

Lemma 3

Let $G = (V, \mathcal{P}, w, \Sigma)$ be an ET0L (E0L, ED0L) system, and $\alpha, \beta \in V^*$. Then $\alpha \Rightarrow^* \beta$ if and only if $(\alpha', A) \Rightarrow^* (\beta, \emptyset)$ for some $A \subseteq V$ and some α' with $\alpha' <^A \alpha$. Note that $A \subseteq V_d$.

Proof

Easy from the two preceding lemmas. $\square$

4. THE MEMBERSHIP PROBLEM

We first establish upper and lower bounds for ET0L membership which are very close to NSPACE(log n), and see that the same bounds apply to various restrictions of the ET0L systems and to some emptiness and infiniteness problems (Theorem 4 through Corollary 7). We then show that E0L membership is in $\mathcal{NP}$ (Theorem 9), ED0L membership is in $\mathcal{P}$ (Theorem 12), and that ED0L membership requires at least logarithmic space (Theorem 13). A lower bound of $\mathcal{NP}$ for E0L membership will result from Corollary 21 of section 5.

Theorem 4

$\mathrm{MEMBER}^{\mathrm{ET0L}} \in \mathrm{NSPACE}(n \log n)$.

Proof

Let $G = (V, \mathcal{P}, w, \Sigma)$ be an ET0L system. By Lemma 3, $x \in L(G)$ if and only if $(w', A) \Rightarrow^* (x, \emptyset)$ for some $A \subseteq V$ and $w' \in V^*$ such that $w' <^A w$. To test $x \in L(G)$ it suffices to guess A and w', and (nondeterministically) generate a sequence

$$(w', A) = (w_0, A_0) \Rightarrow (w_1, A_1) \Rightarrow \ldots \Rightarrow (w_k, A_k),$$

accepting x just in case a pair $(w_k, A_k) = (x, \emptyset)$ is obtained. Note that $|w_0| \leq |w_1| \leq \ldots \leq |w_k|$, and that only two consecutive (w_i, A_i) pairs need be stored at any time.

Recalling that n is the length of $<\bar{G}, \bar{x}>$, we see that this can be done in space n log n by storing A_i as a bit vector and w_i directly. The log n factor comes from the need to encode each symbol v_i of V as the string $V\bar{i}$. □

Corollary 5

MEMBER ^{EDT0L}, MEMBER^{EPT0L} and MEMBER^{EPDT0L} are in NSPACE(n log n).

Theorem 6

MEMBER^{EPDT0L} $\notin$ 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 EPDT0L 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, $|\overline{G}| = O(n \log n)$. Consequently L(Z) is reducible to L(G). Now suppose $\text{MEMBER}^{\text{EPDT0L}} \in \text{NSPACE}(n^{1-\epsilon})$ for some ϵ, $0 < \epsilon < 1$. By [11] there exists $L \in \text{NSPACE}(n) - \text{NSPACE}(n^{1-\epsilon/2})$. Let Z be chosen to recognize L in space n. Then we can decide whether an arbitrary $x \in \Sigma^*$ is in L by first constructing G as above, letting $n = |x|$ and $y = 0^{n+3}$, and then deciding whether $\langle \overline{G}, \overline{y} \rangle \in \text{MEMBER}^{\text{EPDT0L}}$. Now $|\langle \overline{G}, \overline{y} \rangle| = O(n \log n)$, so this process works in space $O((n \log n)^{1-\epsilon}) = O(n^{1-\epsilon}(\log n)^{1-\epsilon}) \subseteq O(n^{1-\epsilon} n^{\epsilon/2}) = O(n^{1-\epsilon/2})$, a contradiction. □

__Corollary 7__

None of the following is in $\text{NSPACE}(n^{1-\epsilon})$ for any $\epsilon > 0$: $\text{MEMBER}^{\text{EDT0L}}$, $\text{NONEMPTY}^{\text{EDT0L}}$, $\text{INFINITE}^{\text{EDT0L}}$, $\text{MEMBER}^{\text{ET0L}}$, $\text{NONEMPTY}^{\text{ET0L}}$, $\text{INFINITE}^{\text{ET0L}}$, or their restrictions to propagating systems.

__Proof__

The construction is easily modified so that L(G) is infinite if and only if Z accepts x, giving the result for $\text{INFINITE}^{\text{EDT0L}}$. The other results are immediate. □

__Remark__

The following somewhat simpler construction yields the same results except for $\text{MEMBER}^{\text{EPDT0L}}$ and $\text{MEMBER}^{\text{EPT0L}}$, and may be interesting in its own right. Given a nondeterministic finite automaton

$M = (K, \Sigma, \delta, q_0, F)$, define the EDT0L-system $G = (K, \{P_a \mid a \in \Sigma\}, q_0, K-F)$, where for each $a \in \Sigma$,

$$P_a = \{ p \to q_1 q_2 \ldots q_k \mid \delta(p, a) = \{q_1, q_2, \ldots, q_k\}\}$$

it is easily seen that $L(G)$ is nonempty just in case $L(M) \neq \Sigma^*$. The $NSPACE(n^{1-\epsilon})$ lower bound obtains from the fact that $\{R \mid L(R) \neq \{0, 1\}^*$ and R is a regular expression$\}$ is known to be in NSPACE(n) but in no smaller space complexity class [10]. Given any R, a nondeterministic finite automaton is easily construction to accept L(R), so an EDT0L system G can be built as just described satisfying $L(R) \neq \{0, 1\}^*$ just in case $L(G) \neq \emptyset$. If λ-productions are allowed it is easy to modify G so $L(G) = \{\lambda\}$ just in case $L(G) \neq \emptyset$, giving the result for $MEMBER^{EDT0L}$.

We now show that $MEMBER^{E0L}$ is in $\mathcal{NP}$. Step-by-step simulation would be inadequate to show this for two reasons: the problems with dying letters mentioned in section 3; and the fact that the shortest derivation of x in L(G) may be of length exponential in $|\langle \overline{G}, \overline{x} \rangle|$. Recall that V_d is the set of all dying letters.

<u>Lemma 8</u>

Let $G = (V, P, w, \Sigma)$ be an E0L system and let $\alpha, \beta \in V^*$ with $|\alpha| = |\beta|$. Then the relation $(\alpha, V_d) \Rightarrow^* (\beta, V_d)$ can be nondeterministically decided in time polynomial in $|\langle \overline{G}, \overline{x} \rangle|$.

<u>Proof</u>

Let $\alpha = a_1 a_2 \ldots a_k$ and $\beta = b_1 b_2 \ldots b_k$ (each $a_i, b_i \in V$) and let $r > 0$. Then the following statements are equivalent:

(1) $(\alpha, V_d) \Rightarrow^r (\beta, V_d)$

(2) $(a_i, V_d) \Rightarrow^r (b_i, V_d)$ for each $i = 1, 2, \ldots, k$.

(3) $a_i \Rightarrow^r x_i b_i y_i$ for some $x_i, y_i \in V_d{}^*$ and each $i = 1, 2, \ldots, k$.

We decide (3) by forming a $V \times V$ connection matrix M, where for each $a, b \in V$

$$m(a,b) = \begin{cases} 1 & \text{if } a \to xby \text{ is in P for some } x, y \in V_d{}^* \\ 0 & \text{otherwise.} \end{cases}$$

Then $M^r(a,b)$ will be 1 exactly when $a \Rightarrow^r xby$ for some $x, y \in V_d{}^*$ (where M^r is the r'th power of M, using and-or matrix multiplication). There are only 2^{p^2} distinct connection matrices, so it suffices to guess an $r \leq 2^{p^2}$, and test condition (3) for this r.

M^r may be obtained by computing $M^1, M^2, M^4, \ldots, M^{2^{p^2}}$ by repeated squaring, and multiplying those matrices which correspond to ones in the binary representation of r. Clearly each of these steps may be done in time polynomial in $|<\overline{G}, \overline{x}>|$. □

<u>Theorem 9</u>

$\text{MEMBER}^{\text{E0L}} \in \mathcal{NP}$.

<u>Proof</u>

Let $G = (V, P, w, \Sigma)$ be an E0L system. According to Lemma 3, $x \in L(G)$ if and only if $x \in \Sigma^*$ and $(w', A) \Rightarrow^* (x, \emptyset)$ for some $A \subseteq V$ and $w' <^A w$. Observe that $A \subseteq V_d$, so V_d could be used instead of A.

Recalling that $(\alpha, A) \Rightarrow (\beta, B)$ implies $|\alpha| \leq |\beta|$, we see that $(w', V_d) \Rightarrow^* (x, \emptyset)$ if and only if

1) $(w', V_d) \Rightarrow^r (x, \emptyset)$ for some $r < p$; or

2) there exist k $(0 \leq k \leq |x|)$ and strings $\alpha_i, \beta_i \in V^*$ $(1 \leq i \leq k)$ such that $|\alpha_1| = |\beta_1| < |\alpha_1| = |\beta_2| < \ldots < |\alpha_k| = |\beta_k|$ and

$$(w', V_d) \Rightarrow (\alpha_1, V_d) \Rightarrow^* (\beta_1, V_d) \Rightarrow (\alpha_2, V_d) \Rightarrow^* (\beta_2, V_d) \Rightarrow \ldots$$
$$\ldots \Rightarrow (\alpha_k, V_d) \Rightarrow^* (\beta_k, V_d) \Rightarrow^p (x, \emptyset)$$

Following is a decision procedure based on these remarks.

choose $\alpha \overset{V_d}{<} w$;

<u>if</u> $(\alpha, V_d) \Rightarrow^r (x, \emptyset)$ for some $r < p$ <u>then</u> accept;

<u>for</u> $i = 1, 2, \ldots, |x|$ <u>do</u>

<u>begin</u> choose β so that $|\alpha| = |\beta|$ and $(\alpha, V_d) \Rightarrow^* (\beta, V_d)$;

    <u>if</u> $(\beta, V_d) \Rightarrow^p (x, \emptyset)$ <u>then</u> accept;

    choose α so that $(\beta, V_d) \Rightarrow (\alpha, V_d)$

<u>end</u>

By Lemma 8 we see that this nondeterministic procedure runs in polynomial time. □

An $\mathcal{NP}$ lower bound for $\text{MEMBER}^{\text{E0L}}$ will appear in Corollary 21, following from the same bound for $\text{NONEMPTY}^{\text{ED0L}}$. We now consider ED0L membership. Previous work includes "feasible" algorithms (Vitányi [19]) for the general membership and finiteness problems for D0L systens, including

<u>Theorem 10</u>

$\text{MEMBER}^{\text{EPD0L}} \in \mathcal{P}$.

His algorithm is based on the following facts, which we shall also use. Suppose $w \Rightarrow^* x$ by an ED0L system $G = (V, P, w, \Sigma)$. Then

(a) all steps after the first $p|x|$ can only use productions $a \rightarrow \alpha$ in which α has at most one nondying letter;

(b) consequently a propagating system can only use productions of the form $a \rightarrow b$ $(a, b \in V)$ after $p|x|$ steps;

(c) the derivation is reversible after the first $p|x| + p$ steps, in the following sense:
If $w \Rightarrow^r a_1 \ldots a_k \Rightarrow b_1 \ldots b_k$ and $r \geq p|x| + p$, then for each $i = 1, 2, \ldots, k$, a_i is the unique symbol such that $b_i \Rightarrow^+ a_i \Rightarrow b_i$.

The algorithms of [19] do <u>not</u> yield polynomial time algorithms for non-propagating systems, since they involve a direct simulation of G's derivation for $p(|x| - |w| + 1)$ steps. This derivation can produce intermediate strings whose length is exponential in p if G has many dying symbols. Our algorithm for $\text{MEMBER}^{\text{ED0L}}$ involves a more efficient way to simulate short derivations, and an application of the Chinese remainder theorem as used in [19].

<u>Lemma 11</u>

Let $G = (V, P, w, \Sigma)$ be an ED0L system and $x \in \Sigma^*$. The relation "$\alpha \Rightarrow^* x$ in k or fewer steps" can be decided in time bounded by a polynomial function of $|<\overline{G}, \overline{x}>|$ and k.

Proof

It is sufficient to show that the following functions $a(i)$ (where $0 \leq i \leq k$ and $a \in V$) can be computed in polynomial time:

$$a(i) = \begin{cases} \beta & \text{if } a \Rightarrow^{i} \beta \text{ and } \beta \text{ is a subword of } x \\ \# & \text{otherwise.} \end{cases}$$

Let $a \in V$ and $0 \leq i \leq k$, and let the unique a-production in P be $a \to b_1 b_2 \ldots b_r$. It is immediate that

$$a(i) = \begin{cases} a & \text{if } i = 0 \text{ and } a \text{ is a subword of } x; \\ b_1(i-1)b_2(i-1)\ldots b_r(i-1) & \text{if } i \neq 0 \text{ and } b_1(i-1)\ldots b_r(i-1) \\ & \text{is a subword of } x; \\ \# & \text{otherwise.} \end{cases}$$

Thus the $a(i)$'s may be computed in the order $i = 0, 1, \ldots, k$; the time bound is immediate, since only subwords of x are stored. A similar technique was used in [7]. □

Theorem 12

$\text{MEMBER}^{\text{ED0L}} \in \mathcal{P}$.

Proof

Let $G = (V, P, w, \Sigma)$ be an ED0L system. Assume $w \Rightarrow^{p|x|} v \Rightarrow^{*} z \Rightarrow^{p} x$. Because of fact (a) above the number of nondying symbols in v, z and x are the same. Let w', v', z', and x' be the words we obtained by removing all the dying letters from w, v, z and x. Then $(w', V_d) \Rightarrow^{p|x|} (v', V_d) \Rightarrow^{*} (z', V_d) \Rightarrow^{p} (x', V_d)$. Since all dying symbols in an ED0L system must derive the empty string in p or fewer steps, we actually have that if $x \in V^{*}$ then:

$$w \Rightarrow^{p|x|} \quad v \Rightarrow^{*} z \Rightarrow^{p} x \text{ for some } v, z \in V^{*}$$

if and only if

$$(w', V_d) \Rightarrow^{p|x|} \quad (v', V_d) \Rightarrow^{*} (z', V_d) \Rightarrow^{p} (x', V_d),$$

where $w' <^{V_d} w$, and $z' \Rightarrow^{p} x$ for some $w', v', z', x' \in (V-V_d)^{*}$.

Consider the following algorithm:

(1) <u>If</u> $w \Rightarrow^{r} x$ for some $r < p|x|+p$ <u>then</u> accept;

(2) Find $w' \in (V-V_d)^{*}$ so $w' <^{V_d} w$;

(3) Find $x' \in (V-V_d)^{*}$ so $x' <^{V_d} x$;

(4) Find $v' \in (V-V_d)^{*}$ so $(w', V_d) \Rightarrow^{p|x|} \quad (v', V_d)$;

(5) Find $z' \in (V-V_d)^{*}$ so $(z', V_d) \Rightarrow^{p} (x', V_d)$;

(6) <u>If</u> $z' \Rightarrow^{p} x$ and $(v', V_d) \Rightarrow^{*} (z', V_d)$ <u>then</u> accept;

Correctness of the algorithm follows from the remarks above. Steps 1, 4 and the first part of 6 can be done in polynomial time by Lemma 11. Steps 2 and 3 are easily done in polynomial time. From above it follows that step 5 can be done in polynomial time. $(v', V_d) \Rightarrow^{*} (z', V_d)$ in step 6 can be tested in polynomial time using the Chinese remainder theorem as in [19], page 82. Note that $|v'| = |z'|$ if the relation holds. □

<u>Theorem 13</u>

There is an EPD0L system G such that if L(G) is in DSPACE(S(n)), then

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

<u>Proof</u>

$L = \{a^n bc^n \mid n \geq 0\}$ is an EPD0L language. By Alt and Mehlhorn [2], if L is in DSPACE(S(n)), then S must satisfy the condition stated.

$\square$

5. THE NONEMPTINESS AND INFINITENESS PROBLEMS

We determine the complexity of nonemptiness rather than emptiness, since sharper bounds may be obtained. In Theorems 14 through 18 we show that for systems with tables these problems have essentially NSPACE(n) complexity. In Theorem 19 we see that E0L nonemptiness can be decided deterministically in space n, but infiniteness seems to require nondeterminism. $\mathcal{NP}$ lower bounds on these problems, and $\mathcal{NP}$ completeness of the same problems for ED0L systems, are proved in Lemma 20 through Theorem 23.

Theorem 14

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

Proof

If $G = (V, \mathcal{P}, w, \Sigma)$ is an ET0L system, then clearly $L(G) \neq \emptyset$ if and only if there is a sequence $A_1 \rightsquigarrow A_2 \rightsquigarrow \ldots \rightsquigarrow A_k$ with $A_1 = \mathrm{Alph}(w)$ and $A_k \subseteq \Sigma$. Such a sequence may be generated nondeterministically one A_i at a time, storing only two consecutive A_i's as bit vectors of size p per step. $\square$

Corollary 15

NONEMPTY$^{\mathrm{EDT0L}}$ $\in$ NSPACE(n).

By Corollary 7, NONEMPTY$^{\mathrm{EDT0L}}$ $\notin$ NSPACE($n^{1-\epsilon}$) for any $\epsilon > 0$.

Theorem 16

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

Proof

Let $G = (V, \mathcal{P}, w, \Sigma)$ be an ETOL system. Define for $C, C', B, B' \subseteq V$:

$(C,B) \Rightarrow (C',B')$ if and only if there are

$\alpha, \beta \in V^*$ such that $C = \text{Alph}(\alpha)$, $C' = \text{Alph}(\beta)$ and

$(\alpha, B) \Rightarrow (\beta, B')$.

$(C,B) \overset{<}{\Rightarrow} (C',B')$ if and only if

$(C,B) \Rightarrow (C',B')$ as above with $|\alpha| < |\beta|$.

It is easily seen that the following three statements are equivalent:

(1) $L(G)$ is infinite.

(2) $(w', A) \Rightarrow^* (\beta, B) \Rightarrow^* (\beta', B) \Rightarrow^* (x, \emptyset)$ for some $w', \beta, \beta' \in V^*$, $x \in \Sigma^*$, $A, B \subseteq V$, $w' <^A w$, $|\beta| < |\beta'|$, and $\text{Alph}(\beta) = \text{Alph}(\beta')$.

(3) $(C_0, A) \Rightarrow^* (C,B) \Rightarrow^* (C_1, B') \overset{<}{\Rightarrow} (C_2, B'') \Rightarrow^* (C,B) \Rightarrow^* (C_3, \emptyset)$ for some $A, B, B', B'', C_0, C_1, C_2, C_3, C \subseteq V$, $C_0 \subseteq \text{Alph}(w)$, and $C_3 \subseteq \Sigma$.

Construction of an algorithm based on (3) above is now straightforward. The C's and B's can be stored as vectors of p bits, and the relations $\Rightarrow$ and $\overset{<}{\Rightarrow}$ can be easily tested in p bits. □

Corollary 17

$\text{INFINITE}^{\text{EDTOL}}, \text{INFINITE}^{\text{EOL}} \in \text{NSPACE}(n)$.

Theorem 18

The membership, emptiness and infiniteness problems for EPDTOL, EDTOL, EPTOL, and ETOL systems are PSPACE complete.

Proof

We have just seen that each is recognizable in polynomial space. It is well known that there is a context-sensitive language L which is PSPACE hard [1]. By Theorem 6 L is reducible to L(G) for an EPDT0L system G, so $\mathrm{MEMBER}^{\mathrm{EPDT0L}}$ and the others are all PSPACE hard. □

Theorem 19

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

Proof

Let $G = (V, P, w, \Sigma)$ be an E0L system. For $A \subseteq V$, define

$$\mathrm{Pred}(A) = \bigcup \{B \mid B \rightsquigarrow A\}.$$

Thus $\alpha \Rightarrow \beta$ for some $\beta \in A^*$ if and only if $\mathrm{Alph}(\alpha) \subseteq \mathrm{Pred}(A)$. Consequently $w \Rightarrow^r x$ for some $x \in \Sigma^*$ if and only if $\mathrm{Alph}(w) \subseteq \mathrm{Pred}^r(\Sigma)$. Each $\mathrm{Pred}^s(\Sigma)$ is a subset of V, so if w derives any strings in Σ^* it must do so for some $r \le 2^p$. Combining these observations we get the following algorithm, which can clearly be implemented in space p.

$A := \Sigma$;

<u>for</u> $r := 1, 2, \ldots, 2^p+1$ <u>do</u>

    <u>if</u> $\mathrm{Alph}(w) \subseteq A$ <u>then</u> accept <u>else</u> $A := \mathrm{Pred}(A)$;

reject

□

We now proceed to show that the infiniteness and nonemptiness problems for ED0L systems are $\mathcal{NP}$ complete.

Lemma 20

$\text{NONEMPTY}^{\text{EPD0L}}$ is $\mathcal{NP}$-hard.

Proof

Stockmeyer and Meyer show in [12] how to build from any propositional formula $\mathfrak{F}$ a regular expression R of the form

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

such that $0^*-L(R)$ is infinite if $\mathfrak{F}$ is satisfiable, and $0^* = L(R)$ if $\mathfrak{F}$ is unsatisfiable.

Construct an EPD0L system $G = (V, P, z_1^0 \ldots z_r^0, \Sigma)$ where $V = \{z_i^j \mid 1 \leq i \leq r,\ 0 \leq j \leq p_i+q_i-1\}$, $\Sigma = V - \{z_1^{p_1}, z_2^{p_2}, \ldots, z_r^{p_r}\}$ and P consists of the productions $(i = 1, \ldots, r)$: $z_i^j \to z_i^{j+1}$ for $j = 0, \ldots, p_i+q_i-2$ and $z_i^{p_i+q_i-1} \to z_i^{p_i}$. Now $L(R) \neq 0^*$ iff $L(G) \neq \emptyset$ iff $\overline{G} \in \text{NONEMPTY}^{\text{EPD0L}}$. Clearly $\overline{G}$ can be constructed from R in polynomial time, so $\text{NONEMPTY}^{\text{EPD0L}}$ is $\mathcal{NP}$-hard. □

Corollary 21

The following problems are $\mathcal{NP}$-hard: $\text{NONEMPTY}^{\text{ED0L}}$, $\text{NONEMPTY}^{\text{E0L}}$, $\text{INFINITE}^{\text{ED0L}}$, $\text{INFINITE}^{\text{E0L}}$, $\text{MEMBER}^{\text{E0L}}$ and their restrictions to propagating systems.

Proof

For $\text{INFINITE}^{\text{EPD0L}}$, obtain a new EPD0L system G' by replacing $z_i^{p_i+q_i-1} \to z_i^{p_i}$ by $z_i^{p_i+q_i-1} \to z_i^{p_i} z_i^{p_i}$ in the above. Now $L(G') = \emptyset$ if $L(R) = 0^*$, and $L(G')$ is infinite if $L(R) \neq 0^*$, so $\text{INFINITE}^{\text{EPD0L}}$ is $\mathcal{NP}$-hard. The other results except for $\text{MEMBER}^{\text{E0L}}$ follow trivially.

Let $G = (V, P, w, \Sigma)$ be any EPD0L system. Construct an EP0L system $G' = (V \cup \{g, 0\}, P', w, \{0\})$ where P' consists of all productions in P, $a \to 0$ for $a \in \Sigma$, $0 \to g$, and $g \to g$. Now $L(G)$ contains words of length i iff $0^i \in L(G')$.

The theorem follows then by observing that in the proof of Theorem 20 $L(R) \neq \emptyset$ iff $L(G) \neq \emptyset$ iff $L(G)$ contains a word of length r. □

Lemma 22

NONEMPTY^{ED0L} and INFINITE^{ED0L} are in $\mathcal{NP}$.

Proof

Let $G = (V, P, w, \Sigma)$ be an ED0L system, and $w = \alpha_0 \Rightarrow \alpha_1 \Rightarrow \ldots$ be its derivation. Clearly $L(G)$ is infinite if and only if $|\alpha_0|, |\alpha_1|, |\alpha_2|, \ldots$ grows infinitely and $\mathrm{Alph}(\alpha_j) \subseteq \Sigma$ for some j with $2^p \leq j < 2^{p+1}$. The infiniteness of $|\alpha_0|, |\alpha_1|, \ldots$ is testable in polynomial time by [19]. To test the j condition we can form a connection matrix M:

$$M(a,b) = \begin{cases} 1 & \text{if } a \to \alpha b \beta \text{ is in } p \text{ for some } \alpha, \beta \in V^* \\ 0 & \text{otherwise.} \end{cases}$$

As in the proof of Lemma 8, we can guess j nondeterministically and compute M^j by repeated squaring. $\mathrm{Alph}(\alpha_j)$ may be read directly from M^j, which completes the proof for INFINITE^{ED0L}. NONEMPTY^{ED0L} is similar but simpler. □

Theorem 23

The infiniteness and nonemptiness problems for ED0L and EPD0L systems are $\mathcal{NP}$ complete.

6. CONCLUSIONS

In general the complexity bounds we have obtained lie between those for the context-free and context-sensitive classes. This might be expected, since every context-free language is E0L and every ET0L language is context-sensitive. For the most part our complexity bounds are tight, in that the lower bounds are near the upper bounds, indicating that our decision algorithms are nearly the best possible. There are three exceptions to this – ED0L membership, with a lower bound of DSPACE(log n) and an upper bound of $\mathcal{P}$; and E0L nonemptiness and infiniteness, with lower bounds of $n\mathcal{P}$ and upper bounds of DSPACE(n) and NSPACE(n) respectively.

The results are indicated in the following table, in which the bounds for context-free and context-sensitive languages are included for comparison. The results of the top and bottom rows and the leftmost column are known, and may be found in [3], [4], [5], [6], [7], [9], [10], [13], [14], [15], [16], [17], [18], [19], and [20].

	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
ET0L, EPT0L	$n\mathcal{P}$	NSPACE(n log n)	NSPACE(n)	NSPACE(n)	UPPER
		NSPACE($n^{1-\epsilon}$)	NSPACE($n^{1-\epsilon}$)	NSPACE($n^{1-\epsilon}$)	LOWER
EDT0L, EPDT0L	$n\mathcal{L}$	NSPACE(n log n)	NSPACE(n)	NSPACE(n)	UPPER
		NSPACE($n^{1-\epsilon}$)	NSPACE($n^{1-\epsilon}$)	NSPACE($n^{1-\epsilon}$)	LOWER
E0L, EP0L	DSPACE($\log^2 n$) DTIME($n^{3.81}$)	$n\mathcal{P}$	DSPACE(n)	NSPACE(n)	UPPER
	$n\mathcal{L}$		$n\mathcal{P}$	$n\mathcal{P}$	LOWER
ED0L, EPD0L	$\mathcal{L}$	$\mathcal{P}$	$n\mathcal{P}$	$n\mathcal{P}$	UPPER
	$\mathcal{L}$	$\mathcal{L}$			LOWER
CONTEXT-FREE	DSPACE($\log^2 n$) DTIME($n^{2.81}$)	$\mathcal{P}$	$\mathcal{P}$	$\mathcal{P}$	UPPER
	$n\mathcal{L}$				LOWER

In this table we use the notations

$$\mathcal{L} = \text{DSPACE}(\log n), \qquad n\mathcal{L} = \text{NSPACE}(\log n)$$

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 $n\mathcal{L}$, $\mathcal{P}$, $n\mathcal{P}$ or NSPACE(n), then P is L-hard.

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

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

Acknowledgement

We wish to thank the referees for helpful suggestions regarding form and presentation.

REFERENCES

[1] Aho, A.V., Hopcroft, J.E., Ullman, J.D., The Design and Analysis of Computer Algorithms, Addison-Wesley, Reading, Mass., 1974.

[2] Alt, H., Mehlhorn, K., Lower bounds for the space complexity of context-free recognition, in Michaelson, S., Milner, R.: Automata, Languages and Programming, (3rd Coll., Edinburgh), University Press, Edinburgh, Scotland, 1976, pp. 338-354.

[3] Harju, T., A polynomial recognition algorithm for the EDT0L languages, Elektron. Inform. Kybernetik 13, 1977, pp. 169-177.

[4] Herman, G.T., Rozenberg, G., Developmental Systems and Languages, North-Holland, Amsterdam, 1975.

[5] Hopcroft, J.E., Ullman, J.D., Formal Languages and their Relation to Automata, Addison-Wesley, Reading, Mass., 1969.

[6] Jones, N.D., Laaser, W.T., Complete problems for deterministic polynomial Time, Theoretical Comp. Sci. 3, 1976, pp. 105-118.

[7] Jones, N.D., Skyum, S., Recognition of deterministic ET0L languages in logarithmic space. Information and Control 35, 1977, pp. 177-181.

[8] Jones, N.D., Skyum, S., Complexity of some problems concerning L systems, in: Salomaa, A., Steinby, M., Automata, Languages and Programming (4th Coll., Turku), Springer Lecture Notes in Computer Science 52, 1977, pp. 301-308.

[9] Lewis, P.M. III, 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, 1965, pp. 191-202.

[10] Meyer, A.R., Stockmeyer, L.J., The Equivalence problem for regular expressions with squaring requires exponential space, Conf. Rec. 13th Annual IEEE Symposium on Switching and Automata Theory, Baltimore, 1972, pp. 125-129.

[11] Seiferas, J., Fischer, M., Meyer, A., Refinements of nondeterministic time and space hierarchies, Conf. Rec. 14th Annual IEEE Symposium on Switching and Automata Theory, Iowa, 1973, pp. 130-137.

[12] Stockmeyer, L.J., Meyer, A.R., Word problems requiring exponential time, Proc. 5th Annual ACM Symposium on Theory of Computing, Austin, May 1973, pp. 1-9.

[13] Sudborough, I.H., A note on tape-bounded complexity classes and linear context-free languages, J. ACM 22, 1975, pp. 499-500.

[14] Sudborough, I.H., The time and tape complexity of developmental languages, in: Salomaa, A., Steinby, M., Automata, Languages and Programming, (4th Coll., Turku), Springer Lecture Notes in Computer Science 52, 1977, pp. 509-523.

[15] Valiant, L.G., General context-free recognition in less than cubic time. J. Computer and Systems Sciences 10, 1975, pp. 308-315.

[16] van Leeuwen, J., Deterministically recognizing E0L languages in time $O(n^{3.81})$. Tech. Rep. Math. Centrum, Amsterdam (1975).

[17] van Leeuwen, J., The membership question for ET0L languages is polynomially complete, Information Processing Letters 3, 1975, pp. 138-143.

[18] van Leeuwen, J., The tape complexity of context-independent developmental languages, J. Computer and Systems Sciences 15, 1975, pp. 203-211.

[19] Vitányi, P.M.B., On the size of DOL languages, in Rozenberg, G. and Salomaa, A. (ed.) L Systems, Springer Lecture Notes in Computer Science 15, 1974, pp. 72-77.

[20] Younger, D.H., Recognition and parsing of context-free languages in Time n^3, Information and Control 10, 1967, pp. 189-208.