ISSN 0105-8517

A NOTE ON THE COMPLEXITY OF GENERAL DOL MEMBERSHIP

By

Neil D. Jones
and
Sven Skyum

Daimi PB-93
January 1979

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

A NOTE ON THE COMPLEXITY OF GENERAL DOL MEMBERSHIP

Neil D. Jones *
Computer Science Department
University of Kansas
Lawrence, Kansas 66045
U.S.A.

Sven Skyum
Department of Computer Science
University of Aarhus
8000 Aarhus C
Denmark

Abstract

In [3] a number of upper and lower bounds were obtained for various problems concerning L systems. In most cases the bounds were rather close ; however, for the general membership problem the upper bound was $\mathcal{P}$, and the lower was deterministic log space. In this note we show that membership can be decided deterministically in $\log^2 n$ space, which makes it very unlikely that the problem is complete for $\mathcal{P}$. We also show that non-membership is as hard as any problem solvable in nondeterministic log n space. Thus both bounds are improved.

* This research was partially supported by National Science Foundation Research Grant, MCS76-80269.

Introduction

Let $G = (V, \delta, a)$ be a DOL system (see Herman and Rozenberg [1]) so V is an alphabet, $a \in V$ a letter from V, and $\delta : V \to V^*$ a mapping. Extending δ to a homomorphism $\delta : V^* \to V^*$, we define

$$L(G) = \{\delta^r(a) \mid r = 0, 1, 2, \ldots\},$$

where δ^r denotes the r-fold composition of δ with itself.

In Sudborough [4] it was shown that for each G, L(G) is in DSPACE(log n) (our notation is from Jones and Skyum [3].) Each set L(G) is a specific membership problem. The general membership problem ($\text{MEMBER}^{\text{DOL}}$) is: Given a DOL system G and a word $v \in V^*$, to determine whether $v \in L(G)$. This problem was first addressed by Vitanyi [5].

In Jones and Skyum [3] it was shown that this problem is in $\mathcal{P}$, and cannot be solved in less than deterministic log space. These two bounds are not particularly close (unless $\mathcal{P}$ = DSPACE(log n) which seems very unlikely.) The purpose of this note is to show that they can be improved. We shall prove:

> Theorem $\text{MEMBER}^{\text{DOL}}$ is in DSPACE($\log^2 n$); and the non-membership problem is hard for NSPACE(log n).

This is nearly the best possible complexity result - it could only be strengthened (aside from a major breakthrough in complexity theory) by showing completeness at one of the bounds. At the lower bound it would be necessary to show that non-membership could be decided nondeterministically in logarithmic space, which seems rather unlikely; and a proof of completeness for DSPACE($\log^2 n$) would require new techniques in complexity theory, as no natural complete problems are presently known for this class.

Proof of the Lower Bound

Define

$$\text{AGAP} = \{\overline{\Gamma} \mid \Gamma \text{ is a digraph with node set } \{1,2,\ldots,n\} \text{ for some } n, \Gamma \text{ has a path from } 1 \text{ to } n, \text{ and } i < j \text{ for each arc } (i,j) \text{ of } \Gamma\}$$

This problem was shown complete for NSPACE($\log n$) in Jones [2]. We now show how to construct from such digraph Γ a DOL system $G = (\{1,\ldots,n\}, \delta, 1)$ such that $\lambda \notin L(G)$ if and only of $\overline{\Gamma}$ is in AGAP. Thus AGAP is reducible to non-membership. Consequently $\text{MEMBER}^{\text{DOL}}$ is in DSPACE($\log n$) if and only if DSPACE($\log n$) = NSPACE($\log n$).

To build G, first add the single arc (n,n) to Γ, obtaining Γ'. Now define $\delta(i) = j_1 \ldots j_k$ to hold just in case $\{j_1, \ldots, j_k\} = \{j \mid (i,j) \text{ is an arc of } \Gamma'\}$ and $j_1 < j_2 < \ldots < j_k$.

Letting Alph(w) equal the smallest alphabet $A \subseteq V$ such that $w \in A^*$, we see that for $r = 1,2,\ldots$ Alph($\delta^r(1)$) is the set of nodes reachable from 1 by path of length exactly r. Now $\overline{\Gamma} \in$ AGAP if and only if Γ' has paths from 1 of arbitrary length if and only if Alph($\delta^r(1)$) $\neq \emptyset$ for all r if and only if $\lambda \notin L(G)$.

Proof of the Upper Bound

For this we give an algorithm which operates in DSPACE($\log^2 n$). Our algorithm is similar to that of Jones and Skyum [3] (which in turn is based on Vitanyi's algorithm [5]), but has several refinements to make it operate in $\log^2 n$ space. Our notation is similar to that of Vitanyi.

Define $b \in V$ to be mortal ($b \in M$) iff $\delta^s(b) = \lambda$ for some s, and monorecursive ($b \in MR$) iff $\delta^s(b) \in M^* bM^*$ for some $s > 0$. The cycle of a monorecursive letter is the least $s > 0$ such that $\delta^s(b) \in M^* bM^*$. Let p be the number of letters in V, and n the number of symbols required to write G and v.

It is easy to see that L(G) is finite iff $\delta^p(a)$ contains only letters in $M \cup MR$, and that if L(G) is infinite, then $v \in L(G)$ iff $v = \delta^r(a)$ for some $r \leq p \mid v \mid$. Our algorithm will have the form "if L(G) infinite then test $v = \delta^r(a)$ for $r = 0, 1, \ldots, p|v|$ else test $v \in L(G)$ by another method". Thus we first show that membership in M and MR, and "$v = \delta^r(a)$" for $r \leq p|v|$ can be determined in DSPACE($\log^2 n$).

Now define the function NUMBER(b, c, s) for $b, c \in V$ and $s \geq 0$ as follows:

$$\text{NUMBER}(b, c, s) = \begin{cases} m & \text{if } \delta^s(b) \text{ contains } m \text{ occurrences} \\ & \text{of } c \text{ and } m \leq n \\ \infty & \text{otherwise} \end{cases}$$

Clearly $b \in M$ iff NUMBER(b, c, p) = 0 for all $c \in V$, and $b \in MR$ iff there is a $0 < i \le p$ such that NUMBER(b, b, i) = 1 and if NUMBER(b, c, i) > 0 then $c \in M \cup \{b\}$.

Setting any partial results greater than n to ∞ and using $0 \cdot \infty = \infty \cdot 0 = 0$ NUMBER can be computed by the following $\log^2(\max(n, s))$ algorithm:

NUMBER(b, c, s) =

  <u>if</u> s = 1 <u>then</u> the number of c's in $\delta(b)$

  <u>else</u> $\sum_{d \in V}$ NUMBER(b, d, $\lceil s/2 \rceil$) * NUMBER(d, c, $\lfloor s/2 \rfloor$).

Thus we can compute M, MR, and CYCLE(b) for all $b \in V$ in $\log^2 n$ space. Further L(G) is infinite iff NUMBER(a, b, p) > 0 for some $b \notin M \cup MR$.

In order to test "$v = \delta^r(a)$" define the functions SYMBOL(b, s, i) for $b \in V$, $0 \le s$, $0 \le i \le n$, and $|w|'$ for $w \in V^*$ as follows:

$$\text{SYMBOL}(b, s, i) = \begin{cases} c & \text{if } c \text{ is the } i\text{-th letter in } \delta^s(b) \\ \# & \text{if } i = 0 \text{ or } i > |\delta^s(b)| \end{cases}$$

$$|w|' = \begin{cases} |w| & \text{if } |w| \le n \\ \infty & \text{if not} \end{cases}$$

$|\delta^s(b)|'$ can easily be computed in $\log^2(n)$ space using NUMBER if s is bounded by a polynomial in n.

Now suppose c = SYMBOL(b, s, i) for some $i \le n$, $s > 0$, and $\delta(b) = b_1 b_2 \ldots b_k$. Then $\delta^s(b) = \delta^{s-1}(b_1)\,\delta^{s-1}(b_2) \ldots \delta^{s-1}(b_k)$, so c = SYMBOL($b_r$, s-1, j) where

$$\sum_{m=1}^{r-1} |\delta^{s-1}(b_m)|' < i \leq \sum_{m=1}^{r} |\delta^{s-1}(b_m)|'$$

and

$$j = i - \sum_{m=1}^{r-1} |\delta^{s-1}(b_m)|' .$$

Repeating, the path leading from a to b = SYMBOL(a, s, i), $i \leq |\delta^s(a)|$ $\delta^s(a)$ may be traversed by the following algorithm:

b : = a;

<u>for</u> h : = s, s−1, ..., 2 <u>do</u>

<u>begin</u>

let $\delta(b) = b_1 b_2 \dots b_k$;

find r such that

$$\sum_{m=1}^{r-1} |\delta^{h-1}(b_m)|' < i \leq \sum_{m=1}^{r} |\delta^{h-1}(b_m)|' ;$$

b : = b_r;

$$i := i - \sum_{m=1}^{r-1} |\delta^{h-1}(b_m)|'$$

<u>end</u>

Using SYMBOL it is easy to test "$v = \delta^r(a)$ for some $r \leq p \cdot |v|$ " in DSPACE($\log^2 n$), which finishes the test if L(G) is infinite.

However, in case L(G) is finite the smallest r such that $v = \delta^r(a)$ may be exponential in n. A different method is needed and the key to this is the following observation, due to Vitanyi [5]:

<u>Observation</u> If L(G) is finite, we can write $\delta^{p}(a) = v_1 a_1 v_2 a_2 \dots a_m v_{m+1}$ where each $a_i \in MR$ and $v_i \in M^*$. If $v = \delta^{r}(a)$ for some $r \geq 2p$, then there exist $\alpha_1, \dots, \alpha_m \in V^*$ such that

a) $v = \alpha_1 \alpha_2 \cdots \alpha_m$

b) for each $j = 1, 2, \dots, m$ there is an r_j such that $p \leq r_j < 2p$ and $\delta^{r_j}(a_j) = \alpha_j$

c) $r_j \equiv r_{j'} \mod \gcd(\mathrm{Cycle}(a_j), \mathrm{Cycle}(a_{j'}))$ for each pair j, j' with $1 \leq j' < j \leq m$.

Conversely, a), b) and c) together imply $v = \delta^{r}(a)$ for some r. In addition $t \not\equiv t' \mod \mathrm{Cycle}(a_j)$ implies $\delta^{t}(a_j) \notin V^* \delta^{t'}(a_j) V^*$; thus α_j is the <u>only</u> prefix of $\alpha_j \cdots \alpha_m$ which is derivable from a_j.

The algorithm testing a, b, and c uses a procedure FIND(i, q, k, r), $0 \leq i, q, k, t \leq n$:

<u>procedure</u> FIND(i, q, k, r); $[v = a_1 a_2 \dots a_{|v|}]$

<u>begin</u>

b : = "the i-th monorecursive letter in $\delta^{p}(a)$" if it exists, otherwise reject;

k : = CYCLE(b);

r : = "the smallest $p \leq t < 2p$ such that $\delta^{t}(b)$ is a prefix of $a_{q+1} a_{q+2} \dots a_{|v|}$" if it exists, otherwise reject;

q : = $q + |\delta^{r}(b)|$; reject if $q > |v|$;

<u>end</u>

Before giving the complete algorithm we will see that FIND can be performed in $\log^2 n$ space.

First to find the i-th monorecursive letter in $\delta^p(a)$ in $\log^2 n$ space, we can simply modify NUMBER and SYMBOL to give the number of non-mortal letters or the j-th nonmortal symbol. Note that $|\delta^p(a)|$ may be exponential in n. r can be found by computing $\delta^t(b)$ for $t = p, 2, \ldots, 2p-1$ one letter at a time and comparing it with v.

The final algorithm for $\text{MEMBER}^{\text{DOL}}$ verifies condition a and b of the observation by calling FIND for $i = 1, 2, \ldots, m$, and verifies condition c by calling FIND for $i' = 1, 2, \ldots, i-1$ in an inner loop for each value of i. The input is a DOL system $G = (V, \delta, a)$ and a word $v \in V^*$.

<u>begin</u>

  <u>if</u> $\delta^p(a) \notin (M \cup MR)^*$ [L(G) is infinite]

  <u>then</u> accept if $v = \delta^r(a)$ for some $r \le p \cdot |v|$ and reject if not

  <u>else</u>

  <u>begin</u> [L(G) is finite]

    accept if $v = \delta^r(a)$ for some $r \le 2p$;

    $q := 0$;

    $m :=$ „the number of monorecursive letters in v";

      [if $v = \delta^r(a)$ for some $r > 2p$ then this equals the number of monorecursive letters in $\delta^p(a)$]

```
for i := 1, 2, ..., m do
begin
      FIND(i, q, k, r) ;
      q' := 0;
      for i' := 1, 2, ..., i-1  do
      begin
          FIND (i', q', k', r') ;
          reject if r ≢ r' mod GCD (k, k')
      end
end;
if q = |v| then accept else reject ;
```

There should be no difficulty in seeing that the algorithm is in DSPACE($\log^2 n$).

REFERENCES

1 Herman, G., Rozenberg, G. Developmental Systems and Languages, North Holland, Amsterdam, 1975.

2 Jones, N.D. Space-bounded reducibility among combinatorial problems. Journal of Computer and System Sciences 11, pp. 68-75, 1975.

3 Jones, N.D., Skyum, S. Complexity of some problems concerning L systems. Submitted for publication. Preliminary version in Automata, Languages and Programming, Lecture Notes in Computer Science, v. 52, G. Goos, J. Hartmanis eds., pp. 301-308, Springer-Verlag, 1977.

4 Sudborough, I.H. The Complexity of the membership problem for some extensions of context-free languages, Technical Report, Northwestern University, Computer Science Department, Evanston, Ill. 1976.

5 Vitanyi, P.M.B. On the size of DOL languages, in Rozenberg, G., Salomaa, A. (Ed.) L Systems, Lecture Notes in Computer Science, v. 15, G. Goos, J. Hartmanis eds., pp. 78-92, Springer-Verlag, 1974.