ON THE EMPTINESS OF THE INTERSECTION OF TWO DOS LANGUAGES PROBLEM

by

A. Ehrenfeucht*

and

G. Rozenberg**

CU-CS-159-79 August, 1979

*A. Ehrenfeucht, Department of Computer Science, University of Colorado at Boulder, 80309 USA

**G. Rozenberg, Institute of Applied Mathematics and Computer Science, University of Leiden, 2300 RA Leiden, Holland

All correspondence to G. Rozenberg.

ON THE EMPTINESS OF THE INTERSECTION OF TWO DOS LANGUAGES PROBLEM

by

A. Ehrenfeucht
Dept. of Computer Science
University of Colorado-Boulder
Boulder, Colorado 80309 USA

and

G. Rozenberg
Institute of Applied Mathematics and Computer Science
University of Leiden
2300 RA Leiden
Holland

All correspondence to G. Rozenberg.

ON THE EMPTINESS OF THE INTERSECTION OF TWO DOS LANGUAGES PROBLEM

by

A. Ehrenfeucht
Dept. of Computer Science
University of Colorado-Boulder
Boulder, Colorado 80309 USA

and

G. Rozenberg
Institute of Applied Mathematics and Computer Science
University of Leiden
2300 RA Leiden
Holland

All correspondence to G. Rozenberg.

ABSTRACT

A DOS system is like a DOL system except that at each derivation step the underlying homomorphism is applied to one occurrence in a word only. It is shown that given two arbitrary DOS systems having the same "rewriting mapping" it is undecidable whether or not their languages intersect. Since the analogous problem is decidable for DOL systems, and a DOS system is a "sequential analogue" of a DOL system, this result points out an essential difference between parallel and sequential rewriting systems.

INTRODUCTION

The classical formal language theory based on the Chomsky hierarchy (see, e.g., [S]) and the theory of L systems, (see, e.g., [RS]) constitute today two most developed areas of formal language theory. The former is based on sequential rewriting while the latter is based on parallel rewriting. The comparison of sequential and parallel rewriting systems constitutes today an important research area in formal language theory.

A distinct feature of the mathematical theory of L systems is that its basic core fits into very systematic and natural mathematical framework. The very basic element within this framework are the so called DOL systems which are based on the iteration of a homomorphism on a free monoid. In our opinion such a systematic framework for the theory of sequential rewriting systems is missing. In an attempt to build up such a framework we introduce in this paper the notion of a DOS system which is a "sequential counterpart" of the notion of a DOL system.(1)

We demonstrate an essential difference between DOL and DOS systems. We prove that the emptiness of the intersection problem for two DOS systems having the same transition mapping is undecidable; whereas it was demonstrated in [ER] that the analogous problem for DOL systems is decidable. Moreover in proving our result we use the undecidability of the word problem for groups, rather than the Post Correspondence Problem usually used in proofs of undecidability results.

In this note we will use the standard language theoretical notation and terminology. Perhaps only the following requires an additional explanation. For a word x, $|x|$ denotes the length of x and, for $1 \le i \le |x|$, $x[i]$ denotes the letter occurring as the i'th from the left element of x. We write the composition of functions in the left to right order.

BASIC DEFINITIONS

In this section we define the basic notion of this paper: a DOS system.

Definition. Let Σ be a finite alphabet.

(i). A *sequential homomorphism* (abbreviated *s-homomorphism*) *on* Σ^* is a mapping h from Σ^* into 2^{Σ^*} defined inductively as follows:

(1). $h(\Lambda) = \{\Lambda\}$,

(2). for each $b \in \Sigma$ there exists a $y \in \Sigma^*$ such that $h(b) = \{y\}$,

(3). for each $x \in \Sigma^+$,

$h(x) = \{x_1 \beta x_2 : x = x_1bx_2$ for some $b \in \Sigma$, $x_1, x_2 \in \Sigma^*$ and $h(b) = \{y\}\}$.

The s-homomorphism h is extended to 2^{Σ^*} by letting $h(K) = \bigcup_{x \in K} h(x)$ for each $K \subseteq \Sigma^*$.

(ii). Given an s-homomorphism h, its *inverse* h^{-1} is the mapping from 2^{Σ^*} into 2^{Σ^*} defined by:

for every $K \subseteq \Sigma^*$, $h^{-1}(K) = \{x \in \Sigma^* : h(x) \cap K \neq \emptyset\}$. □

As usual, we assume that an s-homomorphism on Σ^* is given by providing its values for all singletons from Σ. To simplify the notation, in the sequel we will often identify a singleton $\{x\}$ with its element x.

Definition. A DOS *system* is a construct $G = (\Sigma, h, \omega)$ where Σ is a finite nonempty alphabet, $\omega \in \Sigma^*$ and h is an s-homomorphism on Σ^*. The *language of* G, denoted L(G), is defined by

$L(G) = \{x : x \in h^n(\omega)$ for some $n \geq 0\}$. □

Example. Let $G = (\{a,b,c\}, h, b)$ be a DOS system where $h(a) = a^2$, $h(b) = abc$ and $h(c) = c$. Then $L(G) = \{a^m bc^n : m \geq n \geq 0\}$. □

MAIN RESULT

In this section we prove that it is undecidable whether or not the languages produced by two arbitrary DOS systems having the same s-homomorphism have nonempty intersection.

We need the following technical result first.

Definition. Let h be an s-homomorphism on Σ^*. We define a binary relation $\sim_h$ on Σ^* inductively as follows. For $x, y \in \Sigma^*$,

(1). $x \overset{0}{\sim}_h y$ if and only if $x = y$,

(2). for every positive integer n, $x \overset{n}{\sim}_h y$ if and only if there exist $g_1,\ldots,g_n$ where each g_i, $1 \leq i \leq n$, is either h or h^{-1}, such that $y \in g_1\ldots g_n(x)$, and

(3). $x \sim_h y$ if and only if there exists an $n \geq 0$ such that $x \overset{n}{\sim}_h y$.

Lemma 1. Let $G_1 = (\Sigma,h,\omega_1)$ and $G_2 = (\Sigma,h,\omega_2)$ be DOS systems. Then $L(G_1) \cap LG_2) \neq \emptyset$ if and only if $\omega_1 \sim_h \omega_2$.

Proof.

(i). Obviously if $L(G_1) \cap L(G_2) \neq \emptyset$ then $\omega_1 \sim_h \omega_2$.

(ii). Let us assume that $\omega_1 \sim_h \omega_2$, hence $\omega_1 \overset{n}{\sim}_h \omega_2$ for some $n \geq 0$. We will prove now by the induction on n that $L(G_1) \cap L(G_2) \neq \emptyset$.

<u>$n = 0$</u> and <u>$n = 1$</u>. Obviously $L(G_1) \cap L(G_2) \neq \emptyset$.

Assume that for any two DOS systems G_1 and G, as above, $L(G_1) \cap L(G_2) \neq \emptyset$ whenever $\omega_1 \overset{n}{\sim}_h \omega_2$ for $n \leq k$ for some positive integer k.

Then let us assume that $\omega_1 \overset{k+1}{\sim}_h \omega_2$. That is, there exist $\xi_k \in \Sigma^*$ such that $\omega_1 \overset{k}{\sim}_h \xi_k$ and $\xi_k \overset{1}{\sim}_h \omega_2$. Let $\overline{G} = (\Sigma,h,\xi_k)$. Then, by the inductive assumption, $L(G_1) \cap L(\overline{G}) \neq \emptyset$ and $L(\overline{G}) \cap L(G_2) \neq \emptyset$. Hence, obviously, $L(G_1) \cap L(G_2) \neq \emptyset$ and so the lemma holds. □

Definition. We say that two DOS systems $G_1 = (\Sigma_1, h_1, \omega_1)$ and $G_2 = (\Sigma_2, h_2, \omega_2)$ are *cofunctional* if $\Sigma_1 = \Sigma_2$ and $h_1 = h_2$. □

Theorem. It is undecidable whether or not $L(G_1) \cap L(G_2) \neq \emptyset$ for arbitrary two cofunctional DOS systems G_1 and G_2.

Proof.

Let G be a finitely generated group (where $A_1,\ldots,A_k$, $k \geq 1$, is its set of generators) together with a finite set F of defining equalities (identities). As usual for two words x, y in G we write $x \equiv_G y$ if and only if x can be transformed into y applying finite number of times group identities ($A_iA_i^{-1} = \Lambda$ or $A_i^{-1}A_i = \Lambda$) and equalities from F.

Let $E = (\Sigma^*, R)$ where $\Sigma = \{A_1,\ldots,A_k,A_1^{-1},\ldots,A_k^{-1}\}$ and R consists of the following equalities:

$A_iA_i^{-1} = \Lambda$ for $1 \leq i \leq k$,

$A_i^{-1}A_i = \Lambda$ for $1 \leq j \leq k$, and

$uv^{-1} = \Lambda$ for every equality $u = v$ in F (where $\Lambda^{-1} = \Lambda$ and if $v = a_1\ldots a_m$, $m \geq 1$, where $a_1,\ldots,a_m$ are letters, then $v^{-1} = a_m^{-1}\ldots a_1^{-1}$).

Then, for words x,y in Σ^*, we write $x \equiv_E y$ if and only if one can transform x to y using a finite number of times equalities from R.

(i). For every $x, y \in \Sigma^*$, $x \equiv_E y$ if and only if $x \equiv_G y$.

The proof of (i) is obvious.

Let $R = \{\pi_1 = \Lambda, \pi_2 = \Lambda, \ldots, \pi_r = \Lambda\}$, $\Delta = \{B_1,\ldots,B_r\}$ where $\Delta \cap \Sigma = \Phi$ and let $\Theta = \Sigma \cup \Delta$. Let $\xi = B_1\ldots B_r$. Let h_E be the s-homomorphism on θ^* defined by:

(1). $h_E(a) = a$ for $a \in \Sigma$, and

(2). for $i \in \{1,\ldots,r\}$,

$h_E(B_i) = B_i \ldots B_n(\pi_i[1])\xi(\pi_i[2])\xi\ldots\xi(\pi_i[|\pi_i|])B_1\ldots B_i$.

Let τ be a function from Σ^* into Θ^* given by:

for $z \in \Sigma^*$,

$$\tau(z) = \begin{cases} \xi(z[1])\xi(z[2])\xi\ldots\xi(z[|z|])\xi & \text{for } z \neq \Lambda, \\ \xi & \text{for } z = \Lambda. \end{cases}$$

Let for $z \in \Sigma^*$, $H_z = (\Theta, h_E, \tau(z))$.

(ii). For every $x,y \in \Sigma^*$, $x \equiv_E y$ if and only if $\tau(x) \sim_{h_E} \tau(y)$.

This follows directly from the definitions of τ and h_E.

(iii). For every $x,y \in \Sigma^*$, $x \equiv_G y$ if and only if $L(H_x) \cap L(H_y) \neq \emptyset$.

This follows from (i), (ii) and Lemma 1.

Now we conclude the proof of the theorem as follows. It is undecidable whether or not $L(G_1) \cap L(G_2) = \emptyset$ for arbitrary two cofunctional DOS systems G_1 and G_2, because otherwise (iii) contradicts the well known fact (see, e.g., [B]) that the word problem for groups is undecidable. □

Hence we have the following result.

Corollary. It is undecidable whether or not $L(G_1) \cap L(G_2) \neq \emptyset$ for arbitrary two DOS systems G_1 and G_2. □

ACKNOWLEDGMENTS

The authors gratefully acknowledge the financial support of NSF grant number MCS 79-03838.

REFERENCES

[B]. Britton, J.L., The word problem, Ann. Math., 77 (1963) 16-32.

[ER]. Ehrenfeucht, A. and Rozenberg, G., Simplifications of homomorphisms, Information and Control, 38 (1978) 298-309.

[RS]. Rozenberg, G. and Salomaa, A., The mathematical theory of L systems, Academic Press, London - New York, to appear.

[S]. Salomaa, A., Formal languages, Academic Press, London - New York, 1974.

FOOTNOTES

(1). The letter "S" in the name "DOS system" stands for the sequential rewriting, which is to be contrasted to the letter L in "DOL system" which symbolizes the parallel rewriting (although originally it stands for the name of Lindenmayer).