THE SEQUENCE EQUIVALENCE PROBLEM IS DECIDABLE FOR OS SYSTEMS

by

Andrzej Ehrenfeucht
Department of Computer Science
University of Colorado at Boulder
Boulder, Colorado

Grzegorz Rozenberg
Department of Mathematics
University of Antwerp, U.I.A.
Wilrijk, Belgium

#CU-CS-137-78 September, 1978

ALL correspondence to second author.

Abstract. OS systems generalize context-free grammars without non-terminals. It is shown that it is decidable whether or not two arbitrary OS systems generate the same set of (derivation) sequences. As a corollary we get that it is decidable whether or not two arbitrary context free grammars have the same sets of derivation sequences.

0. INTRODUCTION

When considering a context-free grammar $G = (V_N, V_T, P, S)$ from the "computational point of view" one can restrict oneself to $\overline{G} = (V_N \cup V_T, P, S)$ which is "a context free grammar without nonterminals"; such systems were investigated e.g. in [2] or [5]. Generalized a little bit, such systems give rise to OS systems which can be viewed as the sequential counterpart of OL systems (see e.g. [3]). Studying OS systems is in our opinion a very natural step in a systematic study of the foundations of formal language theory. On the one hand in this way one hopes to build up more thorough foundations of the theory of context free languages, on the other hand when contrasted with the theory of OL systems such a study can shed new light on the basic differences between parallel and sequential rewriting systems.

In this paper we view a OS system as a system to generate sequences of words (all "derivations" in it) and then we consider the basic decision problem concerning this problem area: do two arbitrary OS systems generate the same set of sequences. We prove that this problem is decidable and show that as a corollary it yields the following result: "it is decidable whether or not two arbitrary context free grammars generate the same set of derivation sequences."

I. PRELIMINARIES

We assume the reader to be familiar with basics of the theory of context-free grammars (e.g. in the scope of [4]). Mostly we will use standard terminology and notation with perhaps the following requiring an additional explanation.

(1) For a word α, $|\alpha|$ denotes its length and $alph(\alpha)$ denotes the set of letters occurring in α. For α nonempty and $1 \leq i \leq |\alpha|$, $\alpha(i)$, denotes the i'th letter in α; e.g. $\alpha(1)$ denotes the first and $\alpha(|\alpha|)$ the last letter of α. For a positive integer k, $pref_k\alpha$ denotes the prefix of α of length k (if $k > |\alpha|$ then $pref_k\alpha = \alpha$); analogously $suf_k\alpha$ denotes the suffix of α of length k. For two words α and β, $mpref(\alpha,\beta)$ denotes the maximal common prefix of α and β and $msuf(\alpha,\beta)$ denotes the maximal common suffix of α and β.

(2) For a word α over an alphabet Σ, $\alpha = a_1 \ldots a_n$, $n \geq 1$, $a_i \in \Sigma$ for $1 \leq i \leq n$, the set $\{a_1 \ldots a_n, a_2 a_3 \ldots a_n a_1 a_2, \ldots, a_{n-1} a_n a_1 \ldots a_{n-2}, a_n a_1 \ldots a_{n-1}\}$ is called the set of *cyclic conjugates of* α and denoted by $conj(\alpha)$.

(3) Given a (directed, labeled) tree T we define its *size*, denoted as *size* T, to be the sum of lengths of all its path.

As usual in formal language theory we are faced with the problem of distinguishing between letters and their occurrences in words. In order not to burden our notation too much we will treat this problem rather informally, we hope however that this does not lead to a confusion. For example it should always be clear from the context whether we talk about the i'th occurrence in a word or about the letter that is "the value" of the i'th occurrence in a word.

In dealing with words the following well-known basic lemma (see, e.g., [1]) turns out to be very useful.

Lemma 1. Let Σ be a finite alphabet and let $\alpha_1, \alpha_2, \beta \in \Sigma^*$. If $\alpha_1\beta = \beta\alpha_2$ then there exist $\gamma, \delta \in \Sigma^*$ and $\ell \geq 0$ such that $\alpha_1 = \gamma\delta$, $\alpha_2 = \delta\gamma$ and $\beta = (\gamma\delta)^\ell\gamma = \gamma(\delta\gamma)^\ell$. □

We end this section by proving a combinatorial result on the structure of equal words which will be very crucial in our further considerations.

Lemma 2. Let Σ be a finite alphabet, $n \geq 1$, $a_1,\ldots,a_n \in \Sigma$, $\alpha\beta \in \Sigma^+$ and $1 \leq i < j \leq n$. Then

(I) $\ldots a_1\ldots a_{i-1}\alpha a_{i+1}\ldots a_n = a_1\ldots a_{j-1}\beta a_{j+1}\ldots a_n$

if and only if

(II) ... there exist $\gamma,\delta \in \Sigma^*$ and $\ell \geq 0$ such that $\alpha = a_i\gamma\delta$, $\beta = \delta\gamma a_j$ and $a_{i+1}\ldots a_{j-1} = \gamma(\delta\gamma)^\ell$.

Proof.

(i) Let us assume that (II) holds. Then

$a_1\ldots a_{i-1}\alpha a_{i+1}\ldots a_n = a_1\ldots a_i\gamma\delta\gamma(\delta\gamma)^\ell a_j\ldots a_n$ and

$a_1\ldots a_{j-1}\beta a_{j+1}\ldots a_n = a_1\ldots a_i\gamma(\delta\gamma)^\ell\delta\gamma a_j\ldots a_n$

and so (I) holds.

(ii) Let us assume that (I) holds. Let $a_1\ldots a_{i-1} = \hat{\gamma}$, $a_{i+1}\ldots a_{j-1} = \hat{\delta}$ and $a_{j+1}\ldots a_n = \xi$.

Then (I) implies that $\hat{\gamma}\alpha\hat{\delta}a_j\xi = \hat{\gamma}a_i\hat{\delta}\beta\xi$ and consequently

(III) $\ldots \alpha\hat{\delta}a_j = a_i\hat{\delta}\beta$.

Thus there exist words $\overline{\alpha},\overline{\beta} \in \Sigma^*$ such that $\alpha = a_i\overline{\alpha}$ and $\beta = \overline{\beta}a_j$. Hence (III) implies that $\overline{\alpha}\hat{\delta} = \hat{\delta}\overline{\beta}$ which by Lemma 1 implies that there exist words γ,δ and integer $\ell \geq 0$ such that $\overline{\alpha} = \gamma\delta$, $\overline{\beta} = \delta\gamma$ and $\hat{\delta} = \gamma(\delta\gamma)^\ell$. Consequently $\alpha = a_i\gamma\delta$, $\beta = \delta\gamma a_j$ and $a_{i+1}\ldots a_{i-1} = \gamma(\delta\gamma)^\ell$. Thus (II) holds. □

II. OS SYSTEMS AND PRODUCTIONS

In this section OS systems are introduced and also some basic notions concerning (rewriting) productions are considered.

First of all we owe the reader the explanation why do we give a new name to grammars that are variations of context free grammars without nonterminals. The reason is that the work presented in this paper forms a result in our research concerning foundations of formal language theory. We are convinced that the theory of L systems (see, e.g. [3]) constitutes an example of a systematic build-up of a fragment of formal language theory. There one considers two basic mappings: a homomorphism and a finite substitution and forms a rewriting system by either iterating a homomorphism or a finite number of them or iterating a finite substitution or a finite number of them. Thus e.g. the name "an OL system" denotes the L system (where L symbolizes parallel rewriting) without interactions (that is what 0, *zero*, stands for), which consists of iterations of a finite substitution. We want to consider the theory of Chomsky grammars also to be built up in this way and so e.g. the analogue of an OL system will be an OS system where S will stand for sequential rewriting (the "sequential use" of finite substitution).

Definition. A *sequential system without interactions*, denoted as an OS *system*, is a three-tuple $G = (\Sigma,h,\omega)$ where Σ is a finite nonempty alphabet (called the *alphabet of* G), $\omega \in \Sigma^+$ (called the *axiom of* G) and h is a homomorphism from Σ_1^* into Σ_2^* where $\Sigma_1 \subseteq \Sigma$ and $\Sigma_2 \subseteq \Sigma$ (called the *transition function of* G or the *set of productions of* G; each pair (x,α) with $x \in \Sigma_1$ and $\alpha \in \Sigma_2^*$ such that $h(x) = \alpha$ is called a *production in* G). □

Definition. Let $G = (\Sigma,h,\omega)$ be an OS system.

(1) Let $\beta \in \Sigma^+$ and $\gamma \in \Sigma^*$. We say that β *directly derives* γ *in* G, denoted

as $\beta \underset{G}{\Rightarrow} \gamma$, if $\beta = \beta_1 x \beta_2$, $\gamma = \beta_1 \alpha \beta_2$ and (x,α) is a production in G.

(2) Let $\beta \in \Sigma^+$ and $\gamma \in \Sigma^*$. We say that β *derives* γ *in* G, denoted as $\beta \overset{*}{\underset{G}{\Rightarrow}} \gamma$, if there exist words $\gamma_1,\ldots,\gamma_n$ such that $\gamma_n = \gamma$, $\gamma_{i-1} \underset{G}{\Rightarrow} \gamma_i$ for $2 \leq i \leq n$ and $\beta \underset{G}{\Rightarrow} \gamma_1$.

(3) A finite sequence of words $\gamma_0,\gamma_1,\ldots,\gamma_n$, $n \geq 1$, is called a G-*sequence* if $\gamma_0 = \omega$ and $\gamma_i \underset{G}{\Rightarrow} \gamma_{i+1}$ for $0 \leq i \leq n-1$. The set of all G-sequences is called the *set of sequences of* G and is denoted as E(G).

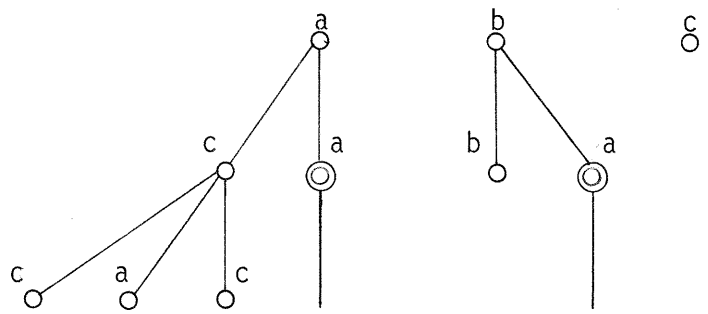
(4) The *language of* G, denoted as L(G), is defined by $L(G) = \{\beta \in \Sigma^* : \omega \overset{*}{\underset{G}{\Rightarrow}} \beta\}$. □

As usual in language theory, to every G-sequence we can assign a *derivation graph* which in the case when the axiom is of length one becomes a *derivation tree*. Given a derivation graph T a node e in it is called *productive* if it contributes a nonempty subword to the last word of a sequence represented by T (this subword is denoted by *contr*(e)); otherwise is called *nonproductive*.

Example. Let $G = (\Sigma,h,\omega)$ be the OS system where $\Sigma = \{a,b,c\}$, $\omega = abc$ and $h(a) = \{ca,\Lambda\}$, $h(b) = \{ba,b^2\}$ and $h(c) = \{cac\}$. Then

abc,cabc,cbc,cacbc,cacbac,cacbc

is a G-sequence and its derivation graph looks as follows



where the improductive nodes are double circled. □

We introduce now some terminology and notation needed in this paper.

Given a production (x,α) in an OS system $G = (\Sigma,h,\omega)$ we write it also in the form $x \to \alpha$ and we also write $(x \to \alpha) \in h$ and say that $x \to \alpha$ is a production in h. As a matter of fact taking a more general look we say that $x \to \alpha$ is a *production (over* Σ) if $x \in \Sigma$ and $\alpha \in \Sigma^*$. For a production $\pi = (x \to \alpha)$ we say that x is its *left-hand side* (denoted as $lhs(\pi)$) and α is its *right-hand side* (denoted as $rhs(\pi)$). The *length of* π is denoted by $|\pi|$ and defined by $|\pi| = |rhs(\pi)|$. (For $G = (\Sigma,h,\omega)$ we use $maxr G$ to denote $\max\{|\pi| : \pi$ is a production in $G\}$.) The *domain of* π, denoted as $dom(\pi)$, is defined by $dom(\pi) = \{\beta \in \Sigma^+ : \beta = \beta_1(lhs(\pi))\beta_2$ for some $\beta_1,\beta_2 \in \Sigma^*\}$. Then for a word β in $\Sigma^+ \setminus dom(\pi)$, $\pi(\beta) = \emptyset$, and for a word β in $dom(\pi)$, $\pi(\beta) = \{\beta_1(rhs(\pi))\beta_2 : \beta = \beta_1(lhs(\pi))\beta_2$ for some $\beta_1,\beta_2 \in \Sigma^*\}$.

Let π and ρ be productions over some alphabet Σ. We say that they are *associated* (written as $\pi \sim \rho$) if there exists an x in Σ^+ such that $\pi(x) \cap \rho(x) \neq \emptyset$.

Obviously $\pi \sim \pi$ for every production π.

We classify now productions into six types.

Let π be a production over an alphabet Σ, $\pi = (x \to \alpha)$.

(1) π is of *type* 1 if $\alpha = \Lambda$.

(2) π is of *type* 2 if $\alpha = x$.

(3) π is of *type* 3 if $\alpha(1) \neq x$ and $\alpha(|\alpha|) \neq x$.

(4) π is of *type* 4 if $\alpha = x\beta$ for $\beta \in \Sigma^+$ such that $\beta(|\beta|) \neq x$.

(5) π is of *type* 5 if $\alpha = \beta x$ for $\beta \in \Sigma^+$ such that $\beta(1) \neq x$.

(6) π is of *type* 6 if $\alpha = x\beta x$ for $\beta \in \Sigma^*$.

Lemma 3.

(1) If π is a production of type 1 and $\rho \sim \pi$ then $\rho = \pi$.

(2) If π is a production of type 2 and $\rho \sim \pi$ then ρ is also a production of type 2.

(3) If π is a production of type 3 and $\rho \sim \pi$ then $\rho = \pi$.

(4) If π is a production of type 4, $\pi = (x \to x\beta)$, and $\rho \sim \pi$ then either $\rho = \pi$ or $\rho = (y \to \overline{\beta}y)$ where $\overline{\beta} \in conj(\beta)$.

(5) If π is a production of type 5, $\pi = (x \to \beta x)$, and $\rho \sim \pi$ then either $\rho = \pi$ or $\rho = (y \to y\overline{\beta})$ where $\overline{\beta} \in conj(\beta)$.

(6) If π is a production of type 6, $\pi = (x \to x\beta x)$, and $\rho \sim \pi$ then either $\rho = \pi$, or $\rho = (y \to \overline{\beta}y)$ where $\overline{\beta} \in conj(\beta x)$, or $\rho = (y \to y\overline{\beta})$ where $\overline{\beta} \in conj(x\beta)$.

Proof.

(1) and (2) are rather obvious and (3) through (6) follows from Lemma 2. □

III. WINNING WORDS

When one considers whether or not $E(G_1) \subseteq E(G_2)$ for OS systems G_1, G_2 it is quite instructive to consider this problem as a game of G_1 against G_2. Given a word α in $L(G_1)$, whenever G_1 applies a production π to (the i'th occurrence in) α we say that G_1 *attacks*, or makes a *move on* α (more specifically the (i,π)-*move on* α). In this way one obtains a word β from $\pi(\alpha)$. We say that G_2 *defends* (this move) if by applying a production ρ to (the j'th occurrence of) α it generates β (more specifically it defends by making the (j,ρ)-move on α).

Thus (assuming that G_1 and G_2 have equal axioms) $E(G_1) \not\subseteq E(G_2)$ if and only if G_1 can win against G_2. We are going to explore this point of view somewhat further now.

It will be very convenient in our considerations to assume that every word starts with the marker $\not\!c$ and ends with the marker $\not\!c$ where $\not\!c$ does not belong to the alphabet Σ fixed in our considerations.

Let $\alpha \in \not\!c\Sigma^+\not\!c$, $\alpha = \not\!c a_1 \ldots a_n \not\!c$, $a_j \in \Sigma$ for $1 \leq j \leq n$, let $\pi \in h_1$ and let $i \in \{1,\ldots,n\}$. We say that α is an (i,π)-*winning word* if $lhs(\pi) = \alpha(i)$ and $\not\!c a_1 \ldots a_{i-1} rhs(\pi) a_{i+1} \ldots a_n \not\!c \notin h_2(\alpha)$.

If α is an (i,π)-winning word for some i and π then we also say that α is a *winning word*.

Let $\alpha \in \not\!c\Sigma^+\not\!c$, $\alpha = \not\!c a_1 \ldots a_n \not\!c$, $a_j \in \Sigma$ for $1 \leq j \leq n$, let π be a production over Σ and let $i \in \{1,\ldots,n\}$.

(i) The *left horizon of the occurrence* a_i *with respect to* π *in* α, denoted as $lhor_{\alpha,\pi}(a_i)$ is defined as follows:

(1) if $\alpha \notin dom(\pi)$ then $lhor_{\alpha,\pi}(a_i) = a_i$,

(2) if $\alpha \in dom(\pi)$ and π is of type j for $j \in \{1,2,3,4\}$ then $lhor_{\alpha,\pi}(a_i) = a_i$,

(3) if $\alpha \in dom(\pi)$ and π is of type 5, $\pi = (x \to \beta x)$, then $lhor_{\alpha,\pi}(a_i) = a_{i-t-1}$ where $t = |msuf(\beta^i, pref_{i-1}\alpha)|$,

(4) if $\alpha \in dom(\pi)$ and π is of type 6, $\pi = (x \to x\beta x)$, then $lhor_{\alpha,\pi}(a_i) = a_{i-t-1}$ where $t = |msuf((x\beta)^i, pref_{i-1}\alpha)|$.

(ii) The *right horizon of the occurrence* a_i *with respect to* π *in* α, denoted as $rhor_{\alpha,\pi}(a_i)$, is defined as follows:

(1) if $\alpha \notin dom(\pi)$ then $rhor_{\alpha,\pi}(a_i) = a_i$,

(2) if $\alpha \in dom(\pi)$ and π is of type j for $j \in \{1,2,3,5\}$ then $rhor_{\alpha,\pi}(a_i) = a_i$,

(3) if $\alpha \in dom(\pi)$ and π is of type 4, $\pi = (x \to x\beta)$, then $rhor_{\alpha,\pi}(a_i) = a_{i+t+1}$ where $t = |mpref(\beta^{|\alpha|}, pref_{|\alpha|}a_{i+1}\ldots a_n)|$,

(4) if $\alpha \in dom(\pi)$ and π is of type 6, $\pi = (x \to x\beta x)$, then $rhor_{\alpha,\pi}(a_i) = a_{i+t+1}$ where $t = |mpref((\beta x)^{|\alpha|}, pref_{|\alpha|}a_{i+1}\ldots a_n)|$.

Note that, because α starts and ends with ¢, both $lhor_{\alpha,\pi}(a_i)$ and $rhor_{\alpha,\pi}(a_i)$ are well defined.

Let $\alpha = ¢a_1\ldots a_n¢ \in ¢\Sigma^+¢$, $a_j \in \Sigma$ for $1 \le j \le n$, be an (i,π)-winning word. Then we write α in the form $\alpha = left_{i,\pi}(\alpha)\ mid_{i,\pi}(\alpha)\ right_{i,\pi}(\alpha)$ where $mid_{i,\pi}(\alpha)$ is a subword of α starting with $lhor_{\alpha,\pi}(a_i)$ and ending with $rhor_{\alpha,\pi}(a_i)$.

We will consider now several cases when one can alter a winning word (α) to obtain another winning word.

Lemma 4. If γ_1, γ_2 are words such that $\gamma_1 mid_{i,\pi}(\alpha)\gamma_2 \in ¢\Sigma^+¢$ and $alph(\gamma_1\gamma_2) \subseteq alph(\alpha)$ then $\gamma = \gamma_1 mid_{i,\pi}(\alpha)\gamma_2$ is an $(i - |left_{i,\pi}(\alpha)| + |\gamma_1|, \pi)$-winning word.

Proof.

We will prove the lemma by demonstrating that if x is not an $(i - |left_{i,\pi}(\alpha)| + |\gamma_1|, \pi)$-winning word then α could not be an (i,π)-winning word.

So let us assume that γ is not an $(i - |left_{i,\pi}(\alpha)| + |\gamma_1|, \pi)$-winning word. (Note that $\gamma(i - |\alpha_1| + |\gamma_1|) = \alpha(i)$ and so $lhs(\pi) = \gamma(i - |\alpha_1| + |\gamma_1|)$.)

We have six cases to consider.

(1) If π is of type 1 then (by Lemma 3) G_2 defends by the $(i - |\alpha_1| + |\gamma_1|,\pi)$-move on α. Hence $\pi \in h_2$. But then G_2 can defend the (i,π)-move on α by G_1 by making the (i,π)-move on α.

(2) If π is of type 2 then (by Lemma 3) G_2 defends by applying a production of type 2 to γ. Since $alph(\gamma) \subseteq alph(\alpha)$, G_2 can defend the (i,π)-move on α by G_1.

(3) If π is of type 3 then (by Lemma 3) G_2 defends by the $(i - |\alpha_1| + |\gamma_1|,\pi)$-move on γ. Consequently the (i,π)-move by G_1 on α is defended by the (i,π)-move by G_2 on α.

(4) If π is of type 4 then (by Lemma 3) we have two cases to consider.

(4.1) If the $(i - |\alpha_1| + |\gamma_1|,\pi)$-move by G_1 on γ is defended by G_2 also by applying the production π then by Lemma 2 it must be the $(i - |\alpha_1| + |\gamma_1|,\pi)$-move by G_2 on γ. But then the (i,π)-move by G_1 on α can be defended by the (i,π)-move by G_2 on α.

(4.2) If the $(i - |\alpha_1| + |\gamma_1|,\pi)$-move by G_1 on γ is defended by G_2 by an application of a production $\rho = (y \to \overline{\beta}y)$, where $\pi = (x \to x\beta)$ and $\overline{\beta} \in conj(\beta)$ then (by the definition of the $rhor_{\alpha,\pi}(a_i)$) G_2 can defend by applying ρ to an occurrence b in γ which lies within $mid_{i,\pi}(\alpha)$. But then the (i,π)-move by G_1 on α can be defended by G_2 by applying ρ to "the same" occurrence b.

The two remaining cases (of π being of type 5 and π being of type 6) can be proved analogously to the case (4). □

Lemma 5. Let $|mid_{i,\pi}(\alpha)| \geq 3$ and let

$mid_{i,\pi}(\alpha) = lhor_{\alpha,\pi}(a_i)\gamma_1\beta_1\gamma_2 a_i \gamma_3\beta_2\gamma_4 rhor_{\alpha,\pi}(a_i)$

where β_1,β_2 are words such that $(|\pi| - 1)$ divides $|\beta_1|$ and $(|\pi| - 1)$ divides $|\beta_2|$. Then the word

$left_{i,\pi}(\alpha)\ lhor_{\alpha,\pi}(a_i)\gamma_1\gamma_2 a_i\gamma_3\gamma_4 rhor_{\alpha,\pi}(a_i) right_{i,\pi}(\alpha)$ is an $(i - |\beta_1|,\pi)$-winning word.

Proof.

Since the proof of this lemma can be carried analogously to the proof of Lemma 4, we leave it to the reader. The crucial observation here is that, because $(|\pi| - 1)$ divides $|\beta_1|$ (and $(|\pi| - 1)$ divides $|\beta_2|$), by removing β_1 (and β_2) we have removed from the periodic word $(\gamma(\delta\gamma)^{\ell}$ — see Lemma 2) subwords the length of which are multiplicities of the period. But then if G_2 can defend on such words it can also defend on the original one and in the "corresponding" positions. □

In the same way we leave to the reader the proofs of the following two lemmas.

Lemma 6. Let $|mid_{i,\pi}(\alpha)| \geq 2$ and let $mid_{i,\pi}(\alpha) = a_i\gamma_3\beta_2\gamma_4 rhor_{\alpha,\pi}(a_i)$ where $(|\pi| - 1)$ divides $|\beta_2|$. Then the word $left_{i,\pi}(\alpha)a_i\gamma_3\gamma_4 rhor_{\alpha,\pi}(a_i)right_{i,\pi}(\alpha)$ is also an (i,π)-winning word. □

Lemma 7. Let $|mid_{i,\pi}(\alpha)| \geq 2$ and let $mid_{i,\pi}(\alpha) = lhor_{\alpha,\pi}(a_i)\gamma_1\beta_1\gamma_2 a_i$ where $(|\pi| - 1)$ divides $|\beta_1|$. Then the word $left_{i,\pi}(\alpha)lhor_{\alpha,\pi}(a_i)\gamma_1\gamma_2 a_i right_{i,\pi}(\alpha)$ is an $(i - |\beta_1|,\pi)$-winning word. □

IV. THE OS SEQUENCE EQUIVALENCE PROBLEM

We will consider now the *sequence equivalence problem for* OS *systems:* "Is $E(G_1)$ equal to $E(G_2)$ for arbitrary two OS systems G_1 and G_2?"

We will demonstrate that this problem is decidable, that is there exists an algorithm which, given arbitrary OS systems G_1 and G_2, decides whether or not $E(G_1) = E(G_2)$.

To this aim let $G_1 = (\Sigma_1, h_1, \omega_1)$ and $G_2 = (\Sigma_2, h_2, \omega_2)$ be two arbitrary OS systems that we fix now for our considerations. Clearly considering the sequence equivalence problem for OS systems we can assume that $\Sigma_1 = \Sigma_2 = \Sigma$, $\omega_1 = \omega_2 = \omega$ and $|\omega| = 1$. Also following our convention from the last section we assume that words in G_1 and G_2 start and end with ¢ which is not an element of Σ and so it is never rewritten.

The following result allows one to check whether G_1 contains winning words.

Lemma 8. Let $C = 4 \cdot (\#\Sigma) \cdot (max r_{G_1})^2$. The following two statements are equivalent:

(1) $L(G_1)$ contains a winning word,

(2) $L(G_1)$ contains a winning word α such that there exists a derivation tree T_α of α with the property that no path in T_α is longer than C.

Proof.

(i). Clearly (2) implies (1).

(ii). To prove that (1) implies (2) we proceed as follows. Let $\alpha = ¢a_1 \ldots a_n ¢$, $a_1, \ldots, a_n \in \Sigma$, be a winning word in $L(G_1)$ with the following property:

(*)... { it has a derivation tree T which is such that among all the derivation trees in G_1 for winning words there is none of the size smaller than T.

Let α be an (i,π)-winning word and let (as in the preceding) it be of the form $\alpha = left_{i,\pi}(\alpha)\ mid_{i,\pi}(\alpha)\ right_{i,\pi}(\alpha)$.

We consider two cases separately.

(ii.1) $|mid_{i,\pi}(\alpha)| = 1$.

Let p be an arbitrary path in T starting at the root of T and ending at the occurrence in α (a leaf).

We have two cases to consider.

(ii.1.1) p contains a node e that is an ancestor of $mid_{i,\pi}(\alpha)$, however the direct descendent of e is not an ancestor of $mid_{i,\pi}(\alpha)$.

Then let p_1 be the "initial" part of p, starting at the root and ending at e, and let p_2 be the "final" part of p, starting at the direct descendent of e and ending at $mid_{i,\pi}(\alpha)$.

Clearly neither p_1 nor p_2 can have nodes with the same labels, because otherwise we could "shorten" p by removing the path between such two nodes and obtain a derivation tree $\overline{T}$ of a winning word (see Lemma 4) such that $size\overline{T} < sizeT$ which contradicts (*). Thus the length of p is bounded by $2 \cdot (\#\Sigma)$.

(ii.2.3) It is not true that (ii.1.1) holds.

Then the reasoning analogous to that from (ii.1.1) yields that the length of p is bounded by $\#\Sigma$.

(ii.2). $|mid_{i,\pi}(\alpha)| \geq 2$.

Then we have three cases to consider; they correspond to Lemma 5, Lemma 6 and Lemma 7 respectively.

Clearly the "worst case" is the one corresponding to Lemma 5 and so we consider only this case.

Thus let $mid_{i,\pi}(\alpha) = lhor_{\alpha,\pi}(a_i)\gamma_1\beta_1\gamma_2 a_i \gamma_3\beta_2\gamma_4 rhor_{\alpha,\pi}(a_i)$ where $(|\pi| - 1)$ divides $|\beta_1|$ and $(|\pi| - 1)$ divides $|\beta_2|$.

Let p be an arbitrary path in T starting at the root and ending at a leaf (an occurrence in α). A node e on p is called *special* if the subword it contributes to α contains one of the occurrences from the set $\{lhor_{\alpha,\pi}(a_i), rhor_{\alpha,\pi}(a_i), a_i\}$ but the direct descendent of e does not contain this occurrence in its contribution to α.

Clearly p can contain at most three distinguished points which yields the division of p (done analogously to that in (ii.1.1)) into at most four parts p_1, p_2, p_3, p_4 (which catenated in this order yield p).

Let us consider an arbitrary part $\overline{p}$ of p ($\overline{p} \in \{p_1, p_2, p_3, p_4\}$). To each node e of $\overline{p}$ we assign its description $des(e) = (x, M, N)$ defined as follows: x is the label of e in T,

if e is a productive node in T and $contr(e) = a_q a_{q+1} \ldots a_{q+s}$ for some $1 \leq q \leq n-1$, $s \geq 1$ then $M = q-1 \pmod{(|\pi| - 1)}$ and $N = q + s \pmod{(|\pi| - 1)}$,

if e is a nonproductive node in T then $M = N = |\pi| - 1$.

We note that no two different nodes on $\overline{p}$ can have the same description. The reason is that by removing the path between them we obtain

— either a smaller derivation tree of α (if those nodes are improductive),

— or a smaller derivation tree of a winning word $\overline{\alpha}$ shorter than α; this follows from Lemmas 4, 5, 6 and 7 and from an observation that the construction of $\overline{p}$ guarantees that the subword we remove to obtain $\overline{\alpha}$ from α lies either to the left of $lhor_{\alpha,\pi}(a_i)$, or between $lhor_{\alpha,\pi}(a_i)$ and a_i, or between a_i and $rhor_{\alpha,\pi}(a_i)$, or to the right of $rhor_{\alpha,\pi}(a_i)$ — moreover the length of these subwords is divisible by $(|\pi| - 1)$.

Thus the length of $\overline{p}$ is bounded by $(\#\Sigma)(maxrG_1)^2$ and consequently the length of p is bounded by $4 \cdot (\#\Sigma) \cdot (maxrG_1)^2$.

This completes the proof of the lemma. □

On the basis of the above lemma we can demonstrate now that the OS sequence equivalence problem is decidable.

Theorem 1. It is decidable whether or not $E(G_1) = E(G_2)$ for arbitrary OS systems G_1 and G_2.

Proof.

Clearly to decide whether or not $E(G_1) = E(G_2)$ it suffices to decide whether or not $E(G_1) \subseteq E(G_2)$ and $E(G_2) \subseteq E(G_1)$. It is also obvious that $E(G_i) \subseteq E(G_j)$, $i,j \in \{1,2\}$, $i \neq j$, if and only if $E(G_i)$ does not contain winning words (with respect to G_2). However, by Lemma 8, to decide whether or not G_i contains a winning word it suffices to generate in G_i all G_i-sequences for which derivation trees do not have a path longer than $4 \cdot (\#\Sigma) \cdot (maxr G_i)^2$ and then check whether or not among those words there is a winning word. This is clearly effective since for each such word α it suffices to check whether or not $h_i(\alpha)$ contains a word which is not in $h_j(\alpha)$. □

Let $G = (V_N, V_T, P, S)$ be a context-free grammar. (As usual we assume that every nonterminal in G can be rewritten, in a number of steps, in such a way that it yields a word in V_T^*). The weakest notion of a *derivation in* G (used quite often in the literature) is defined as the sequence of words $\beta_0, \beta_1, \ldots, \beta_n$, $n \geq 1$, such that $\beta_0 = S$, $\beta_n \in V_T^*$ and $\beta_i \underset{G}{\Rightarrow} \beta_{i+1}$ for $0 \leq i \leq n-1$. We will call the set of all derivations (in the sense as above) the *computation set of* G and denote it as *comp*G. Clearly in considering *comp*G one can consider the OS system $\overline{G} = (V,H,S)$ where $V = V_N \cup V_T$ and h is defined by productions in P (so $h : V_N^* \to V^*$). Then, obviously, $compG_1 = compG_2$ if and only if $E(\overline{G}_1) = E(\overline{G}_2)$ where G_1, G_2 are two arbitrary context-free grammars.

Clearly the question whether or not $compG_1 = compG_2$ for arbitrary context-free grammars G_1 and G_2 is one of the most natural questions about context-free grammars. The above reasoning and Theorem 1 yields now the following result.

Theorem 2. It is decidable whether or not $comp G_1 = comp G_2$ for arbitrary context-free grammars G_1 and G_2.

It is very instructive to compare this result with the well known result, see, e.g. [5] that it is not decidable whether or not two context-free grammars generate the same set of sentential forms.

REFERENCES

[1] M. Harrison, Formal languages, Addison-Wesley, Reading, Mass., 1978.

[2] T. Harju and M. Penttonen, Some decidability problems of sentential forms, Techn. Rep. 78-CS-8, Dept. of Appl. Math., McMaster University, 1978.

[3] G.T. Herman and G. Rozenberg, Developmental systems and languages, North-Holland, Amsterdam, 1975.

[4] A. Salomaa, Formal languages, Academic Press, London, 1973.

[5] A. Salomaa, On sentential forms of context-free grammars, Acta Informatica 2, 40-49.

Acknowledgments. The second author is very much indebted to IBM Belgium and NFWO foundation for supporting his stay at the University of Colorado at Boulder which made the work on this and other papers possible.