KYBERNETIKA — VOLUME 19 (1983), NUMBER 3

DESCRIPTIONAL COMPLEXITY MEASURES OF CONTEXT-FREE LANGUAGES

ANTON ČERNÝ

The properties of several new descriptional complexity measures of context-free languages are discussed. Though these measures seem to be very simple the basic algorithmic problems remain to be undecidable.

1. INTRODUCTION

In [2], [3] and [4] several measures of descriptional complexity of context-free grammars (cfg's) and context-free languages (cfl's) have been investigated, most of them having the following properties:

1. The corresponding hierarchy of complexity classes of languages over two-letter alphabets is infinite.

2. The basic algorithmic problems are undecidable. (For example, the problems to determine the complexity of a language generated by a given grammar, to decide whether the given grammar is minimal or to construct an equivalent minimal grammar.)

In the present paper an attempt is made to investigate measures which seem to be simpler from two points of view. Over a fixed alphabet they induce a finite hierarchy of languages, and to determine the complexity of a grammar only a part of the grammar has to be considered. In spite of this for most of these new measures the basic algorithmic problems remain to be undecidable.

2. PRELIMINARIES

A survey of the descriptional complexity theory of formal languages is given in [4]. The basic notions of context-free languages theory to be used here are from [1].

A *cfg* is a quadruple $G = (V, \Sigma, P, S)$ with V and $\Sigma \subseteq V$ being finite sets of symbols

P a finite set of productions of the form $A \to x$ where $A \in V - \Sigma$, $x \in V^*$, and $S \in V - \Sigma$ the start symbol. The elements of Σ, resp. $V - \Sigma$, are called *terminals*, resp. *variables*. We write $w_1 A w_2 \Rightarrow w_1 x w_2$ iff $A \to x$ is in P and $w_1, w_2 \in V^*$. The relation $\Rightarrow^*$ is a reflexive and transitive closure of $\Rightarrow$. The language defined by a cfg G is $L(G) = \{w \in \Sigma^*; S \Rightarrow^* w\}$.

In Section 4 the ε-reduced form (ε being the empty word) of cfg's will be often used. A cfg $G = (V, \Sigma, P, S)$ is said to be in *ε-reduced* form iff

a) for no variable $A \neq S$, $A \to \varepsilon$ is in P

b) if $S \to \varepsilon$ is in P, then S does not appear in any of the right sides of productions in P.

The undecidability results in Sections 3 and 4 will be obtained using a reduction to the Post correspondence problem. In doing that a class of languages, denoted by $L'_{x,y}$, will be used. To define $L'_{x,y}$ the languages $L_{x,y}$ described in [1] are used.

Let $x = (x_1, \ldots, x_n)$, $y = (y_1 \ldots, y_n)$ be ordered n-tuples ($n \geqq 1$) of nonempty words over the alphabet $\{a, b\}$. Let

$$L(x) = \{ba^{i_k} \ldots ba^{i_1} c x_{i_1} \ldots x_{i_k};\ k \geqq 1,\ n \geqq i_j \geqq 1,\ j = 1, 2, \ldots, k\}$$

$$L(x, y) = L(x)\, c\, L(y)^R$$

$$L_s = \{w_1 c w_2 c w_2^R c w_1^R;\ w_1, w_2 \in \{a, b\}^*\}$$

$$L_{x,y} = \{a, b, c\}^* - L_s \cap L(x, y)$$

According to [1], $L_{x,y}$ is a cfl and a cfg $G_{x,y}$ generating $L_{x,y}$ can be effectively constructed given x and y. Moreover, $L_{x,y} = \{a, b, c\}^*$ iff the Post correspondence problem for the pair (x, y) has no solution.

The languages $L'_{x,y}$ can now be defined by $L'_{x,y} = h(L_{x,y})$ where $h : \{a, b, c\}^* \to \{a, b\}^*$ be the homomorphism defined by $h(a) = ab$, $h(b) = a^2b^2$, $h(c) = a^3b^3$.

Clearly, a cfg $G'_{x,y}$ generating $L'_{x,y}$ can be effectively constructed given x and y, and $L'_{x,y} = h(\{a, b, c\}^*)$ iff the Post correspondence problem for the pair (x, y) has no solution.

In our last definition, the notion of descriptional complexity measure ([4]) is introduced. Let $\mathscr{G}(\mathscr{L})$ denote the class of all cfg's (cfl's) and N the set of all nonnegative integers. A *descriptional complexity measure* of cfg's (cfl's) is an arbitrary mapping

$$\mathbf{K} : \mathscr{G} \to N \quad (\mathbf{K} : \mathscr{L} \to N)$$

Every complexity measure $\mathbf{K}$ of cfg's induces a complexity measure of cfl's. This measure is also denoted $\mathbf{K}$ and is defined as follows

$$\mathbf{K}(L) = \min \{\mathbf{K}(G);\ G \in \mathscr{G},\ L(G) = L\}$$

for every cfl L.

3. PRODUCTIONS WITH TERMINALS

The number of productions of a cfg G – *Prod* (G) is one of the basic complexity measures of cfg's. One way to define a simpler measure seems to be to count some special productions only.

Since variables and terminals play such an important role in the definition of cfg's, it is natural to consider as a complexity measure the number of those productions in which terminals and/or variables satisfy some special condition. For example, productions with at least one (with no) terminal, productions with at least one (with no) variable in the right side and so on. The measures of the former type are investigated in this section. We consider the number of productions with a terminal as a complexity measure.

In the following definition two terminal-based complexity measures are defined. They differ only in the case when a cfg contains ε-productions.

Definition 1. Let $G = (V, \Sigma, P, S)$ be a cfg. Then

$PT(G)$ = the number of productions in P with right hand side containing at least one terminal

$PT_\varepsilon(G) = PT(G)$ + the number of ε-productions in P

The basic relations between these two measures and the number of symbols of the underlying alphabet are summarized in the following lemma.

Lemma 1. Let Σ be a finite alphabet and $L \subseteq \Sigma^*$ a cfl. Then

(i) $PT(L) \leqq PT_\varepsilon(L) \leqq PT(L) + 1$

(ii) $PT(L) \leqq |\Sigma|$

In spite of property (ii) of Lemma 1 measures PT and PT_ε induce an infinite hierarchy of complexity classes of cfl's with no gaps. Indeed, it is easy to verify that for any integer $n \geqq 0$ $PT(L_n) = n$ $(PT_\varepsilon(L_n) = n)$ holds for the languages $L_0 = \emptyset$, $L_n = \{a_1, \ldots, a_n\}$ over the alphabets $\Sigma_n = \{a_1, \ldots, a_n\}$ $(n \geqq 1)$.

Though measures PT and PT_ε are relatively simple, the basic algorithmic problems for them are undecidable. The proofs of undecidablility are based on the following lemma.

Lemma 2. There is no algorithm to decide for an arbitrary cfg G whether or not $PT(L(G)) = 1$ $(PT_\varepsilon(L(G)) = 2)$.

Proof. Let D denote the Dyck language generated by the grammar with two productions

$$S \to SaSb \mid \varepsilon$$

Let h be the homomorphism from Section 2. For every pair (x, y) of n-tuples of non-empty words over $\{a, b\}$ denote

$$L''_{x,y} = (D - ah(\{a, b, c\}^*)\, b) \cup a\, L'_{x,y} b$$

Since the language $a\boldsymbol{h}(\{a, b, c\}^*)\, b$ is regular, a cfg $G''_{x,y}$ can be constructed with $L(G''_{x,y}) = L''_{x,y}$.

Two cases will be considered now.

I. The Post correspondence problem for (x, y) possesses no solution. In such a case $L'_{x,y} = \boldsymbol{h}(\{a, b, c\}^*)$ and therefore $L''_{x,y} = D$, $PT(L'_{x,y}) = 1$, $PT_\varepsilon(L''_{x,y}) = 2$.

II. The Post correspondence problem for (x, y) has a solution. We shall prove by a contradiction that $PT(L''_{x,y}) \geqq 2$ and $PT_\varepsilon(L''_{x,y}) \geqq 3$ in this case.

Let $G = (V, \{a, b\}, P, S)$ be a grammar generating $L''_{x,y}$. Suppose $PT(G) = 1$. Since $ab \in a\, L'_{x,y} b \subseteq L''_{x,y}$, the only production in P with a terminal in the right side has to be of the form

$$A \to uavbz,\ uvz \in (V - \{a, b\})^* \tag{1}$$

and there exist words $v_1, v_2 \in (V - \{a, b\})^*$ such that

$$S \Rightarrow^* v_1 A v_2 \Rightarrow v_1 uavbz v_2 \Rightarrow^* ab$$

However, this is possible only if

$$S \Rightarrow^* v_1 A v_2 \Rightarrow^* A,\ v \Rightarrow^* \varepsilon,\ uz \Rightarrow^* \varepsilon \tag{2}$$

Let us now denote $L(v) = \{w \in \{a, b\}^*;\ v \Rightarrow^* w\}$ and suppose for a moment that the inclusion $D \subseteq L(v)$ has been proven. Since there exists a solution of the Post correspondence problem for (x, y), there is a word

$$t \in \boldsymbol{h}(\{a, b, c\}^*) - L'_{x,y} \tag{3}$$

Then $t \in D \subseteq L(v)$, and by (1), (2)

$$S \Rightarrow^* A \Rightarrow uavbz \Rightarrow^* avb \Rightarrow^* atb \tag{4}$$

is a derivation in G. Consequently, $atb \in L''_{x,y}$. On the other hand, by (3) the following relations are simultaneously valid:

$$atb \notin D - a\boldsymbol{h}(\{a, b, c\}^*)\, b,\ atb \notin a\, L'_{x,y} b$$

From here we obtain that $atb \notin L''_{x,y}$ what contradicts to (4). This means that $PT(L''_{x,y}) \geqq 2$ (and $PT_\varepsilon(L''_{x,y}) \geqq 3$) in this case. Therefore to finish the proof of lemma the inclusion $D \subseteq L(v)$ remains to be shown. Since ε is in $L(v)$, in order to prove $D \subseteq L(v)$ it is sufficient to show that $w_1, w_2 \in D \cap L(v)$ implies $w_1 a w_2 b \in D \cap L(v)$. Suppose $w_1, w_2 \in D \cap L(v)$. The word $w = a^4 w_1 a w_2 b b^4$ is from $D - a\boldsymbol{h}(\{a, b, c\}^*)b$. Indeed, w is in D and none of the words from $a\boldsymbol{h}(\{a, b, c\}^*)\, b$ contains the subword b^5. Consequently, w is from $L''_{x,y} = L(G)$. Then $w = x_1 x_2$ where $A \Rightarrow uavbz \Rightarrow^* x_1$ (an initial substring has to be generated using (1)). From (2) we obtain

$$S \Rightarrow^* A \Rightarrow^* x_1 \in L''_{x,y} \subseteq D$$

The equality $x_1 = w$ can be easily shown, as each initial substring of w except w itself contains more a's than b's what is impossible for a word from D. Thus each derivation of x_1 from $uavbz$ has to be of the form

(5) $\quad uavbz \Rightarrow^* av_3b \Rightarrow^* x_1 = w\,, \quad$ where $\quad v \Rightarrow^* v_3 \Rightarrow^* a^3w_1aw_2bb^3,\ v_3 \in V^*$

Since in (5) the production (1) has to be used again, the existence of the following derivations can be shown similarly:

$$v \Rightarrow^* a^2w_1aw_2bb^2$$
$$v \Rightarrow^* aw_1aw_2bb$$
$$v \Rightarrow^* w_1aw_2b$$

Therefore $D \subseteq L(v)$ and this completes the proof of lemma. □

The unsolvability of determining $PT(L(G))$ $(PT_\varepsilon(L(G)))$ can be formulated now in a stronger form.

Theorem 1.

(i) For no integer $n \geqq 1$ there is an algorithm to decide for a given grammar G whether or not $PT(L(G)) = n$ $(PT_\varepsilon(L(G)) = n + 1)$.

(ii) The problems $PT(L(G)) = 0$, $PT_\varepsilon(L(G)) = 0$, $PT_\varepsilon(L(G)) = 1$ are decidable.

Proof. (i) For $n = 1$ the theorem follows from Lemma 2. Let now $n \geqq 2$. Let $\Sigma_{n-1} = \{a_1, \ldots, a_{n-1}\}$ be an alphabet, such that $\Sigma_{n-1} \cap \{a, b\} = \emptyset$. Denote $L^n_{x,y} = L''_{x,y} \cup \Sigma_{n-1}$. Clearly, $PT(\Sigma_{n-1}) = PT_\varepsilon(\Sigma_{n-1}) = n - 1$. By the same reasoning as in the proof of Lemma 2 one can show that $PT(L^n_{x,y}) = n$ $(PT_\varepsilon(L^n_{x,y}) = n + 1)$ iff the Post correspondence problem has no solution for (x, y). (i) now follows from undecidability of the Post correspondence problem.

(ii) From Definition 1 we easily obtain

$$PT(L(G)) = 0 \quad \text{iff} \quad L(G) = \emptyset \quad \text{or} \quad L(G) = \{\varepsilon\}$$
$$PT_\varepsilon(L(G)) = 0 \quad \text{iff} \quad L(G) = \emptyset$$

Since $L(G) = \emptyset$ and $L(G) = \{\varepsilon\}$ is decidable we have decidability of the first two problems in (ii). The decidability of the third problem we get from the decidability of the inclusion $L(G) \subseteq w^*$ ([1]) since

(6) $\quad PT_\varepsilon(L(G)) = 1 \quad$ iff $\quad L(G) \neq \emptyset \quad$ and there is a word w such that

$$L(G) \subseteq w^+$$

The only trouble in the proof of (6) seems to be with the if-part for the case $w \neq \varepsilon$. Let $L(G) \subseteq w^+ \subseteq \Sigma^*$ with Σ being a finite alphabet, $w \neq \varepsilon$. Let $h_1 : a^* \to \Sigma^*$ be a homomorphism, $h_1(a) = w$. By [1] $h_1^{-1}(L(G))$ is a cfl not containing the empty

word. It is easy to see that

$$PT_\varepsilon(\mathrm{L(G)}) \leqq PT_\varepsilon(h_1^{-1}(\mathrm{L(G)}))$$

and by Lemma 1 $PT_\varepsilon(h_1^{-1}(\mathrm{L(G)})) = 1$. □

There are two interesting problems concerning the minimality of a cfg with respect to $PT(PT_\varepsilon)$. The undecidability of the first one is an immediate corollary of Lemma 2, the undecidability of the second one needs a short proof.

Corrollary 1. There is no algorithm to construct for a given cfg G an equivalent cfg G′ such that $PT(\mathrm{G}') = PT(\mathrm{L(G')})\,(PT_\varepsilon(\mathrm{G}') = PT_\varepsilon(\mathrm{L(G')}))$.

Theorem 2. There is no algorithm to decide for a given cfg G whether or not $PT(\mathrm{G}) = PT(\mathrm{L(G)})\,(PT_\varepsilon(\mathrm{G}) = PT_\varepsilon(\mathrm{L(G)}))$.

Proof. It is easily to see that given x and y a cfg G can be constructed such that $\mathrm{L(G)} = \mathrm{L}''_{x,y}$, $PT(\mathrm{G}) = 2$ and $PT_\varepsilon(\mathrm{G}) = 3$. It has been shown that $PT(\mathrm{L}''_{x,y}) < 2$ $(PT_\varepsilon(\mathrm{L}''_{x,y}) < 3)$ iff the Post correspondence problem for (x, y) has no solution. However, this implies that G is a minimal cfg with respect to the complexity measure $PT(PT_\varepsilon)$ iff the Post correspondence problem for (x, y) possesses a solution. The theorem now follows from undecidability of the Post correspondence problem.

4. END PRODUCTIONS

The structure of leaf-parts of derivation trees in cfg's depends on the number and form of productions with no variable in the right hand side. Thus the number of such "end productions" as the complexity measure for cfg's could be of some importance. Similarly, the minimal number of distinct left sides of the end productions gives us some information about the intrinsic complexity of the language.

Definition 2. Let $\mathrm{G} = (\mathrm{V}, \Sigma, \mathrm{P}, S)$ be a cfg. A production $A \to w$ will be called *end production* iff $w \in \Sigma^*$. A variable A will be called *end variable* iff there is an end production $A \to w$ in P.

Definition 3. Let $\mathrm{G} = (\mathrm{V}, \Sigma, \mathrm{P}, S)$ be a cfg. If G is a grammar in ε-reduced form, then

$$EP(\mathrm{G}) = \text{the number of end productions in P}$$

$$EV(\mathrm{G}) = \text{the number of end variables in P}$$

Otherwise

$$EP(\mathrm{G}) = EV(\mathrm{G}) = |\Sigma| + 1$$

Definition 3 needs some explanation. For an arbitrary grammar G an equivalent grammar G′ with no end productions can be constructed. Indeed, by concatenating

the right side of each production with a new symbol X and adding $X \to \varepsilon$ to the productions of G such a grammar is obtained. The separate definition of the measures EP and EV for grammars not being in ε-reduced form makes both measures non-trivial, as shown in Lemma 3.

Lemma 3. Let $n \geqq m \geqq 1$ be integers and $\Sigma_n = \{a_1, \ldots, a_n\}$ an alphabet. Then

(i) for any cfl $L \subseteq \Sigma_n^*$ $EV(L) \leqq EP(L) \leqq n$

(ii) there is a language $L_{m,n} \subseteq \Sigma_n^*$ such that $EV(L_{m,n}) = m$, $EP(L_{m,n}) = n$.

Proof. (i) is obvious. To prove (ii) denote $L_{m,n} = a_1^+ \cup \ldots \cup a_{m-1}^+ \cup a_m \cup \ldots \cup a_n$. This language is defined by the grammar with productions

$$S \to A_i \mid a_j, \quad i = 1, 2, \ldots, m-1$$

$$A_i \to a_i A_i \mid a_i, \quad j = m, m+1, \ldots, n$$

and therefore

$$EV(L_{m,n}) \leqq m, \quad EP(L_{m,n}) \leqq n \tag{7}$$

Let now $G = (V, \Sigma_n, P, S)$ be an arbitrary grammar in ε-reduced form generating $L_{m,n}$. $L_{m,n}$ contains at least one word from each of the languages a_i^+, $i = 1, 2, \ldots, n$. Hence for each integer $i = 1, 2, \ldots, n$ there is a production

$$B_i \to a_i^{k_i}, \quad k_i \geqq 1 \tag{8}$$

in P. Clearly, we may assume that $k_i = 1$ for $i = m, \ldots, n$ and that (8) is used in the derivation of some word $w_i \in a_i^+$, $|w_i| > k_i$ for $i = 1, 2, \ldots, m-1$.

From (8) we immediately get $EP(L_{m,n}) \geqq n$, and consequently, $EP(L_{m,n}) = n$ by (7).

Suppose there are integers r, s, $1 \leqq r < s \leqq m$, such that $B_r = B_s$. Then the following two derivations exist in G:

$$S \Rightarrow^* b_r^p B_r b_r^q \Rightarrow b_r^{p+q+k_r} = w_r \in L_{m,n}, \quad p + q > 0$$

$$S \Rightarrow^* b_r^p B_r b_r^q \Rightarrow b_r^p B_s b_r^q \Rightarrow b_r^p b_s^{k_s} b_r^q \in L_{m,n}$$

This contradicts to the fact that no word in $L_{m,n}$ contains two different symbols. Thus $EV(L_{m,n}) \geqq m$ and by (7) $EV(L_{m,n}) = m$. □

The decidability results for EP, EV and PT, PT_ε complexity measures are quite similar and the following lemma corresponds to Lemma 2.

Lemma 4. There is no algorithm to decide for an arbitrary cfg G whether or not $EP(L(G)) = 1$ $(EV(L(G)) = 1)$.

Proof. Denote $L'''_{x,y} = L'_{x,y}c$. In order to prove the lemma it is sufficient to show that $EP(L'''_{x,y}) = 1$ $(EV(L'''_{x,y}) = 1)$ iff the Post correspondence problem for (x, y) possesses no solution. To this end two cases will be considered.

I. If the Post correspondence problem for (x, y) has no solution, then $L'''_{x,y} =$

$= \boldsymbol{h}(\{a, b, c\}^*)\, c$ ($\boldsymbol{h}$ is the homomorphism defined in Section 2). This language is generated by the grammar $S \to abS \mid a^2b^2S \mid a^3b^3S \mid c$ and therefore $EP(\mathrm{L}'''_{x,y}) = EV(\mathrm{L}'''_{x,y}) = 1$.

II. Let the Post correspondence problem for (x, y) have a solution, i.e. there exist words $u, v, z, t \in \{a, b\}^*$, $z = v^{\mathrm{R}}$, $t = u^{\mathrm{R}}$, such that for no integer $m \geqq 1$ the word $\boldsymbol{h}(u^m c v^m c z^m c t^m)\, c$ is in $\mathrm{L}'''_{x,y}$. Suppose there exists a cfg $\mathrm{G} = (\mathrm{V}, \{a, b, c\}, \mathrm{P}, S)$ in ε-reduced form which generates $\mathrm{L}'''_{x,y}$, and $EV(\mathrm{G}) = 1$.

Since $c \in \mathrm{L}'''_{x,y}$, the production $A \to c$ where A is the only end variable of G has to be in P.

At first we shall show that $EV(\mathrm{G}) = 1$ implies that $\mathrm{L}'''_{x,y}$ is a regular language. Let

$$B \to w_1 C w_2 \tag{9}$$

be an arbitrary production from P used in some derivation of a terminal word, with $w_1 \in \{a, b, c\}^*$, i.e. C is the first variable from left in the right hand side of (9). Then there are words $w_3, \ldots, w_8 \in \mathrm{V}^*$ and the derivation

$$S \Rightarrow^* w_3 B w_4 \Rightarrow w_3 w_1 C w_2 w_4 \Rightarrow^* w_3 w_1 w_5 A w_6 w_2 w_4 \Rightarrow$$

$$\Rightarrow w_3 w_1 w_5 c w_6 w_2 w_4 \Rightarrow^* w_7 c w_8 \in \mathrm{L}'''_{x,y}$$

(A is the only end variable of G!). Since the only possible position for a symbol c in $\mathrm{L}'''_{x,y}$ is at the end of the words, we get $w_8 = w_2 = \varepsilon$. Hence every non-superfluous production of G is right-linear and $\mathrm{L}'''_{x,y}$ is a regular language.

However, from the existence of a solution of the Post correspondence problem for (x, y) the opposite follows. Indeed, by Theorem 5.6 of [5] if $\mathrm{L}'''_{x,y}$ is regular, then the equivalence relation $\mathbf{E}$ on $\{a, b, c\}^*$ induced by $\mathrm{L}'''_{x,y}$ is of finite index. Thus there exist integers $j, k \geqq 1$, $j \neq k$, such that $\boldsymbol{h}(u^j)\, \mathbf{E}\, \boldsymbol{h}(u^k)$. This contradicts to the fact that $\boldsymbol{h}(u^j)\, \boldsymbol{h}(cv^k cz^k ct^k)\, c$ is from $\mathrm{L}'''_{x,y}$ while $\boldsymbol{h}(u^k)\, \boldsymbol{h}(cv^k cz^k ct^k)\, c$ does not belong to $\mathrm{L}'''_{x,y}$. Consequently, grammar G with the described properties cannot exist, and $EP(\mathrm{L}'''_{x,y}) \geqq EV(\mathrm{L}'''_{x,y}) \geqq 2$. The lemma follows now from the undecidability of the Post correspondence problem. □

Using Lemma 4 the following theorem can be proven.

Theorem 3.

(i) For no integer $n \geqq 1$ there is an algorithm to decide for an arbitrary cfg G whether or not $EP(\mathrm{L}(\mathrm{G})) = n$ ($EV(\mathrm{L}(\mathrm{G})) = n$).

(ii) The problem $EP(\mathrm{L}(\mathrm{G})) = 0$ ($EV(\mathrm{L}(\mathrm{G})) = 0$) is decidable.

Proof. (i) For $n = 1$ the theorem follows from Lemma 4. Let $n \geqq 2$. Let $\Sigma_{n-1} = \{a_1, \ldots, a_{n-1}\}$ be an alphabet, $\Sigma_{n-1} \cap \{a, b, c\} = 0$. Denote $\mathrm{L}^n_{x,y} = \mathrm{L}'''_{x,y} \cup \mathrm{L}_{n-1}$ where $\mathrm{L}'''_{x,y}$ is the language from the proof of Lemma 4 and $\mathrm{L}_{n-1} = a_1^+ \cup \ldots \cup a_{n-1}^+$. Since the languages $\mathrm{L}'''_{x,y}$ and L_{n-1} are over distinct alphabets we get easily

$$\mathbf{K}(\mathrm{L}^n_{x,y}) = \mathbf{K}(\mathrm{L}'''_{x,y}) + \mathbf{K}(\mathrm{L}_{n-1}) \quad \text{for} \quad \mathbf{K} \in \{EP, EV\}\,.$$

By a similar reasoning as in the proofs of Lemma 3 and Lemma 4 we get $EP(L^n_{x,y}) = n$ ($EV(L^n_{x,y}) = n$) iff the Post correspondence problem for (x, y) has no solution. The theorem follows now from the undecidability of the Post correspondence problem.

(ii) Clearly, $EP(L(G)) = 0$ ($EV(L(G)) = 0$) iff $L(G) = \emptyset$ or $L(G) = \{\varepsilon\}$. The last two problems are known to be decidable. □

Corollary 2. There is no algorithm to construct for a given cfg G an equivalent cfg G′ minimal with respect to EP (EV).

The problem of minimality is undecidable for EP, EV, too:

Theorem 4. There is no algorithm to decide for a given cfg G whether or not $EP(G) = EP(L(G))$ ($EV(G) = EV(L(G))$).

Proof. Obviously, a cfg G in ε-reduced form generating $L_{x,y}$ can be constructed, with $A \to a$, $B \to b$, $C \to c$ as the only end productions. If these productions are replaced by the productions $A \to Db$, $B \to Dab^2$, $C \to Da^2b^3$, $D \to a$, a new grammar generating $L'_{x,y}$ is obtained. From this grammar by a small modification a grammar G′ generating $L'''_{x,y}$ can be obtained such that $EP(G') = EV(G') = 2$. From the proof of Lemma 4 it follows that G′ is minimal with respect to $EP(EV)$ iff the Post correspondence problem for (x, y) has a solution. The theorem now follows from the undecidability of the Post correspondence problem. □

To finish this section three another complexity measures similar to the measures EP and EV are defined and some results concerning these measures are summarized.

Definition 4. Let $G = (V, \Sigma, P, S)$ be a cfg. If G is in ε-reduced form, then

$PTB(G)$ = the number of productions in P with a terminal at the beginning of the right hand side

$PTE(G)$ = the number of productions in P with a terminal at the end of the right hand side

$PTBE(G)$ = the number of productions in P with a terminal at the beginning and at the end of the right hand side.

Otherwise

$PTB(G) = PTE(G) = PTBE(G) = |\Sigma| + 1$

Theorem 5.

(i) Let Σ be a finite alphabet, $L \subseteq \Sigma^*$ a cfl. Then
$EP(L) \leqq PTBE(L) \leqq PTB(L) \leqq |\Sigma|$
$PTBE(L) \leqq PTE(L) \leqq |\Sigma|$

(ii) Let $\mathbf{K} \in \{PTB, PTE, PTBE\}$. There is no algorithm

1. to decide for an arbitrary cfg G and integer $n \geqq 1$ whether or not $\mathbf{K}(L(G)) = n$
2. to decide for a given cfg G whether or not $\mathbf{K}(G) = K(L(G))$
3. to construct for a given cfg G and equivalent cfg G′ minimal with respect to $\mathbf{K}$.

5. COMPLEMENTARY MEASURES

Measures of complexity in Sections 2 and 3 have been defined according to the following schema:

For each cfg $G = (V, \Sigma, P, S)$ and each production $p \in P$ an integer $k(G, p) \in \in \{0, 1\}$ has been defined. The complexity of a cfg G has then been defined by

$$\mathbf{K}(G) = \sum_{p \in P} k(G, p)$$

In other words, only a part of productions of a cfg G contributes to the overall complexity of G. Once this point of view is accepted it is quite natural to investigate the so-called complementary complexity measures to which the remaining part of productions contributes. They are defined by

$$\mathbf{CK}(G) = \sum_{p \in P} (1 - k(G, p))$$

and will be discussed in this section.

Definition 5. For a cfg $G = (V, \Sigma, P, S)$ let us define

$CPT(G)$ = the number of productions in P without terminals in the right side
$CPT_\varepsilon(G)$ = $CPT(G)$ − the number of ε-productions in P
$CEP(G)$ = the number of productions in P not being end productions
$CEV(G)$ = the number of variables in V not being end variables
$CPTB(G)$ = the number of productions in P with the right hand side not beginning with a terminal
$CPTE(G)$ = the number of productions in P without terminal at the end of the right hand side
$CPTBE(G)$ = the number of productions in P with the right hand side not beginning or not ending by a terminal

The main results of this section are summarized in two theorems.

Theorem 6. There is no algorithm to determine $CEP(L(G))$ for an arbitrary cfg G.

Theorem 7. Let L be a cfl. Then

(i) If $L = \emptyset$ or $L = \{\varepsilon\}$ then $CEV(L) = 1$, otherwise $CEV(L) = 0$
(ii) $CPT_\varepsilon(L) = 0$
(iii) If $\varepsilon \in L$, then

$$CPT(L) = CPTB(L) = CPTE(L) = CPTBE(L) = 1$$

Otherwise

$$CPT(\mathrm{L}) = CPTB(\mathrm{L}) = CPTE(\mathrm{L}) = CPTBE(\mathrm{L}) = 0$$

Hence, the basic algorithmic problems are decidable for all complementary measures of Definition 5 with the exception of the measure *CEP*.

Proofs. The proof of Theorem 6 is similar to that of Lemma 2. Let us denote $\overline{\mathrm{L}}_{x,y} = \mathrm{L}_{x,y} - \{\varepsilon\}$ and let us consider two cases.

I. If the Post correspondence problem for (x, y) possesses no solution, then $\overline{\mathrm{L}}_{x,y} = \{a, b, c\}^+$. This language is generated by the grammar $S \to SS \mid a \mid b \mid c$ and therefore $CEP(\overline{\mathrm{L}}_{x,y}) = 1$.

II. Let the Post correspondence problem for (x, y) have a solution. Suppose there is a cfg G with $\mathrm{L(G)} = \overline{\mathrm{L}}_{x,y}$, $CEP(\mathrm{G}) = 1$. Then the start symbol is the only recursive variable in G. Now a grammar G′ equivalent to G can be constructed containing just one variable. However, this contradicts to Lemma 3.2 in [3], which says that at least two variables are necessary to generate $\overline{\mathrm{L}}_{x,y}$. It means that there is no grammar G for $\overline{\mathrm{L}}_{x,y}$ with $CEP(\mathrm{G}) = 1$ and therefore $CEP(\overline{\mathrm{L}}_{x,y}) \geqq 2$. Theorem 6 follows now from the undecidability of the Post correspondence problem.

If G is a cfg and A a variable of G such that $A \Rightarrow^* w \in \mathrm{L(G)}$, then by adding $A \to w$ to the productions of G an equivalent grammar is obtained. From that (i) of Theorem 7 follows easily. The parts (ii) and (iii) of Theorem 7 follow from the well known fact that to every cfg there exists an equivalent cfg G′ in ε-reduced form all productions of which are of the $A \to a$ or $A \to axb$ with a, b being terminals, or $S' \to \varepsilon$ where S' is the start symbol of G′. □

ACKNOWLEDGEMENT

The author is indebted to Dr. J. Gruska for valuable and stimulating discussions and for reading the manuscript.

(Received July 15, 1977.)

REFERENCES

[1] S. Ginsburg: The Mathematical Theory of Context-Free Languages. McGraw-Hill, New York 1966.

[2] J. Gruska: Some classifications of context-free languages. Information and Control *14* (1969), 2, 152–179.

[3] J. Gruska: Complexity and unambiguity of context-free grammars and languages. Information and Control *18* (1971), 5, 502–519.

[4] J. Gruska: Descriptional complexity of context-free languages. Proceedings MFCS' 73, High Tatras, 71–83.

[5] A. Salomaa: Formal Languages. Academic Press, New York and London 1973.

RNDr. Anton Černý, Katedra teoretickej kybernetiky Matematicko-fyzikálnej fakulty Univerzity Komenského (Department of Theoretical Cybernetics, Faculty of Mathematics and Physics — Comenius University), Mlynská dolina, 842 15 Bratislava. Czechoslovakia.