MATHEMATICAL PROBLEMS IN COMPUTATION THEORY
BANACH CENTER PUBLICATIONS, VOLUME 21
PWN – POLISH SCIENTIFIC PUBLISHERS
WARSAW 1988

EFFECTIVE CONSTRUCTIONS OF GRAMMARS

MIROSLAV NOVOTNÝ

Institute of Mathematics, Czechoslovak Academy of Sciences, Brno, Czechoslovakia

1. Motivation

In its beginning in the late fifties, theory of formal languages was motivated linguistically. From the algebraic point of view, a language is an ordered pair (V, L) where V is a finite set and L is a subset of the free monoid V^* over V. The elements in V^* are called *strings*, Λ denotes the empty string, xy the string obtained by concatenation of strings x and y; $|x|$ denotes the length of the string x. The elements in V are interpreted to be word-forms, the elements in L are considered to be correct sentences of the language. Hence, considering a language (V, L) we are in a position similar to that of a linguist who investigates an unknown language written in an unknown alphabet; he recognizes only word-forms and correct sentences but does not understand them.

Theory of formal languages developed in two directions. First of them, which can be called analytic, tried to formalize the fundamental linguistic notions as, e.g., morphologic categories [13] and syntactic configurations [5]. If we take an English sentence, e.g.,

LITTLE CHILDREN DRANK OFTEN GOOD MILK,

then it can be developed from a simpler sentence

CHILDREN DRANK

by replacing the string CHILDREN of length 1 by the string LITTLE CHILDREN of length 2 and, similarly, the string DRANK by the string DRANK OFTEN MILK, and the string MILK by the string GOOD MILK.

Then the string LITTLE CHILDREN is a syntactic configuration of English and CHILDREN is its resultant. Similarly, DRANK OFTEN MILK, GOOD MILK are syntactic configurations whose resultants are DRANK, MILK, respectively. Configurations and their resultants are characterized by the property that the resultant appearing in any correct sentence may be replaced by the corresponding configuration in such a way that the resulting string is a correct sentence; the reverse replacement of a

configuration by its resultant in a correct sentence leads to a correct sentence only under certain restrictions.

Thus, we imagine that we start with an arbitrary correct sentence containing no configuration. If it contains a resultant of a configuration, we replace the resultant by this configuration. If the resulting string contains a resultant, we replace it by the corresponding configuration. All strings obtained after a finite number of such steps are correct sentences.

Clearly, if starting with the string CHILDREN DRANK and if using the ordered pairs (CHILDREN, LITTLE CHILDREN), (DRANK, DRANK OFTEN MILK), (MILK, GOOD MILK) in the above described way, we obtain, e.g.,

(1) LITTLE LITTLE CHILDREN DRANK OFTEN MILK OFTEN GOOD GOOD GOOD MILK.

This and similar strings are consired to be correct. The string

(2) LITTLE GOOD CHILDREN DRANK OFTEN CACAO OFTEN SWEET GOOD FRESH MILK

is a correct sentence and string (1) is obtained from (2) by replacing any word-form by another that is able to play the same grammatical role in all correct sentences.

This idea leads us to the notion of a special [15] or a pure [4], [14], [10] grammar. A pure grammar is an ordered triple $G = \langle V, S, R \rangle$, where V, S, R, are finite sets and $S \subseteq V^*$, $R \subseteq V^* \times V^*$. The set $L(G)$ of all strings generated by G is obtained by starting with a string $s \in S$, by taking an ordered pair (or production) $(y, x) \in R$ such that s contains y, i.e., $s = uyv$ for some $u, v \in V^*$, and by replacing y by x, i.e., by forming the string uxv, then by taking another production $(y', x') \in R$ such that uxv contains y' and by replacing y' by the string x', and so on; all strings obtained in this way form the set $L(G)$. Then $(V, L(G))$ is said to be the *language generated by G*.

In [5], [16], [17] some constructions of pure grammars for certain families of languages can be found.

It is very surprising that no attempts of formalizing syntactic categories can be found in the theory of formal languages till the end of seventies.

The synthetic direction in formal language theory dealt particularly with classification of grammars (not only of pure grammars), with relations of grammars to automata. In this direction a language is supposed to be generated by a grammar [1], [2], [3]. Grammars proved to be very important tools for programming languages and formal language theory became one of the theoretical backgrounds of computer science [9], [22]. This implies that effective methods were preferred in the study of formal languages. Since the results obtained in the analytic direction were obtained mostly by non-effective methods, both directions developed almost independently.

A group of mathematicians in Brno tried to find regions of common interest between the analytic and synthetic direction. Various methods of constructing grammars (in non-effective ways) were found [16], [17]. Syntactic categories of formal languages were introduced and grammars were defined on the basis of these categories for certain families of languages [11], [12].

Some variants of effective constructions of grammars were described [19], [18]. One of these constructions will be studied in this lecture. The main idea is as follows. A pure grammar $G_k(V, L)$ is assigned to any finite language (V, L) and to any integer k that is large enough in an effective way where the grammar $G_k(V, L)$ need not generate (V, L). If (V, L) is an arbitrary language and k a non-negative integer, then we define the set kL to be the set of strings in L of length $\leqslant k$; then (V, kL) is a finite language called the kth fragment of (V, L). The following can be proved. If there exists an integer k_0 such that $G_{k_0}(V, k_0 L) = G_k(V, kL)$ for any $k \geqslant k_0$, then $G_{k_0}(V, k_0 L)$ generates (V, L) and (V, L) belongs to a certain family $\mathscr{B}^m$. If $(V, L) \in \mathscr{B}^m$, then there exists k_0 such that $G_k(V, kL) = G_{k_0}(V, k_0 L)$ for any $k \geqslant k_0$. Thus, if we know the number k_0, we have an effective construction of a grammar for a language $(V, L) \in \mathscr{B}^m$ such that the grammar generates the language.

The above results can be used in syntactic pattern recognition [6].

2. Pure grammars

Let V be a finite non-empty set. An ordered pair $(y, x) \in V^* \times V^*$ is called a *production* over V. We put

$$\left|(y, x)\right| = \max\{|y|, |x|\}.$$

For any $s, t \in V^*$, we set $s \Rightarrow t$ $(\{(y, x)\})$ if there exist $u, v \in V^*$ such that $s = uyv$, $uxv = t$.

Let us have $R \subseteq V^* \times V^*$. We put $s \Rightarrow t$ (R) if there exists $(y, x) \in R$ such that $s \Rightarrow t$ $(\{(y, x)\})$. If $s \Rightarrow t$ (R), we can have more productions $(y, x) \in R$ such that $s \Rightarrow t$ $(\{(y, x)\})$. We put

$$|(s, t)|_R = \min\left\{\left|(y, x)\right|;\ (y, x) \in R \text{ and } s \Rightarrow t\ (\{(y, x)\})\right\}.$$

Let $s, t \in V^*$, $n \geqslant 0$, $s_0, s_1, \ldots, s_n \in V^*$ be such that $s = s_0$, $s_n = t$, $s_{i-1} \Rightarrow s_i$ (R) for $i = 1, \ldots, n$. Then the finite sequence $(s_i)_{i=0}^n$ is said to be an *s-derivation* of t in R. We put

$$\|(s_i)_{i=0}^n\|_R = \begin{cases} 0 & \text{if } n = 0, \\ \max\{|(s_{i-1}, s_i)|_R;\ i = 1, \ldots, n\} & \text{if } n > 0. \end{cases}$$

Finally, we put $s \overset{*}{\Rightarrow} t$ (R) if there exists an s-derivation of t in R and $s \overset{+}{\Rightarrow} t$ (R) if $s \overset{*}{\Rightarrow} t$ (R) and $s \neq t$. We define

$$\|(s, t)\|_R = \min \{\|(s_i)_{i=0}^n\|_R;\ (s_i)_{i=0}^n \text{ is an } s\text{-derivation of } t \text{ in } R\}.$$

Let V be a finite non-empty set, $S \subseteq V^*$, $R \subseteq V^* \times V^*$. Then the ordered triple $G = \langle V, S, R\rangle$ is said to be a *generalized pure grammar* [15], [16], [17], [18], [19]; it is said to be a *pure grammar* if the sets S, R are finite [4], [10], [14].

If $G = \langle V, S, R\rangle$ is a generalized pure grammar, we set

$$L(G) = \{w \in V^*;\ \text{there exists } s \in S \text{ with } s \overset{*}{\Rightarrow} w\ (R)\}.$$

Then $(V, L(G))$ is said to be the *language generated by* G. For any $z \in L(G)$, we put

$$\|z\|_R^S = \min \{\|(s, z)\|_R;\ s \in S,\ s \overset{*}{\Rightarrow} z\ (R)\}.$$

We shall investigate pure grammars of special type. We put

$$K^m(V) = \{(y, x) \in V^* \times V^*;\ 1 \leqslant |y| < |x|\}.$$

Ordered pairs in $K^m(V)$ are said to be *monotone productions*. A generalized pure grammar $\langle V, S, R\rangle$ is said to *have monotone productions* if $R \subseteq K^m(V)$. We denote by $\mathscr{B}^m$ the family of languages generated by pure grammars with monotone productions.

Let (V, L) be an arbitrary language. We put

$$N(V, L) = \{x \in V^*;\ \text{there exist } u, v \in V^* \text{ such that } uxv \in L\},$$

$$D(V, L) = \{(y, x) \in V^* \times V^*;\ \text{for any } u, v \in V^*,\ uyv \in L \text{ implies } uxv \in L\},$$

$$E(V, L) = D(V, L) \cap (D(V, L))^{-1},$$

$$D^m(V, L) = (N(V, L) \times N(V, L)) \cap D(V, L) \cap K^m(V).$$

The strings in $N(V, L)$ are said to be *Necessary* in (V, L), $D(V, L)$ is called *Domination* relation for (V, L), $E(V, L)$ is an Equivalence on V^*.

Any language can be generated by a generalized pure grammar.

1. Example. Let (V, L) be a language. Then $\langle V, L, D^m(V, L)\rangle$ generates (V, L).

Clearly, $L(\langle V, L, D^m(V, L)\rangle) \supseteq L$. On the other hand, if $s \in L$, $t \in V^*$, $s \overset{*}{\Rightarrow} t$ $(D^m(V, L))$, then $s \overset{*}{\Rightarrow} t$ $(D(V, L))$ which implies $t \in L$. Thus, $L(\langle V, L, D^m(V, L)\rangle) \subseteq L$. ∎

We now characterize languages in $\mathscr{B}^m$.

2. Theorem. *Let (V, L) be a language. Then $(V, L) \in \mathscr{B}^m$ if and only if there exists a pure grammar $G = \langle V, S, R\rangle$ generating (V, L) such that $S \subseteq L$ and $R \subseteq D^m(V, L)$.*

Proof. If S and R have the above mentioned properties, then $(V, L) = (V, L(G)) \in \mathscr{B}^m$.

Let $G = \langle V, S, R \rangle$ be a pure grammar generating (V, L) such that $R \subseteq K^m(V)$. Clearly, $S \subseteq L(G) = L$.

If $(y, x) \in R$ and $y \notin N(V, L)$, then the production (y, x) cannot be used for generating a string $z \in L(G) = L$. Thus, $\langle V, S, R - \{(y, x)\} \rangle$ generates the same language as $\langle V, S, R \rangle$ does. Thus, we can suppose that $y \in N(V, L)$. If $u, v \in V^*$ are such that $uyv \in L = L(G)$, there exists $s \in S$ with $s \overset{*}{\Rightarrow} uyv$ (R). Since $uyv \Rightarrow uxv$ (R), we obtain $s \overset{*}{\Rightarrow} uxv$ (R) which implies that $uxv \in L(G) = L$. Thus, we have $x \in N(V, L)$ and $(y, x) \in D(V, L)$ which implies that $(y, x) \in D^m(V, L)$. We have proved that $R \subseteq D^m(V, L)$. ∎

We now incorporate the family $\mathscr{B}^m$ into the usual hierarchy of languages. Let $\mathscr{R}$ denote the family of all regular languages, $\mathscr{CF}$ the family of all context-free languages.

3. THEOREM. $\mathscr{R} \subseteq \mathscr{B}^m$, $\mathscr{R} \neq \mathscr{B}^m$. (Cf. [16].)

Proof. (1) Let (V, L) be a non-empty regular language. There exists a positive integer N such that $E(V, L)$ has exactly N blocks. We set $S = \{x \in L;\ |x| \leqslant N\}$, $R = \{(y, x) \in D(V, L) \cap K^m(V);\ |x| \leqslant N+1\}$. Then $G = \langle V, S, R \rangle$ is a pure grammar with monotone productions.

Let $t \in L$ be arbitrary. If $|t| \leqslant N$, then $t \in S \subseteq L(G)$. If $n = |t| > N$, there exist $a_1, \ldots, a_n$ in V such that $t = a_1 \ldots a_n$. We set $t_i = a_1 \ldots a_i$ for any i with $1 \leqslant i \leqslant n$. There exist indices i, j such that $1 \leqslant i < j \leqslant N+1$, $(t_i, t_j) \in E(V, L)$. Hence, $|t_i| = i < j = |t_j| \leqslant N+1$ and, thus, $(t_i, t_j) \in R$. We put $z = a_{j+1} \ldots a_n$, $s = t_i z$. Since $t_j z = t \in L$ and $(t_j, t_i) \in D(V, L)$, we obtain $s \in L$. Furthermore, $s \Rightarrow t$ (R) and $|s| < |t|$. Thus, to any $t \in L$ with $|t| > N$, there exists $s \in L$ with $|s| < |t|$, $s \Rightarrow t$ (R). Hence, to any $t \in L$ with $|t| > N$, there exists $s \in L$ with $|s| \leqslant N$ such that $s \overset{*}{\Rightarrow} t$ (R). Thus, $s \in S$ and $t \in L(G)$.

We have proved that $L \subseteq L(G)$.

On the other hand, $S \subseteq L$, $R \subseteq D(V, L)$ imply $L(G) \subseteq L$.

We have proved that $L(G) = L$ and, hence, $\mathscr{R} \subseteq \mathscr{B}^m$.

(2) Clearly, if $V = \{a, b, c\}$, $L = \{a^i bc^{2i};\ i \geqslant 0\}$, then $(V, L) \notin \mathscr{R}$ but it is easy to see that $(V, L) \in \mathscr{B}^m$. ∎

4. THEOREM. $\mathscr{B}^m - \mathscr{CF} \neq \emptyset$, $\mathscr{CF} - \mathscr{B}^m \neq \emptyset$. (Cf. [16].)

Sketch of proof. (1) Put

$$V = \{a, b, c, d\},\quad L' = \{ba^{2k}ca^{2^{2n}-k}b;\ n \geqslant 0,\ 1 \leqslant k \leqslant 2^{2n}\},$$

$$L'' = \{ba^{2^{2n+1}-k}da^{2k}b;\ n \geqslant 0,\ 1 \leqslant k \leqslant 2^{2n+1}\},$$

$$L = L' \cup L'',$$

$$S = \{ba^2cb\},\quad R = \{(acb, da^2b), (ad, da^2), (bda, ba^2c), (ca, a^2c)\},$$

$$G = \langle V, S, R \rangle.$$

Then G generates (V, L) and, hence, $(V, L) \in \mathscr{B}^m$. Using the well-known pumping lemma, we prove $(V, L) \notin \mathscr{C}\mathscr{F}$.

(2) Put $V = \{a, b\}$, $L = \{x\tilde{x};\ x \in V^*\}$, where $\tilde{x}$ denotes the mirror image of x. Then $D(V, L) = \mathrm{id}_{V^*}$ and, thus $D^m(V, L) = \emptyset$ which implies $(V, L) \notin \mathscr{B}^m$. Clearly $(V, L) \in \mathscr{C}\mathscr{F}$. ∎

3. Effective constructions of grammars

A language (V, L) is said to be *finite* if the set L is finite.

To any finite language (V, L) and to any integer k that is large enough, we assign a pure grammar in an effective way (cf. [19], [18]). We put

$$m(V, L) = \begin{cases} 0 & \text{if } L = \emptyset, \\ \max\{|z|;\ z \in L\} & \text{if } L \neq \emptyset, \end{cases}$$

we choose an integer $k \geqslant m(V, L)$ and we set

$$b_k(V, L) = \{(y, x) \in V^* \times V^*;\ \text{for any } u, v \in V^*,\ uyv \in L \text{ implies either } uxv \in L \text{ or } |uxv| > k\},$$

$$(n \times n)(V, L) = \{(y, x) \in V^* \times V^*;\ \text{there exist } u, v \in V^* \text{ with } uyv \in L,\ uxv \in L\},$$

$$d_k(V, L) = (n \times n)(V, L) \cap b_k(V, L) \cap K^m(V),$$

$$B_k(V, L) = \{s \in L;\ t \in L,\ t \overset{*}{\Rightarrow} s\ (d_k(V, L)) \text{ imply } |t| \geqslant |s|\},$$

$$X_k(V, L) = \{(y, x) \in d_k(V, L);\ \text{there exists } z \in L \text{ with } |(y, x)| \leqslant \|z\|_{d_k(V,L)}^{B_k(V,L)}\},$$

$$G_k(V, L) = \langle V, B_k(V, L), X_k(V, L)\rangle.$$

The reader may observe that the sets $b_k(V, L)$, $(n \times n)(V, L)$, $d_k(V, L)$ are analogous to the sets $D(V, L)$, $N(V, L) \times N(V, L)$, $D^m(V, L)$, respectively. Furthermore, $G_k(V, L)$ is a pure grammar that can be constructed effectively if the finite sets, V, L and the integer k are given. We give an illustration.

1. Example. Let us have $V = \{a, b\}$, $L(i) = \{ab^{j-1};\ 0 < j \leqslant i\}$ for any $i \geqslant 3$. We obtain $m(V, L(i)) = i$, $N(V, L(i)) = \{\Lambda, a, b, ab, b^2, ab^2, b^3, \ldots, b^{i-1}, ab^{i-1}\}$. For any $(y, x) \in (N(V, L(i)) \times N(V, L(i))) \cap K^m(V)$ we test whether it belongs to $b_i(V, L(i))$; in the positive case, we test whether the set $M(y, x) = \{(z, t) \in L(i) \times L(i);\ z \Rightarrow t\ (\{(y, x)\})\}$ is non-empty. The pairs satisfying these conditions form the set $d_i(V, L(i))$. We obtain $d_i(V, L(i)) = \{(ab^s, ab^r);\ 0 \leqslant s < r \leqslant i-1\} \cup \{(b^s, b^r);\ 1 \leqslant s < r \leqslant i-1\}$, $M(ab^s, ab^r) = \{(ab^s, ab^r), \ldots, (ab^{i-1+s-r}, ab^{i-1})\}$, $|(ab^s, ab^r)| = r+1$ for any s, r with $0 \leqslant s < r \leqslant i-1$, $M(b^s, b^r) = \{(ab^s, ab^r), \ldots, (ab^{i-1+s-r}, ab^{i-1})\}$, $|(b^s, b^r)| = r$ for any s, r with $1 \leqslant s < r \leqslant i-1$. It follows that

$$\{(z, t) \in L(i) \times L(i);\ z \Rightarrow t(d_i(V, L))\} = \{(ab^s, ab^r);\ 0 \leqslant s < r \leqslant i-1\}$$
$$= \{(z, t) \in L(i) \times L(i);\ z \overset{+}{\Rightarrow} t(d_i(V, L))\}$$
$$= \{(z, t) \in L(i) \times L(i);\ z \overset{+}{\Rightarrow} t(\{(a, ab), (b, b^2)\})\}.$$

This implies that $B_i(V, L(i)) = \{a\}$, $X_i(V, L(i)) = \{(a, ab), (b, b^2)\}$, $G_i(V, L(i)) = \langle \{a, b\}, \{a\}, \{(a, ab), (b, b^2)\} \rangle$ for any $i \geqslant 3$. Clearly, $G_i(V, L(i))$ generates the language (V, L) such that $L = \{ab^i;\ i \geqslant 0\}$ which is in $\mathscr{B}^m$. ∎

2. THEOREM. *Let (V, L) be a finite language, $k \geqslant m(V, L)$ an integer. Then* $L \subseteq L(G_k(V, L))$.

Proof. If $z \in L$ and $s \overset{+}{\Rightarrow} z$ $(d_k(V, L))$ does not hold for any $s \in L$, then $z \in B_k(V, L) \subseteq L(G_k(V, L))$. If there exists $s \in L$ with $s \overset{+}{\Rightarrow} z$ $(d_k(V, L))$, we take such an s with minimal length. Then $s \in B_k(V, L)$. Among all $s \in B_k(V, L)$ with $s \overset{+}{\Rightarrow} z$ $(d_k(V, L))$, there exists s_0 such that $\|(s_0, z)\|_{d_k(V,L)}$ is minimal; clearly

$$\|(s_0, z)\|_{d_k(V,L)} = \|z\|_{d_k(V,L)}^{B_k(V,L)} \text{ which implies that } s_0 \overset{+}{\Rightarrow} z \ (X_k(V, L))$$

and, therefore, $z \in L(G_k(V, L))$. ∎

The effective construction of $G_k(V, L)$ can be given in the form of an algorithm or of a program. This program was written in the language ASSEMBLER for the computer EC 1021.

This construction will be applied to fragments of an arbitrary language. To this aim, we introduce some new symbols. We denote by N the set of all non-negative integers and by P the set of all positive integers. For any language (V, L) and any $k \in N$, we put

$$kL = \{x \in L;\ |x| \leqslant k\}, \quad a_k(V) = \{(y, x) \in V^* \times V^*;\ |y| \leqslant k,\ |x| \leqslant k\}.$$

Our results are based on the following

3. THEOREM. *Let (V, L) be a language. For any $i \in N$, there exists $v(i) \in N$ such that $v(i) \geqslant i$ and that $D^m(V, L) \cap a_i(V) = d_k(V, kL) \cap a_i(V)$ for any $k \geqslant v(i)$.*

Proof. Let us have $(y, x) \in (N(V, L) \times N(V, L)) \cap K^m(V) \cap a_i(V)$. Then we have two possibilities.

(A) $(y, x) \in D(V, L)$.

Then $(y, x) \in b_k(V, kL)$ for any $k \in N$. Since $y \in N(V, L)$, there exists $u, v \in V^*$ such that $uyv \in L$, $uxv \in L$. We take u, v in such a way that uxv has the least possible length and we put $i(y, x) = |uxv|$. For any $k \geqslant i(y, x)$, we have $(y, x) \in (n \times n)(V, kL)$. Furthermore, we put $j(y, x) = 0$.

(B) $(y, x) \notin D(V, L)$.

Then there exists $(u, v) \in V^* \times V^*$ such that $uyv \in L$, $uxv \notin L$. We take u, v in such a way that uxv has the least possible length and we put $j(y, x) = |uxv|$.

For any $k \geqslant j(y, x)$, we have $(y, x) \notin b_k(V, kL)$. Furthermore, we put $i(y, x) = 0$.

We set $I = 0 = J$ if the set $(N(V, L) \times N(V, L)) \cap K^m(V) \cap a_i(V)$ is empty and

$$I = \max \{i(y, x);\ (y, x) \in (N(V, L) \times N(V, L)) \cap K^m(V) \cap a_i(V)\},$$

$$J = \max \{j(y, x);\ (y, x) \in (N(V, L) \times N(V, L)) \cap K^m(V) \cap a_i(V)\}$$

if it is non-empty; furthermore, we put

$$v(i) = \max \{i, I, J\}.$$

Since $K^m(V) \cap a_i(V)$ is a finite set, $v(i)$ is correctly defined. Then, for any $k \geqslant v(i)$ and any $(y, x) \in (N(V, L) \times N(V, L)) \cap K^m(V) \cap a_i(V)$, we have $(y, x) \in D(V, L)$ if and only if $(y, x) \in d_k(V, kL)$. It follows that

$$\begin{aligned} D^m(V, L) \cap a_i(V) &= (N(V, L) \times N(V, L)) \cap K^m(V) \cap a_i(V) \cap D(V, L) \\ &= (N(V, L) \times N(V, L)) \cap K^m(V) \cap a_i(V) \cap d_k(V, kL) \\ &= d_k(V, kL) \cap a_i(V) \end{aligned}$$

because $d_k(V, kL) \subseteq K^m(V)$ and $d_k(V, kL) \subseteq (n \times n)(V, kL) \subseteq N(V, L) \times N(V, L)$. ∎

4. THEOREM. *Let (V, L) be a language such that there exists $k_0 \in N$ with the property $G_k(V, kL) = G_{k_0}(V, k_0 L)$ for any $k \geqslant k_0$. Then $(V, L) \in \mathscr{B}^m$ and the pure grammar $G_{k_0}(V, k_0 L)$ generates (V, L).*

Proof. By 2, we have $kL \subseteq kL(G_k(V, kL))$ for any $k \in N$. Let us have $k \geqslant k_0$, $t \geqslant v(k)$, $z \in kL(G_k(V, kL)) = kL(G_t(V, tL))$. Since $G_t(V, tL) = \langle V, B_t(V, tL), X_t(V, tL)\rangle$, we have $|z| \leqslant k$ and there exists $s \in B_t(V, tL)$ such that $s \overset{*}{\Rightarrow} z$ $(X_t(V, tL))$. Thus, $s \overset{*}{\Rightarrow} z$ $(d_t(V, tL))$, $\|(s, z)\|_{d_t(V, tL)} \leqslant k$, i.e., we have $s \overset{*}{\Rightarrow} z$ $(d_t(V, tL) \cap a_k(V))$. By 3, it follows that $s \overset{*}{\Rightarrow} z$ $(D^m(V, L) \cap a_k(V))$. Since $s \in tL \subseteq L$, we have $z \in L(\langle V, L, D^m(V, L)\rangle)$ which implies that $z \in kL$ by 2.1.

We have proved that $kL(G_k(V, kL)) = kL$ for any $k \geqslant k_0$.

Thus,

$$\begin{aligned} L &= \bigcup_{k \geqslant k_0} kL = \bigcup_{k \geqslant k_0} kL(G_k(V, kL)) = \bigcup_{k \geqslant k_0} kL(G_{k_0}(V, k_0 L)) \\ &= L(G_{k_0}(V, k_0 L)). \end{aligned}$$

Hence, (V, L) is generated by the pure grammar $G_{k_0}(V, k_0 L) = \langle V, B_{k_0}(V, k_0 L), X_{k_0}(V, k_0 L)\rangle$, where $X_{k_0}(V, k_0 L) \subseteq d_{k_0}(V, k_0 L) \subseteq K^m(V)$. Thus, $(V, L) \in \mathscr{B}^m$. ∎

5. EXAMPLE. We have proved that $G_k(V, kL) = G_3(V, 3L)$ for any $k \geqslant 3$

where $V = \{a, b\}$, $L = \{ab^i;\ i \geq 0\}$; cf. Example 1. Thus $G_3(V, 3L)$ generates (V, L). ∎

The last theorem can be reversed.

6. THEOREM. *Let (V, L) be a language. Then the following two assertions are equivalent.*

(i) $(V, L) \in \mathscr{B}^m$.

(ii) *There exists $k_0 \in N$ such that $G_k(V, kL) = G_{k_0}(V, k_0 L)$ for any $k \geq k_0$* (cf. [19].) ∎

4. Two complexity measures

We have seen that we obtain an effective construction of a pure grammar $G_{k_0}(V, k_0 L)$ for a language $(V, L) \in \mathscr{B}^m$ if we know the number k_0 appearing in 3.4. The number k_0 is not defined uniquely; thus, for any $(V, L) \in \mathscr{B}^m$, we put

$$Z^m(V, L) = \min \{k_0 \in \boldsymbol{P};\ G_k(V, kL) = G_{k_0}(V, k_0 L) \text{ for all } k \geq k_0\}.$$

If we have two languages (V, L_1), (V, L_2) in $\mathscr{B}^m$ such that $k_1 = Z^m(V, L_1)$ is small and $k_2 = Z^m(V, L_2)$ is large, then, for the construction of the pure grammar $G_{k_1}(V, k_1 L_1)$, we operate only with strings in L_1 of length $\leq k_1$ while for the construction of $G_{k_2}(V, k_2 L_2)$ we need strings in L_2 of length $\leq k_2$. From this point of view, the first language is simpler than the second. Thus, the function $Z^m(V, L)$ defined for $(V, L) \in \mathscr{B}^m$ can be considered to be a complexity measure of (V, L).

The set $\{k \in N;\ k < Z^m(V, L)\}$ can be divided into intervals such that $G_k(V, kL)$ is constant on any of them; some of these intervals may have length equal to 0. If we know the maximum of these lengths – say p – then $Z^m(V, L)$ has been found whenever we know a number $k > 0$ such that $G_k(V, kL) = G_{k+1}(V, (k+1)L) = \ldots = G_{k+p+1}(V, (k+p+1)L)$; clearly, $Z^m(V, L) \leq k$. Furthermore, if (V, L_1), (V, L_2) in $\mathscr{B}^m$ are given such that the above mentioned maximum p has a small value p_1 for (V, L_1) and a large value p_2 for (V, L_2), then (V, L_1) is simpler than (V, L_2) because looking for $Z^m(V, L_1)$ we test only short intervals of length $\leq p_1$ while looking for $Z^m(V, L_2)$ we have to test also large intervals of length p_2. From this point of view, this p is a complexity measure for $(V, L) \in \mathscr{B}^m$. The exact definition is as follows.

For any $(V, L) \in \mathscr{B}^m$ we put

$$P^m(V, L) = \begin{cases} 0 & \text{if } Z^m(V, L) = 1, \\ \max \{t \in N;\ \text{there exists } k \in \boldsymbol{P} \text{ with } k < Z^m(V, L) - t \text{ such} \\ \qquad \text{that } G_k(V, kL) = G_{k+i}(V, (k+i)L) \text{ for any } i \in N \\ \qquad \text{with } 0 \leq i \leq t\} & \text{if } Z^m(V, L) > 1. \end{cases}$$

The usual problems concerning complexity measure of languages can be formulated for $Z^m(V, L)$ and $P^m(V, L)$ (cf. [7], [8], [20]). The first of them is as follows.

P1. Are there languages $(V, L) \in \mathscr{B}^m$ with arbitrarily large $Z^m(V, L)$ $[P^m(V, L)]$?

Particularly, if the answer to this question were negative, there would exist an integer c such that $Z^m(V, L) \leqslant c$ for any $(V, L) \in \mathscr{B}^m$; thus the effectively constructed grammar $G_c(V, cL)$ would generate (V, L) for any $(V, L) \in \mathscr{B}^m$. Unfortunately, the answer is positive as it follows from the solution of the following problem.

P2. For arbitrary $i \in \boldsymbol{P}$ $[i \in \boldsymbol{N}]$ is there $(V, L) \in \mathscr{B}^m$ such that $Z^m(V, L) = i$ $[P^m(V, L) = i]$?

1. THEOREM. *Problem* P2 *has a positive solution.* (Cf. [21].) ∎

Thus, if there is $k \in \boldsymbol{P}$ and a very large $p \in \boldsymbol{N}$ such that $G_k(V, kL) = G_{k+1}(V, (k+1)L) = \ldots = G_{k+p}(V, (k+p)L)$ for some language (V, L), we have no mathematical reason to believe that $(V, L) \in \mathscr{B}^m$ and that $Z^m(V, L) \leqslant k$. Thus, we may try to find a subfamily of $\mathscr{B}^m$ such that finding $Z^m(V, L)$ is simpler in the subfamily than in $\mathscr{B}^m$. For this reason, we introduce the following family of languages.

A language $(V, L) \in \mathscr{B}^m$ is said to be *faithful* if $P^m(V, L) = 0$. This means that either $Z^m(V, L) = 1$ or $Z^m(V, L) > 1$ and $G_k(V, kL) \neq G_{k+1}(V, (k+1)L)$ for any k with $1 \leqslant k < Z^m(V, L)$, i.e., $Z^m(V, L)$ is the least integer $k \geqslant 1$ such that $G_k(V, kL) = G_{k+1}(V, (k+1)L)$. Regarding this property, faithful languages are of some interest. Particularly, if some simple languages prove to be faithful and if the family of faithful languages is closed under some operations with languages, then we could obtain faithful languages (V, L) of more complex structure without testing the condition $P^m(V, L) = 0$ directly. Unfortunately

2. THEOREM. *The family of faithful languages is an anti-AFL that is not closed under intersections.* (Cf. [21].) ∎

These results may provoke a pessimistic prognose concerning applications of the results contained in Section 3. We shall demonstrate that this pessimism is not justified.

5. Applications

The results of Section 3 can be applied in artificial intelligence. A typical problem of artificial intelligence is the following. A set of objects and a classification of these objects are given where the classification is supposed to have a finite number of blocks. The problem is to find the block corresponding to any object. An example is the problem of finding a botanical name of

any given plant. There exist well-known algorithms solving this problem in the framework of botany. Anybody intending to use such algorithm must be an expert in botany, must know various types of leaves, flowers, etc. Artificial intelligence tries to solve this and similar problems without the use of special knowledge, i.e., algorithms of artificial intelligence need not experts and can be performed by means of computers. This is based on the idea to replace real objects and their classification by mathematical objects (patterns) and their classification. E.g., classification of three types A, B, C of the botanical genus Iris was replaced by the classification of two-dimensional vectors (cf. [6], p. 4); a set of individuals was classified by experts and the petal length and petal width was stated for any individual. Thus, any individual was represented by a point in the plane and a line separating points corresponding to the type A from the points corresponding to the types B, C was found. Hence, if an unknown individual is to classify it suffices to find the corresponding point and to state whether it is in the region corresponding to the type A or in the region corresponding to the types B, C. Thus, the type can be stated mechanically without any special botanical knowledge.

Syntactic pattern recognition consists in replacing real objects by mathematical objects composed of a finite number of primitives; the set of all primitives is supposed to be finite. In the simplest case a real objects is replaced by a string of primitives; generally, the collection of primitives corresponding to an object has a more complicated structure (e.g., a tree with labelled nodes, etc.). In the simplest case, a block of strings corresponds to any block of real objects. Hence any block of real objects defines a language which is supposed to be generated by a grammar. If we know the grammar corresponding to any block and if a string corresponding to an object is given, we start with the grammar corresponding to the first block. We test whether the string is generated by the grammar. If the answer is positive, the problem is solved, the object belongs to the first block. In the negative case, we go to the second grammar and we continue in this way.

Two types of chromosomes were described in the literature (cf. [6], p. 33) by means of their contour lines. These contour lines can be obtained as composites of primitive arcs of five types and, therefore, any chromosome can be expressed as a string formed of five symbols. The chromosomes are of various extents and a language corresponds to the set of chromosomes of the first type; another language is assigned to the set of chromosomes of the second type. Grammars generating these languages were constructed on the basis of observing the languages. The methods of syntactic analysis lead to decision whether a given chromosome belongs to the first or to the second type.

Naturally, there is a serious problem how to obtain the grammar generating a given language. If a language (V, L) is given, we usually suppose that only a finite subset $F \subseteq L$ is to our disposal. On the basis of the finite

language (V, F) a grammar generating an infinite language (V, L') with $F \subseteq L'$ is to be constructed; since the language (V, L) is not known it is supposed to coincide with (V, L'). This is the so-called Grammatical Inference Problem. Several heuristic algorithms solving this problem are known from the literature (cf. [6], Chapter 6). Our grammar $G_k(V, L)$ assigned to a finite language (V, L) and an integer $k = m(V, L)$ can be considered to be a solution of this problem, too.

Suppose that a language (V, L) belongs to $\mathscr{B}^m$. Theorems 3.4 and 3.6 express the fact that the effectively constructed grammar $G_k(V, kL)$ generates (V, L) if k is large enough. If the language (V, L) corresponds to a set of real objects, we have a reason to believe that this k has been obtained if $G_k(V, kL) = G_{k+1}(V, (k+1)L) = \ldots = G_{k+p}(V, (k+p)L)$ for a large number p.

1. Example. Let a, ab, ab^2 be schemes of some observed organisms. We set $L = \{a, ab, ab^2\}$, $V = \{a, b\}$. Using 3.1, we construct $G_3(V, L) = \langle\{a, b\}, \{a\}, \{(a, ab), (b, b^2)\}\rangle$. We can have biological reasons to believe that strings corresponding to all observed organisms form a set that can be described by means of a language (V, M) such that $G_i(V, iM) = G_3(V, 3M) = G_3(V, L)$ for any $i \geqslant 3$. Then $M = \{ab^i;\ i \geqslant 0\}$ and all strings in M can be easily recognized. ∎

Another possibility of applying the construction of Section 3 is the prediction of growing systems. Suppose that the types of cells of a growing organism form a finite set V and that the ith generation of cells can be described by a string $w_i \in V^*$ for $i = 1, 2, \ldots, n$; let $1 \leqslant i < j \leqslant n$ imply $|w_i| < |w_j|$. Hence, all past and the present generation are known and the problem is to predict the structure of future generations, i.e., the structure of the ith generation for any $i > n$. Put $L = \{w_i;\ 1 \leqslant i \leqslant n\}$, $|w_n| = K$. Then (V, L) is finite language and $G_K(V, L)$ and $M = L(G_K(V, L))$ can be constructed. Suppose that $M = \{v_i;\ i \geqslant 1\}$, where $1 \leqslant i < j$ imply $|v_i| < |v_j|$ and that $v_i = w_i$ for any i with $1 \leqslant i \leqslant n$. Then it is natural to believe that v_i represents the ith generation for any $i > n$.

A special case of the above situation occurs if a non-negative integer $f(i)$ is assigned to any i with $1 \leqslant i \leqslant n$. E.g., $f(i)$ is the number of cells of the ith generation of a growing organism, the number of inhabitants in a certain area born in the year i, etc. We put $V = \{a, b, c\}$, $L = \{a^i bc^{f(i)};\ 1 \leqslant i \leqslant n\}$, $K = f(n) + n + 1$, and construct $G_K(V, L)$, $M = L(G_K(V, L))$. If there exists a function $g(i)$ such that $M = \{a^i bc^{g(i)};\ i \geqslant 1\}$, then, by 2.1, $g(i) = f(i)$ for any i with $1 \leqslant i \leqslant n$. Hence, g is an extrapolation of f and can serve to predict the value of $f(i)$ for $i > n$.

2. Example. Let us have $f(1) = 2$, $f(2) = 4$, $f(3) = 6$. Put $V = \{a, b, c\}$, $L = \{abc^2, a^2bc^4, a^3bc^6\}$. We obtain $G_{10}(V, L) = \langle\{a, b, c\}, \{abc^2\}, \{(b, abc^2)\}\rangle$ which implies that $L(G_{10}(V, L)) = \{a^i bc^{2i};\ i \geqslant 1\}$. Thus, we can predict that $f(i) = 2i$ for any $i \in P$. ∎

Naturally, these examples of problems are very simple and can impress that the results are obtainable without our theory. More complex problems can be solved by means of a program that has been written in the language ASSEMBLER for the computer EC 1021.

References

[1] N. Chomsky, *Three models for the description of language*, IRE Transactions on Information Theory IT-2, No. 3 (1956), 113–124.

[2] –, *Syntactic structures*, Mouton and Co., The Hague, 1957.

[3] –, *On certain formal properties of grammars*, Information and Control 2 (1959), 137–167.

[4] A. Gabrielian, *Pure grammars and pure languages*, Univ. of Waterloo, Dept. of Appl. Anal. and Comput. Sci. Rep. CSRR 2027 (1970).

[5] A. V. Gladkij, *Konfiguracionnye charakteristiki jazykov* (*Configurational characteristics of languages*), Problemy Kibernetiki 10 (1963), 251–260.

[6] R. C. Gonzalez, M. G. Thomason, *Syntactic pattern recognition: an introduction*, Addison-Wesley, Reading, 1978.

[7] J. Gruska, *On a classification of context-free languages*, Kybernetika 3 (1967), 22–29.

[8] –, *Descriptional complexity of context-free languages*, Mathematical Foundation of Computer Science. Proceedings of Symposium and Summer School. High Tatras, Sept. 3–8 (1973), 71–83.

[9] J. E. Hopcroft, J. D. Ullman, *Formal languages and their relation to automata*, Addison-Wesley, Reading, 1969.

[10] M. Jantzen, M. Kudlek, *Homomorphic images of sentential form languages defined by semi-Thue systems*, Universität Hamburg, Fachbereich Informatik, Bericht No. 89 (1983).

[11] B. Kříž, *Zobecněné gramatické kategorie* (*Generalized grammatical categories*), Thesis, University J. E. Purkyně, Brno 1980.

[12] –, *Generalized grammatical categories in the sense of Kunze*, Archivum Math. Brno 17 (1981), 151–158.

[13] J. Kunze, *Versuch eines objektivierten Grammatikmodells*, I and II, Z. Phonetik Sprachwiss. Kommunikat. 20 (1967), 415–448, and 21 (1968), 421–466.

[14] H. A. Maurer, A. Salomaa, D. Wood, *Pure grammars*, Inst. für Informationsverarbeitung, Technische Universität Graz, Bericht No. 24 (1979).

[15] M. Novotný, *Bemerkung über ableitbare Sprachen*, Spisy Přírod. Fak. Univ. J. E. Purkyně, Brno, No. 468 (1965), 503–508.

[16] –, *Constructions of grammars for formal languages*, Mathematical Foundations of Computer Science. Proceedings of Symposium and Summer School. High Tatras, September, 3–8 (1973), 125–133.

[17] –, *On some operators reducing generalized grammars*, Information and Control 26 (1974), 225–235.

[18] –, *S algebrou od jazyka ke gramatice a zpět* (*With algebra from language to grammar and back*), Academia, Publishing House of the Czechoslovak Academy of Sciences (to appear).

[19] J. Ostravský, *Effective constructions of grammars for languages of two particular classes*, Fundamenta Informaticae 8 (1985), 235–252.

[20] G. Păun, *On the smallest number of nonterminals required to generate a context-free language*, Mathematica-Revue d'analyse numérique et de la théorie de l'approximation (18) 18 (41), 2 (1976), 203–208.

[21] –, M. Novotný, *On some parameters occurring in certain effective constructions of grammars*, Fundamenta Informaticae, 10 (1987), 69–80.
[22] A. Salomaa, *Formal languages*, Academic Press, New York and London 1973.

Presented to the semester
Mathematical Problems in Computation Theory
September 16–December 14, 1985