KYBERNETIKA — VOLUME 8 (1972), NUMBER 1

Precedence Relations and Their Connection with Unambiguity of Context-free Grammars

MIROSLAV HLADKÝ

The paper deals with the use of so called precedence relations either defined between the symbols of a context-free grammar or generalized on strings. It shows how by using these relations one may recognize unambiguous context-free grammars and how to use them in syntactical analysis.

1. PRELIMINARIES

A *context-free grammar is a quadruple* (V, V_T, P, σ), where V is a finite nonempty set of symbols (vocabulary), $V_T \subset V$ is a nonempty set of terminal symbols, P is a finite nonempty set of productions of the form $v \to x$, $v \in V - V_T$, $x \in V^*$ (V^* is free monoid of strings over V including the empty string ε), $\sigma \in V - V_T$ is the initial symbol.

About the strings y, $z \in V^*$ we say that:

y immediately generates z in G and conversely *z reduces into y in G* (we denote $y \Rightarrow z$) if there exist strings u, $v \in V^*$ such that $y = uvv$, $z = uxv$ and $v \to x$ is a production from P; *y nontrivialy generates z in G* (denoted $y \overset{+}{\Rightarrow} z$) if there exist strings $z_0, z_1, \ldots, z_r \in V^*$ $(r > 0)$ such that $z_i \Rightarrow z_{i+1}$ $(i = 0, 1, \ldots, r - 1)$ and $z_0 = y$ and $z_r = z$. The sequence $z_0, z_1, \ldots, z_r$ is called a *y-derivation of the string z in G.* If we admit also $r = 0$, we say that *y generates z* and we denote $y \overset{*}{\Rightarrow} z$.

The *language $\mathscr{L}(G)$ of the context-free grammar G (context-free language)* is the set of strings $\mathscr{L}(G) = \{x \mid x \in V_T^*, \sigma \overset{*}{\Rightarrow} x\}$. By a *sentence of G* we denote any string x such that $\sigma \overset{*}{\Rightarrow} x$ (in G). To every context-free grammar G there exists a context-free grammar $G' = (V', V'_T, P', \sigma')$ such that $\mathscr{L}(G) = \mathscr{L}(G') - \{\varepsilon\}$ while:

a) no production in P' is of the form $\xi \to \varepsilon$, $\xi \to \xi$, $\xi \to \sigma'(\xi \in V' - V'_T)$,

b) $V' - V'_T$ contains only those nonterminal symbols on which σ' depends (ξ depends on v, ξ and $v \in V - V_T$, if $\xi \overset{*}{\Rightarrow} uvv$ for some u, $v \in V^*$),

c) for no $U \in V' - V'_T$ there holds $U \overset{+}{\Rightarrow} U$.

The generative power of both grammars G and G' is the same with the exception of the ability to generate the empty string ε. So in the following by the notion grammar we shall understand the context-free grammar with properties a) – c) and by the language the language of such a grammar.

By the *leftmost* (*rightmost*) *derivation* of the string w from the string ξ we denote the derivation $\xi = w_0 \Rightarrow w_1 \Rightarrow \ldots \Rightarrow w_r = w$, where $w_i = u_i v_i y_i$, $w_{i+1} = u_i z_i y_i$, $v_i \to z_i \in P$ and $u_i \in V_T^*$ ($y_i \in V_T^*$ in the case of the rightmost derivation) for all $0 \leqq i \leqq r - 1$. A grammar G is said to be *ambiguous* if there is some string in $\mathscr{L}(G)$ generated by two different leftmost (rightmost) derivations from σ. A grammar which is not ambiguous is said to be *unambiguous*.

2. SIMPLE PRECEDENCE RELATIONS AND SIMPLE PRECEDENCE GRAMMARS

In [1] the following problem is presented: Find a family of unambiguous context-free grammars. By introducing so-called simple precedence relations between the symbols of context-free grammar we may decide if such a grammar is so called simple precedence grammar or not. We shall show in this part that every simple precedence grammar is unambiguous.

Let $G = (V, V_T, P, \sigma)$ be a given grammar.

Definition 2.1. Let $U \in V - V_T$. We define the sets $L(U)$, $R(U)$:

$$L(U) = \{A \mid U \overset{+}{\Rightarrow} Ax,\ A \in V,\ x \in V^*\},$$

$$R(U) = \{A \mid U \overset{+}{\Rightarrow} xA,\ A \in V,\ x \in V^*\}.$$

Definition 2.2. Let $S_1, S_2 \in V$. We define the relations $\doteq$, $\lessdot$, $\gtrdot$:

a) $S_1 \doteq S_2$ if there exists in P a production of the form $U \to uS_1S_2v$ for some $u, v \in V^*$.

b) $S_1 \lessdot S_2$ if there exists in P a production of the form $U \to uS_1U_2v$ and $S_2 \in L(U_2)$ for some $U, U_2 \in V - V_T$ and $u, v \in V^*$.

c) $S_1 \gtrdot S_2$ if there exists in P a production of the form $U \to uU_1U_2v$ and there holds $S_1 \in R(U_1)$ and either $S_2 = U_2$ or $S_2 \in L(U_2)$ for some $u, v \in V^*$, $U_1 \in V - V_T$.

The relations $\doteq$, $\lessdot$, $\gtrdot$ will be called *simple precedence relations.*

Remark 2.1. In this part we shall be limited only to the simple precedence relations. The word "simple" will be therefore omitted.

Remark 2.2. Simple precedence relations are not generally symmetric. If we say in the following "the precedence relation R holds between the symbols S_1, S_2" it will indicate that (S_1, S_2) is in the relation R.

Definition 2.3. A grammar $G = (V, V_T, P, \sigma)$ such that between every two symbols of V at most one precedence relation holds will be called *simple precedence grammar.*

Notation. For any string $x \neq \varepsilon$ we shall denote by $l(x)$ and $r(x)$ the leftmost and the rightmost symbol of x respectively.

Lemma 2.1. *Let $G = (V, V_T, P, \sigma)$ be a simple precedence grammar and let $x = x_1 x_2 \ldots x_n$ be an arbitrary sentence of G. Then between every two adjacent symbols x_i, x_{i+1} $(i = 1, 2, \ldots, n - 1)$ of x just one precedence relation holds.*

Proof. According to Definition 2.3 there is sufficient to prove that between every two adjacent symbols of x at least one precedence relation holds. Let $x \Rightarrow y$, $x = uZv$, $y = uzv$ and let between every two adjacent symbols of x at least one precedence relation hold. We shall prove that at least one precedence relation holds between every two adjacent symbols of y. By Definition 2.2 the relation $\doteq$ holds between every two adjacent symbols of z and

$$r(u) \lessdot l(z) \quad \text{when} \quad r(u) \doteq Z \quad \text{or} \quad r(u) \lessdot Z\,,$$

$$r(u) \gtrdot l(z) \quad \text{when} \quad r(u) \gtrdot Z \quad \text{and}$$

$$r(z) \gtrdot l(v) \quad \text{for an arbitrary relation between } Z \text{ and } l(v)\,.$$

Then there holds at least one precedence relation between every two adjacent symbols of y. The assertion of the Lemma follows from above by induction according to the length of the derivation.

Lemma 2.2. *Let $G = (V, V_T, P, \sigma)$ be a grammar. Then for no sentence u of G there holds $u \overset{+}{\Rightarrow} u$.*

Proof. Let there exist the sentence u of G such that $u \overset{+}{\Rightarrow} u$.

a) If $|u| = 1$ then we get a contradiction with conditions laid on G in Part 1.

b) Let $|u| > 1$ i.e. $u = u_1 u_2 \ldots u_n$ $(u_i \in V$ for $i = 1, 2, \ldots, n,$ $n > 1)$. Then according to Lemma 1.4.6 of [1]* there holds $u_1 \overset{*}{\Rightarrow} u_1$, $u_2 \overset{*}{\Rightarrow} u_2, \ldots, u_n \overset{*}{\Rightarrow} u_n$ and as $u \overset{+}{\Rightarrow} u$ at least one of these derivations is nontrivial. But if for any i $(1 \leq i \leq n$ there holds $u_i \overset{+}{\Rightarrow} u_i$ we get a contradiction with the assertion proved in a).

Theorem 2.1. *Let $G = (V, V_T, P, \sigma)$ be a simple precedence grammar such that no two productions in P have the same right sides. Then G is unambiguous.*

* [1], page 21: If $v_1 v_2 \ldots v_r \overset{*}{\Rightarrow} w$, then there exist $w_1, \ldots, w_r$ such that $w = w_1 \ldots w_r$ and $v_i \overset{*}{\Rightarrow} w_i$ for each i. Furthermore, each occurence of a production used in the generation of $v_1 \ldots v_r \overset{*}{\Rightarrow} w$ occurs in the generation of some $v_i \overset{*}{\Rightarrow} w_i$ and conversely.

Proof. We shall suppose that G is ambiguous. Then there exists a sentence x of G in $\mathscr{L}(G)$ such that it has two different rightmost derivations

$$\sigma = w_0 \Rightarrow w_1 \Rightarrow \ldots \Rightarrow w_k = x \,,$$

$$\sigma = w'_0 \Rightarrow w'_1 \Rightarrow \ldots \Rightarrow w'_h = x \,.$$

Let $w_{k-1} = uZv$, $w'_{h-1} = u'Z'v'$ and $Z \to z$, $Z' \to z'$ be two productions of P. Plainly $u'z'v' = uzv = x$. According to the properties of the rightmost derivation there hold

$$(1) \qquad v \in V_T^* \,, \quad v' \in V_T^* \,.$$

Hence $v = v'$ (otherwise either $Z'v' \subseteq v$ or $Zv \subseteq v'$ both in a contradiction with (1)). Since $z \subset z'$ or $z' \subset z$ leads to a contradiction with Lemma 2.1 (if for example $z \subset z'$ then between every two adjacent symbols of z' there holds the relation $\doteq$ but furthermore between at least two adjacent symbols of z' there holds the relation $\lessdot$; we shall get similar result for $z' \subset z$) there must hold $z = z'$ and according to the assumption also $Z = Z'$. Hence $w_{k-1} = w'_{h-1}$. Repeating these arguments and applying Lemma 2.2 we get $k = h$ and for all $0 \leqq i \leqq k$ $w_i = w'_i$ a contradiction with the assumption that both derivations are different. Hence G is unambiguous.

3. PRECEDENCE RELATIONS OF HIGHER ORDER

In [2] the use of simple precedence relations in syntactical analysis is studied. Using simple precedence relations we have no difficulties in the parse of sentences of simple precedence grammars but great difficulties arise when between some symbols more than one simple precedence relation hold. By generalization of the definition of simple precedence relations on strings we may reach similar results, i.e. we may obtain another class of unambiguous context-free grammars. The result depends on the manner of such generalization. The way suggested in [2] is too general and it leads to many difficulties both in the study of the properties of these grammars and in the parse. We shall now present one such generalization and state sufficient conditions for the given grammar (not a simple precedence one) to be unambiguous. At the end of this part we put the algorithm of syntactical analysis and its use on several examples of the practical parse may be seen in Part 4.

Definition 3.1. Let k be an integer, $U \in V - V_T$. We define the sets $L^k(U)$ and $R^k(U)$:

$$L^k(U) = \{z_1 z_2 \ldots z_k \mid U \overset{+}{\Rightarrow} z_1 \ldots z_k u',\ u' \in V^*,\ z_1 \ldots z_k \in V^* - \{\varepsilon\}\} \,,$$

$$R^k(U) = \{z_1 z_2 \ldots z_k \mid U \overset{+}{\Rightarrow} u' z_1 \ldots z_k,\ u' \in V^*,\ z_1 \ldots z_k \in V^* - \{\varepsilon\}\} \,.$$

Definition 3.2. Let $x, y \in V^* - \{\varepsilon\}$, $x = S_{-m} \ldots S_{-1}$, $y = S_1 \ldots S_n$, m and n are positive integers. We define the relations $\doteq$, $\lessdot$ and $\gtrdot$:

a) $x \doteq y$ if there exists in P a production of the form $U \to uxyv$ for some $u, v \in V^*$.

b) $x \lessdot y$ if there exists in P a production of the form $U \to uxU_1v$ and $y \in L^n(U_1)$ for some $U, U_1 \in V - V_T$ and $u, v \in V^*$.

c) $x \gtrdot y$ if either there exists a production in P of the form $U \to uU_{-1}yv$ and $x \in R^m(U_{-1})$ for some $U, U_{-1} \in V - V_T$ and some $u, v \in V^*$ or there exists in P a production of the form $U \to uU_{-1}U_1v$ and $x \in R^m(U_{-1})$, $y \in L^n(U_1)$ for some $U, U_{-1}, U_1 \in V - V_T$ and $u, v \in V^*$.

In all these cases we speak about *precedence relations of the order* (m, n).

Remark 3.1. We see that simple precedence relations are precedence relations of the order $(1,1)$.

Remark 3.2. If we say that the precedence relation R holds between the strings x, y it will indicate that:

a) $(x, y) \in R$,

b) the relations is of the order $(|x|, |y|)$.

Lemma 3.1. *Let R be one of precedence relations, $S_{-p} \ldots S_{-1}, S_1 \ldots S_q \in V^* - \{\varepsilon\}$ and let $(S_{-p} \ldots S_{-1}, S_1 \ldots S_q) \in R$. Then $(S_{-h} \ldots S_{-1}, S_1 \ldots S_k) \in R$ for any integers h, k such that $1 \leqq h \leqq p$, $1 \leqq k \leqq q$.*

Proof. a) If $R = \doteq$ then P contains the production of the form $U \to uS_{-p} \ldots \ldots S_{-h} \ldots S_{-1}S_1 \ldots S_k \ldots S_qv$ for some $u, v \in V^*$ and by Definition 3.2 $(S_{-h} \ldots \ldots S_{-1}, S_1 \ldots S_k) \in R$.

b) If $R = \lessdot$ then P contains the production of the form $U \to uS_{-p} \ldots S_{-h} \ldots \ldots S_{-1}U_1v$, where $U_1 \overset{+}{\Rightarrow} S_1 \ldots S_k \ldots S_qv'$ for some $u, v, v' \in V^*$. By Definition 3.2 $(S_{-h} \ldots S_{-1}, S_1 \ldots S_k) \in R$.

c) If $R = \gtrdot$ then P contains the production of the form $U \to uV_{-1}V_1v$ where $V_{-1} \in V - V_T$, $V_1 \in V$, u and $v \in V^*$ and there hold $V_{-1} \overset{+}{\Rightarrow} u'S_{-p} \ldots S_{-1}$ and either $V_1v = S_1 \ldots S_qv'$ or $V_1 \overset{+}{\Rightarrow} S_1 \ldots S_qv'$ for some $v' \in V^*$. Then by Definition 3.2 there holds also $S_{-h} \ldots S_{-1} \gtrdot S_1 \ldots S_k$ for any integers h, k such that $1 \leqq h \leqq p$, $1 \leqq k \leqq q$.

Definition 3.3. We say that the grammar $G = (V, V_T, P, \sigma)$ (in the sense of Part 1) has the *property* A when it complies the following conditions: If there hold between $S_i, S_{i+1} \in V$ more than one (simple) precedence relations then to every sentence of G of the form $uS_iS_{i+1}v$ there exist $u_1, u_2, v_1, v_2 \in V^*$ such that $u = u_1u_2$, $v = v_1v_2$, $uv \neq \varepsilon$ and between uS_i, $S_{i+1}v$ only one precedence relation holds and between every two adjacent symbols of the strings u_2S_i, $S_{i+1}v_1$ only one simple precedence relation holds.

Theorem 3.1. *Let $G = (V, V_T, P, \sigma)$ be a grammar such that:*

a) *P contains no productions with the same right sides,*

b) *G has the property* A *from Definition* 3.3.

 Then G is unambiguous grammar.

Proof. Let $x = S_1 S_2 \dots S_i S_{i+1} \dots S_k$ be a sentence of G such that between S_i, S_{i+1} more than one simple precedence relations hold. But according to b) there exist substrings $S_h \dots S_i$, $S_{i+1} \dots S_l$ of the strings $S_1 \dots S_i$, $S_{i+1} \dots S_k$ respectively such that between $S_h \dots S_i$, $S_{i+1} \dots S_l$ only one precedence relation holds. By Lemma 3.1 and Definition 3.2 the same relation holds between S_i, S_{i+1} and when all relation symbols which are between S_i, S_{i+1} are replaced by that which holds between $S_h \dots S_i$, $S_{i+1} \dots S_l$ we get the same situation as we had in the parse of sentences of simple precedence grammars. And simple precedence grammars are unambiguous.

4. APPLICATION

Merely to clarify what follows, we shall at first give an example of the parse of several sentences of simple precedence grammar. It is really simple. To the investigated string x we construct the "associated" string $\perp x \perp$ where $\perp \notin V$ and we put $\perp \lessdot \alpha$ and $\alpha \gtrdot \perp$ for any $\alpha \in V$. Going from the left to the right we find out the first occurence of the relation symbol $\gtrdot$ and then we return back to the left seeking the first occurence of the relation symbol $\lessdot$. If the string between $\lessdot$ and $\gtrdot$ is the right side of some production of P we replace it by the nonterminal symbol which is on the left side of this production. Then we add missing precedence relations and continue in the same way. The parse is succesfully finished when we obtain the string $\perp$ *initial symbol* $\perp$. If no production in P has the same right side as the string in $\lessdot$, $\gtrdot$ the parse is finished because x is not a sentence of our simple precedence grammar. We get the same result if between some two adjacent symbols of x no simple precedence relation holds.

Example 1.

$$
\begin{aligned}
G' &= (V', V'_T, P', \tilde{I}),\\
V' &= \{\tilde{I}, \tilde{L}, \tilde{D}, a, b, \dots, z, A, B, \dots, Z, 0, 1, \dots, 9\},\\
V'_T &= \{a, b, \dots, z, A, B, \dots, Z, 0, 1, \dots, 9\},\\
P' &: \tilde{I} \to \tilde{L},\\
&\quad \tilde{I} \to \tilde{I}\tilde{L},\\
&\quad \tilde{I} \to \tilde{I}\tilde{D},\\
&\quad \tilde{L} \to a,\ \tilde{L} \to b,\ \dots,\ \tilde{L} \to z,\\
&\quad \tilde{L} \to A,\ \tilde{L} \to B,\ \dots,\ \tilde{L} \to Z,\\
&\quad \tilde{D} \to 0,\ \tilde{D} \to 1,\ \dots,\ \tilde{D} \to 9.
\end{aligned}
$$

We can easily find out that $\mathscr{L}(G')$ is a fragment of the language ALGOL 60 describing the syntactical definition of identifier.

The corresponding precedence matrix is

	$\tilde{I}$	$\tilde{L}$	$\tilde{D}$	a	b	c	$\ldots$	z	A	B	C	$\ldots$	Z	0	1	2	$\ldots$	9
$\tilde{I}$		$\doteq$	$\doteq$	$\lessdot$	$\lessdot$	$\lessdot$	$\ldots$	$\lessdot$	$\lessdot$	$\lessdot$	$\lessdot$	$\ldots$	$\lessdot$	$\lessdot$	$\lessdot$	$\lessdot$	$\ldots$	$\lessdot$
$\tilde{L}$		$\gtrdot$	$\gtrdot$	$\gtrdot$	$\gtrdot$	$\gtrdot$	$\ldots$	$\gtrdot$	$\gtrdot$	$\gtrdot$	$\gtrdot$	$\ldots$	$\gtrdot$	$\gtrdot$	$\gtrdot$	$\gtrdot$	$\ldots$	$\gtrdot$
$\tilde{D}$		$\gtrdot$	$\gtrdot$	$\gtrdot$	$\gtrdot$	$\gtrdot$	$\ldots$	$\gtrdot$	$\gtrdot$	$\gtrdot$	$\gtrdot$	$\ldots$	$\gtrdot$	$\gtrdot$	$\gtrdot$	$\gtrdot$	$\ldots$	$\gtrdot$
a		$\gtrdot$	$\gtrdot$	$\gtrdot$	$\gtrdot$	$\gtrdot$	$\ldots$	$\gtrdot$	$\gtrdot$	$\gtrdot$	$\gtrdot$	$\ldots$	$\gtrdot$	$\gtrdot$	$\gtrdot$	$\gtrdot$	$\ldots$	$\gtrdot$
b		$\gtrdot$	$\gtrdot$	$\gtrdot$	$\gtrdot$	$\gtrdot$	$\ldots$	$\gtrdot$	$\gtrdot$	$\gtrdot$	$\gtrdot$	$\ldots$	$\gtrdot$	$\gtrdot$	$\gtrdot$	$\gtrdot$	$\ldots$	$\gtrdot$
c		$\gtrdot$	$\gtrdot$	$\gtrdot$	$\gtrdot$	$\gtrdot$	$\ldots$	$\gtrdot$	$\gtrdot$	$\gtrdot$	$\gtrdot$	$\ldots$	$\gtrdot$	$\gtrdot$	$\gtrdot$	$\gtrdot$	$\ldots$	$\gtrdot$
.																		
.		$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$
.																		
z		$\gtrdot$	$\gtrdot$	$\gtrdot$	$\gtrdot$	$\gtrdot$	$\ldots$	$\gtrdot$	$\gtrdot$	$\gtrdot$	$\gtrdot$	$\ldots$	$\gtrdot$	$\gtrdot$	$\gtrdot$	$\gtrdot$	$\ldots$	$\gtrdot$
A		$\gtrdot$	$\gtrdot$	$\gtrdot$	$\gtrdot$	$\gtrdot$	$\ldots$	$\gtrdot$	$\gtrdot$	$\gtrdot$	$\gtrdot$	$\ldots$	$\gtrdot$	$\gtrdot$	$\gtrdot$	$\gtrdot$	$\ldots$	$\gtrdot$
B		$\gtrdot$	$\gtrdot$	$\gtrdot$	$\gtrdot$	$\gtrdot$	$\ldots$	$\gtrdot$	$\gtrdot$	$\gtrdot$	$\gtrdot$	$\ldots$	$\gtrdot$	$\gtrdot$	$\gtrdot$	$\gtrdot$	$\ldots$	$\gtrdot$
C		$\gtrdot$	$\gtrdot$	$\gtrdot$	$\gtrdot$	$\gtrdot$	$\ldots$	$\gtrdot$	$\gtrdot$	$\gtrdot$	$\gtrdot$	$\ldots$	$\gtrdot$	$\gtrdot$	$\gtrdot$	$\gtrdot$	$\ldots$	$\gtrdot$
.																		
.		$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$
.																		
0		$\gtrdot$	$\gtrdot$	$\gtrdot$	$\gtrdot$	$\gtrdot$	$\ldots$	$\gtrdot$	$\gtrdot$	$\gtrdot$	$\gtrdot$	$\ldots$	$\gtrdot$	$\gtrdot$	$\gtrdot$	$\gtrdot$	$\ldots$	$\gtrdot$
1		$\gtrdot$	$\gtrdot$	$\gtrdot$	$\gtrdot$	$\gtrdot$	$\ldots$	$\gtrdot$	$\gtrdot$	$\gtrdot$	$\gtrdot$	$\ldots$	$\gtrdot$	$\gtrdot$	$\gtrdot$	$\gtrdot$	$\ldots$	$\gtrdot$
2		$\gtrdot$	$\gtrdot$	$\gtrdot$	$\gtrdot$	$\gtrdot$	$\ldots$	$\gtrdot$	$\gtrdot$	$\gtrdot$	$\gtrdot$	$\ldots$	$\gtrdot$	$\gtrdot$	$\gtrdot$	$\gtrdot$	$\ldots$	$\gtrdot$
3		$\gtrdot$	$\gtrdot$	$\gtrdot$	$\gtrdot$	$\gtrdot$	$\ldots$	$\gtrdot$	$\gtrdot$	$\gtrdot$	$\gtrdot$	$\ldots$	$\gtrdot$	$\gtrdot$	$\gtrdot$	$\gtrdot$	$\ldots$	$\gtrdot$
.																		
.		$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$	$\ldots$
.																		
9		$\gtrdot$	$\gtrdot$	$\gtrdot$	$\gtrdot$	$\gtrdot$	$\ldots$	$\gtrdot$	$\gtrdot$	$\gtrdot$	$\gtrdot$	$\ldots$	$\gtrdot$	$\gtrdot$	$\gtrdot$	$\gtrdot$	$\ldots$	$\gtrdot$

a) The investigated string is "$A23KB6$":

$$\begin{array}{l}
\perp \lessdot A \gtrdot 2 \gtrdot 3 \gtrdot K \gtrdot B \gtrdot 6 \gtrdot \perp \\
\perp \lessdot \tilde{L} \gtrdot 2 \gtrdot 3 \gtrdot K \gtrdot B \gtrdot 6 \gtrdot \perp \\
\perp \lessdot \tilde{I} \lessdot 2 \gtrdot 3 \gtrdot K \gtrdot B \gtrdot 6 \gtrdot \perp \\
\perp \lessdot \tilde{I} \doteq \tilde{D} \gtrdot 3 \gtrdot K \gtrdot B \gtrdot 6 \gtrdot \perp \\
\perp \lessdot \tilde{I} \lessdot 3 \gtrdot K \gtrdot B \gtrdot 6 \gtrdot \perp \\
\perp \lessdot \tilde{I} \doteq \tilde{D} \gtrdot K \gtrdot B \gtrdot 6 \gtrdot \perp \\
\perp \lessdot \tilde{I} \lessdot K \gtrdot B \gtrdot 6 \gtrdot \perp \\
\perp \lessdot \tilde{I} \doteq \tilde{L} \gtrdot B \gtrdot 6 \gtrdot \perp \\
\perp \lessdot \tilde{I} \lessdot B \gtrdot 6 \gtrdot \perp \\
\perp \lessdot \tilde{I} \doteq \tilde{L} \gtrdot 6 \gtrdot \perp \\
\perp \lessdot \tilde{I} \lessdot 6 \gtrdot \perp \\
\perp \lessdot \tilde{I} \doteq \tilde{D} \gtrdot \perp \\
\perp \quad \tilde{I} \quad \perp
\end{array}$$

b) We investigate the string "$2aB73$":

$$\perp \lessdot 2 \gtrdot a \gtrdot B \gtrdot 7 \gtrdot 3 \gtrdot \perp$$

$$\perp \lessdot \tilde{D} \gtrdot a \gtrdot B \gtrdot 7 \gtrdot 3 \gtrdot \perp$$

The parse finishes because $\tilde{D}$ is the right side of no production of P'. The string "$2aB73$" is not a sentence of G'.

c) The string "$Ab7\tilde{I}2$" is not the sentence of G' because no simple precedence relation holds between its adjacent symbols 7 and $\tilde{I}$.

The parsing algorithm for the class of grammars described in Part 3 is given by the flowchart (Fig. 1). It follows from above that we get only one sequence of the left reductions and thus the rightmost derivation for every sentence of G.

Comments to the parsing algorithm:

1. We construct the associated string to the investigated string in the same way as we did in the case of simple precedence grammars.

2. When between two adjacent symbols S_i, S_{i+1} of the investigated string more than one precedence relations hold we add at first the left context of the symbol S_i to it. If it is not possible to continue to the left we start to add the right context of the symbol S_{i+1} to it. We continue until the precedence relation between arizing strings is the only one and then we replace by its relation symbol the relation symbol between S_i, S_{i+1}.

3. The investigated string is not the sentence of the given grammar of supposed properties in these cases:

a) between some two adjacent symbols of it no precedence relation holds,

b) the string in $\lessdot$, $\gtrdot$ is not the right side of any production of P,

c) we do not obtain strings such that between them only one precedence relation holds (there is not satisfied the condition b) from Theorem 3.1).

Example 2.

$$G = (V, V_T, P, S)$$

where

$$V = \{S, U, W, a, b, c, d, e, f, g\},$$

$$V_T = \{a, b, c, d, e, f, g\},$$

$$P: S \to aUWd, \quad U \to bcU, \quad U \to c,$$

$$W \to Wefg, \quad W \to ef.$$

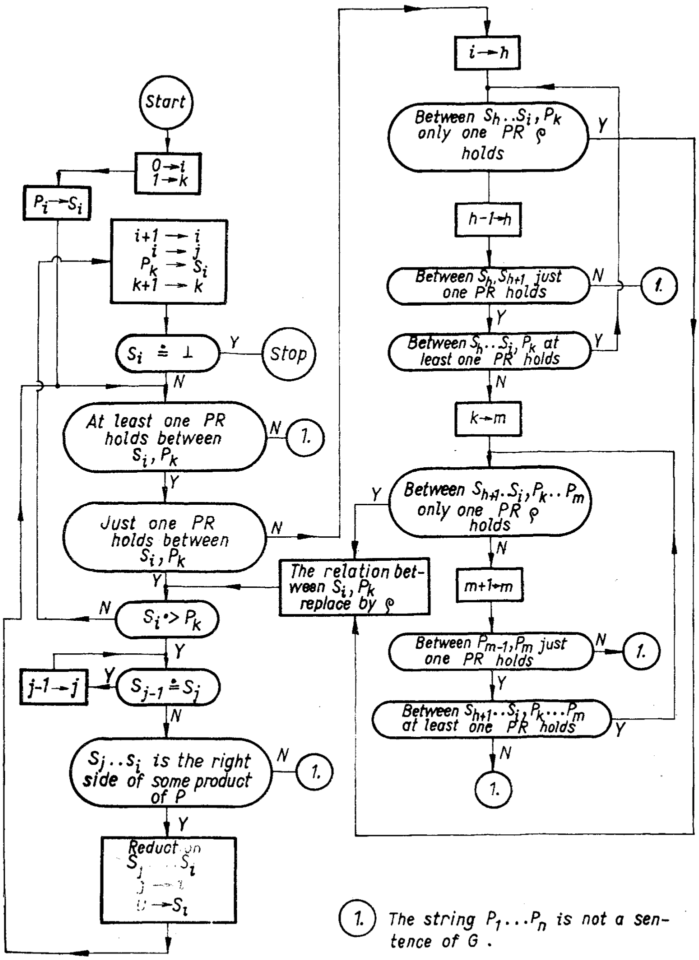

Fig. 1. Parsing algorithm flowchart (PR = precedence relation).

The corresponding precedence matrix of simple precedence relation is

	U	W	a	b	c	d	e	f	g
U		$\gtrdot, \doteq, \lessdot$					$\lessdot, \gtrdot$		
W						$\doteq$	$\doteq$		
a	$\doteq$			$\lessdot$	$\lessdot$				
b					$\doteq$				
c	$\doteq$	$\gtrdot$		$\lessdot$	$\lessdot$		$\gtrdot$		
d									
e								$\doteq$	
f						$\gtrdot$	$\gtrdot$		$\doteq$
g						$\gtrdot$	$\gtrdot$		

A) The investigated string is "*abcbccefefgd*":

a) $\perp \lessdot a \lessdot b \doteq c \lessdot b \doteq c \lessdot c \gtrdot e \doteq f \gtrdot e \doteq f \doteq g \gtrdot d \gtrdot \perp$
b) $\perp \lessdot a \lessdot b \doteq c \lessdot b \doteq c \doteq U \lessdot, \underline{\gtrdot}\ e \doteq f \gtrdot e \doteq f \doteq g \gtrdot d \gtrdot \perp$
c) $\perp \lessdot a \lessdot b \doteq c \doteq U \lessdot, \underline{\gtrdot}\ e \doteq f \gtrdot e \doteq f \doteq g \gtrdot d \gtrdot \perp$
d) $\perp \lessdot a \doteq U \gtrdot, \underline{\lessdot}\ e \doteq f \gtrdot e \doteq f \doteq g \gtrdot d \gtrdot \perp$
e) $\perp \lessdot a \doteq U\ \underline{\lessdot}, \doteq, \gtrdot\ W \doteq e \doteq f \doteq g \gtrdot d \gtrdot \perp$
f) $\perp \lessdot a \doteq U \lessdot, \underline{\doteq}, \gtrdot\ W \doteq d \gtrdot \perp$
g) $\perp\ S\ \perp$

Comments to the parse:

a) Between every two adjacent symbols there holds just one precedence relation.
b), c) There hold $U \lessdot e$ and simultaneously $U \gtrdot e$ but $cU \gtrdot e$. We put between U, e the relation symbol $\gtrdot$.
d) There hold simultaneously $U \gtrdot e$ and $U \lessdot e$ but $aU \lessdot e$. We put $\lessdot$ between U, e.
e) There hold $U \gtrdot W$, $U \doteq W$, $U \lessdot W$ and also $aU \lessdot W$ and $aU \doteq W$ but $aU \lessdot We$. We put $\lessdot$ between U, W.
f) There hold $U \lessdot W$, $U \doteq W$, $U \gtrdot W$, $aU \lessdot W$, $aU \doteq W$ but $aU \doteq Wd$.
g) The parse finishes.

B) The investigated string is "*abccefg*":

a) $\perp \lessdot a \lessdot b \doteq c \lessdot c \gtrdot e \doteq f \doteq g \gtrdot \perp$
b) $\perp \lessdot a \lessdot b \doteq c \doteq U\ \underline{\gtrdot}, \lessdot e \doteq f \doteq g \gtrdot \perp$
c) $\perp \lessdot a \doteq U \gtrdot, \underline{\lessdot}\ e \doteq f \doteq g \gtrdot \perp$

Comments to the parse:

b) There hold $U \gtrdot e$ and $U \lessdot e$ but $cU \gtrdot e$.
c) In P is no production with the right side "*efg*". The string $abccef \notin \mathscr{L}(G)$.

C) The string "*abcbcfgd*" does not also belong to $\mathscr{L}(G)$ because between its two adjacent symbols c, f no simple precedence relation holds

(Received July 2, 1969.)

REFERENCES

[1] S. Ginsburg: The Mathematical Theory of Context-free Languages. New York 1967.
[2] N. Wirth, H. Weber: Euler — A Generalization of ALGOL. Communication of the ACM (1966), 1, 2.

VÝTAH

Precedenční relace a jejich souvislost s jednoznačností nekontextových gramatik

MIROSLAV HLADKÝ

V [1] je předložen následující problém: Najděte třídu jednoznačných nekontextových gramatik. Na tento problém navazuje předložená práce, kterou lze v podstatě rozdělit do dvou částí:

a) V první části (odstavce 1 a 2) jsou zavedeny tzv. jednoduché precedenční relace mezi symboly slovníku nekontextové gramatiky. Dále jsou definovány tzv. jednoduché precedenční gramatiky jakožto nekontextové gramatiky takové, že mezi každými dvěma symboly slovníku dané nekontextové gramatiky platí nejvýše jedna jednoduchá precedenční relace. V závěru této části je ukázáno, že každá jednoduchá precedenční gramatika taková, že žádná dvě pravidla nemají stejné pravé strany, je jednoznačná.

b) Možnost zobecnění jednoduchých precedenčních relací je nadhozena v [2]. Navržený způsob je však příliš obecný a vede k zásadním potížím jak ve studiu vlastností těchto relací, tak při syntaktické analýze. Přesto je však možno získat širší třídu nekontextových, gramatik než je třída jednoduchých precedenčních gramatik rovněž jednoznačných. V druhé části práce (odstavce 3 a 4) je navržen způsob zobecnění jednoduchých precedenčních relací na relace mezi řetězy a možnost využití těchto relací k získání širší třídy jednoznačných nekontextových gramatik. V závěru je uveden algoritmus syntaktické analýzy pro tyto gramatiky a jeho aplikace na příkladech.

Dr. Miroslav Hladký, Katedra aplikované matematiky FEVUT (Department of Applied Mathematics — Technical University), Hilleho 6, Brno.