Jan Ostravský

Acceptors and generalized lc-grammars

Archivum Mathematicum, Vol. 18 (1982), No. 3, 151--162

Persistent URL: http://dml.cz/dmlcz/107137

ARCH. MATH. 3, SCRIPTA FAC. SCI. NAT. UJEP BRUNENSIS
XVIII: 151—162, 1982

ACCEPTORS AND GENERALIZED LC-GRAMMARS

JAN OSTRAVSKÝ, Gottwaldov

(Received May 5, 1980)

Formal languages studied in the literature are mostly generated by finite devices (grammars, finite acceptors). Infinite devices generating languages appear only exceptionally. We mention machines of Pawlak [7] and acceptors studied by Novotný [2]. These acceptors, though infinite, define interesting classes of languages that can be completely characterized in an instrinsic way.

In the present paper, we find other infinite devices generating the classes of languages studied by Novotný, the so called generalized labelled contextual grammars. They are introduced as generatizations of contextual grammars due to Marcus [1]. We find two classes of such generalized grammars that generate exactly the languages that are accepted by α-acceptors (β-acceptors) in the sense of Novotný and exactly all computations of machines in the sense of Pawlak.

I. Generalized LC-grammars

Let V be a set (of any cardinal number), $L \subseteq V^*$ be a language over V. Then the ordered pair of strings $(u, v) \in V^* \times V^*$ is called a context over V. The context $(u, v) \in V^* \times V^*$ is said to accept the string $x \in V^*$ if $u \times v \in L$.

Let V be a set, $L \subseteq V^*$ a set of strings over V, $C \subseteq V^* \times V^*$ be a set contexts over V. Then the ordered triple $G = \langle V, L, C \rangle$ is called a generalized contextual grammar. Let K be the least set of strings over V having the following properties:

1° $L \subseteq K$,

2° If $x \in K$ and $(u, v) \in C$, then $u \times v \in K$.

Then K is called the language generated by G and denoted by $L(G)$.

1.1. Proposition. *Any language can be generated by a suitable generalized contextual grammar.*

Proof. If $L \subseteq V^*$ is a language, we put $G = \langle V, L, \emptyset \rangle$. □

Let $\langle V, L, C \rangle$ by a generalized contextual grammar and S a set disjoint from V. The elements of S are called labels. Let $T \subseteq S$ be a fixed subset of S, it is said to be the marked subset. Suppose that a mapping φ: $L \to 2^S$ is given assigning a set of labels $\varphi(x) \subseteq S$ to any $x \in L$. Furthermore, let $\psi: C \to 2^{S \times S}$ be a mapping assigning a set of ordered pair $(u, v) \in S \times S$ to any context $(u, v) \in C$. Any pair $(s, s') \in \psi(u, v)$ will be called a label of the context (u, v).

The ordered 7-tuple $G = \langle V, L, C, S, T, \varphi, \psi \rangle$ is said to be a generalized labelled contextual grammar (generalized lc-grammar). A generalized lc-grammar is said to be an lc-grammar if the sets V, L, C, S are finite. A generalized lc-grammar is said to be simple if the set T contains only one element. If $T = \{\sigma\}$, we write $\langle V, L, C, S, \sigma, \varphi, \psi \rangle$ instead of $\langle V, L, C, S, \{\sigma\}, \varphi, \psi \rangle$. A generalized simple lc-grammar will be called briefly a generalized slc-grammar.

For any string $x \in V^*$, we define a set of labels $\Phi(x) \subseteq S$ as the least subset in S satisfying the following conditions:

1. $\varphi(x) \subseteq \Phi(x)$ for any $x \in L$,
2. If $x = uzv$, where $u, z, v \in V^*$ and if $s' \in \Phi(z)$, $(s, s') \in \psi(u, v)$, then $s \in \Phi(x)$.

For any $s \in S$, we define $\Psi(s)$ as the set of all $x \in V^*$ such that $s \in \Phi(x)$. Furthermore, we put $L(G) = \bigcup_{t \in T} \Psi(t)$ and $L(G)$ is called the language generated by the generalized lc-grammar G.

1.1. Example. *Express the language of Curry by means of an slc-grammar.*

Let $V = \{a, b, c\}$, $S = \{\sigma, A, B, C, D\}$, $L = \{a\}$, $C = \{(\Lambda, a), (\Lambda, b), (\Lambda, c)\}$, $\varphi(a) = \{D\}$, $\psi(\Lambda, a) = \{(A, B)\}$, $\psi(\Lambda, b) = \{(\sigma, A), (A, A), (C, C), (C, D)\}$, $\psi(\Lambda, c) = \{(B, C)\}$, $G = \langle V, L, C, S, \sigma, \varphi, \psi \rangle$.

Example of a string in $L(G)$:

We have $D \in \varphi(a) \subseteq \Phi(a)$, $(C, D) \in \psi(\Lambda, b)$ and hence $C \in \Phi(ab)$. Since $(C, C) \in \psi(\Lambda, b)$, we obtain $C \in \Phi(ab^2)$. Further $(B, C) \in \psi(\Lambda, c)$ and therefore, $B \in \Phi(ab^2c)$. From the condition $(A, B) \in \psi(\Lambda, a)$, it follows that $A \in \Phi(ab^2ca)$. Since $(\sigma, A) \in \psi(\Lambda, b)$ we have finally $\sigma \in \Phi(ab^2cab)$ and then $ab^2cab \in L(G)$.

An ordered quadruple $H = \langle U, V, \sigma, R \rangle$, where $V \subseteq U$, $\sigma \in U - V$, $R \subseteq (U - V) \times V^*(U - V) V^* \cup (U - V) \times V^*$, is said to be a generalized linear grammar.

The relation $\Rightarrow (R)$ of direct derivation and the relation $\overset{*}{\Rightarrow} (R)$ of derivation with respect to R are defined in the same way as for grammars. The set $L(H) = \{w \in V^*, \sigma \Rightarrow w(R)\}$ is said to be the language generated by H. We write $\Rightarrow$, $\overset{*}{\Rightarrow}$ instead of $\overset{*}{\Rightarrow} (R)$, $\overset{*}{\Rightarrow} (R)$, respectively, if R is clear from the context.

A generalized linear grammar is said to be a linear grammar if the sets U, R are finite.

Let $G = \langle V, L, C, S, \sigma, \varphi, \psi \rangle$ be a generalized slc-grammar. We put

$$R_1 = \{(s, x); x \in L, s \in \varphi(x)\},$$
$$R_2 = \{(s, us'v); (u, v) \in C, (s, s') \in \psi(u, v)\},$$
$$H = \langle V \cup S, V, \sigma, R_1 \cup R_2 \rangle.$$

Then, clearly, H is a generalized linear grammar. It will be called the generalized linear grammar associated with G and denoted by $A(G)$. Note that $A(G)$ is a linear grammar for a slc-grammar G.

1.2. Example. *Construct the linear grammar associated with the slc-grammar G which defines the language of Curry.*

For the slc-grammar G of 1.1., we define $H = A(G)$. We obtain

$$R_1 = \{(D, a)\},$$
$$R_2 = \{(A, Ba), (\sigma, Ab), (A, Ab), (C, Cb), (C, Db), (B, Cc)\},$$
$$H = \langle \{\sigma, A, B, C, D, a, b, c\}; \{a, b, c\},$$
$$\{(\sigma, Ab), (A, Ab), (A, Ba), (B, Cc), (C, Cb), (C, Db), (D, a)\}, \sigma\rangle.$$

Let $H = \langle U, V, \sigma, R\rangle$ be a generalized linear grammar. We put

$$S = U - V,$$
$$R_1 = \{(s, x); (s, x) \in R, s \in S, x \in V^*\},$$
$$R_2 = R - R_1.$$

Thus, for any $(s, x) \in R_2$, there exists exactly one $s' \in S$ and $(u, v) \in V^* \times V^*$ such that $x = us'v$.

We define

$$L = \{x; (s, x) \in R_1 \text{ for some } s \in S\},$$
$$\varphi(x) = \{s; (s, x) \in R_1\} \text{ for any } x \in L,$$
$$C = \{(u, v); (s, us'v) \in R_2 \text{ for some } s, s' \in S, u, v \in V^*\},$$
$$\psi(u, v) = \{(s, s'); (s, us'v) \in R_2\} \text{ for any } (u, v) \in C.$$
$$G = \langle V, L, C, S, \sigma, \varphi, \psi\rangle.$$

Then G is a generalized slc-grammar. It will be called the generalized slc-grammar corresponding to H and denoted by $C(H)$. Note that $C(H)$ is a slc-grammar if H is a linear grammar.

By a direct calculation according to the definition of the operators A, C we obtain

1.2. Lemma. *$A(C(H)) = H$ for any generalized linear grammar H.* □

Our main result is the following.

1.3. Theorem. *$L(G) = L(A(G))$ for any generalized slc-grammar G.*

Proof. Let $G = \langle V, L, C, S, \sigma, \varphi, \psi\rangle$ be an arbitrary generalized slc-grammar G, $A(G) = \langle U, V, \sigma, R\rangle$. Let $s \in S$, $x \in V^*$. We prove that $s \overset{*}{\Rightarrow} x$ iff $s \in \Phi(x)$.

We denote by $V(n)$ the following assertion. If there exists an s-derivation of a string x of length $\leqq n$, then $s \in \Phi(x)$.

If there exists an $s-$derivation x of length 1 then $(s, x) \in R$ and, therefore, $(s, x) \in R_1$ whence $s \in \varphi(x) \subseteq \Phi(x)$. Thus $V(1)$ holds.

Let $m \geqq 1$ be an integer and suppose the validity of $V(m)$. Let $x \in V^*$ have an s-derivation of length $m + 1$: $s = t_0, t_1, \ldots, t_{m+1} = x$. Then $t_1 = us'v$ for some suitable $(s, us'v) \in R$ where $u, v \in V^*$. Hence $(s, s') \in \psi(u, v)$ and, clearly, $x = ux'v$, where x' has an s'-derivation of length m. Therefore $s' \in \Phi(x')$ by induction hypothesis and thus $s \in \Phi(x)$. We have obtained $V(m + 1)$.

By induction, we obtain: if $s \overset{*}{\Rightarrow} x$ for $x \in V^*$ and $s \in S$ then $s \in \Phi(x)$ holds.

For $s \in S$, $x \in V^*$ we put $s \in \Phi'(x)$ if $s \overset{*}{\Rightarrow} x$. If $s \in \varphi(x)$ we have $(s, x) \in R$ and thus $s \overset{*}{\Rightarrow} x$ and therefore $s \in \Phi'(x)$. Thus $\varphi(x) \subseteq \Phi'(x)$.

If $x = uzv$, $u, z, v \in V^*$, $s' \in \Phi'(z)$, $(s, s') \in \psi(u, v)$ then $s' \overset{*}{\Rightarrow} z$ and $(s, us'v) \in R$. This implies $s \overset{*}{\Rightarrow} uzv = x$ and, hence, $s \in \Phi'(x)$. Then, for any $x \in V^*$, $\Phi'(x)$ is a set with the properties 1. a 2. From the minimality of $\Phi(x)$, it follows that $\Phi(x) \subseteq \subseteq \Phi'(x)$. Thus, if $s \in \Phi(x)$ then $s \overset{*}{\Rightarrow} x$.

We have proved $L(G) = L(A(G))$. □

1.4. Proposition. *$L(H) = L(C(H))$ for any generalized linear grammar H.*,

Proof. By 1.3 and 1.2, we obtain $L(C(H)) = L(A(C(H))) = L(H)$. □

1.5. Corollary. *A language is generated by an slc-grammar iff it is linear.* □

1.6. Proposition. *A language is generated by a contextual grammar iff it is generated by a linear grammar with one nonterminal symbol.*

See, for example, Păun [10], Gruska [11], [12].

Proof. Clearly a contextual grammar $G' = \langle V, L, C \rangle$ may be considered as a special case of a lc-grammar $G = \langle V, L, C, S, \sigma, \varphi, \psi \rangle$ if putting $S = \{\sigma\}$ where $\sigma \notin V$ is a new symbol, $\varphi(x) = \{\sigma\}$ for any $x \in L$ and $\psi(u, v) = (\sigma, \sigma)$ for any $(u, v) \in C$. Then, clearly, $L(G) = L(G')$. □

II. Acceptors

The ordered triple (S, E, v) where S, E are sets and v is a mapping $v\colon E \to S \times S$ is said to be a graph. The elements in S are called nodes and the elements in E edges. A mapping v assigns to any edge $e \in E$ the ordered pair of nodes $v(e) = = (s', s'')$; s' is called the initial node and s'' the final node of the edge e.

Let $s \in S$, $t \in S$ be nodes; the sequence of edges $(e_1, e_2, \ldots, e_n)$, $(n \geqq 0)$ is said to be a path from s to t if the conditions $s'_1 = s$, $s''_i = s'_{i+1}$ for $i = 1, 2, \ldots, n - 1$; $s''_n = t$ are satisfied where $(s'_i, s''_i) = v(e_i)$ for $1 \leqq i \leqq n - 1$.

The ordered 5-tuple (S, E, V, v, u), where (S, E, v) is a graph, V is a set and u is a mapping $u\colon E \to V$, is said to be a graph with labelled edges. If $(e_i)_{i=1}^n$ is a path from the node $s \in S$ to the node $t \in S$ then the sequence $(u(e_i))_{i=1}^n$ is called the description of the way.

The ordered 7-tuple $A = (S, E, V, I, F, v, u)$ such that (S, E, V, v, u) is a graph with labelled edges and $I \subseteq S$, $F \subseteq S$ are sets, is said to be an accepting graph. We shall call the elements in I start nodes and in F final nodes. The sequence $(a_i)_{i=1}^n$ is said to be accepted by an accepting graph A if there exist nodes $s \in I$ and $t \in F$

and a path from s to t such that its description is just $(a_i)_{i=1}^n$. We shall write the sequences accepted by accepting graphs as strings; thus, mostly $a_1, \ldots, a_n$ instead of $(a_i)_{i=1}^n$. We denote by $L(A)$ the set of all strings accepted by an accepting graph A.

Generally for $s \in S$, $t \in S$, it can happen that there exist $e_1 \in E$, $e_2 \in E$, $e_1 \neq e_2$ such that $v(e_1) = (s, t) = v(e_2)$, $u(e_1) = u(e_2)$. It is easy to see that $L(A) = L(A')$ if we denote by A' an accepting graph that we obtain from A omitting the edge e_2 and its description.

We can confine ourselves to accepting graphs in which for any $s \in S$, $t \in S$ and any $a \in V$ there exists at most one edge $e \in E$ such that $v(e) = (s, t)$, $u(e) = a$. This graph is called simple. Such a simple accepting graph can be described by indicating to each node $s \in S$ and to each symbol $a \in V$, the set $f(s, a)$ of nodes t such that any edge leaving the node s and entering the node t is labelled by a. Clearly, E, V, u can be reconstructed from S, V, f, I, F.

An acceptor is a 5-tuple $N = \langle S, V, f, I, F \rangle$ where S, V are sets, f a mapping of $S \times V$ into 2^S and I, F are subsets of S. Elements of S are often called states of N. A string $a_1 \ldots a_n$ of elements in V where $n \geqq 0$ is said to be accepted by N if there is a string $s_0 s_1 \ldots s_n$ of elements in S such that $s_0 \in I$, $s_n \in F$, and $s_i \in f(s_{i-1}, a_i)$ for $i = 1, 2, \ldots, n$. The string $s_0 s_1 \ldots s_n$ is also called the calculation of the string $a_1 \ldots a_n$; s_n is called the last state of this calculation. Let $L(N)$ denote the set of all strings accepted by an acceptor N; it is called the language accepted by N.

We denote by $R(A)$ an acceptor corresponding to a simple accepting graph A. Further let us denote by $S(N)$ a simple accepting graph reconstructed from an acceptor N. Clearly:

2.1. Proposition. $S(R(A)) = A$, $R(S(N)) = N$, $L(A) = L(R(A))$, $L(N) = L(S(N))$. □

We now exhibit simple examples of acceptors. Let U be a set, $D \subseteq U$ and m: $D \to U$ a mapping. Then the ordered pair $P = (U, m)$ is said to be a machine of Pawlak, or briefly, a P-machine. Clearly, $D = D_m$ is the domain of the mapping m. A string $a_1 \ldots a_n$ where $n \geqq 2$, $a_i \in D$, $a_{i+1} = m(a_i)$ for $i = 1, 2, \ldots, n-1$, $a_n \in U - D$ is said to be a calculation of the P-machine. We denote by $L(P)$ the set of all calculations of the P-machine P. $L(P)$ is said to be the language generated by P. For any P-machine we construct Pawlak's accepting graph in the following way: we choose a new element $\omega \notin U$ and we put $S = U \cup \{\omega\}$, $E = \{(a, m(a));\ a \in D\} \cup \{(a, \omega);\ a \in U - D\}$, $V = U \cup \{\omega\}$, $I = D$, $F = \{\omega\}$, $v = \mathrm{id}_E$, $u(a, b) = a$ for any $(a, b) \in E$. Then the ordered 7-tuple $A = (S, E, V, I, F, v, u)$ is said to be Pawlak's accepting graph or a P-graph assigned to the P-machine P. We denote it by $Q(P)$. It is easy to see from the definition that $Q(P) = A$ has the following properties:

(1) any $a \in D$ is left by precisely one edge, the edge enters $m(a)$ and it is labelled by the symbol a,

(2) any $a \in U - D$ is left by precisely one edge; the edge enters ω and it is labelled by the symbol a,

(3) there are no other edges in A.

This implies immediately:

2.2. Theorem. *Let P be a P-machine. Then $L(P) = L(Q(P))$.* □

In what follows, we consider the following properties of an acceptor $\langle S, V, f, I, F\rangle$.

(i) $S = V$.

(ii) $f(s, a) \neq \emptyset$ for $s \neq a$, $s \in S$, $a \in S$.

(iii) There exists an element $\omega \in S$ such that $F = \{\omega\}$, $f(\omega, \omega) = \emptyset$.

(iv) for arbitrary $s \in I$, there exists precisely one $t \in S - \{\omega\}$ such that $f(s, s) = \{t\}$.

(v) $f(s, s) = \{\omega\}$ for an arbitrary $s \in S - I - \{\omega\}$.

2.3. Proposition. *For any acceptor $N = \langle S, V, f, I, F\rangle$ with the properties (i)–(v), there exists a P-machine P such that $S(N) = Q(P)$.*

Proof. We put $U = V - \{\omega\}$, $m(a) = f(a, a)$ for any $a \in I$. Then $P = (U, m)$ is a P-machine. If constructing $Q(P)$, it is easy to see that it equals $S(N)$. □

Let $N = \langle S, V, f, I, F\rangle$, $N' = \langle S', V, f', I', F'\rangle$ be acceptors, $h: S \to S'$ a bijection. The bijection h is said to be an isomorphism of N onto N' if it holds: $I' = h[I]$, $F' = h[F]$, $f'(s', a) = h[f(h^{-1}(s'), a)]$ for any $a \in V$, $s' \in S'$.

Let $N = \langle S, V, f, I, F\rangle$, S' be a set, $h: S \to S'$ a bijection. We put $f'(s', a) = h[f(h^{-1}(s'), a)]$ for any $s' \in S'$, $a \in V$, $I' = h[I]$, $F' = h[F]$. Then $N' = \langle S', V, f', I', F'\rangle$ is an acceptor and h is an isomorphism of N onto N'. We say that this acceptor is defined by means of the bijection h.

We summarize

2.4. Fact. *An arbitrary bijection of the set S of states of an acceptor $N = \langle S, V, f, I, F\rangle$ defines an acceptor isomorphic with N.* □

We now define acceptors with special properties. Let $N = \langle S, V, f, I, F\rangle$ be an acceptor. We say that N is normal if $S \cap V = \emptyset$, N is special when card $V =$ card S and N is exceptional when $V = S$.

As special cases of 2.4. we obtain

2.5. Proposition. *There exists a special and also a normal acceptor that is isomorphic with an arbitrary given acceptor.* □

For any isomorphic acceptors N, N' the graphs $S(N)$ and $S(N')$ differ only in nodes while the structures on the sets of nodes are the same in both cases. If replacing any node in $S(N)$ by the corresponding node in $S(N')$, a labelled path in $S(N')$ corresponds to any labelled path in $S(N)$ in such a way that the labellings are the same. Hence, by 2.1.

2.6. Proposition. *Two isomorphic acceptors accept the same language.* □

Let $N = \langle S, V, f, I, F \rangle$ be a special acceptor, $h\colon S \to V$ a bijection. The bijection h is said to be an α-bijection if $f(s, a) = \emptyset$ holds for any $a \in V$, $s \in S$ with the property $h(s) \neq a$. We say that an acceptor has the property (α') if it has at least one α-bijection. Particularly, if $S = V$ and if id_S is an α-bijection of an acceptor $N = \langle S, V, f, I, F \rangle$ we say that N has the property (α).

It follows immediately from the definition that an acceptor with the property (α) is exceptional. It is easy to see that any acceptor with the property (α) has the property (α'), too.

Then

2.7. Proposition. *To any acceptor with the property* (α') *there exists an isomorphic acceptor with the property* (α). □

2.8. Proposition. *Acceptors with the property* (α) *and acceptors with the property* (α') *accept the same class of languages.* □

Let $N = \langle S, V, f, I, F \rangle$ be an acceptor with the property (α'), $h\colon S \to V$ its α-bijection. The α-bijection is said to be a β-bijection if it has the following properties:

(β_1) $f(s, h(s)) \neq \emptyset$ iff $s \in I$,

(β_2) $F = S - I$.

We say that an acceptor has the property (β') if it has at least one β-bijection. Particularly if $S = V$ and id_S is a β-bijection of an acceptor $N = \langle S, V, f, I, F \rangle$ we say that N has the property (β).

Clearly, any acceptor with the property (β) has the property (β'), too.

Then

2.9. Proposition. *To any acceptor with the property* (β') *there exists an isomorphic acceptor with the property* (β). □

2.10. Proposition. *Acceptors with the property* (β) *and acceptors with the property* (β') *accept the same class of languages.* □

Let $N = \langle S, V, f, I, F \rangle$ be an acceptor with the property (α'), $h\colon S \to V$ its α-bijection. We say that the α-bijection is a π-bijection if it has the following properties:

(g_1) there exists $\omega \in S - I$ such that $F = \{\omega\}$, $f(\omega, h(\omega)) = \emptyset$,

(g_2) for any $s \in I$ there exists just one $t \in S - \{\omega\}$ such that $f(s, h(s)) = \{t\}$,

(g_3) $f(s, h(s)) = \{w\}$ for any $s \in S - I - \{\omega\}$.

We say that an acceptor has the property (π') if it has at least one π-bijection. Particularly if $S = V$ and id_S is a π-bijection of an acceptor $N = \langle S, V, f, I, F \rangle$, we say that N has the property (π).

Clearly any acceptor with the property (π) has the property (π'), too.

2.11. Proposition. *To any acceptor with the property* (π') *there exists an isomorphic acceptor with the property* (π). □

2.12. Proposition. *Acceptors with the property* (π) *and acceptors with the property* (π') *accept the same class of languages.* □

If we compare the properties $(i)-(v)$ with the property (π) we see that an acceptor has the property (π) iff it has the properties $(i)-(v)$. From here and from 2.3, 2.2, 2.1 we have

2.13. Proposition. *To any acceptor* N *with the property* (π) *there exists a P-machine* P *such that* $S(N) = Q(P)$, $L(N) = L(P)$. □

Simillary 2.2 gives

2.14. Proposition. $R(Q(P))$ *has the property* (π) *for any P-machine* P *and* $L(P) = L(R(Q(P)))$ *holds.* □

2.15. Proposition. *Acceptors with the property* (π) *accept the same class of languages as P-machines generate.* □

III. Acceptors and generalized LC-grammars

For an arbitrary normal acceptor $N = \langle S, V, f, I, F\rangle$ we put,

$$L = \{\Lambda\},$$
$$C = \{(\Lambda, a); a \in V\},$$
$$\varphi(\Lambda) = I,$$
$$\psi(\Lambda, a) = \{(s, s'); s \in f(s', a)\} \quad \text{for any } a \in V,$$
$$G = \langle V, L, C, S, F, \varphi, \psi\rangle,$$
$$K(N) = G.$$

Then $K(N)$ is a generalized lc-grammar that has the following properties:

(*a*) $L = \{\Lambda\}$,

(*b*) $C \subseteq \{\Lambda\} \times V$.

This generalized lc-grammar is said to be a generalized right regular lc-grammar (shortly rrlc-grammar). Note that $K(N)$ is an rrlc-grammar if the acceptor is finite.

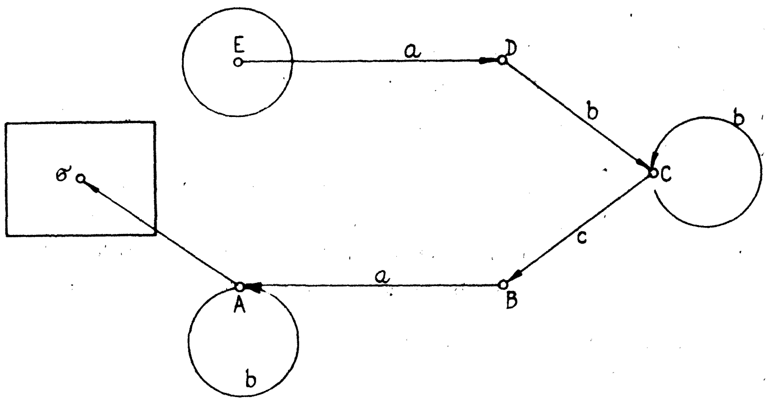
Now let $G = \langle V, L, C, S, T, \varphi, \psi\rangle$ be a generalized rrlc-grammar. We put

$$I = \varphi(\Lambda),$$
$$f(s, a) = \{t; (t, s) \in \psi(\Lambda, a)\} \quad \text{for any } s \in S, a \in V,$$
$$N = \langle S, V, f, I, F\rangle,$$
$$M(G) = N.$$

Then $M(G)$ is a normal acceptor. Note that the normal acceptor $M(G)$ is finite if G is an rrlc-grammar.

3.1. Example. *Let* $N = \langle S, V, f, I, F\rangle$ *be a normal acceptor such that* $S = \{\sigma, A, B, C, D, E\}$, $V = \{a, b, c\}$, $I = \{E\}$, $F = \{\sigma\}$ *and let* f *be given by the following table:*

f	a	b	c
σ	$\emptyset$	$\emptyset$	$\emptyset$
A	$\emptyset$	$\{\sigma, A\}$	$\emptyset$
B	$\{A\}$	$\emptyset$	$\emptyset$
C	$\emptyset$	$\{C\}$	$\{B\}$
D	$\emptyset$	$\{C\}$	$\emptyset$
E	$\{D\}$	$\emptyset$	$\emptyset$



We suppose that E is the only start state and σ the only final state. By definitions, we have

$$\varphi(\Lambda) = \{E\}, \qquad \psi(\Lambda, a) = \{(A, B), (D, E)\},$$
$$\psi(\Lambda, b) = \{(\sigma, A), (A, A), (C, C), (C, D)\}, \qquad \psi(\Lambda, c) = \{(B, C)\},$$
$$K(N) = \langle V, \{\Lambda\}, \{(\Lambda, a), (\Lambda, b), (\Lambda, c)\}, S, \{\sigma\}, \varphi, \psi\rangle$$

3.1. Proposition. $M(K(N)) = N$ *for any normal acceptor* N.

Proof. Let $N = \langle S, V, f, I, F\rangle$. Then $K(N) = \langle V, L, C, S, F, \varphi, \psi\rangle$ and $M(K(N)) = (S, V, f', I', F\rangle$ where $I' = \varphi(\Lambda) = I$ and $f'(s, a) = \{t; (t, s) \in \psi(\Lambda, a)\} = f(s, a)$. □

3.2. Theorem. $L(G) = L(M(G))$ *for any generalized rrlc-grammar* G.

Proof. Let $G = \langle V, L, C, S, T, \varphi, \psi\rangle$, $M(G) = N = \langle S, V, f, I, T\rangle$. We denote by $V(n)$ the following assertion: If for $s \in S$, $x \in V^*$, there exists a calculation of the string of length $\leqq n$ with the last state s, then $s \in \Phi(x)$.

If for $s \in S$, $x \in V^*$ there exists a calculation of length 0 with the last state s then $x = \Lambda$ and $s \in I \cap T$. Thus $s \in I = \varphi(\Lambda) \subseteq \Phi(x)$. Therefore $V(0)$ holds.

Suppose that $m \geqq 0$ and $V(m)$ holds. Let $s \in S$, $x \in V^*$ and suppose the existence of an calculation of length $m + 1$ of the string x with the last state s.

Then there exist $s_0, s_1, \ldots, s_{m+1}$ in S and $a_1, a_2, \ldots, a_{m+1}$ in V such that $x = a_1 a_2 \ldots a_{m+1}$, $s_0 \in I$, $s_{m+1} = s$ and $s_i \in f(s_{i-1}, a_i)$ for $i = 1, 2, \ldots, m + 1$. Therefore $s_0 s_1 \ldots s_m$ is a calculation of the string $x' = a_1 a_2 \ldots a_m$ of length m with the last state s_m. By induction hypothesis, we obtain $s_m \in \Phi(x')$. Since $s_{m+1} \in f(s_m, a_{m+1})$ we have $s_{m+1} \in \Phi(x' a_{m+1}) = \Phi(x)$.

By induction, it follows that the existence of a calculation of the string x with the last state s implies $s \in \Phi(x)$.

Let us put $s \in \Phi'(x)$ if there exists a calculation of the string x with the last state s.

If $s \in \varphi(\Lambda)$, then $s \in I$ holds and thus there exists a calculation of Λ with the last state s. Therefore $\varphi(\Lambda) \subseteq \Phi'(\Lambda)$.

Let us have $x = uzv$, $s' \in \Phi'(z)$, $(s, s') \in \psi(u, v)$. Then $u = \Lambda$, $v = a \in V$. There exists a calculation of z with the last state s'. Since $(s, s') \in \psi(\Lambda, a)$, we obtain $s \in f(s', a)$ and thus there exists a calculation of $za = uzv = x$ with the last state s; this implies $s \in \Phi'(x)$.

We have seen that $\Phi'(x)$ has the properties 1 and 2 for any $x \in V^*$. From the minimality of $\Phi(x)$ it follows that $\Phi(x) \subseteq \Phi'(x)$, i.e., if $s \in \Phi(x)$, there exists a calculation of x with the last state s. This can be expressed as follows: $x \in \Psi(s)$ iff there exists a calculation of x with the last state s. Therefore: $x \in L(G) = \bigcup_{s \in T} \Psi(s)$ iff there exists a calculation of x with the last state in T, i.e., iff $x \in L(N)$. □

By 3.1 and 3.2, we obtain

3.3. Proposition. *$L(N) = L(M(K(N))) = L(K(N))$ for any normal acceptor N.* □

3.4. Theorem. *A language is regular iff it is generated by rrlc-grammar.* □

Let $N = \langle S, V, f, I, F \rangle$ be a normal acceptor with the property (α'). Then $K(N)$ is a generalized rrlc-grammar that has the following properties:

(*c*) there exists a bijection $h : S \to V$

(*d*) for any $a \in V$ there exists $S(a) \subseteq S$ such that $\psi(\Lambda, a) = \{(s, h^{-1}(a)), s \in S(a)\}$. [It is sufficient to put $S(a) = f(h^{-1}(a), a)$.]

Such a generalized rrlc-grammar is said to be a generalized α'rrlc-grammar. We obtain

3.5. Proposition. *If N has the property (α') then $K(N)$ is a generalized α'rrlc-grammar.* □

3.6. Proposition. *If G is a generalized α'rrlc-grammar, then $M(G)$ has the property (α').* □

3.7. Corollary. *Let L be a language. Then the following conditions are equivalent:*

(i) *L is accepted by an acceptor of the class (α').*

(ii) *L is accepted by an acceptor of the class* (α).

(iii) *L is generated by a generalized α'rrlc-grammar.* $\square$

Let $N = \langle S, V, f, I, F\rangle$ be a normal acceptor with the property (β'). Then $K(N) = \langle V, L, C, S, F, \varphi, \psi\rangle$ is a generalized $\alpha'rrlc$-grammar that has the following properties:

(e_1) $F = S - \varphi(\Lambda)$

(e_2) For any $a \in V$, the condition $S(a) \neq \emptyset$ is satisfied iff $h^{-1}(a) \in \varphi(\Lambda)$.

Such a generalized α'rrlc-grammar is said to be a generalized β'rrlc-grammar. Immediately it follows

3.8. Proposition. *If N has the property (β') then $K(N)$ is a generalized β'rrlc-grammar.* $\square$

3.9. Proposition. *If G is a generalized rrlc-grammar, then $M(G)$ has the property (β').* $\square$

3.10. Corollary. *Let L be a language. Then the following conditions are equivalent:*

(i) *L is accepted by an acceptor of the class* (β').

(ii) *L is accepted by an acceptor of the class* (β).

(iii) *L is generated by a generalized β'rrlc-grammar.* $\square$

If $N = \langle S, V, f, I, F\rangle$ is a normal acceptor with the property (π') then $K(N) = \langle V, L, C, S, F, \varphi, \psi\rangle$ is a generalized α'rrlc-grammar having the following properties:

(i) It is simple, $F = \{\omega\}$, $\omega \in S - I$, $\psi(\Lambda, h(\omega)) = \emptyset$

(j) for any $h(a) \in V$ with the property $a \in I$ there exists exactly one $b \in S - \{\omega\}$ such that $\psi(\Lambda, h(a)) = \{(b, a)\}$,

(k) for any $h(a) \in V$ with the property $a \in S - I - \{\omega\}$ the condition $\psi(\Lambda, h(a)) = \{(\omega, a)\}$ is satisfied.

Such a generalized α'rrlc-grammar is said to be a generalized π'rrlc-grammar.

Clearly

3.11. Proposition. *If N is an acceptor of the class (π') then $K(N)$ is a generalized π'rrlc-grammar.* $\square$

3.12. Proposition. *If G is a generalized π'rrlc-grammar then $M(G)$ is an acceptor of the class* (π'). $\square$

3.13. Corollary. *Let L be a language. Then the following conditions are equivalent:*

(i) *L is accepted by an acceptor of the class* (π').

(ii) *L is accepted by an acceptor of the class* (π).

(iii) *L is generated by a P-machine.*

(iv) *L is generated by a generalized π'rrlc-grammar.*

REFERENCES

[1] Marcus, S.: *Contextual grammars*. Rev. Roum. Math. Pures et Appl. 14 (1969), 1525—1934.

[2] Novotný, M.: *On a class of contextual grammars*. Cahiers de linguistique théorique et appliquée 11 (1974), Fasc. 2, 313—314.

[3] Novotný, M.: *On multicontextual grammars* (manuscript).

[4] Novotný, M.: *On some variants of contextual languages*. Rev. Roum. Math. Pures et Appl. 21 (1976), 1053—1062.

[5] Novotný, M.: *Each generalized contextual language is context sensitive*. Rev. Roum. Math. Pures et Appl. 21 (1976), 353—362.

[6] Ginsburg, S.: *The mathematical theory of context-free languages*. Mc Graw Hill Book Company New York, 1966.

[7] Pawlak, Z.: *Maszyny programowane*. Algorytmy 10, volume 5, 1969, 5—19.

[8] Čulík, K.: *Some notes on finite state languages and events represented by finite automata using labelled graphs*. Čas. pěst. mat., 86, 43—55 (1961).

[9] Marcus, S.: *Gramatici şi automate finite*. Editura academiei republicii populare Romine, 1964.

[10] Păun, G.: *On the smallest number of nonterminals required to generate a context-free language*. Rev. Anal. Numér. Théor. Approx. 18 (41), t. 2 (1976), 203—208.

[11] Gruska, J.: *On a classification of context-free languages*. Kybernetika (Prague) 3, Nr. 1 (1967), 22—29.

[12] Gruska, J.: *Descriptional complexity of context-free languages*. Mathematical Foundations of Computer Science. Proceedings of Symposium and Summer School. High Tatras, September 3 to 8, 1973 (1973), 71—83.

J. Ostravský
762 72 Gottwaldov, nám. Rudé armády 275
Czechoslovakia