ANALELE ŞTIINŢIFICE ALE UNIVERSITAŢII „AL. I. CUZA" DIN IAŞI
Tomul XXXIII, s. I a, Matematică, 1987, fasc. 4

GENERATING CONTEXT SENSITIVE LANGUAGES BY ATTRIBUTE GRAMMARS

BY

G. GRIGORAŞ

1. **Introduction.** In his historic paper, Knuth [4] introduces a formalism, called attribute grammar, which constitutes a formal definition of all context-free and context-sensitive language properties on the one hand, and a formal specification of the semantic analysis on the other. In [1] the main results on attribute grammars, including theoretical and practical aspects, are surveyed. In [6] the attribute grammars are applied to semantic analysis in compiler construction.

It is known from Knuth [4] that purely synthesized attribute grammar have the same power as Turing machines. However other classes of attributes grammars are introduced and the relationships between languages defined of them are studied in the literature. In this paper we give a construction of an attribute grammar starting from a context sensitive grammar, which generates the same language as the second one.

2. **Preliminaries.** We use the formal language terminology from [2] and [3]. A context sensitive grammar is a grammar $G=(V, T, P, S)$ whose productions are of the form $\alpha A\gamma \to \alpha\beta\gamma$, where $A \in V$, α, $\gamma \in (V \cup T)^*$ and $\beta \in \in (V \cup T.)^+$ or $S \to \lambda$, in which case S does not appear in the righthand side of any rule.

An attribute grammar is composed by a reduced context-free grammar $G_0=(V, T, P, S)$, called the support grammar, a finite set of attributes $\mathcal{A} = \bigcup_{X \in V} \mathcal{A}(X)$ and a finite set of semantic rules $\mathcal{R} = \bigcup_{p \in P} \mathcal{R}(p)$. For each nonterminal symbol $X \in V$, $\mathcal{A}(X)$ is the set of attributes of X and is a disjoint union of $\mathcal{A}^s(X)$-synthesized attribute of X and $\mathcal{A}^i(X)$-inherited attributes of X. An attribute occurence of X is denoted $X.\,a$, where a is the name of attribute. For each production $p : X_0 \to u_0 X_1 u_1 X_2 u_2 \ldots X_n u_n$, the set $\mathcal{R}(p)$ of semantic rules for p, contains a rule for each attribute a from $\mathcal{A}^s(X_0) \bigcup_{k=1}^{n} \mathcal{A}^i(X_k)$. The rule is of the form : $X_i\,.\,a := f(X_j\,.\,b, \ldots, X_k\,.\,c)$, where f is a calculable function. We note that each attribute has a set from which this attribute takes values.

Example : $G_0=(V, T, P, S)$, $V=\{S, A, B, C\}$, $T=\{a, b, c\}$, S has the synthesized attribute *Answer* with values in $\{true, false\}$ and A, B, C have the synthesized attribute n whose values are in N. The syntactic (context free) and semantic rules are :

1. $S \to ABC$

$$S.\ Answer := \textit{if } A.n = B.n = C.n \textit{ then true}$$
$$\textit{else false}$$

2. $A \to aA.\quad A_0.n := A_1.n+1$

3. $A \to \lambda.\quad A.n := 0$

4. $B \to bB.\quad B_0.n := B_1.n+1$

5. $B \to \lambda.\quad B.n := 0$

6. $C \to cC.\quad C_0.n := C_1.n+1$

7. $C \to \lambda.\quad C.n := 0$

Each node labeled by $X \in V$ in a tree of a word $w \in L(G_0)$ is associated with a particular set of values for the attributes of X. These values are calculated by semantic rules. If all these values can be computed for each derivation tree in grammar G_0 then the attribute grammar is called well-formed (or noncircular) [4].

In our example, $L(G_0) = \{a^i b^j c^k / i, j, k \geq 0\}$. It is obvious that for each $w \in L(G_0)$ all attribute values can be calculated and S. Answer $=$ *true* if and only if $w = a^n b^n c^n$, for a $n \in \mathbf{N}$.

In the sequel, we consider attribute grammars AG_0 (with the support grammar G_0) in which the start symbol S has a synthesized attribute *Answer* and we define the language generated by AG_0 by:

$$L(AG_0) = \{w / w \in T^*,\ S \overset{*}{\underset{G_0}{\Rightarrow}} w,\ S.\ Answer = true\}.$$

In the next section, starting from a context sensitive grammar G, we construct an attribute grammar AG_0 such that $L(AG_0) = L(G)$.

3. **The attribute grammar of a context sensitive-grammar.** Let Σ be an alphabet and $\mathbf{N}$ be the set of natural numbers. We define a binary operation $\circ$ on $\Sigma^* \times \mathbf{N}$ in the following way: if $(\alpha, n), (\beta, m) \in \Sigma^* \times \mathbf{N}$ then $(\alpha, n) \circ (\beta, m) = (\alpha\beta, k)$, where $\alpha\beta$ is the concatenation of α and β and $k = \max\{n, m\}$. It is obvious that $(\Sigma^* \times \mathbf{N}, \circ)$ is a monoid: the binary operation $\circ$ is associative and $(\lambda, 0)$ is a two-sided identity for $\Sigma^* \times \mathbf{N}$.

We extend the operation just defined to sets in $2^{\Sigma^* \times \mathbf{N}}$ in a natural way: for $U, V \in 2^{\Sigma^* \times \mathbf{N}}$ we put $U \circ V = \{(\alpha, n) \circ (\beta, m) / (\alpha, n) \in U,\ (\beta, m) \in V\}$.

We define $\varnothing \circ U = U \circ \varnothing = U$. This operation will be used in the semantic rules of an attribute grammar.

Let $G = (V, T, P, S)$ be a context sensitive grammar. Without loss of generality, we suppose that S does not appear in the right hand side of any productions. Let us consider a restriction now: if $\alpha A \gamma \to \alpha\beta\gamma$ and $\alpha_1 A \gamma_1 \to \alpha_1 \beta_1 \gamma_1$ are in P then $\alpha = \alpha_1$ and $\gamma = \gamma_1$. We consider the context free grammar $G_0 = (V, T, P_0, S)$ where $P_0 = \{A \to \beta / \alpha A \gamma \to \alpha\beta\gamma \in P\}$. Now, we define an attribute grammar with G_0 the support grammar. The set of attribute are:

— the start symbol S has an synthesized attribute *Answer* whose values are in $\{true, false\}$;

— each nonterminal $A (A \neq S)$ from V has the synthesized attributes: *Answer* with values $\{true, false\}$, *rlc* (rule left context), *rrc* (rule right context) and *fr* (frontier) with values in the set $2^{(V \cup T)^* \times \mathbf{N}}$, and inherited attributes: *level* with values in $\mathbf{N}$, *lc* (left context) and *rc* (right context) with values in $2^{(V \cup T)^* \times \mathbf{N}}$.

Now we define the semantic rules for each type of production in such a way that in a derivation tree of G_0 for the word $w \in T^*$ the value of S. *Answer* is *true* if and only if the word w can be derived from S in the context sensitive grammar G. These rules are:

1. For the production $S \to \lambda$ (if this is a production in G_0) or $S \to w$, $w \in T^*$, we put

$$S.\ Answer := true.$$

2. For the production of the form

$$S \to u_0 A_1 u_1 A_2 u_2 \ldots A_m u_m,$$

where $u_i \in T^*$, $A_i \in V$ $0 \leq i \leq m$, we consider the rules:

$$S.\ Answer := \begin{cases} true \text{ iff } A_i.\ Answer = true,\ 1 \leq i \leq m, \text{ and } rlc,\ rrc \\ \qquad\qquad \text{are compatible for } A_1, A_2, \ldots A_m; \\ false \text{ otherwise.} \end{cases}$$

for $i = 1, 2, \ldots, m$:

$$A_i.level := 1;$$

$$A_i.lc := (u_0, 1) \circ A_1.fr \circ (u_1, 1) \circ \ldots A_{i-1}.fr \circ (u_{i-1}, 1);$$

$$A_i.rc := (u_i, 1) \circ A_{i+1}.fr \circ \ldots \circ A_m.fr \circ (u_m, 1).$$

We will explain later what we mean by „*rlc*, *rcc* are compatible for $A_1, A_2, \ldots, A_m$".

3. For the productions of the form

$$A \to v_0 B_1 v_1 B_2 v_2 \ldots B_m v_m \quad (A \neq S),$$

for which in G there is the production $\alpha A \gamma \to \alpha\beta\gamma$, $\beta = v_0 B_1 v_1 B_2 v_2 \ldots B_m v_m$ and $|\alpha| = k$, $|\gamma| = l$ we define:

$$B_i.level := A.level + 1,\ 1 \leq i \leq m;$$

$$A.fr := \{(A, A.level)\} \cup (v_0, B_1.level) \circ B_1.fr \circ \ldots \circ B_m.fr \circ$$
$$\circ (v_m, B_m.level);$$

for $i = 1, 2, \ldots, m$:

$$B_i.lc := A.lc \circ (v_0, B_1.level) \circ B_1.fr \circ \ldots \circ B_{i-1}.fr \circ$$
$$\circ (v_{i-1}, B_{i-1}.level)$$

$$B_i.rc := (u_i, B_i.level) \circ B_i.fr \circ \ldots \circ B_m.fr \circ (v_m, B_m.level) \circ A.rc$$

$$A\,.\,rlc := \begin{cases} \{(x,\ n) \mid (x,\ n) \in A\,.\,lc,\ \alpha \in \mathrm{LAST}_k(x)\} & \text{if } \alpha \neq \lambda; \\ \{(\lambda,\ A\,.\,level)\} & \text{if } \alpha = \lambda. \end{cases}$$

$$A\,.\,rrc := \begin{cases} \{(x,\ n) \mid (x,\ n) \in A\,.\,rc,\ \gamma \in \mathrm{FIRST}_l(x)\} & \text{if } \gamma \neq \lambda; \\ \{(\lambda,\ A\,.\,level)\} & \text{if } \gamma = \lambda. \end{cases}$$

$$A\,.\,Answer := \begin{cases} true \text{ if } B_i\,.\,Answer = true,\ 1 \leq i \leq m, \text{ and } rlc,\ rrc, \text{ are compatible for } B_1, \ldots, B_m; \\ false \text{ otherwise.} \end{cases}$$

4. For the productions of the form $A \to w$, $w \in T^*$, and in G there is the production

$$\alpha A \gamma \to \alpha w \gamma,\ |\alpha| = k,\ |\gamma| = l, \text{ we put:}$$

$$A\,.\,fr := \{(A,\ A\,.\,level),\ (w,\ A\,.\,level + 1)\}$$

$$A\,.\,rlc := \begin{cases} \{(x,\ n) \mid (x,\ n) \in A\,.\,lc,\ \alpha \in \mathrm{LAST}_k(x)\}, & \text{if } \alpha \neq \lambda; \\ \{(\lambda,\ A\,.\,level)\} & \text{for } \alpha = \lambda. \end{cases}$$

$$A\,.\,rrc := \begin{cases} \{(x,\ n) \mid (x,\ n) \in A\,.\,rc,\ \gamma \in \mathrm{FIRST}_l(x)\}, & \text{if } \gamma \neq \lambda; \\ \{(\lambda,\ A\,.\,level)\}, & \text{if } \gamma = \lambda. \end{cases}$$

$$A\,.\,Answer := \begin{cases} true \text{ if } A\,.\,rlc \neq \emptyset \text{ and } A\,.\,rrc \neq \emptyset; \\ false \text{ otherwise.} \end{cases}$$

Let now explain what means „*rlc*, *rrc* are compatible for $A_1, A_2, \ldots, A_m$". Let consider the sets $A_1\,.\,rlc$, $A_1\,.\,rrc$, $A_2\,.\,rlc$, $A_2\,.\,rrc$, $\ldots$, $A_m\,.\,rlc$, $A_m\,.\,rrc$ in this order. We say that *rlc*, *rrc* are compatible for $A_1, \ldots, A_m$ iff for each i, $1 \leq i \leq m$, the sets $A_i\,.\,rrc$ and $A_{i+1}\,.\,rlc$ have one of the following two properties:

1) one of them is $\{(\lambda,\ n)\}$ for some $n \in \mathbb{N}$ or,

2) there exists in $A_i\,.\,rrc$ (or in $A_{i+1}\,.\,rlc$) a pair of the form $(\alpha,\ n)$ such that $n = A_i\,.\,level$ (respectively $n = A_{i+1}\,.\,level$).

Let us denote by AG_0 the attribute grammar constructed above, starting from the context sensitive grammar G.

Theorem 1. *The attribute grammar AG_0 is noncircular.*

Proof. Let t be a derivation tree in the support grammar G_0. From the definitions of the semantic rules, an algorithm that calculates all the attribute occurrences in t, performs this in the following order:

1. Calculates the values of *level* occurrences;
2. Calculates the values of *fr* occurences. These values depend only on the values of *level*;
3. Calculates the values of *lc* and *rc* occurrences; these values depend only on *level* and *fr* values;
4. Calculates the values of *rlc* and *rrc* occurrences. These values depend on the *level*, *lc* and *rc* values;
5. Calculates the values of the *Answer* occurrences.

We note that five visits in the derivation tree t are necessary for computing all the attribute occurrences.

Theorem 2. $L(G) = L(AG_0)$.

Proof. Let $w \in L(G)$. There exists a derivation $S \overset{*}{\underset{G}{\Rightarrow}} w$ and in G_0 we can construct a derivation tree t such that, if in $S \overset{*}{\underset{G}{\Rightarrow}} w$ the production $\alpha A \gamma \Rightarrow \alpha \beta \gamma$ is performed than in t the production $A \to \beta$ is used. Now we can use the algorithm from theorem 1 for decorate the tree t with the value of attribute in every node. From definitions of attribute rules it follows that $S.\ Answer = true$, hence $w \in L(AG_0)$.

Conversely, if $w \in L(AG_0)$ then in G_0 we have a derivation tree t in which, after attribute values computing, the value of $S\,.\,Answer$ is *true*. It follows that $A\,.\,Answer$ is *true* for all $A \in V$ appearing in the tree t and, on every level of the tree the values of the attributes *rlc* and *rrc* are compatible for $B_1, \ldots, B_m$ (if the production is $A \to u_0 B_1 u_1 \ldots B_m u_m$). From the definition of compatibility of contexts, for every production of the form $A \to \beta$ from G_0 we can choose a production of the form $\alpha A \gamma \to \alpha \beta \gamma$ in G such that replacing $A \to \beta$ in the derivation $S \overset{*}{\underset{G}{\Rightarrow}} w$ by $\alpha A \gamma \to \alpha \beta \gamma$ we find a derivation $S \overset{*}{\underset{G}{\Rightarrow}} w$ so $w \in L(G)$.

Let us discuss now the case of existing two (or more) productions of the form $\alpha A \gamma \to \alpha \beta \gamma$ and $\alpha_1 A \gamma_1 \to \alpha_1 \beta \gamma_1$ in G, with $\alpha \neq \alpha_1$ or/and $\gamma \neq \gamma_1$. In this case in G_0 we consider the productions $A \to \beta X_1$, $A \to \beta X_2$, $X_1 \to \lambda$, $X_2 \to \lambda$, X_1, X_2 being new variables with the same attributes like old variables and the following semantic rules:

$$\begin{aligned} A \to \beta X_1: \quad & X_1\,.\,level := A\,.\,level + 1; \\ & X_1\,.\,lc := \emptyset; \\ & X_1\,.\,rc := \emptyset. \\ X_1 \to \lambda: \quad & X_1\,.\,fr := \emptyset; \\ & X_1\,.\,rlc := \emptyset \\ & X_1\,.\,rrc := \emptyset; \end{aligned}$$

and the same rules for X_2.

By our convention about the operation $\circ$ ($U \circ \emptyset = \emptyset \circ U = U$) these definition does not perturb the previous construction.

REFERENCES

1. Deransart P., Jourdan M., Lorho B. — *A survey on Attribute Grammars part I: Main results on Attribute Grammars*, Rapport RR 485, INRIA, Rocquencourt, (1986).
2. Grigoraş G. — *Formal Languages and Compiler Construction* (in romanian), Iaşi University, 1985.

3. Hopcroft J., Ullman J. D. — *Introduction to Automata Theory, Languages and Computation*, Addison Wesley, Reading Massachussetts, 1979.

4. Knuth D. E. — *Semantics of Context-free Languages*, Math. System Theory 2(1968) 127–145, 5(1971), 95–96.

5. Waite W. M., Goos G. — *Compiler Construction*, Springer-Verlag, New York, Berlin, Heidelberg, Tokyo, 1984.

Received 15.XI.1987

Faculty of Mathematics
University of Iaşi
Iaşi, R.S. Romania

ANALELE ŞTIINŢIFICE ALE UNIVERSITĂŢII „AL. I. CUZA" DIN IAŞI
Tomul XXXIII, s. I a. Matematică, 1987, fasc. 4

ON THE MYHILL-NERODE RELATIONS

BY

ALEXANDRU MATEESCU

1. **Definitions and notations.** As usually, Σ^* is the free monoid generated by the alphabet Σ and $P(\Sigma^*)$ is the set of all subsets of Σ^*, i.e. the set of all languages over Σ. The null element of Σ^* is λ. If $L\subseteq\Sigma^*$ then the relation $\sim_L ({}_L\sim)$ (cf. [2], [4], [5], [8] etc.) is: $u\sim_L v\,(u\,{}_L\sim v)$ if and only if for every $x\in\Sigma^*$, $ux\in L\Leftrightarrow vx\in L\,(xu\in L\Leftrightarrow xv\in L)$. The relation $\approx_L$ is defined $u\approx_L v$ if and only if for every x, $y\in\Sigma^*$, $xuy\in L\Leftrightarrow xvy\in L$.

$\sim_L ({}_L\sim)$ (see [1], [2], [5], [8]) is the greatest equivalence relations on Σ^*, right (left) invariant, such that L is closed with respect to $\sim_L ({}_L\sim)$. Moreover, $\approx_L$ is the greatest congruence on Σ^* such that L is closed with respect to it.

The Myhill-Nerode theorem (1957–1958) says that a language L is regular if and only if $\approx_L$ is of finite index, respectively, if and only if $\sim_L$ $({}_L\sim)$ is of finite index.

Notations. If L, $L_1\subseteq\Sigma^*$ then, left quotient of L by L_1 is $L_1\backslash L$ and *right quotient of L* by L_1 is L/L_1, where:

$$L_1\backslash L=\{x\mid \exists y\in L_1,\ yx\in L\},\quad L/L_1=\{x\mid \exists y\in L_1,\ xy\in L\}.$$

and let $\#$ be such that $\#\notin\Sigma$, the quotient of L by L_1 is:

$$\frac{L}{L_1}=\{x\#y\mid \exists t\in L_1,\ xty\in L\}.$$

Remark. $P(\Sigma^*)$ will be considered as a monoid with respect to the product of languages and the null element $\{\lambda\}$.

2. **Main results. Definition 1.** *If $L\subseteq\Sigma^*$ is a fixed language, then we define the relations $\simeq_L$, ${}_L\simeq$, $\underset{L}{\simeq}$ in $P(\Sigma^*)$:*

$$\text{for } L_1,\ L_2\subseteq\Sigma^*,\qquad L_1\simeq_L L_2\Leftrightarrow L_1\backslash L=L_2\backslash L.$$

$$\text{for } L_1,\ L_2\subseteq\Sigma^*,\qquad L_1\,{}_L\simeq L_2\Leftrightarrow L/L_1=L/L_2,$$

$$\text{for } L_1,\ L_2\subseteq\Sigma^*,\qquad L_1\underset{L}{\simeq} L_2\Leftrightarrow \frac{L}{L_1}=\frac{L}{L_2}.$$

Definition 2. *A language $L\subseteq\Sigma^*$ is weak closed with respect to the equivalence relation $\sim$ on $P(\Sigma^*)$ if and only if for any L_1, $L_2\subseteq\Sigma^*$*

if $L_1\cap L\neq 0$ *and* $L_1\sim L_2$ *then* $L_2\cap L\neq 0$ is true: