229. Covering-Languages of Grammars

By Takumi KASAI
Research Institute for Mathematical Sciences, Kyoto University

(Comm. by Kinjirô KUNUGI, M. J. A., Dec. 13, 1971)

1. Introduction.

Two derivation trees (phrase-markers) are called *congruent* in [1] if merely by relabelling of the nonterminal nodes they may be made the same. A *marker* is an equivalence class of congruent derivation trees. In this note we introduce a new type of language, called a *covering language*, which can describe the set of markers generated by a context-free grammar. The intrinsic structure of a context-free grammar G is characterized by the covering language $K(G)$ of G.

Let $G=(N, \Sigma, P, S)$ be a context-free grammar with the set of nonterminal symbols N, the set of terminal symbols Σ, the set of productions P and the initial symbol S. Each production π is usually expressed in a unique way in the following canonical form

$$\pi : X \rightarrow t_0 Y_1 t_1 \cdots t_{n-1} Y_n t_n$$

where X and Y_i $(1 \leq i \leq n)$ are nonterminal symbols and the t are possibly empty terminal words. The integer $n \geq 0$ determines the number of occurrences of nonterminal symbols at the right side of the production π and is said to be the *rank* of π. The rank of a production π is denoted by $\sigma_P(\pi)$. For each production $\pi : X \rightarrow t_0 Y_1 t_1 \cdots Y_n t_n$, let $\langle t_0, t_1, \cdots, t_n \rangle$ be an abstract symbol. We shall call this the *form* of π and the integer n is said to be the *rank* of this form. The form of π will be denoted by $f(\pi)$ and the set of all forms of the productions in P will be denoted by $f(P)$, i.e. $f(P)=\{f(\pi) \mid \pi \text{ in } P\}$. We extend f to a length preserving homomorphism $f : P^* \rightarrow \{f(P)\}^*$ by defining $f(\varepsilon)=\varepsilon$ and $f(\pi_1 \cdots \pi_k)=f(\pi_1) \cdots f(\pi_k)$.

The notation $x \overset{\alpha}{\Longrightarrow} y$ or $\alpha : x \Longrightarrow y$ means that there exists a leftmost derivation

$$D : x=x_0 \overset{\pi_1}{\Longrightarrow} x_1 \overset{\pi_2}{\Longrightarrow} \cdots \overset{\pi_n}{\Longrightarrow} x_n = y$$

such that $\alpha=\pi_1 \pi_2 \cdots \pi_n$, where in the transition from x_i to $x_{i+1} (0 \leq i < n)$ the production π_i is applied. The word $\pi_1 \pi_2 \cdots \pi_n$ is called the *associate* of D and $f(\pi_1 \pi_2 \cdots \pi_n)$ is called the *form* of D.

In this paper, unless stated otherwise, by "grammar" we shall mean context-free grammar and by "derivation" we shall mean leftmost derivation.

Given a grammar $G=(N, \Sigma, P, S)$, let

$$L(G)=\left\{w \text{ in } \Sigma^* \mid S \overset{\alpha}{\Longrightarrow} w,\ \alpha \text{ in } P^*\right\}$$

$$A(G)=\left\{\alpha \text{ in } P^* \mid S \overset{\alpha}{\Longrightarrow} w,\ w \text{ in } \Sigma^*\right\}$$

and

$$K(G)=f(A(G)).$$

The set $L(G)$ is the context-free language generated by G. The set $A(G)$ will be called the *associate language* of G, and the set $K(G)$ will be called the *covering language* of G. Given a grammar G, each element of $A(G)$ can be regarded as a derivation tree in G, and for α and β in $A(G)$, $f(\alpha)=f(\beta)$ means that α and β realize the same tree except for a relabelling of nonterminal nodes. Thus the set $K(G)$ can be regarded the set of markers generated by G.

2. Subgrammars.

Let G_1 and G_2 be grammars. If $K(G_1)\subset K(G_2)$, then G_1 is said to be a *subgrammar* of G_2 and we write $G_1 \underset{s}{\subset} G_2$. A subgrammar G_1 of G_2 is said to be *spanning* if $L(G_1)=L(G_2)$. G_1 and G_2 are *structurally equivalent* [1], written $G_1 \underset{s}{=} G_2$, if $G_1 \underset{s}{\subset} G_2$ and $G_2 \underset{s}{\subset} G_1$.

This definition differs from the definition of structural equivalence as used in [1]. It can be shown, although not done here, that these two definitions of structural equivalence are equivalent.

Example. Let $G_1=(\{S, X, Y\}, \{a, b\}, P_1, S)$ and $G_2=(\{S, X\}, \{a, b\}, P_2, S)$ be grammars, where P_1 and P_2 consist of the following productions.

P_1: $\pi_1: S\to aXb$, $\pi_2: S\to ab$ $\pi_3: X\to YXb$,
$\pi_4: X\to aSb$, $\pi_5: X\to ab$ $\pi_6: Y\to a$

P_2: $\hat{\pi}_1: S\to aSb$, $\hat{\pi}_2: S\to XSb$, $\hat{\pi}_3: S\to ab$, $\hat{\pi}_4: X\to a$.

Then we have

$A(G_1)=\{\pi_1\{\pi_3\pi_6\}^*\pi_4\}^*\{\pi_2 \cup \pi_1\{\pi_3\pi_6\}^*\pi_5\}$

$K(G_1)=\{\langle a, b\rangle\{\langle \varepsilon, \varepsilon, b\rangle\langle a\rangle\}^*\langle a, b\rangle\}^*\{\langle ab\rangle \cup \langle a, b\rangle\{\langle \varepsilon, \varepsilon, b\rangle\langle a\rangle\}^*\langle ab\rangle\}$

$A(G_2)=\{\hat{\pi}_1 \cup \hat{\pi}_2\hat{\pi}_4\}^*\hat{\pi}_3$, $K(G_2)=\{\langle a, b\rangle \cup \langle \varepsilon, \varepsilon, b\rangle\langle a\rangle\}^*\langle ab\rangle$

$L(G_1)=L(G_2)=\{a^n b^n \mid n\geq 1\}$.

Thus G_1 is a spanning subgrammar of G_2.

A grammar G is said to be *inherently ambiguous* if all grammars generating the same language are ambiguous. A grammar G is said to be *completely ambiguous* if any spanning subgrammar of G is ambiguous. A grammar G is said to be *structurally unambiguous* [1] if the restriction $f/A(G): A(G)\to K(G)$ is bijective. By definition it should be clear that any inherently ambiguous grammar is completely ambiguous.

Basic results are the following Theorems. Detailed proofs will appear elsewhere.

Theorem 2.1. *There exists a completely ambiguous grammar which is not inherently ambiguous.*

Theorem 2.2. *For any grammar G, there exists structurally unambiguous grammar G' such that $G \underset{s}{=} G'$.*

Theorem 2.3. *Let G_1, G_2 and G_3 be arbitrary grammars such that $G_1 \underset{s}{\subseteq} G_3$ and $G_2 \underset{s}{\subseteq} G_3$. Then it is unsolvable to determine whether $L(G_1) = L(G_2)$.*

Corollary. *Let G_1 be a subgrammar of G_2. Then it is unsolvable whether G_1 is a spanning subgrammar of G_2.*

Theorem 2.4. *Let G_1, G_2 and G_3 be grammars such that $G_1 \underset{s}{\subseteq} G_3$ and $G_2 \underset{s}{\subseteq} G_3$, and let G_3 be unambiguous. Then it is solvable to determine whether $L(G_1) = L(G_2)$.*

Theorem 2.5. *It is unsolvable to determine for an arbitrary grammar G where G is completely ambiguous.*

3. Graded context-free languages.

In this section we reduce consideration of a covering language to consideration of the language generated by a new type of grammar, called graded grammar.

By a *graded set* we mean a set Σ with a map $\sigma : \Sigma \to N = \{0, 1, 2, \cdots\}$. We denote by Σ_n the set $\sigma^{-1}(n)$. σ is called the *grading map* of Σ. For a in Σ, $\sigma(a)$ is called the *rank* of a. A finite graded set is called a *graded alphabet*. Thus, in a grammar $G = (N, \Sigma, P, S)$, P will be treated as a graded alphabet with the grading map σ_P.

Let Σ be any set. We denote by $[\Sigma^*]^n$ the set of all n-tuples of words over Σ, i.e., $[\Sigma^*]^n = \Sigma^* \times \cdots \times \Sigma^*$ (n-times). A subset Δ of $\bigcup_{i=1}^{\infty} [\Sigma^*]^i$ is called a *stencil set* over Σ if Δ is graded by the condition

$$\Delta_n \subset [\Sigma^*]^{n+1} \qquad \text{for all } n \geq 0.$$

A finite stencil set is called a *stencil alphabet.* We henceforth treat each element of Δ as an abstract symbol, and, in a grammar $G = (N, \Sigma, P, S)$, the set $f(P)$ will be treated as a stencil alphabet over Σ. Note that π and $f(\pi)$ have the same rank for each π in P.

Let Σ be a graded set. The set Σ^T of *trees* over Σ is defined by the following fundamental inductive definition.

(i) If a is in Σ_0, then a is in Σ^T

(ii) If $n > 0$, a in Σ_n and $\alpha_1, \cdots, \alpha_n$ in Σ^T, then

$$a\alpha_1 \cdots \alpha_n \qquad \text{is in } \Sigma^T.$$

A *graded grammar* is a grammar $G = (N, \Sigma, P, S)$ in which

(i) Σ is a graded alphabet

(ii) each production in P is of the form $X \to aY_1 \cdots Y_{\sigma(a)}$, where X and Y_i $(1 \leq i \leq \sigma(a))$ are in N, a is in Σ and $\sigma(a)$ is the rank of a.

A set L is a *graded context-free language* if $L = L(G)$ for some graded grammar G.

Theorem 3.1. *Let Δ be a stencil alphabet over Σ, and let $L\subset\Delta^*$. Then L is a graded context-free language if and only if $L=K(G)$ for some grammar G with the terminal alphabet Σ.*

Theorem 3.2. *For any grammar G, $A(G)$ is a graded context-free language.*

A *graded pushdown automaton* (abbreviated g-pda) is a pushdown automaton $M=(K, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ in which

i) Σ is a graded alphabet

ii) $\delta(p, a, Z)\subseteq K\times\Gamma^{\sigma(a)}$ for all (p, a, Z) in $K\times(Z\cup\{\varepsilon\})\times\Gamma$,

where $\sigma(a)$ is the rank of a for each a in Σ and $\sigma(\varepsilon)=1$.

For each g-pda M we define $T(M)$, the language *accepted by empty store*, to be

$$T(M)=\{w \text{ in } \Sigma^* \mid (q_0, w, Z_0)\vdash^*(q, \varepsilon, \varepsilon),\ q \text{ in } F\}.$$

Theorem 3.3. *L is a graded context-free language if and only if $L=T(M)$ for some g-pda M.*

Theorem 3.4. *Let M_1 be a g-pda. Then there exists a deterministic ε-free g-pda M_2 with $T(M_1)=T(M_2)$.*

Corollary 1. *Let Δ be a stencil alphabet. Let $L\subset\Delta^*$ be a covering language and let $R\subset\Delta^*$ be a regular set. Then*

(i) *$L\subset\Delta^T$*

(ii) *Δ^T-L is a covering language*

(iii) *L is a deterministic context-free language*

(iv) *Δ^*-L is a deterministic context-free language*

(v) *$L\cap R$ is a covering language.*

Corollary 2. *The family of covering language is closed under union, intersection and relative complementation.*

Let Σ_1 and Σ_2 be graded alphabets with grading map σ_1 and σ_2, respectively. A length preserving homomorphism $h: \Sigma_1^*\to\Sigma_2^*$ is said to be a *projection* if $\sigma_1(a)=\sigma_2(h(a))$ for all a in Σ_1.

Corollary 3. *The family of covering languages is closed under projections.*

Acknowledgements. The author wishes to express his gratitude to Professor Satoru Takasu for his advice. The author is also indebted to Professor Shigeru Igarashi and Mr. Teruyasu Nishizawa for their suggestions toward this paper.

References

[1] M. C. Paull and S. H. Unger: Structural equivalence of context-free grammars. JCSS, **2**, 427–463 (1968).

[2] S. Ginsburg and M. A. Harrison: Bracketed context-free languages. JCSS, **1**, 1–23 (1967).

[3] J. W. Thatcher: Characterizing derivation trees of context-free grammars through a generalization of finite automata theory. JCSS, **1**, 317–322 (1967).

[4] E. Altman and R. Banerji: Some problem of finite representability. Information and Control, **8**, 251–263 (1965).

[5] T. Kasai: An hierarchy between context-free and context-sensitive languages. JCSS, **5**, 492–508 (1970).