Information Processing in Japan Vol. 12, 1972

On Existence of Precedence Functions of Precedence Grammars

KIYOSHI ASAI*

Following results for precedence grammars are given ;
(1) the necessary and sufficient condition that a precedence grammar has precedence functions,
(2) the existence of equivalent precedence grammars with precedence functions to any given precedence grammar,
(3) the existence of equivalent precedence grammars with precedence functions to any given context-free grammar.

1. *Introduction*

The operator precedence grammar presented by R. W. Floyd [1] has been extended to more general precedence grammars by Wirth and Weber [2], A. Colmerauer [3] and others. At the same period M. Nagao [4] has independently obtained the idea of precedence grammar using boolean matrices. A. Colmerauer has also used same types of boolean matrices as M. Nagao's to extend the concept of the precedence grammar. Thus the class of precedence grammars tends to become wider. Practical applications of precedence grammars, however, meet with a difficulty since sizes of precedence matrices used in the analyses of the grammars become larger. Two methods have been proposed to remove the difficulty. The one is reduction of sizes of precedence matrices (Inoue [5] and others.) The other is the use of precedence functions (Floyd [1].)

Precedence functions, however, have not been considered as tools for analyses of the grammars since precedence grammars generally have no precedence function. They are very useful in analysis and construction of a compiler. Fortunately there exist equivalent precedence grammars with precedence functions for a given precedence grammar and the grammars are natural extensions of the given grammar. The author uses the fact in construction of a compiler of a modified PL360 language [6], [10].

2. *Basic definitions and Concepts*

In this section we give definitions and theorems of simple precedence grammars. Notations and theorems shown in this section are due to A. Colmerauer [3].

This paper first appeared in Japanese in Joho-Shori (Journal of the Information Processing Society of Japan), Vol. 13, No. 4 (1972), pp. 218~224.
* Japan Atomic Energy Research Institute

2.1 *Pair Relations*

We denote a subset of pair relations over a set E by ρ. $a\rho b$ is an abbreviation of a pair relation $(a, b)\in\rho$, where $a, b\in E$. The set of pair relations is denoted by $E\times E$ and complement of ρ is denoted by $\bar{\rho}=E\times E-\rho$. The product $\rho\sigma$ of relations ρ, σ is defined as

$$a\rho\sigma b\equiv[\text{there exist } c\in E, a\rho c\cap c\sigma b].$$

The closure ρ^+ of ρ is defined as $\rho^+=\bigcup_{i=1}^{\infty}\rho^i$, where $\rho^i=\rho\rho^{i-1}$, $a\rho^0 b\equiv[a=b]$. If the number of elements of the set E is n, $\rho^+=\bigcup_{i=1}^{\infty}\rho^i=\bigcup_{i=1}^{n}\rho^i$.

2.2 *Context-free Grammars*

A grammar $G=(V_N, V_T, S, P)$ is a context-free grammar if it satisfies the following conditions;

(1) V_N, V_T and S are finite sets such that $V_N\cap V_T=\phi, S\in V_N$,

(2) P is a set of finite pair relations $\rightarrow$ over the set $(V_N\cup V_T)^*$ of finite sequences of elements of $(V_N\cup V_T)$, including the empty sequence,

(3) if $x\rightarrow y$ for $x, y\in(V_N\cup V_T)^*$ and the length of x is unity, then x is an element of V_N,

(4) if $x\rightarrow y$ for $x, y\in(V_N\cup V_T)^*$, $Z\in V_N$, $x=uZv$, $y=uwv$, then there exist $u, v, w \in(V_N\cup V_T)^*$ such that $Z\rightarrow w$.

The sets V_N, V_T and P are called the set of variables, terminal symbols and rewriting rules, respectively. $x\Rightarrow y_n$ is an abbreviation of relations $x\rightarrow y_1\rightarrow\cdots\rightarrow y_n$, where $x\in V_N$ and $y_1, \cdots, y_n\in(V_N\cup V_T)^*$. The language generated by a context-free grammar G is denoted as $L(G)=\{t\in V_T^*\,|\,S\Rightarrow^+ t\}$.

2.3 *Precedence Grammars*

The pair relations α, λ and ρ between $A, B\in(V_N\cup V_T)$ of a context-free grammar $G=(V_N, V_T, S, P)$ are defined as follows;

$$A\alpha B\equiv[\text{there exists } U\in V_N,\ x, y\in(V_N\cup V_T)^*,\ U\rightarrow xABy],$$

$$A\lambda B\equiv[\text{there exists } y\in(V_N\cup V_T)^*,\ A\rightarrow By],$$

$$A\rho B\equiv[\text{there exists } x\in(V_N\cup V_T)^*,\ B\rightarrow xA].$$

New pair relations over $V_N\cup V_T$

$$\doteq\,=\alpha,\ \lessdot\,=\alpha\lambda^+ \text{ and } \gtrdot\,=\rho^+\alpha\cup\rho^+\alpha\lambda^+$$

are called the Wirth-Weber type simple precedence relations [2], [3]. A context-free grammar G is said to be a simple precedence grammar if for any $A, B \in(V_N\cup V_T)$, there exists one of pair relations $\gtrdot, \lessdot, \doteq$ and ϕ (empty relation). Hereafter we use the term precedence grammars for only simple precedence grammar. A simple precedence grammar is said to be unambiguous simple precedence grammar if it satisfies the following two conditions;

(1) if $X\rightarrow u, Y\rightarrow u$, then $X=Y$,

(2) $(\lessdot\cap\gtrdot)\cup(\lessdot\cap\doteq)\cup(\doteq\cap\gtrdot)=\phi$.

A. Colmerauer has shown that (a) the condition (2) is equivalent to the following cndition (2′), and that (b) G is a precedence grammar if its precedence relations $\gtrdot$, $\lessdot$ and $\doteq$ satisfy the following condition (3);

(2′) $(\alpha\lambda^+ \cap \alpha) \cup (\rho^+\alpha \cap \alpha) \cup (\rho^+\alpha\lambda^+ \cap \alpha) \cup (\rho^+\alpha \cap \alpha\lambda^+) = \phi$,

(3) $\alpha \subset \doteq$, $\alpha\lambda^+ \subset \lessdot$, $\rho^+\alpha\lambda \subset \gtrdot$, $\rho^+\alpha\lambda^+ \subset \lessdot \cup \gtrdot$.

3. *Existence Theorem of Precedence Functions*

In this section let us show the existence of precedence functions for any given precedence grammar.

Definition 1. Precedence matrix

A matrix M is called a precedence matrix if its (i, j) element is the pair relation (S_i, S_j) of a precedence grammar $G=(V_N, V_T, S, P)$, $S_i, S_j \in (V_N \cup V_T)$.

Example 1.

For simple precedence grammar $G=(V_N, V_T, A, P)$, $V_N=\{A, B, C\}$, $V_T=\{[,], \lambda\}$, $P=\{\varphi_1, \cdots, \varphi_6\}$, $\varphi_1: A \to CB$, $\varphi_2: A \to [\ ]$, $\varphi_3: B \to \lambda$, $\varphi_4: B \to \lambda A$, $\varphi_5: B \to A$, $\varphi_6: C \to [$, assuming $\gtrdot \supset \rho^+\alpha\lambda^+$, we can get its precedence matrix as Fig. 3. 1.

	A	B	C	]	[	λ
A				$\gtrdot$		
B				$\doteq$		
C	$\lessdot$	$\doteq$	$\lessdot$		$\lessdot$	$\lessdot$
]				$\gtrdot$		
[	$\gtrdot$	$\gtrdot$	$\gtrdot$	$\doteq$	$\gtrdot$	$\gtrdot$
λ	$\doteq$		$\lessdot$	$\gtrdot$	$\lessdot$	

Fig. 3. 1

As is shown in the above example, the matrix M is composed of $N \times N$ elements when the number of elements of set $(V_N \cup V_T)$ is N. The values of N are not small for practical programming languages, for examle, the value of N is approximately 500 for Fortran IV language, but we can compress this matrix to two vectors f and g with N elements, respectively. These two vectors are called precedence functions.

Definition 2. Precedence functions

Two functions f and g with N values respectively are called the precedence functions of a precedence grammar $G=(V_N, V_T, S, P)$ if they satisfy the following relations for any $S_i, S_j \in (V_N \cup V_T)$;

if $S_i \lessdot S_j$ then $f(S_i) < g(S_j)$,

if $S_i \doteq S_j$ then $f(S_i) = g(S_j)$,

if $S_i \gtrdot S_j$ then $f(S_i) > g(S_j)$, where N is the number of elements of $(V_N \cup V_T)$

Definition 3. Symbols f_i and g_j.

f_i and g_j are symbols which have one-to-one correspondence with function values $f(S_i)$ and $g(S_j)$, respectively. These symbols have precedence $f_i \lessdot g_j$ if $S_i \lessdot S_j$, $f_i \doteq g_j$ if $S_i \doteq S_j$, or $f_i \gtrdot g_j$ if $S_i \gtrdot S_j$.

We sometimes denote $f_i \leq g_j$ if f_i and g_j have precedence $f_i \lessdot g_j$ or $f_i \doteq g_j$.

Definition 4. Sets $H_1, \cdots, H_n$

We denote by H_1 a set $\{f_1, \cdots, f_n, g_1\}$ and by H_n a set $\{f_1, \cdots, f_n, g_1, \cdots, g_n\}$.

Definition 5. Cycle, Monotone cycle and set $B(h)$

If there exists a sequence $h_1R_1h_2R_2\cdots h_mR_mh_1$ for nonempty relation $R_i \in \{\gtrdot, \lessdot, \doteq\}$ and $H_n \ni h_j$, $j=1, \cdots, m$, we call the sequence as a cycle of h_1. This cycle is said to be a monotone cycle if a transitive relation $h_1 \lessdot h_1$ holds for the cycle. In this case we say that the set H_n contains a monotone cycle. We also denote $h \in B(h)$ if there exists a monotone cycle of h and $h \notin B(h)$ otherwise.

Theorem 1.

A precedence grammar G has precedence functions if and only if every element h in the set H of the definition 4 for the grammar G has no monotone cycle.

For the proof see [8].

The precedence grammar G_1 of the example 1 has no precedence functions since it includes a monotone cycle $f(\lambda) < g([) < f([) = g(]) < f(\lambda)$ if we assume $\gtrdot \supset \rho^+ \alpha \lambda^+$.

By the theorem 1 a precedence grammar G has no precedence function if the set $H = \{f_i, g_j | i, j=1, \cdots, n\}$ of a precedence grammar contains monotone cycles. We can, however, obtain a new set H' which has no monotone cycle by transforming H to $H' = \{f_i, g_j | i, j=1, \cdots, m\}$, $m \geq n$, $H' \supseteq H$ and G to G'. Let us note that this transformation preserves a condition $L(G) = L(G')$. From this fact we have the following theorem.

Theorem 2.

For a given precedence grammar, there exist precedence grammars with precedence functions.

For the proof see [8].

Example 2.

Introducing a new variable A' and changing the rewriting rule φ_2: $A \to [\ ]$ to φ'_2: $A \to A']$, φ''_2: $A' \to [$ of the grammar G_1, we can obtain a grammar G_2 with precedence functions. As is shown by A. Learner and A. L. Lim [9], there exist precedence grammars for a given context-free grammar, we have the following corollary as a direct consequence of the theorem 2.

Corollary

There exist precedence grammars with precedence functions for a given context-free grammar.

Recent publication shows [11] that David F. Martin, although the method is different, has independently obtained the same results as the author's.

4. *Concluding Remarks*

The author had not yet obtained these theorems when he began to construct the GPL compiler [6], [10]. At that time the author adopted the following method [7];

(i) there exists a family of precedence grammars with precedence functions,

(ii) by introducing new variables and terminals we can transform any precedence grammar to a grammar of the family.

The above method has, however, some defficulties since the introduction of new terminals enforces the scanning routine of the input symbol to scan symbols context sensitively.

If a grammar has precedence functions depends on frequencies of appearances of terminals in the right parts of rewriting rules [7], so that it is easy to obtain precedence grammars with precedence functions since numbers of such troublesome terminals are usually a few.

We can use the same procedures to obtain precedence grammars from a context-free grammar and to obtain precedence grammars with precedence functions from a precedence grammar.

This fact assures us that we can obtain an equivalent precedence grammar with precedence functions immediately when we have obtained a precedence grammar from a context-free grammar, and that we may take the term "equivalent grammars" as "equivalent grammars with the same analysis mechanism."

References

[1] R. W. Floyd : "Syntactic analysis and operator precedence", *J. ACM*, Vol. 10. No. 3, pp. 316–333 (July 1963).

[2] N. Wirth & H. Weber : "Euler-A generalization of ALGOL and its formal definition : part I", *Comm. ACM*, Vol. 9, No. 1, PP. 13–23, 25 (Jan. 1966).

[3] A. Colmerauer : "Total precedence relations", *J. ACM*, Vol. 17, No. 1, pp. 14–30 (Jan. 1970).

[4] M. Nagao : "Syntactic analysis for phrase structure grammar", *Joho-Shori* (*Journal of the Information Processing Society of Japan*), Vol. 4, No. 4, pp. 186–193 (July 1963).

[5] K. Inoue : "Right precedence grammars", *Joho-Shori*, Vol. 11, No. 8, pp. 449–456 (Aug. 1970).

[6] K. Asai, Y. Inami and T. Fujimura : "The grammar and compiler of a modified PL360 language", Reports on 12th programming symposium of the Infor. Proc. Soc. of Japan, pp. 197–207 (Jan. 1971).

[7] K. Asai : "Precedence grammars with precedence functions", *Joho-Shori*, Vol. 12, No. 5, pp. 264–271 (May 1971).

[8] K. Asai : "On existence of precedence functions of precedence grammars", *Joho-Shori*, Vol. 13, No. 4, pp. 218–224 (April 1972).

[9] A. Learner & A. L. Lim : "A note on transforming context-free grammars to Wirth-Weber precedence form", *The Computer Jour.*, Vol. 13, No. 2, pp. 142–144 (May 1970).

[10] K. Asai & M. Tomiyama : "GPL-Genken Programming Language", JAERI M-Report 4762, Japan Atomic Energy Research Institute (Febr. 1972).
[11] D. F. Martin : "A Boolean matrix method for the computation of linear precedence functions", *Comm. ACM*, Vol. 15, No. 6, pp. 448–454 (June 1972).