順位関数の存在について*

浅井 清**

Abstract

In this paper the author shows following two results for precedence grammars;

(1) the necessary and sufficient condition that precedence grammars have precedence functions,

(2) the existence of equivalent precedence grammars with precedence functions to any given precedence grammar.

1. はじめに

R. W. Floyd[1] によって考案された演算子文法は，Wirth-Weber[2]，A. Colmerauer[3] などによってより一般の順位文法へと発展させられた．R. W. Floyd とは独立に，同じころ長尾[4] もプログラムの流れの解析に使われるブール行列を利用して順位文法の考えを得ている． A. Colmerauer による一般化の基本的な方法も，やはりこのブール行列の利用にある．このようにして順位文法のわくは広くなってきているが，その現実的な応用については問題が残っている．順位文法の解析は順位行列を使って行なわれるが，実用的なプログラミング言語を順位文法で規定したとき，この順位行列は非常に大きなものになるからである．この難点を取り除くために，いままで二つの方法が考えられている．その一つは順位行列を小さくする方法で，これは井上[5] などによって考えられている．他の一つは順位行列のかわりに順位関数を使う方法である[1),2)]．順位関数は Floyd によって考えられたものであるが，一般に順位文法は順位関数をもたないので，この方法が文法解析の手段として正面から取り上げられたことはなかった．順位関数は，それが存在すれば実用的なコンパイラの作成には非常に便利なものである．幸いなことに，与えられた順位文法について，これと同値な，しかも与えられた文法の自然な拡張となっている順位文法で順位関数をもつものが存在する．この事実を使って筆者は変形 PL360 言語[6),10]のコンパイラを順位関数で解析している．

* On Existence of Precedence Functions of Precedence Grammars, by Kiyoshi Asai (Japan Atomic Energy Research Institute).

** 日本原子力研究所

2. 順位文法

まず，順位文法の定義と定理について述べよう．2.1 から 2.3 までの記法と定理は A. Colmerauer[3] による．

2.1 対関係

集合 E の上の対関係の部分集合を ρ で表わし，$a, b\in E$，$(a, b)\in\rho$ を $a\rho b$ と略記する．対関係の集合を $E\times E$ で，これについての ρ の補集合を $\bar{\rho}=E\times E-\rho$ で表わす．E についての二つの対関係 ρ，σ の積 $\rho\sigma$ は，次のように定義される：

$$a\rho\sigma b\equiv[\exists c\in E,\ a\rho c\cap c\sigma b].$$

ρ の閉包 ρ^+ は，

$$\rho^i=\rho\rho^{i-1},\quad a\rho^0 b\equiv[a=b]$$

として，

$$\rho^+=\bigcup_{i=1}^{\infty}\rho^i$$

と表わされるが，E の要素の数が $n<\infty$ であるときは，

$$\rho^+=\bigcup_{i=1}^{\infty}\rho^i=\bigcup_{i=1}^{n}\rho^i$$

となる．

2.2 *CF* 文法

文法 $G=(V_N, V_T, S, P)$ が CF 文法であるとは，$V_N\cap V_T=\phi$，$S\in V_N$ となる有限集合 V_N，V_T と，$V_N\cup V_T$ の要素の有限列（長さ 0 の列も含む）の集合 $(V_N\cup V_T)^*$ の上で定義された有限個の対関係→の集合 P が存在し，長さ 1 の x と，$y\in(V_N\cup V_T)^*$ なる y について $x\to y\in P$ であるときは $x\in V_N$ であり，また $x, y\in(V_N\cup V_T)^*$，$Z\in V_N$，$x=uZv$，$y=uwv$ なる x，y，Z について $x\to y$ ならば，$Z\to$

$w \in P$ となる $u, v, w \in (V_N \cup V_T)^*$ が存在するときをいう．ここで V_N, V_T, P をそれぞれ変数，端記号，生成規則の集合と呼び，$x \in V_N$, $y_1, \cdots, y_n \in (V_N \cup V_T)^*$ なる x と $y_1, \cdots, y_n$ について $x \to y_1 \cdots \to y_n$ となるときは $x \Rightarrow y_n$ と略記する．このとき文法 G によって生成される言語 $L(G)$ は，

$$L(G)=\{t \in V_T^* \mid S \Rightarrow^+ t\}$$

と表現される．

2.3 順位文法

CF 文法 $G=(V_N, V_T, S, P)$ の任意の要素 $A, B \in (V_N \cup V_T)$ について，対関係 $\gtrdot$, $\lessdot$, $\doteq$, または ϕ（空なる対関係）のいずれか一つが存在するとき，文法 G は単純順位文法であるという．これ以後の議論においてはすべて単純順位文法を取り扱い，それを順位文法と略記する．次に CF 文法 $G=(V_N, V_T, S, P)$ の任意の $A, B \in (V_N \cup V_T)$ について対関係 α, λ, ρ を次のように定義する．

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

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

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

この α, λ, ρ を組み合わせて得られる $(V_N \cup V_T)$ 上の新しい対関係，

$$\doteq\ =\alpha,\quad \lessdot\ =\alpha\lambda^+,\quad \gtrdot\ =\rho^+\alpha \cup \rho^+\alpha\lambda^+$$

を Wirth-Weber 形の単純順位関係と呼ぶ[2),3)]．文法 G についてこの順位関係が定義されているとき，G があいまいさのない単純順位文法であるとは，

（1） $X \to u$, $Y \to u$ なら $X=Y$,

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

なる条件を満たすときをいう．ここで ϕ は空集合を表わす．A. Colmerauer は上の条件（2）が次の条件（2′）と同値であること，および文法 G の順位関係 $\lessdot$, $\doteq$, $\gtrdot$ が条件（3）を満たせば，その文法は順位文法として解析できることを示した[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） $\alpha \subset \doteq, \alpha\lambda^+ \subset \lessdot, \rho^+\alpha \subset \gtrdot, \rho^+\alpha\lambda^+ \subset \lessdot \cup \gtrdot$.

記法を簡単にするために，以後の議論では ρ^+, λ^+ をそれぞれ ρ, λ と略記する．

3. 順位関数をもつ順位文法

この節では，順位文法が順位関数をもつための必要十分条件，および任意の順位文法に対して順位関数をもつ同値な順位文法の存在を示そう．

集合 $(V_N \cup V_T)$ の上の新しい対関係 $R=\{\lessdot, \doteq, \gtrdot, \phi\}$ のすべてを行列で表現することができる．

定義 0. 順位行列

$(V_N \cup V_T) \ni S_i, S_j$ の二つの文法要素の間に $R=\{\lessdot, \doteq, \gtrdot, \phi\}$ のいずれか一つの関係が存在するとき，この関係を第 i 行，第 j 列の要素とする行列を順位行列と呼ぶ．

順位文法 G_1[2)] の順位行列は次のようになる．

例 1

$G_1=(V_N, V_T, A, P)$,

$V_N=\{A, B, C\}$, $V_T=\{〔, 〕, \lambda\}$,

$P=\{\varphi_1, \cdots, \varphi_6\}$,

$\varphi_1: A \to CB〕$, $\varphi_2: A \to 〔\ 〕$, $\varphi_3: B \to \lambda$,

$\varphi_4: B \to \lambda A$, $\varphi_5: B \to A$, $\varphi_6: C \to 〔$

において，$\gtrdot \supset \rho\alpha\lambda$ とすれば，その順位行列は次のようになる．

	A	B	C	〕	〔	λ
A				$\gtrdot$		
B				$\doteq$		
C	$\lessdot$	$\doteq$	$\lessdot$		$\lessdot$	$\lessdot$
〕				$\gtrdot$		
〔	$\gtrdot$	$\gtrdot$	$\gtrdot$	$\doteq$	$\gtrdot$	$\gtrdot$
λ	$\doteq$		$\lessdot$	$\gtrdot$	$\lessdot$	

図 3.1 Precedence matrix for G_1

この例でわかるように，集合 $(V_N \cup V_T)$ の要素の数を N とすると順位行列は $N \times N$ 行列となる．実用的なプログラミング言語，たとえば FORTRAN IV などの文法を厳密に記述すると N は 500 を越える．この行列を $2N$ 個の関数値に縮約できる場合がある．いま任意の文法要素 $S_i \in (V_N \cup V_T)$ を変数とする二つの実数値関数 $f(S_i)$, $g(S_i)$ を考えよう．これは，いいかえれば一つの文法要素 S_i に二つの値 $f(S_i)$, $g(S_i)$ を対応させることにほかならない．このように一つの文法要素に二つの値を対応させ，それによって順位関係 $R=\{\lessdot, \doteq, \gtrdot\}$ を論じなければならない理由は，さきにあげた例1の文法 G_1 の順位行列をみれば明らかになる．ここでは $S_i \gtrdot S_i$ となる場合が存在し，一つの値 $f(S_i)$ では，この関係を表現できない．それでは二つの値で十分かといえば，それも必らずしもあたらない．それは例1の順位行列からわかる

ように $f(\lambda)<g(()<f(()=g())<f(\lambda)$ となって，文法 G_1 に対する $f(S_i)$, $g(S_i)$ が存在しないことから明らかである．しかしこのような f, g をもつことができない任意の文法 G に対して，これと同値な文法で f, g をもつことができるものが必ず存在する．それをこの節で示そう．

定義 1.　順位関数

単純順位文法 G の任意の $S_i, S_j \in (V_N \cup V_T)$ について，

$S_i \lessdot S_j$ なら $f(S_i)<g(S_j)$,

$S_i \doteq S_j$ なら $f(S_i)=g(S_j)$,

$S_i \gtrdot S_j$ なら $f(S_i)>g(S_j)$

となるそれぞれが N 個の値をとる関数 f と g が存在するとき，この f と g を文法 G の順位関数と呼ぶ．ここで N は集合 $V=(V_N \cup V_T)$ の要素の数である．

定義 2.　f_i, g_j

f_i, g_j は関数値 $f(S_i)$, $g(S_j)$ と1対1に対応する記号で，これらの記号は $S_i \lessdot S_j$ のとき $f_i \lessdot g_j$, $S_i \doteq S_j$ のとき $f_i \doteq g_j$, $S_i \gtrdot S_j$ のとき $f_i \gtrdot g_j$ となる順序をもつ．

これら f_i, g_j について $f_i \doteq g_j$, または $f_i \lessdot g_j$ が成立するとき，このことを $f_i \leq g_j$ と略記することがある．

定義 3.　$H_1, \cdots, H_n$

集合 $\{g_1, \cdots, g_N, f_1\}$ を H_1, $\{g_1, \cdots, g_N, f_1, \cdots, f_n\}$ を H_n で表わす．

定義 4.　サイクル，単調サイクル，$B(g_j)$

空ではない関係 $R_i \in \{\gtrdot, \lessdot, \doteq\}$ が存在して $H_n \ni h_j$, $j=1, \cdots, l$ なる h_j について $h_1R_1h_2R_2 \cdots h_l R_l h_1$ となるとき，この列 $h_1R_1h_2R_2 \cdots h_lR_lh_1$ を h_1 のサイクルと呼び，この列で $h_1 \lessdot h_1$ となるときこのサイクルを単調サイクルと呼ぶ．このとき H_n は単調サイクルを含むという．

h の単調サイクルが存在するとき，$h \in B(h)$ と書き，そうでないときは $h \notin B(h)$ と書く．

順位文法について次の定理が成立する．

定理 1

順位文法が順位関数をもつための必要十分条件は，定義3によって作られる集合 H の記号 h が単調サイクルをもたないことである．

この条件の成立は，当然のようにみえるが自明ではない．そこで付録に証明を掲げておく．例1の順位文法 G_1 は $\gtrdot ⊐ \rho\alpha\lambda$ とすれば，単調サイクル $f(\lambda)<g(()<f(()=g())<f(\lambda)$ をもつから順位関数を定義できない．

集合 $H=\{f_i, g_j|i, j=1, \cdots, n\}$ に含まれる任意の記号 h が単調サイクルをもたないときは，定理1によってこの集合 H に対応する順位文法は順位関数をもつことがわかった．順位関数をもたない順位文法 G に対応する集合 $H=\{f_i, g_j|i, j=1, \cdots, n\}$ は単調サイクルをもつ記号を含んでいる．ところが文法 G を適当に変形して G' とし，H を $H'=\{f_i, g_j|i, j=1, \cdots, m\}$, $m \geq n$, $H' \supseteq H$ と変えることによって H' に含まれる記号が単調サイクルをもたないようにできる．しかも，この変形は $L(G)=L(G')$ という制約を満たす．この事実から次の定理が得られる．

定理 2

任意の順位文法について，これと同値な順位関数をもつ順位文法が存在する．

証明は付録に掲げるが，その要点を簡単に述べると次のようになる．

（1）順位文法 G に対応する集合 $H=\{f_i, g_j|i, j=1, \cdots, n\}$ が単調サイクル，

$f_i \lessdot g_j \lessdot f_k \lessdot \cdots \lessdot f_i$

を含んでいるとしよう．

（2）文法 G に新しい変数と生成規則を導入することによって，このサイクルを消去することができる．その方法は次のようになる．

（i）$f_i \lessdot g_j$ の関係が $f_i\alpha\lambda g_j$ なら，新しい変数と生成規則を導入することによって，$f_i\alpha\lambda g_j$ を $f_i\rho\alpha\lambda g_j$ かつ $f_i \gtrdot g_j$ とすることができる．

（ii）$f_i \gtrdot g_j$ の関係が $f_i\rho\alpha\lambda g_j$ なら，これを $f_i \lessdot g_j$ とすることができる．

（iii）$g_j \lessdot f_k$ の関係が $f_k\rho\alpha g_j$ なら，新しい変数と生成規則を導入することによって，$f_k\rho\alpha g_j$ を $f_k\rho\alpha\lambda g_j$, かつ $f_k \lessdot g_j$ とすることができる．

（iv）$g_j \lessdot f_k$ の関係が $f_k\rho\alpha\lambda g_j$ なら，これを $f_k \lessdot g_j$ とすることができる．

（3）上の（2）で述べた方法によって単調サイクルを消去することができるが，ここに一つ問題がある．一つの単調サイクルの消去が新しい単調サイクルを生むことがあるから．次の二つのサイクル C_1 と C_2 において，$f_1 \lessdot g_1$ を $f_1 \gtrdot g_1$ とすることによって単調サイクル C_2 を作ることになる．

C_1: $f_1 \lessdot g_1 \lessdot f_2 \lessdot g_2 \lessdot f_1$,

C_2: $f_1 < g_1 > f_3 > g_3 > f_1$.

しかし，ここではひとまず C_1 が C_1'，C_2 が C_2' となるように文法を変形し，

C_1': $f_1 > g_1 < f_2 < g_2 < f_1$,

C_2': $f_1 > g_1 > f_3 > g_3 > f_1$

とする．次に $g_1 > f_3$ を $g_1 < f_3$ となるように文法を変形し，新しく単調サイクルができれば同様の操作を行なう．これを続けていくと最後には g_1 が $f_i R g_1$（R は順位関係を表わす）となる任意の f_i について $g_1 \leq f_i$ となり，操作は有限回で終わる．

(4) g_1 を含む単調サイクルはこのようにして消去された．同様の操作で $g_2, \cdots, g_n$ を含む単調サイクルを消去する．この操作で文法の変数と生成規則の数は増加するから，集合 H は $H' = \{f_i, g_j | i, j = 1, \cdots, n, n+1, \cdots, m\}$ と変化するが，$H \ni h$ の単調サイクルを消去すれば，$H' \ni h$ の単調サイクルも同時に消去することができる．したがって H' に対応する順位文法は順位関数をもつ．

例 2

先に述べた文法 G_1 に新しい変数 A' を導入し，生成規則 φ_2: $A \to 〔〕$ を φ_2': $A \to A'〕$, φ_2'': $A' \to 〔$ と変更して得られる文法 G_2 は，次のように順位関数をもつ．

S =	A	A'	B	C	〕	〔	λ
$f(S)$=	2	1	1	1	2	4	2
$g(S)$=	2	3	1	3	1	3	2

4. おわりに

筆者は先に述べたコンパイラ[6),10)]の作成にとりかかったときには，これらの定理を得ていなかった．そこで常に順位関数をもつ順位文法の族を求め[7)]，その文法族を一つの近似として順位関数をもつ順位文法を得るという方法を採用していたが，この方法にはいくぶんか難点がある．入力記号を走査するルーチンによけいな負担を負わせることがあり，また文法を修正するときは，コンパイラ設計者が勘をはたらかせて順位関数をもつように文法を修正しなければならないからである．

定理 1 の証明の手順を計算機プログラムにすることができるが，そうすれば単調サイクルを機械的に拾い上げることができる．定理 2 の証明の手順も計算機プログラムにすることができるが，この部分はコンパイラ設計者が手作業で行なって文法を修正するほうがよい．同一の右辺をもつ生成規則の出現を避けうることが多いからである．

Wirth-Weber 形式の順位文法については，$\gtrdot \supset \rho\alpha\lambda$ という制約があるため，一般には定理 2 が成立しない．A. Colemerauer の全順位文法になると，条件が $(\gtrdot \cup \lessdot) \supset \rho\alpha\lambda$ とゆるやかになるので定理 2 が得られる．定理 1 の証明は順位関数を得る手順であるが，Wirth のプログラム[8)]を FORTRAN に書き替えて $V_N \cup V_T$ の要素の数が 150，順位関係の数が 1900 の文法の順位関数を求めるに要する FACOM 230-60 の中央処理装置時間は 5～6 秒である．ある順位文法が順位関数をもつかどうかは，その文法の生成規則の右辺に出現する端記号の出現頻度に依存する[7)]．幸い現在のプログラミング言語ではこのような端記号の数は少なく，また利用できる文字セットの数が増加する傾向にあるから将来も事情は変わらないであろう．したがって順位関数をもつ順位文法を得ることは容易である．

CF 文法から順位文法を求める手順[9)]と，順位文法からこれと同値な順位文法を求める手順とは同じである．このことには二つの意味がある．第 1 はコンパイラ設計者が CF 文法から順位文法を得たときには，設計者は順位関数をもつ順位文法を手に入れうる状態にほぼ到達しているということ，第 2 には，「同値な」ということばを「解析機構まで同値な」という強い意味に解釈してよいということである．

5. 付　録

定理 1，2 の証明を与えよう．

定義 5. f_i **線**, $g(S_j, f_i)$, $L(f_i)$, $U(f_i)$

順位行列の第 i 行 K_i を取り出し，この行の記号 g_j と f_i に整数値 $g(S_j)$, $f(S_i)$ を対応させる．第 1 列から始めて $f_i < g_j$ となるときは $f(S_i) < g(S_j)$, $f_i \doteq g_j$ となるときは $f(S_i) = g(S_j)$, $f_i > g_j$ となるときは $f(S_i) > g(S_j)$ となるように $f(S_i)$ と $g(S_j)$ の値を定める．このようにして得られた集合 $\{f_i, f(S_i), g_1, g(S_1), \cdots, g_N, g(S_N)\}$ と $f_i \leftrightarrow f(S_i), g_j \leftrightarrow g(S_j)$ の対応づけを f_i 線と呼ぶ．

f_i 線上の $g(S_j)$ を $g(S_j, f_i)$ で表わすことがある．また集合 $L(f_i)$, $U(f_i)$ をそれぞれ，

$L(f_i) = \{g_j | f_i > g_j$, または $f_i \doteq g_j, g_j \in K_i\}$,

$U(f_i) = \{g_j | f_i < g_j, g_j \in K_i\}$ で定義する．

定義 6.　右へのスライド

H_N の部分集合 A をある関数値 $h(S_i)$ を基準にして右へスライドするとは，$h_i<h_j$, $(h_j \in A)$ となる h_j に対応する順位関数の値を $h(S_i)+1$ とし，これにつれて $h_j \doteq h_k$, または $h_j<h_k$ となる $h_k\,(\in A)$ に対応する関数値を $h(S_j)$, または $h(S_j)+1$ とおく操作を A のすべての要素について行なうことをいう．

定理1の証明

条件が必要であることは明らかであるから，十分であることを示そう．

証明は次のような手順で行なう．

(1)　$h \notin B(h)$, $h \in H_2$ なる条件が成立すれば，f_1 線と f_2 線を合わせて新しく $f_1 \cdot f_2$ 線を作れることを示し，

(2)　$n=N-1$ のとき $h \notin B(h)$, $h \in H_{N-1}$ なる条件のもとで新しい線分 $f_1 \cdots f_{N-1}$ が作られていると仮定し，

(3)　$n=N$ のとき $h \notin B(h)$, $h \in H_N$ の条件があれば $f_1 \cdots f_{N-1}$ 線と f_N 線を合わせて新しく一つの線分を作れることを示す．

ステップ 1

(1)　$L(f_1) \cap L(f_2) \ni g_i$ のとき

(i)　$g(S_i, f_1)=g(S_i, f_2)$ なら f_1 線，f_2 線上の $g(S_i)$ はそのままの値をもつとする．

(ii)　$g(S_i, f_1)>g(S_i, f_2)$ なら $g(S_i, f_2) \Leftarrow g(S_i, f_1)$ とし，f_2 線上の $U(f_2)$ を右にスライドする．このとき，

$U(f_2) \cap L(f_1) \ni g_j$,

かつ $L(f_2) \cap U(f_1) \ni g_k$

なら $g(S_i, f_2)$ の定義は $g(S_i, f_1) \to g(S_i, f_2) \to g(S_j, f_2) \to g(S_j, f_1) \to g(S_k, f_1) \to g(S_j, f_2) \to \cdots$ となる g の定義列を引き起こすが，これは g_j, g_k がそれぞれ $g_j \notin B(g_j)$, $g_k \notin B(g_k)$ という仮定に反する．したがって $U(f_2) \cap L(f_1) = \phi$, または $L(f_2) \cap U(f_1)=\phi$ であり，この場合には $g(S_i, f_2)$ の定義は，g_j の値の定義のループを引き起こさない．

(iii)　$g(S_i, f_1)<g(S_i, f_2)$ なら $g(S_i, f_1) \Leftarrow g(S_i, f_2)$ とし，f_1 線上の $U(f_1)$ を右にスライドする．このときも(ii)と同様に $g(S_i, f_1)$ の定義が g_j の値の定義のループを引き起こすことはない．

(2)　$L(f_1) \cap U(f_2) \ni g_i$ のとき

$L(f_1) \cup \{h \mid h<g_i,\ h \in H_2\}$ を $L(f_1)$,

$U(f_2) \cup \{h \mid g_i<h,\ h \in H_2\}$ を $U(f_2)$

として次の操作を行なう．

(i)　$g(S_i, f_1)=g(S_i, f_2)$ なら f_1, f_2 線上の g_i はそのままの値をもつとする．

(ii)　$g(S_i, f_1)>g(S_i, f_2)$ なら $g(S_i, f_2) \Leftarrow g(S_i, f_1)$ とおき，f_2 線上の g_i を右へスライドする．この $g(S_i, f_2)$ の定義は他の g_j の値の定義を変えない．

(ii)　$g(S_i, f_1)<g(S_i, f_2)$ なら $g(S_i, f_1) \Leftarrow g(S_i, f_2)$ とおき，f_1 線上の $U(f_1)$ を右にスライドする．このとき(1),(2)の議論からわかるように，$U(f_1) \cap L(f_2) \neq \phi$ とはならないから，この $g(S_i, f_1)$ の定義がループを引き起こすことはない．

(3)　$U(f_1) \cap L(f_2) \ni g_i$ のとき

上述の(2)と同じような議論によって g_i の値を定義するが，このとき g_i の値の定義が g_j の値の定義のループを引き起こすことはない．

(4)　$U(f_1) \cap U(f_2) \ni g_i$ のとき

(i)　$g(S_i, f_1)=g(S_i, f_2)$ なら f_1, f_2 線上の g_i は，そのままの値をもつとする．

(ii)　$g(S_i, f_1)>g(S_i, f_2)$ なら $g(S_i, f_2) \Leftarrow g(S_i, f_1)$ とし，f_2 線上の g_i を右にスライドする．このとき $L(f_1) \not\ni g_i$, $L(f_2) \not\ni g_i$ であるから，f_2 線上の g_i のみが動く．したがってこの g_i の値の定義はループしない．

(iii)　$g(S_i, f_1)<g(S_i, f_2)$ なら $g(S_i, f_1) \Leftarrow g(S_i, f_2)$ として $g(S_i, f_1)$ を定義する．このときも上記(4)(ii)と同じように $g(S_i, f_1)$ の定義はループしない．

(5)　積集合が空集合となるとき

上記の(1)～(5)のいずれの場合にもあてはまらないときは，f_1, f_2 線上の $g(S_i, f_1)$, $g(S_i, f_2)$ はそれぞれもとの値をもつとする．

定義 7.　$f_1 \cdot f_2$ 線，$L(f_1/f_1 \cdot f_2)$, $U(f_1/f_1 \cdot f_2)$, $g(S_i, f_1 \cdot f_2)$

上述の(1)～(5)の操作を S_i, $i=1, \cdots, n$ についてくり返すと，f_1 線と f_2 線上の同じ名まえ g_i の値 $g(S_i, f_1)$ と $g(S_i, f_2)$ は互いに等しくなっている．したがってこの f_1 線と f_2 線を合わせて一つの線分を作ることができる．これを $f_1 \cdot f_2$ 線と名づけ，この線分の上での g_i の値を $g(S_i, f_1 \cdot f_2)$ で表わし，また $L(f_1)$, $U(f_1)$ をそれぞれ $L(f_1/f_1 \cdot f_2)$, $U(f_1/f_1 \cdot f_2)$ で表わす．

ステップ 2

$f_1 \cdots f_{n-1}$ 線が作られたと仮定しよう．この線分と f_n 線を組み合わせて新しく $f_1 \cdots f_n$ 線を作ることができることを示せばよい．

(6) $L(f_n) \cap L(f_p/f_1 \cdots f_{n-1}) \ni g_i$, $(1 \leq p \leq n-1)$ のとき

$L(f_p/f_1 \cdots f_{n-1}) \cup \{h \mid h < g_i,\ h \in H_n\}$ を改めて $L(f_p/f_1 \cdots f_{n-1})$ と定義し，また $U(g_i/f_1 \cdots f_{n-1}) \cup \{h \mid g_i < h,\ h \in H_n\}$ を改めて $U(g_i/f_1 \cdots f_{n-1})$ と定義する．

(ⅰ) $g(S_i, f_n) = g(S_i, f_1 \cdots f_{n-1})$ なら $g(S_i, f_n)$, $g(S_i, f_1 \cdots f_{n-1})$ はそのままの値とする．

(ⅱ) $g(S_i, f_n) > g(S_i, f_1 \cdots f_{n-1})$ なら $g(S_i, f_1 \cdots f_{n-1}) = g(S_i, f_n)$ とし，この $g(S_i, f_1 \cdots f_{n-1})$ を基準にして $f_1 \cdots f_{n-1}$ 線上の $U(g_i/f_1 \cdots f_{n-1})$ を右にスライドする．$g_i \notin B(g_i)$ という仮定によって，$f_1 \cdots f_{n-1}$ 線上では $g_i \lhd g_i$ となっている．また，$g_j \in L(f_n)$, $g_k \in U(f_n)$, $g_j \in U(f_q/f_1 \cdots f_{n-1})$, $g_k \in L(f_q/f_1 \cdots f_{n-1})$ のときは $f_1 \cdots f_{n-1}$ 線上で $g_k \leq f_q < g_i$, $(1 \leq q \leq n-1)$, f_n 線上で $g_j \leq f_n < g_k$ となって g_j の値の定義がループするが，これは $g_i \notin B(g_j)$ という仮定に反する．

(ⅲ) $g(S_i, f_n) < g(S_i, f_1 \cdots f_{n-1})$ なら $g(S_i, f_n) = g(S_i, f_1 \cdots f_{n-1})$ とし，この $g(S_i, f_n)$ を基準にして $f_n \cup U(f_n)$ を右にスライドする．このとき $g_k \in (U(f_n) \cap (f_q/f_1 \cdots f_{n-1}))$, $g_j \in (L(f_n) \cap U(f_q/f_1 \cdots f_{n-1}))$, $(1 \leq q \leq n-1)$ となる g_k, g_j は仮定によって $f_1 \cdots f_{n-1}$ 線上ではそれぞれ $g_k \lhd g_k$, $g_j \lhd g_j$ となっているが，$f_1 \cdots f_{n-1}$ 線上で $g_k \leq f_q < g_j$, f_n 線上で $g_j \leq f_n < g_k$ となって $g_j \notin B(g_j)$, $g_k \notin B(g_j)$ の仮定に反する．

(7) $L(f_n) \cap U(f_p/f_1 \cdots f_{n-1}) \ni g_i$ のとき

$U(f_p/f_1 \cdots f_{n-1}) \cup \{h \mid g_i < h,\ h \in H_n\}$ を新たに $U(f_p/f_1 \cdots f_{n-1})$ とおき，$U(g_i/f_1 \cdots f_{n-1}) \cup \{h \mid g_i < h,\ h \in H_n\}$ とおく．$f_1 \cdots f_{n-1}$ 線上で $g_i < f_q$, $(1 \leq q \leq n-1)$ となる f_q が存在するときは (6) と同じ議論となる．このような f_q が存在しなければ $f_1 \cdots f_{n-1}$ 線上の g_i を右にスライドすることに問題はない．f_n 線上で $f_n \cup U(f_n)$ を右へスライドする場合は (6)(ⅲ) に似た議論となる．

(8) $U(f_n) \cap L(f_p/f_1 \cdots f_{n-1}) \ni g_i$ のとき

$U(g_i/f_1 \cdots f_{n-1}) \cup \{h \mid g_i < h,\ h \in H_n\}$ を改めて $U(g_i/f_1 \cdots f_{n-1})$ とおき，$U(f_n) \cup \{h \mid g_i < h,\ h \in H_n\}$ を改めて $U(f_n)$ とおいて，$g(S_i, f_n) \gtreqless g(S_i, f_1 \cdots f_{n-1})$ に応じて g_i を右にスライドする．この場合も $g_j \notin B(g_j)$ という仮定から有限回の操作で g_i の値を決定できる．

(9) $U(f_n) \cap U(f_p/f_1 \cdots f_{n-1}) \ni g_i$ のとき

$U(g_i/f_1 \cdots f_{n-1}) \cup \{h \mid g_i < h,\ h \in H_n\}$ を $U(g_i/f_1 \cdots f_{n-1})$ とおき，$U(f_n) \cup \{h \mid g_i < h,\ h \in H_n\}$ を $U(f_n)$ とおいて $g(S_i, f_n) \gtreqless g(S_i, f_1 \cdots f_{n-1})$ に応じて g_i を右にスライドする．この場合も $g_j \notin B(g_j)$ という仮定から有限回の操作で g_i の値を決定できる．

(10) 積集合が空集合のとき

この場合は f_n 線と $f_1 \cdots f_{n-1}$ 線上の f と g は，それぞれもとの値をもつとする．

以上の議論から H_N の要素は一つの実数直線上に並べることができ，したがって $h \notin B(h)$, $h \in H_N$ のときは順位関数が存在する．

定理 2 の証明

定理 2 の証明は次の補題 1，2 から得られる．

補題 1

順位文法 $G^{(n)} = (V_N^{(n)}, V_T, S, P^{(n)})$ に下記の $(P1)$—$(P3)$ の操作のいずれかを施して得られる文法 $G^{(n+1)} = (V_N^{(n+1)}, V_T, S, P^{(n+1)})$ は，次の (1)，(2) の条件を満たす．

$(P1)$ φ_1: $D \rightarrow x_1 C B y_1$, かつ $C \Rightarrow x_2 A$ $(x_2 \neq \phi)$ のとき，新しい記号 $B' (\notin (V_N^{(n)} \cup V_T))$ と生成規則 φ_1', φ_1'' $(\notin P^{(n)})$ を導入し，φ_1': $D \rightarrow x_1 C B' y_1$, φ_1'': $B' \rightarrow B$ で置き替える．

$(P2)$ φ_1: $D \rightarrow x_1 A C y_1$, $C \Rightarrow B y_2$ $(y_2 \neq \phi)$ のとき，これを，φ_1': $D \rightarrow x_1 A' B y_1$, φ_1'': $A' \rightarrow A$ で置き替える．ここで $A' \notin (V_N^{(n)} \cup V_T)$, φ_1', $\varphi_1'' \notin P^{(n)}$ とする．

$(P3)$ φ_1: $D \rightarrow x_1 A B y_1$ $(A \neq \phi \neq B)$ のとき，これを，φ_1': $D \rightarrow x_1 C B y_1$, φ_1'': $C \rightarrow A$, または，φ_1': $D \rightarrow x_1 A C y_1$, φ_1'': $C \rightarrow B$ で置き替える．ここで $C \notin (V_N^{(n)} \cup V_T)$, φ_1', $\varphi_1'' \notin P^{(n)}$ とする．

条件

(1) $L(G^{(n+1)}) = L(G^{(n)})$.

(2) $G^{(n+1)}$ は順位文法である．

証明

(1) については $L(G^{(n+1)})\supset L(G^{(n)})$, かつ $L(G^{(n+1)})\subset L(G^{(n)})$ となることから明らか.

(2) は $(P1)$ について考えると, この操作は $C\Rightarrow xF_i$ となるすべての F_i について関係 $F_i\rho\alpha B$ を, 関係 $F_i\rho\alpha B$, および $F_i\rho\alpha\lambda B'$ で置き替えることに相当する. B' は新しい要素であるから, $F_i\ (\neq C)$ について $F_i\alpha B'$ となることはない. したがって B' は第2節の条件 (2′), (3) を満たす.

$(P2)$, $(P3)$ についても同様の手順で証明できる. なお $(P1)$ は関係 $\rho\alpha$ を $\rho\alpha\lambda$ に, $(P2)$ は $\alpha\lambda$ を $\rho\alpha\lambda$ に, $(P3)$ は α を $\rho\alpha$, または $\alpha\lambda$ に置き替える手順である.

補題 2.

上の補題1の文法 $G^{(n)}$ で順位関数をもつものがある.

証明

$G^{(1)}$ を順位関数をもたない順位文法としよう. したがって $G^{(1)}$ は単調サイクルをもつ. その一つを,

$$f_1\lessdot g_2\lessdot f_2\lessdot\cdots\lessdot f_1$$

で表わし, この単調サイクルを消去する方法について考える.

(1) 順位関係がすべて $\rho\alpha\lambda$ のときは, $\lessdot$ を $\gtrdot$, または $\gtrdot$ を $\lessdot$ とすることによって, この単調サイクルは消去される.

(2) 順位関係が $\rho\alpha\lambda$ を含んでいないときは, 補題1の手順で文法を変更し, 順位を逆転させることで単調サイクルは消去される.

(3) 順位関係を逆転させることで新しい単調サイクルが生まれないときは, すべての単調サイクルが消去されたときの $G^{(n)}$ が求める文法である.

(4) 順位関係 $f_1\lessdot g_2$ を $f_1\gtrdot g_2$ と逆転させて新しい単調サイクル $f_1\gtrdot g_2\gtrdot f_i\gtrdot\cdots\gtrdot f_1$, $f_i\neq f_1$ が生まれるときは, g_2 を基準にして $g_2\lessdot f_i$ となるよう文法を変更する. g_2 の順位がいかなる f_i よりも小さくなるように文法を変更することができるから, g_2 を含む単調サイクルは消去可能である.

上の手順 (1), (2), (4) をくり返すことによって $G^{(n)}$ を得る.

補題 1, 2 から定理2が証明された.

参考文献

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) 長尾真: "Phrase structure grammar の構造分析について", 情報処理, Vol. 4, No. 4, pp. 186—193 (昭和38年7月).
5) 井上謙蔵: "右順位文法", 情報処理, Vol 11, No. 8, pp. 449-456 (昭和45年8月).
6) 浅井, 稲見, 藤村: "変形 PL360 言語の文法とコンパイラ", 第12回プログラミング報告集, 情報処理学会, pp. 197-207 (昭和46年1月).
7) 浅井清: "順位関数をもつ順位文法", 情報処理, Vol. 12, No. 5, pp. 264-271 (昭和46年5月).
8) N. Wirth: "Alogrithm 265, Find precedence functions", Comm. ACM, Vol. 8, No. 10, pp. 604-605 (Oct. 1965).
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) 浅井 清, 富山峯秀: "GPL-Genken Programming Language", JAERI M レポート 4762, 日本原子力研究所 (昭和47年2月).

(昭和46年11月11日受 付)
(昭和47年 2 月 7 日再受付)