拡張右順位文法とその解析法*

関　本　彰　次**

Abstract

The definition of the right precedence grammars and the parsing method for them has been given by K. Inoue.

In this paper, we attempt to miden further the definition and to make some corrections for its original representation and the parsing method, giving a new definition of extended right precedence grammars.

1. まえがき

本論文で取り上げた右順位文法は井上謙蔵氏[1)]によって創出されたものであり，その詳細な内容については，先に本紙において明らかにされている．

右順位文法の特長は文法解析の過程で把手を識別するための順位表として把手の右端を得るための右順位表のみをもち，左端を得るための左順位表を不要とするところにある．

このことは左順位についての文法上の制限を設けず，その意味で広い言語を取り扱えること，また左順位表をもつ文法よりも保持する表が少ないという有利さをもっている．

そこで注目しなければならないのは，右順位文法を規定する定義が充分広い言語範囲に及んでいるかどうかという点である．

これは現存するプログラミング言語への適用の可否とも関係する．

本論文では先に井上氏によって与えられた文法規定について考察し，文法規定をゆるめること，解析法を修正することによって右順位文法としてより広い範囲を包含しうるものに改めうることを示す．

2. 右順位文法

はじめに右順位文法について文献 1) に従って定義を与える．文脈独立 (Context free) 文法を

$$G=(\boldsymbol{V_N}, \boldsymbol{V_T}, \boldsymbol{P}, S)$$

と表わす．ただし，$\boldsymbol{V_N}$，$\boldsymbol{V_T}$，および $\boldsymbol{P}$ はそれぞれ非端記号 (non terminal symbol)，端記号 (terminal symbol) および生成規則 (production) の集合であるとする．また，S は

$$S\in \boldsymbol{V_N}$$

で出発記号 (start symbol) とする．

文形 (sentential form) η と ξ との間に

$$\eta=\beta A\gamma,\ \xi=\beta\alpha\gamma,\ A\to\alpha\in \boldsymbol{P}$$

なる関係があるとき，η は ξ を直接導く (directry derive) といい，ξ は η に直接還元される (directry reduced) という．この関係を

$$\eta\Rightarrow\xi$$

と表わす．ただし，

$$A\in \boldsymbol{V_N},\ \alpha,\beta,\gamma,\eta,\xi\in \boldsymbol{V}^*$$
$$\boldsymbol{V}\equiv \boldsymbol{V_N}\cup \boldsymbol{V_T}$$

で，$\boldsymbol{V}^*$ は $\boldsymbol{V}$ の要素からなる任意の有限個の記号の列からなる集合である．また文形 η，η_1，η_2，……η_n および ξ があって，

$$\eta\Rightarrow\eta_1\Rightarrow\eta_2\Rightarrow\cdots\cdots\Rightarrow\eta_n\Rightarrow\xi$$

のとき，これを

$$\eta\overset{*}{\Rightarrow}\xi$$

と表わし，η は ξ を導く，または ξ は η に還元されるという．右順位文法を示す．

文脈独立文法にたいして，文法，

$$\left.\begin{array}{l} G'=(\boldsymbol{V_N}', \boldsymbol{V_T}', \boldsymbol{P}', S'),\ \boldsymbol{V_N}'\equiv \boldsymbol{V_N}\cup\{S'\} \\ \boldsymbol{V_T}'\equiv \boldsymbol{V_T}\cup\{\vdash, \dashv\},\ \boldsymbol{V}'\equiv \boldsymbol{V_N}'\cup \boldsymbol{V_T}' \\ \boldsymbol{P}'\equiv \boldsymbol{P}\cup\{S'\to\vdash S\dashv\} \end{array}\right\}$$

がつぎの制限をもつとき，G' を右順位文法という．

---

* Extended Right Precedence Grammars and an Analyzing Technique for Them by Shoji Sekimoto (Mitsubishi Electric Corporation Kamakura Works)

** 三菱電機株式会社鎌倉製作所電子機器研究部

（i） $A \in \boldsymbol{V}_N'$ に対して，必ず

$A \overset{*}{\Rightarrow} U$, $U \in \boldsymbol{V}_T'^*$

を作ることができる．ただし，$\boldsymbol{V}_T'^*$ は $\boldsymbol{V}_T'$ の要素の任意の有限個の記号からなる列である．

（ii） $A \to \alpha$, $B \to \alpha$, $A \neq B$ である生成規則はない．

（iii） $S' \overset{*}{\Rightarrow} \alpha AB\beta$, A, $B \in \boldsymbol{V}'$

である記号 A, B の間に，非可換的で一義的な右順位関係が存在する．

右順位関係は，次のようにして作られる．まず，$A \in \boldsymbol{V}_N'$ に対して，集合 $\boldsymbol{L}(A)$ および $\boldsymbol{R}(A)$ を次のように定義する．

$$\boldsymbol{L}(A) \equiv \{F \mid A \overset{*}{\Rightarrow} F\alpha, A \in \boldsymbol{V}_N', F \in \boldsymbol{V}', \alpha \in \boldsymbol{V}_T'^*\}$$

$$\boldsymbol{R}(A) \equiv \{F \mid A \overset{*}{\Rightarrow} \alpha F, A \in \boldsymbol{V}_N', F \in \boldsymbol{V}', \alpha \in \boldsymbol{V}'^*\}$$

このほかに集合 $\boldsymbol{L}_T(A)$ を

$A \in \boldsymbol{V}_T'$ ならば $\boldsymbol{L}_T(A) \equiv \{A\}$

$A \in \boldsymbol{V}_N$ ならば $\boldsymbol{L}_T(A) \equiv \{F \mid A \overset{*}{\Rightarrow} F\alpha, F \in \boldsymbol{V}_T', \alpha \in \boldsymbol{V}'^*\}$

と定義する．さて，$B \in \boldsymbol{V}'$, β, $\gamma \in \boldsymbol{V}'^*$ として，

a） $A \to \beta BE\gamma$, で $C \in \boldsymbol{L}_T(E)$ ならば $B \lessdot C$

b） $A \to \beta DE\gamma$, で $B \in \boldsymbol{R}(D)$, $C \in \boldsymbol{L}_T(E)$ ならば $B \gtrdot C$ と右順位位を定義する．

ここで以下の記述中に使用される記号列についての関数 $H_i(\alpha)$ と $T_i(\alpha)$ についての規定をしておく．

$\alpha \in \boldsymbol{V}'^*$ であって，記号列 α の長さ $|\alpha| = n$ とするとき，$H_i(\alpha)$ は α の頭部の長さ i の部分記号列であり，また $T_i(\alpha)$ とは α の尾部の長さ i の部分記号列である．ここでは $0 \leq i \leq n$ とし，$H_0(\alpha)$, $T_0(\alpha)$ はいずれも空なる記号列，また $H_n(\alpha)$, $T_n(\alpha)$ はいずれも α を指すものとする．

（iv） 生成規則 $A \to \alpha\beta$, $B \to \beta$ があるときは

$C \to rH_i(\alpha)D\delta$, $D \overset{*}{\Rightarrow} Tn-1(\alpha)B\delta'$

は存在しない．ここで $n = |\alpha|$, $n \geq i > 0$ である．

また，任意の ξ, η について，$\xi \overset{*}{\Rightarrow} \eta$ は，$\xi \overset{*}{\Rightarrow} \eta$ または $\xi = \eta$ を表わす．

以上（i）～（iv）によって定義される文法を単純右順位文法と呼んでいる．

いま Wirth および Weber によって示さた順位文法とこの右順位文法の定義を対応してみると（i），（ii）については両者同じで，（iii）は順位文法では把手 (handle) を調べる際の左，右両端での $\boldsymbol{V}'$ の要素間での順位について言及しているのに対し，右順位文法では右端だけでのそれについて規定していること，さらに右順位文法では（iv）という規則が加わっていることである．順位文法では（iv）に対応する規則は述べていない．

いいかえると順位文法のように把手の左端に関する順位を規定しないかわりに規則（iv）という制限規則を設けているのである．このことは（iv）そのものによって，どの程度文法としての適用範囲がせばめられるかについて明らかにすることも必要である．

本文では右順位文法と他の文法との比較において上記について触れることは割愛して，右順位文法の（iv）についてこれをどのていどゆるやかなものにしうるか，解析法にどのような手をいれるべきかについて論ずる．

ここで拡張への論を進める前に，（iv）をそれと同等な（iv°）の表現をとってみる．規定表現のとり方は（iv）の簡潔な，まとまったものをとるのが正統ではあるが，ここでは，それを手続きとして，たとえばプログラム的に表現する場合のことなども考えて，多少冗長ではあるが，（iv°）に示したものを便宜上とることにした．

（iv°） 生成規則 $A \to \alpha\beta$, $B \to \beta$ があるときは，$n = |\alpha|$, $0 \leq i < n$ に対して，

$$C_1 \to \gamma_1 H_{n-i}(\alpha) D_1 \varepsilon_1 \tag{2.1}$$

$$D_1 \overset{*}{\Rightarrow} T_i(\alpha) B \delta_1 \tag{2.2}$$

が同時には存在せず，特に，(2.2) が存在するときは

$$0 \leq i + i_1 < n$$

なる i_1 に対して，

$$C_2 \to \gamma_2 H_{n-i-i_1}(\alpha) D_2 \varepsilon_2 \tag{2.3}$$

$$D_2 \overset{*}{\Rightarrow} T_{i_1}(H_{n-i}(\alpha)) D_1 \delta_2 \tag{2.4}$$

は同時には存在しない．これを l 回繰り返し

$$0 \leq i + i_1 + \cdots\cdots + i_{l-1} < n,$$

$$k = i + i_1 + \cdots\cdots + i_{l-1}$$

として，

$$D_l \overset{*}{\Rightarrow} T_{i\,l-1}(H_{n-k+i_{l-1}}(\alpha)) D_{l-1} \tag{2.5}$$

が存在するとき

$$0 \leq i + i_1 + \cdots\cdots + i_{l-1} + i_l < n$$

なる i_l に対して，

$$C_{l+1} \to \gamma_{l+1} H_{n-k-i_l}(\alpha) D_{l+1} \varepsilon_{l+1} \tag{2.6}$$

$$D_{l+1} \overset{*}{\Rightarrow} T_{il}(H_{n-k}(\alpha)) D_l \delta_{l+1} \tag{2.7}$$

は同時には存在してはならない．

なお，$1 \leq j \leq l+1$, に対し，γ_j, δ_j, ε_j はいずれも $\boldsymbol{V}'^*$ の要素である．また，任意の ξ, η について，$\xi \overset{*}{\Rightarrow} \eta$ とは $\xi = \eta$ または $\xi \overset{*}{\Rightarrow} \eta$ を表わす．

3章以下では (ⅳ) のかわりに (ⅳ°) を用いることにする.

3. 右順位文法の拡張

3.1 文法規則 (ⅳ°) について

前節の文法規則で (ⅳ°) が述べていることは右順位表に従って文章を左から右へ解剖 (parsing) していく過程で部分的に還元された記号列を A_1, A_2, ……A_n とし, これが push-down stack に貯えられているとして, あらたに読み込まれた端記号 T との間に,

$$A_n > T$$

の関係があったときに, $\boldsymbol{P}$ の要素の中から右辺の右端が A_n であって, かつ, 右辺を右側から照合して stack の途中までの部分列と一致する生成規則のうち, 右辺の長さが最長のものが把手であること, したがって, 還元に適用するべき生成規則はこれを右辺とする $\boldsymbol{P}$ の要素であると一意に決めるための制限であり, 換言すると最長でない右辺が把手となりうるような生成規則は除くということにほかならない.

ところでわれわれにとって右順位表を用いて解析を進めて行くとき必要なことは把手が一意に決定しうることであって, 最長列のものに整えるというのは便宜的な一つの制限の付け方であって必ずしも要求されることではない. そこで, この (ⅳ°) をゆるめた書き方に改めて, 文法の適用範囲をひろげることにする.

準備として, まず, $\boldsymbol{V}_T$ の部分集合,

$$\boldsymbol{Q}_T(A),\ \boldsymbol{Q}_T(\alpha|A)$$

を定義する. ここで, A は $\boldsymbol{V}_N'$ の要素, また α は $\boldsymbol{V}'^*$ の要素とする. そのために,

$$A \in \boldsymbol{V}_N'$$

に対して, A を少なくとも1つ右辺の右端以外のところに含む生成規則の集合を $\boldsymbol{P}_0(A)$ とする.

ここで A が出発記号のときは,

$$\boldsymbol{P}_0(A) \equiv \phi,\ \phi \text{ は空集合} \tag{3.1}$$

であって, このとき,

$$\boldsymbol{Z}(A) \equiv \boldsymbol{P}_0(A) \equiv \phi \tag{3.2}$$

と定義する.

次に, A を右辺の右端に含む生成規則であって, 左辺が出発記号でないものがあるときの左辺非端記号を

$$P_{0i} \in \boldsymbol{V}_N',\ i=1, 2, \cdots\cdots, n_0$$

とするとき, 各 P_{0i} について $\boldsymbol{P}_0(P_{0i})$ を求めて,

$$\boldsymbol{P}_1(A) \equiv \bigcup_{j=1}^{n_0} \boldsymbol{P}_0(P_{0i}) \tag{3.3}$$

とする. このような P_{0i} が存在しないときは,

$$\boldsymbol{Z}(A) \equiv \boldsymbol{P}_0(A) \tag{3.4}$$

とする.

つづいて, P_{0i} について, これらを右辺の右端に含む生成規則があって, 左辺が出発記号でないものが存在するとき, それら生成規則の左辺を, すべての P_{0i} について求め, これを

$$P_{1i} \in \boldsymbol{V}_N',\ i=1, 2, \cdots\cdots, n_1$$

とする. そして P_{1i} について P_{0i} のときと同様に $\boldsymbol{P}_0(P_{1i})$ を求めて,

$$\boldsymbol{P}_2(A) \equiv \bigcup_{i=1}^{n_1} \boldsymbol{P}_0(P_{1i}) \tag{3.5}$$

とする. 各 P_{0i} についてみて P_{1i} が存在しないときは

$$\boldsymbol{Z}(A) \equiv \boldsymbol{P}_0(A) \cup \boldsymbol{P}_1(A) \tag{3.6}$$

とする. 以下同様の手順を繰り返すことによって, A に対してある m が定まって,

$$\boldsymbol{P}_0(A),\ \boldsymbol{P}_1(A), \cdots\cdots,\ \boldsymbol{P}_m(A)$$

$$\boldsymbol{Z}(A) \equiv \bigcup_{j=1}^{m} \boldsymbol{P}_j(A) \tag{3.7}$$

が定まる. これは生成規則の数が有限であること, およびその作り方から明らかである.

なお, $\boldsymbol{P}_i(A)$ と $\boldsymbol{P}_j(A)$ の中には同一生成規則が両者の要素として, それぞれ含まれていることはある.

ここで $\boldsymbol{Z}(A)$ の部分集合 $\boldsymbol{P}_j(A)$ について次のことを考える. まず, $\boldsymbol{P}_0(A)$ について, その要素の右辺は A を1個以上含み, A の右隣りには必ず $\boldsymbol{V}'$ の要素が並んでいる. すなわち,

$$P_{0i}' \to \gamma A W_{0k} \gamma'$$

の形をしている. そこで, $\boldsymbol{P}_0(A)$ の要素の右辺の A と右に隣合っている W_{0k} のすべてについて,

$$\boldsymbol{L}_T(W_{0k}),\ k=1,\ 2, \cdots\cdots,\ n_0'$$

を求める.

$$\boldsymbol{P}_j(A),\ j=1,\ 2, \cdots\cdots,\ m$$

についても要素の右辺の P_{j-1i} の右に隣り合う W_{jk} のすべてについて,

$$\boldsymbol{L}_T(W_{jk}),\ k=1,\ 2, \cdots\cdots,\ n_j'$$

を求める. 以上の結果で

$$\boldsymbol{Q}_T(A) \equiv \bigcup_{j=0}^{m} \bigcup_{k=1}^{n_j'} (\boldsymbol{L}_T(W_{jk})) \tag{3.8}$$

によって $\boldsymbol{Q}_T(A)$ を定義する.

$\boldsymbol{Q}_T(\alpha|A)$ は $\boldsymbol{Q}_T(A)$ のときの A のかわりに αA を用いて同様に定義したものである.

$\boldsymbol{Q}_T(A)$ はその作られ方から, A の右隣りで A と順位関係をもつ端記号の全体であることは明らか.

$\boldsymbol{Q}_T(A)$, $\boldsymbol{Q}_T(\alpha|A)$ を定義することによって, 右順位

文法の規則 (iv°) を拡張的に，次の (iv′) のように，まず，改めることができる．

(iv′) 生成規則 $A\to\alpha\beta$, $B\to\beta$ があって，

$$\boldsymbol{Q}_T(A)\cap\boldsymbol{Q}_T(B)\fallingdotseq\phi,\ \phi \text{ は空集合,}$$

のときには，$n=|\alpha|$, $0\le i<n$ に対して，

$$C_1\to\gamma_1 H_{n-i}(\alpha)\text{Đ}_1\varepsilon_1 \qquad (3.9)$$

$$\text{Đ}_1\overset{*}{\Rrightarrow}T_i(\alpha)B\delta_1 \qquad (3.10)$$

が同時には存在せず，$0\le i<n$ で (3.10) が存在するときは，

$$0\le i+i_1<n$$

なる i_1 に対して，

$$G_2\to\gamma_2 H_{n-i-i_1}(\alpha)\text{Đ}_2\varepsilon_2 \qquad (3.11)$$

$$\text{Đ}_2\overset{*}{\Rrightarrow}T_{i_1}(H_{n-i}(\alpha))\text{Đ}_1\delta_2 \qquad (3.12)$$

は同時には存在しない．これを l 回繰り返し，

$$0\le i+i_1+\cdots\cdots+i_{l-1}<n,$$

$$k=i+i_1+\cdots\cdots+i_{l-1}$$

として，

$$\text{Đ}\overset{*}{\Rrightarrow}Ti_{l-2}(H_{n-k+i_{l-1}}(\alpha))\text{Đ}_{l-1}\varepsilon_l \qquad (3.13)$$

が存在するときは，

$$0\le i+i_1+\cdots\cdots+i_{l-1}+i_l<n$$

なる i_l に対して，

$$C_{l+1}\to\gamma_{l+1}H_{n-k-i_l}(\alpha)\text{Đ}_{l+1}\varepsilon_{l+1} \qquad (3.14)$$

$$\text{Đ}_{l+1}\overset{*}{\Rrightarrow}Ti_l(H_{n-k}(\alpha))\text{Đ}_l\delta_{l+1} \qquad (3.15)$$

は同時に存在してはならない．

なお，$1\le j\le l+1$ なる j に対し，γ_j, δ_j および ε_j はいずれも $\boldsymbol{V'^*}$ の要素である．また任意の ξ, η について $\xi\overset{*}{\Rrightarrow}\eta$ とは $\xi=\eta$ または $\xi\overset{*}{\Rrightarrow}\eta$ を表わす．

(iv′) によって，

$$\boldsymbol{Q}_T(A)\cap\boldsymbol{Q}_T(B)\equiv\phi$$

のときは (iv°) の条件からはずさせること，すなわち一意に把手が決定されることは明らかである．

さらに (iv′) は (iv″) に拡張することができる．(iv″) について述べた後で，解析法の修正の仕方について触れる．

(iv″) 生成規則 $A\to\alpha\beta$, $B\to\beta$ があって，

$$\boldsymbol{Q}_T(A)\cap\boldsymbol{Q}_T(B)\fallingdotseq\phi,\ \phi \text{ は空集合,}$$

のときは，$n=|\alpha|$, $0\le i<n$ に対して，

$$\boldsymbol{Q}_T(A)\cap\boldsymbol{Q}_T(T_i(\alpha)|B)\fallingdotseq\phi$$

となるような i に対しては，

$$C_1\to\gamma_1 H_{n-1}(\alpha)\text{Đ}_1\varepsilon_1 \qquad (3.16)$$

$$\text{Đ}_1\overset{*}{\Rrightarrow}T_i(\alpha)B\delta_1 \qquad (3.17)$$

が同時には存在しない．さらに，

$$0\le i<n$$

のとき，(3.17) を認めるならば，

$$0\le i+i_1<n$$

なる i_1 に対して，

$$C_2\to\gamma_2 H_{n-i-i_1}(\alpha)\text{Đ}_2\varepsilon_2 \qquad (3.18)$$

$$\text{Đ}_2\overset{*}{\Rrightarrow}T_{i_1}(H_{n-i}(\alpha))\text{Đ}_1\delta_2 \qquad (3.19)$$

は同時には存在しない．これを l 回繰り返し，

$$0\le i+i_1+\cdots\cdots+i_{l-1}<n,$$

$$k=i+i_1+\cdots\cdots+i_{l-1}$$

として，

$$\text{Đ}_l\overset{*}{\Rrightarrow}Ti_{l-1}(H_{n-k+i_{l-1}}(\alpha))\text{Đ}_{l-1}\varepsilon_l \qquad (3.20)$$

が存在するとき，

$$C_{l+1}\to\gamma_{l+1}H_{n-k-i_l}(\alpha)\text{Đ}_{l+1}\varepsilon_{l+1} \qquad (3.21)$$

$$\text{Đ}_{l+1}\overset{*}{\Rrightarrow}Ti_l(H_{n-k})\alpha))\text{Đ}_l\delta_{l+1} \qquad (3.22)$$

は同時には存在してはならない．

なお，$1\le j\le l+1$ なる j について，γ_j, δ_j および ε_j はいずれも $\boldsymbol{V'^*}$ の要素である．また任意の ξ, η について $\xi\overset{*}{\Rrightarrow}\eta$ とは $\xi=\eta$ または $\xi\overset{*}{\Rrightarrow}\eta$ を表わす．

文法規則 (iv°) の代わりに (iv″) を用いた右順位文法では，文章解剖中に stack の頭部の状態が，

$$\cdots\cdots\alpha\beta$$

となり，入力された端記号 T と $T_1(\beta)$ との間に，

$$T_1(\beta)>T$$

のとき，$A\to\alpha\beta$, $B\to\beta$ があったとしても把手は $\alpha\beta$ とはかぎらない．B と T との間に右順位表での関係があり，A と T とに関係がないときは $B\to\beta$ が適用されるべき生成規則である．この場合は，

$$T\notin\boldsymbol{Q}_T(A),\quad T\in\boldsymbol{Q}_T(B)$$

ということである．A と T, B と T のいずれの間にも順位関係があるときは，

$$T\in\boldsymbol{Q}_T(A)\cap\boldsymbol{Q}_T(B)$$

であって，(iv″) によって β が把手となる場合を除いている．すなわち，この場合は長い方の記号列 $\alpha\beta$ が把手となる．A と T に順位関係があり，B と T にはそれがないときは，$T\in\boldsymbol{Q}_T(A)$, $T\notin\boldsymbol{Q}_T(B)$ で $\alpha\beta$ の方が取られる．(iv) の制限では $A\to\alpha\beta$, $B\to\beta$ があるときは，本質的に $\text{Đ}\to T_i(\alpha)B\delta$, $i=n$, という生成規則が許せなかったが，

$$\boldsymbol{Q}_T(A)\cap\boldsymbol{Q}_T(T_i(\alpha)|B)\equiv\phi \qquad (3.23)$$

であれば，$T\in\boldsymbol{Q}_T(T_i(\alpha)|B)$ ならば，

$$T\notin\boldsymbol{Q}_T(A),\quad T\in\boldsymbol{Q}_T(B)$$

であるから，

$$\cdots\cdots\alpha\beta \quad \supset$$
$$\cdots\cdots\alpha\beta$$
$$\cdots\cdots\alpha\beta\delta \quad \supset$$
$$\cdots\cdots H_{n-i}(\alpha)\text{Đ} \quad \supset$$

なる変換がありうること，(3.23) が成立しないときでも条件 (iv″) からは $Đ\to T_i(\alpha)B\delta$ を必ずしも除去しなくてもよいことは明らかである．

3.2 文法規則 (ii) について

文法規則 (ii) は同一右辺を持つ異なる生成規則を許さないことを規定しているのであるが，現実の言語では，ほぼこのようなことは認められない．(ii) をゆるめるやり方にはいろいろあるが，ここでは右順位文法の解析法だけで処理しうる規定として修正する．

(ii′) $\boldsymbol{Q}_T(A)\cap\boldsymbol{Q}_T(B)\fallingdotseq\phi$ のとき，

$$A\to\alpha,\ B\to\alpha,\ A\fallingdotseq B$$

である生成規則はあってはならない．

上の規則によれば，α が把手であるとき，入力の端記号 T との間に順位関係をもつ A または B に対応して，$A\to\alpha$ または $B\to\alpha$ が一意に適用される．

3.3 拡張された右順位文法

文法規則 (i)，(ii)，(iii) および (iv°) によって定まる右順位文法に対して，文法規則 (i)，(ii′)，(iii) および (iv″) によって定まる文法を拡張された右順位文法と呼ぶことにする．

3.4 拡張された右順位文法の解析法

右順位文法の解析法では，右順位表を用いて把手の右端を見つけたとき，それを右端とする生成規則のうちで stack 中で照合一致する最長のものを把手としうるという手続きをとった．

これに対して拡張された方法では，把手の右端を見つけたならば，右辺の右端がそれと一致する生成規則について，その右辺が stack 中の記号列と照合一致するものをみいだし，その左辺の非端記号が，入力された端記号との間に右順位関係の成り立つものから，右辺の記号列の最長のものを把手としてとり，それを右辺にもつ生成規則を適用する．

拡張された右順位文法では，生成規則，

$$A\to\alpha\beta,\ B\to\beta$$

が右辺がともに stack の頭部の記号列と一致し，他に

$$C\to\gamma\alpha\beta$$

の形で，右辺が stack の頭部と一致するものがないとき，入力端記号 T に対して， $T_1(\beta)\gtrdot T$ とすれば，

$$A\gtrdot T \text{ または } A\lessdot T$$

のときは $\alpha\beta$ が把手であって，$A\to\alpha\beta$ が適用される．このとき

$$B\gtrdot T \text{ または } B\lessdot T$$

の関係があったとしても，この場合は，

$$\boldsymbol{Q}_T(A)\cap\boldsymbol{Q}_T(B)\fallingdotseq\phi$$

であるから

$$\boldsymbol{Q}(A)\cap\boldsymbol{Q}_T(T_i(\alpha)|B)\fallingdotseq\phi$$

のときは，(iv″) の制限によって β が把手となることはない．また B と T との間には右順位関係はあるが A と T との間にはそれがないときは，β が把手であり $B\to\beta$ が適用される．

4. 生成規則の付加について

右順位が一意に定まらない場合について，これを一義化する方法は文献 1) によって Presser の解法として示されたが，右順位文法へのこの方法の適用上の補足をここで述べておく．

文献 1) では，この部分は次のように説明されている．右順位が一義的でなくなるのは，次のような生成規則がある場合である．すなわち，$C\in\boldsymbol{V}_T{}'$ として，

(r_1) $A\to\beta BE\gamma$

かつ，$B\in\boldsymbol{R}(B)$，$C\in\boldsymbol{L}_T(E)$

(r_2) $A\to\beta BE\gamma$，$A\to\beta' ĐE\gamma'$ に対して，

$B\in\boldsymbol{R}(B)$，$C\in\boldsymbol{L}_T(E)$，$C\in\boldsymbol{L}_T(E')$

いずれの場合でも，$B\lessdot C$，かつ，$B\gtrdot C$ となる．

このような場合には，新しい非端記号 B' とその生成規則 $B'\to B$ を導入し，

$$A\to\beta BE\gamma$$

を，

$$A\to\beta B'E\gamma \text{ および } B'\to B$$

に書き換える必要がある．

上の説明では (r_2) に対し，$B\in\boldsymbol{V}'$ で，

$$F\to\beta B$$

という生成規則があるときは単純右順位文法の解析では誤った解析におちいる．このような生成規則についても，

$$F\to\beta B'$$

と書き改めておかなくてはならない．さらに，

$$G\to B$$

という生成規則がある場合には，

$$G\to B'$$

と書き改めなくては同一右辺の生成規則で異なるものが二つ以上生ずることになる．以上の二点を補足しておく．

このように新しい非端記号とその生成規則を導入するという発想で，次のような右順位文法の拡張の方法も考えられる．だたし，これは意味付け法に依存的であって，文法拡張法としては一般的なものではないことはあらかじめ断わっておく．

5. 右順位文法の意味付け依存的な拡張

2章の右順位文法を規定する条件のうちで，（iv）に従うと，

$A\to\alpha\beta$，$B\to\beta$ があるときは，

$n=|\alpha|$，$n\geq i>0$ に対して，

$$C\to\gamma\alpha D\delta,\ D\overset{*}{\Rightarrow}B\delta' \qquad (5.1)$$

は同時には存在しないのであるが，（iv）の前提のもとに (5.1) が同時に起こる場合でも，特別の場合には（iv）を満足するように，書き換えることが可能なことがある．(5.1) が同時に起こるとは，

換言すると，stack の状態が，

$\cdots\cdots\gamma\alpha\beta$

の並びのときに，生成規則 $B\to\beta$ の適用によって，β を B に還する場合が起こる．

この場合でも，$G\to B$ なる生成規則がないかぎり，

$A\to\alpha B$ を $A\to\alpha B'$，$B'\to\beta$

とし，新しい非端記号 B' とその生成規則を導入することによって（iv）を満足するように変換することができる．

さて，この方法で問題になるのは，前にも断わったように，実際の言語プロセッサを考えるときに必要な意味付け法に独立でない点である．

というのは，4章で $\boldsymbol{B'}$ を導入したときとは本質的に異なることが起こっているからである．4章での B' の生成規則にはなんらの意味付けを伴わず，また B を B' で置き換えられた生成規則についても意味の変化はなかったのであるが，今回の場合は $B'\to B$ の意味付けとして $B\to\beta$ で行なった意味付けを無効にするという意味付けが与えられなければならない．したがって，もとの言語に与えられた意味付けの仕方とは独立でなくなる．それゆえ，いつも可能とはいえない．

ただ，ここでいえることは，こうした意味付け変更が許されるようなプロセッサ作成法がとれるときには，この方法は簡単な手続によるものとして採用しうるということであって，自動作成を含めた一般的な方法としてはとり得ないであろう．

6. あとがき

コンパイラ作成の自動化という立場での一問題として，われわれはこれまでも様々な文法およびその解析法について考察してきたが，この方面で最近最も興味のあるのは，適用範囲が現存の言語をカバーし，最も効率のよい文法規定とその解析法についてである．

一般的・理論的という意味で包括的な文法規定というものについては，上述の目的への考察の過程としては重要であるが，あくまでわれわれは現存の言語をカバーしうると同時に，実際的な効率を期待しうるものを求めている．この意味で，われわれは右順位文法に興味をもち，この文法の適用範囲について考察する過程で，この文法の適用範囲の拡大をはかってみることにした．

本稿では右順位文法の立場のみからの拡張定義を行なっているが，これまでの発想をもとにして，さらには右順位文法をこえての拡大の方向が考えられる．

右順位文法をもとにして，ある意味での左順位をもつようにすれば，かなり満足のいく範囲に及ぶ文法が得られるのではないかという期待をもっている．

それは右順位表としては $\boldsymbol{V'}$ と $\boldsymbol{V_T'}$ の要素間での，この並びでの順位関係であったが，これに対し，$\boldsymbol{V'}$ と $\boldsymbol{V_N'}$ の要素間でのこの並びでの左順位をもつということである．この表の使用は，変換 $A\to B$ を適用するときに β の左隣りの記号 X と A の間の関係を調べるもので，この表の導入によって，もはや右順位文法ではなくなるが，もとの右順位文法の手法を生かして，もとの文法がどのように拡張，変更されるか，また現存の言語との比較については稿を改めて明らかにしたい．

本稿を閉じるにあたり，本文法の発想を最初に与えられた井上謙蔵氏に深く敬意を表しますとともに，本文発表の機会を与えられ，適切な指示をいただいた首藤 勝氏，ならびに内容について検討いただいた春原猛氏に厚く感謝します．

参 考 文 献

1) 井上謙蔵：右順位文法，情報処理，Vol. 11, No. 8, Aug. 1970.
2) 井上謙蔵：順位言語の右順位解析，情報処理，Vol. 11, No. 4, April 1970.
3) Irons, E. T.: A syntax directed Compiler for ALGOL 60. Comm. ACM 4 (Jan. 1961).
4) Feldman, J. and Gries, D.: Translator Writing Systems. Comm. ACM 11 (Feb. 1968)
5) McKeeman‘ W. M.: An approach to Computer lungcage design. Tech. Rpt. CS48, Computer Science Dept., Stanford U. Aug. 1966.
6) Hopgood, F. R. A.: COMPILING TECHNIQUE. COMPUTER MONOGRAPH SERIES., American Elsevier Publishing Company, Inc. New York. 1969.

(昭和46年3月18日受付)