



パーシングテーブルによる弱順位パーザの構成法*

青江 順一** 山本 米雄**
原田 尚文*** 島田 良作**

Abstract

This paper proposes a method for the construction of a weak precedence parser by parsing tables, which are considered as finite items with "side effects" that manipulate a pushdown list. We present algorithm to construct the parsing tables from Right Graph and Left Graph. The obtained parser π excels the parser M with weak precedence matrix in memory size and speed. The parser π also excels the parser with weak precedence functions in error detecting capability and the class of grammars.

1. ま え が き

弱順位パーザの縮小法として、弱順位行列を関数に変換する方法がある^{2)~4)}。しかし、これらの方法によるパーザでは受理する言語のクラス及びエラー処理能力が低下するという欠点がある。

また、弱順位行列は還元に適用すべき生成規則の番号をもたないので、還元動作ごとに生成規則表を探索する必要がある¹⁾。従って、還元動作の速度は $LR(k)$, $LL(k)$ 文法によるパーザより劣る。

本論文では、パーシングテーブル⁵⁾による弱順位パーザの構成法を提案する。パーシングテーブルは弱順位パーザの状態にそれぞれ対応する有限の項からなっており、本方法では次の2つのパーシングテーブルを構成する。

(1) シフト、還元の決定を行うパーシングテーブル $T_S(G)$ 。

(2) 還元動作と還元適用された生成規則番号の決定を同時に行うパーシングテーブル $T_R(G)$ 。

そして、 $T_S(G)$ と $T_R(G)$ の能率的な連結方法を用いたパーザの操作法を提案し、次の点が改善されることを示す。

(i) 受理する言語のクラス及びエラー処理能力を低下することなしに、パーザの規模を縮小できる。

(ii) シフト、還元の決定及び還元動作が能率的となり、解析速度を高速にできる。

以上の理論を展開するにあたり、2.では、本論文の基礎となる用語、諸定義について述べ、3.では、順位関係のグラフ化として RG , LG を定義し、その性質について述べる。4.では、 RG , LG からパーシングテーブルを構成する方法とそのテーブルによる弱順位パーザについて述べる。5.では、パーザの簡単化の方法について述べ、解析速度、パーザの規模に対する評価を与える。

2. 諸 定 義

2.1 文 法

文脈自由文法 (context free grammar) を $G_c = (V_N, V_T, P, S)$ と表わす。ここで、 V_N, V_T はそれぞれ非終端語、終端語の有限集合である。 P は $A \rightarrow \alpha$ なる生成規則の有限集合である。 S は V_N の要素で始記号を表わす。

$V = (V_N \cup V_T)$ なる V に対し、有限長の記号列のすべてからなる集合を V^* で表わし、 V^* は V^* から空記号 ϵ を除いた集合を表わす。また、 $\$$ を文の両端にのみ現れる記号とし、 $\$ \in V$ とする。以後、次の記号を特に断わらずに使用する。

$$A, B, C \dots \in V_N$$

* The Construction of Weak Precedence Parsers by Parsing Tables by Jun-ichi AOE, Yoneo YAMAMOTO, Naohumi HARADA and Ryouzaku SHIMADA (Faculty of Engineering, Tokushima University).

** 徳島大学工学部情報工学科

*** 徳島大学工学部電子工学科

$$\begin{aligned} a, b, c, \dots &\in V_T \cup \{\$\} \\ \dots, X, Y, Z &\in V_U \{\$\} \\ \dots, x, y, z &\in (V_T \cup \{\$\})^* \\ a, \beta, \gamma, \dots &\in (V_U \{\$\})^* \end{aligned}$$

また, “#” で任意の記号を表わすものとする

文法 G_c において, 正当な文法 G_p (proper grammar) は次の条件を満足する.

- (1) $S \xrightarrow{*} xAy \xrightarrow{*} xzy$ が存在する.
- (2) $A \rightarrow \epsilon \in P$.
- (3) $A \xrightarrow{+} A$ が存在しない.
- (4) $A \rightarrow \alpha, B \rightarrow \alpha$ ならば, $A = B$ である.

ここで, $\xrightarrow{*}, \xrightarrow{+}$ はそれぞれ導出記号 \Rightarrow の反射推移閉包 (reflexive and transitive closure), 推移閉包 (transitive closure) を表わす. 以後, ある関係の添字 $*$, $+$ はその関係の反射推移閉包, 推移閉包を表わすものとする. また, 文法 G による言語を $L(G) = \{w \mid S \xrightarrow{*} w, w \in V_T^*\}$ で表わす.

2.2 順位関係と弱順位文法

集合 E 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. また, 2つの対関係 ρ, σ の積 $\rho\sigma$ を

$$a\rho\sigma b \equiv [\exists c \in E; a\rho c \sigma b]$$

で定義する.

文法 G_p の任意の要素 X, Y について 3つの対関係 μ, ρ, λ を次のように定義する.

$$\begin{aligned} X\mu Y &\equiv [\exists \alpha, \exists \beta; A \rightarrow \alpha XY \beta] \\ A\rho X &\equiv [\exists \alpha; A \rightarrow \alpha X] \\ A\lambda X &\equiv [\exists \alpha; A \rightarrow X\alpha] \end{aligned}$$

順位関係 (precedence relations) $\doteq, <, >$ はこれらの対関係より次のように定義される.

$$\doteq = \mu, < = \mu\lambda^+, > = \rho^*\mu\lambda^*$$

[定義 1] 文法 G_p において弱順位文法 G_w (weak precedence grammar) は次の条件を満足する²⁾.

- (1) $\leq \cap > = \phi$, ϕ は空集合を表わす.
- (2) $A \rightarrow \alpha X \beta, B \rightarrow \beta \in P$ ならば, $X \leq B$ が成立しない. 但し, $\leq = (< \cup \doteq)$.

2.3 弱順位行列によるパーザ

文法 G_p を $G_p = (V_N \cup \{S_0\}, V_T, P \cup \{S_0 \rightarrow S\}, S_0)$ とし, 以後, この文法を $G_p = (V_N, V_T, P, S_0)$ として改めて使用する. P の要素は $S_0 \rightarrow S$ を 0 番目と

して, 整数 1 より適当に順序づけられているものとし, 第 p 番目の要素を $A_p \rightarrow \alpha_p$ で表わす ($0 \leq p \leq u$). 以後, p, q, r など生成規則番号を表わす*. また, α の尾を $t(\alpha)$ で表わし, 特に, 長さを指定したい場合は $t_i(\alpha)$ と書く ($i \geq 1$).

パーザの第 n 番目の計算状況を

$$Q_n = (\$ \beta, a x \$, t)$$

で表わす. ここで, () 内の第一, 第二, 第三要素はそれぞれプッシュダウンスタックの内容, 入力記号列, right parse を表わす. このとき, 弱順位行列 M によるパーザ (以後, パーザ M とよぶ) は次のように定義される. 但し, $t_1(\beta) = X$ とする.

- [定義 2]** (1) $M(X, a) = >^{**}$ であって, $t(\beta) = \alpha_p, t(\alpha_p) = a\alpha_p$ ならば, $Q_n \vdash (\$ \gamma A_p, a x \$, t p), \beta = \gamma \alpha_p$.
 $t(\beta) = \alpha_r$ ならば, $Q_n \vdash (\$ \delta A_r, a x \$, t r), \beta = \delta \alpha_r$.
- (2) $M(X, a) = \leq$ ならば, $Q_n \vdash (\$ \beta a, x \$, t)$.
- (3) $(\$ S_0, \$, t) = \text{accept}$.
- (4) (1), (2), (3) 以外は error である.

ここで, \vdash は計算状況の変化を表わす記号である.

パーザ M では, ハンドル及び生成規則の決定を最長一致法で行う. 従って, $>$ 関係が還元を表わし, \leq 関係がシフトを表わす. またそれらの関係も $(V_U \{\$\}) \times (V_T \cup \{\$\})$ 上で定めればよい.

3. RG, LG とその性質

本章では, パーズングテーブル (parsing table, 以後, 文法 G に対するパーズングテーブルを $T(G)$ と略記する) を構成するための準備として, グラフ RG, LG を定義し, その性質について述べる.

3.1 RG, LG の性質

グラフ K を $K = (N, R)$ で表わす. ここで, N はノードの有限集合を表わし, R は集合 N 上の関係を表わす. $a, b \in N, (a, b) \in R$ ならば, ノード a からノード b にアークが引かれているものとする.

[定義 3] 文法 $G_p = (V_N, V_T, P, S_0)$ に対して, グラフ $(V_1, \rho), (V_2, \lambda)$ を定義し, それらのグラフに次の (1), (2) を行ったものをそれぞれ RG, LG という. ここで, $V_1 = \{X \mid A \rho^* X, \forall A \in V_N\}, V_2 = \{Y \mid A \lambda^* Y, \forall A \in V_N\}$.

- (1) $a \in V_1, b \in V_2, a\mu b$ なる b をグラフ (V_2, λ) のノードとする.
- (2) グラフ (V_1, ρ) のノード X とグラフ (V_2, λ) のノード Y にそれぞれ $\mu_R(X), \mu_L(Y)$ を隣接集合として加える. ここで, $\mu_R(X) = \{Y \mid X\mu Y\}, \mu_L(Y) =$

* 単に, A_p, α_p と書けば, 生成規則 $A_p \rightarrow \alpha_p$ の左辺と右辺を表わすものとする.
 ** 行ラベル X と列ラベル a に対応する行列 M の要素を $M(X, a)$ で表わす.

$\{X|X\mu Y\}$, 但し, $\mu_R(S_0), \mu_L(S_0)$ には S をその要素に入れる。

以後, RG のノード A から k 番目にあるすべての子孫を要素とする集合を $R^k(A)$ で表わし, $R_N^k(A) = R^k(A) \cap V_N, R_T^k(A) = R^k(A) \cap V_T$ とする。また, LG についても集合 $L^k(A), L_N^k(A), L_T^k(A)$ を同様に定義する。

3.2 RG, LG の性質

(性質 1) 文法 G , のすべての順位関係を, RG, LG は次のように与える。

(1) RG のすべてのノード X と $\forall Y \in \mu_R(X)$ なる Y において $X \leq Y$ である。 LG すべてのノード Y と $\forall X \in \mu_L(Y)$ なる X において $X \leq Y$ である。

(2) RG のすべてのノード A による $\forall X \in R^+(A)$ なる X と $\forall Z \in \mu_R(A), \forall Y \in L^*(Z)$ なる Y において $X > Y$ である。

(3) LG のすべてのノード A による $\forall X \in L_L(A)$ なる X と $\forall Y \in L^+(A)$ なる Y において $X < Y$ である。

(証明) RG は $V_1 \times V$ 上のすべての \leq 関係を, また, LG は $V \times V_2$ と $\bar{V}_1 \times \bar{V}_2$ 上 (定義 3 の (1) による) のすべての \leq 関係を (1) で与える。よって, RG, LG が (1) において $V \times V$ 上のすべての \leq 関係を与えることは明らかである。ここで, \bar{V}_1, \bar{V}_2 はそれぞれ V に対する V_1, V_2 の補集合を表わす。また, RG, LG のすべてのノード A に対して $R^+(A) = \{X|A\rho^+X\}, L^+(A) = \{Y|A\lambda^+Y\}$ が成立するので, RG, LG が (2), (3) において $V \times V$ 上のすべての $>, <$ 関係を与えることは明らかである。(証明終)

(性質 2) 文法 G , が弱順位文法であるための必要十分条件は, RG, LG において次の (1), (2) を満足することである。

(1) $X \in R^+(A), \forall Y_1 \in \mu_R(A), \forall Y_2 \in \mu_R(X)$ において, $L_T^*(Y_1) \cap L_T^*(Y_2) = \phi$ 。

(2) $A \rightarrow \alpha X \beta, B \rightarrow \beta \in P$ において, $X \in \mu_L(C), B \in L_N^*(C)$ なるノード C が LG に存在しない。

(証明) 必要: (1) において, $a \in (L_T^*(Y_1) \cap L_T^*(Y_2))$ なる a が存在するとき (Fig. 1-(a)), $X \leq a$ かつ $X > a$ となるので, 定義 1 の (1) に反する。(2) において, $X \in \mu_L(C), B \in L_N^*(C)$ なるノード C が存在するとき (Fig. 1-(b)), $X \leq B$ となるので, 定義 1 の (2) に反する。

十分: $X < a$ が成立するのは, $X \in R^+(A), Y_1 \in \mu_R(A), a \in L_T^*(Y_1)$ なる A, Y_1 が存在する場合に限

* 要素 A を $A\lambda^+a$ なるすべての記号 a に代えることを意味する。

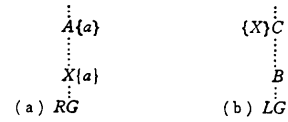


Fig. 1 Examples of RG and LG

る。 $X \leq a$ が成立するのは, $X \in \mu_L(Y_2), a \in L_T^*(Y_2)$ なる Y_2 が存在する場合に限る。よって, (1) が成立する場合は明らかに $\leq \cap > = \phi$ である。また, (2) が成立すれば, $X \leq B$ なる関係が存在しないことも明らかである。故に, (1), (2) を満足する文法 G , は弱順位文法である。(証明終)

〈例 1〉 算術式を文法 G_1 として, RG, LG の構成例を Fig. 2 に示す。

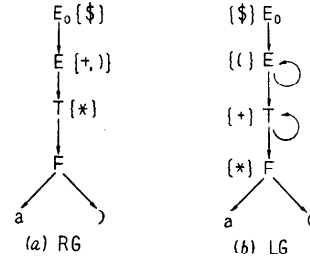


Fig. 2 RG and LG for grammar G_1

$G_1 = (\{E_0, E, T, F\}, \{a, (,), +, *\}, P, E_0)$
 $P: 0. E_0 \rightarrow E, 1. E \rightarrow E+T, 2. E \rightarrow T, 3. T \rightarrow T*F, 4. T \rightarrow F, 5. F \rightarrow (E), 6. F \rightarrow a$

これらの RG, LG が順位関係を与えることは容易にわかる。例えば, RG の $T\{*\}$ のノードから $R^+(T) = \{F, a, \}$ であるので, $T \leq *, F > *, a > *, \} > *$ である。また, $L_T^*(+, \}) \cap L_T^*(*) = \{+, \} \cap \{*\} = \phi$ であり, $E \rightarrow E+T, E \rightarrow T$ に対して $+ \in \mu_L(T), E \in L_N^*(T); T \rightarrow T*F, T \rightarrow F$ に対して $* \in \mu_L(F), T \in L_N^*(F)$ である。よって, 文法 G_1 は弱順位文法である。

以上のように, RG, LG は順位関係の決定及び弱順位文法の検定に対して有効なものである。

以後, 簡単のために RG の $A \in \mu_R(X)$ なる要素 A を, $L_T^+(A)$ の要素に代えておく。

次に, RG のパスに関する性質について述べる。

Fig. 3 のような RG が存在するとき,

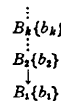


Fig. 3 Path of RG

$\beta_k B_k \Rightarrow \beta_{k-1} B_{k-1} \Rightarrow \dots \Rightarrow \beta_1 B_1$; $B_2, \dots, B_k \in V_N, B_1 \in V_T, k \geq 2$ なる導出が必ず存在する。性質 1 によりパーザ M は、

$$Q_n = (\$ \gamma \beta_1 B_1, b_k x \$, t_1)$$

なる計算状況に対して

$Q_n \vdash (\$ \gamma \beta_2 B_2, b_k x \$, t_2) \vdash \dots \vdash (\$ \gamma \beta_k B_k, b_k x \$, t_k) \vdash (\$ \gamma \beta_k B_k b_k, x \$, t_k) = Q_{n+k}$ と変化する。従って RG のアークを逆にたどるパスのノードラベルは連続した還元動作に対するスタックのトップ記号の変化を示す。

(性質 3) スタックシンボル a , 入力記号 b で還元動作が始まれば, $a \in R^+(B), b \in \mu_R(B)$ なる B がスタックシンボルに現れるまで還元動作が続き, その後, 必ずシフト動作が起こる。但し, 途中でエラーとなる場合を除く。

(証明) 性質 1, 2 より明らかである。(証明終)

また, RG のアーク (A, X) には, $t_1(\alpha_p) = X$ なる p を対応させることができるので, 連続した還元動作に適用されるべき生成規則番号の順序もこの RG のパスより知ることができる。

4. $T(G)$ による弱順位パーザ

本章では, $T(G)$ による弱順位パーザ (以後, パーザ π と略記する) について述べる。

3.2 で述べたようにパーザ M では, Q_n から Q_{n+k-1} までの各状況に対して常にシフト, 還元の決定を行う必要がある。本論文では, 性質 3 を利用することにより Fig. 4 の操作法を提案する。そして, I の範囲に相当する $T_S(G)$ と II の範囲に相当する T_R

(G) を構成する。

この方法では, $T_S(G)$ で還元を決定すれば, $T_R(G)$ で連続還元動作を行い, 次に 1 回のシフト動作を行うまで, シフト, 還元の決定を行う必要がない。

以下, $T(G)$ の定義, $T(G)$ の構成法, パーザ π の操作法の順に話を進める。

4.1 $T(G)$ の定義

$T(G)$ は状態 S をもつ $T_S(G)$ と状態 R をもつ $T_R(G)$ よりなっており, それぞれの内容を Table 1 に示す。

Table 1 Contents of $T(G)$

(a) $T_S(G)$				
状態名	入力記号	入口状態	出口状態	行先状態
S_a	b_1	R_a	R_{b_1}	S_{b_1}
	\vdots		\vdots	\vdots
	b_n		R_{b_n}	S_{b_n}
(b) $T_R(G)$				
状態名	スタックシンボル	動作	出力	行先状態
R_r	X_1	$\leftarrow A$	p_1	R_1
	\vdots	\vdots	\vdots	\vdots
	X_i	\rightarrow	p_i	R_i

状態 S_a のときの各入力記号に対する内容を

$S_a = [\text{入力記号}, \text{入口状態}, \text{出口状態}, \text{行先状態}]$

で表わす*。入口状態と出口状態はそれぞれ $T_R(G)$ で還元動作を行うべき最初の状態と還元動作を終るべき最後の状態を表わす。行先状態は $T_S(G)$ で次に進むべき状態を表わす。状態 S_a に対する入口状態は 1 個であって, 出口, 行先状態は各入力記号に対応して存在する。

状態 R_r のときの各スタックシンボルに対する内容を

$R_r = [\text{スタックシンボル}, \text{動作}, \text{出力}, \text{行先状態}]$ で表わす。動作には, $\rightarrow, \leftarrow A, \leftrightarrow A$ があって, それぞれスタックシンボルを pop up すること, 記号 A を push down すること, スタックシンボルを pop up して記号 A を push down することを表わす。出力, 行先状態は各スタックシンボルに対応して存在し, それぞれ出力すべき生成規則番号, $T_R(G)$ で次に進むべき状態を表わす。

入口状態, 出口状態, 動作, 出力がないことを b で表わす (表中では空白である)。

4.2 $T_S(G), T_R(G)$ の構成

$T_S(G)$ は性質 1, 3 より容易に得られる。

(手順 1) $T_S(G)$ の構成

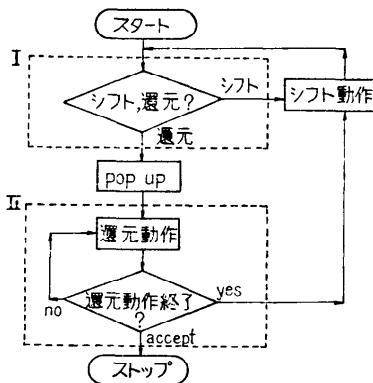


Fig. 4 Algorithm of parser π

* 記号 a に対応する状態 S を S_a で表わす。状態 R_r も同様である。また出口状態を $R_{b'}$ で表わす。

(1-1) RG のノード a , $a \in R^+(B)$ なるノード B に対し $b \in \mu_R(B)$ とするとき, $S_a = [b, R_a, R_{B'}, S_b]$ とする. また, 状態 $R_{B'}$ が一意に決定できない場合, それらを $R_{B'_1}, R_{B'_2}, \dots, R_{B'_r}$ とすれば ($r \leq 2$), $R_{B'} = R_{B'_1} = R_{B'_2} = \dots = R_{B'_r}$ とする.

(2-2) LG の $c \in \mu_L(Y)$, $d \in L_T^*(Y)$ なるすべての c, d を求め, $S_c = [d, \mathbf{b}, \mathbf{b}, S_d]$ とする.

$T_R(G)$ は RG の各アークに生成規則番号を対応させることで得られる.

(手順 2) $T_R(G)$ の構成

(2-1) RG の (A_p, X_i) なるアークを求める. $\alpha_p = X_m X_{m-1} \dots X_1$ とすれば, $m \geq 2, 1 \leq i \leq m-2$ なる X_i に対して $R_{X_i} = [X_{i+1}, \rightarrow, \mathbf{b}, R_{X_{i+1}}]$ とし, X_{m-1} に対して $R_{X_{m-1}} = [X_m, \leftrightarrow, A_p, p, R_{A_p}]$ とする. また, $m=1$ ならば, $R_{X_1} = [\#, \leftarrow, A_p, p, R_{A'_p}]$ とする.

(2-2) $\alpha_p = Z_l Z_{l-1} \dots Z_1 X_m X_{m-1} \dots X_1$, $\alpha_q = Y_k Y_{k-1} \dots Y_1 X_m X_{m-1} \dots X_1$, $Z_i \neq Y_1$ ならば, $l \geq 1, k \geq 1, 1 \leq j < m$ なる X_j , $1 \leq t \leq l$ なる Z_t , $1 \leq s \leq k$ なる Y_s に対して手順(2-1)と同様に行い*, X_m に対して $R_{X_m} = [Z_l, \rightarrow, \mathbf{b}, R_{Z_l}]$, $R_{X_m} = [Y_1, \rightarrow, \mathbf{b}, R_{Y_1}]$ とする. また, $k=0$ ならば, $Y_1 = \#$ として, $R_{X_m} = [\#, \leftarrow, A_q, q, R_{A'_q}]$ とする.

(2-3) $R_{B'} \neq R_{S'}$ なる状態 $R_{B'}$ に対して $R_{B'} = [B, \rightarrow, \mathbf{b}, R_B]$ とする. $R_{B'} = R_{B'_1} = R_{B'_2} = \dots = R_{B'_r}$ ならば, $R_{B'} = [B_i, \rightarrow, \mathbf{b}, R_{B_i}]$ とする.

〈例 2〉 文法 G_1 に対する $T(G_1)$ を Table 2₁ に示す.

4.3 パーザ π の操作法

文 w が $w \in L(G_w)$ ならば, 文 w に対する right parse を出力し, $w \notin L(G_w)$ ならば, "error" を出力するパーズングアルゴリズムを示す.

(アルゴリズム 1)

(1) 計算状況 $Q_0 = (\$, w\$, \epsilon)$, $T_S(G)$ の状態 S'_0 でスタートし, $Q_n = (\$ \gamma X, a x \$, t)$, $X \in V_T$ まで解析が進んでいるものとする. このときの $T_S(G)$ の状態を S_X とする.

(2) $S_X = [a, \mathbf{b}, \mathbf{b}, S_a]$ ならば, (5) へ行く.

$S_X = [a, R_a, R_{B'}, S_a]$ ならば, 次へ行く.

(3) pop up を 1 回行い. 次へ行く.

(4) 状態 R_a から状態 $R_{B'}$ まで還元動作を行い, 次へ行く.

但し, $R_{B'} = R_{S'}$ ならば, 計算状況を確認し

* 但し, X_{m-1} に対しての出力 p は省く.

** シフト, 還元の設定ができない場合をシフトエラーという. また, 還元動作ができない場合を還元エラーという.

Table 2 $T(G_1)$ for grammar G_1

(a) $T_S(G_1)$

状態名	入力記号	入口状態	出口状態	行先状態
S_a	*	R_a	$R_{r'}$	S_a
)		$R_{g'}$	S_i
	+		$R_{g'}$	S_a
	\$		$R_{g'}$	S_g
S_i	*	R_i	$R_{r'}$	S_a
)		$R_{g'}$	S_i
	+		$R_{g'}$	S_a
	\$		$R_{g'}$	S_g
S_a	a			S_a
	(S_c
S_c	a			S_a
	(S_c
S_i	a			S_a
	(S_c
S_g	a			S_a
	(S_c

(b) $T_R(G_1)$

状態名	スタックシンボル	動作	出力	行先状態
R_a	#	$\leftarrow F$	6	$R_{r'}$
R_i	E	\rightarrow		$R_{g'}$
$R_{g'}$	($\leftrightarrow F$	5	$R_{r'}$
$R_{r'}$	F	\rightarrow		R_f
R_f	#	$\leftarrow T$	4	$R_{r'}$
	*	\rightarrow		R_a
R_a	T	$\leftrightarrow T$	3	$R_{r'}$
$R_{r'}$	T	\rightarrow		R_f
R_r	#	$\leftarrow E$	2	$R_{g'}$
	+	\rightarrow		R_a
R_a	E	$\leftrightarrow E$	1	$R_{g'}$
$R_{g'}$	E	\rightarrow		R_g
R_g	#	$\leftarrow E_0$	0	$R_{r'}$

て $(\$ S_0, \$, t')$ なる場合, t' を出力して操作を終る.

(5) シフト動作を 1 回行い, (2) へ行き, 以上の操作を繰り返す. このときの $T_S(G)$ の状態は状態 S_a である.

(6) (2), (4) 以外の場合は, "error" を出力して操作を終る.

(補題 1) パーザ π は, シフト動作, シフトエラー** の検出についてパーザ M と等価である.

(証明) $X \in V_T$ なる X に対して, 性質 1, 3 より

$$M(X, a) = \leq S_X = [a, \mathbf{b}, \mathbf{b}, S_a]$$

なる対応関係が成立する。また、 $X \in V_N, X\rho^*c$ なる X, c に対して

$$M(X, a) = \leq \Rightarrow S_c = [a, R_c, R_{X'}, S_a]$$

なる対応関係が成立する。この場合は、スタックシンボルが c から X に変化するまで還元動作が行われ、次にシフト動作が行われる。シフトエラー検出については自明である。故に、補題は明らかである。(証明終)

(補題 2) パーザ π は還元動作、還元エラーの検出及び accept についてパーザ M と等価である。

(証明) 付録参照。(証明終)

〈定理 1〉 アルゴリズム 1 は正当な (valid) 弱順位パーザを構成する。

(証明) 補題 1, 2 より明らかである。(証明終)

5. パーザ π の簡単化と評価

本章では、パーザ π の簡単化について述べる。そして、解析速度、パーザの規模に対する評価を与える。

5.1 パーザ π の簡単化

(手順 3) パーザ π の簡単化

(3-1) 同じ状態 S_a, S_b を求め、状態 $S_{a,b}$ にまとめる。

(3-2) $S_a = [c, R_a, R_{B'}, S_c], S_b = [c, R_b, R_{B'}, S_c]$ なる状態を求め、 $\alpha_p = \beta a, \alpha_q = \gamma b$ に対して $\beta \neq \gamma$ ならば、 $S_{a,b} = [c, R_{a,b}, R_{B'}, S_c]$ とする。

(3-3) $R_X = [A_p, \leftrightarrow A_p, p, R_{A'_p}]$ なる状態を求め、 $R_X = [A_p, b, p, R_{A'_p}]$ とする。

(3-4) RG の $\mu_R(A_q) = \phi$ なるノード A_q に対する状態 $R_{A'_q}$ を $T_R(G)$ より除き、 $R_X = [a, \leftrightarrow A_q, q, R_{A'_q}]$ なる状態を $R_X = [a, \rightarrow, q, R_{A'_q}]$ とする。

(3-5) 手順(1-1)ですべての状態 $R_{B'}$ が一意的に決まる場合、あるいは一意的でなくて $R_{B'} = R_{B'_1} = R_{B'_2}$ とした場合でも、 $\alpha_p = \beta B_1, \alpha_q = \gamma B_2$ に対して $\beta \neq \gamma$ ならば、 $T_R(G)$ よりすべての状態 $R_{B'}$ を除く。但し、この方法では $T_S(G)$ の出口状態 $R_{B'}$ を R_B に変更し、操作法も Fig. 4 の還元動作に入るループを pop up へ入るように変更する。また、手順(3-4)、(3-5)は同時に行わない。

Fig. 2-(a) のノード F が手順(3-4)に相当する。すべての状態 $R_{B'}$ の動作は pop up だけであるから、手順(3-5)により統一して行え、かつ状態数も軽減される。

(補題 3) 手順 3 はパーザ π に対して等価な変換である。

(略証) 手順(3-1)、(3-3)は明らかである。また、手順 3-2 で $\beta \neq \gamma$ ならば、 a, b によらなくてもハンドル α_p, α_q の区別が可能である。手順(3-4)の状態 $R_{A'_q}$ は性質 3 より出口状態とはならない。手順(3-5)で状態 $R_{B'}$ が一意的な場合は明らかである。そうでないときも、 $\beta \neq \gamma$ ならば、手順(3-2)と同様にハンドル α_p, α_q の区別が可能である。(略証終)

〈例 3〉 手順(3-1)、(3-2)、(3-3)、(3-5)により簡単化された $T(G_1)$ を Table 3 に示す。

Table 3 Simplified $T(G)$ for grammar G_1

(a) $T_S(G_1)$

状態名	入力記号	入口状態	出口状態	行先状態
$S_{a,}$	*	$R_{a,}$	R_T	$S_{*,*}(.+)$
)		$R_{E,}$	$S_{a,}$
	+		$R_{E,}$	$S_{*,*}(.+)$
	\$		$R_{E,}$	$S_{*,*}(.+)$
$S_{*,*}(.+)$	a			$S_{a,}$
	($S_{*,*}(.+)$

(b) $T_R(G_1)$

状態名	スタックシンボル	動作	出力	行先状態
$R_{a,}$	#	$\leftarrow F$	6	R_F
	E	\rightarrow		$R_{E,}$
$R_{E,}$	($\leftrightarrow F$	5	R_F
R_T	#	$\leftarrow T$	4	R_T
	*	\rightarrow		R_*
R_*	T		3	R_T
R_T	#	$\leftarrow E$	2	$R_{E,}$
	+	\rightarrow		R_+
R_+	E		1	$R_{E,}$
$R_{E,}$	#	$\leftarrow E_0$	0	$R_{E,}$

5.2 パーザ π の評価

パーザ π は解析速度に対して次の利点がある。

(1) 1回の還元を決定するだけで、連続した還元動作と1回のシフト動作が行える。

(2) $T_R(G)$ により還元動作と還元に適用された生成規則番号の決定がハンドルの長さに比例した時間で行える。

(1)、(2)により、パーザ π の解析速度はパーザ M より速いことが期待される。

パーザ π の規模は、 $T(G)$ の状態数によって決まる。 $T_S(G)$ の状態数は手順 1 より V_T の要素数に等しい。また、 $T_R(G)$ の状態数は生成規則数を u 、生成規則の右辺の長さの平均値を k とすれば、約 $k \cdot u$ となる。ここで、 u は V の要素数に比例して増加し、

* 簡単のために、手順(1-1)の r を $r=2$ とする。

k は定数*であるから、 $T_R(G)$ の状態数も V の要素数の線形のオーダーとなる。行列 M は V の要素数の2乗のオーダーであるから、パーザ π はコンパクト性においてもパーザ M より優れている。

手順3の変換を行えば、パーザ π の解析速度、コンパクト性は更に改善される。

7. むすび

以上、 $T(G)$ による弱順位パーザについて述べた。この方法では、還元動作を記述する $T_R(G)$ を導入したが、 $T_S(G)$ と $T_R(G)$ の能率的な連結方法により $T_S(G)$ の状態数が軽減されるので、全体としてコンパクト性、高速性に優れたパーザが実現可能となった。また、関数による方法と異なり文法のクラス、エラー処理能力を制限する必要もなくなった。

参考文献

- 1) 井上: 右順位文法, 情報処理, Vol. 11, No. 8 p. 449~(1970)
- 2) A. V. Aho and J. D. Ullman: The theory of parsing, translation, and compiling, ch 5, ch 7, Prentice-Hall. (1973)
- 3) 海尻, 妹尾, 打浪, 手塚: 誤り検出遅延による順位関数実現法, 信学論 (D), Vol. J 59-D, No. 11 (1976)
- 4) 青江, 山本, 原田, 島田: 弱順位関数の一構成法, 信学論 (D), Vol. J 60-D, No. 1 (1977)
- 5) 関本: LR (1) 文法の諸性質とそれに基づく parser の構成法, 情報処理, Vol. 13, No. 3, p. 170~ (1973)



付録

(補題2の証明)

連続した還元動作が起こる直前の計算状況を $Q_n = (\$ \beta, \alpha x \$, t)$, $\beta = X_n X_{n-1} \dots X_1$, $X_i \in V_T$, $n \geq 1$ とおくことができる。このとき、性質 1, 3より

$$M(X_1, \alpha) \Rightarrow S_{X_1} = [\alpha, R_{X_1}, R_{B'}, S_e]$$

なる対応関係が成立するので、パーザ π は1回の pop up の後、 $T_R(G)$ の状態 R_{X_1} より還元動作を行う。

(a) $\alpha_p = X_m X_{m-1} \dots X_1 (n \geq m \geq 1)$, $A_p \rightarrow \alpha_p \in P$ の場合、手順 2-1より

$$Q_{n+m} = (\$ X_n X_{n-1} \dots X_{m+1} A_p, \alpha x \$, t p)$$

となる。

(b) $A_p \rightarrow \alpha_p \notin P$ ならば、スタックシンボル X_{r+1}

をもたない状態 R_{X_r} が $T_R(G)$ に必ず存在し ($2 \leq r \leq n-1$)、 Q_{n+r} =還元エラーとなる。

(c) $t_1(\alpha_p) = t_2(\alpha_q) = X_1$ の場合は、 α_p, α_q を手順 (2-2)のようにおくと、

$$Q_{n+m} = (\$ X_n X_{n-1} \dots X_{m+1}, \alpha x \$, t)$$

まで pop up される。そして、

$$R_{X_m} = [Z_1, \rightarrow, \mathbf{b}, R_{Z_1}], R_{X_m} = [Y_1, \rightarrow, \mathbf{b}, R_{Y_1}]$$

であるから、状態 R_{X_m} からは α_p, α_q が独立に還元される。特に、 $k=0$ の場合は、 $Y_1 = \#$ であって

$$R_{X_m} = [\#, \leftrightarrow, A_q, q, R_{A'_q}]$$

となる。従って、パーザ π の還元動作は常に最長のハンドルを適用する。以下、(a), (b)と同様である。

最初の還元動作の後、 $T_R(G)$ の状態が $R_{A'_p}$ になったとする。このとき、 $R_{A'_p} = R_{B'}$ ならば、還元動作は終る。 $R_{A'_p} \neq R_{B'}$ ならば、 $R_{A'_p} = [A_p, \rightarrow, \mathbf{b}, R_{A_p}]$ により1回の pop up が起こり、状態 R_{A_p} より次の還元動作が始まる。そして、 $T_R(G)$ の状態が $R_{B'}$ になるかパーザ π が還元エラーを検出するまで還元動作が続く。

以上のことは、手順(1-1)で状態 $R_{B'}$ が一意的に決った場合ならば、明らかである。次に、状態 $R_{B'}$ が一意的に決定しなかった場合について証明する。この場合が生ずる RG は Fig. 1-(a)か Fig. 5の場合であるが、Fig. 1-(a)の場合は弱順位文法でないので、Fig. 5の場合についてのみ証明する。

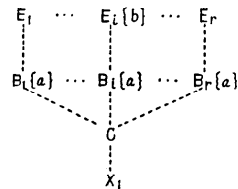


Fig. 5 An illustration of lemma 2

$R_{B'} = R_{B'_1} = R_{B'_2} = \dots R_{B'_r}$ とおくとき ($r \geq 2$),

(d) $S_{X_1} = [a, R_{X_1}, R_{B'}, S_e]$ の場合は、明らかである。

(e) $S_{X_1} = [b, R_{X_1}, R_{E'_i}, S_e]$ の場合は ($1 \leq i \leq r$)、状態 $R_{B'}$ のときのスタックシンボルが B_i となるから $R_{B'} = [B_i, \rightarrow, \mathbf{b}, R_{B_i}]$ が参照され、 $R_{E'_i}$ が出口状態となることは明らかである(手順 (2-3))。故に、出口状態が一意的に決定されない場合でもパーザ π の還元動作は正当である。acceptについては明らかである。以上より、補題は明らかである。

(昭和51年2月14日受付)

(昭和51年8月23日再受付)

* アルゴルの例では、 $k=1.4$ である。