自然言語処理 47－1
（1985. 1. 24）

句構造文法の特性化と構文解析

日高 達 ， 中村 貞吾 ， 吉田 将
（ 九州大学 工学部 ）

1. まえがき

句構造文法の構文解析は、正規文法、LR(k)とLL(k)文法、文脈自由文法については能率的な解析手法が開発されている(1)が、文脈依存文法やタイプ0の文法の解析手法については、文脈依存文法の構文解析法に関する若干の研究がある(2)~(6)ものの、有効な方法は未だ発表されていない。

本研究で提案する標準形句構造文法の構文解析法は、標準形句構造文法と同等なBracket文法の構文解析に還元する手法である。一般に、句構造文法は単純な変換により容易に標準形に変換できるので、標準形という制約は本質的なものではない。本稿では、(1) Bracket文法を定義し、(2) 句構造文法とBracket文法の同等性を示し、(3) 句構造文法の構文解析は対応するBracket文法の構文解析から簡単に還元できることを示し、(4) Bracket文法の構文解析法を与える。

Bracket文法は特殊記号(bracket)を含む文脈自由文法に、bracket記号列の消去関数eを導入した文法で、文脈自由文法の拡張になっている。消去関数eは単純な pushdown transducer と解釈することもできる。

本稿で議論する構文解析法はタイプ0の文法の general parser だから、入力 $w \notin L(G)$ に対しては手順の停止が必ずしも保障されていないことは当然であるが、タイプ1(文脈依存)の文法Gに対しても手順の停止が保障されない欠点をもっている。この欠点を改善した構文解析法については、間もなく発表する予定である。

2. 諸定義

ブラケット文法(Bracket Grammar, BG)は、文脈自由文法に、ブラケットと呼ばれる新しい補助的な記号を導入した形の文法である。以下に、標準形の句構造文法、ブラケット文法、ブラケット言語、標準形の句構造文法とブラケット文法の対応を定義する。

［ 定義 1 ］

標準形の句構造文法 $G = (N, \Sigma, P, S)$ は生成規則が次の(i)、(ii) のいずれかの形をした句構造文法である。ただし、N、Σ、P、S はそれぞれ非終端記号の集合、終端記号の集合、生成規則の集合、開始記号 ($S \in N$) である。

(i) $X \to \alpha$

(ii) $X_1 X_2 \to Y_1 Y_2$

ここで、$X \in N$, $X_1 \in N$, $X_2 \in N$, $Y_1 \in (N \cup \Sigma)$, $Y_2 \in (N \cup \Sigma)$, $\alpha \in (N \cup \Sigma)^*$。

(i) の形式の生成規則を自由規則、(ii) の形式の生成規則を文脈規則と呼ぶ。以後、Gの文脈規則に順番をつけて π_1, π_2, π_3, … と記し、Gの自由規則の集合を P_{CF} 、文脈規則の集合を $P_{CS} = \{ \pi_1, \pi_2, \cdots \}$ と記す。 □

任意の句構造文法は単純な変換により標準形の句構造文法に等価変換できることが一般に知られている。

また、自由規則がε- free ($X \to \varepsilon$ の形の生成規則がない)ならば G は文脈依存文法、また文脈規則をもたないならば G は文脈自由文法である。

［ 定義 2 ］

BG は6項組 $G = (N, \Sigma, B_L, B_R, P, S)$ で表される。ここで、

(1) N: 非終端記号の有限集合

(2) Σ: 終端記号の有限集合で、$N \cap \Sigma = \phi$ (空集合)。

(3) B_L : 左括弧(Left bracket) の可付番集合。

$B_L = \{ [_1, [_2, [_3, \cdots \}$.

B_R : 右括弧(Right bracket) の可付番集合。

$B_R = \{]_1,]_2,]_3, \cdots \}$.

B_L と B_R の和集合を B で表す。

添字の等しい左括弧と右括弧、すなわち、$[_i$ と $]_i$ は互に対をなすと言う。また、$[_i$ と $]_i$ の組を単に対と呼ぶ。

(4) P : 生成規則の有限集合で、P の各要素は、次の3つのうちのいずれかの形をしている。

(1) $X \to \alpha$

(2) $X \to Y[_i$

(3) $X \to]_i Y$

ここで、$X \in N$, $Y \in (N \cup \Sigma)$.

$\alpha \in (N \cup \Sigma)^*$

(i) の形の規則を自由規則(CF規則)、(ii) の形の規則を左括弧規則(LB規則)、(iii) の形の規則を右括

弧規則(RB規則)と呼ぶ。また、LB規則とRB規則をまとめて括弧規則(B規則)と呼ぶ。そして、CF規則、LB規則、RB規則の集合をそれぞれP_{CF}, P_{LB}, P_{RB} で表す。

$$P = P_{CF} \cup P_{LB} \cup P_{RB}.$$

(5) S : Nの要素で開始記号 □

BGにおける導出($\xi \overset{*}{\underset{(G)}{\Rightarrow}} \eta$)は、通常の文脈自由文法の場合と全く同様に定義されるものとする。

[定義 3]

整った形をした括弧の系列を、整列(Well formed brackets) と呼ぶ。整列は、次のように帰納的に定義される。

(1) 空系列εは整列である。

(2) αが整列であるならば、$[_i\ \alpha\ ]_i$ も整列である。

(3) α、βが整列であるばらば、αβも整列である。

(4) (1) ~ (3) の規則を有限回用いて作られる系列のみが整列である。

また、整列の中で規則(3)を用いずに作られるものを、特に単純整列(Simple well formed brackets) と言う。

αが整列、および、単純整列であることを、それぞれWFB(α), SB(α)と書く。 □

[定義 4]

BG を、G = (N, Σ , B_L, B_R, P, S)であるとする。$(\Sigma \cup B)^*$の系列 w が次のように書けるとき、w は整語(Well formed word) であると言い WFW(w) と書く。

$$w = b_0\ a_1\ b_1\ a_2\ b_2 \cdots b_{n-1}\ a_n\ b_n$$

ここで、$a_j \in \Sigma \quad j = 1, \cdots, n,$

$WFB(b_j) \quad j = 0, \cdots, n$

また、 整語 w が次の条件を満たすとき、特に単純整語(Simple well formed word) と言い、SW(w)と書く。

$SB(b_j) \quad j = 1, \cdots, n-1$

$b_0 = b_n = \varepsilon$ □

w が整語であるということは、w の中のΣの要素に注目したときに、各々のΣの要素の間および両端に現れる括弧の列は整列となっていることを意味する。ここで、整列の定義より、空系列εもまた整列であるということに注意する。

また、w が単純整語であるということは、w の中のΣの要素に注目したときに、各々のΣの要素の間に現れる括弧の列は単純整列であり、両端には括弧列は現れないということを意味する。

[例 1]

$\Sigma = \{ a_1, a_2, a_3 \}$

$b_1 = [_3\ [_1\ [_4\ 4\ ]_4\ 1\ ]_1\ [_2\ 2\ ]_2\ 3\ ]_3\ [_4\ 4\ ]_4$: 整列

$b_2 = [_2\ 2\ ]_2\ [_1\ [_1\ 1\ ]_1\ [_3\ 3\ ]_3\ 1\ ]_1$: 整列

$b_3 = [_4\ [_1\ [_2\ [_3\ 3\ ]_3\ 2\ ]_2\ 1\ ]_1\ 4\ ]_4$: 単純整列

$b_1\ a_1\ a_2\ b_3\ a_3\ b_2$: 整語

$a_1\ b_3\ a_2\ a_3$: 単純整語

[定義 5]

整列を消去する関数 e は、次のような $(N \cup \Sigma \cup B)^*$ から$(N \cup \Sigma)^*$への mapping である。

$$w = b_0\ c_1\ b_1\ c_2\ b_2 \cdots c_n\ b_n$$

$$\left.\begin{array}{l} c_j \in N \cup \Sigma \quad j = 1, \cdots, n \\ b_j \in B^* \quad j = 0, \cdots, n \end{array}\right\} \text{に対し}$$

$$e(W) \overset{\text{def}}{=} \begin{cases} c_1 c_2 \cdots c_n \cdots WFB(b_j) & j = 0, \cdots, n\text{の場合} \\ \phi\ (\text{定義されない}) \cdots \text{他の場合} \end{cases}$$ □

関数 e は一種の pushdown transducer である。

[定義 6]

BG G = (N, Σ, B_L, B_R, P, S)によって生成される言語 $L_B(G)$ を次のように定義する。

$L_B(G) = \{ e(w) \mid S \overset{*}{\underset{(G)}{\Rightarrow}} w\ ,\ WFW(w) \}.$

$L_B(G)$をブラケット言語(Bracket Language, BL)と呼ぶ。 □

[定義 7]

標準形句構造文法 G = (N, Σ, $P_{CF} \cup P_{CS}$, S),

BG G'=(N, Σ, B_L, B_R, $P_{CF} \cup P_{LB} \cup P_{RB}$, S)

において、

$P_{CS} = \{ XX' \rightarrow YY' \mid X \rightarrow Y[_i \in P_{LB},\ X' \rightarrow {}_i]Y' \in P_{RB} \}$

であるとき、GとG'は互に対応するという。 □

[例 2]

$P_{CS} = \{ BA \rightarrow AB,\ BD \rightarrow Da_2,\ AD \rightarrow CE,\ AC \rightarrow Ca_1 \},$

$P_{LB} = \{ A \rightarrow C[_1,\ B \rightarrow A[_2 \mid D[_3 \},$

$P_{RB} = \{ A \rightarrow {}_2]B,\ C \rightarrow {}_1]a_1,\ D \rightarrow {}_1]E \mid {}_3]a_2 \}.$

このとき、標準形句構造文法 G = (N, Σ, $P_{CF} \cup P_{CS}$, S)と BG G' = (N, Σ, B_L, B_R, $P_{CF} \cup P_{LB} \cup P_{RB}$, S) は互に対応する。

任意に与えられた標準形句構造文法 G に対し、G に対応する BG を容易に構成することができ、また逆の操作も容易である。

4章において、対応する標準形句構造文法 G とBG G' に対し、$L(G) = L_B(G')$ であること、および Gの構文解析と G' の構文解析が同等であることが示される。

3. ブラケット文法の性質

ここでは、BG, $G = (N, \Sigma, B_L, B_R, P, S)$ による導出における性質を示す。

以下では、$\gamma \in (N \cup \Sigma)^*$ とする。また、CF規則、 LB規則、RB規則、B規則だけを用いた導出を、それぞれ $\xRightarrow[CF]{}$, $\xRightarrow[LB]{}$, $\xRightarrow[RB]{}$, $\xRightarrow[B]{}$ で表し、一般的な導出 $\Longrightarrow$ と区別する。

[補題 1]

整語wの導出 $\gamma \overset{*}{\Longrightarrow} w$ に対して、次のことが成り立つ。

(i) 導出中のどのステップにおいても、左括弧の右隣にΣの要素が現れることはない。

(ii) 最左導出においてRB規則が適用されたステップでは、そのRB規則によって生成された右括弧よりも左側の部分は、すべてΣ ∪ Bの要素になっている。

(iii) 最左導出において最初に用いられるB規則はLB規則である。

(証明) (i), (ii) は、w が整語であることおよび最左導出の定義より自明。また、(iii) は、(ii) より明らかである。 □

[定義 8]

整語w の導出 $\gamma \overset{*}{\Longrightarrow} w$ において、w 内にある各々の括弧に対して、最左導出によって生成された順番に従って昇順に番号を付ける。このとき、次の2つの条件を満たすような左括弧 $[_i$ を優先左文脈と呼ぶ。

(i) $[_i$ の右側に現れる括弧は番号がすべて $[_i$ より大きい。

(ii) $[_i$ の右側に現れる括弧の中で最も番号が小さい括弧が $[_i$ の右隣にあり、かつそれが $[_i$ と互に対をなす右括弧 $_i]$ である。 □

[補題 2]

整語wの導出 $\gamma \overset{*}{\Longrightarrow} w$ があるとすると、w の中には優先左文脈が少なくとも1つは存在する。

(証明) 補題 1-(ii) より、右括弧については、左側にあるものの方が右側にあるものよりも番号が小さい。したがって、w の中で最も右側にある左括弧は、優先左文脈である。 □

[定理 1]

整語 wの 導出 $\gamma \overset{*}{\Longrightarrow} w$ に対して次のような導出が存在する。

$$\gamma \overset{*}{\underset{CF}{\Longrightarrow}} \alpha\, X\, Y\, \beta \overset{2}{\underset{B}{\Longrightarrow}} \alpha\, X'\, [_k\, _k]\, Y'\, \beta$$

$$\overset{*}{\Longrightarrow} w_1\, [_k\, _k]\, w_2 = w$$

ここで、$\alpha, \beta \in (N \cup \Sigma)^*$, $X, Y \in N$,

$X', Y' \in (N \cup \Sigma)$,

$X \to X'\, [_k \in P_{LB}$, $Y \to _k]\, Y' \in P_{RB}$,

かつ、$[_k$ は導出 $\gamma \overset{*}{\Longrightarrow} w$ における優先左文脈の中で最も左側の優先左文脈とする。

(証明) $\gamma \overset{*}{\Longrightarrow} w$ の最左導出を考える。補題 1-(iii) より、初めて現れる括弧は左括弧である。したがって、補題 1-(i) より、最左導出の過程を次のように書くことができる。

$$\gamma \overset{*}{\underset{CF}{\Longrightarrow}} \alpha_1\, X_1\, \beta_1 \xRightarrow[LB]{} \alpha_1\, X_1'\, [_{i_1}\, \beta_1$$

$$\overset{*}{\Longrightarrow} \alpha_1\, t\,(X_1')\, [_{i_1}\, \beta_1$$

$$\beta_{j-1} \overset{*}{\underset{CF}{\Longrightarrow}} X_j\, \beta_j \xRightarrow[LB]{} X_j'\, [_{i_j}\, \beta_j$$

$$\overset{*}{\Longrightarrow} t\,(X_j')\, [_{i_j}\, \beta_j$$

ただし、$j = 2, 3, \cdots, m$

$$\beta_m \overset{*}{\underset{CF}{\Longrightarrow}} X_{m+1}\, \beta_{m+1} \xRightarrow[RB]{} {}_{i_{m+1}}]\, X'_{m+1}\, \beta_{m+1}$$

$$\overset{*}{\Longrightarrow} {}_{i_{m+1}}]\, t\,(X'_{m+1})\, \beta_{m+1}$$

$$\beta_{m+1} \overset{*}{\Longrightarrow} w'$$

ここで、$\alpha_1 \in \Sigma^*$, $w' \in (\Sigma \cup B)^*$,

$X_j \in N$, $X_j' \in (N \cup \Sigma)$,

$\beta_j \in (N \cup \Sigma)^*$

$j = 1, 2, \cdots, m+1$

$X_j \to X_j'\, [_{i_j} \in P_{LB}$

$j = 1, 2, \cdots, m$

$X_{m+1} \to {}_{i_{m+1}}]\, X_{m+1}' \in P_{RB}$

また、t(X) は、Xから導出された $(\Sigma \cup B)^*$ の系列を表す。

補題 2 より導出 $\gamma \overset{*}{\Longrightarrow} w$ における優先左文脈が存在するが、上の $[_{i_m}$ は最も左側に位置する優先左文脈である。さらに w は整語だから、$[_{i_m}$ と $_{i_{m+1}}]$ が対をなす($i_m = i_{m+1}$) ことは明らかである。

さて、この導出において、各 X_j (j = 1, 2, … , m-1) からの導出 $X_j \overset{*}{\Longrightarrow} t(X'_j)\ [_{i_j}$ を後回しにした導出は次のようになる。

$$\gamma \overset{*}{\underset{cF}{\Longrightarrow}} \alpha_1 X_1 \beta_1 \overset{*}{\underset{cF}{\Longrightarrow}} \alpha_1 X_1 X_2 \beta_2 \overset{*}{\underset{cF}{\Longrightarrow}} \cdots$$

$$\overset{*}{\underset{cF}{\Longrightarrow}} \alpha_1 X_1 X_2 \cdots X_{m-1} X_m X_{m+1} \beta_{m+1}$$

$$\overset{2}{\underset{B}{\Longrightarrow}} \alpha_1 X_1 X_2 \cdots X_{m-1} X_m' [_{i_m}\ _{i_{m+1}}] X_{m+1}' \beta_{m+1}$$

$$\overset{*}{\Longrightarrow} \alpha_1\ t(X_1')\ [_{i_1} \cdots t(X_{m-1}')\ [_{i_{m-1}}$$

$$t(X_m')\ [_{i_m}\ _{i_{m+1}}]\ t(X_{m+1}')\ w'$$

この導出は最左導出ではないが、確かに γ から w の導出になっている。ゆえに、

$\alpha = \alpha_1 X_1 X_2 \cdots X_{m-1}$, $\beta = \beta_{m+1}$
$X = X_m$, $Y = X_{m+1}$,
$k = i_m$ ($= i_{m+1}$)

とおくと、本定理となる。 □

定理 1 は、$\gamma \overset{*}{\Longrightarrow} w$ の導出の順序を適当に入れ替えることによって、最初に適用される LB 規則と RB 規則が連続した2ステップで適用され、かつ、それによって互に対をなす隣接した左右の括弧が生成されるような導出に構成できることを示している。

次に、ε- free BG の性質を示す。

[補題 3]

G は、ε - free BG であるとする。このとき、$\gamma \in (N \cup \Sigma)^*$, $w \in (\Sigma \cup B)^*$, $\gamma \overset{*}{\Longrightarrow} w$ であるような w に対して次のことが成り立つ。

(i) w の中の任意の左括弧 $[_i$ に対して $w = w_1 [_i w_2$ としたとき、$w_1 \neq \varepsilon$。

(ii) w の中の任意の右括弧 $_i]$ に対して $w = w_1\ _i] w_2$ としたとき、$w_2 \neq \varepsilon$。

(iii) w の中の任意の右括弧 $_i]$ とその右側に現れる任意の左括弧 $[_i$ に対して $w = w_1\ _i] w_2 [_j w_3$ としたとき、$w_2 \neq \varepsilon$。

(証明) ε - free であることと LB 規則 ($X \to Y [_i$) RB 規則 ($X \to\ _i] Y$) の形より、(i) (ii) は自明である。いま、$w = w_1\ _i] [_j w_3$ と仮定すると、次のような最左導出が存在することになる。

$\gamma \overset{*}{\Longrightarrow} w_1 X \alpha \Longrightarrow w_1\ _i] Y \alpha \overset{*}{\Longrightarrow} w_1\ _i] [_j w_3$

ただし、$X \to\ _i] Y \in P_{RB}$.

G は ε - free だから、$w' \in (\Sigma \cup B)^*$ が存在し

$$Y \overset{*}{\Longrightarrow} [_j w' \qquad \cdots (a)$$

RB 規則の定義より、$Y \in N \cup \Sigma$ であるが、本補題の(i) より $Y \in N$ の場合には(a) は成立しないし、$Y \in \Sigma$ の場合にも勿論 (a) は成立しない。 □

[定理 2]

G が ε - free BG であるとき、

$S \overset{*}{\Longrightarrow} w$, WFW(w) ならば SW(w) である。

(証明) $w = b_0 a_1 b_1 a_2 \cdots a_n b_n$

とすると、整語の定義と補題 3-(i), (ii) より、$b_0 = b_n = \varepsilon$ である。また、補題 3-(iii) より $SB(b_j)$ (ここで、j = 1, … , n-1) である。したがって、WFW(w) であるならば、SW(w) が成り立つ。 □

[例 3]

ε - free BG , $G = (N, \Sigma, B_L, B_R, P, S)$ を次のようにする。

$N = \{ S, A, B, C, D, E \}$
$\Sigma = \{ x, y, z \}$
$P_{CF} = \{ S \to ABSz , S \to ADz , C \to x , E \to y \}$
$P_{LB} = \{ B \to A [_1 , B \to D [_2 , A \to C [_3 , A \to C [_4 \}$
$P_{RB} = \{ A \to\ _1] B , D \to\ _2] y , D \to\ _3] E , C \to\ _4] x \}$

G によって生成される BL は、

$$L_B(G) = \{ x^n y^n z^n \mid n \geq 1 \}$$

BG, G における導出の一例を示す。

$S \Longrightarrow A B S z$
$\Longrightarrow C [_4 B S z$
$\Longrightarrow x [_4 B S z$
$\Longrightarrow x [_4 A [_1 S z$
$\Longrightarrow x [_4 C [_3 [_1 S z$
$\Longrightarrow x [_4\ _4] x [_3 [_1 S z$
$\Longrightarrow x [_4\ _4] x [_3 [_1 A D z z$
$\Longrightarrow x [_4\ _4] x [_3 [_1\ _1] B D z z$
$\Longrightarrow x [_4\ _4] x [_3 [_1\ _1] D [_2 D z z$
$\Longrightarrow x [_4\ _4] x [_3 [_1\ _1]\ _3] E [_2 D z z$
$\Longrightarrow x [_4\ _4] x [_3 [_1\ _1]\ _3] y [_2 D z z$
$\Longrightarrow x [_4\ _4] x [_3 [_1\ _1]\ _3] y [_2\ _2] y z z$

$w = x\ [_4\ 4]\ x\ [_3\ [_1\ 1]\ 3]\ y\ [_2\ 2]\ y\ z\ z$

$SW(w)$

$e(w) = x\ x\ y\ y\ z\ z \in LB(G)$

4. 句構造文法と BG の言語同等性

本節では標準形句構造文法 G とそれに対応する BG G' が言語的に同等 ($L(G) = L_B(G')$) であるばかりでなく、一方の文法での導出から他方の文法での導出が容易に構成できることを示す。以上のことは、句構造文法の構文解析を対応する Bracket 文法の構文解析に置き換えることができることを意味する。

[定理 3]

標準形句構造文法 $G = (N, \Sigma, P_{CF} \cup P_{CS}, S)$ とそれに対応する BG $G' = (N, \Sigma, B_L, B_R, P_{CF} \cup P_{LB} \cup P_{RB}, S)$ に対し、$L(G) = L_B(G')$ であり、一方の文法での導出から他方の文法での導出を構成できる。

(証明) $w \in L(G)$ の G での導出 $\gamma_0 (=S) \xrightarrow[(G)]{} \gamma_1 \xrightarrow[(G)]{} \gamma_2 \cdots \xrightarrow[(G)]{} \gamma_m (=w)$ に対し、次のような G' での導出が存在する。すなわち、G の CF 規則の適用には G' における同じ CF 規則の適用を対応させ、G の CS 規則 $X\ X' \to Y\ Y' \in P_{CS}$ の適用には G' の LB 規則 $X \to Y\ [_i \in P_{LB}$ および RB 規則 $X' \to {}_i]\ Y'$ の連続適用を対応させる(このような i は必ず存在する)。

$\gamma_\ell = c_{\ell 1}\ c_{\ell 2} \cdots c_{\ell m_\ell}$

$(c_{\ell j} \in N \cup \Sigma\ ,\ j=1, \cdots, m_\ell\ ,\ \ell=0, \cdots, m)$

とすると、整列 $b_{\ell j}$ $(j= 0, \cdots, m_\ell)$ が存在し、$\gamma_0 \xrightarrow[(G)]{} \gamma_1 \xrightarrow[(G)]{} \cdots \xrightarrow[(G)]{} \gamma_m$ には、GB での導出 $\gamma_0' \xrightarrow[(G')]{+} \gamma_0' \xrightarrow[(G')]{+} \cdots \xrightarrow[(G')]{+} \gamma_m'$ が対応する。ただし、

$\gamma_\ell' = b_{\ell 0}\ c_{\ell 1}\ b_{\ell 1} \cdots c_{\ell m_\ell}\ b_{\ell m_\ell}$

$(\ell = 0, \cdots, m).$

以上のことから、次のことが分る。

$L(G) \subseteq L_B(G').$

次に、整語 w の G' による導出 $\gamma \xrightarrow[(G')]{*} w$ が与えられたとき、これを G による e(w) の導出 $\gamma \xrightarrow[(G)]{*} e(w)$ に変換できることを示す。ただし、$\gamma \in (N \cup \Sigma)^*$ である。

定理 1 より、$\gamma \xrightarrow[(G')]{*} w$ は次のような導出に変換できる。

$\gamma \xrightarrow[CF]{*} \alpha\ X\ X'\ \beta \xrightarrow[B]{2} \alpha\ Y\ [_j\ j]\ Y'\ \beta \xrightarrow[(G')]{*} w_1\ [_j\ j]\ w_2 = w$

ただし、$\alpha, \beta \in (N \cup \Sigma)^*$, $X \to Y\ [_j \in P_{LB}$, $X' \to {}_j]\ Y' \in P_{RB}$.

定義より、$X\ X' \to Y\ Y' \in P_{CS}$ だから、G による導出

$\gamma \xrightarrow[CF]{*} \alpha\ X\ X'\ \beta \xrightarrow[CS]{} \alpha\ Y\ Y'\ \beta$

が存在し、かつ

$\alpha\ Y\ Y'\ \beta \xrightarrow[(G')]{*} w_1\ w_2$

となる。w は整語だから、$w_1\ w_2$ は w より bracket 記号が少ない整語である。導出 $\alpha\ Y\ Y'\ \beta \xrightarrow[(G')]{*} w_1\ w_2$ に同様の操作を繰返すことにより、G' による導出 $\gamma \xrightarrow[(G')]{*} w$ は G による導出 $\gamma \xrightarrow[(G)]{*} e(w)$ に完全に変換することができる。以上のことから、次のことが分る。

$L_B(G') \subseteq L(G).$

$\therefore\ L_B(G') = L(G).$ □

5. Bracket 文法の構文解析法

Bracket 文法は、生成規則は文脈自由文法のそれと同じ形体であるが、整語から整列を除去する pushdown transducer e によって強化されている。したがって、入力 $e(w) = a_1\ a_2 \cdots a_n \in \Sigma^*$ の BG G による構文解析では、e によって除去される整列 $b_0, b_1, \cdots b_n$ を解析の過程で補完しながら、導出 $S \xrightarrow[(G)]{*} b_0\ a_1\ b_1 \cdots a_n\ b_n$ を求める必要がある。

本節で与える構文解析法は、文脈自由文法の最も高速な構文解析法である Earley の方法に pushdown transducer e の機能を組み込んで強化した手法である。本手法は、Earley の方法と n 長さの入力 $a_1\ a_2 \cdots a_n \in \Sigma^*$ とBG G よりパーズリスト PL を作成する PL 作成アルゴリズムと PL から $G = (N, \Sigma, B_L, B_R, P, (= P_{CF} \cup P_{LB} \cup P_{RB}), S)$ における最左導出を抽出する構文構造抽出アルゴリズムから成るが、本手法の本質はパーズリスト PL にあり、構文構造抽出アルゴリズムは PL に格納される情報 (item) の意味さえ理解されれば Earley の方法から自然に推察されると思われるので、本節では主として PL に格納される項目 (item) の意味を述べる。

[定義 9]

パーズリスト PL は次のような形式の item の集合である。

$< X \to \alpha \cdot \beta,\ i,\ j,\ L_i,\ L_j >$

ただし、$X \to \alpha\beta \in P$, $0 \leq i \leq j \leq n$, $L_i \in B_L^*$, $L_j \in B_L^*$.

$< X \to \alpha \cdot \beta,\ i,\ j,\ L_i,\ L_j > \in PL$ であることと、次の i、ii が成立することは同等である。ただし、$L_i = [_{i_1}\ [_{i_2} \cdots [_{i_p}$, $L_j = [_{j_1}\ [_{j_2} \cdots [_{j_q}$ とす

る。

$e(w_i) = a_1 \cdots a_i$, $e(w_j) = a_1 \cdots a_j$ であるような整語 w_i, w_j と整列 $b_{i_1}, \cdots, b_{i_p}, b_{j_1}, \cdots, b_{j_q}$ および $\gamma \in (N \cup \Sigma \cup B)^*$ が存在し、

i $S \overset{*}{\underset{(G)}{\Longrightarrow}} w_i [_{i_1} b_{i_1} \cdots [_{i_p} b_{i_p} X \gamma$

ii $w_i [_{i_1} b_{i_1} \cdots [_{i_p} b_{i_p} \alpha \overset{*}{\underset{(G)}{\Longrightarrow}} w_j [_{j_1} b_{j_1} \cdots [_{j_q} b_{j_q}$ □

$\langle S \to \alpha\cdot, 0, n, \varepsilon, \varepsilon \rangle$ の形式の item が PL に少なくとも1つ存在することと、$a_1 \cdots a_n \in L_B(G)$ であることは同等である。

次に PL の作成アルゴリズムを与える。

[PL 作成アルゴリズム]

入力 : $w = a_1 a_2 \cdots a_n \in \Sigma^*$

STEP 1. $S \to \alpha \in P$ ならば、$\langle S \to \cdot\alpha, 0, 0, \varepsilon, \varepsilon \rangle$ を PL に加える。

STEP 2. PL に新しい item が加えられなくなるまで i) ~ iv) を繰返す。

i) PL に登録された item $\langle X \to \alpha\cdot Y\beta, i, j, L_i, L_j \rangle$ と $Y \to \gamma \in P$ に対して、$\langle Y \to \cdot\gamma, j, j, L_j, L_j \rangle$ を PL に加える。

ii) PL に登録された2つの item
$\langle X \to \alpha\cdot Y\beta, i, j, L_i, L_j \rangle$
$\langle Y \to \gamma\cdot, j, k, L_j, L_k \rangle$ に対して、
$\langle X \to \alpha Y\cdot\beta, i, k, L_i, L_k \rangle$ を PL に加える。

iii) PL に登録された item
$\langle X \to \cdot_j]Y, i, i, L[_j, L[_j \rangle$ に対し、
$\langle X \to {}_j]\cdot Y, i, i, L[_j, L \rangle$ を PL に加える。

iv) PL に登録された item
$\langle X \to \alpha\cdot a\beta, i, j, L_i, \varepsilon \rangle$ に対し、
$j < n$ かつ $a = a_{j+1}$ ならば
$\langle X \to \alpha a\cdot\beta, i, j+1, L_i, \varepsilon \rangle$ を PL に加える。

STEP 3. $\langle S \to \alpha\cdot, 0, n, \varepsilon, \varepsilon \rangle$ の形の item が PL に登録されていれば、PL を出力して停止する。

STEP 4. $\langle X \to Y\cdot[_k, i, j, L_i, L_j \rangle$ の形の item が PL に登録されていれば、
$\langle X \to Y[_k\cdot, i, j, L_i, L_j[_k \rangle$ を PL に加える。新しい item が加えられれば STEP 2 へ戻り、そうでなければ reject を出力して停止する。 □

PL 作成アルゴリズムで作られる PL は定義 9 で与えられたパーズリストの部分集合になるが、STEP 3で正常に停止した場合、PL は B 規則を最小回数使った導出に関する情報を全て保有している。

PL 作成アルゴリズムが PL を出力して正常停止した場合、PL 上の表操作により最左導出を抽出することになる。

次の抽出アルゴリズムに現れる R は PL から最左導出への mapping で、定義 9 のような item $\langle X \to \alpha\cdot\beta, i, j, L_i, L_j \rangle \in$ PL に対し、$R(\langle X \to \alpha\cdot\beta, i, j, L_i, L_j \rangle)$ は定義 9 の ii の最左導出である。最左導出は適用される生成規則を適用の順に従って左から右方向へ連鎖して表すものとし、連鎖の記号を $*$ とする。

[構文構造抽出アルゴリズム]

item $\langle S \to \delta\cdot, 0, n, \varepsilon, \varepsilon \rangle \in$ PL を PL から検索し、最左導出 $(S \to \delta) * R(\langle S \to \delta\cdot, 0, n, \varepsilon, \varepsilon \rangle)$ を出力する。ここで、定義 9 のような item $x = \langle X \to \alpha\cdot\beta, i, j, L_i, L_j \rangle \in$ PL に対し、R(x) は次の i ~ v のように recursive に定義される。

i $\alpha = \varepsilon$ ならば、$R(x) \overset{\text{def}}{=} \varepsilon$.

ii $\alpha = \alpha' a_j$ ならば、$R(x) \overset{\text{def}}{=} R(\langle X \to \alpha'\cdot a_j\beta, i, j-1, L_i, \varepsilon \rangle)$.

iii $\alpha = Y[_{j_q}$ ならば、$R(x) \overset{\text{def}}{=} R(\langle X \to Y\cdot[_{j_q}, i, j, L_i, [_{j_1} [_{j_2} \cdots [_{j_{q-1}} \rangle)$.

iv $\alpha = {}_k]$ ならば、$R(x) \overset{\text{def}}{=} R(\langle X \to \cdot_k]\beta, i, i, L_i, L_i \rangle)$.

v $\alpha = \alpha' X'$, $X' \in N$ ならば、$\langle X' \to \gamma\cdot, k, j, L_k, L_j \rangle \in$ PL であるような $X' \to \gamma \in P$, 整数 k および $L_k \in B_L^*$ が存在する。このような item を PL から検索し、
$R(x) \overset{\text{def}}{=} R(\langle X \to \alpha'\cdot X'\beta, i, k, L_i, L_k \rangle) * (X' \to \gamma) * R(\langle X' \to \gamma\cdot, k, j, L_k, L_j \rangle)$.
□

[定義 10]

$X \in N$ および整列 b, b' が存在し、

$X \overset{+}{\underset{(G)}{\Longrightarrow}} bXb'$

であるとき、BG G は cycle をもつといい、そうでなければ BG G は cycle free であるという。 □

BG G に対応する句構造文法 G' に対し、
$X \overset{+}{\underset{(G)}{\Longrightarrow}} bXb'$ であることと $X \overset{+}{\underset{(G)}{\Longrightarrow}} X$ であることは同等である。

抽出アルゴリズムは、BG G が cycle free ならば必ず停止することが証明できるが、本稿では省略する。

6. あとがき

本稿では、句構造文法の構文解析を、対応する Bracket 文法の構文解析に還元して行う手法を示した。要点を概略すると、先ず (1) Bracket 文法を定義し、(2) 句構造文法とそれに対応する Bracket 文法の同等性を示し、(3) 句構造文法の構文解析は対応する Bracket 文法の構文解析から簡単な変換により求められることを示し、(4) Bracket 文法の構文解析法を与えた。

Bracket 文法は文脈自由文法の拡張で、その構文解析には文脈自由文法の構文解析手法を応用し易い。本稿の構文解析法は、文脈自由文法の構文解析法である Earley の方法の拡張になっている。

参考文献

(1) Aho,A.V.,Ullman,J.D.:"The Theory of Parsing, Translation,and Compiling,Vol.1:Parsing", Printice-Hall(1972)

(2) Revesz,G.:"Unilateral context sensitive grammars and left-to-right parsing",J. Comput.& Syst.Sci.,5,4,pp.337-352(1971)

(3) 田中, 桜本, 笠井:"文脈言語の下降構文解析法", 信学論(D),J62-D,2,pp.97-103,(1979-02)

(4) 田中:"文脈言語の上昇構文解析法",信学論(D), J63-D,6,pp.454-460,(1980-06)

(5) 鎌田, 岡本:"右側文脈言語の上昇構文解析方法", 信学論(D),J66-D,7,pp.857-863,(1983-07)

(6) 岡本, 鎌田:"右側文脈言語の下降構文解析方法", 信学論(D),J66-D,10,pp.1137-1144(1983-10)