

# 文脈自由型誤り訂正文法によるパターン認識について

森田孝一郎\*・金岡泰保\*\*・富田真吾\*\*・岡田敏彦\*\*

Pattern Recognition by Context-Free Grammar with Error-Correcting

Koichiro MORITA, Taiho KANAOKA, Shingo TOMITA  
and Toshihiko OKADA

## Abstract

In this paper, the validity of syntactic pattern recognition by context-free grammar with error-correcting is made sure from a theoretical point and results on a simulation for hand-written Katakana characters.

Firstly, we define three types of syntax errors to debase strings and propose a distance between one syntactic pattern (a string) and a group of syntactic patterns (a set of strings) showing the smallest number of errors to parse. Secondly, we introduce an algorithm on the error-correcting parser for context-free grammar to find the distance, and apply the algorithm to the pattern recognition of hand-written characters. Finally, we show the results on a simulation for hand-written Katakana characters.

## 1. ま え が き

構文的なパターン認識は、与えられたパターンを基本要素上の一次元記号列として表現し、この記号列があらかじめ推論されたパターン生成文法の生成規則に従っているか否かによって認識を行う方法であり、従来の統計的な手法とは趣を異にしている。

近年、木文法、文脈自由型文法および確率文法等のいわゆる形式文法を用いた構文的なパターン認識に関する理論的、実験的な興味ある報告<sup>1-4)</sup>がなされており手書き文字への文脈自由型文法、木文法の有用性が手書きアルファベットの実験結果と共に示されている。しかしながら、これらは i) 入力文字の細線化, ii) 基本要素上の列としてのストリング化, iii) 文法推論, iv) 誤り訂正文法の構成, v) 誤り訂正パーズング, といった一連の認識システムに関わる具体的方法論の明確さに欠け、理論および実験に対する体系的な議論も十分とは言い難い。

本論文の目的は、構文的パターン認識機構の有用性を、文脈自由型誤り訂正文法を用いて、理論的な立場からと、手書き片仮名文字に対する実験結果を基に検討することである。

2. では、準備として形式言語に関する種々の定義を与え、認識を行う際に重要な役割を演ずる変換および距離を定義しこれらに関する若干の命題を示す。3. では、本論文で議論の対象となる文脈自由型誤り訂正文法を定義する。4. では、文脈自由型誤り訂正文法を用いた構文解析のための誤り訂正パーズングについて述べ、そのアルゴリズムを示す。5. では、認識システムの概要を述べ、6. では、手書き片仮名文字を用いた入力文字の細線化、ストリング化、文法推論等の認識実験システムについて述べる。7. では、そのシミュレーション結果について検討する。

本論文で、特に定義されることなしに用いられる諸記法、諸用語等については文献5) に従うものとする。

## 2. 準 備

本節では形式文法とこれに伴う種々の定義を行い、それらに関する若干の命題を示す。

〔定義1〕形式文法  $G$  を次で定義する。

$$G \triangleq (V_N, V_T, P, S)$$

ここで、 $V_N, V_T$  はそれぞれ非終端記号および終端記号の有限集合を表わし、 $V_N \cap V_T = \phi$  である。 $V_N^*$  上の元をストリングという ( $U^*$  は空語  $\lambda$  を含む  $U$  上の

\* 大学院電子工学専攻 (現、日本電気株式会社)

\*\* 電子工学科

全ての記号系列の集合を表わす). ストリング  $\omega \in V_T^*$  を構成する記号の数をストリングの長さといひ  $|\omega|$  で表わす (例えば,  $|\text{aba}|=3$ ). ただし, 空語  $\lambda$  に対して  $|\lambda|=0$  である. 非終端記号はストリングの生成の中間の過程にのみ現れるもので, 終端記号は生成されたストリングを形成するものである.  $P$  は次のような生成規則の有限集合である.

$$\alpha \rightarrow \beta \quad \text{ここで } \alpha \in (V_N \cup V_T)^* V_N (V_N \cup V_T)^*, \\ \beta \in (V_N \cup V_T)^*$$

$S$  は初期記号を表わし, それから出発してストリングが生成される特殊な非終端記号である.

次に, 形式文法  $G$  によって生成されるランゲージ  $L(G)$  の定義を与える.

[定義2] ランゲージ  $L(G)$  を次で定義する.

$$L(G) \triangleq \{\omega \mid \omega \in V_T^*, S \xrightarrow{*} \omega\}$$

ここで,  $S \xrightarrow{*} \omega$  は文法  $G$  の生成規則を何回か適用することによって  $\omega$  が生成されることを示す.

次に, 本論文で議論の対象となる文脈自由型文法の定義を与える.

[定義3]  $P$  に属する各生成規則が,

$$A \rightarrow \beta \quad (A \in V_N, \beta \in (V_N \cup V_T)^*)$$

なる形の時,  $G$  を文脈自由型文法という.

形式文法を用いた構文的パターン認識においては, 代入, 削除, 挿入なる変換が重要な役割を演ずるが, 次にこれらの定義を与える.

[定義4]  $V_T = \{a_1, a_2, \dots, a_n\}$  とする.

(4.1) 代入変換  $T_S$ : 各  $a_k$  ( $1 \leq k \leq n$ ) に対して

$$T_S(a_k) \in V_T - \{a_k\}$$

(4.2) 削除変換  $T_D$ : 各  $a_k$  ( $1 \leq k \leq n$ ) に対して

$$T_D(a_k) \triangleq \lambda$$

(4.3) 挿入変換  $T_I$ :  $x = a_{i_1} a_{i_2} \dots a_{i_r}$  (各  $j$  ( $1 \leq j \leq r$ ) に対して  $a_{i_j} \in V_T$ ) とする. このとき, 各  $k$  ( $1 \leq k \leq r$ ) に対して,

$$T_I(a_{i_k}) \in \{a_{i_k} a_l \mid 1 \leq l \leq n\}$$

ただし,  $x$  の左端への挿入は  $T_I(\lambda) \in V_T$  で表わす. 例えば,  $a_{i_1} a_{i_2} a_{i_3} a_{i_4} \in V_T^*$ ,  $T_S(a_{i_3}) = a_{i_\mu}$ ,  $i_3 \neq i_\mu$ ,  $T_I(a_{i_2}) = a_{i_2} a_{i_\nu}$ , とすると,  $a_{i_1} a_{i_2} T_S(a_{i_3}) a_{i_4} = a_{i_1} a_{i_2} a_{i_\mu} a_{i_4}$ ,  $a_{i_1} T_D(a_{i_2}) a_{i_3} a_{i_4} = a_{i_1} a_{i_3} a_{i_4}$ ,  $a_{i_1} T_I(a_{i_2}) a_{i_3} = a_{i_1} a_{i_2} a_{i_\nu} a_{i_3}$  である.

次に, ストリングとストリングの距離およびストリングとランゲージとの距離の定義を与える.

[定義5] 任意の  $x, y \in V_T^*$  に対して,  $y$  から  $x$  を導くための変換の列  $T_{i_1} T_{i_2} \dots T_{i_u}$  (各  $j$  ( $1 \leq j \leq u$ ) に対して  $i_j \in \{S, D, I\}$ ) を  $J$  で示し, 用いた変換の数を  $|J|$  で示す. このとき,  $x$  と  $y$  の距離  $d(x, y)$  を次式で定義する.

$$d(x, y) \triangleq \inf_J \{|J|\}$$

例として,  $x = caabcd$ ,  $y = cbabdb$  の変換の一例を示す.

$$\begin{aligned} y &= cbabdb \\ &\rightarrow cT_S(b)abdb \\ &= caabdb \\ &\rightarrow caaT_I(b)db \\ &= caabcdb \\ &\rightarrow caabcdT_D(b) \\ &= caabcd \\ &= x \end{aligned}$$

従って, この場合  $|J|=3$  で, かつ  $d(x, y)=3$  である.

以上の定義より容易に次の命題が成立する.

[命題1] 有限長の任意の  $x, y \in V_T^*$  について, 代入, 削除, 挿入変換を有限回行えば,  $x$  から  $y$  を導くことができる.

[定義6]  $x$  と  $L(G)$  の距離  $d(x, L(G))$  を次式で定義する.

$$d(x, L(G)) \triangleq \inf \{d(x, y) \mid y \in L(G)\}.$$

定義6において,  $y$  は無限にあるので  $d(x, y)$  も無限に生じるが, 4. で述べる誤り訂正パーズングにより  $d(x, L(G))$  を有限回の操作で求めることができる.

### 3. 文脈自由型誤り訂正文法の構成

あるパターン集合を表現する文法  $G$  が推論<sup>6)</sup> された後, 未知パターン  $x$  に対して  $d(x, L(G))$  を算出するためには,  $G$  をもとにした誤り訂正文法  $G'$  の構成が重要となる. そこで, この節では誤り訂正文法の構成法について述べる.

$G = (V_N, V_T, P, S)$  を文脈自由型文法とする. このとき, 文脈自由型誤り訂正文法  $G' = (V'_N, V_T, P', S)$  は以下の手順に従って構成される.

- (1)  $A \rightarrow \alpha_0 a_1 \alpha_1 a_2 \alpha_2 \dots a_m \alpha_m$  ( $m \geq 0, A \in V_N, \alpha_i \in V_N^*, a_i \in V_T$ ) なる生成規則が  $P$  に存在すれば,  $A \rightarrow \alpha_0 E_{a_1} \alpha_1 E_{a_2} \alpha_2 \dots E_{a_m} \alpha_m$  ( $E_{a_i} \in V'_N$ ) なる生成

規則を  $P$  に加える.

(2) 全ての  $a_i \in V_T$  に対して, 以下の生成規則を  $P$  に加える.

- ①  $E_{a_i} \rightarrow a_i$
- ②  $E_{a_i} \rightarrow a_j \ (a_i \neq a_j)$
- ③  $E_{a_i} \rightarrow Ha_i \ (H \in V_N')$
- ④  $I \rightarrow a_i \ (I \in V_N')$
- ⑤  $E_{a_i} \rightarrow \lambda$

(3)  $P$  に以下の生成規則を加える.

- ①  $S' \rightarrow S \ (S' \in V_N')$
- ②  $S' \rightarrow SH$
- ③  $H \rightarrow HI$
- ④  $H \rightarrow I$

ここで,  $V_N' = V_N \cup \{S', H, I\} \cup \{E_{a_i} \mid a_i \in V_T\}$  である. このようにして得られた生成規則を  $P'$  とする. また,  $P'$  に含まれる  $E_{a_i} \rightarrow a_j, E_{a_i} \rightarrow \lambda, I \rightarrow a_i$  なる生成規則は誤り訂正規則と呼ばれ, それぞれ, 代入変換, 削除変換, 挿入変換に相当している.  $H$  は空でないストリングを生成するので,  $E_{a_i} \rightarrow Ha_i$  は  $a_i$  の前への挿入を表わし,  $S' \rightarrow SH$  はストリングの後への挿入を表わしている.

$G'$  の構成法から容易に  $L(G') = V_T^*$  となることがわかる. 今, 各  $i \ (1 \leq i \leq m)$  に対して  $G_i = (V_N, V_T, P_i, S)$  とし,  $G_i$  の誤り訂正文法を  $G'_i$  とすると,  $L(G_i) = L(G'_i) = \dots = L(G'_m) = V_T^*$  である. (Fig.1 参照)

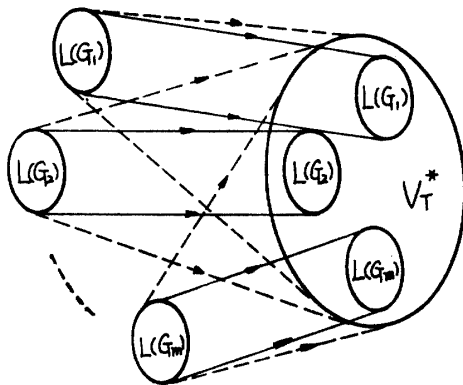


Fig. 1 Relation between  $L(G_i)$  and  $L(G'_i)$ .

以上のことから,  $L(G)$  と  $L(G')$  について容易に次の命題が成立する.

[命題 2] 文法  $G$  と誤り訂正文法  $G'$  に対して

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

が成立する.

#### 4. 誤り訂正パーズング

文法  $G = (V_N, V_T, P, S)$  と  $x \in V_T^*$  に対して,  $x \in L(G)$  か否かを調べる手順をパーズングといい, 誤り訂正規則を含む誤り訂正文法  $G'$  を用いて,  $x$  と  $L(G)$  との距離  $d(x, L(G))$  を求める手順を誤り訂正パーズングという.  $d(x, L(G)) = \inf \{d(x, y) \mid y \in L(G)\}$  を求める際,  $d(x, y)$  は無限に生じるが, 誤り訂正パーズングにより有限回の操作で求めることができる. パーズングには種々のアルゴリズムが提案されているが<sup>7)8)</sup>, 本論文では, Earley のパーズングアルゴリズムを基本にした誤り訂正パーズングについて考察する. 以下, そのアルゴリズムについて述べる.

文法  $G$  に対する誤り訂正文法を  $G' = (V_N', V_T, P', S')$  とし,  $x = a_1 a_2 \dots a_n \ (x \in V_T^*)$  とする. このとき誤り訂正パーズングを行うために, リスト  $I_0, I_1, \dots, I_n$  を以下の手順に従って作成する.

リストは

$$[A \rightarrow \alpha \cdot \beta, i, k] \ (A \in V_N, \alpha, \beta \in (V_N \cup V_T)^*)$$

なる項からなっている. ここで,  $A \rightarrow \alpha \beta$  は  $G'$  の生成規則を表わす.  $\cdot$  は, 最初  $[A \rightarrow \cdot r_1 r_2 \dots r_t] \ (r_i \in (V_N \cup V_T)^*, t \geq 1)$  のように, 生成規則の右辺の左端におかれ, パーズングを行う毎に  $r_1 \cdot r_2 \dots r_n, r_1 r_2 \cdot \dots r_n, \dots, r_1 r_2 \dots r_n \cdot$  と逐次移動する特殊な記号である.  $i \ (0 \leq i \leq n)$  は  $\alpha$  からの導出が, リスト  $I_i$  から始まったことを示し,  $k \ (0 \leq k)$  は  $\alpha$  からの導出に必要な誤り訂正規則の数を示す. 最後のリスト  $I_n$  に現れる  $[S' \rightarrow S \cdot, i, k]$  もしくは  $[S' \rightarrow SH \cdot, i, k]$  なるいくつかの項の  $k$  の最小値が  $d(x, L(G))$  を表している.

- (1)  $I_0$  に  $[S' \rightarrow \cdot S, 0, 0]$  と  $[S' \rightarrow \cdot SH, 0, 0]$  を加える.
- (2)  $I_0$  に  $[A \rightarrow \alpha \cdot B \beta, 0, k]$  があり,  $B \rightarrow \gamma \ (\gamma \in (V_N \cup V_T)^*)$  が  $P$  にあれば,  $I_0$  に  $[B \rightarrow \cdot \gamma, 0, 0]$  を加える.
- (3)  $I_0$  に  $[A \rightarrow \alpha \cdot B \beta, 0, k]$  と  $[B \rightarrow \beta \cdot, 0, l]$  があれば,  $I_0$  に  $[A \rightarrow \alpha B \cdot \beta, 0, m]$  を加える. ここで,  $B \rightarrow \beta$  が誤り訂正規則ならば  $m = k + l + 1$ , そうでなければ  $m = k + l$  とする.
- (4)  $I_0$  に加える項がなくなるまで (2), (3) を繰り返す.
- (5)  $I_{j-1} \ (1 \leq j \leq n)$  において  $a = a_j \ (x = a_1 a_2 \dots a_n)$  なるすべての  $[A \rightarrow \alpha \cdot a \beta, i, k]$  について,  $I_j$  に  $[A \rightarrow \alpha a \cdot \beta, i, k]$  を加える.
- (6)  $I_j$  に  $[B \rightarrow \gamma \cdot, i, k]$  があり,  $I_i$  に  $[A \rightarrow \alpha \cdot B \beta,$

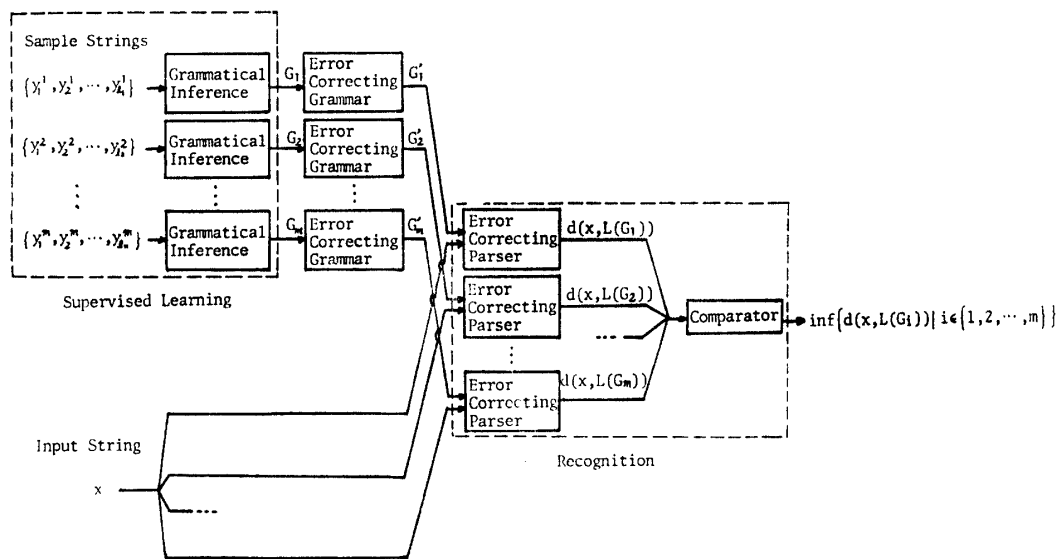


Fig. 2 Recognition system.

$h, l$  があれば,  $I_j$  に  $[A \rightarrow \alpha B \cdot \beta, h, m]$  を加える. ここで,  $B \rightarrow \gamma$  が誤り訂正規則ならば  $m = k + l + 1$ , そうでなければ  $m = k + l$  とする. ここで,  $m' \leq m$  なる  $[A \rightarrow \alpha B \cdot \beta, h, m']$  がすでに  $I_j$  にあれば,  $[A \rightarrow \alpha B \cdot \beta, h, m]$  を  $I_j$  に加えない. 同様に,  $m' > m$  なる  $[A \rightarrow \alpha B \cdot \beta, h, m']$  が  $I_j$  にあれば, これを  $I_j$  から削除し,  $[A \rightarrow \alpha B \cdot \beta, h, m]$  を加える.

- (7)  $I_j$  に  $[A \rightarrow \alpha \cdot B \beta, i, k]$  があり,  $B \rightarrow \gamma$  が  $P'$  にあれば,  $[B \rightarrow \cdot \gamma, j, 0]$  を  $I_j$  に加える.
- (8)  $I_j$  に加える項がなくなるまで (6), (7) を繰り返す.
- (9)  $I_n$  において  $[S' \rightarrow S \cdot, i, k]$  および  $[S' \rightarrow S H \cdot, i, k]$  を探す. ここで,  $k$  の最小値が用いられた誤り訂正規則の最小回数, すなわち,  $d(x, L(G))$  である.

5. 認識システム

本節では, 誤り訂正パーズングを用いた認識システムについて述べる. まず, 各  $i (1 \leq i \leq m)$  に対して, カテゴリー  $i$  に属するサンプルストリングの集合  $Y^i = \{y_1^i, y_2^i, \dots, y_{k_i}^i\}$  から  $Y^i \subseteq L(G_i)$  となるような文法  $G_i$  を, 教師付き学習によって推論する. 次に,  $G_i$  をもとにして誤り訂正文法  $G'_i$  を構成し, さらに, 入力ストリング  $x$  と  $G'_i$  を用いて, 誤り訂正パーズングを行い, 比較器により  $\inf \{d(x, L(G_i)) \mid 1 \leq i \leq m\} = d(x, L(G_j))$  が得られたならば,  $x$  のカテゴリーは  $j$  であると判定する (Fig. 2 参照).

6. 手書き片仮名文字認識実験システム

本節では, 手書き片仮名文字認識実験システムの概要を Fig. 3 に従って説明する.

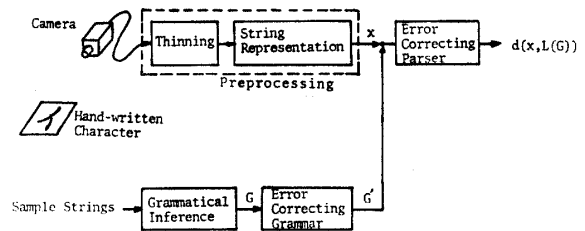


Fig. 3 Block diagram of system.

(6.1) 細線化

手書き文字をカメラで入力し,  $AD$  変換器を用いて  $64 \times 64$  メッシュで二値化した後,  $20 \times 20$  メッシュへ変換し 8 連結性を考慮した Hilditch のアルゴリズムを用いて細線化を行う<sup>9)</sup>.

(6.2) ストリング表現

細線化された二値パターンを右上から走査し, メッシュの連続したセルをその前のメッシュに対する相対的位置 (Fig. 5 参照) に従って, 記号序列化する. 次に,  $AAA \rightarrow a, CBB \rightarrow b$  のように長さ 3 の記号列を一つの基本パターン (Fig. 6 参照) にわりあて, 各パターン  $a, b, c, d$  とその接続関係 ( , + , \* , \* , ) , : 等を用いてストリング表現する.

(6.3) 文法推論

サンプルストリングからサブストリングを取り出し, その構成を表わす核生成規則を作る. 次に, 各



Table 1 Results of Simulation

Input character	String $X$	$d(X, L(G_i))$									
		$G_1$	$G_2$	$G_3$	$G_4$	$G_5$	$G_6$	$G_7$	$G_8$	$G_9$	$G_{10}$
イ	$a+a \times bb$	0	5	4	3	3	4	5	5	6	5
イ	$a+a \times cb$	1	6	4	4	2	5	5	5	6	5
イ	$b+a \times bb$	1	5	3	2	3	3	6	4	6	4
イ	$baa+d \times ba$	3	5	4	4	5	5	5	6	7	5
イ	$bba+a \times cd$	3	3	5	3	4	5	5	5	7	5
イ	$bb+ad \times cb$	4	6	5	4	5	5	5	6	7	4
ス	$d \times a+a \times c$	3	0	5	5	3	2	4	5	4	4
ス	$d \times b+a \times c$	3	0	6	5	3	3	4	5	3	5
ス	$dd \times a+a \times c$	4	0	6	5	4	4	5	6	4	6
ス	$d \times ba+a \times c$	4	0	7	6	4	4	5	6	3	6
ス	$d \times bb+a \times c$	4	1	6	5	5	3	6	6	3	5
ス	$d \times bb \times a \times cd$	6	3	7	6	6	5	7	5	3	5
ハ	$bbb: bb$	4	5	0	1	5	3	6	6	4	4
ハ	$c: bb$	5	5	0	1	4	4	4	6	5	5
ハ	$bbc: bba$	5	7	1	1	6	4	6	6	6	4
ハ	$bbb: bbaa$	6	7	2	2	8	3	8	6	6	4
ハ	$bbcd: bbb$	5	7	1	1	7	3	6	6	6	5
ハ	$bbbc: bbba$	6	8	1	1	8	4	9	6	7	5
ル	$bbba: bb$	4	6	1	0	6	4	6	6	4	5
ル	$bbbc: bb$	5	6	0	0	6	4	6	6	5	4
ル	$bbbbd: bb$	6	6	1	2	8	4	7	6	5	4
ル	$bbbcd: bb$	6	6	1	2	7	5	7	6	6	4
ル	$bbca: bbb$	4	7	1	1	7	4	6	7	6	5
ル	$bbbda: bbb$	5	7	1	1	8	5	7	7	6	6
へ	$a \times ccc$	4	6	3	4	1	4	2	7	4	5
へ	$aa \times cbc$	5	5	4	5	1	5	3	6	5	6
へ	$da \times ccc$	5	6	3	4	1	3	3	7	4	6
へ	$bb \times ccc$	5	6	3	5	1	4	3	6	5	5
へ	$ab \times ccc$	5	5	3	5	3	4	4	7	5	6
へ	$b+ad \times ccc$	5	7	5	6	3	4	4	4	6	4
ト	$bb+bb \times c$	5	5	3	4	5	0	6	4	5	1
ト	$bb+b \times d$	4	5	3	3	5	0	5	6	4	3
ト	$bb+ad \times c$	4	5	4	4	4	1	5	5	6	2
ト	$bbb+b \times c$	5	3	3	3	4	0	6	5	5	3
ト	$bc+bb \times cc$	5	5	2	4	4	0	6	5	6	1
ト	$bb+bb \times dd$	6	5	4	4	4	1	7	5	6	2
シ	$aaad: c: d$	5	6	3	3	4	4	0	6	5	6
シ	$aa: c: c$	4	4	4	4	2	5	0	7	5	6
シ	$aa: c: d$	4	5	4	4	3	5	0	7	4	5
シ	$aadd: b: d$	5	7	5	5	6	6	2	5	3	6
シ	$aaad: c: c$	5	5	6	6	3	6	0	5	3	7
シ	$baaaa: c: c$	5	5	6	5	2	6	1	7	6	7
ナ	$b+d \times ba \times d$	5	6	6	5	6	3	6	0	3	3
ナ	$b+d \times bb \times d$	4	7	5	5	6	2	6	1	4	3
ナ	$b+dd \times ba \times d$	6	6	6	6	7	4	7	0	3	4
ナ	$b+d \times ba \times dd$	6	6	7	6	7	4	7	0	3	3
ナ	$b+d \times bba \times dd$	6	5	7	6	8	4	8	0	4	3
ナ	$bb+dd \times ba \times d$	7	5	7	6	8	3	7	2	4	3
ラ	$ddd \times ba: dd$	7	4	5	5	8	5	6	3	0	6
ラ	$da \times baa: a$	5	3	4	4	5	7	5	7	2	7
ラ	$dd \times baa: ad$	6	4	6	6	7	7	5	7	2	7
ラ	$ddd \times bb: dd$	6	4	6	6	7	6	6	4	1	6
ラ	$ddd \times cba: ad$	7	5	7	7	7	7	6	5	2	7
ラ	$ddd \times baa: ad$	7	4	7	7	7	8	7	5	1	8
ヒ	$bb+bcdd \times dd$	8	8	6	7	9	4	8	5	5	1
ヒ	$b+abbdd \times d$	6	6	5	6	7	3	6	4	5	1

Table 1 (Continued)

Input character	String $X$	$d(X, L(G_i))$									
		$G_1$	$G_2$	$G_3$	$G_4$	$G_5$	$G_6$	$G_7$	$G_8$	$G_9$	$G_{10}$
ヒ	$bb+bbdd \times d$	7	7	5	5	8	2	6	6	5	1
ヒ	$ad+b \times abbdd$	5	7	7	5	8	5	7	5	5	4
ヒ	$aaa+c \times dbdd$	4	7	8	7	6	6	5	6	6	5
ヒ	$bb+bbcd \times dd$	6	9	6	7	7	4	6	7	7	1

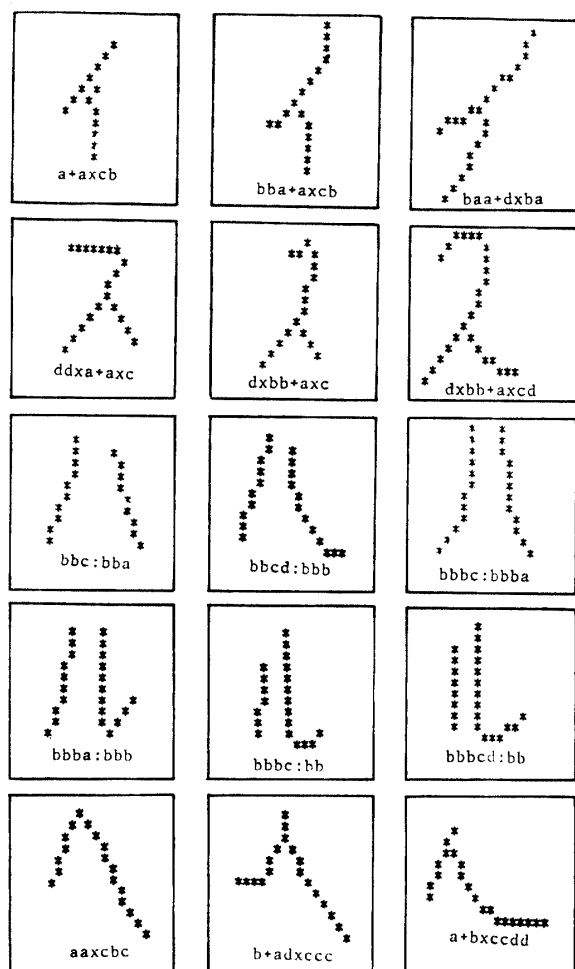


Fig. 8 Example of digitized patterns.

リングの長さは認識率と密接に関係していると考えられるが、細線化、ストリング化等の計算時間はストリング長に比例して大幅に増し、特に誤り訂正パーズングの段階ではストリング長の3乗に比例する。従ってストリングをできるだけ短くして認識率を高めることが望まれるが、計算時間と認識率との兼ね合いからの最適なストリング長に関する考察は今後の課題である。また、認識率を高める方法としては、誤り訂正規則に重みをつけることも考えられる。

なお、本実験で  $d(x, L(G))$  の算出に要した時間は、 $x$  と  $G$  によって異なるが、平均すると 8.4 sec (FACOM M-200) である。今回のシミュレーション

では、あるストリング  $x$  とすべてのランゲージ  $L(G_i)$  ( $1 \leq i \leq m$ ) との距離を求め、その最小値により認識を行ったが、さらに計算時間を短縮するために、入力文字をその構造的特徴、例えばストロークが離れているもの(例、ハ、ル、...)、ストロークがループを形成するもの(例、ヌ、ロ)等によって、ストリング表現の段階で、あらかじめ類別を行うことが考えられる。その後、Cocke-Kasami-Younger のアルゴリズム<sup>7)</sup> をもとにした誤り訂正パーズングによる実験では、計算時間がかなり短縮されており、これらについては、別途報告する予定である。

### 8. むすび

文脈自由型誤り訂正文法を用いた手書き片仮名文字に対する実験結果をふまえて、構文的パターン認識の手書き文字への有用性を考察した。基本要素の選択の問題を含めて、入力文字のストリング表現は、いわゆる特徴抽出に相当しており、最小限の特徴量で最大限の認識を行うことが要求されるが、ストリング長と計算時間および認識率に関する考察が望まれる。

### 謝 辞

日ごろ御指導御助言頂く本学高浪五男教授、井上克司助教授に感謝致します。また、御協力頂いた富田研の乙部由美子教務員および卒研生の諸氏に感謝すると共に、細線化に関する文献を御教示下さった電総研田村秀行博士に感謝致します。

### 参 考 文 献

- 1) K.S. Fu and S.Y. Lu: "A Clustering Procedure for Syntactic Patterns", IEEE Trans. on Syst., Man, Cybern., SMC-7, 734 (1978).
- 2) S.Y. Lu and K.S. Fu: "Error-Correcting Tree Automata for Syntactic Pattern Recognition", IEEE Trans. on Comp., C-27, 1040 (1978).
- 3) S.Y. Lu and K.S. Fu: "Stochastic Error-Correcting Syntax Analysis for Recognition of Noisy Patterns",

- IEEE Trans. on Compt., C-27, 1268 (1977).
- 4) 米崎, 佐藤, 榎本: “パターンの統計的性質を用いた構文的認識機構の自動構成とその評価”, 信学論 (D), 63-D, 859 (1980).
  - 5) 本多波雄: “オートマトン・言語理論”, コロナ社 (1978)
  - 6) K.S. Fu and T.L. Booth: “Grammatical Inference: Introduction and Survey Part 1”, IEEE Trans. on Syst., Man, Cybern., SMC-5, 95 (1975).
  - 7) A.V. Aho and J.D. Ullman: “The Theory of Parsing, Translating and Compiling”, Prentice Hall Inc., Vol. 1 (1972) p. 314.
  - 8) A.V. Aho and T.G. Peterson: “A Minimum Distance Error-Correcting Parser for Context-Free Languages”, SIAM J. Compt., 1, (1972) p. 305.
  - 9) 田村秀行: “細線化法についての諸考察”, 信学技報, PRL 75-66 (1975)

(昭和 56 年 4 月 15 日 受理)