

# 構文的手法による手書き片仮名文字認識

金岡泰保\*・岡田敏彦\*・富田真吾\*・鹿田泰治\*\*・田中恭治\*\*\*・綿谷正巳\*\*\*\*

## Recognition of Handwritten Katakana Characters by Syntax Analysis

Taiho KANAOKA, Toshihiko OKADA, Shingo TOMITA,  
Taiji SHIKADA, Kyoji TANAKA and Masami WATATANI

### Abstract

This paper investigates the validity of syntax analysis for recognition of handwritten Katakana characters.

A method of string representation which minimizes the length of the string is proposed and a weighted model for error-correcting parsing with substitution and insertion error is introduced. String representation of this paper has a advantage that some consecutive identical primitives in other methods are denoted only one primitive. Related to introducing a weighted error-correcting grammer, the original grammer is inferred from only one sample string which is the shortest and appears frequently in sample strings. From some experimental results for handwritten Katakana characters, the validity of our method is discussed.

### 1. まえがき

近年、形式言語理論における構文解析法を文字、図形などのパターン認識へ応用しようとする研究が盛んに行われている<sup>1),2)</sup>。構文的パターン認識は、1)入力パターンの細線化、2)基本要素およびオペレータの記号系列としてのストリング化、3)サンプルストリングを用いた文法推論と誤り訂正文法の構成、4)誤り訂正パーキング等の一連の過程を経てなされる。1)は入力パターンの前処理であり、その方法論はほとんど確立されている<sup>11),12)</sup>。2)は1)で得られた細線化パターンを文法に適合した語に変換する操作である。3)の文法推論は2)で得られたあるカテゴリーの複数個のサンプル語を基にそのカテゴリーを文法で表現する過程である<sup>9)</sup>。又、推論された文法(基本文法ともいう)によって生成される語に対して代入、削除、挿入等の記号の変換を施こしてあらゆる語が生成できるように構成されたのが誤り訂正文法である。4)は未知入力ストリングに

対して、それをある基本文法で生成される語とするのに要する変換の最小回数(語と言語の距離)を求めようとするものである。

このような方法を手書き文字認識などの具体的問題へ適用しようとするとき、種々の問題点が生じてくる。特に、4)の操作に要する時間は一般にストリングの長さの3乗の比例<sup>10)</sup>し、誤り訂正文法の生成規則の数に比例する。従って、処理時間の観点からは、ストリングの長さの短縮化と文法の終端記号に相当する基本要素とオペレータの数は可能な限り小さいことが望ましい。このことは認識率と密接に関係し、ストリングの過剰短縮化および基本要素数とオペレータ数の過剰削減は字種の増加に伴い認識率低下の要因となる。又、言語と語の距離として変換の回数を採用すると混読の原因となる<sup>6)</sup>。

本論文は、長さを短縮化するストリング化法と重み付き誤り訂正文法の一構成を提案し、手書き片仮名文字に対するこれらの手法の有用性を実験的に検討している。

長さには無関係な方向性だけに注目した基本要素を有効に用いる本ストリング化法は、従来の手法<sup>1)</sup>で見られた連続した基本要素を短縮化して一つの基本要素で

\*電子工学科

\*\* // (現, 東京芝浦電気)

\*\*\* // (現, 日本製鋼所)

\*\*\*\* // (現, 三菱電機)

表示するという特長があり、人力文字の大きさには依存しない一種の正規化的な効果を有している。重み付き誤り訂正文法の構成と関連して、基本文法の推論には、サンプルパターンの中から長さが最小でかつ出現頻度の高い1個のサンプルを基に推論するという簡潔な方法を用いており、又、誤り訂正文法に重みを導入することによってストリング間の距離を連続量に変換することができ、混読が解消される。

## 2. ストリング化

この章では、長さを短縮化する入力文字のストリング化法について述べる。なお、ストリング化の前処理段階である細線化は文献(11)に従っている。以下、入力パターンというときは細線化された  $\{0, 1\}$  上のパターンを意味するものとする。

まず、入力パターンに関する二、三の定義を与えよう。

[定義1] ある1の点の8近傍内に1の点が只1個存在するとき、その点を端点という。

[定義2] ある1の点の8近傍内に1の点が3個以上存在するとき、その点を分岐点という。

又、端点と分岐点を総称して特異点ともいう。

[定義3] となり合う2つの特異点間の任意の2点(特異点も含む)  $a, b$  に対して、 $a$  と  $b$  の間の1の列を枝  $ab$  という。

例えば、Fig. 1 では  $a, b, c$  および  $d$  が端点で  $e$  は分

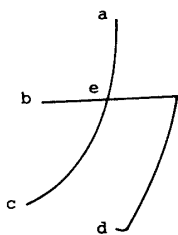


Fig.1 Example of singular points.

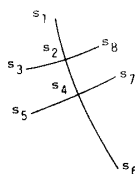


Fig.2 Numbering of singular points.

岐点である。なお、口などのようにループのある文字では、ループの部分において最初と最後に検出された1の点を端点としている。又、シミュレーションでは、ある分岐点の一定内周内に他の分岐点が存在する場合

はそれらを同一の分岐点とみなしている。

### 2.1 基本要素化

入力パターンにおける1の点の検出順序に従って、特異点に Fig. 2 のような番号付けを行う。その後には各枝の基本要素化がなされるが、この節では、枝  $s_\alpha s_\beta$  ( $\alpha < \beta$ ) の基本要素化法について述べる。ここで用いる基本要素  $u_k$  ( $u_k \in U = \{u_i \mid 1 \leq i \leq m\}$ ) は Fig. 3 のように任意の長さの方向性を持つものである。

[基本要素化アルゴリズム]

- (1) 枝  $s_\alpha s_\beta$  上の点  $x$  に対して、 $\angle s_\alpha x s_\beta$  が最小とある点を  $x_1$  とし、 $\angle s_\alpha x_1 s_\beta \leq \theta$  ならば枝  $s_\alpha x_1$  上で  $\angle s_\alpha x x_1$  が最小となる点  $x_2$  を求める。このとき、 $\angle s_\alpha x_2 x_1 \leq \theta$  ならば枝  $s_\alpha x_2$  上で  $\angle s_\alpha x_3 x_2$  が最小となる点  $x_3$  を求める。同様の操作をくり返して、 $x_\tau$  ( $\tau \geq 1$ ) に対して  $\angle s_\alpha x_{\tau+1} x_\tau > \theta$  が満たされたならば、ベクトル  $\overrightarrow{s_\alpha x_\tau}$  の方向に最も近い基本要素  $u_k$  ( $u_k \in U$ ) を枝  $s_\alpha x_\tau$  に割り当てる。
- (2) 枝  $x_\tau s_\beta$  に対して(1)と同様の操作をくり返す。

以上の手順によって、枝  $s_\alpha s_\beta$  は  $u_{i_1} u_{i_2} \dots u_{i_t}$  (各  $j$  ( $1 \leq j \leq t$ ) に対して  $u_{i_j} \in U$ ) と基本要素化される。Fig. 4 の枝  $s_\alpha s_\beta$  において角度の検出は次の順

- 1)  $\angle s_\alpha x_1 s_\beta = \min \{ \angle s_\alpha x s_\beta \mid x \text{ は } s_\alpha s_\beta \text{ 上の点} \} \leq \theta$
- 2)  $\angle s_\alpha x_2 x_1 = \min \{ \angle s_\alpha x x_1 \mid x \text{ は } s_\alpha x_1 \text{ 上の点} \} \leq \theta$
- 3)  $\angle s_\alpha x_3 x_2 = \min \{ \angle s_\alpha x x_2 \mid x \text{ は } s_\alpha x_2 \text{ 上の点} \} > \theta$
- 4)  $\angle x_2 x_4 s_\beta = \min \{ \angle x_2 x s_\beta \mid x \text{ は } x_2 s_\beta \text{ 上の点} \} \leq \theta$
- 5)  $\angle x_2 x_5 x_4 = \min \{ \angle x_2 x x_4 \mid x \text{ は } x_2 x_4 \text{ 上の点} \} > \theta$
- 6)  $\angle x_4 x_6 x_\beta = \min \{ \angle x_4 x x_\beta \mid x \text{ は } x_4 x_\beta \text{ 上の点} \} > \theta$

但し、 $\theta$  は角度閾値

になされ、ベクトル  $\overrightarrow{s_\alpha x_2}$ ,  $\overrightarrow{x_2 x_4}$  および  $\overrightarrow{x_4 s_\beta}$  の方向に最も近い基本要素をそれぞれ  $u_{i_1}$ ,  $u_{i_2}$ ,  $u_{i_3}$  とすると、枝  $s_\alpha s_\beta$  は  $u_{i_1} u_{i_2} u_{i_3}$  と基本要素化される。従来の手法<sup>1)</sup>で見られた  $u_1 u_1 u_2 u_2 u_2$  等のように同じ基本要素のくり返しは、本

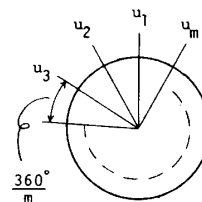


Fig.3  $m$  directed primitives.

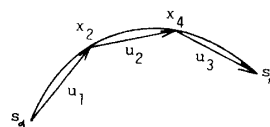


Fig.4 Example of string representation of a branch  $s_\alpha s_\beta$ .

手法を用いると  $u_1 u_2$  と短縮化される。従って、本手法は入力パターンの大きさには依存しない一種の正規化的効果を有していると考えられる。

### 2.2 スtring化

String化は基本要素化された各枝をオペレータによって接続する操作であり、この操作によって入力パターンの記号系列化がなされる。枝と枝の接続関係を表示するオペレータとして +, ×, \*, () および : を用いているが、各オペレータについては文献(3)を参照されたい。

Fig. 5 に細線化パターンとそのString化のいくつかの例を示した。但し、シミュレーションでは  $m =$

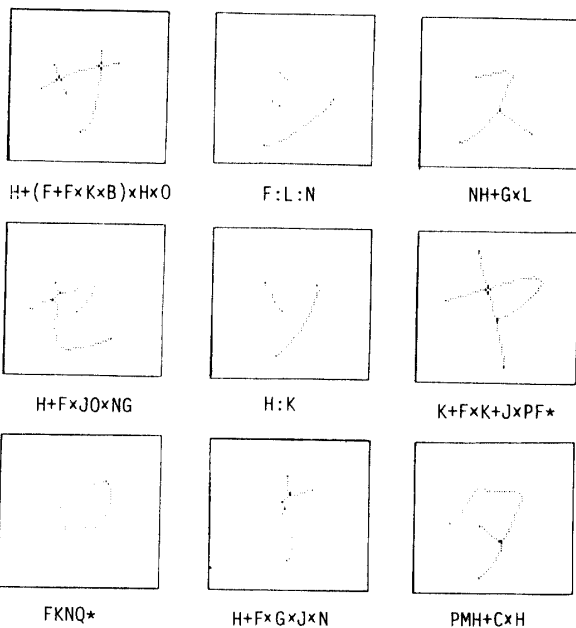


Fig.5 Character patterns and thier string representations.

16 とし、Fig. 5 において A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q はそれぞれ基本要素  $u_1, u_2, \dots, u_{16}$  を意味する。なお、入力パターンは  $64 \times 64$  量子化レベルで 2 値細線化を行い、分岐点からのびている枝の内、3 メッシュ以内の短い枝は細線化で生じた“ヒゲ”とみなし削除した。

### 3. String間の距離

この章では、まず、形式文法の定義を与え、更に、代入変換および挿入変換なるString上の変換を用いてString間の距離を導入し、これを基にStringと言語間の距離を定義する。

[定義4] 文法とは  $G = [V_N, V_T, P, S]$  である。ここで、 $V_N, V_T, P$  はそれぞれ非終端記号、終端記号および生成規則の有限集合であり、 $S$  は開始記号である。

又、 $G$  で生成される言語の集合を  $L(G)$  で表わす。文法の詳細は文献(4)を参照されたい。

[定義5]

(5.1) 代入変換  $T_s$ : 任意の  $a, b \in V_T$  ( $a \neq b$ ) に対して

$$\omega_1 a \omega_2 \xrightarrow{T_s, S(a, b)} \omega_1 b \omega_2, \omega_1, \omega_2 \in V_T^*$$

$S(a, b)$  は  $a = b$  を代入するときの重みである。

(5.2) 挿入変換  $T_i$ :

(i)  $\omega_1 a \omega_2 \xrightarrow{T_i, I(a, b)} \omega_1 b a \omega_2, \omega_1, \omega_2 \in V_T^*$   
 $I(a, b)$  は  $a$  の前に  $b$  を挿入するときの重みである。

(ii)  $\omega \xrightarrow{T_i, I(b)} \omega b, \omega \in V_T^*$   
 $I(b)$  は  $\omega$  の後へ  $b$  を挿入するときの重みである。

[定義6] 任意の  $x, y$  ( $x, y \in V_T^*, |x| \geq |y|$ ) に対して、 $y$  から  $x$  を導くために用いた変換の系列を  $J = T_{i_1} T_{i_2} \dots T_{i_n}$  (各  $j$  ( $1 \leq j \leq n$ ) に対して  $T_{i_j} \in \{T_s, T_i\}$ ) とし、各  $j$  に対して変換  $T_{i_j}$  に付随する重みを  $W_j$  とする。このとき、 $x$  と  $y$  の距離を次で与える。

$$d(x, y) = \min_J \left\{ \sum_{k=1}^n W_k \right\}$$

$d(x, y)$  を与える変換の系列は重みの総和が最小となる系列である。例えば、 $x = cbbabdbb, y = cbabdbb$  とすると、 $y = cdabdbb \xrightarrow{T_s, S(d, b)} cbabdbb$

$$\xrightarrow{T_s, S(b, d)} cbabdbb \xrightarrow{T_i, I(a, b)} cbbabdbb = x$$

であり、 $\sum W_k = s(d, b) + s(b, d) + I(a, b)$  である。

[定義7] String  $x$  と言語  $L(G)$  の距離  $d(x, L(G))$  を次で定義する

$$d(x, L(G)) = \begin{cases} \min \{d(x, y) \mid y \in L(G)\}, & |x| \geq |\omega_M| \text{ のとき} \\ \infty, & |x| < |\omega_M| \text{ のとき} \end{cases}$$

但し、 $\omega_M$  は  $L(G)$  の中で最も長さの小さいStringを示している。

任意のString  $x, y$  に対して、 $y$  を変換して  $x$  を得るには代入変換、挿入変換の他に削除変換を考慮しなければならないが、これら三つの変換を用いると、構文解析が複雑<sup>(6)</sup>となり処理時間の点で問題となる。本章の基本文法の推論過程で最も短いStringを利用することにより、実際上は、上で定義した二つの変換を考慮すればよいことがわかる。すなわち、定義7で示したように、 $|x| < |\omega_M|$  のときは  $d(x, L(G)) = \infty$  と考え、 $x$  は文法  $L(G)$  で表わされるカテゴリーには属しないとするわけである。

### 4. 基本文法と重み付き誤り訂正文法の構成

#### 4.1 基本文法 $G$ の構成

あるカテゴリーに属するいくつかのサンプルストリ

ングの中から長さが最小で出現頻度の高いストリングを1個選ぶ。今、そのストリングを

$$\omega = S_1 \Delta_1 S_2 \Delta_2 \cdots S_n \Delta_n$$

とする。但し、各  $i$  ( $1 \leq i \leq n$ ) に対して  $S_i \in U^*$ ,  $\Delta_i \in Q$ ,  $\Delta_n \in \{QU\phi\}$  で  $Q$  はオペレータの集合である。このとき、 $\omega$  から次のような文法  $G$  を構成する。

$$G = (V_N, V_T, P, S)$$

ここで、

$$V_N = \{S\} \cup \{A_i \mid 1 \leq i \leq n\}$$

$$V_T \equiv \{U \cup Q\}$$

$$P = \{S \rightarrow A_1 \Delta_1 A_2 \Delta_2 \cdots \Delta_{n-1} A_n \Delta_n,$$

$$A_1 \rightarrow S_1, A_2 \rightarrow S_2, \dots, A_n \rightarrow S_n\}$$

明らかに、 $G$  によって生成される言語は  $L(G) = \{\omega\}$  である。又、 $G$  は文脈自由文法であるから Chowsky の標準形<sup>(14)</sup>に変換することができる。

## 4.2 誤り訂正文法 $G^c$ の構成

4.1 で構成された Chowsky 標準形の文法  $G' = (V'_N, V'_T, P', S)$  を基にして誤り訂正文法  $G^c = (V'_N \cup \{I\}, V'_T^c, P' \cup P^c, S)$  を次のようにして構成する。

(1) 生成規則  $P'$  の中に、 $A \rightarrow \alpha$  ( $A \in V'_N, \alpha \in V'_T$ ) なる形の生成規則があれば、この規則に対して次の生成規則を  $P^c$  に加える。

a)  $A \rightarrow \beta$  ( $\beta \in V'_T^c, V'_T^c = U \cup Q, \beta \neq \alpha$ )

b)  $A \rightarrow IA, c) I \rightarrow \gamma$  ( $\gamma \in V'_T^c$ )

(2)  $P^c$  に次の生成規則を加える。

d)  $S \rightarrow SI, e) I \rightarrow II$

a) は代入変換、b) ~ e) は挿入変換に相当する。又、 $L(G^c) = V'_T^{c*} - K$  ( $K$  は長さが  $|\omega_K|$  未満の全てのストリングの集合) である。

## 4.3 重み付き誤り訂正文法 $G_w^c$ の構成

4.2 で構成した誤り訂正文法  $G^c = (V_N^c, V_T^c, P^+, S)$  ( $V_N^c = V'_N \cup \{I\}$ ,  $P^+ = P' \cup P^c$ ) の生成規則  $P^+$  に重みを付けた文法  $G_w^c = (V_N^c, V_T^c, P_w^+, S)$  は次のようにして構成される。

今、生成規則  $P^+$  の中で  $A \rightarrow \alpha$  ( $A \in V_N^c, \alpha \in V_T^c$ ) の形をした生成規則の集合を  $R = \{R_i \mid 1 \leq i \leq l\}$  とする。又、重みを付けるためのサンプルストリングを  $T = \{\omega_i \mid 1 \leq i \leq \nu\}$  とする。このとき、 $\omega_i \in T$  に対して  $d(\omega_i, L(G))$  を与える変換の系列  $J$  が  $m_i$  ( $m_i \geq 1$ ) 個存在するなら、 $m_i$  個の各々の  $G^c$  による導出経路を調べて、 $R_j \in R$  の平均使用回数  $N_{ij}$

$$N_{ij} = n_{ij} / m_i$$

但し、 $n_{ij}$  は  $m_i$  の全ての導出経路に対して  $R_j$  を用いた回数である。

$$W_j = \begin{cases} t & \nu / \sum_{i=1}^{\nu} N_{ij} \geq t \text{ のとき} \\ \sum_{i=1}^{\nu} N_{ij} & \leq t \text{ のとき} \end{cases}$$

を求め、生成規則  $R_j$  に対する重み  $W_j$  を

与える。但し、 $t$  は  $W_j$  の上限を与える閾値である。従って、使用頻度に逆比例して重みを付けることになる。又、 $R$  以外の生成規則は変換の回数には関与しないからその重みは考慮しない。

あるカテゴリーの基本文法  $G$  と重み付き誤り訂正文法  $G_w^c$  が構成されたとすると、ストリング  $x, y$  に対して  $x \in L(G), y \notin L(G)$  ならば、 $G_w^c$  によって  $x$  と  $y$  をパーズングすることによって  $0 = d(x, L(G)) < d(y, L(G))$  が得られることが容易にわかる。重みを考慮しない場合は未知入力ストリング  $x$  に対して  $d(x, L(G_i))$  ( $i=1, 2, \dots$ ) が最小となる  $i$  は複数個存在することがあり混読の原因<sup>(5),(6)</sup>となるが、重みを導入することによって混読はほとんど解消される。このことはカテゴリー数が大なるほど有効であると考えられる。Table 1 にいくつかの字種に対する基本文法と、基本文法と重み付き誤り訂正文法を構成するために用いたサンプルストリング数を示し、Table 2 に  $K$  の重み付き誤り訂正文法を示している。

## 5. 誤り訂正パーズング

ストリングと言語の距離  $d(\omega, L(G))$  は誤り訂正パーズングを行うことによって得られるが、ここでは、Cock-Younger-Kasami の上昇形構文解析法を基礎にした重み付き誤り訂正パーズングアルゴリズムを用いる。

[アルゴリズム]

(入力) ストリング  $\omega = a_1 a_2 \cdots a_n$  ( $|\omega| = n$ ) と重み付き誤り訂正文法  $G_w^c = (V_N^c, V_T^c, P^+, S)$ 。

(出力) 距離  $d(\omega, L(G))$

対  $(A, m)$  ( $A \in N_N^c, m$  は非負実数) を要素とするパーズテーブル  $[T_{ij}]$  ( $1 \leq i \leq n, 1 \leq j \leq n-i+1$ ) を次のように構成する。

- (1) 各  $i$  ( $1 \leq i \leq n$ ) に対して、 $A \xrightarrow{w_i} a_i \in P^+$  ( $w_i$  は生成規則  $A \rightarrow a_i$  ( $A \in V_N^c, a_i \in V_T^c$ ) の重み) であれば  $(A, W_i)$  を  $T_{i1}$  に加える。
- (2) 各  $i, j$  ( $1 \leq i \leq n, 1 \leq j < j$ ) に対して  $T_{ij}$  が完成したならば、 $T_{ij} = \{(c, l+m) \mid \text{ある } k (1 \leq k < j) \text{ に対して } C \rightarrow AB \in P^+ \text{ かつ } (A, l) \in T_{ik}, (B, m) \in T_{i+k, j-k}\}$  とする。
- (3) (2) の操作を各  $i$  ( $1 \leq i \leq n$ ), 各  $j$  ( $1 \leq j \leq n-i+1$ ) に

文字	G	N	文字	G	N	文字	G	N
ク	S→A+B A→j B→C×D C→g D→n h	8	ワ	S→A+B A→j B→C×D C→g D→n h	10	ス	S→A+B A→n g B→C×D C→g D→l	40
ヒ	S→A+B A→j B→C×D C→j n D→n	42	ヨ	S→A+B A→o j B→C×D C→f D→j f	32	マ	S→A+B A→n g B→C×D C→c D→l	25
ア	S→A+B A→n g B→C×D C→b D→h	26	ム	S→A+B A→h o B→C×D C→l D→b	25	ユ	S→A+B A→n j B→C×D C→e D→n	46
タ	S→A+B A→p c C→n h B→D×E D→c E→g	17	ヲ	S→A+B A→n h B→C×D C→e D→h	44	ウ	S→A+B A→j B→C×D C→f j D→n k	40

Table.1 Some original grammers and number of sample strings.

S ·A1AA 0.	A1 ·A 10.00	A2 ·A 10.00	BB ·A 10.00	I ·A 10.00
AA ·A2BB 0.	A1 ·B 10.00	A2 ·B 10.00	BB ·B 10.00	I ·B 10.00
S ·S I 0.	A1 ·C 10.00	A2 ·C 10.00	BB ·C 10.00	I ·C 10.00
I ·I I 0.	A1 ·D 10.00	A2 ·D 10.00	BB ·D 10.00	I ·D 10.00
A1 ·I A1 0.	A1 ·E 10.00	A2 ·E 10.00	BB ·E 10.00	I ·E 1.56
A2 ·I A1 0.	A1 ·F 10.00	A2 ·F 10.00	BB ·F 10.00	I ·F 1.56
BB ·I BB 0.	A1 ·G 10.00	A2 ·G 10.00	BB ·G 0.36	I ·G 0.85
	A1 ·H 10.00	A2 ·H 10.00	BB ·H 0.16	I ·H 10.00
	A1 ·J 10.00	A2 ·J 10.00	BB ·J 2.28	I ·J 1.74
	A1 ·K 10.00	A2 ·K 10.00	BB ·K 10.00	I ·K 10.00
	A1 ·L 10.00	A2 ·L 4.93	BB ·L 1.97	I ·L 1.97
	A1 ·M 10.00	A2 ·M 1.18	BB ·M 10.00	I ·M 4.93
	A1 ·N 10.00	A2 ·N 0.11	BB ·N 10.00	I ·N 10.00
	A1 ·O 10.00	A2 ·O 4.93	BB ·O 10.00	I ·O 10.00
	A1 ·P 0.10	A2 ·P 10.00	BB ·P 10.00	I ·P 10.00
	A1 ·Q 4.93	A2 ·Q 10.00	BB ·Q 10.00	I ·Q 10.00
	A1 ·+ 10.00	A2 ·+ 10.00	BB ·+ 10.00	I ·+ 10.00
	A1 ·× 10.00	A2 ·× 10.00	BB ·× 10.00	I ·× 10.00
	A1 ·( 10.00	A2 ·( 10.00	BB ·( 10.00	I ·( 10.00
	A1 ·) 10.00	A2 ·) 10.00	BB ·) 10.00	I ·) 10.00
	A1 ·: 10.00	A2 ·: 10.00	BB ·: 10.00	I ·: 10.00
	A1 ·* 10.00	A2 ·* 10.00	BB ·* 10.00	I ·* 10.00

Table.2 Weighted error-correcting grammar  $G_w^c$  for “ク”

対して行う。但し、(1), (2), (3)において  $T_{ij}$  に前項が同じ対があれば、後項の数が最も小さい対以外は削除する。

- (4)  $T_{in}$  中の開始記号をもつ対 (S, h) における h が  $d(\omega, L(G))$  である。

### 6. スtringのクラス分けと認識方法

認識過程での処理時間の短縮化のために、ストリン

グのオペレータ並びに着目して、手書き片仮名文字を Table 3 で示すような 26 種類にクラス分けした。すなわち、入力パターンのストリング化が完了すると、まず、そのストリングが Table 3 のどのクラスに属するかを判定し、さらに、そのクラス内のどのカテゴリーに属するかをパーキング処理によって決定するのである。従って、字種が 1 個のクラスはパーキングを行うことなく直ちにその文字を認識結果とする。 $\xi$  ( $\xi \geq 2$ ) 個の字種を有するクラスに属すると判定されたストリング  $x$  に対しては、重み付き誤り訂正パーキングを行って

$$\min \{d(x, L(Gc_i)) \mid 1 \leq i \leq \xi\} = d(x, L(Gc_r))$$

が得られたならば、 $x$  はカテゴリー  $c_r$  に属すると判定される。

### 7. 認識実験および検訂

データは、18~25 歳の男女 60 名の濁点、半濁点を除く手書き片仮名文字 45 種で、総データ数 2497 個である。全データの内 1,820 個を用いて基本文法および重み付き誤り訂正文法を構成し、残りの 677 個で認識実験を行った。又、データはタテ 33mm ヨコ 33mm の枠内に書かれたものである。

認識率は 94.5% であった。クラス別認識率と誤認識

文字を Table 3 に示した。クラス 1 の“フ”と“コ”の誤読の改善法として、曲線や折線を直線近似するストリング化段階で角度閾値  $\theta$  を適切に設定することが考えられる。すなわち、本実験では全ての入力文字に対して  $\theta$  を一定としたが、クラスに応じて  $\theta$  を変えてストリング化を行うことも必要であろう。クラス 3 の“ハ”“リ”“ソ”“ン”およびクラス 5 の“ツ”“シ”の誤読は、ストリング化において離れた枝相互間の位置関係を表わす情報がないためと考えられる。従って、クラス 3 あるいはクラス 5 と判定されたストリングに

対しては離れた枝の位置関係をも考慮したストリング化を行う必要がある。

Table 4 にクラス 1 のストリングと認識例を示した。又、入力から認識に至るまでの処理時間は約 400ms/文字 (FA COM-200) であり、細線化、ストリング化、パーキング処理の比率は約 1 : 7 : 6 である。細線化およびストリング化で用いた 64×64 量子レベルの減少、重み付き誤り訂正パーキング処理時に閾値を設けて d

クラス番号	演算子並び	文	字	認識率	誤認識文字
1	演算子なし	ノ, ヘ, フ, レ, ク, ワ, コ, マ, ム, ア, ロ		96.0	コ→ロ
2	*	ロ		100	
3	:	ハ, リ, ソ, シ, ニ, ル, ラ, ア, ス, ヒ, ウ		86.8	ソ→ル
4	+×	イ, ト, ア, マ, ム, ス, ユ, ヒ, ヲ, ワ, ク, ヨ, タ, ウ		98.1	ア→ス ラ→ヨ
5	::	ミ, ツ, シ		78.7	ミ→ツ, シ
6	+*	タ, ロ		100	
7	+×	ナ, メ, ヤ, カ, ス, セ, モ		97.6	ヤ→カ
8	::+	テ, ケ, ウ		94.4	テ→ケ
9	+×	オ		100	
10	+×	エ, ケ		100	
11	++*	ア, タ		100	
12	::+	チ, ネ, モ		100	
13	+×	チ, モ		100	
14	+×	ヤ		100	
15	+×	ホ		100	
16	+×	キ, サ		100	
17	+(+×)×	オ, モ		100	
18	+(+××)×	キ		100	
19	+(+××)××	サ		100	
20	+×::	ホ		100	
21	+×	ネ		100	
22	::+	ネ		100	
23	+×	ネ		100	
24	::+	ネ		100	
25	+×	ネ		100	
26	+(+××)×	モ		100	

Table.3 Classification of Katakana characters and recognition results.

INPUT STRING	dw (ω, L (Gi))											認識結果	CPU (MS)
	L(Gノ)	L(Gヘ)	L(Gフ)	L(Gレ)	L(Gク)	L(Gワ)	L(Gコ)	L(Gマ)	L(Gム)	L(Gア)	L(Gロ)		
ア OGJ	11.0	21.6	3.8	30.0	22.3	21.2	20.3	10.4	30.0	0.3	—	ア	53
ア NGJ	11.0	20.9	3.7	21.1	22.3	21.2	20.1	10.6	21.9	0.9	—	ア	52
ア OGHG	11.3	31.6	4.8	40.0	21.0	20.9	11.6	11.4	30.2	2.1	21.5	ア	118
ア OGJG	11.5	31.6	6.0	30.1	21.2	21.3	11.1	11.4	23.9	0.8	12.1	ア	120
ク PNH	20.2	11.0	6.6	21.1	0.4	1.3	30.0	12.1	20.4	21.4	—	ク	52
ク PNG	20.4	11.0	6.8	21.1	0.6	1.7	30.0	12.0	20.4	21.4	—	ク	52
ク QNH	25.1	21.8	15.2	31.1	6.8	12.0	11.2	20.6	30.4	30.7	35.7	ク	119
ク PNLG	22.9	11.1	8.5	22.0	2.5	2.6	30.1	3.2	11.4	22.1	29.0	ク	118
ク QNH	20.2	11.8	10.3	21.1	5.2	10.3	30.0	21.3	20.4	21.4	—	ク	52
ク PNHF	25.1	21.0	11.5	22.5	1.9	3.0	11.2	11.6	23.9	30.7	35.7	ク	119
ク PNJG	21.0	21.0	10.0	12.6	2.3	2.9	20.3	13.1	17.5	20.6	26.3	ク	120
ク QNG	20.4	11.8	10.5	21.1	5.4	10.7	30.0	21.2	20.4	21.4	—	ク	52
ク QMG	20.4	11.7	20.3	30.0	6.5	20.5	30.0	30.0	30.0	21.4	—	ク	52
ク PNHE	25.1	21.0	11.5	22.5	1.9	4.0	10.9	11.6	23.9	30.7	26.3	ク	119
コ NJE	15.5	20.9	11.4	11.2	30.0	25.5	0.4	20.4	15.6	20.7	—	コ	52
コ OJE	15.5	21.6	11.5	20.1	30.0	25.5	0.6	20.2	23.6	20.1	—	コ	52
コ OJF	15.5	21.6	11.5	20.1	30.0	30.0	0.9	20.2	23.6	20.1	—	コ	53
コ NLJE	18.1	30.1	13.1	12.1	18.8	24.0	4.9	22.4	15.3	20.8	2.1	ロ	119
コ NHGE	15.7	30.9	7.4	31.1	21.9	23.3	1.5	11.6	22.5	12.2	11.2	コ	119
コ NHE	15.1	20.9	5.2	21.1	30.0	25.5	0.9	10.6	21.9	20.7	—	コ	52
コ NJGE	15.9	30.9	8.6	21.2	21.9	23.3	1.1	14.1	16.5	11.5	11.2	コ	119
ノ Q	10.0	—	—	—	—	—	—	—	—	—	—	ノ	2
ノ H	0.2	—	—	—	—	—	—	—	—	—	—	ノ	1
ノ G	0.4	—	—	—	—	—	—	—	—	—	—	ノ	1
ノ JG	1.0	20.0	3.1	10.1	—	—	—	—	—	—	—	ノ	10
ノ JHF	5.7	30.0	7.8	20.1	30.0	30.0	11.1	13.6	23.2	30.0	—	ノ	51
ノ LJG	3.5	30.0	4.8	11.0	20.4	20.5	20.1	21.9	17.3	21.4	—	ノ	52

Table.4 Examples of distance between a string and a language.

$(x, L(G\sigma)) > \delta$ ならばパーシングの途中で  $x$  キ  $\sigma$  とする等の方法によって処理時間は改善されるものと考えられる。

## 8. むすび

長さを短縮するストリング化法と重み付き誤り訂正文法を用いた構文的手法の手書き片仮名文字認識への有用性を実験的に検討した。未知パターンに対する認識率は 94.5% であり、本手法の有用性が確認された。

重み付き誤り訂正文法の構成と関連して、サンプルパターンの中から長さが最小でかつ出現頻度の高いサンプル 1 個から基本文法を構成するという本手法は簡潔な文法推論法といえよう。又、重み付き誤り訂正文法の採用は、混読の解消と同時に認識率の向上にも関与していると考えられるが、このことの理論的、実験的な考察は今後の課題である。更に、本手法の平仮名、漢字などの他の文字への応用も興味ある課題である。

謝辞 本研究に際し、ご助言頂いた本学高浪五男教授、井上克司助教授ならびに御討論頂いた富田研の諸氏に感謝します。

## 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 Compt., 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) K. C. You and K. S. Fu: "A Syntactic Approach to Shape Recognition Using Attributed Grammers" IEEE Trans. on Syst, Man, Cybern., SMC-9, 334 (1979).
- 5) 森田, 金岡, 富田, 岡田: "文脈自由型誤り訂正文法によるパターン認識について", 山口大学工学部研究報告 32, 195 (1981).
- 6) 岡村, 金岡, 富田, 岡田: "Bottom-up 的構文解析による手書き片仮名文字認識", 山口大学工学部研究報告 32, 319 (1982).
- 7) 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).
- 8) 山崎, 外村: 文脈自由言語の bottom-up 的最小誤り訂正アルゴリズムについて, 情報処理学会論文誌, 18, 781 (1977)
- 9) K. S. Fu and T. L. Booth: "Grammatical Inference: Introduction and Survey-Part I", IEEE Trans, on Syst, Man, Cybern., SMC-5, 95 (1975).
- 10) A. V. Aho and J. D. Ullmann: "The Theory of Parsing, Translation and Compiling", Prentice Hall Inc., Vol. 1 (1972).
- 11) C. J. Hilditch: "Linear skeletons from square cupboards", In Mechine Intelligence IV, B. Meltzer and D. Michie, Eds., University Press, Edinburgh, 403 (1969).
- 12) 田村秀行: "細線化法についての諸考察", 信学技報, PRL 75-66 (1975).
- 13) 衣松, 浜本, 他: "手書き片仮名文字を対象としたストリング化の一手法", 山口大学工学部研究報告, 32, 203 (1981).
- 14) J. E. Hopcroft and J. D. Ullman: "Formal Languages and their Retation to Automata", Addison-wesley (1969).

(昭和 57 年 10 月 15 日受理)