Bottom-up的構文解析による手書き片仮名文字認識

岡村健史郎*・金岡泰保**・富田真吾**・岡田敏彦**

Bottom-up Algorithm for syntactic Pattern Recognition of Handwritten Katakana Characters

Kenshiro OKAMURA, Taiho KANAOKA, Shingo TOMITA, Toshihiko OKADA

Abstract

Recently, several methods for pattern recognition by context-free grammar have been published. However, most of them use top-down algorithms based on three types of error-transformations, substitution, deletion and insertion, and these methods have rather time-consuming parser.

In this paper, we propose a new error-correcting parser for the improvement of parsing time. This parser bases on Cocke-Younger-Kasami's bottom-up algorithm and uses only two types of error-transformations, substitution and deletion. From some experimental results for handwritten Katakana characters, the validity of our method is discussed.

1. まえがき

近年，構文的パターン認識における文脈自由型文法の有用性が報告されている．しかしながらそのほとんどは，代入，削除および挿入なる3種類の変換を基にしたものである．特に，認識実験においてはその大半が top-down 的であり，bottom-up 的手法によるものはほとんどなされていない[1~4]．

本論文では，代入，削除なる2種類の変換を用いて bottom-up 的手法である Cocke-Younger-Kasami のアルゴリズム[5] を拡張した誤り訂正アルゴリズムを提案する．さらに，手書き片仮名文字に対するシミュレーション結果を基に本手法の有用性を検討する．

2. では準備として文脈自由型言語に関する種々の定義を与え，構文的パターン認識において重要となる代入変換，削除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.** では文脈自由型誤り訂正文法の構成法を述べる．**4.** では Cocke-Younger-Kasamiのアルゴリズムを拡張した文脈自由型文法の bottom-up 的誤り訂正アルゴリズムを提案する．**5.** では**4.** で提案したアルゴリズムの有用性を示すためにシミュレーションを行ない，**6.** 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6)に従うものとする．

2. 準　備

本節では，構文的パターン認識に必要な種々の定義を行なう．

〔定義1〕 文脈自由型文法とは

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

である．ここで，V_N は非終端記号の有限集合，V_T は終端記号の有限集合を表わし，$V_N \cap V_T=\phi$ である．V_T^* 上の元をストリングという（U^*は空語 λを含む U 上の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い $|\omega|$で表わす．ただし，$|\lambda|=0$ 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われるものである．Pは次のような生成規則の有限集合である．

$\alpha \to \beta$ ここで $\alpha \in V_N,\ \beta \in (V_N \cup V_T)^*$

Sは開始記号を表わし，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，文法 Gによって生成される言語 $L(G)$ の定義を与える．

〔定義2〕 文法 G によって生成される言語（ストリングの集合）を

$$L\ (G)=\{\omega \mid \omega \in V_T^*,\ S \overset{*}{\underset{G}{\Rightarrow}} \omega\}$$

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

〔定義3〕 文脈自由型文法 $G=(V_N,\ V_T,\ P,\ S)$ に

* 大学院電子工学専攻
** 電子工学科

おいて生成規則Pが次の形のいずれかである場合にGは Chomsky 標準形であるという.

(1) $A\to BC$　　$A, B, C\in V_N$

(2) $A\to a$　　$A\in V_N$, $a\in V_T$

次の命題[5]は, 全ての文脈自由型文法が Chomsky 標準形の文法と等価であることを示している.

〔命題1〕 任意の文脈自型文法 Gに対し

$L(G)-\{\lambda\}=L(G_1)$ となる Chomsky 標準形の文脈自由型文法 G_1が存在する.

構文的パターン認識においては代入, 削除, 挿入なる3種類の変換が重要な役割りを演ずることは, 文献1)7)等で知られているが, 本論文では挿入変換を除く代入, 削除の2種類の変換を導入して議論を行なう. ここで, これらの変換の定義を与える.

〔定義4〕 $V_T=\{a_1, a_2, \cdots, a_n\}$ 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

$T_D(a_k)=\lambda$

例えば, $a_{i1}a_{i2}a_{i3}a_{i4}\in V_T^*$, $T_S(a_{i3})=a_{i\mu}$, $i_3\neq i_\mu$ とすると, $a_{i1}a_{i2}T_S(a_{i3})\,a_{i4}=a_{i1}a_{i2}a_{i\mu}a_{i4}$, $a_{i1}T_D(a_{i2})a_{i\mu}a_{i4}=a_{i1}a_{i\mu}a_{i4}$ 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言語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, xからyを導くための変換の列 $T_{i1}T_{i2}\cdots T_{iu}$ (各 j $(1\leq j\leq u)$ に対して $i_j\in\{S, D\}$ 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.

$$d(x, y)=\min_J\{|J|\} \tag{1}$$

但し, $|x|<|y|$ のときは, xをx tail $(y, |y|-|x|)$ として変換を行なう. ここで, tail (y, l) はyの右端から lの長さの部分ストリングを示している,

〔例1〕 $x=cbabb$, $y=cabd$ の変換の一例を示す.

$x=cbabb\to c\ T_D(b)\ abb$

$=cabb\ \to cab\ T_S(b)$

$=cabd=y$

従って, この場合$|J|=2$かつ $d(x,y)=2$ である.

〔例2〕 $x=cba$, $y=cabd$ の場合には

$x=cbad$ とし

$x=cbad\to c\ T_S(b)\ ad$

$=caad\to ca\ T_S(a)\ d$

$=cabd$

従って, この場合 $d(x, y)=3$ である.

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

$$d(x, L(G))=\min\{d(x,y)\,|\,y\in L(G)\} \tag{2}$$

定義6においてyは一般に無限に存在するので$d(x, y)$ も無限に生じるが, 生成規則が有限個であることから, **4.** で述べる誤り訂正パージングにより $d(x, L(G))$ を有限回の操作で求めることができる.

3. 誤り訂正文法の構成

この節では誤り訂正文法 $G'=(V_N\cup\{I\}, V_T, P\cup P', S)$ の構成法について述べる.

〔誤り訂正文法 G'の構成法〕 Chomsky 標準形文脈自由型文法 $G=(V_N, V_T, P, S)$ が与えられたとき

(1) 生成規則 Pの中に $A\to\alpha(A\in V_N, \alpha\in V_T)$ なる形の規則があれば, 全てのαに対し次の生成規則を P'に加える.

a) $A\to\beta$, $\beta\in V_T$, $\beta\neq\alpha$

b) $A\to IA$

c) $I\to\gamma$, $\gamma\in V_T$

(2) P'に次の生成規則を加える.

d) $S\to SI$

e) $I\to II$

a)～e) において a) は代入変換, b)～e) は削除変換に相当する. すなわち, G'は定義4で述べた代入変換および削除変換と等価な生成規則が Pに付加されており, $L(G')$は各 $x\in L(G)$ に対して2種類の変換によって可能な全てのストリングを含むことになる.

以上のことから, $L(G)$ と $L(G')$ について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. 誤り訂正パージング

文脈自由型文法のbottom-up的誤り訂正アルゴリズムに関しては種々の報告[8~10]がなされているが, この節では Cocke-Younger-Kasami のアルゴリズムを基にして, 代入, 削除なる2種類の変換を考慮した誤り訂正パージングアルゴリズムを提案する.

〔アルゴリズム〕

(入力) ストリング $\omega=a_1a_2\cdots a_n$, 誤り訂正文法 G'

(出力) ストリング ωと言語 $L(G)$ との距離 $d(\omega, L(G))$

対 (A, m) $(A\in V_N\cup\{I\}$, m：非負整数) を要素とする Parse Table $[T_{ij}]$ $(1\leq i\leq n,\ 1\leq j\leq n-i+1)$ を次のように構成する.

(1) 各 $i(1\leq i\leq n)$ に対し, $A\to a_i\in P$であれば$(A,0)$

を，$A\to a_i\in P'$ であれば $(A, 1)$ を T_{i1} に加える．

(2) 各 i, j' $(1\leq i\leq n,\ 1\leq j'<j)$ に対し $T_{ij'}$ が完成したとすると，$T_{ij}=\{(C, m+n)|$ ある $k(1\leq k<j)$ に対し $C\to AB\in P\cup P'$ かつ $(A, m)\in T_{ik}$, $(B,n)\in T_{i+k,j-k}\}$

(3) (2) を各 i $(1\leq i\leq n)$ 各 $j(1\leq j\leq n-i+1)$ に対して行なう．

但し，(1) (2) (3) において T_{ij} に対の前の部分が同じものがあれば，後ろの数が最も小さい対以外は削除する．

(4) T_{1n} の中で開始記号をもつ対 (S, l) における l が距離 $d(\omega, L(G))$ である．

Parse Table はまず (1) によって入力ストリング ω と生成規則 $P\cup P'$ を比較しながら底辺の部分 $T_{11}, T_{21}, T_{31}, \cdots, T_{n1}$ が構成される．各 i, j' $(1\leq i\leq n,\ 1\leq j'<j)$ に対して $T_{ij'}$ が完成したとする．次に T_{ij} を構成するには T_{i1} にある対と $T_{i+1,\ j-1}$ にある対を調べ，次に T_{i2} と $T_{i+2,\ j-2}$，その次に $T_{i+3,\ j-3}$ というように調べて行けばよい．結局，セル T_{ij} を構成する場合には

$[T_{i1}, T_{i+1,\ j-1}]$, $[T_{i2}, T_{i+2,\ j-2}]$, $\cdots$, $[T_{i,j-1}, T_{i+j-1,1}]$

なる組を調べる．例えば Fig.1 で示したように，[①, ①'], [②, ②'], [③, ③'] の順で調べて行く．そしていま T_{i1} に対 (A,m)，$T_{i+1,\ j-1}$ に対 (B, n) が存在するなら，$C\to AB$ が $P\cup P'$ の要素であるか否かを調べ，存在すれば T_{ij} に対 $(C, m+n)$ を加える．

$[T_{i2},\ T_{i+2,\ j-2}]\cdots[T_{i,j-1},\ T_{i+j-1,1}]$ に対しても同様に行なう．

T_{ij} に対 (A,m) が存在するということは

$$A\overset{*}{\underset{G}{\Rightarrow}}a_ia_{i+1}\cdots a_{i+j-1} \tag{3}$$

なる導出が可能であることを意味している．これは T_{ij} の要素を調べるときに

$A\to BC\in P\cup P'$ とすると

$$\begin{cases}B\to a_i\\ C\overset{*}{\Rightarrow}a_{i+1}a_{i+2}\cdots a_{i+j-1}\end{cases}$$

$$\begin{cases}B\overset{*}{\Rightarrow}a_ia_{i+1}\\ C\overset{*}{\Rightarrow}a_{i+2}a_{i+3}\cdots a_{i+j-1}\end{cases} \tag{4}$$

$$\vdots$$

$$\begin{cases}B\overset{*}{\Rightarrow}a_ia_{i+1}\cdots a_{i+j-2}\\ C\to a_{i+i-1}\end{cases}$$

これらの導出の組のいずれが可能であるかを

$[T_{i1}, T_{i+1,\ j-1}]$, $[T_{i2}, T_{i+2,\ j-2}]$, $\cdots$, $[T_{i,j-1}, T_{i+j-1,1}]$

の組を調べて，(4)式において一組でも成り立てば T_{ij} の要素に (A, m) を加え

$$A\overset{*}{\underset{G}{\Rightarrow}}a_ia_{i+1}\cdots a_{i+j-1}$$

なる導出が可能になる．

〔例3〕 文脈自由型文法 $G=(V_N, V_T, P, S)$ を $V_N=\{S,A\}$, $V_T=\{a,b\}$, $P=\{S\to AA, S\to AS, S\to b, A\to SA, A\to AS, A\to a\}$ とし，入力ストリングを $\omega=ddab$ とした場合の誤り訂正文法および Parse Table を示す．

誤り訂正文法 $G'=(V_N\cup\{I\}, V_T, P\cup P', S)$

$P'=\{S\to a, S\to c, S\to d, S\to +, S\to\times, S\to(, S\to), S\to *, S\to :, S\to IS, A\to b, A\to c, A\to d, A\to +, A\to\times, A\to(, A\to), A\to *, A\to :, A\to IA, I\to a, I\to b, I\to c, I\to d, I\to +, I\to\times, I\to(, I\to), I\to *, I\to :, S\to SI, I\to II\}$，誤り訂正パージングアルゴリズムにより Parse Table は Fig. 2 のようになる．T_{14} を見ると $(S, 2)$ なる開始記号を持った対があるよって入力ストリング ω と言語 $L(G)$ との距離は d

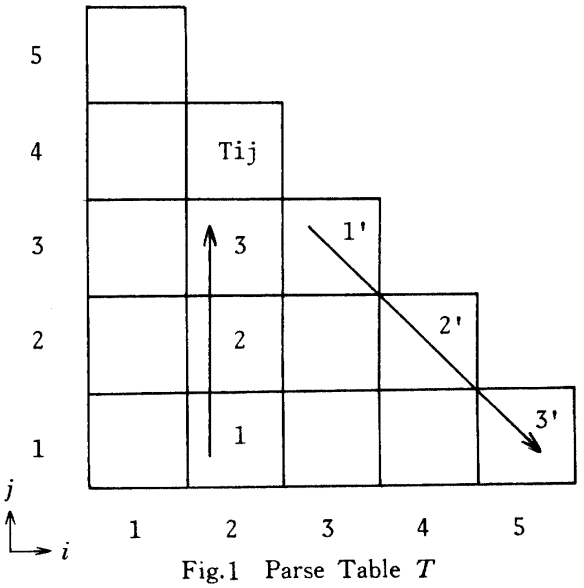

Fig.1 Parse Table T

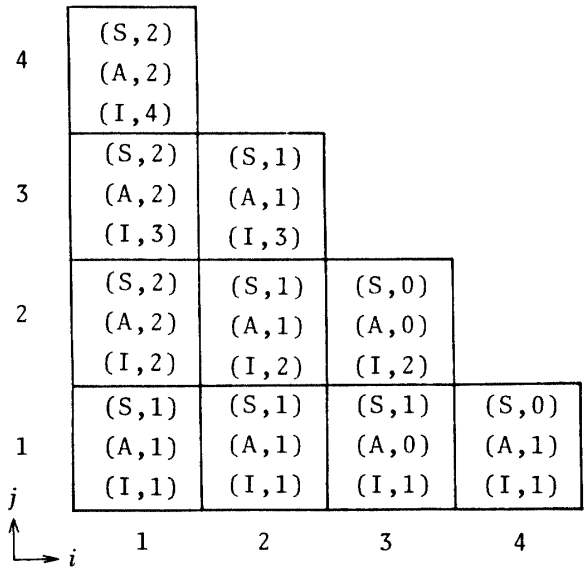

Fig.2 Parse Table for $w=ddab$

$(\omega, L(G))=2$ となる.

5. シミュレーション

4. で提案した誤り訂正パージングアルゴリズムの有用性を検討するために手書き片仮名文字に対するシミュレーションを行なった. シミュレーションに用いた認識システムの概略を Fig.3 に示している.

シミュレーションは, イ, ス, ハ, ル, ヘ, エ, ナ, ヒ, ラ, ト, シ, タ, ソ, ンの14文字について, それぞれ 7～8 個, 全体で 100個のストリングについて行なった. この14文字は, 片仮名の中から比較的構造が簡単で, かつ類似した文字を選んだ. なお, 入力文字のストリング化は文献 11), 文法推定は文献12) に従った.

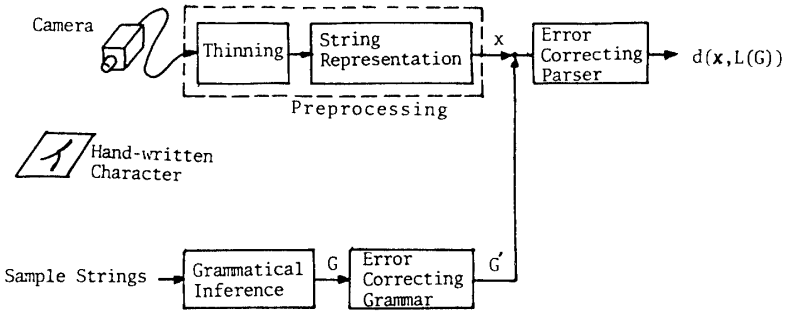

Fig.3 Block diagram of system

手書き片仮名イ, ス, ハ, ル, ヘ, エ, ナ, ヒ, ラ, ト, シ, タ, ソ, ンを入力した場合のシミュレーション結果を Table 1～Table 14に示している. 各 Tableには, それぞれ7～8個のストリングに対する認識結果が示してある. Table の左端の列が未知入力ストリングで第2列から15列がそれぞれのストリング ω と言語 $L(G_i)$ (i=イ, ス, …, ン) との距離で, 右端の列が認識結果である. 認識方法はストリングと最短距離にある言語に未知入力ストリングが属するとした.

Table 1 d(イ, $L(Gi)$)

input string W	d (W , L (Gi))														category
	$G_イ$	$G_ス$	$G_ハ$	$G_ル$	$G_ヘ$	$G_エ$	$G_ナ$	$G_ヒ$	$G_ラ$	$G_ト$	$G_シ$	$G_タ$	$G_ソ$	$G_ン$	
aa+aaxbbb	0	7	6	5	4	9	7	7	7	7	5	5	5	5	イ
aa+axbb	0	7	5	4	3	9	7	6	6	5	5	6	4	4	イ
a+axbb	0	6	4	3	3	6	5	5	6	4	5	6	4	4	イ
baa+dxba	3	5	6	4	5	9	6	5	7	5	5	5	4	5	イ
bba+axcb	3	3	5	4	4	9	5	5	7	3	6	5	3	5	イ,ス,ト,ソ
aa+axbb	0	7	5	4	3	9	7	6	6	5	5	6	4	4	イ
bb+adxcb	4	6	5	4	5	9	6	3	7	3	6	7	4	5	ヒ　ト

Table 2 d(ス, $L(Gi)$)

input string W	d (W , L (Gi))														category
	$G_イ$	$G_ス$	$G_ハ$	$G_ル$	$G_ヘ$	$G_エ$	$G_ナ$	$G_ヒ$	$G_ラ$	$G_ト$	$G_シ$	$G_タ$	$G_ソ$	$G_ン$	
dxba+aaxcc	5	0	9	7	4	8	7	7	4	6	7	5	5	6	ス
ddxba+aaxc	5	0	9	7	5	6	7	7	3	6	8	6	5	6	ス
dxa+axc	3	0	7	5	3	8	5	6	4	4	5	5	4	4	ス
dxba+axc	4	0	7	5	4	8	6	6	3	4	6	8	4	5	ス
dxbb+axcd	6	2	7	6	6	7	6	5	4	4	7	5	5	7	ス
dxbb+axc	5	1	6	5	5	8	6	5	4	3	6	8	4	6	ス
a+d+dxa+aacc	7	4	12	10	7	6	7	9	7	9	8	5	8	7	ス

Table 3 d(ハ, L (Gi))

input string W	d (W , L (Gi))														category
	$G_イ$	$G_ス$	$G_ハ$	$G_ル$	$G_ヘ$	$G_エ$	$G_ナ$	$G_ヒ$	$G_ラ$	$G_ト$	$G_シ$	$G_タ$	$G_ソ$	$G_ン$	
bbc:bbb	4	7	0	1	6	8	6	5	6	4	5	6	3	6	ハ
bbbb:bb	5	6	0	1	7	9	6	4	5	4	6	7	2	6	ハ
bb:ba	5	6	1	1	5	9	5	5	5	4	5	7	2	4	ハ, ル
bbbc:bbba	6	8	1	1	8	9	7	5	7	4	7	7	4	8	ハ, ル
bbcd:bbb	5	8	1	2	7	9	6	5	6	5	6	6	4	6	ハ
bbb:bbaa	6	7	2	2	8	8	6	4	6	3	7	7	4	7	ハ, ル
bbc:bba	5	7	1	1	6	8	6	5	6	4	5	6	3	6	ハ, ル

Table 4 d(ル, L (Gi))

input string W	d (W , L (Gi))														category
	$G_イ$	$G_ス$	$G_ハ$	$G_ル$	$G_ヘ$	$G_エ$	$G_ナ$	$G_ヒ$	$G_ラ$	$G_ト$	$G_シ$	$G_タ$	$G_ソ$	$G_ン$	
bbba:bbb	4	6	1	0	7	9	7	5	5	4	6	7	2	6	ル
bbbbd:bb	6	7	1	1	8	9	7	4	6	4	6	7	2	6	ハ, ル
bba:bb	3	6	1	0	5	8	6	5	5	4	5	8	1	4	ル
bbbcd:bb	6	7	1	2	7	9	7	4	6	5	6	7	3	6	ハ
bbd:b	4	7	1	1	5	7	5	5	4	4	3	8	1	3	ハ, ル, ソ
bbbc:bb	5	6	0	1	6	9	6	4	5	4	5	7	2	6	ハ
bbca:bbb	4	7	1	1	7	9	7	6	6	5	6	6	3	6	ハ, ル

Table 5 d(ヘ, L (Gi))

input string W	d (W , L (Gi))														category
	$G_イ$	$G_ス$	$G_ハ$	$G_ル$	$G_ヘ$	$G_エ$	$G_ナ$	$G_ヒ$	$G_ラ$	$G_ト$	$G_シ$	$G_タ$	$G_ソ$	$G_ン$	
aaxcc	4	7	5	4	0	9	7	6	5	4	3	7	3	2	ヘ
axcc	4	6	4	3	0	9	7	5	4	5	3	6	3	2	ヘ
axccc	4	6	4	4	0	9	7	5	4	4	2	6	4	3	ヘ
b+adxccc	5	5	6	7	3	8	4	4	6	4	4	7	5	5	ヘ
bbxccc	6	7	3	4	2	9	6	5	5	3	4	8	3	5	ヘ
a+bxccdd	5	7	6	6	4	6	4	4	5	4	5	6	6	6	ヘ, ナ, ヒ, ト
abxccc	5	7	4	5	1	9	7	6	5	4	3	7	4	4	ヘ

Table 6 d(エ, L (Gi))

input string W	d (W , L (Gi))														category
	$G_イ$	$G_ス$	$G_ハ$	$G_ル$	$G_ヘ$	$G_エ$	$G_ナ$	$G_ヒ$	$G_ラ$	$G_ト$	$G_シ$	$G_タ$	$G_ソ$	$G_ン$	
daxdxbb+adxdd	9	6	11	9	10	2	5	6	6	8	9	8	10	9	エ
dd+ddxbb+adxcdd	11	5	12	12	12	1	5	8	6	10	11	8	11	12	エ
ad+dxbb+dxddd	8	7	11	10	11	1	4	5	5	8	9	9	11	10	エ
dd+dxbb+adxddd	10	6	12	11	12	0	4	6	5	9	10	8	11	11	エ
da+dxb+axdd	7	4	10	8	8	2	3	6	4	7	8	5	9	8	エ
dd+dxbbb+adxdd	9	6	11	10	12	0	4	6	6	8	11	8	10	11	エ
dd+ddxbb+adxdd	10	5	12	11	12	0	4	7	5	9	11	7	11	11	エ

Table 7 d(ナ, L (Gi))

input string W	d (W , L (Gi))														category
	$G_イ$	$G_ス$	$G_ハ$	$G_ル$	$G_ヘ$	$G_エ$	$G_ナ$	$G_ヒ$	$G_ラ$	$G_ト$	$G_シ$	$G_タ$	$G_ソ$	$G_ン$	
bb+ddxbaxd	7	7	7	6	8	6	1	4	4	4	8	8	6	8	ナ
bb+ddxbaxd	7	7	7	6	8	6	1	4	4	4	8	8	6	8	ナ
b+dxbaxdd	6	6	7	6	7	4	0	3	3	4	7	6	6	7	ナ
b+dddxbaaxd	7	4	9	8	8	3	1	5	3	6	8	7	7	8	ナ
b+dxbaxd	5	6	6	5	6	5	0	3	3	3	6	7	5	6	ナ
b+ddxbbadxd	7	5	8	7	9	2	1	4	5	5	8	9	7	8	ナ
b+dxbaxd	5	6	6	5	6	5	0	3	3	3	6	7	5	6	ナ

Table 8 d(ヒ, L (Gi))

input string W	d (W , L (Gi))														category
	$G_イ$	$G_ス$	$G_ハ$	$G_ル$	$G_ヘ$	$G_エ$	$G_ナ$	$G_ヒ$	$G_ラ$	$G_ト$	$G_シ$	$G_タ$	$G_ソ$	$G_ン$	
b+abbddxd	6	7	6	6	7	4	4	1	5	3	6	8	6	6	ヒ
add+abcddxb	6	7	8	8	8	6	7	5	6	8	7	8	9	8	ヒ
aaa+bxbcddd	5	7	8	8	6	6	6	5	6	7	6	7	8	6	イ, ヒ
b+bbbdddxda	9	9	7	6	10	6	6	1	6	4	8	9	7	9	ヒ
bb+bcddxdd	8	8	6	7	9	6	5	0	5	4	8	7	7	8	ヒ
aaa+cxdbdd	4	7	8	8	6	7	6	5	7	7	5	6	7	5	イ
b+bbcdddxddd	10	10	8	9	11	6	5	0	5	6	9	9	9	10	ヒ

Table 9 d(ラ, L (Gi))

input string W	d (W , L (Gi))														category
	$G_イ$	$G_ス$	$G_ハ$	$G_ル$	$G_ヘ$	$G_エ$	$G_ナ$	$G_ヒ$	$G_ラ$	$G_ト$	$G_シ$	$G_タ$	$G_ソ$	$G_ン$	
dddxbaa:ad	8	4	8	6	8	4	5	8	1	8	6	6	6	6	ラ
dddd:dddxbaa	10	7	10	9	11	7	6	7	3	11	8	7	10	9	ラ
dddxbaa:add	8	3	9	7	8	4	5	8	1	9	7	7	7	7	ラ
ddxbaa:ad	7	4	7	5	7	7	7	7	1	7	6	6	5	5	ラ
dddxba:dd	7	4	7	6	8	4	3	6	0	7	5	7	6	6	ラ
dddxbaa:dd	8	4	8	7	8	4	4	7	0	8	6	7	6	6	ラ
dddxbaaa:dd	8	4	9	8	8	4	5	8	0	9	7	7	6	6	ラ

Table 10 d(ト, L(Gi))

input string W	d (W , L (Gi))														category
	$G_イ$	$G_ス$	$G_ハ$	$G_ル$	$G_ヘ$	$G_エ$	$G_ナ$	$G_ヒ$	$G_ラ$	$G_ト$	$G_シ$	$G_タ$	$G_ソ$	$G_ン$	
bb+bbxc	5	5	3	4	5	8	5	2	6	0	6	7	3	6	ト
bb+bbxcc	6	5	4	4	5	8	6	3	7	1	7	7	3	7	ト
bb+abxc	4	5	4	4	4	8	5	3	6	1	6	7	3	5	ト
cc+bbxdd	6	7	5	6	7	7	5	3	5	3	7	7	6	7	ヒ, ト
bb+bbxc	5	5	3	4	5	8	5	2	6	0	6	7	3	6	ト
bb+bbxd	5	6	3	4	6	7	4	1	5	0	6	7	4	6	ト
bb+bbbxc	6	5	3	4	6	8	5	2	7	0	7	8	3	7	ト

Table 11 $d(シ, L(Gi))$

input string W	d (W , L (Gi))														category
	$G_イ$	$G_ス$	$G_ハ$	$G_ル$	$G_ヘ$	$G_エ$	$G_ナ$	$G_ヒ$	$G_ラ$	$G_ト$	$G_シ$	$G_タ$	$G_ソ$	$G_ン$	
aaad:c:c	5	6	6	6	3	9	7	7	6	7	0	7	4	2	シ
baaaa:c:c	5	5	6	6	3	9	7	8	6	7	1	8	2	3	シ
ddd:d:d	7	6	6	6	7	6	5	5	2	6	2	8	6	4	ラ, シ
aa:c:c	4	7	4	4	2	9	7	6	5	5	0	7	3	2	シ
aaad:c:d	5	7	6	6	4	9	6	6	5	7	0	7	4	2	シ
aa:d:c	4	7	5	4	3	9	7	5	5	5	0	7	3	1	シ
aaa::	3	6	4	3	2	9	7	6	4	5	3	7	2	1	ン

Table 12 $d(タ, L(Gi))$

input string W	d (W , L (Gi))														category
	$G_イ$	$G_ス$	$G_ハ$	$G_ル$	$G_ヘ$	$G_エ$	$G_ナ$	$G_ヒ$	$G_ラ$	$G_ト$	$G_シ$	$G_タ$	$G_ソ$	$G_ン$	
ddda+c+axaa*	7	6	10	9	8	8	8	8	7	9	9	0	8	8	タ
da+c+axab*	5	6	8	8	7	8	8	8	7	9	9	0	7	7	タ
dda+c+aaxaab*	5	6	11	11	8	8	10	10	7	11	10	0	8	8	タ
dda+cc+aaxaa*	8	7	12	11	9	8	10	10	8	11	10	0	9	9	タ
dd+c+baxab*	7	6	8	8	9	7	7	8	6	8	9	2	8	9	タ
ddb+axc+bxab*	8	7	9	9	10	7	8	9	8	9	11	4	10	11	タ
dda+c+bxab*	7	7	8	8	9	7	8	8	7	8	9	1	9	9	タ
da+c+axab*	5	6	8	8	7	8	7	7	7	8	8	0	7	7	タ

Table 13 $d(ソ, L(Gi))$

input string w	d (W , L (Gi))														category
	$G_イ$	$G_ス$	$G_ハ$	$G_ル$	$G_ヘ$	$G_エ$	$G_ナ$	$G_ヒ$	$G_ラ$	$G_ト$	$G_シ$	$G_タ$	$G_ソ$	$G_ン$	
bbaa:b	3	6	2	1	4	9	6	5	4	4	5	8	0	3	ソ
baaa:b	3	6	3	2	3	9	6	5	4	5	4	8	0	3	ソ
baba:c	4	6	3	2	3	9	6	4	3	3	3	8	1	2	ソ
bbba:c	5	6	2	1	4	9	6	4	3	3	4	8	0	3	ソ
bbaa:c	4	6	3	2	3	9	6	5	4	5	4	8	0	2	ソ
abaaa:c	3	4	5	4	2	9	6	6	4	5	3	6	1	1	ソ, ン
babaa:b	3	5	3	2	4	9	5	5	3	5	5	6	1	3	ソ
babaa:c	4	4	4	3	2	9	5	5	3	4	4	7	1	2	ソ

Table 14 $d(ン, L(Gi))$

input string W	d (W , L (Gi))														category
	$G_イ$	$G_ス$	$G_ハ$	$G_ル$	$G_ヘ$	$G_エ$	$G_ナ$	$G_ヒ$	$G_ラ$	$G_ト$	$G_シ$	$G_タ$	$G_ソ$	$G_ン$	
aa:c	4	7	3	2	1	9	7	6	5	5	4	7	1	0	ン
aaa:c	3	6	4	3	1	9	7	6	4	4	2	7	1	0	ン
baad:c	5	6	4	3	3	9	6	4	5	4	2	8	1	1	ソ, ン
cbaadd:d	6	6	6	5	6	9	6	4	6	6	3	8	4	3	シ, ン
aaa+adxcd:c	5	5	9	9	4	9	8	7	8	4	3	7	6	4	シ
aaaa:c	3	6	5	4	1	9	7	6	4	5	2	7	1	0	ン
aa:c	4	7	3	2	1	9	7	6	5	5	4	7	1	0	ン

6. シミュレーションに対する検討

Table 1～Table 14によると混読16，誤読5で，全体として認識率は79%であった．認識率が特に低かったのは，イ，ハ，ル，ソ，ンの5文字で，イの場合は手書きの癖が多かったため，又，ト等と混読される結果となった．ハとル，ソとンは $L(G_{ハ})$ と $L(G_{ル})$，$L(G_{ッ})$ と $L(G_{ン})$ がそれぞれ類似しており互いのストリングと言語との距離が小さくなっているからと考えられる．類似した文字ハとル，ソとンにおいて，ハとンを除いた認識率は92%である．

以上のシミュレーション結果より認識率を向上させるには，

（1） ストリング化する際の基本要素数[11]をふやす．
（2） 前処理段階で入力文字を2値化する際のメッシュ数をふやす（現在20×20）．
（3） 生成規則[12]をふやすことにより類似している文字の互いのストリングと言語との距離をなるべく大きくする．
（4） 生成規則に重みを付ける．

などが考えらる．（1）～（3）は，入力ストリング長に大きく影響し，結局，計算時間* にも影響することになる．なお距離 $d(\omega, L(G))$ を求めるために要した時間は ω と G によって多少異なるが，平均すると約30秒（ミニコン NOVA システムモデル 01，FACOM M-200では約0.3秒）であり，文献1）と比較するとかなり短縮されている．また認識率が文献1）とほぼ同程度であることから，top-down 的な文献1）の手法よりも本手法の方がより有効であると考えられる．さらに，本論文では挿入変換は考慮していないが手書き片仮名文字においては代入，削除の2種類の変換を用いれば十分であると考えられる．

7. むすび

本論文ではCocke-Younger-Kasamiのアルゴリズムを基にし，代入，削除なる2種類の変換を導入した誤り訂正パージングアルゴリズムを提案した．さらに，手書き片仮名文字認識に対するシミュレーション結果から本手法の有用性を明らかにした．また，代入，削除挿入なる3種類の変換を用いた手法[1]に比べ，本論文の手法は認識率を低下させることなく計算時間の点で優れていることがわかった．

本論文では，手書き片仮名文字を対象にシミュレーションを行なったが，本手法は平仮名，アルファベット等の他の手書き文字のみならず，ストリングへ変換可能なあらゆる図形認識に有効であると思われる．

また，6. で述べた認識率向上のための種々の考察および計算時間に関する理論的な立場からの詳細な議論は今後の課題である．

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します．

文　献

1) 森田，金岡，富田：“文脈自由型誤り訂正文法を用いた手書き片仮名文字認識について”，信学技報 PRL80-89.
2) K.S. Fu and S.Y. Lu: “A Clustering Procedure for Syntactic Patterns”, IEEE Trans. on Syst., Man and Cybern., **SMC-7**, 734 (1977)
3) S.Y. Lu and K.S.Fu: “Error-correcting Tree Automata for Syntactic Pattern Recognition”, IEEE Trans. on Compt., **C-27**, 1040 (1978)
4) 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)
5) A.V. Aho and J.D. Ullman: “The Theory of Parsing, Translation and Compiling”, Prentice Hall Inc., Vol.1 (1972)
6) 本田波雄：“オートマトン・言語理論”，コロナ社 (1978)
7) A.V. Aho and T.G. Peterson: “A Minimum Distance Error-Correcting Parser for Context-Free Languages”, SIAM J. Compt., **1**, 305 (1972)
8) 岩元，沢野：“文脈自由語の誤り訂正”，信学誌，**56-D**, 675 (1973)
9) L.W. Fung and K.S. Fu: “Maximum-Likelihood Syntactic Decoding”, IEEE Trans. on Information Theory **IT-21**, 423 (1975)
10) 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)
11) 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)
12) 上本，甲斐，他：“手書き片仮名文字認識のための文法推論”，山口大学工学部研究報告，**32**, 315 (1982)

（昭和56年10月15日　受理）

* 計算時間 $d(\omega, L(G))$ は $O(n^3)$ であることが知られている[5]．