手書き片仮名文字を対象としたストリング化の一手法

衣松　博*・浜本義彦**・金岡泰保***・富田真吾***・岡田敏彦***

String Representation of Handwritten Katakana Characters

Hiroshi KINUMATSU, Yoshihiko HAMAMOTO, Taiho KANAOKA, Shingo TOMITA and Toshihiko OKADA

Abstract

In syntactical pattern recognition, a pattern must be represented by a sentence (or string) in a language. So string representation of a input pattern is important pre-processing for syntactical pattern recognition. In this paper, we propose a method of string representation for handwritten katakana characters. Each character is a continuous line pattern on a 20×20 mesh. Starting from its upper right corner, each input pattern is chain-coded cell by cell. Namely, each successive cell is coded as A, B, C or D according to its position relative to that of the current one. After three consecutive cells have been coded one of four primitives a, b, c and d, by using four concatenation relations ×, +, *,: and the parentheses ( ), the concatenation of branches, which is a string representation, completes. From some experimental results, the validity of our method is discussed.

1. まえがき

近年，木文法，文脈自由型文法等のいわゆる形式文法を用いた構文的パターン認識に関する理論的実験的な興味ある報告がなされている[1)~3)]．

構文的パターン認識は主に

（1） 入力パターンの細線化[4)]

（2） ストリング化[5)]

（3） 文法推論[6)]

（4） 誤り訂正パージング[1),7)]

といった一連の過程を経て行われるが，特に，(1) と (2) は前処理過程として重要な位置を占めている．

本論文の目的は，手書き片仮文字を対象に，細線化された入力パターンのストリング化法を確立し，シミュレーション結果を基にその有用性を検討することである．

2. 構文的パターン認識機構におけるストリング化過程

構文的パターン認識機構は概略 Fig. 1 のようなブロック図で表されるが，前処理過程としてのストリング化は細線化された二次元パターンを一次元の記号系列として表示する過程である．

情報圧縮という観点からすると，ストリング化では，プリミティブの数，連接記号の数およびストリングの長さをできるだけ小さくして後の過程における処理を容易にすることが要求される．認識率との兼合いからの最適なプリミティブ数，連接記号の数およびストリング長に関する考察は興味ある課題であるが，このことは場を改めて論じることにして，本論文では当面の

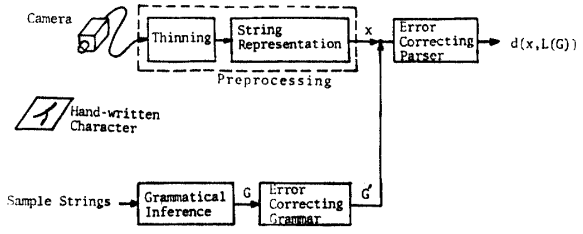

Fig. 1 Block diagram of system.

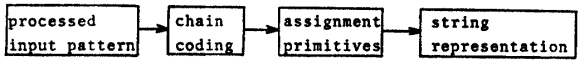

Fig. 2 Procedure of string representation.

* 電子工学科（現，日本製鋼所）
** 大学院電子工学専攻
*** 電子工学科

課題であるストリング化の方法論の確立ということに焦点を当てて議論する.

3. 手書き片仮名文字のストリング化

ストリング化を行うには Fig. 2 に示すように細線化された入力文字パターンのチェインコード化, プリミティブ化等の処理が必要となるが, この章では, 本論文で提案する, 手書き片仮名文字を対象とした, ストリング化法の詳細について述べる.

3.1 チェインコード化

まず, Fig. 3.1 のような細線化された2値パターンを Fig. 3.2 のようにある点から始めて逐次その前の点に対する相対的位置関係に従って記号化していく. このとき, ある点 x の相対的位置関係として Fig. 4 のように x の近傍の8点を A, B, C, D の4個の方向記号で表わす. さらに, A, B, C, D の方向記号とは別に, 点 x と連続した点が x の8近傍内に2個以上あるときは点 x を特異点といい S で表わす. すなわち, 点 x が特異点の場合は A, B, C, D の何れかの記号と S が2重に記されることになる. 従って, 細線化された入力パターンの1要素は全て A, B, C, D の何れかに変換され, さらに特異点には S が記される. (但し, 始点が特異点のときは S のみを記し, そうでないときは B を記す.)

チェインコード化とはこのように変換された二次元パターンを, 新たに端点を表す記号 E を追加して, 一次元の記号系列に変換することをいう. Fig. 3.2 の例では, 右上から始めて, 特異点から分岐した枝を左の枝から優先的にチェインコード化すると

$$EBAAAASAAAESCBBBBBBBE$$

となる. ここで, 左から6番目の A は5番目の A に対する7番目の特異点 S の方向を示しており, 12番目の S は7番目の S と同等で, $CBB\cdots$ なる枝は7番目の S から出ていることを示している.

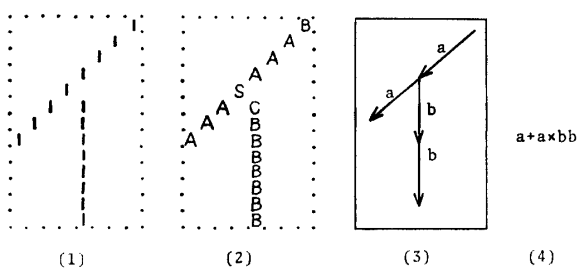

Fig. 3 Illustrative example of procedure of string representation.

C	B	A
D	X	D
A	B	C

Fig. 4 Symbols of 8-neighbors.

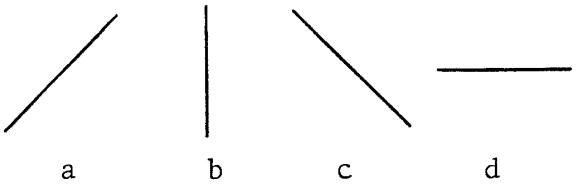

Fig. 5 Primitives.

3.2 プリミティブ化

$K=\{A, B, C, D\}$, $Z=\{E, S\}$ とし, プリミティブの集合を $P=\{a, b, c, d\}$ とおく. 今,

$$\omega=r_1u_1r_2u_2\cdots r_{n-1}u_{n-1}r_n \tag{1}$$

但し, $\omega\in\{K\cup Z\}^*$, $r_i\in Z^*$, $u_i\in K^*$, $|u_i|\geqq 1$ $(1\leqq i\leqq n)$ であり, 記号系列 η に対して $|\eta|$ は η の長さを, U^* は集合 U のクリーネ閉包を示す.

とすると, 各 $u_i(1\leqq i\leqq n-1)$ に対して

$$f(u_i)=t_i \tag{2}$$

但し, f は単射で $t_i\in P\cup\lambda$ (λ は空語).

なる変換 f がプリミテッブ化である.

Fig. 3 の例では

$$\omega=Eu_1Su_2ESu_3u_4E$$

但し, $u_1=BAAAA$, $u_2=AAA$, $u_3=CBB$, $u_4=BBBBB$ であり, $f(u_1)=a, f(u_2)=a, f(u_3)=b, f(u_4)=b$ とおいて

$$\tilde{\omega}=EaSaESbbE \tag{3}$$

を得る.

プリミティブ化では, 式 (1) で示される ω の部分系列化 (又は分割の仕方) と式 (2) の写像 f の定め方が重要な問題となる. このことに関して, 本論文では基本方針を次のように示している. 今, $u_i=\sigma_1\sigma_2\cdots\sigma_m$ (各 i $(1\leqq i\leqq m)$ に対して $\sigma_i\in K$) とする. 前から3個づつを順次プリミティブに置き換えるが, $m=3r+1$ $(r=0, 1, 2,\ldots)$ のときは $f(\sigma_m)=\lambda$, $m=3r+2$ のときは $f(\sigma_{m-1}\sigma_m)$ は σ_m に対応するプリミティブに

Table 1　Assignment primitives
(x is not defined)

u_i	$f(u_i)$	u_i	$f(u_i)$	u_i	$f(u_i)$	u_i	$f(u_i)$
AAA	a	AAB	a	AAC	a	AAD	a
ABA	a	ABB	b	ABC	b	ABD	x
ACA	a	ACB	b	ACC	c	ACD	d
ADA	a	ADB	x	ADC	d	ADD	d
BAA	a	BAB	b	BAC	b	BAD	a
BBA	b	BBB	b	BBC	b	BBD	b
BCA	b	BCB	b	BCC	c	BCD	c
BDA	d	BDB	x	BDC	d	BDD	d
CAA	a	CAB	b	CAC	c	CAD	d
CBA	b	CBB	b	CBC	c	CBD	x
CCA	c	CCB	c	CCC	c	CCD	c
CDA	d	CDB	x	CDC	c	CDD	d
DAA	a	DAB	a	DAC	d	DAD	d
DBA	x	DBB	x	DBC	x	DBD	x
DCA	d	DCB	c	DCC	c	DCD	d
DDA	d	DDB	x	DDC	d	DDD	d

置き換える．すなわち，$\sigma_{m-1}\sigma_m=AB$ なら $f(\sigma_{m-1}\sigma_m)=B$ とする．従って，u_i をストリング化したときのストリングの長さは $n=3r$ および $3r+1$ のときは r で，$n=3r+2$ のときは $r+1$ である．$|u_i|=3$ のときの各 $u_i\in K^*$ に対する $f(u_i)$ は Table 1 のように定めている．

3.3. 部分系列間の連接関係を表わすオペレータとストリング化

プリミティブ化が完了した段階では，入力パターンは式 (3) のようにプリミティブと特異点および端点を表わす記号になる一次元の記号系列として表わされている．ストリング化はこの記号系列から E と S を削除し，プリミティブよりなる部分系列をオペレータによって連接することによりなされる．本論文では，部分系列間の連接関係を示すものとして，( )，×，＋，＊，：の順に優先順位をもつ5種類のオペレータを用いる．以下，$Q=\{(\ ),\ \times,\ +,\ *,\ :\}$, $Z=P\cup Q$ とし，ストリング $S_1, S_2,\cdots, S_l$ ($S_i\in Z^*$ ($1\leqq i\leqq l$), $l\geqq 2$) の連接関係を $[S_1, S_2,\cdots, S_l]$ で示す．

3.3.1. オペレータ ( )

×，＋，＊および：は部分系列間の位置関係を示すオペレータであるが，( )は $(S_1\Delta_1S_2\Delta_2\cdots\Delta_lS_l)\ \Delta_{l+1}S_{l+1}$ のように S_{l+1} と Δ_{l+1} なる連接関係にある部分系列を指定するオペレータである．すなわち，$(S_1\Delta_1S_2\Delta_2\cdots\Delta_lS_l)\Delta_{l+1}S_{l+1}$ はストリング $S_1\Delta_1S_2\Delta_2\cdots\Delta_lS_l$ と S_{l+1} は Δ_{l+1} なる関係にあることを意味している．但し，$\Delta_i\in Q$ ($1\leqq i\leqq l+1$).

3.3.2 オペレータ×

S_1 と S_2 の始点と終点が Fig. 6.1 のような関係にあるとき

$$[S_1, S_2]=S_1\times S_2$$

とする．

3.3.3 オペレータ＋

S_1 と S_2 が Fig. 6.2 のような関係にあり，しかもストリング S_2 に Q の元が含まれるとき，

$$[S_1, S_2]=S_1+S_2,$$

とする．但し，始点から矢の方向に向って右側のストリングを優先するから $S_1+S_2\neq S_2+S_1$ である．

3.3.4 オペレータ＊

S_1 と S_2 が Fig. 6.3 のような連接関係にあるとき

$$[S_1, S_2]=\begin{cases}S_1S_2^* & ,\ S_2\in P^*\\ S_1+S_2^*, & S_2 \text{ の中に } Q \text{ の元が含まれるとき}\end{cases}$$

とし，ストリング S_1 の最左端のプリミティブの始点

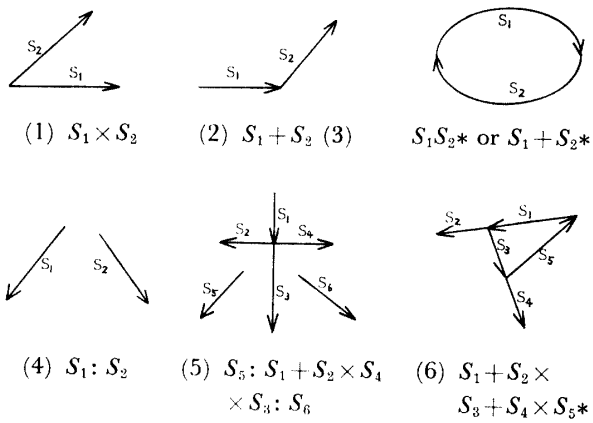

Fig. 6　Concatenation relations of branches.

と S_2 の最右端の プリミティブの終点が一致していることを表す.

3.3.5 オペレータ：

Fig. 6.4 のようにストリング S_1 と S_2 が離れているとき

$$[S_1, S_2] = S_1 : S_2$$

とし, S_2 が S_1 の右側に位置していることを表わす.

Example (1) Fig. 6.5 の例では

$$[S_1, S_2, \ldots, S_6] = S_5 : S_1 + S_2 \times S_4 \times S_3 : S_6$$

(2) Fig. 6.6 の例では

$$[S_1, S_2, \ldots, S_5] = S_1 + S_2 \times S_3 + S_4 \times S_5 *$$

4. シミュレーション結果および検討

3章で述べたストリング化法を用いてシミュレーションを行なった. メインプログラムは Fig. 7 に示したフローチャートに従って実行される. シミュレーション結果を Fig. 8 と Table 2 に示している. Fig. 8 には, 細線化された手書き片仮名文字に A, B, C, D, S の記号を割当てたものと, さらにこれをストリング化したいくつかの例を示している. また, Table 2 には全ての片仮名文字のストリング化例を示している.

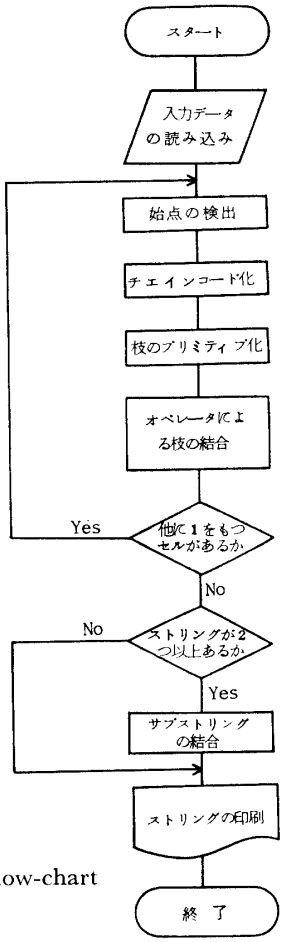

Fig. 7. Flow-chart

Table 2 Example of string representation for handwritten katakana characters.

		string representation			string representation
1	ア	$dddxba+bxaad$	6	カ	$b+ddxbaxddbba$
2	イ	$aa+axbbb$	7	キ	$b+(axb+ddxcxd)xa$
3	ウ	$b+ddbxddbaaa$	8	ク	$ddaaxbaaa$
4	エ	$a+dxbb+ddxdd$	9	ケ	$b+axdd+baxdd$
5	オ	$b+dxaabbbxd$	10	コ	$dddxbbaddd$
11	サ	$b+(d+dxbxb)xbbbxd$	16	タ	$dda+axc+aaxab*$
12	シ	$baaa:d:d$	17	チ	$a+axb+dxbxdd$
13	ス	$dddxba+aaxcc$	18	ツ	$baaa:b:b$
14	セ	$b+dxbbdddxdd$	19	テ	$dddd:dd+ddxaa$
15	ソ	$bbaaa:b$	20	ト	$bb+bbxc$
21	ナ	$bb+ddxbbxdd$	26	ハ	$bcb:bbb$
22	ニ	$ddddd:dd$	27	ヒ	$b+bbcdddxdd$
23	ヌ	$ddxba+cxaxc$	28	フ	$dddxbaaa$
24	ネ	$b+ddxdda+daxbbxc$	29	ヘ	$aaxccc$
25	ノ	$baaa$	30	ホ	$b:bb+ddxbbbxdd:a$
31	マ	$ddddxba+cxc$	36	ヤ	$bb+axbbxdadb$
32	ミ	$cc:b:cc$	37	ユ	$dxbb+adxcd$
33	ム	$aaabddd+cxb$	38	ヨ	$dddxbb+dddxbbddd$
34	メ	$aa+cxaxc$			
35	モ	$ad+ddxb+dxbcddxdd$			
39	ラ	$dddd:ddddxbaa$	44	ワ	$dddaxbaaa$
40	リ	$bbba:bb$	45	ン	$baaa:c$
41	ル	$bbbbaa:bbb$			
42	レ	$bbbbaa$			
43	ロ	$ddddbbcdddabb*$			

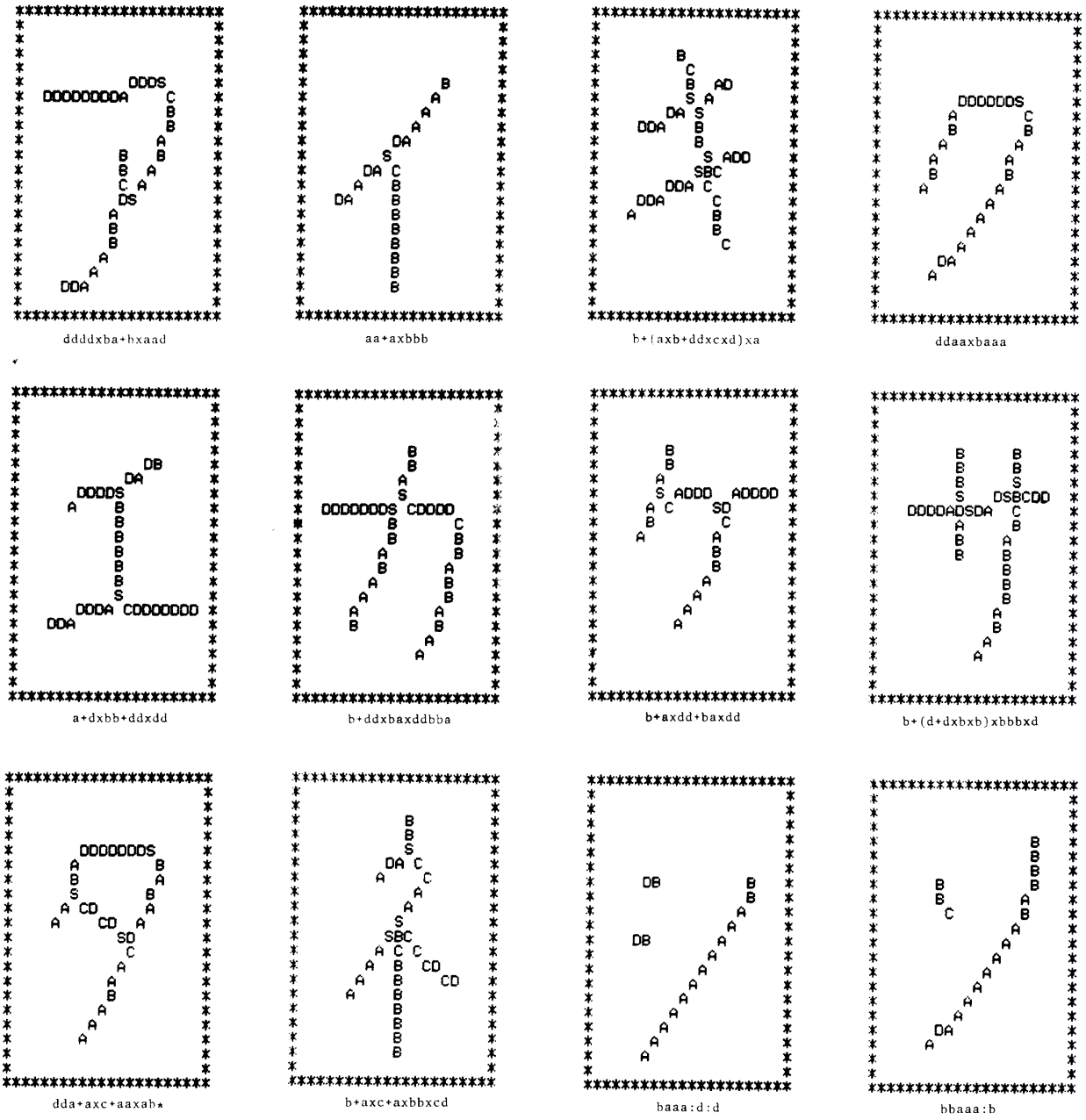

Fig. 8　Character patterns and their string representations.

ストリング化に本手法を用いて手書き片仮名文字の認識実験を行ったが[3]，認識率は，イ，ス，ル，ヘ，ト，シ，ナ，ラ，ヒの9文字各々6個の計54文字に対して95%という良好な結果が得られており，このことから，本手法は手書き片仮名文字に対して有効であろうと考えられる．

本論文では，チェインコード化の段階で8近傍を4個の記号で表わし，さらに，方向性を考慮しない4個のプリミティブを採用したために，Table 2 の「リ」，「ル」等のようにストリングで見る限りは区別がつかない場合が生じる．すなわち，本手法によると入力パターンに対してストリングは一意に決まるが，逆にストリングから入力パターンは一般に一意には決まらない．工学的には，入力データはできるだけ情報の圧縮を施すことが望ましいが，情報を圧縮しすぎたためにこのような問題が生じたものと考えられる．プリミティブに方向性を持たせることによりこの問題は解消されるものと思われる．

始点は，メッシュの最上行から下の行へ向って右から左へ操作して，最初の1の点としたが，入力パターンの僅かの相違でストリングが大きく変化することがある．このことは後の文法推論とも密接に関連しており，始点の採り方も今後の課題である．

なお，シミュレーションで使用した計算機は日本ミ

ニコンピュータ NOVA システムモデル01で，細線化された入力パターン1個をストリング化するのに要した時間は約2.5秒である．

謝　辞

本研究に際し，有益なる御意見をお寄せいただいた本学高浪五男教授，井上克司助教授ならびに御討論いただいた富田研の諸氏に深謝する．

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 (1977).
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) 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，PRL 80-89, (1981) p. 5.
4) 田村秀行："細線化法についての諸考察"，信学技報，PRL 75-66, (1975)
5) A.S. Shaw: "A formal picture description sheme as a basis for picture processing systems" Inform. Contr., vol. 14, (1969) p. 9.
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 T.G. Peterson: "A minimum distance error-correcting parser for context-free languages", SIAM J. Compt., **1**, 305 (1972)

(昭和56年4月15日　受理)