T2R2 東京工業大学リサーチリポジトリ
Tokyo Tech Research Repository

論文 / 著書情報
Article / Book Information

題目(和文)	漢字パターンの表現法とその応用に関する基礎的研究
Title(English)	
著者(和文)	長橋宏
Author(English)	HIROSHI NAGAHASHI
出典(和文)	学位:工学博士, 学位授与機関:東京工業大学, 報告番号:甲第1195号, 授与年月日:1980年3月26日, 学位の種別:課程博士, 審査員:
Citation(English)	Degree:Doctor of Engineering, Conferring organization: , Report number:甲第1195号, Conferred date:1980/3/26, Degree Type:Course doctor, Examiner:
学位種別(和文)	博士論文
Type(English)	Doctoral Thesis

Powered by T2R2 (Tokyo Institute Research Repository)

漢字パターンの表現法とその応用に関する基礎的研究

指導教官　　安居院　猛　助教授

提出者　　大学院博士課程　物理情報工学専攻

長橋　宏

東京工業大学 437973

目次

第一章 序論[1),(2)]

歯車の自動計数に始まったディジタル計算機が、続いて継電器、そして真空管を利用した電子計算機へと発展したのは、僅か30数年前の1940年代の初めである。1946年、フォン・ノイマンによってストアドプログラム方式が提案されたことにより、多くの大学、研究所等で盛んにコンピュータの開発が行われ、計算機の第一世代を迎える。第一世代の各所で研究された成果は、第二世代に入って、磁気コア、トランジスタ、パラメトロン等の素子として出現した。

一方、新しい素子の開発によって演算速度が高速化し、それに伴って生じてきた入出力装置と演算装置間の速度差の問題を解決するために、割り込み機能やデータチャンネルが考え出された。1964年のIBMシステム/360の発表に始まる第三世代においては、主要な素子のIC化が進み、大規模で複雑な電子回路が飛躍的に小形化し経済的になると共に、仮想記憶方式や新しいプログラム言語の開発によって計算機利用者の負担が大幅に軽減された。そして、計算機に関する個々の概念がまとめられ、コンピュータ・アーキテクチャの成熟期を迎える。

ハードウェア、ソフトウェア両分野における絶ゆまぬ開発により、素子の発展はICからLSIへと進み、今日では超LSI技術の実用化段階に入っている。こうした技術の進歩に伴い、電子計算機への期待も大きくなってきている。計算機を利用する分野も、単に科学技術計算用としてばかりでなく、銀行業務のオンライン化、医療システムへの応用、工場、ビルの管理、通信システム、さらに、パターン識別による製品の選別、品質評価、あるいは、自然言語情報処理等、多くの分野に渡っている。

ところで、現在の電子計算機の入力装置としては、紙テープ、カード、磁気テープ等、種々のものがあるが、その元のデータの始どはキーパンチャーと呼ばれる人々によって作成されたもので、一人が一日に生産できるデータの量は

計算機の演算速度と比較した場合、極めて僅かな量である。現代社会における各種の情報の発生、伝達量は年毎に増え続け、これらをすべて人手で処理することは不可能になっていくであろう。情報の大半が数字や文字、図形であり、この種の形態の情報を機械的に、しかも高速かつ正確に処理する方法や伝達する方法を確立することが望まれている。音声や文字のパターン認識技術は、これらの要望に答える有力な方法と考えられ、その実現に向けて研究が進められている。

一方、我々が日常使用している日本語を計算機で処理しようとする日本語情報処理システムの研究が進み、伝票や帳簿の作成、人名、地名等の印字等が可能なシステムが比較的廉価な費用で利用できるようになってきた。このようなシステムの利用によって、一層、事務の機械化、効率化が計られるであろう。しかしながら、その処理内容は初歩的な作業に限られており、これからの研究、開発次第で、広く普及するか否かが決定されるであろう。例えば、日本語で使う文字種の大部分を占める漢字の入出力方法、記憶方式、符号化法等によってそのシステムの良否が決定される。従って、日本語情報処理システムが広く一般に普及するためには、漢字の扱い方ができるだけ容易で、なおかつ、様々な要望に答えられるものでなければならない。

本論文は、このような点を踏まえ、漢字パターンの構造的な特徴に注目した漢字パターンの表現法について述べたものである。また、その表現法の有効性を調べることを目的に、手書き漢字パターンの表現と漢字パターンの生成モデルに応用した場合の結果についても述べている。本論文は8つの章から構成されており、以下に、各章の内容を概略的に述べる。

第二章では、計算機による言語処理について簡単に触れる。特に、最近盛んに研究が行われている形式言語理論の中の生成文法について述べる。また、日本語情報処理における入出力方法や、それに付随する各種の問題点を述べると共に、入力方法の一つとして期待される文字のパターン認識技術に触れ、これまでに報告されている識別方法や認識理論を方法論的に分類する。

第三章では、漢字の持つ構造的な特徴について述べる。また、その特徴を有効的に利用するためには、どのような処理を行えばよいかを考え、工学的に漢字の構造を定義する。

第四章において、第三章で述べる構造的定義をもとに、漢字パターンを効率良く表現するために行うべき処理について考え、また、その処理過程で得られる幾つかの性質について論じる。

第五章では、第四章で述べる表現法によって得られる漢字パターンの表現式を符号化するために、ある生成文法を定義する。そして、その生成文法によって生成される文と、第四章で述べる漢字パターンの表現式との関係についても述べる。

第六章では、第五章までに述べる漢字パターンの表現方法と符号化法を、手書き漢字パターンに応用し、その符号化法を利用した手書き漢字パターンの分類、識別の可能性について検討する。

第七章では、本論文で述べる漢字パターンの表現法のもう一つの応用である漢字パターンの生成モデルについて述べる。また、その生成モデルと漢字パターンの表現式とを組み合せた場合の可能性を、漢字情報処理システムと関連づけて簡単に論じる。

最後の第八章において、本論文で述べる漢字パターンの表現法について要約すると共に、今後の検討課題等について触れる。

第二章 計算機による言語処理

2-1 自然言語と形式言語

2-1-1 自然言語処理について

19世紀のヨーロッパで発達した言語学は、言語の系統や言語の歴史的変化等についての研究が主なものであった。

20世紀になると、言語の構造を記述する構造言語学が発達し、続いて20世紀後半に生成言語理論が発表されると、これを機に、さまざまな理論や批判が出され今日の言語学へと進展してきた。(3)

一方、近年になって、情報検索、機械翻訳、人工知能等の諸分野において高度な自然言語処理が必要になってきたことなどから、言語学に限らず、学際的な諸科学からのアプローチや研究が盛んに行われるようになった。とりわけ、人工知能の分野では、自然言語の処理が最も重要な研究テーマの一つとなり、数多くの研究成果が報告されている。

自然言語を扱う場合に問題になることとして、自然言語の理解をどう捉えるかということがある。(4)我々、人間が言葉を理解するということは、脳の働きと密接な関係にあり、今だ、確かなことがわからないのが現状である。しかしながら、ある質問に対してどのような応答を示すかという機能的な面から、理解するという行動を捉えることができる。人工知能などで使われる自然言語理解のモデルも、殆どが質問応答システムとして考えられ、これらのシステムの評価は、多数の入力文に対する各応答の適切さと柔軟性によって行われる。そして、その質問応答システムにおいては、入力文の解析、内容の記憶、内部情報との関係による演繹、推論、連想といったことが、入力文に対する応答の適切性、柔軟性に大きく係わってくる。

計算機による言語処理研究には種々のアプローチがあるが、その大きな分野の一つが、自然言語の構造的な側面を計算機によって処理しようとする研究である。また、自然言語文を単なるデータとみて統計的処理を行う研究も、一つの大きな分野となっている。人工知能で扱う自然言語理解のモデルの研究は、主として前者の分野の研究であり、ある印刷物における使用文字の頻度調査等は後者の研究に属する。以下、簡単にそれぞれの分野について述べる。

まず、自然言語文を構造的な面から解析しようとする場合、[5],[6]

（1） 構文解析
（2） 意味解析
（3） 文脈解析

の三段階に分類することができる。構文解析は、文の各要素の文法的な役割を明らかにするもので、この構文解析には、近年発展してきた生成文法理論が大きく影響を及ぼした。意味解析は、構文的に可能な構造であっても意味の上で成立しないものを除いて文法適用の場合の数を減らすためのもので、種々の同義語、類似語等による分類表現などを利用して行うことができる。また、文脈解析は、文と文との関係、一般社会知識などの関係から文を解析することである。その場合、ある文の記述の結果から、次に何が生じるかを推論する知識を導入することが必要で、その知識も、一般的に成立する知識と、その場の文脈上から得られる情報とを区別することが必要である。

文章解析の方法を、上述のように三段階に分類したが、これらはそれぞれ独立に可能なものではなく、文法、意味、文脈、さらには社会的な知識などが混然一体となった形で文章解析が成り立っている。

ところで、自然言語文の文章解析を行うためには大量の情報が必要であるが、これらの大量の情報を効率的に利用するためには、各情報の具体的な内容、計算機内でのデータ構造、その活用方法などが大きな問題であり、文章解析技術と同時に、情報検索技術の発展や組織的な情報作成作業等が行われる必要がある。[7]

一方、自然言語の統計的処理は、計算機の発達する以前から行われていたが、[8] 厖大な量のデータを扱うために、その作業には多くの労力と時間を必要とした。

日本において本格的に日本語の統計的な性質に関する調査が行われたのは、戦後のことで、昭和21年に毎日新聞社が、昭和24年に朝日新聞社がそれぞれ日刊新聞を対象として調査を実施した。

2616種の活字29,507,387個について使用頻度をとった朝日新聞社の調査結果に依れば、使用頻度の最も多い活字は、"の"で2.88128%、続いて、"に"が1.83332%となっている。表2.1.1は、その調査結果の一部を示したものである。また、この二社の調査結果を総合すると、図2.1.1のようになることが報告されている。[9] その結果に依れば、百位までの文字で全体の52%強、一千位で90%、二千位までで98.5%をカバーする。しかしながら、その収束性は次第に緩やかになり、一万字種を用意しても100%にはならない。また、このような統計的性質は、対象とする分野によって文字順位は大幅に変化する。即ち、科学技術論文と文学作品とでは、使用される文字の頻度順位が大きく異なることは容易に考えられる。このような分野別の文字使用度数分布や、雑誌における文字使用度数分布などが、国立国語研究所によって調査、報告されている。[10]

表 2。1。1　文字使用度数の順序の一例

順　位	文　字
1 ～ 24	のにはる　をいた、　とでてし　がなつか　十一れら　二三日も
25 ～ 48	へりうこ　すあまよ　さ五く、　会わけき　四大本そ　どめ中田
49 ～ 72	んだ出国　同長上え　事人ン六　おせ行八　名員部　0円生ち
73 ～ 96	合業者時　月地ろ方　村は下山　東町みス　七内新議　年市トｌ

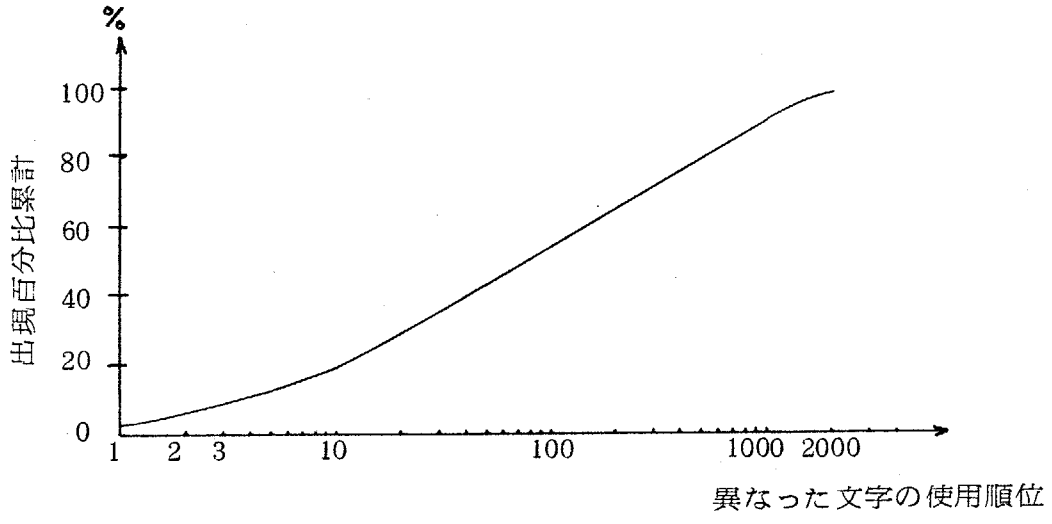


図 2・1・1　日刊新聞記事における異なる文字の分布を示すグラフ

2-1-2　形式言語理論(11)

形式言語の研究は、1956年頃、ノーム・チョムスキー（Noam.Chomsky）が自然言語の研究に関連して、英文法に対する一つの数学的モデルを与えたときを以って始まった。そして、今や言語理論およびオートマトン理論の技巧と、その結果の知識なしには計算機科学の本格的な研究を完全に遂行することは不可能であると言える。

ところで、言語を表現する場合、その方法としては、ある与えられた文がその言語に属するかどうかを判定するアルゴリズムを示すことが考えられる。非常に一般的な方法として、ある手続きによってその言語に属する文に対して'yes'を出し、そうでないものには'no'を出すことである。このような手続き、またはアルゴリズムは、その言語を認識（recognize）すると言われる。一方、言語を認識の面から捉えるのとは別に、生成の面から表現することも可能である。即ち、或る言語の文を、或る順序で次々と組織的に生成する手続きを考えるのである。前者がオートマトン理論的立場であり、後者が生成文法、即ち形式言語理論的立場である。ここでは、後者の生成文法理論について、そのあらましと今後の本論文の展開に必要な部分を簡単に説明する。

文法の概念は、本来言語学者によって自然言語の研究の過程において定式化された。文とは何かを正確に定義するだけでなく、文の構造を記述する為の仕方を与え、その形式的文法理論によって自然言語を記述することを目的としたものであった。しかしながら、そのような目的は現在も達成されてはいない。その事の大きな理由として、何が文を構成するかを正確に決定する明瞭な規則が無いためである。

さて、次に示すような英文があるとする。

"The little boy ran quickly."

この英文を、我々が知っている文法の知識によって'解剖'すれば、まず、こ

の文は、名詞句"The little boy"と、それに続く動詞句"ran quickly"とから成り立っている。さらに名詞句は単数名詞"boy"と、それに懸かる2つの形容詞(冠詞)"The"と"little"とに分解できる。同様に動詞句は、単数動詞"ran"とそれに懸かる副詞"quickly"とに分解できる。この解剖を図示すると、図2.1.2のように表わされる。

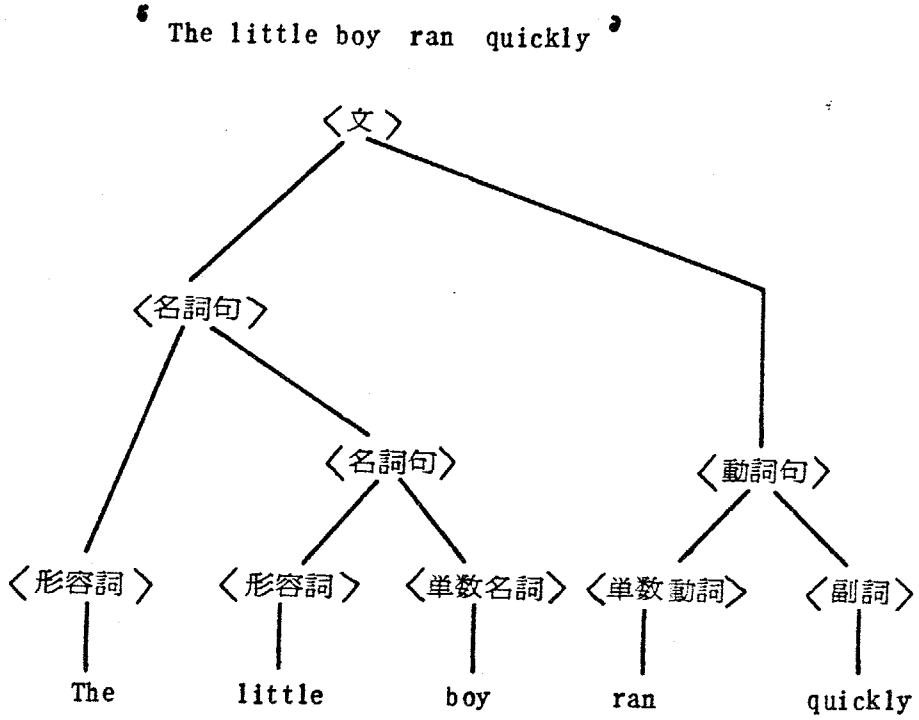


図 2．1．2 文"The little boy ran quickly"の構文木

上記の文を解剖するために用いた規則は次のように書くことができる。

〈文〉→〈名詞句〉〈動詞句〉，〈名詞句〉→〈形容詞〉〈名詞句〉

〈名詞句〉→〈形容詞〉〈単数名詞〉，〈動詞句〉→〈単数動詞〉〈副詞〉

〈形容詞〉→The，〈形容詞〉→little，〈単数名詞〉→boy

〈単数動詞〉→ran，〈副詞〉→quickly

これらの規則における矢印は、その左側にあるものが右側にあるもので生成できることを表わす。この図2.1.2の例に示した英文の構文において4つの概念が現われている。まず、或る文法的な類―〈単数名詞〉,〈動詞〉,〈文〉

等があり、そこから単語の列が導き出された。これらの文法的類に対応するものを、非終端記号(nonterminal symbol)、または変数(variable)と呼ぶ。次に単語それ自身があった。この単語の役割を果たすものを、終端記号(terminal symbol)と呼ぶ。第三の概念は、いろいろな変数や記号の列の間の関係である。そのような関係を生成規則(production rule)と呼ぶ。最後に、或る特別な非終端記号があって、その言語に属すると考えられるすべての終端記号列は、ちょうどその非終端記号から生成される。このような特別の非終端記号を開始記号と呼ぶ。

以上に述べた文法Gを形式的に、(V_N, V_T, P, S) で表わす。記号 V_N, V_T, P, S は、それぞれ非終端記号の集合、終端記号の集合、生成規則の集合および開始記号である。但し、$V_N \cap V_T = \phi$ で、$V = V_N \cup V_T$ とするとき、V上のすべての文の集合を V^*, V^* から空文{ε}を除いた集合を V^+ とすれば、生成規則の集合Pは、$\alpha \to \beta$ という形式の集まりで、$\alpha \in V^+$, $\beta \in V^*$ である。特別の指定のない限り、非終端記号としてローマ字の大文字を、終端記号としてローマ字の前の方の小文字を使う。また、終端記号の列はローマ字の後の方の小文字を、非終端記号と終端記号の混った列にはギリシャ文字の小文字を使う。

次に、文法Gの生成する言語について簡単に述べる。集合 V^* に属する列の間の関係 $\underset{G}{\Longrightarrow}$ は、もし、

$\alpha \to \beta \in P$ で、$\gamma, \delta \in V^*$ であるとき、

$\gamma\alpha\delta \underset{G}{\Longrightarrow} \gamma\beta\delta$

と記し、生成規則 $\alpha \to \beta$ が、列 $\gamma\alpha\delta$ に適用され、列 $\alpha\beta\delta$ が得られたことを意味する。また、$\alpha_1, \alpha_2, \cdots, \alpha_m \in V^*$ で、$\alpha_1 \underset{G}{\Longrightarrow} \alpha_2$, $\alpha_2 \underset{G}{\Longrightarrow} \alpha_3$, $\cdots$ $\alpha_{m-1} \underset{G}{\Longrightarrow} \alpha_m$ であるとき、

$\alpha_1 \overset{*}{\underset{G}{\Longrightarrow}} \alpha_m$ (あるいは単に $\alpha_1 \overset{*}{\Longrightarrow} \alpha_m$)

と記す。このとき、文法Gによって生成される言語を $L(G)$ とすると、

$L(G) = \{ w \mid w \in V_T^*, \text{かつ } S \overset{*}{\Longrightarrow} w \}$

と定義する。そして、文法G_1、G_2は$L(G_1)=L(G_2)$のとき、等価、または同値であるという。

一般に、$S \overset{*}{\Rightarrow} \alpha$, $\alpha \in V^*$である場合、αは文形式(sentential form)と呼ばれる。

さて、以上に説明した文法Gにおいて、その生成規則に制限を加えることによって種々の文法が得られる。

$G=(V_N, V_T, P, S)$をある文法とする。

(1) Pのすべての生成規則$\alpha \to \beta$が、

$$|\beta| \geqq |\alpha| \quad \text{但し}, |\cdot| \text{は記号の数を表わす。}$$

を満たすとき、Gを1型文法と言う。この1型文法においては、

$A \in V_N$, $\alpha_1, \alpha_2, \beta \in V^*$, $\beta \neq \varepsilon$に対して

$$\alpha_1 A \alpha_2 \to \alpha_1 \beta \alpha_2$$

の制限を加えても言語の範囲は変化しない。この生成規則は、Aが文脈$\alpha_1 - \alpha_2$の中に現われたときだけAをβに置き換えることが許される。このことから、1型文法は文脈依存文法とも呼ばれる。

(2) Pの生成規則 $\alpha \to \beta$ において、

(イ) αは一個の変数、(ロ) βはε以外の任意の列

であるとき、この文法を2型文法と呼ぶ。この文法においては、

$$A \to \beta$$

という形の生成規則は、Aがどのような文脈に現われるかに無関係にAをβで置き換えることを許すものである。このことから、2型文法は、文脈自由文法とも呼ばれる。

(3) Pの任意の生成規則が、

$$A \to aB, \text{ または } A \to a \quad , A, B \in V_N, a \in V_T$$

という形であるとする。このとき、Gを3型文法、または正規文法と呼ぶ。

一方、(1), (2), (3)で述べたような制限を加えない文法を0型文法という。

本論文において利用する文法は文脈自由文法であるが、その文法に関連する定理を述べる前に、最左導出(leftmost derivation)の定義について述べる。

導出が最左であるとは、各導出のステップにおいて、置き換えが一番左側の変数に関してなされていくことを言う。即ち、もし、

$$S \Rightarrow \alpha_1 \Rightarrow \alpha_2 \Rightarrow \cdots \Rightarrow \alpha_n$$

が文法 $G = (V_N, V_T, P, S)$ の最左導出であれば、$1 \leq i \leq n$ に対して、

$$\alpha_i = x_i A_i \beta_i \qquad x_i \in V_T^*,\ \beta_i \in V^*,\ A_i \in V_N$$

と書ける。$A_i \longrightarrow \gamma_i \in P$ であり、$\alpha_{i+1} = x_i \gamma_i \beta_i$ で、α_{i+1} は α_i から A_i を γ_i で置き換えることによって導出されている。

文脈自由文法と最左導出性について次の定理が成立する。

〔定理2-1〕

与えられた文脈自由文法 $G = (V_N, V_T, P, S)$ に対して、もし $S \overset{*}{\underset{G}{\Longrightarrow}} w$ ならば、G による w の最左導出が存在する。〔証明 略〕

以上、簡単に形式言語に関して必要な概念を述べた。

2-2 日本語情報処理

2-2-1 日本語情報処理の目的と意義[12],[13]

欧米諸国は長年に渡って事務の機械化を促進してきたが、その基盤は自国の文化(言語)にあった。そして、それはタイプライタという非常に安価で操作性の良い入出力装置の存在によってさらに促進された。一方、日本に於いては、殆どの情報処理が英数字で行われており、僅かにカタカナが使われているに過ぎなかった。

我々日本人は今後も漢字仮名混り文を使用していくであろう。従って、事務の機械化、情報量の増加に対応していくためには、電子計算機による日本語情報処理システムは不可欠なものになると考えられる。この日本語情報処理システムを高度、かつ有効なものに発展させていくためには、それに伴うハードウェア技術とソフトウェア技術の発展が必要である。ハードウェア技術に於いては、超LSI技術の発展や印字方式の技術革新によって、漢字フォントのLSI・ROM化、実用的なワードプロセッサの開発、漢字ディスプレイの低価格化等が可能となった。しかしながら、日本語文の入力方法や日本語文の解析、理解等のソフトウェア技術においては、まだ多くの解決すべき問題があり、今後の研究成果に負う所も多い。

この節では、現在までに実用化されている入出力方法について、それぞれの方式と特徴、あるいは問題点等を簡単に述べる。

2-2-2 日本語文の入力方法(13)

日本語情報処理においては、日本語文の解析の困難さの他に、日本語の持つ特殊性、とりわけ異なる字種が極めて多数であるという理由から、文字の符号化法や入出力の方法などが問題となる。符号化に関しては、漢字のＪＩＳコードも決定され、これまでまちまちであった漢字の符号が統一された。しかしながら、入力方法に関しては、決定的な方法は現在も見あたらず、種々の方法が考えられている。日本語の情報処理システムとしての健全な発展のためには、漢字を主体とした日本語の入出力問題は絶対に避けて通ることのできない問題である。そこで、これまでに使用されてきた各種の入力方式について簡単に説明する。

〔１〕 フルキー方式キーボード

これは、１個のキートップ面に２～１５種の文字が刻印されており、必要な文字を打鍵するには、まず、その文字が刻印されているキートップを押し下げる。続いて、そのキートップ上の刻印文字配列に従ってシフトキーを選択操作するもので、入力文字数は２４００字程度のものから、３,０００字、多いものになると一万字打鍵可能を目指したものもある。

入力可能な文字数は、キーの数と、各キーに刻印する文字の数によって決定される。その場合、キーの数と刻印文字数との関係は、打鍵能率に大きく影響するが、この関係は各メーカーによって様々であり、英文タイプライタの様に、どのタイプライタでもほぼ同等の作業ができるという訳にはいかず、各入力装置別に訓練を要する。

〔２〕 テーブル・ルックアップ方式

この方式は、フルキー方式をより簡略化し、ある程度の小形化を計ると共に、打鍵操作の簡便性向上を目指したものである。

フルキー方式が１個のキートップ面に複数個の文字を刻印する場合が多いの

に対し、テーブル・ルックアップ方式は、漢字表を参照しながら一文字一文字入力していくものである。この方式の主流となるものは、図形入力装置などにしばしば用いられているタブレットを利用するもので、タブレット上に配置された漢字表において入力したい文字を検出ペンで指示すると、その指示した点のX座標、Y座標の値をもとに入力文字の符号を検出するという原理である。

〔3〕 読みかえ入力方式

日本語は、音韻的には56音で表現可能である。そこで、漢字を56音のかなに読み換えて、かなタイプで日本語表音記号情報として入力し、入力後に漢字かな混り日本語文に変換するかなタイプ入力漢字変換方式がある。(14),(15) また、英数字キーボードを利用して、ローマ字表記で入力を行うローマ字入力漢字変換方式などがある。

かなタイプ入力漢字変換方式は、フルキー方式やテーブル・ルックアップ方式などのような漢字鍵盤装置を必要とせず、廉価で、しかも一般性のある仮名鍵盤装置を用いて、鍵盤を注視することもなく打鍵入力が可能なため、入力装置の経費の安さや操作員の習熟の容易さ、打鍵入力の高速化等の利点から、日本語文の入力方式として期待されている。しかしながら、かなタイプ入力漢字変換方式、ローマ字入力漢字変換方式のいずれの方法においても、表音記号による入力データ列を漢字変換するという処理が必要であり、その変換の際に利用する、「漢字語、熟語辞書」の良否如何によって、入力表音記号を誤りなく漢字に変換する率、いわゆる適合率が決定的に左右されることになる。さらに、入力時の文法的制限や条件もあるため、フルキー方式やテーブル・ルックアップ方式と比較すると、その普及度は低い。

以上に述べた方式の他にも、ライノプットという呼び名で報告されている入力方式や、オンラインの手書き文字認識方式、(16) パターン合成入力方式等が考えられているが、これらの方式によって全ての文字を入力することには無理があるように思われ、次に述べる外字入力方式等に利用される場合が多い。

2-2-3 外字入力方式と入力文字の検定[13]

最近の漢字入力装置においては、一般的に、処理し得る文字数が増大する傾向にある。しかしながら、いかに収容文字数を増したところで、確率的には低いものの、通常の打鍵操作やその他の入力操作で処理しきれない場合が生じる。一般に、入力機器に標準的に用意されている文字種を盤内文字、それ以外の文字種を盤外文字、あるいは単に外字と呼んでいる。この外字に遭遇した場合の処理方法としては、あらかじめ定めてある外字符号を入力する方法や、外字を各部画に分解し、それぞれの盤内文字で代替して入力する方法等がある。

漢字コード表によって外字を検索する場合、音訓順、画数順、部首順等によって検索を行うが、外字については日常における使用頻度が少ないことなどから、オペレータがその音訓を記憶していることを期待できない場合が多く、一般的には部首順によって検索を行うことが多い。しかし、部首順による分類にも多くの問題点があり、外字索引コード表には音訓順、部首順とも重複をいとわずに編集してある場合が多い。

一方、パターン合成入力方式は、合成パターンの型を予め定めておき、そのパターンの型と各部パターンの情報を入力する方法である。例えば、「湘」という文字に対しては、「氵」、「木」、「目」の三つのパターンに分割する。その分割は2分割、或は3分割とし、図2.2.1に示すような8種類のパターンとする。即ち、「湘」を入力する場合、パターンの型情報として7の型を入力し、次に「氵」、「木」、「目」の順で入力を行う。[17]

この方式は、外字符号入力方式と比較すると、盤内文字のみで入力を行うことができるが、確実性という点では外字符号入力方式の方が、より安全である。

ところで、文字の入力段階での誤入力の検査、訂正を行う機構を、一般的にベリファイ機構（verify system）と呼ぶが、一文字打鍵毎にベリファイを行うことは打鍵能率を著しく低下させ、しかも、如何に丹念に行っても誤入力は必

ずと言っていい程存在する。従って、一般的には打鍵毎のベリファイは行わず、一旦、すべてを入力し、その初期入力を校正出力する。その出力情報をもとに初期入力ファイルを訂正する方法がとられる。初期入力で形成された入力ファイルにエラーが発見された場合、それをどのような方法によって訂正、編集を行うかということも、その入力システムの良否に大きく影響する。

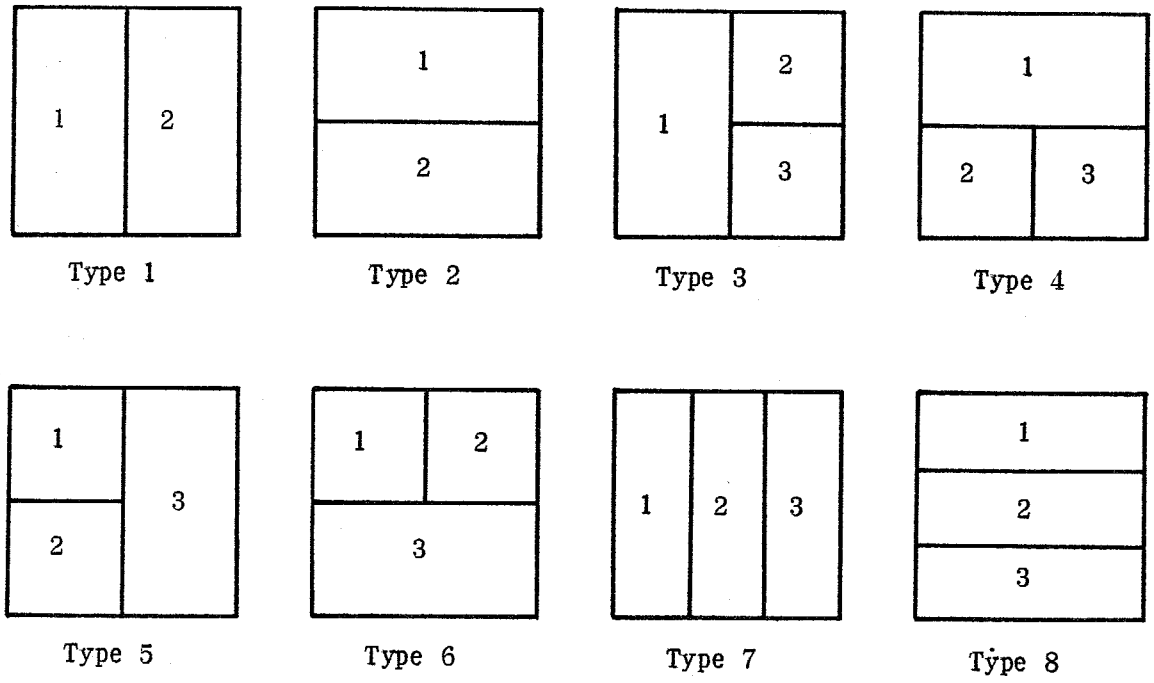


図 2・2・1 合成パターンの型

2-2-4 日本語文の出力方法

電子計算機システムを用いた日本語情報処理への社会的要求と、ハードウェア技術の発達に伴い、各種の方式の出力プリンタや植字機が考案、実用化されてきた。これらのプリンタは、仮名や英数字をも含めて一般的に漢字プリンタと呼ばれている。この漢字プリンタについて以下に簡単に述べる。

漢字プリンタは、その装置で用いられる各種の方式によって分類することができる。

インパクト方式は、記録紙に直接的な衝撃を与えて記録を行うもので、活字を用いたものはその代表的な例である。また、ドットパターンに分解記憶した文字を印字する場合に、ワイヤピンをソレノイドで駆動して点印字を行うような方式もインパクト方式である。

一方、直接的な衝撃を与えないで印字を行う方式を一般にノンインパクト方式と呼ぶが、このノンインパクト方式においても、各種の方式が存在する。文字書体の発生方式に注目して分類を行えば、各文字をドット構成で記憶しておく方式や、写真フィルム上に文字パターンを記録しておき、必要に応じてそのパターンをビジコンやフライングスポット走査管で撮像する方式、あるいは、漢字パターンをホログラムに記憶させておく方式などに分類することができる。

また、印字方式に注目して分類を行った場合は、多針電極による静電および放電破壊記録、感熱素子を用いた熱記録、あるいは、インクジェット記録、オプティカル・ファイバ管による印画紙への記録等がある。これまでに開発された各種の漢字プリンタを方式別に分類したものが表2.2.1である。(13)

一方、これまで述べたものがハードコピーを対象とした出力装置であるのに対し、ソフトコピーを対象とした装置として、漢字のディスプレイ装置がある。情報検索や、既に蓄積されている文章メッセイジ等の、単に閲読するだけでこと足りるような業務が対象の場合、省資源の点で非常に期待される出力装置で

表 2。2。1　漢字プリンタの印字方式の分類

	方式名	印字方式	記録方式
インパクト方式	活字印字式	活字	インクリボン
	機械式ドット文字	ワイヤピン	インクリボン
ノンインパクト方式	電子式ドット文字	多針電極	静電記録
		オプティカルファイバ管	湿式電子写真
		イオン流電界制御	インクジェット
		半導体発熱マトリックス	感熱記録紙
	字母ビディコン式	オプティカルファイバ管	銀塩安定化紙
	フライングスポット式	オプティカルファイバ管	銀塩安定化紙
		CRT	乾式電子写真

ある。

漢字プリンタの場合と同じく、文字パターンの発生方式には、ドット文字記憶方式と字母走査方式とがある。通常、漢字ディスプレイ装置においても毎秒30回程度のリフレッシュを行うが、その際、リフレッシュメモリでの記憶形式を文字符号形式とするか画像形式とするかによってメモリ容量が大幅に異なる。一般に、ドット文字表示形式では、文字発生速度が画像走査速度に追従しないために、画像パターンリフレッシュ方式である。従って、24×24ドットの文字パターンの場合、表示一文字につき576ビットのメモリ容量が必要となる。一方、字母走査方式では、文字符号リフレッシュ方式でも十分追従できる可能性があり、その場合、1文字につき12～16ビットのメモリ容量ですむことになる。

ところで、ドット方式は字母方式などのアナログ式に比べ、電子計算機の内部処理と同じようにディジタル処理を行うことができるため、出力文字品質の安定性に優れており、さらに記憶素子も電子計算機と同じものが使用できることなどから、今後も文字パターン発生方式としては最も有利な方法の一つであると考えられる。しかしながら、文字品質の向上を計ったり、文字数を増やそうとすると、それだけ記憶容量が増大し、システムが高価になると共に高速アクセスメモリが使えなくなり、出力速度が低下してしまう。このような点から、使用頻度別に文字を分類し、高速アクセス可能なメモリに使用頻度の高い文字を記憶させてシステム全体としての高速化を計ったり、あるいはドットパターンの圧縮記憶等が行われる。

漢字パターンの圧縮法の一つに、漢字パターンの構造的な特徴を利用して圧縮を行う方法がある。その方法は、漢字パターンがいくつかの基本的なパターンから合成されているという特性と、漢字パターンをストローク表現で記憶し、出力する場合にストロークデータをドットパターンに変換する方式とを組み合せたものである。図2.2.2に示したような基本パターンに対し、各ストロークの始点と終点の位置をストローク情報として記憶しておき、必要なときにこ

のストローク情報から図2.2.2のような32×32のドットパターンに変換するのがストローク方式である。図2.2.3は、32×32のドットパターンに変換されたそれぞれの基本パターンの合成によって別の文字の作字を行う過程を示したものである。(17),(21)

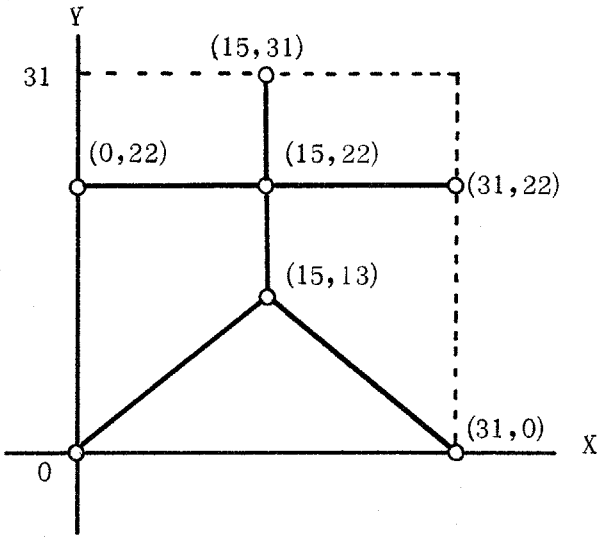


図 2・2・2 基本パターンのストロークの始点、終点の位置

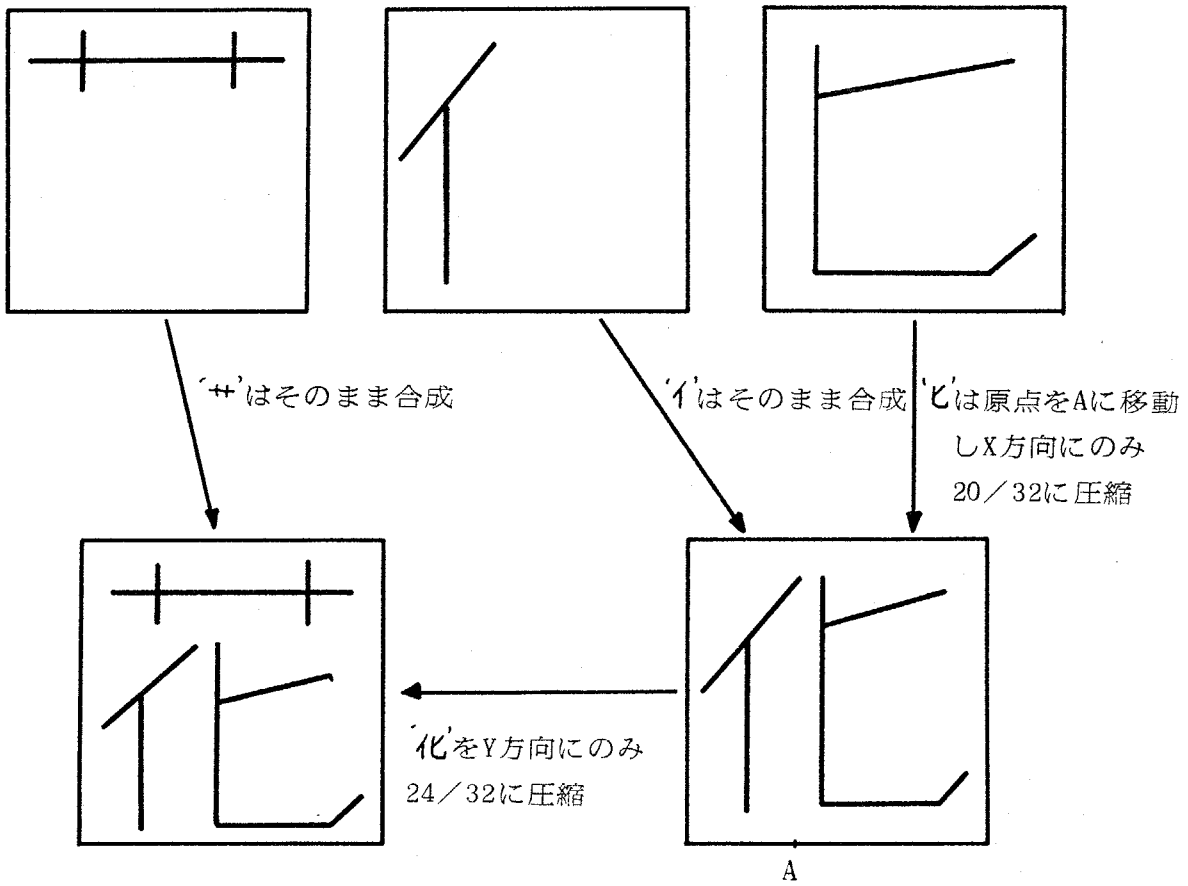


図 2・2・3 基本パターンによる合成作字

2-3 文字図形のパターン認識

2-3-1 文字認識技術と認識理論[2),(22)~(24)]

文字図形を機械的に読み取るという試みは、今世紀初頭から幾つか行われた。しかし、文字や図形の認識装置が今日のように各方面でその必要性が叫ばれ、また熱心に研究されるようになったのは、電子計算機の発達と深く係わりあっている。文字読取技術が注目され始めた1955年頃は、電子計算機の年代で言えば第二世代が始まろうとしていた時期に相当しており、電算機システムにおける有力な入力装置としての期待があった為と思われる。文字読取による入力方法は、前節で述べた種々の入力方法と異なり、我々が書く文字をそのまま入力できる点でより自然な形での入力方法であるため、現在までに数多くの読取方法が考え出されている。しかしながら、現在実用化されている装置は、限られた文字種で、しかも非常に丁ねいに書かれた文字や印刷文字等に限られており、一般的に普及するためには一層の研究開発が必要である。その文字認識研究の経過について以下に簡単に述べる。

初期の認識技術では計算機の演算速度、記憶容量の制限等から、今日ほど高度な処理を行うことは不可能であった為、なるべく簡便な論理回路で入力図形から直ちに適切かつ有効な特徴量を得ようとするものであった。即ち、文字の位置を固定して文字の存在する平面内のいくつかの観測点に文字の部分が存在するか否かを検出し、その検出結果の組み合せによって識別を行おうとする定点サンプリング法や、平面上に適当に配置された数本のゾンデとなる線の上に文字の部分が存在するかどうかの組み合せにより判定を行うゾンデ法、あるいはまた、適当なスリットを設け、そのスリットのそれぞれの出力波形の組み合せによって文字の判別を行うスリット法などが代表的なものであった。しかも、認識の対象となった文字は、読取を容易にするために特別に設計されたものが

多く、印字の規格も厳しいものであった。そして、変換の容易さから磁気インクが使用される場合が多かった。一方、認識論理はトランジスタによるフリップフロップとゲート回路から構成されており、認識処理の内容は単純なものであった。

その後、郵便の区分を目的とした手書き文字認識装置が実用化された。IBM 1827は1967年に、実用的な手書き数字認識装置としては初めて実用化された装置で、ブラウン管のビーム走査をサークルスキャンと呼ぶ特殊な方式で制御し、入力文字の輪郭と傾きを高速に検出して、これらの情報から認識を行うものであった。IBM 1287とほぼ同時に、我が国で開発された郵便区分機は、郵便番号を読み取って行先別に区分する装置である。この装置の手書き数字読み取り処理は、大きさの正規化、線素方向抽出、水平特徴抽出および判別の四段階で行われ、特徴抽出後の未知パターンの照合にはマイクロプログラム方式を採用することによって、従来のハードウェアの結線では不可能だった多彩な手書き数字の読み取りが可能になった。

現在では、手書き英数字、手書き片仮名文字にまで読み取り技術が進歩し、金融機関の会計機等の印字データの処理、電気、ガス等の公共料金徴収業務データの処理、あるいは伝票、帳票類のデータ処理に利用されている。さらに、これまで実現が困難であると考えられていた印刷漢字についても、2,000字以上もの漢字を読み取ることの可能な装置が実用化されるようになってきており、文字読取方式が日本語情報処理システムの入力方法の一環を担う時期が遠からず来ると考えられる。

ところで、このような文字読み取り技術の発展は、単に読み取り方式の進歩のみに依るのではなく、最近の集積回路技術の急速な発達によって相当高度な読み取り方式を安価な装置として構成することが可能になったことや、光電変換装置として従来使われてきたビジコンやイメージ・ダイセクタ管などに代って、ホトダイオードやC.C.D（charge coupled device）等の半導体素子センサが用いられるようになったことによって装置の小形、安定化が計られたとい

うことが大きく貢献しているね。

以上、文字読取技術の歴史的経過と利用状況について簡単に述べた。現在、装置として実用化されている文字読取方式は、その装置の目的である高速かつ正確な読み取りということを前提に、限られた字種のみを対象とした読取方式であり、一般的な文字認識法として良い方法であるとは必ずしも言えない。読取装置的制約にあまり束縛されない、広い意味での文字図形の認識法の研究は、今後も続けていかなければならない。

文字読取装置の開発と平行して、人間を含めた生物系で日常的に行われているパターン認識の機構について本質的に解明しようとする認識理論の研究も盛んに行われた。これまで行われてきた認識理論の研究を方法論的に眺めると、与えられたパターン集合自体の表層的な情報を用いて認識系を構成する決定論的認識論と、パターンが持つ内部的な情報構造を捉え、これを組織化して認識系を構成する図形言語学的認識論の2つに大別することができね。前者は、数理統計学の分野で始まった決定関数の理論を認識論に応用したものであり、後者は形式言語理論の方法をパターン認識論に応用したものであね。この図形言語学的方法の研究は近年盛んに行われ、重要な研究分野の一つになってきているね。

我々人間が、文字や図形を認識する機構は、脳の生理学的な働きと深い関係にあり、依然として明らかにはなっていないが、上述したようなそれぞれの認識論によって全て解決するような単純なものではないということは容易に理解できね。むしろ、それぞれの認識論の持つ長所を合せ持った、より総合的なものであると考えられね。

2-3-2 文字認識法について[23]

これまで多くの文字読み取り方式が研究され、報告されてきたが、以下にそれぞれの方法を概略的に述べる。

文字認識の研究における認識のための処理行程には、必ずしも明確な区別は存在しないが、一般的に分けるとすれば、未知文字入力（光電変換・一文字切り出し）、前処理、特徴抽出、分類及び判定、等になる。

文字入力は、実際の文字読取装置では読み取りの成否に大きく影響するため重要な部分であるが、ハードウェア的要素が強いため、本論文ではサンプリング後の量子化された文字パターンをG(i,j)として扱う。ただし、i,jはそれぞれの座標を示す正整数で、$1 \leq i \leq M$, $1 \leq j \leq N$とする。また、$0 \leq G(i,j) \leq 1$と仮定し、0ならば白、1ならば黒、0と1の間の値であれば灰色とする。

前処理としては、手書き文字を扱う場合、構造分析法が多く用いられる関係上、大きさ、位置の正規化、整形、細線化、輪郭抽出等が行われることが多い。一方、印刷文字の場合は主としてパターン整合法が多く用いられるため、文字線幅の正規化、位置の正規化、あるいはボケ変換等が行われる。

特徴抽出は、分類手法と密接な関係にあり、分類手法抜きに特徴抽出のみを論じても有効な議論にはならないが、それぞれの認識法において利用されている特徴を挙げると、

（a） 端点、分岐点、交差点等の点情報やループ数、連結領域数、ストロークの長さ、方向、連結関係、曲線性、等の情報、あるいは、文字の輪郭線や凹凸構造、文字をある軸方向へ射影した時の周辺分布情報など。

（b） フーリエ変換、ウォルシュ・アダマール変換、あるいは、カルマン・リーベ変換等の線形変換によって文字パターン空間から別の空間に写像し、写像後の空間を特徴空間とするもの。

(a)は位相幾何学的な特徴で、殆どのものは構造解析的な手法に多く用いられる。一方、(b)は直交変換によって周波数領域で扱うもので、文字の微細な構造表現よりも二次元空間的な広がりを重要視する場合に多く用いられる特徴である。

分類及び判定は、未知文字から抽出された特徴と、あらかじめ用意された標準文字の特徴(以後、標準パターンと呼ぶ。)との一致の度合いを調べることによって行う。この一致度の評価には、相関値、類似度、距離、論理関数の真偽、状態遷移図等が用いられる。類似度は相関値の特殊な場合と見なすことができるが、相関値を用いる方法と類似度を用いる方法について簡単に説明する。

重み関数 $w_{i,j}^{(\ell)}$ (i,jは座標を、ℓは文字を表わす添字とする。) の形で用意した標準パターンと、未知文字 $G_{i,j}$ との相関値 $S_\ell(G)$ は次式によって計算される。

$$S_\ell(G)=\sum_{i=1}^{M}\sum_{j=1}^{N} w_{i,j}^{(\ell)} \times G_{i,j}$$

判定は、$S_\ell(G)$ の値が最大となる標準パターンが代表するカテゴリーを未知入力文字のカテゴリーとする。重み関数 $w_{i,j}^{(\ell)}$ にも種々のものが考えられる。

類似度法では、標準パターン $Q_{i,j}^{(\ell)}$ に各カテゴリーの理想図形(例えば文字の規格図形や各カテゴリー毎のサンプル文字の平均図形など。)を用い、次式によって類似度を計算する。

$$S_\ell(G)=\frac{\sum_{i=1}^{M}\sum_{j=1}^{N} Q_{i,j}^{(\ell)} \times G_{i,j}}{\|Q^\ell\| \cdot \|G\|}$$

但し、$\|Q^\ell\|$, $\|G\|$ は、$Q_{i,j}^{(\ell)}$, $G_{i,j}$ のノルムで、次式で与えられる。

$$\|Q^\ell\|=\left\{\sum_{i=1}^{M}\sum_{j=1}^{N}\left(Q_{i,j}^{(\ell)}\right)^2\right\}^{1/2}$$

$$\|G\|=\left\{\sum_{i=1}^{M}\sum_{j=1}^{N}\left(G_{i,j}\right)^2\right\}^{1/2}$$

類似度$S_\ell(G)$には、$0 \leq |S_\ell(G)| \leq 1$の性質があり、未知パターン$G_{ij}$が標準パターン$Q_{ij}^{(\ell)}$に完全に一致した場合に$S_\ell(G)=1$になることから、相関値の場合と同様にして判定を行う。

特徴抽出法と分類判定法の組み合せによって種々の認識法が考えられる。次に、これまでに報告された幾つかの認識法について概略的に説明する。

〔1〕特徴整合法

文字が複数個の部分パターンから構成されていると考え、直感的に多数の特徴点を選定し、これと標準パターンのそれとを重ね合せ法によって比較する。この方法は文字読取装置開発の初期の頃に見られた方法である。

〔2〕ストローク・アナリシス法

ストローク・アナリシス法の認識論理は、特徴整合法の論理と同じく、多数の特徴を抽出するが、特徴整合法のように文字パターンを二次元的に保存しておくのではなく、スリット状の部分的なデータから特徴を抽出するもので、一般に、文字を縦方向に2ないし3分割し、その各々の領域にどのような特徴が現われるかを調べる。このため、文字の大きさをあらかじめ知ることと、これを縦割りするための分割論理を備えている。

〔3〕構造解析法

構造解析法は、ストローク・アナリシス法のもつ可能性を生かし、ハードウェア上の制約を解放したものである。1967年に実用化されたIBM1287や、日本の郵便区分機などはこの構造解析法を利用した文字認識装置である。IBM1287は未知文字の輪郭の傾きと位置を特徴として検出し、その特徴を用いて論理回路によって認識を行う。一方、日本の郵便区分機は、大きさの正規化後、線素方向、水平特徴を抽出し、その系列を状態遷移への入力とする。各標準パターンを示す状態遷移の最終状態を通過するか否かによって認識を行う。

〔4〕線図形化法

線図形化法も構造解析法の一種であるが、上述のIBM1287や郵便区分

機の持つハードウェア的制約、抽出する特徴の制約等を除き、それぞれの長所を生かすために考えられた方法である。この方法は、大きさを正規化した後、細め処理（細図形化）を行う。そして、線図形をチェイン・コード化し、このコード系列を未知文字の特徴とする。認識処理は郵便区分機のように状態遷移図を用いて行うが、全体としてソフトウェア処理を前提としているため、自由度の高い、高度な処理が可能である。しかしながら、この方法は二次元図形を一次元的な状態遷移図で表わす所に限界がある。即ち、二次元的に見れば僅かな変化であっても、文字の位相的な性質を変化させることがあり、線の一部の切れ、ループのつぶれ、不要な線の接触等に対し、1個の状態遷移図で表わすことが難しく、各変形に対応した遷移図を用意しなければならない。

［5］ 場の効果法[25]

文字の変形に対して比較的安定な凹凸構造に着目した方法で、論理マスクを利用する方法や、アナログ的にパターンを扱う方法、2値化して扱う方法等、各種の方法が研究された。その中の2値化法は、文字線上の各点で8方向に触手を出し、その触手の重なり具合から、文字部を除く白地部分の閉じ率を求める方法である。文字の輪郭線に沿って閉じ場を走査し、その閉じ率の大小によってその輪郭部分が凹か凸かを判定する。走査された輪郭を輪郭ストロークとして抽出し、その各々のストロークの凹凸の別、重心、長さ、方向、凹量、角度分布などを調べる。その他にも、各ストローク間の関係、例えば結合量、結合方向、相対位置、面積比、ストローク差などを特徴として求める。また、文字の全体的な特徴として、パターンの大きさ、重心、単連結領域数、ストローク数などを求める。そして、標準パターンとの照合に際しては、未知パターンと標準パターンのストローク間の対応関係を調べ、その特徴量の変動が許容範囲内か否か、またストローク間の関係量が許容範囲内にあるかどうかを調べて識別を行う。

この方法を手書き片仮名文字や平仮名文字に適用して高い認識率を得たという実験結果が報告されている。[26]

[6] AbS法によるストローク抽出[27]

認識系の中に、ストロークの生成モデルを含む一つのフィードバック系を構成し、提示されたパターンとの差が小さくなるようなストロークを生成することによってストロークの抽出を行う方法である。そのストロークの生成モデルは、次のようなバッカス記法で示される。

<stroke> ::= <dot> | <bar> | <complex>
<complex> ::= <bar> <curve> | <curve> <bar> <curve>
<dot> ::= 孤立点
<bar> ::= 直線ストローク要素
<curve> ::= 曲線ストローク要素

ここで、ペンの速度ベクトルをv、ペンの位置を(x, y)とすれば、直線ストロークについては、

$$v(t) = B \cdot \exp(i\theta)$$
$$x(t) = Bt\cos\theta + x_0$$
$$y(t) = Bt\sin\theta + y_0$$

であり、曲線ストローク要素については

$$v(t) = (at + b)\exp[i(\omega t + \theta)]$$
$$x(t) = \int_0^t Re[v(t)]dt$$
$$y(t) = \int_0^t Im[v(t)]dt$$

但し、iは虚数単位、B, θ, x_0, y_0, a, bは定数である。

以上の生成モデルにより、ほとんどすべてのストロークが合成可能である。このストローク抽出法とファジーオートマトンとを用いた認識系、手書き漢字75種(47都道府県名)に適用した実験例が報告されている。[28]

第三章 漢字の構造とその構成要素

3-1 漢字の成り立ちと構造解析の意義

我々が今日使用している漢字の起源を遡ると、紀元前千年以上も前の殷の時代の甲骨文字にまで遡ることができると言われている。そして、紀元前後の漢の時代に於って隷書から楷書へと変化し、現在使用している漢字とほぼ同じ形態の文字が完成され、一連の造字法や造語法も確立された。これらの造字造語法を踏まえながらも、我々は、今日の日本の風土、習慣、思想にあった漢字の使い方を行っている。

これらの漢字を集録した辞典類の索引は、「扁」、「旁」「構」等の部首によって分類整理がなされ、その筆順は「上から下」「左から右」の大法則が確立されている。そして、これらの漢字の学習過程においては、個々の文字の扁旁構等の情報からその文字のある程度の意味内容を理解することができ、文字習得の大きな手懸りとなっている。

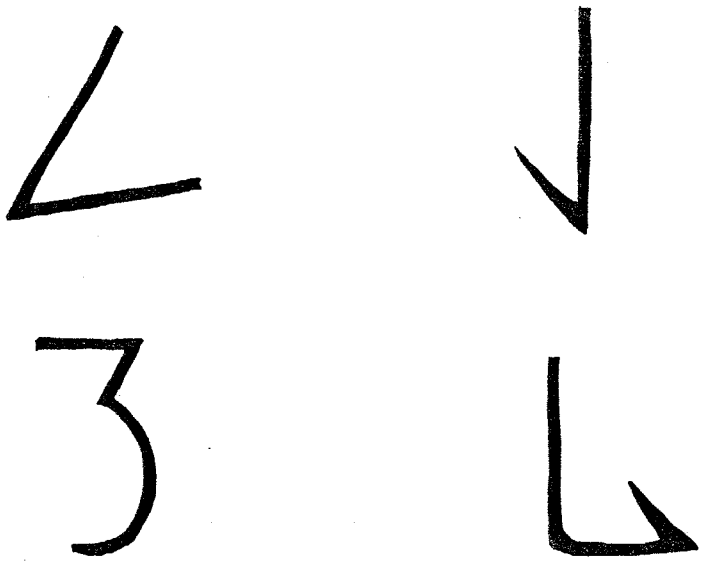
ところで、前章で述べた日本語情報処理システムにおいては、これまでのところ、上で述べたような漢字の扁旁構等の情報や、あるいは何らかの構造的な情報を一貫して有効的に利用しているものは殆どないように思われる。現在の日本語情報処理システムの目的が主として事務の機械化、即ち、帳簿伝票の作成、住所氏名の記録等の印刷機器としての役割にあることを考えれば、個々の文字の構造的情報の必要性はあまりないように思われる。しかしながら、今後の日本語情報処理について考えた場合、個々の漢字が「扁」や「旁」「構」等に相当する情報を完備していれば、なお一層利用価値の高いシステムを構成することが可能であると考えられる。しかも、個々の漢字がどのように構成され、どんな構造や特徴を有しているかを明らかにすることは、個々の漢字を如何に適切に、また如何に少ない情報で表現できるかということに直接的に係わって

くる。さらに、人工知能的立場に立った文字の生成（文字の印刷ではなく、文字を書くこと）等にとっても構造的情報は不可欠な要素であると考えられる。

これらのことを考慮すれば、漢字の構造を工学的に解析することは重要なことであり、より一層の研究が必要であると考えられる。

3-2 漢字の構造表現の背景

我々がある漢字を書く場合、その書き順が人によってまちまちになることは殆どない。これは、各文字ともその書き順が定められており、我々がその漢字を習得する際に、その書き順に従って練習を行うからである。この筆順には、「上から下」、「左から右」という大原則があり、この原則に従った幾つかの基本となる要素が存在する。その各要素においては、筆の運び、すわわち、筆跡が連続しており、これらの要素の複数個が組み合さって1個の漢字を構成する。本論文では、これらの基本となる要素をストロークと呼ぶ。図3.2.1にそのストロークの例をあげる。



（a）　（b）

図 3.2.1 ストロークの例

図3.2.1に示したストロークの特徴は、線の向きとその連続性との関係である。この筆の向きを具体的な角度として捉えるとすれば、大抵の場合、図3.2.2に示した角度において、斜線を引いた部分の方向を向いている。しかしながら、図3.2.1(b)に示した場合のように、ストロークの最後で線をはねる場合は例外的に斜線部分以外の方向へ筆が向くことがある。この線のはねという動作は一般の筆の運びと異なり、独特な筆の運び方である。

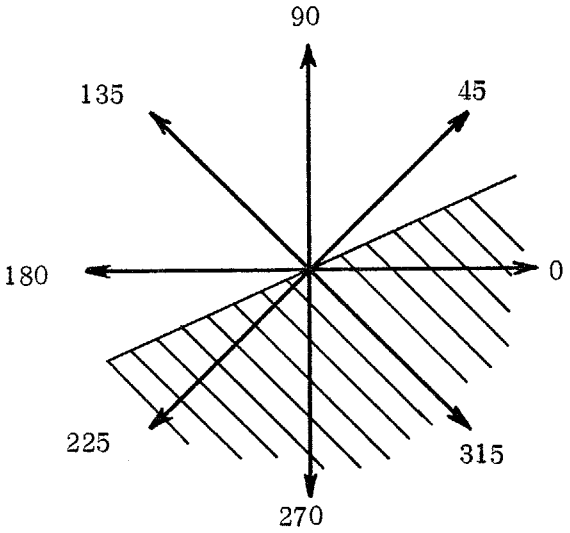
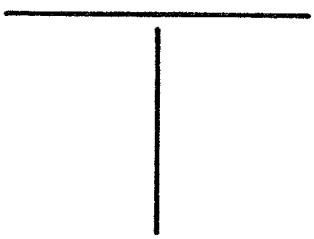


図 3・2・2
ストロークの方向

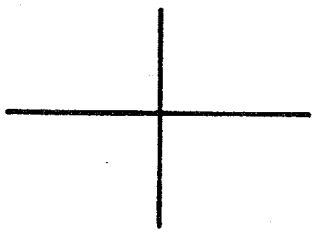
ところで、図3.2.1で示したようなストロークは、それぞれの文字の中に於いては、各部分の長さが微妙に変化し、我々が目で見て均整のとれた美しい漢字となる。従って、各ストロークの長さを固定し、その長さでもって全ての漢字を書くとすれば、およそ漢字らしくない文字となることが容易に理解できる。ストロークの持つ長さの情報を含めて考えれば、殆ど全ての漢字に対してそれぞれのストロークを定義しなければならなくなる。これは非常に不合理なことであり、本研究の主題である漢字の表現法の主旨に沿わない。ストロークの長さに関する情報は必要最小限の情報にとどめなければならない。

本論文では、図3.2.2に示したストロークの方向を量子化し、その量子化方向と連結関係によって、これまで述べてきたストロークというものに相当する基本ブロックを定義する。従って、漢字パターンはこの基本ブロックが複数個組み合さって構成されていると考えることができる。次に、その基本ブロックの組み合せ方について考える。

図3.2.3(a),(b)の例について考えてみる。両図とも同じ基本ブロック「ー」「丨」の組み合さったパターンであるが、この2つのパターンには明らかな違いがある。即ち、それぞれの基本ブロックパターンの位置的な関係に注目した場合、同図(a)については明らかに「ー」が「丨」の上にあるが、同図(b)については、2つの基本ブロックは重なり合っており、位置的関係を明確に述べることは難しい。むしろ、位置的重なりが特徴であると言ってよく、重なり合ったもの同士をまとめて一つの要素と考えれば、位置的関係を決定することが容易に行えると考えられる。



(a) 上下の位置関係にある例



(b) 重なりの位置関係にある例

図3・2・3
基本ブロックの組み合せの例 (1)

以上のことからもわかるように、基本ブロックの組み合せには2つの種類が考えられる。即ち、位置配置的な組み合せと、複数の基本ブロックをまとめて新しい要素を構成する組み合せ方である。

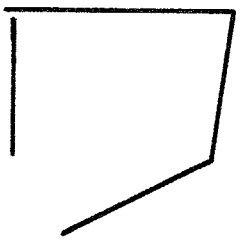


図3。2・4
基本ブロックの組み合せの例 (2)

複数の基本ブロックをまとめなければならない理由の主なものは、位置的関係を明確に決定することが難しいということであったが、図3.2.3(b)の他に図3.2.4に示したパターンにおける「丨」と「フ」の基本ブロックの間の位置関係についても同様のことが言える。このようなことを考慮すれば、複数の基本ブロックをまとめて新しい要素を構成する組み合せ方にも2つの種類が考えら

れる。本論文では、図3.2.3(b)のような関係にある基本ブロックをクロス関係にあると言い、まとめて1つのクロスブロックとする。また、図3.2.4の例のような場合、その2つの基本ブロックは近接関係にあると言い、まとめて1個の近接ブロックとする。

一方、位置的な関係について考えてみると、図3.2.3(a)の場合、「一」のパターンが「丨」の上にあると述べたが、このような上下関係の他にも、図3.2.5に示したような左右関係、あるいは図3.2.6に示した包含関係の位置関係を考えることができる。ところで、このような位置関係を決定するための要素は何であろうか。すなわち、それぞれの位置関係を決定する尺度を定義しなければならない。しかも、その尺度はブロックの種類に無関係に一様な方法で定義されるものでなければならない。このようなことから、本論文では各ブロックの占める大きさを定義し、その大きさをブロックの領域と呼ぶ。そして、それぞれのブロックの領域の縦方向、横方向の重なり具合をもとに目的に応じて位置関係を決定することができる。

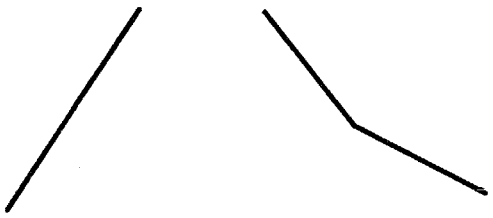


図3。2。5
左右関係にある例

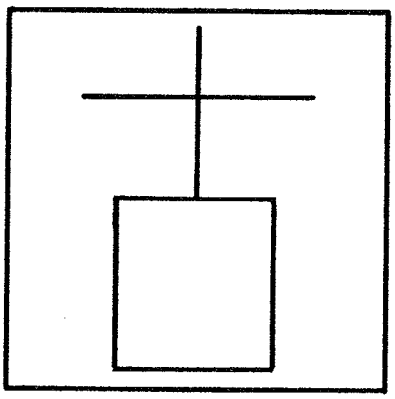


図3。2。6
包含関係にある例

以上に、漢字を構造的に記述する上で基本となる構成要素とその位置的な関係について簡単に述べた。

次節以降で、各ブロックと位置関係に関する詳細な定義を行う。

3-3 ブロックとその構成法[29),(30)(32)]

3-3-1 線素とその連結関係

前節で、漢字の構成上の基本となる幾つかのストロークに注目したが、そのストロークを工学的に定義する上で必要な要素を線素と呼ぶ。この線素は目的と用途に応じて決定しなければならないが、先に述べた漢字のストロークを定義する上で自然なものでなければならない。そこで、線素を次のように定義する。

〔定義3-1〕

線素は、始点と終点の2つの点によって定義され、その向きを始点から終点へ向かう方向とする。また、その長さを1ユニットとし、線素上の、始点、終点以外の点はその線素の種類には影響しない。

例えば、図3.2.2に示したストロークの方向角を8方向で量子化した場合は、図3.3.1に示すような8種類の線素を定義することができる。

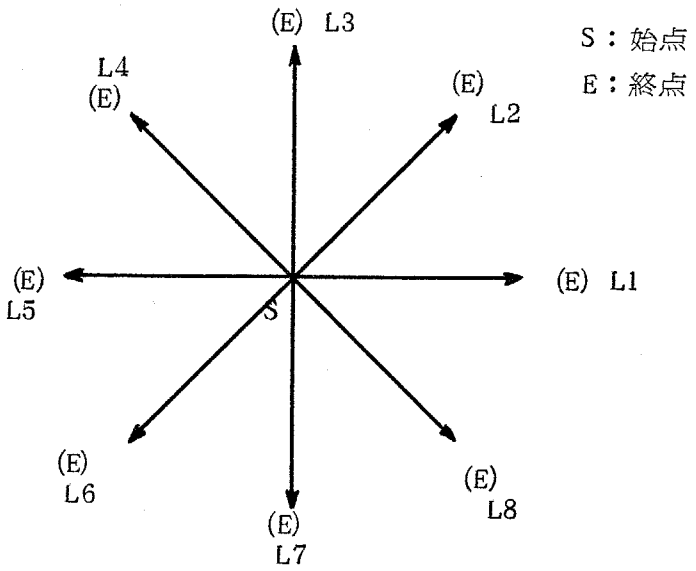


図3。3。1 線素の例

〔規則3-1〕

ある線素をl_iとするとき、l_iの始点、終点をそれぞれ head(l_i)、tail(l_i)で表現する。

線素l_iの終点と線素l_jの始点が一致しているとき、その2つの線素は連結関係にあると言い、連結関係を表わす記号$\mathbb{C}$を用いて、

$$\text{tail}(l_i)\,\mathbb{C}\,\text{head}(l_j) \tag{3-3-1}$$

と表わす。また、この連結関係によって新たに定義される要素を$[l_i, l_j]$と表わし、

$$\begin{aligned}\text{head}([l_i, l_j]) &= \text{head}(l_i)\\ \text{tail}([l_i, l_j]) &= \text{tail}(l_j)\end{aligned} \tag{3-3-2}$$

と定める。

式(3-3-1)の連結関係と式(3-3-2)の関係から、2個以上の線素からなる要素$[l_i, l_j, \cdots, l_m, l_n]$に対し、

$$\begin{aligned}\text{head}([l_i, l_j, \cdots, l_m, l_n]) &= \text{head}(l_i)\\ \text{tail}([l_i, l_j, \cdots, l_m, l_n]) &= \text{tail}(l_n)\end{aligned} \tag{3-3-3}$$

が成立する。

3-3-2 基本ブロック

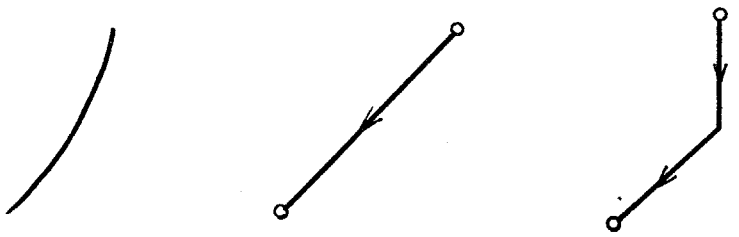
式(3-3-1)によって線素の合成が定義されたが、基本ブロックは、その線素の合成によって構成することができる。

今、n本の線素$l_1, l_2, \cdots l_n$の合成によって基本ブロックf_iが構成されているとき、f_iを

$$f_i = [l_1, l_2, \cdots, l_n] \tag{3-3-4}$$

と表わす。

この基本ブロックf_iの始点、終点は式(3-3-3)より、



(a) あるストローク　　(b) 基本ブロック 1　　(c) 基本ブロック 2

図３。３、２　ストロークと対応する基本ブロックの例

$$head(f_i) = head(l_1)$$
$$tail(f_i) = tail(l_n) \qquad (3\text{-}3\text{-}5)$$

となる。ところで、式(3-3-1)に示した連結関係によって合成される要素の数は無数に存在するが、漢字の基本要素であるストロークに相当する基本ブロックの数は、実際にはそれ程多くはない。例えば、図3.3.2(a)に示したストロークを、図3.3.1に示したストロークの８方向量子化において定義される線素の合成として表現した場合の基本ブロックf_iは、図3.3.2(b)に示したものか、あるいは、図3.3.2(c)に示した基本ブロックとして対応づけることが可能である。

3-3-3　近接ブロック

式(3-3-1)の連結関係によって基本ブロックを定義したが、ここで、基本ブロックの組み合せの一つである近接ブロックを定義するために、連結関係と類似した擬似連結関係を定義する。

〔規則3-2〕

ある２点P_1, P_2の間に、

$$|P_1 - P_2| \leq L_T$$

但し、$|P_1-P_2|$ は P_1P_2 間の距離で、L_T はあるいき値である。

であるとき、$P_1 \mathbb{C}' P_2$ と表わし、この関係 $\mathbb{C}'$ を近接関係と呼ぶ。

〔定義3-2〕

2つの基本ブロック f_i, f_j が、次に示すような近接関係にあるとき、その2つの基本ブロックで1つの近接ブロックを構成する。

(1) f_i, f_j の始点、終点に対し、

$$head(f_i)\,\mathbb{C}'\,head(f_j),\ かつ\ tail(f_i)\,\mathbb{C}'\,tail(f_j) \qquad (3-3-6)$$

の関係が成立するとき、f_i, f_j は第一種近接関係にあると言い、$N_1(f_i, f_j)$ で表わす。

(2) f_i, f_j の始点に対し、

$$head(f_i)\,\mathbb{C}'\,head(f_j) \qquad (3-3-7)$$

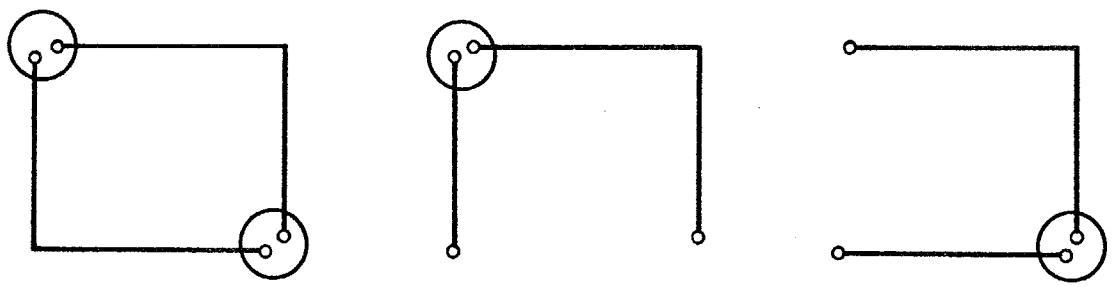
の関係が成立するとき、f_i, f_j は第二種近接関係にあると言い、$N_2(f_i, f_j)$ で表わす。

(3) f_i, f_j の終点に対し、

$$tail(f_i)\,\mathbb{C}'\,tail(f_j) \qquad (3-3-8)$$

の関係が成立するとき、f_i, f_j は第三種近接関係にあると言い、$N_3(f_i, f_j)$ で表わす。

定義3-2で述べた各近接関係の例を図3.3.3(a)～(c)に示す。



（a）第一種近接関係　　（b）第二種近接関係　　（c）第三種近接関係

図3．3．3　各近接関係の例

3-3-4 クロスブロック

前に述べた近接ブロックは、それを構成する基本ブロックの始点と終点の近接関係によって定義されたが、ここで、始点、終点以外の交点をもとに構成するクロスブロックについて述べる。

〔定義3-3〕

ある交点C_pを共有する2つのブロックをクロス関係にあると呼び、クロス関係で関係づけられるすべての基本ブロックによって1つのクロスブロックを構成する。

定義3-3によってクロス関係にある基本ブロックは1つのクロスブロックとして構成されるが、二次元的な基本ブロックの組み合せによって構成されるため、各々のクロスブロックの構造関係が一意的に表現されなければクロスブロックの判別を行うことができない。そこで、クロスブロックの構造表現の中心となる基本ブロックを定め、その基本ブロックを幹ブロックと呼ぶ。幹ブロックの選出は次の規則に従って行う。

〔規則3-3〕

幹ブロックの選出を次の順序で行う。

(1) クロス関係にあるブロックの中で最多交点数を持つブロック。

(2) クロス関係にあるブロックの中で最多線素数を持つブロック。

(3) クロス関係にあるブロックの中でカテゴリ番号最小のブロック。

規則3-3で定めた選出順序で始どの場合に幹ブロックを選出することができるが、例外的に幹ブロックを選出できない場合が存在する。例えば、図3.3.4に示したクロスブロックの場合、規則3-3の順序の第三番目の選出規定によってもf_1とf_2の2つが幹ブロックの候補として残る。このような場合、その候補の中でも、左上にある交点を含むブロックを幹ブロックとして選出する。従って、図3.3.4の例ではf_1を幹ブロックとして選出する。

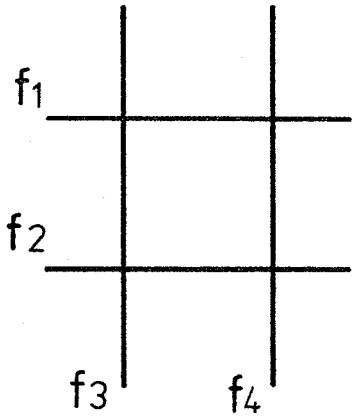


図3・3。4
幹ブロックが一意に選出されない
場合の例

幹ブロックの決定後、クロスブロックの構造表現を図3.3.5に示したアルゴリズムに従って行う。今、着目している交点をC_p、C_pを共有する基本ブロックをf_i, f_jとする。また、f_iとf_jの選出順序番号をF_i, F_j（但し、$F_i < F_j$）とする。このとき、C_pに関して次のように表現する。

$$F_i(n_i) \otimes F_j(n_j) \qquad (3\text{-}3\text{-}9)$$

式(3-3-9)において、記号⊗はクロス関係を表わす記号で、n_i, n_jはF_i, F_j番目に選出された基本ブロックf_i, f_jの線素順を表わす番号である。すなわち、交点C_pがf_i, f_jを構成する線素のn_i番目とn_j番目の線素上にあることを示す。

また、F_i, F_j番目に選出された基本ブロックの情報を$F_i = f_i$, $F_j = f_j$と与える。ところで、クロス関係にある基本ブロックの中に、さらに近接関係にあるようなブロックが存在することがある。近接関係にある基本ブロックを近接ブロックとして構成した後にクロスブロックを構成しようとしても、式(3-3-9)に示した構造表現が行えない。そこで、クロス関係と近接関係の両方の関係にある基本ブロックに対しては近接ブロックを構成せずに、交点のある基本ブロックに対する近接関係を情報として与える。即ち、F_i番目に選出された基本ブロックがF_k番目に選出される基本ブロックf_kとN_α（$\alpha = 1, 2, 3$）の近接関係にあるとき、

$$F_i = f_i \wedge N_\alpha(F_i, F_k) \qquad (3\text{-}3\text{-}10)$$

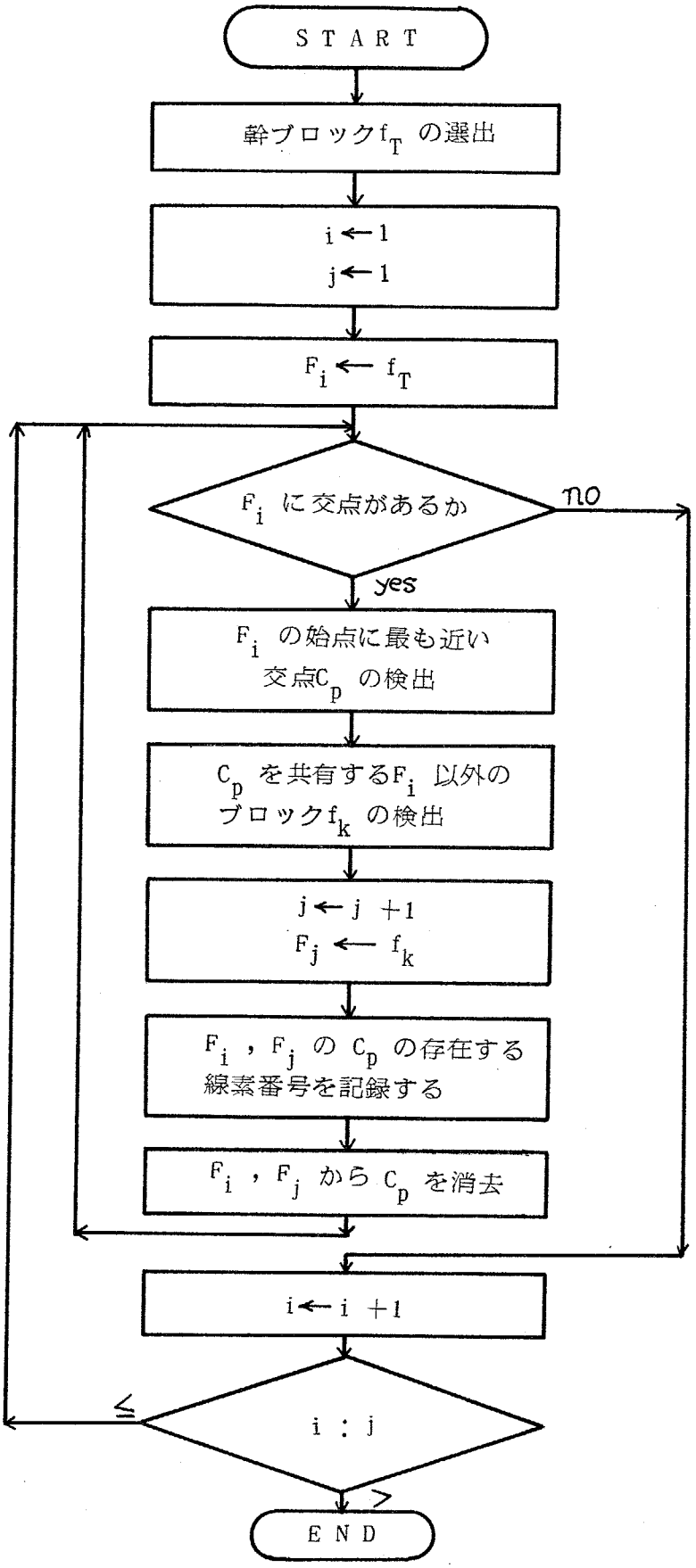
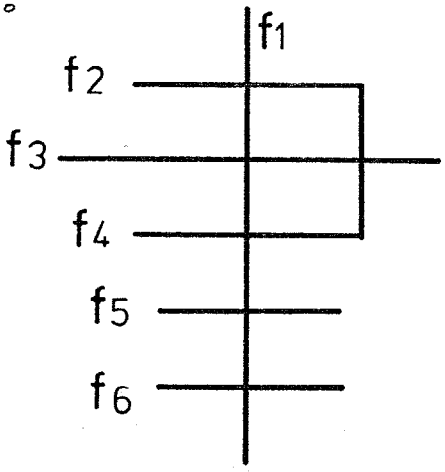


図 3.3.5 クロスブロックの記述アルゴリズム

と表わす。式(3-3-10)において、もしf_Rが他のどの基本ブロックともクロス関係にない場合は、図3.3.5に示したアルゴリズムで得られる最終選出基本ブロックの次の選出順序番号をつけてその基本ブロックを表わす。従って、近接関係のみの基本ブロックに対しては式(3-3-9)に示したクロス構造表現式は存在しない。

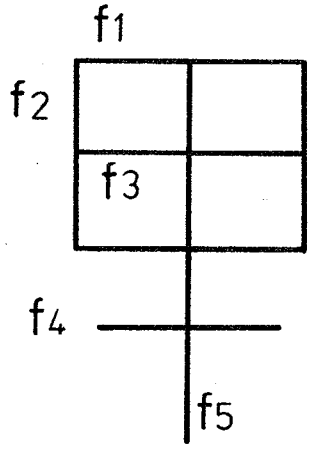
図3.3.6(a),(b)にクロスブロックの表現例を示す。同図(b)における基本ブロックf_1が近接関係のみのブロックである。

$F_1(1) \otimes F_2(1)$　$F_1 = f_1$
$F_1(1) \otimes F_3(1)$　$F_2 = f_2 \wedge N_3(F_2, F_4)$
$F_1(1) \otimes F_4(1)$　$F_3 = f_3$
$F_1(1) \otimes F_5(1)$　$F_4 = f_4 \wedge N_3(F_4, F_2)$
$F_1(1) \otimes F_6(1)$　$F_5 = f_5$
$F_2(2) \otimes F_3(1)$　$F_6 = f_6$



(a)

$F_1(1) \otimes F_2(1)$　$F_1 = f_5$
$F_1(1) \otimes F_3(2)$　$F_2 = f_3$
$F_1(1) \otimes F_4(1)$　$F_3 = f_2 \wedge N_1(F_3, F_5)$
$F_4 = f_4$
$F_5 = f_1 \wedge N_1(F_5, F_3)$



(b)

図3．3．6　クロスブロックの例

3-4 ブロック間の位置関係

前節までにおいて、漢字の構造上の基本となる線素とその連結関係を定義し、基本ブロックを構成する方法について述べた。さらに、近接関係やクロス関係を導入することによって近接ブロック、クロスブロックを定義した。この節においては、漢字を構成する各ブロック間の位置関係について述べる。

ここで、文字パターンを扱う際の座標を定める。文字画像の左上端を座標の原点にとり、横方向にy軸、縦方向にx軸をとる。文字画像がディジタル画像である場合はx座標、y座標とも整数値で、その最大値は、縦、横それぞれの方向の画素数である。

任意の漢字は、基本ブロック、近接ブロック、クロスブロックの3種類のブロックによって構成されるが、以後、これらのブロックの区別を行わない。そして、任意の漢字Kを構成するブロック集合を$\{b\}_K$で表わす。

今、$\{b\}_K$内の任意のブロックb_iを構成する各線素をl_{ji} $(j=1\sim n_i)$とする。そして、$head(l_{ji})$, $tail(l_{ji})$の座標を(x_{jiS}, y_{jiS}), (x_{jiE}, y_{jiE})とし、これらの座標から次の値を求める。

$$\left.\begin{aligned} x_{1i} &= \min_j\{x_{jiS}, x_{jiE}\}, \quad x_{2i} = \max_j\{x_{jiS}, x_{jiE}\} \\ y_{1i} &= \min_j\{y_{jiS}, y_{jiE}\}, \quad y_{2i} = \max_j\{y_{jiS}, y_{jiE}\} \end{aligned}\right\} \quad (3\text{-}4\text{-}1)$$

$$x_i = (x_{1i} + x_{2i})/2 \qquad y_i = (y_{1i} + y_{2i})/2$$

〔定義3-4〕

4点、(x_{1i}, y_{1i}), (x_{1i}, y_{2i}), (x_{2i}, y_{2i}), (x_{2i}, y_{1i})を順に結んだ線で囲まれる部分をブロックb_iの領域と呼び、$[\![b_i]\!]$で表わす。また、(x_i, y_i)を領域$[\![b_i]\!]$の代表点と呼び、p_iで表わす。

定義3-4で定めたブロックの領域は、図3.4.1に示したように、そのブロックを包含し、縦横の辺がそれぞれx軸、y軸に平行な最小の長方形であり、代表点はその長方形の対角線の交点を表わす。

さて、ブロックb_iの領域$[\![b_i]\!]$をx軸へ射影したときの領域を$[\![b_i]\!]_y$、y軸へ射影したときの領域を$[\![b_i]\!]_x$で表わす。そして、$\{b\}_K$内の任意のブロックb_i, b_jに対して、$\alpha_1(i,j)$, $\alpha_2(i,j)$を次のように定める。

$$\alpha_1(i,j)=\frac{[\![b_i]\!]_x \cap [\![b_j]\!]_x}{\min\{[\![b_i]\!]_x, [\![b_j]\!]_x\}} \qquad (3\text{-}4\text{-}2)$$

$$\alpha_2(i,j)=\frac{[\![b_i]\!]_y \cap [\![b_j]\!]_y}{\min\{[\![b_i]\!]_y, [\![b_j]\!]_y\}} \qquad (3\text{-}4\text{-}3)$$

但し、式(3-4-2), (3-4-3)における記号$\cap$は領域の交わりを表わす。この2つの式で得られる$\alpha_1(i,j)$, $\alpha_2(i,j)$は、図3.4.2の例からもわかるように、ブロックb_i, b_jの領域をx軸、y軸方向へ射影したときのb_iの領域とb_jの領域の重なりを表わし、

$$0 \le \alpha_1(i,j)=\alpha_1(j,i) \le 1, \quad 0 \le \alpha_2(i,j)=\alpha_2(j,i) \le 1 \qquad (3\text{-}4\text{-}4)$$

の関係がある。この$\alpha_1(i,j)$、$\alpha_2(i,j)$の値により、b_iとb_jの間の位置関係を、用途に応じて定義することができる。

以上のような観点から漢字を眺めた場合、各ストロークが複雑に組み合さって構成されているにもかかわらず、この章で定義したブロックの概念と、その各ブロック間の位置的な関係によって漢字を構造的に記述することが可能であると考えられる。しかも、これまで述べた各種のブロックの概念は、幾つかの例外を除き、漢字の構造上の特質を十分に捉えたものであり、我々が漢字に対して持っているイメージと相通じるものであると考えられる。

このような漢字の構成要素である各種のブロックと、それらの位置的な関係による漢字の構造表現法について次章以降で述べる。

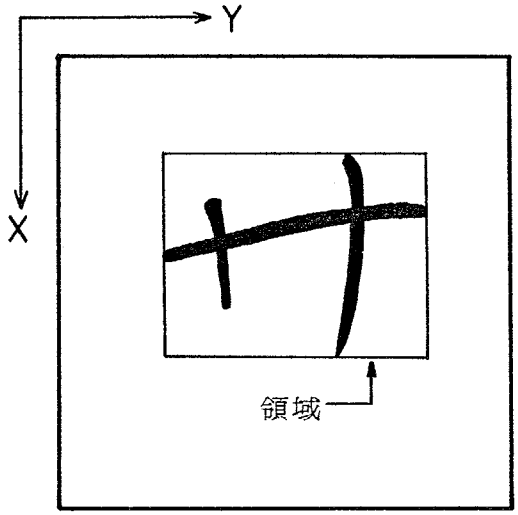


図 3・4・1　ブロックの領域の例

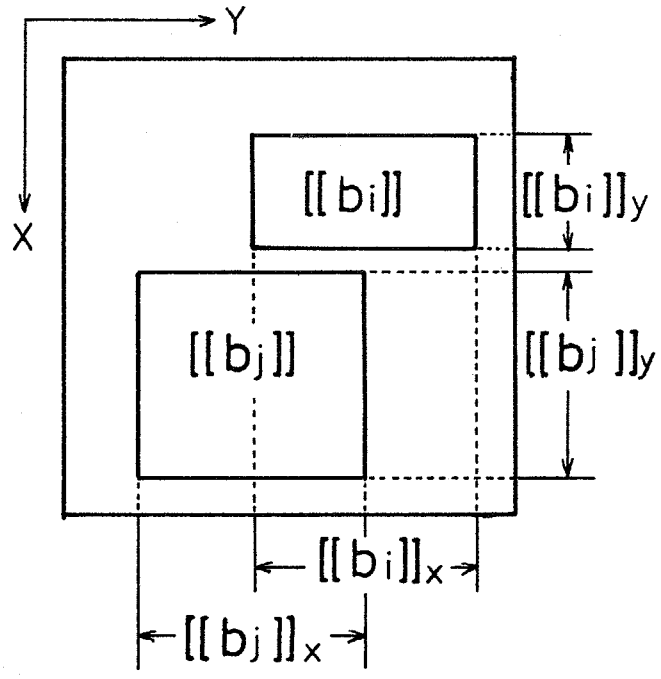


図 3・4・2　ブロックの領域の射影例

第四章　漢字パターンの表現法[30]～[33].

4-1　漢字パターンの表現の目的とその意義

前章で定義した漢字パターンの要素となるブロックとその位置的な関係を利用することによって、漢字パターンを構造的に表現することが可能であることを述べた。

ところで、現在ある日本語情報処理システムにおいて、個々の漢字を構造的に表現することによって情報量の圧縮を計ったり、あるいは、漢字の検索等に利用しているようなシステムは殆ど見当らない。では、漢字を構造的に表現することによってどんな利点が生じるであろうか。もし、我々が漢和辞典を引くときのように、「冫」や「冖」、「構」等の部首、あるいは画数等の情報が個々の漢字パターンの表現式に含まれているとすれば、その表現式を計算機に記憶させておくことによって漢和辞典と同様な利用法が可能になり、さらには、文字の合成、生成等を自由に行うことが可能となるであろう。

一方、文字として与えられている場合、即ち、印刷された漢字や手書きによる漢字パターンに対して、何らかの方法によって構造的な表現式を得ることができるとすれば、あらかじめ個々の漢字パターンの標準的な表現式を作成しておくことによってパターン認識への応用も可能になると考えられる。しかしながら、手書き漢字のパターン認識への応用を目的とするような場合には、表現の正確さ、緻密さ等を要求すればする程、表現の再現性が低下してしまい、パターン認識という目的を達成することが困難になる。従って、パターン認識への応用を目的とする場合の表現式においては、表現の正確さ、緻密さ等よりも、位置や大きさの変化、あるいは線の歪曲等の文字変形に対して安定で、表現の再現性の高い表現方法でなければならない。

4-2 ブロックの領域による位置関係

漢字パターンを表現する場合、その表現式に対して文字パターンの再現性を要求する場合と、パターンの再現性までは要求せず、単に構造的な表現を目的とする場合とでは、表現式の持つ情報量や表現の形態が異なる場合が多い。この章において述べる表現法は、パターンの再現性を要求されない場合の表現式を得る方法について述べるものである。

前章の式(3-4-2),(3-4-3)によって{b}K内の任意のブロックbi, bjの領域をx軸、y軸へ射影した場合のそれぞれの重なり度を定義した。そのα1(i,j), α2(i,j)を用いて3種類の位置関係を定義する。(31)

[定義4-1]

(1) {b}K内の任意のブロックbi, bjにおいて、

$$\left.\begin{array}{l}[\![b_i]\!] > [\![b_j]\!] \text{ かつ、} \\ \alpha_1(i,j)=1,\ \alpha_2(i,j)=1\end{array}\right\} (4\text{-}2\text{-}1)$$

のとき、biはbjを含む。但し、記号>は領域の面積の大小関係を示す不等号である。

(2) {b}K内の任意のブロックbi, bjにおいて、

$$\left.\begin{array}{l}x_i < x_j \quad \text{かつ} \\ \alpha_1(i,j) \geq C_1\end{array}\right\} (4\text{-}2\text{-}2)$$

のとき、biはbjの上にある。

(3) {b}K内の任意のブロックbi, bjにおいて、

$$\left.\begin{array}{l}y_i < y_j \quad \text{かつ} \\ \alpha_2(i,j) \geq C_2\end{array}\right\} (4\text{-}2\text{-}3)$$

のとき、biはbjの左にある。

但し、(x_i, y_i), (x_j, y_j)は領域$[\![b_i]\!]$, $[\![b_j]\!]$の代表点の座標であり、パラメータC1, C2は、上下、左右の関係を決定するための値である。

定義4-1に従って、包含関係、上下関係、左右関係の3種類の位置関係に限定してしまえば、その位置関係からだけではパターンの正確な再構成は不可能である。しかし、これらの3種類の位置関係によって、前節で述べたような構造表現の特質を十分に備えた、簡潔な表現式を得ることができる。

位置関係に関し、次のような規則を定める。

〔規則4-1〕

2つのブロック間に同時に2つ以上の位置関係が成立するとき、その優先順序を、包含関係、上下関係、左右関係の順とする。

〔規則4-2〕

$\{b\}_k$内の任意のブロックb_i, b_jに対し、定義4-1で得られた3種類の位置関係を次の式によって表わす。

$$\langle i / R_x / j \rangle \quad \text{但し、} x \in X, X=\{0, \pm1, \pm2, \pm3\} \qquad (4\text{-}2\text{-}4)$$

即ち、

(1) $x=3$のとき、b_iはb_jを含む。

(2) $x=2$のとき、b_iはb_jの上にある。

(3) $x=1$のとき、b_iはb_jの左にある。

(4) $x=0$のとき、b_iとb_jの間にはいずれの位置関係も成立しない。

また、式(4-2-4)に関して

$$\langle i / R_x / j \rangle \simeq \langle j / R_{-x} / i \rangle \qquad (4\text{-}2\text{-}5)$$

が成立する。但し、$\simeq$は同値関係を表わす記号である。

一方、$\{b\}_k$内のブロック数がmであるとき、$m \times m$のマトリックスPに対して、次のように各成分を求める。

$$P(i,j) = g(\langle i / R_x / j \rangle) = x \qquad (4\text{-}2\text{-}6)$$

但し、$x \in X$, $g(\cdot)$はb_iとb_jの位置関係を集合Xの元xへ対応づける関数である。このマトリックスPを相対位置表現マトリックスと呼ぶ。

式(4-2-5), (4-2-6)より、

$$|P(i,j)| = |P(j,i)| \qquad (4\text{-}2\text{-}7)$$

である。

ところで、図4.2.1, 図4.2.2の例のような場合、b_iとb_kの間には定義4-1で述べたいずれの位置関係も成立しない。このような場合、次の補題に従って位置関係を修正する。

〔補題4-1〕

相対位置表現マトリックスPにおいて、

$|P(i,j)|=|P(j,k)|=2$, かつ、

$|P(i,k)|=0$ のとき、

$x_i < x_k$ ならば

$P(i,k)=2$, $P(k,i)=-2$

$x_i > x_k$ ならば、

$P(i,k)=-2$, $P(k,i)=2$

とする。

また、

$|P(i,j)|=2$, $|P(j,k)|=1$, かつ

任意の$b_\ell \in \{b\}_k$に対して、

$|P(k,\ell)| \neq 2$ のとき、

$x_i < x_k$ ならば

$P(i,k)=2$, $P(k,i)=-2$

$x_i > x_k$ ならば、

$P(i,k)=-2$, $P(k,i)=2$

とする。

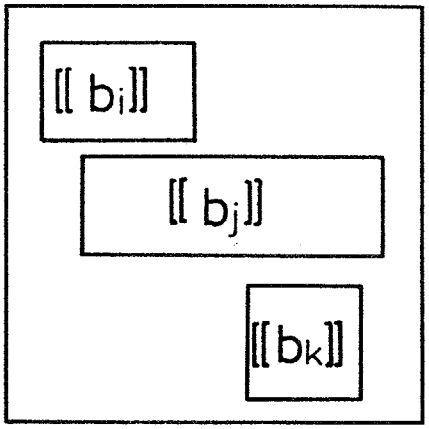


図 4・2・1 ブロックの配置例 (1)

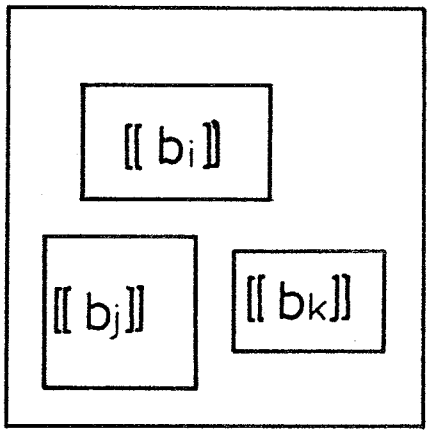


図 4・2・2 ブロックの配置例 (2)

以上に述べてきた3種類の位置関係は、いずれも異なる2つのブロック間で定義されたものであり、同一のブロックに対する位置関係は未定義である。そのため、相対位置表現マトリックスの対角成分が定義されない。そこで、同一のブロックに対する位置関係を次に示す定義で定める。

〔定義4-2〕

同一ブロックに対する位置関係は、すべての$x \in X$に対して、

$$\langle i / Rx / i \rangle$$

の関係が成立するとする。しかし、$|x_1| \neq |x_2|$, $x_1, x_2 \in X$に対して同時には成立しない。また、相対位置表現マトリックスの対角成分を便宜上0として表わす。

この節で定義した3種類の位置関係と、それを表わす相対位置表現マトリックスPをもとに、$\{b\}_K$内のブロックを分類する。その分類法については次の節で述べる。

4-3 位置関係によるブロックの分類

漢字を構成するブロック間の位置関係は、前節の式(4-2-6)によって相対位置表現マトリックスPによって表現されている。このマトリックスPは、任意の2つのブロック間の位置関係を明確に表わすことができるが、逆に、冗長性の少ない情報となってしまう。また、情報量の圧縮という点に関しても、マトリックスP自体による位置関係の表現法はそれ程有効な方法とは考えられない。そこで、さらに少ない情報量で、しかも冗長性のある漢字パターンの表現法について述べる。

最初に、マトリックスPを用いて任意の漢字を構成するブロック集合$\{b\}_K$を、以下に示すアルゴリズムによって分類する。

〔分類アルゴリズム〕

ステップ1

$R_{\{3\}}$、即ち包含関係にあるブロックを分類する。

ある$b_i \in \{b\}_K$に対し、

$$Q_i = \{ b_j \mid P(i,j) = 3,\ b_j \in \{b\}_K,\ b_i \neq b_j \} \qquad (4-3-1)$$

を求め、Qiが空集合でないときbiをbi°とする。

各ブロックに対して式(4-3-1)で表わされる集合を求めた後、各Qiに含まれるブロックを{b}kから削除する。

ステップ2

ステップ1の操作後、残ったブロック集合を$\{\overline{b}\}_k$とし、$\{\overline{b}\}_k$を上下関係で分類する。

今、ある集合Sを{b}kとする。S内のブロックbiに対し、

$$\left.\begin{aligned} W_i^+ &= \{ b_j \mid |P(i,j)| = 2,\ b_j \in S \} \\ W_i^- &= \bigcup_{b_j \in W_i^+} \{ b_R \mid |P(j,R)| = 2,\ b_R \in S \} \\ W_i &= W_i^+ \cup W_i^- \end{aligned}\right\} \quad (4-3-2)$$

を求める。式(4-3-2)より、

$$W_i^- = \bigcup_{b_j \in W_i^+} W_j^+$$

であり、

$$W_i = W_i^+ \cup \left(\bigcup_{b_j \in W_i^+} W_j^+ \right) \quad (4-3-3)$$

となる。Wiに含まれるブロックをSから削除する。削除後、分類されないブロックがあれば同様の操作を繰り返す。

ステップ3

ステップ2で得られた、ある1つの集合Weを左右関係で分類する。

今、ある集合TをWeとする。T内のブロックbiに対し、

$$\left.\begin{aligned} We_i^+ &= \{ b_j \mid |P(i,j)| = 1,\ b_j \in T \} \\ We_i^- &= \bigcup_{b_j \in We_i^+} \{ b_R \mid |P(j,k)| = 1,\ b_R \in T \} \end{aligned}\right\} \quad (4-3-4)$$

を求め、

$$We_i = We_i^+ \cup We_i^-$$

とする。W_{ki} に含まれるブロックを T から削除し、T 内に残るブロックがなくなるまで同様の操作を繰り返す。

ステップ4

式(4-3-2)、(4-3-4)の分類操作を分類の対操作として、ステップ3の操作で得られた部分集合の中で最も上にある部分集合から分類し、その部分集合に含まれるすべてのブロックが1個に分類されるまで同様の操作を繰り返す。

ステップ5

包含関係を有するブロック b_i^* が1個に分類されたとき、b_{ki} に対応する集合 Q_{ki} に対し、ステップ2からステップ4までの操作を適用して Q_{ki} に含まれるブロックを分類する。

ステップ3の分類は、ステップ2で得られる部分集合のうち最も左にある部分集合から行い、その集合の分類が完全に終了した後に次の部分集合の分類へ移る。

以上のブロックの分類を図示したものが図4.3.1である。

次に、この分類法によって得られる幾つかの定理について述べる。

〔定理4-1〕

ステップ1を除くブロックの分類の各段階で得られる任意の部分集合内のどのブロックに対しても、式(4-3-2)、または式(4-3-4)で得られる集合は等しくなる。

〔証明〕

ステップ2の場合について証明する。

今、式(4-3-2)によって $b_i \in \overline{\{b\}}_k$ に対して得られた部分集合 W_i に含まれる任意のブロックを b_j, b_j に対して式(4-3-2)で得られる部分集合を W_j とする。

W_i に含まれ、W_j に含まれないブロック b_ℓ が存在すると仮定し、b_ℓ が W_n^+ ($b_n \in W_i^+$) に含まれているとすれば、W_n^+ 内の任意のブロック b_g に

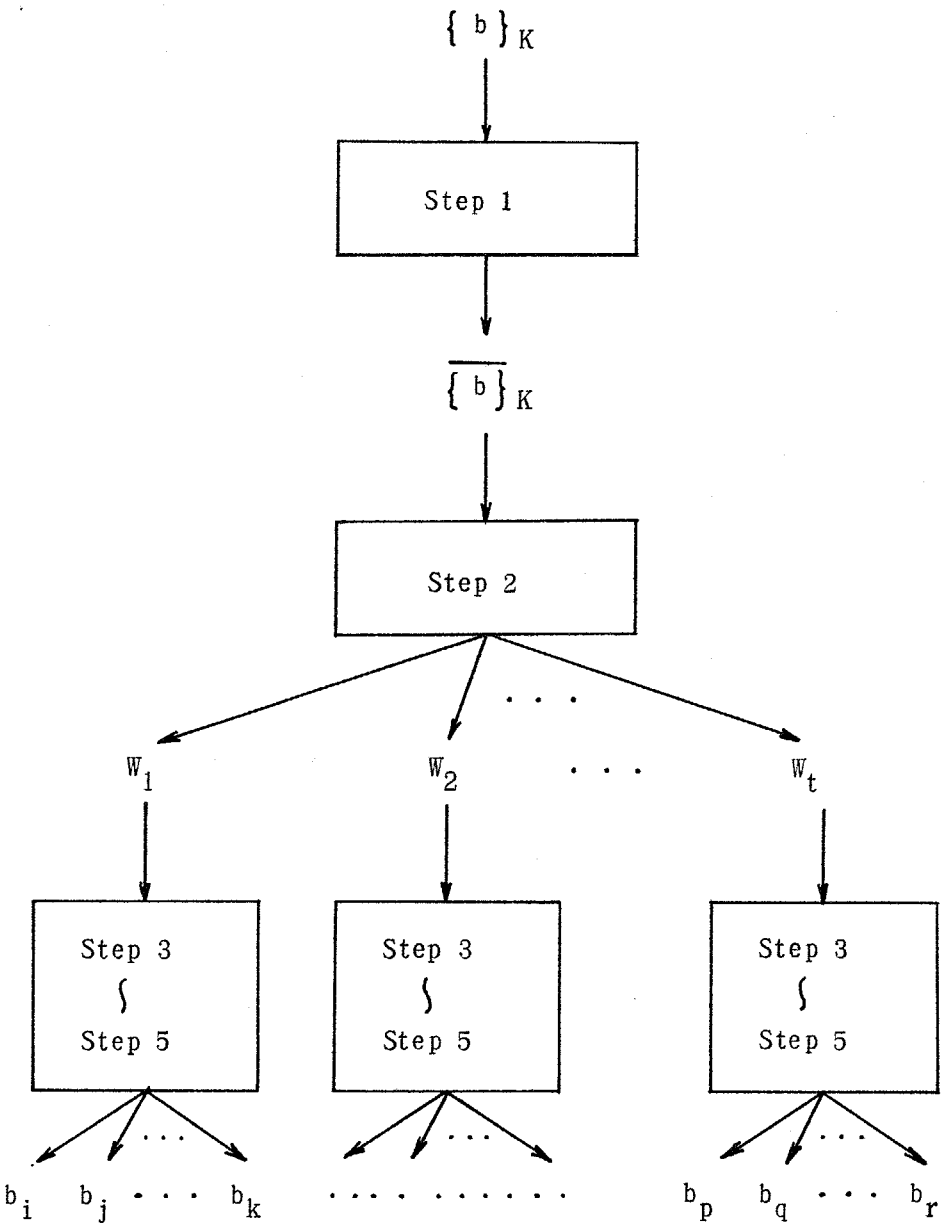


図 4.3.1 ブロックの分類の概略図

対して、$|P(\ell,g)|=2$が成立する。従って、W_jはW_m^+内のすべてのブロックを含まないことになる。同様なことを繰り返せば、W_jはW_i内のすべてのブロックを含まないことになり、b_jがW_iに含まれるということに矛盾する。従って、W_iとW_jは等しくなければならない。

他の分類段階の場合についても同様である。　〔証明終り〕

定理4-1より、どのブロックに着目して分類を行ったかということは、得られる部分集合に無関係となる。そこで、$\{b\}_k$の分類に関して各分類段階で得られる集合の添字を分類順に付けなおす。即ち、

ステップ2で得られる集合を、$W=\{W_1, W_2, \cdots, W_r\}$

ステップ3で、$W_\ell \in W$に対して得られる集合を、$W_\ell=\{W_{\ell 1}, W_{\ell 2}, \cdots, W_{\ell n}\}$

以下、$W_{mn} \in W_m$に対しても、$W_{mn}=\{W_{mn1}, W_{mn2}, \cdots, W_{mnt}\}$等とする。

ところで、ブロックの分類段階で、ある集合に対して得られる部分集合間にも、ブロックの分類法の定義から上下関係と左右関係の位置関係が定義される。

〔規則4-3〕

ブロックの各分類段階で得られる各部分集合間の位置関係も式(4-2-4)に従って表現する。

〔定理4-2〕

$W_i, W_j, W_k, \in W$, $W_{mni}, W_{mnj}, W_{mnk} \in W_{mn} \in W_m \in W$

に対して、

(1) $\langle i / R_{|\pm 1|} / i \rangle$

(2) $\langle i / R_{|\pm 1|} / j \rangle \Rightarrow \langle j / R_{|\pm 1|} / i \rangle$

(3) $\langle i / R_{|\pm 1|} / j \rangle$かつ$\langle j / R_{|\pm 1|} / k \rangle \Rightarrow \langle i / R_{|\pm 1|} / k \rangle$

が成立する。

従って、WとW_{mn}は$R_{|\pm 1|}$に関して同値類となる。

［定理4-3］

$W_{mi}, W_{mj}, W_{mk} \in W_m \in W$に対して

(1) $\langle i / R_{|\pm 2|} / i \rangle$

(2) $\langle i / R_{|\pm 2|} / j \rangle \Rightarrow \langle j / R_{|\pm 2|} / i \rangle$

(3) $\langle i / R_{|\pm 2|} / j \rangle$かつ$\langle j / R_{|\pm 2|} / k \rangle \Rightarrow \langle i / R_{|\pm 2|} / k \rangle$

が成立する。

従って、W_mは$R_{|\pm 2|}$に関して同値類となる。

定理4-2、定理4-3はブロックの分類法より明らかである。また、定理4-2定理4-3より、$\{W_1, W_2, \cdots, W_r\}$は$R_{|\pm 2|}$によるWの商集合であり、$\{W_{m1}, W_{m2}, \cdots, W_{mt}\}$は$R_{|\pm 1|}$による$W_m$の商集合になっている。

以上に述べてきた分類方法を、図4.3.2の例をもとに説明する。但し、図4.3.2の例に対する相対位置表現マトリックスは、表4.3.1の様に作成されているものとする。

まず、ステップ1において、包含関係が成立するb_1, b_2に対し、$Q_1 = \{b_2\}$を求め、b_2を$\{b\}_K$から削除する。次に、ステップ2の分類によってW_iを求める。分類の結果、$W_1 = \{b_1^\circ, b_3, b_4\}$, $W_2 = \{b, b_6, b_7, b_8, b_9\}$が得られる。続いてステップ3の分類により$W_1 = \{W_{11}, W_{12}\}$, $W_{11} = \{b_1^\circ\}$, $W_{12} = \{b_3, b_4\}$が得られ、ステップ4によってW_{12}がさらに分類されて$W_{12} = \{W_{121}, W_{122}\}$, $W_{121} = \{b_3\}$, $W_{122} = \{b_4\}$となる。W_1の分類が終了したので、次にW_2に対して同様にステップ3を適用すると、$W_2 = \{W_{21}, W_{22}, W_{23}\}$, $W_{21} = \{b_5\}$, $W_{22} = \{b_6\}$, $W_{23} = \{b_7, b_8, b_9\}$が得られる。続いて$W_{23}$にステップ4を適用して$W_{23} = \{W_{231}, W_{232}\}$, $W_{231} = \{b_7\}$, $W_{232} = \{b_8, b_9\}$と分類する。さらに、W_{232}は$W_{2321} = \{b_8\}$, $W_{2322} = \{b_9\}$に分類され、すべてのブロックの分類が終了する。包含関係にあるブロックb_1°に対応する集合Q_1の要素も1個であるため、ステップ5の分類を適用する必要がない。

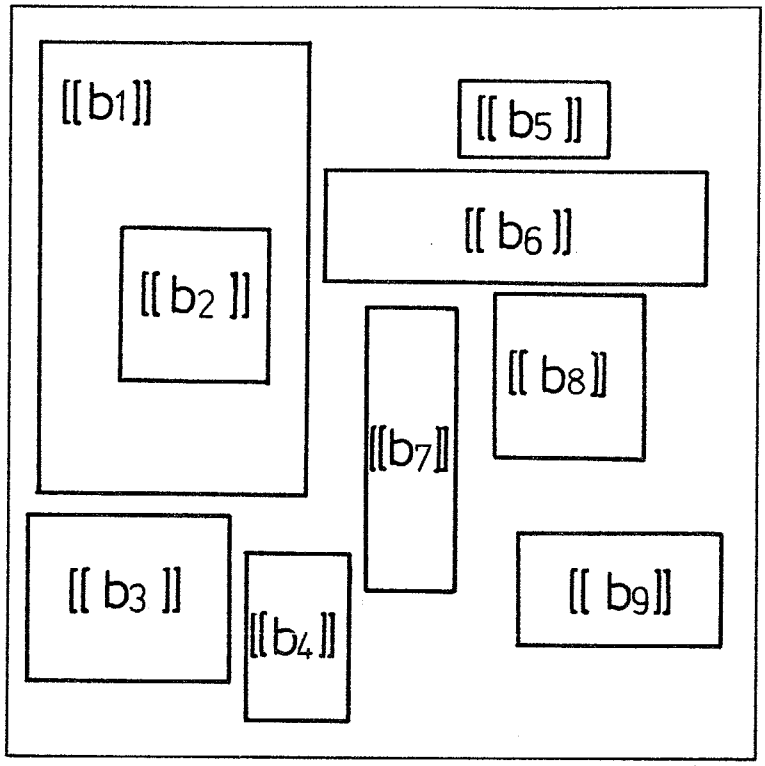


図４・３・２　ブロックの領域とその配置例

表 ４・３・１　　図 ４・３・２の例に対する相対位置表現マトリックス

	b_1	b_2	b_3	b_4	b_5	b_6	b_7	b_8	b_9
b_1	0	3	2	2	1	1	1	1	0
b_2	-3	0	2	0	0	0	1	1	0
b_3	-2	-2	0	1	0	0	0	0	1
b_4	-2	0	-1	0	0	0	0	0	1
b_5	-1	0	0	0	0	2	0	2	2
b_6	-1	0	0	0	-2	0	2	2	2
b_7	-1	-1	0	0	0	-2	0	1	1
b_8	-1	-1	0	0	-2	-2	-1	0	2
b_9	0	0	-1	-1	-2	-2	-1	-2	0

4-4 分類結果による漢字パターンの表現

前節で述べた分類法によって $\{b\}_k$ を分類し、その結果をもとに漢字パターンを表現する。その表現のために、R_1, R_2, R_3 のそれぞれの位置関係を、→, ↓, ＊の記号で表わす。

集合Wについて考える。

$W=\{W_1, W_2, \cdots, W_r\}$ の要素間に成立する位置関係は、定理4-2 により $R_{[1]}$, すなわち左右関係である。従って、Wの任意の要素 W_k に対して次に示す関係式を満たすある要素 W_s が存在する。

$$\langle S / R_1 / k \rangle \quad S \neq k,\ W_s, W_k \in W \qquad (4\text{-}4\text{-}1)$$

式(4-4-1)を満たすSを

$$\sigma(1) = S \qquad (4\text{-}4\text{-}2)$$

とおき、W_s をWから削除する。同様にして $\sigma(2), \sigma(3), \cdots \sigma(r)$ を求める。その結果、前に定めた記号→によって

$$W_{\sigma(1)} \rightarrow W_{\sigma(2)} \rightarrow \cdots \rightarrow W_{\sigma(r)} \qquad (4\text{-}4\text{-}3)$$

と表わすことができる。

同様に、ステップ3で得られるある商集合 $W_m=\{W_{m1}, W_{m2}, \cdots W_{mt}\}$ に対しても、R_2 を表わす記号↓によって

$$W_{m\sigma(1)} \downarrow W_{m\sigma(2)} \downarrow \cdots \downarrow W_{m\sigma(t)} \qquad (4\text{-}4\text{-}4)$$

と表わすことができる。ステップ3以下の分類で得られる各商集合に対しても式(4-4-3), (4-4-4)のように表わすことができる。

もしも、各部分集合に含まれるブロックが1個である場合は、その部分集合の所へ、その要素であるブロックを入れる。

一方、包含関係を有するブロック b_v° が1個に分類された場合は、

$$b_v * Q_v \qquad (4\text{-}4\text{-}5)$$

と記号＊を用いて包含関係を表わし、Q_v の分類結果に対して式(4-4-3),

式(4-4-4)と同様に各表現式を求める。

ある分類段階で得られた任意の部分集合を、次の分類段階で得られるその部分集合の商集合の表現式で置き換えていき、すべての部分集合がその要素である1個のブロックによって置き換えられたときの表現式によって、漢字パターンKの表現式とする。

図4.3.2の例に対する各分類段階における部分集合間の関係を各記号を用いて表現すると、

$$\left.\begin{array}{ll} W=\{W_1, W_2\} & W_1 \rightarrow W_2 \\ W_1=\{W_{11}, W_{12}\} & W_{11} \downarrow W_{12} \\ W_{12}=\{W_{121}, W_{122}\} & W_{121} \rightarrow W_{122} \\ W_2=\{W_{21}, W_{22}, W_{23}\} & W_{21} \downarrow W_{22} \downarrow W_{23} \\ W_{23}=\{W_{231}, W_{232}\} & W_{231} \rightarrow W_{232} \\ W_{232}=\{W_{2321}, W_{2322}\} & W_{2321} \downarrow W_{2322} \end{array}\right\} \quad (4\text{-}4\text{-}6)$$

となる。各部分集合を、それに含まれるブロックで置き換え、さらに包含関係にあるブロックをも表現すると、この例に与えられたパターンの表現式は次のように得られる。

$$(b_1 * b_2 \downarrow (b_3 \rightarrow b_4)) \rightarrow (b_5 \downarrow b_6 \downarrow (b_7 \rightarrow (b_8 \downarrow b_9))) \quad (4\text{-}4\text{-}7)$$

第五章 漢字パターンの符号化法

5-1 ブロックの分類法と表現式の変換[33]~[34]

前章において、漢字パターンを式(4-4-7)のような形式で表現する表現法について述べた。ところで、式(4-4-7)の形式の表現式は、ブロックの分類の最終結果を表わす式で、その表現式の中には位置関係表現記号や括弧が含まれている。従って、文字認識等に式(4-4-7)の形式の表現式を利用するためには構文解析的方法を用いるか、あるいは、何らかの形に変換するなどの操作を必要とし、その処理過が複雑になると考えられる。そこで、この問題を解決すると共に、式(4-4-6)で与えられるようなブロックの分類の各段階で得られる中間結果を効果的に利用でき、しかも、式(4-4-7)の表現式と同等な情報を持つ漢字パターンの符号化法について述べる。

今、ある部分集合$a_1, a_2, \cdots, a_t \in A$があり、$\{a_1, a_2, \cdots, a_t\}$が、ある位置的な同値関係によるAの商集合になっているとする。そして、これらの部分集合間に、位置的な関係を表わす2項間位置関係記号Δが定義され、各部分集合において

$$a_1 \Delta a_2 \Delta a_3 \Delta \cdots \Delta a_t \qquad (5\text{-}1\text{-}1)$$

の関係が成立しているものとする。

また、式(5-1-1)において、2項間位置関係記号Δに関して結合則が成立するものとする。このとき、式(5-1-1)に対して演算子$\mathbb{D}$を次のように定義する。

〔定義5-1〕

$$\left.\begin{aligned} &\mathbb{D}(a_1 \Delta a_2 \Delta \cdots \Delta a_t) = \mathbb{D}(a_1) \cdot \mathbb{D}(a_2 \Delta a_3 \Delta \cdots \Delta a_t) \Delta \\ &\mathbb{D}(a_1) = a_1 \end{aligned}\right\} \quad (5\text{-}1\text{-}2)$$

式(5-1-1)で与えられる集合間の関係式に、定義5-1で定めた演算子𝔻を作用させ、すべてのa_iを演算子𝔻の外に出すためには、次に示すような(t-1)回の一連の演算が必要である。

$$\left.\begin{aligned}
&\mathbb{D}(a_1 \Delta a_2 \Delta \cdots \Delta a_t) = a_1 \cdot \mathbb{D}(a_2 \Delta a_3 \Delta \cdots \Delta a_t) \Delta \\
&\mathbb{D}(a_2 \Delta a_3 \Delta \cdots \Delta a_t) = a_2 \cdot \mathbb{D}(a_3 \Delta a_4 \Delta \cdots \Delta a_t) \Delta \\
&\quad\vdots \\
&\mathbb{D}(a_{t-1} \Delta a_t) = a_{t-1} \cdot a_t \Delta
\end{aligned}\right\} \quad (5\text{-}1\text{-}3)$$

式(5-1-1)で与えられる関係式に対して、式(5-1-3)で示したように各部分集合を演算子𝔻の外に出す一連の演算を$\mathbb{D}^+$で表わす。この演算$\mathbb{D}^+$によって得られる変換式は、式(5-1-3)より、

$$a_1 a_2 \cdots a_{t-1} a_t \Delta \Delta \cdots \Delta \qquad (5\text{-}1\text{-}4)$$

となる。

さて、式(5-1-3)で与えられる各式に対して演算子𝔹を次のように定義する。

〔定義5-2〕

$$\left.\begin{aligned}
&\mathbb{B}(\mathbb{D}(a_i) \cdot \mathbb{D}(a_{i+1} \Delta a_{i+2} \Delta \cdots \Delta a_t)) \\
&= \mathbb{B}(\mathbb{D}(a_i)) \cdot \mathbb{B}(\mathbb{D}(a_{i+1} \Delta a_{i+} \Delta \cdots \Delta a_t)) \Delta \\
&\mathbb{B}(\mathbb{D}(a_i \Delta a_{i+1} \Delta \cdots \Delta a_t)) = \sum_{j=i}^{t} |a_j|
\end{aligned}\right\} \quad (5\text{-}1\text{-}5)$$

但し、$|a_j|$は集合a_jの要素数を表わす。

演算$\mathbb{D}^+$によって得られる一連の式に対して演算子𝔹を作用させる演算を、$\mathbb{B}\cdot\mathbb{D}^+$で表わす。

ところで、前章の4節で述べたように、各分類段階で得られる各部分集合間の位置的な関係は、R_1, R_2の2項間の位置関係を表わすそれぞれの記号→, ↓によって式(5-1-1)のような形式で表現することができた。また、包含関係の場合も、式(4-4-5)のように2項間の位置的関係を示しており、包含関係を表わす記号＊による表現式も式(5-1-1)と同形式の表現式とみなすことができる。

従って、式(5-1-1)の2項間位置関係記号Δとしては、→, ↓, ＊の3種類の記号がすべてあてはまり、ブロックの分類の各段階で得られる表現式に対して演算$B \cdot D^{+}$を作用させることができる。

さて、第3章で述べた漢字の構造的な性質は、漢字の本質的な構成要素という観点から眺めたものであったが、その構成要素の部分的なまとまりとして眺めた場合、「語」、「誤」等の文字は、「言」、「吾」、「呉」の漢字が組み合さって　つの文字を構成していることがわかる。このような種類の文字が多いことも漢字の構造上の特徴である。このような特徴を利用すれば、漢字パターンの効率よい符号化を行うことができると考えられる。

ところで、前章で述べたブロックの分類における基本的分類操作は、次に示す2つの操作であった。

（操作1）

ある部分集合$U \subseteq \overline{\{b\}}_K$内の任意のブロック$b_i$に対し、

$$S_i^{+} = \{ b_j \mid |P(i,j)| = 2,\ b_j \in U \}$$

$$S_i^{-} = \bigcup_{b_j \in S_i^{+}} \{ b_k \mid |P(j,k)| = 2,\ b_k \in U \}$$

$$S_i = S_i^{+} \cup S_i^{-}$$

を求め、S_iに含まれるブロックをUから除く。Uに含まれるブロックがすべて分類されるまで上の操作を繰り返す。

（操作2）

ある部分集合$U \subseteq \overline{\{b\}}_K$内の任意のブロック$b_i$に対し、

$$S_i^{+} = \{ b_j \mid |P(i,j)| = 1,\ b_j \in U \}$$

$$S_i^{-} = \bigcup_{b_j \in S_i^{+}} \{ b_k \mid |P(j,k)| = 1,\ b_k \in U \}$$

$$S_i = S_i^{+} \cup S_i^{-}$$

を求め、S_iに含まれるブロックをUから除く。Uに含まれるブロックがすべて分類されるまで上の操作を繰り返す。

以上の2つの基本的分類操作と演算$\mathbb{B}\cdot\mathbb{D}^+$とを組み合せた分類法について述べるが、その分類法は前章で述べたブロックの分類法と本質的には異ならない。

分類アルゴリズム

〔ステップ1〕

$\{b\}_K$内の各ブロックb_iに対し、

$$Q_i = \{ b_j \mid P(i,j)=3,\ b_j \in \{b\}_K \} \tag{5-1-6}$$

$$n(i) = 1 + |Q_i| \tag{5-1-7}$$

但し、$|Q_i|$はQ_iの要素数を表わす。

を求め、$|Q_i| \neq 0$のとき、b_iをb_i°とする。次に、空でない2つの集合Q_j, Q_kにおいて、$Q_k \subset Q_j$であるとき、Q_jからQ_kの要素を除く。以上の操作の後、各Q_iに含まれるブロックを$\{b\}_K$から除く。

ステップ2

ステップ1の後、残ったブロック集合$\overline{\{b\}}_K$に対して操作1を行う。その結果、t個の部分集合が得られたとすれば、t個の部分集合間には左右関係のみが成立する。最も左にあるものから順に$u_1, u_2, \cdots u_t$とすれば、

$$u_1 \to u_2 \to u_3 \cdots \to u_t \tag{5-1-8}$$

と表わすことができる。この表現式に演算$\mathbb{B}\cdot\mathbb{D}^+$を行えば、次に示すような$(t-1)$個の変換式が得られる。

$$\left.\begin{aligned}
&\left(\sum_{j=1}^{t} |u_j|\right) = |u_1| \cdot \left(\sum_{j=2}^{t} |u_j|\right) \to \\
&\left(\sum_{j=2}^{t} |u_j|\right) = |u_2| \cdot \left(\sum_{j=3}^{t} |u_j|\right) \to \\
&\quad\vdots \qquad\qquad\qquad\quad \vdots \\
&\left(\sum_{j=t-1}^{t} |u_j|\right) = |u_{t-1}|\,|u_t| \to
\end{aligned}\right\} \tag{5-1-9}$$

ここで、各$|u_i|$の値を、ステップ1で求めた$n(i)$の値を用いて次

のように定義する。

$$|U_i| = \sum_{b_j \in U_i} n(j) \qquad (5\text{-}1\text{-}9)$$

もし、U_i の中に包含関係を有するブロック b_j° が含まれているとき、Q_j に含まれるブロックはステップ1によって除かれているため、U_i の中には含まれないが、本来は U_i に含まれると考えられる。従って、広義の意味で U_i の要素数を式(5-1-9)で定義する。

(ステップ3)

ステップ2で得られた、ある一つの部分集合 U_i に対し、操作2を適用する。その結果、得られた部分集合が g 個であるとき、g 個の部分集合間の位置関係は上下関係のみである。最も上にあるものから順に $U_{i1}, U_{i2}, \cdots U_{ig}$ とすれば、

$$U_{i1} \downarrow U_{i2} \downarrow U_{i3} \cdots \downarrow U_{ig} \qquad (5\text{-}1\text{-}10)$$

と表わすことができる。この表現式に対して演算 $B \cdot D^+$ を行えば、次に示すような $(g-1)$ 個の変換式が得られる。

$$\left.\begin{aligned}
&\left(\sum_{j=1}^{g} |U_{ij}|\right) = |U_{i1}| \cdot \left(\sum_{j=2}^{g} |U_{ij}|\right) \downarrow \\
&\left(\sum_{j=2}^{g} |U_{ij}|\right) = |U_{i2}| \cdot \left(\sum_{j=3}^{g} |U_{ij}|\right) \downarrow \\
&\quad\vdots \qquad\qquad\qquad\qquad \vdots \\
&\left(\sum_{j=g-1}^{g} |U_{ij}|\right) = |U_{ig-1}|\,|U_{ig}| \downarrow
\end{aligned}\right\} \qquad (5\text{-}1\text{-}11)$$

ステップ2の場合と同様に $|U_{ij}|$ を次のように定義する。

$$|U_{ij}| = \sum_{b_k \in U_{ij}} n(k) \qquad (5\text{-}1\text{-}12)$$

ステップ4

ステップ3で得られた部分集合の中に、ブロックを2個以上含む部分集合があるとき、その中で最も上にある部分集合から操作1と操作2によってすべてのブロックが1個に分類されるまで分類し、同時に各変換式を求める。

ステップ5

包含関係を有するブロックb_i°が1個に分類されたとき、b_iに対応する集合Q_iに対し、ステップ2からステップ4によって分類を行ない、各変換式を求める。

なお、ステップ3以後の操作は、ステップ3で得られたも個の部分集合の中で最も左に位置する部分集合から適用する。以上に述べた分類法を概念的に示したものが図4.3.1であった。

また、包含関係を有するブロックb_i°については、$b_i * Q_i$と表わされるが、ステップ2からステップ4までの分類段階においてb_i°が1個に分類されたとき、その時点で$b_i * Q_i$の関係式に演算$B \cdot D^{+}$を作用させて変換式を求める。続いて、Q_iに対してステップ2からステップ4を適用し、Q_iに含まれるブロックを分類して各変換式を求める。Q_iの分類が終了した時点で、Q_iの分類へ移る直前の分類段階へ戻る。もしも、Q_iの分類段階において包含関係を有するブロックがさらに1個に分類された場合、その処理操作は上述の操作と同様に行う。

以上に述べた分類法は、ステップ2で得られる各部分集合がそれぞれ、ステップ3からステップ5までの操作によって分類されるものであるが、これは、前に述べた漢字パターンの構造上の特徴を、漢字パターンの効果的な符号化に利用するための分類法である。

5-2 文生成文法の導入[35)~37)]

前節において、演算子𝔻と𝔹を定義し、ブロックの分類段階の各結果から変換式を求めたが、その一連の変換式を文生成過程に対応づけるために、文生成文法を導入する。

符号化の対象とする漢字パターンの中で、最も多くブロックを含む漢字パターンのブロック数をMとする。

今、ある変換式

$$\begin{aligned}&\mathbb{B}(\mathbb{D}(u_i \Delta u_{i+1} \Delta \cdots \Delta u_t)) \\ &= \mathbb{B}(\mathbb{D}(u_i)) \cdot \mathbb{B}(\mathbb{D}(u_{i+1} \Delta u_{i+2} \cdots \Delta u_t)) \Delta\end{aligned} \qquad (5\text{-}2\text{-}1)$$

に於いて、

$\mathbb{B}(\mathbb{D}(u_i \Delta u_{i+1} \cdots \Delta u_t))$, $\mathbb{B}(\mathbb{D}(u_i))$, $\mathbb{B}(\mathbb{D}(u_{i+1} \Delta u_{i+2} \cdots \Delta u_t))$の各項が、前節の式(5-1-9),(5-1-12)に従って求めた結果、p, q, r の各値になるとき、各項に対してB_p, B_q, B_rという記号をつける。このとき、p, q, r の各値の間には、

$$p = q + r \qquad (5\text{-}2\text{-}2)$$

の関係が成立する。

ここで、ある文法Gを次のように定義する。[11)]

[定義5-3]

$$G = (V_N, V_T, P, S) \qquad (5\text{-}2\text{-}3)$$

但し、V_N, V_T, P, Sはそれぞれ、非終端記号集合、終端記号集合、文生成規則集合、そして開始記号であり、V_N, V_Tは

$$\begin{aligned}V_N &= \{B_2, B_3, B_4, \cdots, B_M, S\} \\ V_T &= \{B_1, \rightarrow, \downarrow, *\}\end{aligned} \qquad (5\text{-}2\text{-}4)$$

である。また、文生成規則集合の各要素を次のように定める。

$$
\begin{array}{ll}
P_1 \quad S \to B_1 & P_{M+6};\ B_3 \to B_1B_2 \rightarrow \\
P_2 \quad S \to B_2 & P_{M+7};\ B_3 \to B_1B_2 \downarrow \\
\vdots \quad\ \vdots & P_{M+8};\ B_3 \to B_1B_2 * \\
P_M \quad S \to B_M & P_{M+9};\ B_4 \to B_3B_1 \rightarrow \\
P_{M+1} \quad B_2 \to B_1B_1 \rightarrow & P_{M+10};\ B_4 \to B_3B_1 \downarrow \\
P_{M+2};\ B_2 \to B_1B_1 \downarrow & P_{M+11};\ B_4 \to B_2B_2 \rightarrow \\
P_{M+3};\ B_2 \to B_1B_1 * & P_{M+12};\ B_4 \to B_2B_2 \downarrow \\
P_{M+4};\ B_3 \to B_2B_1 \rightarrow & \vdots \qquad \vdots \\
P_{M+5};\ B_3 \to B_2B_1 \downarrow & P_{MAX};\ B_M \to B_1B_{M-1} *
\end{array}
\qquad (5\text{-}2\text{-}5)
$$

この文法G_Tを導入することによって、演算$B\cdot D^+$により得られる各変換式は文法Gで定めた生成規則のいずれかと一対一に対応させることができる。さて、各変換式がどの生成規則と対応しているかを判定しなければならないが、その判定は、ブロックの分類法から一意に決定することができる。

式(5-2-1)で示した変換式に対応する生成規則は、次式で与えられるyの値をもつ生成規則P_yである。

$$y = p^2 - 2g - 2 + M + \chi(\Delta) \qquad (5\text{-}2\text{-}6)$$

但し、

$$\chi(\Delta) = \begin{cases} 1 \cdots \Delta\text{が} \rightarrow \text{のとき} \\ 2 \cdots \Delta\text{が} \downarrow \text{のとき} \\ 3 \cdots \Delta\text{が} * \text{のとき} \end{cases}$$

式(5-2-6)は文法Gの中で定義した生成規則の順序によって決まる関数で、生成規則の順序が変われば式(5-2-6)の関数式も変化する。

式(5-2-6)より、生成規則の個数MAXは、

$$MAX = M^2 + M - 1 \qquad (5\text{-}2\text{-}7)$$

となり、Mの2乗に比例して増大する。

一方、ステップ2の分類において得られる部分集合が$u_1, u_2, \cdots, u_t$である

とき、ステップ2の分類に対し、開始記号で始まる規則として、

$$P_{m_0}\ ;\ S \rightarrow B_{m_0} \quad \text{但し、} m_0 = \sum_{j=1}^{t} |u_j| \qquad (5\text{-}2\text{-}8)$$

を対応させる。又、各部分集合u_iに対してステップ3以後の分類を行うときは、開始記号で始まる生成規則として、

$$P_{m_i}\ ;\ S \rightarrow B_{m_i} \quad \text{但し、} m_i = |u_i| \qquad (5\text{-}2\text{-}9)$$

を対応させる。

なお、文法Gの中で定義した生成規則の中には、図5.2.1に示したような、包含関係にない複数のブロックが同一のブロックを含むような場合に相当する生成規則は含まれていない。本論文で述べた表現法においては、図5.2.1の例のような関係を定義していないからである。しかしながら、包含関係の成立条件等を変えることによって、本研究の表現方法で扱うことも可能である。

また、本方法は漢字パターンの認識や分類、検索等を行うための表現、符号化を目的として、ブロックの分類過程を生成規則に対応づけているものであり、定義5-3で導入した生成文法のみによっては漢字パターンの生成を行うことはできない。

ブロックの分類に対応して、任意の漢字パターンに対し、生成規則列が得られるがその生成規則列とは別に、式(4-4-7)の形式の表現式の最も左にあるブロックから順に並べたものをブロックのカテゴリー列として求める。このブロックのカテゴリー列は、ブロックの分類段階に於いて求めることができる。このようにして得られる生成規則列とブロックのカテゴリー列とで、漢字パターンを符号化する。

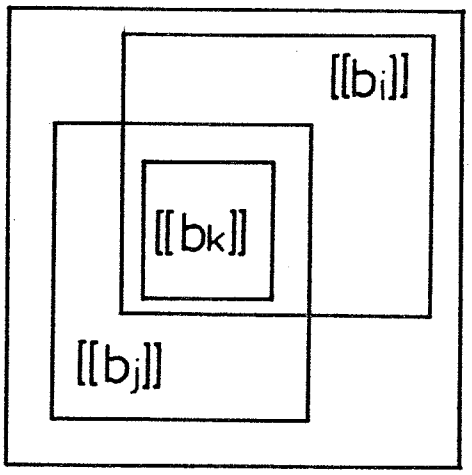


図 5.2.1
複数のブロックが同一の
ブロックを含む例

図5.2.2の例をもとに、ブロックの分類段階の各結果から生成規則列とブロックのカテゴリー列を求める方法を概略的に説明する。なお、図5.2.2のブロックの配置例に於ける各ブロックの位置関係は表5.2.1のように与えられているものとする。

ブロック集合$\{b\}_K$は$\{b_1, b_2, \cdots, b_8\}$で、まずステップ1で包含関係にあるブロックを分類する。すなわち、$Q_2=\{b_3, b_4\}$, $Q_i=\{\phi\}$ ($i=1\sim8$, 2を除く)が得られ、b_2をb_2°と改める。各Q_iの値と式(5-1-7)より、$n(i)$の値は、$n(2)=3$, $n(i)=1$ ($i=2$を除く)となる。ステップ2の上下関係による分類によって2つの部分集合$u_1=\{b_1, b_2^\circ\}$、$u_2=\{b_5, b_6, b_7, b_8\}$が得られ、その関係は$u_1 \rightarrow u_2$と表わされる。次に、$u_1$と$u_2$に対してステップ3以後の操作を行う。$u_1$については、$u_{11}=\{b_1\}$, $u_{12}=\{b_2^\circ\}$が得られ、$u_{11} \downarrow u_{12}$の関係で表わされる。包含関係を有するブロックb_2°が1個に分類されたので、$b_2 * Q_2$と表わし、この表現式の変換式を求める。続いてQ_2を分類する。その結果、$Q_{21}=\{b_3\}$, $Q_{22}=\{b_4\}$が得られ、$Q_{21} \downarrow Q_{22}$と表わされる。同様に、u_2についても、$u_{21}=\{b_5\}$, $u_{22}=\{b_6\}$, $u_{23}=\{b_7, b_8\}$, $u_{231}=\{b_7\}$, $u_{232}=\{b_8\}$と分類され、それぞれ、$u_{21} \downarrow u_{22} \downarrow u_{23}$ $u_{231} \rightarrow u_{232}$の各表現式が得られる。以上の各分類段階において得られる変換式に対応する生成規則と生成過程、それに、ブロックが1個に分類された時点、とそのブロックを示したものが表5.2.2である。この例では、ブロックが1個に分類された順に並べることによってブロックのカテゴリー列が得られる。なお、同表に示した文生成過程は、ブロックの分類と対応させて参考のために示したものであり、生成文法によって独立に導かれるものではない。

この節において定義した生成文法によって導出される文と、四章で述べた表現式との関係について次節で概略的に述べる。

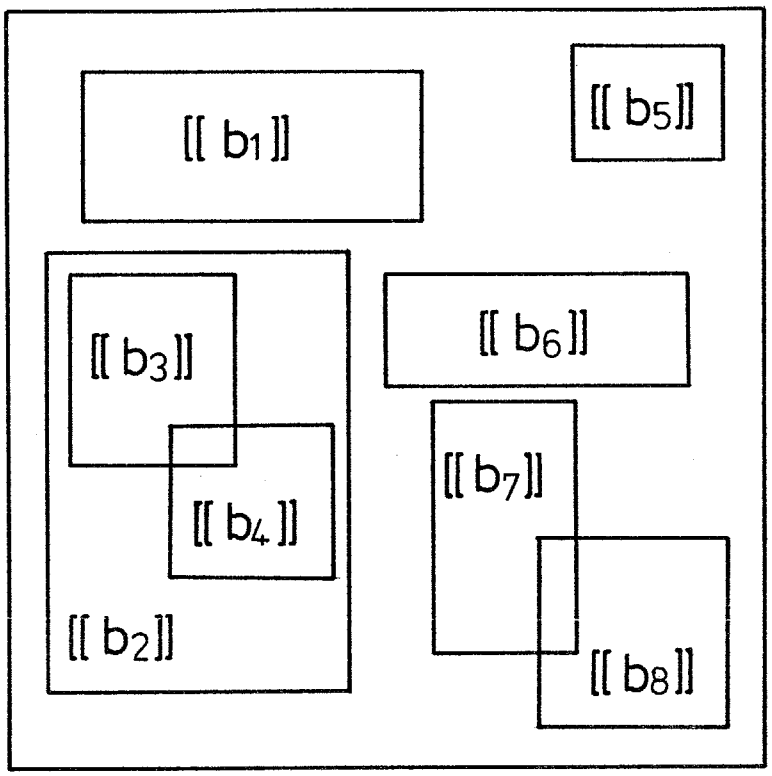


図 5・2・2　ブロックの領域とその配置例

表 5・2・1　図5・2・2の例に対する相対位置表現マトリックス

	b_1	b_2	b_3	b_4	b_5	b_6	b_7	b_8
b_1	0	2	2	2	1	0	0	0
b_2	-2	0	3	3	0	1	1	1
b_3	-2	-3	0	2	0	1	0	0
b_4	-2	-3	-2	0	0	0	1	1
b_5	-1	0	0	0	0	2	0	2
b_6	0	-1	-1	0	-2	0	2	2
b_7	0	-1	0	-1	0	-2	0	1
b_8	0	-1	0	-1	-2	-2	-1	0

表 5・2・2

図 5・2・2の例におけるブロックの分類に対応して得られる生成規則とブロック

生成規則			ブロック	文生成過程	分類段階
1	P_8	$S \rightarrow B_8$			ステップ 2
2	P_{M+55}	$B_8 \rightarrow B_4 B_4 \rightarrow$		$B_4 B_4 \rightarrow$	
1	P_4	$S \rightarrow B_4$			ステップ 3 (U_1の分類)
2	P_{M+14}	$B_4 \rightarrow B_1 B_3 \downarrow$	b_1, b_2	$B_1 B_3 \downarrow$	
3	P_{M+8}	$B_3 \rightarrow B_1 B_2 *$		$B_1 B_1 B_2 * \downarrow$	
4	P_{M+2}	$B_2 \rightarrow B_1 B_1 \downarrow$	b_3, b_4	$B_1 B_1 B_1 B_1 \downarrow * \downarrow$	
1	P_4	$S \rightarrow B_4$			ステップ 3 (U_2の分類)
2	P_{M+14}	$B_4 \rightarrow B_1 B_3 \downarrow$	b_5	$B_1 B_3 \downarrow$	
3	P_{M+7}	$B_3 \rightarrow B_1 B_2 \downarrow$	b_6	$B_1 B_1 B_2 \downarrow \downarrow$	
4	P_{M+1}	$B_2 \rightarrow B_1 B_1 \rightarrow$	b_7, b_8	$B_1 B_1 B_1 B_1 \rightarrow \downarrow \downarrow$	

5-3 漢字パターンの表現式と生成文法

前章において導いた式(4-4-7)の形式の表現式は、漢字パターンの構造とその構成要素との関係を端的に表現していると考えられる。ところで、ブロックの分類によって得られる商集合の各要素は同一の位置関係によって表現されるが、それらの商集合の各表現式から式(4-4-7)の形式の最終的な表現式を得る作業はそれほど容易なことではない。従って、漢字パターンを表現すること自体が目的でなく、表現式を種々の情報処理の手段として利用するということを考慮した場合、式(4-4-7)の形式の表現式では、少ない情報で効率的処理可能な表現形式という利点が損われてしまう。式(4-4-7)の表現式のもつ利点を活用し、なおかつ、その表現式のもつ短所を補うことを目的として、この章において生成文法を導入した。この生成文法と式(4-4-7)の形式の表現式との関係について、算術式の例と対比しながら簡単に説明する。

我々が一般に使っている日本語や英語などの自然言語に対して、計算機のプログラム言語等の形式言語があるが、算術式も形式言語の一種と考えられる。各言語にはそれぞれ文法が定められており、各文はその文法に従って書かれている。算術式の文法については省略するが、[39)~(41)]

$$(a+b)\times c+(d-e)\div f \qquad (5\text{-}3\text{-}1)$$

という式は、文法に照らして見るまでもなく正しく表現された算術式であることが容易に分かる。我々は、算術式を見るときは全体を見渡せるため、どこから演算を開始するか、演算結果と次の演算の相手、またその演算の種類は何かということを容易に判定することができる。これに対して、計算機によって演算を行わせようとした場合、計算機はシリアルに一項毎に見ていかなければならないために、その演算式が正しい表現式であるかを直ちに判定することもできず、また、正しい演算式だからと言ってすぐに演算を開始することもできない。計算機によって式(5-3-1)のような演算を行うためには、構文解析等によ

って慣用表現の演算式を機械にとって便利な表現形式に変換してやらなければならない。その中間形式の表現式の一つにポストフィックス(postfix)表現というものがある。この表現法は、演算子をオペランド(operand:被演算項)のあとにおく表現方法で、逆ポーランド記法(Reverse Polish notation)と同等な表現法である。(42) この表現法によって表現された式においては、その式の左から順に見ていき、最初に現われる2項演算記号を、その前にある2個のオペランドに働らかせて1個の値とするという規則があり、括弧の不要な数式の表現法である。(40) 例えば、式(5-3-1)の数式をポストフィックス表現すると、

$$a\,b+c\times d\,e-f\div +\qquad (5\text{-}3\text{-}2)$$

と表わすことができる。

式(5-3-2)の形式の表現が機械にとって便利であるのは、スタックを用いて容易に演算を行うことができるからである。すなわち、左から見ていき、一つの変数や定数を読み出すたびにスタックにプッシュダウンして書き込む。演算記号が現われたら、スタックの最上位とつぎの最上位のデータをポップアップ(pop up)して読み出し、その間で演算を行って、その結果をスタックにプッシュ・ダウンする。これを続けていき、中間形式表現の最後までいくと、スタックの最上位にその演算の結果が残る。式(5-3-2)の例に対するスタックの変化を図示したものが図5.3.1である。

式(5-3-1)のように慣用表現された算術式を式(5-3-2)のようなポストフィックス表現に変換するアルゴリズムは確立され、計算機のコンパイリング等に利用されているが、その方法等については省略する。

一方、前章で述べた漢字パターンの表現式も算術式の場合と同様にポストフィックス表現で表わすことができる。例えば、式(4-4-7)の表現式、

$$(b_1 * b_2 \downarrow (b_3 \rightarrow b_4)) \rightarrow (b_5 \downarrow b_6 \downarrow (b_7 \rightarrow (b_8 \downarrow b_9)))\qquad (5\text{-}3\text{-}3)$$

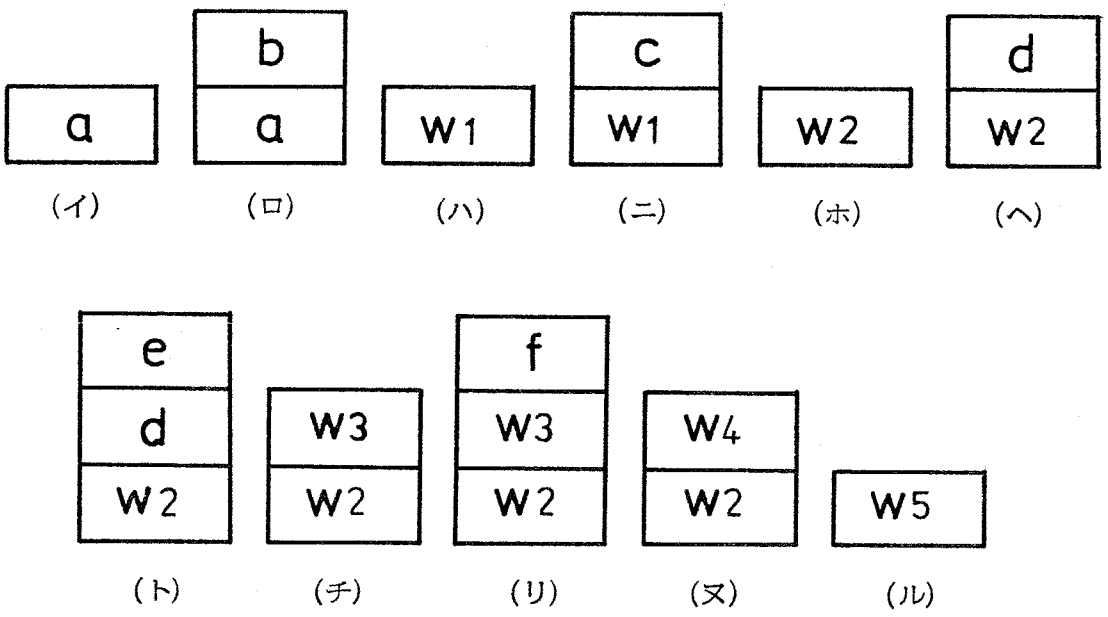
に対しては、

$$b_1 b_2 * b_3 b_4 \rightarrow \downarrow\ b_5 b_6 \downarrow b_7 b_8 b_9 \downarrow \rightarrow \downarrow \rightarrow \qquad (5\text{-}3\text{-}4)$$

と表わすことができる。

a b + c × d e - f ÷ +

(イ) (ロ) (ハ) (ニ) (ホ) (ヘ) (ト) (チ) (リ) (ヌ) (ル)



$$w1 = a + b$$
$$w2 = w1 \times c$$
$$w3 = d - e$$
$$w4 = w3 \div f$$
$$w5 = w2 + w4$$

図 5・3・1 スタックの変化

計算機で処理を行う場合、式(5-3-1)の形式よりも式(5-3-2)の形式の表現式の方が都合がいいのと同様に、漢字パターンの表現式においても式(5-3-3)よりも式(5-3-4)の形式の方が処理の容易さ、記憶容量等において有利である。しかし、式(5-3-4)の表現式を得るためには、まず式(5-3-3)の表現式がなければならない。ところで、ある漢字パターンが与えられたとき、そのパターンから式(5-3-4)の表現式を得るまでの過程を、大まかに算術式の変換過程にたとえてみると、図5.3.2のようになると考えられる。同図において、これまで述べたように、慣用的なインフィックス表現からポストフィックス表現に変換することは比較的容易に行うことができた。しかしながら、自然言語表現をインフィックス表現の算術式に変換する操作は、現時点においては非常に困難な作業である。同様に、漢字パターンの表現式においても、ブロックの位置関係による分類からインフィックス表現式を求めることは、インフィックス表現をポストフィックス表現に変換する操作に比べて労力の要する作業である。そこで、ブロックの分類操作から直接的にポストフィックス表現に相当する情報を得ようとしたのが、この章での文法導入であった。しかも、ポストフィックス表現におけるブロック情報と構造情報の独立的扱いの困難性を解決するために、ブロックの個数と位置関係を表わす情報とブロックのカテゴリ情報とに分離し、それぞれ独自に参照できるような情報構造になっている。

図5.3.2の漢字パターンのポストフィックス表現式の生成過程とブロックの分類過程とを対比してみると、

$W_1=\{b_1,b_2,b_3,b_4\}$; $S\rightarrow B_4$;($S\Rightarrow B_4$

$W_{11}=\{b_1\}, W_{12}=\{b_2,b_3,b_4\}$

$W_{11}\downarrow W_{12}$; $B_4\rightarrow B_1B_3\downarrow$;($\Rightarrow B_1B_3\downarrow$

$W_{121}=\{b_2\}, W_{122}=\{b_3,b_4\}$

$W_{121}\rightarrow W_{122}$; $B_3\rightarrow B_1B_2\rightarrow$;($\Rightarrow B_1B_1B_2\rightarrow\downarrow$

$W_{1221}=\{b_3\}, W_{1222}=\{b_4\}$

$W_{1221}\downarrow W_{1222}$; $B_2\rightarrow B_1B_1\downarrow$;($\Rightarrow B_1B_1B_1B_1\downarrow\rightarrow\downarrow$

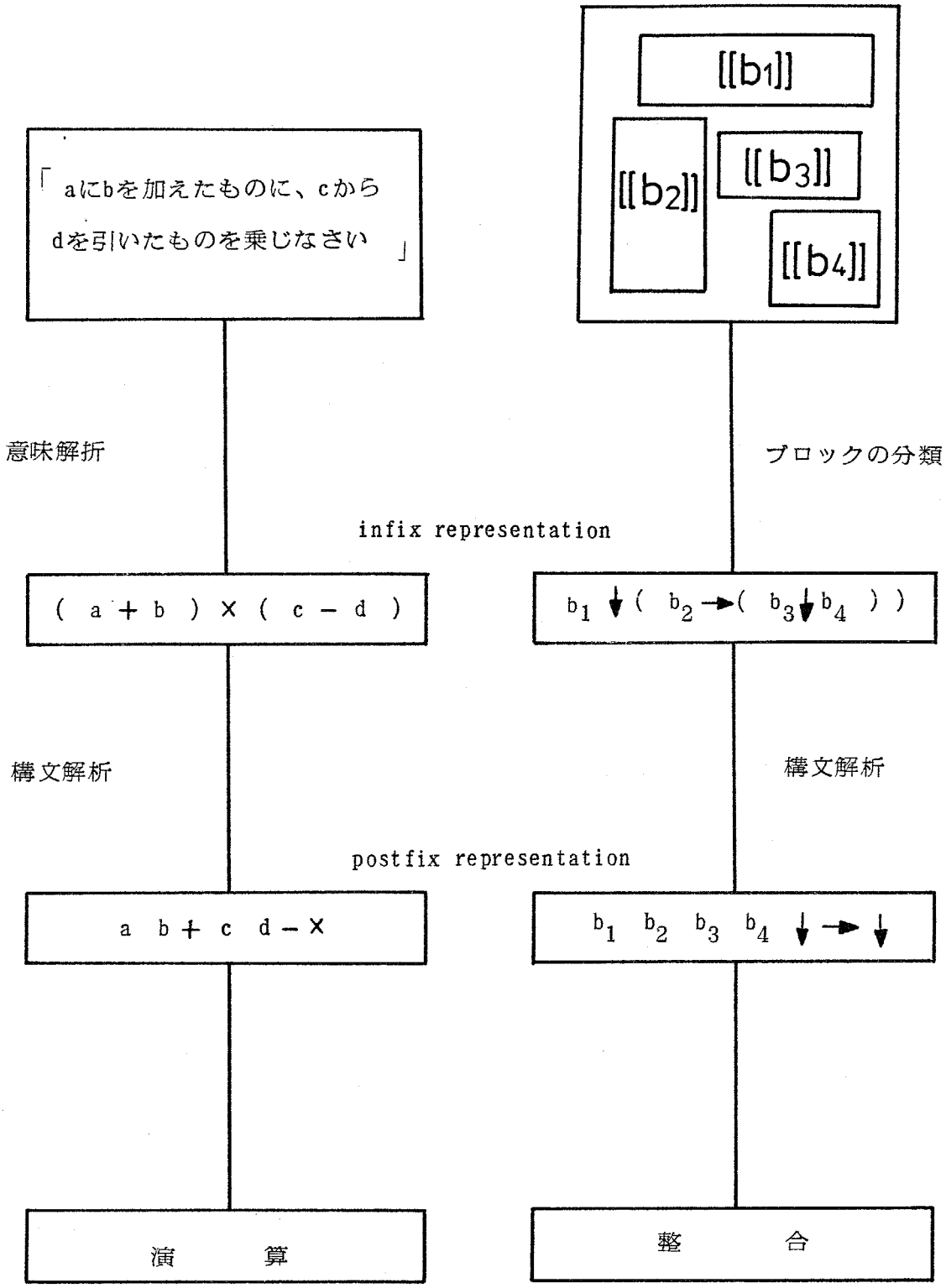


図 5・3・2　算術式と漢字パターンの表現式との関係

となる。また、このパターンのポストフィックス表現式は、

$b_1\ b_2\ b_3\ b_4$ ↓→↓

となり、各ブロックを終端記号 B_1 でおきかえた表現式は、ブロックの分類に対して得られる生成規則列によって生成される文と等しくなる。

ところで、この章で定義した文法は、ブロックの分類法と切り離して純粋な生成文法としてみた場合、曖昧な文法である。しかしながら、この章で述べた手法の目的とするところは、ブロックの分類過程を符号化することであり、ポストフィックス形式の表現式を文法によって生成することではない。

なお、曖昧でない文法については第七章において述べる。また、この章で定義した文法に対して最左導出性の条件を付加した場合の効果についても述べる。

第六章　手書き漢字パターン表現への応用

6-1　手書き漢字パターン表現の目的

前章までにおいて、漢字パターンのもつ構造的な特徴に注目したパターンの表現法と生成文法を利用した符号化法とについて述べたが、その表現法の応用の一つとして手書き漢字パターンの表現がある。

ところで、既に書かれた文字、あるいは印刷された文字を何らかの方法で表現するという場合、単なる情報の圧縮を目的とするような場合を除けば、殆どの場合、文字のパターン認識が目的となると言ってもよい。即ち、それぞれの文字の標準となる表現式と未知入力文字の表現式とを比較し、その結果から該当文字の判定を行うという作業である。

文字のパターン認識法は、第二章でも述べたように多くの方法が報告されているが、手書き文字、特に漢字に対しては現在のところ、実用化へ結び付けられるような有効な方法は考えられていない。その最も大きな要因は、認識の対象となる文字種が極めて多いことと文字自体が複雑であることである。さらに手書きによって生じる位置や大きさの変化、線と線の不要な接触、あるいは、連結すべき点の切断等の現象が一層認識を困難なものにしている。

これまでに、手書き漢字パターンを対象とした認識方法が多く報告されている。[27],[28],[43] 印刷漢字の場合と同様に混合類似度法を応用して認識を行おうとする研究も行われているが、線の断続等の局部的な位相構造の変化に対しては比較的安定ではあるものの、文字の歪みや変形に弱いため、印刷漢字の場合ほど有効な方法とは考えられない。また、漢字の構造的な特徴に注目して漢字パターンを表現し、その表現式を利用して認識を行おうとする方法も報告されているが、線の連結関係を重視し過ぎる傾向にあり、線の断続に対処するための処理が複雑になる。[44] しかも、構造的な表現法においては、その表現が正確に、あ

るいは細密になればなる程、表現の再現性が低下してしまうという矛盾が生じてしまう。一方、漢字の周辺の形状やストロークの密度情報を利用して漢字パターンの分類識別を行おうとする方法が報告されているね。[45] この方法は、漢字の分類等には適しているね考えられるが、分類後の識別判定をどのように行うかが問題点として残されているね。

以上に述べた各種の方式にはそれぞれ長所、短所があり、一つの方式で全ての漢字の認識を行うことは非常に困難であるね考えられね。それぞれの問題点を踏まえ、前章までに述べた漢字パターンの表現法、符号化法を手書き漢字に応用し、その表現法を利用した手書き漢字パターンの認識法について検討するね。

6-2 ブロックの構成と位置関係

6-2-1 細線化処理と特徴抽出

これまでに述べてきた表現法を手書き漢字パターンの表現に応用するためには、まず、それぞれのブロックを構成しなければならない。その構成方法の如何に依って表現法の良否が決定されてしまうため、このブロックの構成が最も大事な処理過程となるね。

ブロックを構成する方法としては、細線化処理を行った後に幾何学的な特徴点を抽出し、この特徴点をもとにブロックを構成する方法や、細線化処理を行わずに、直接、ストロークを抽出してブロックを構成する方法等が考えられね。細線化処理はすべての文字に統一的に適用することができ、後の処理も容易になるなどの利点があるが、ストロークが消滅したり、あるいは逆に、ヒゲと呼ばれる不要なストロークが生じたりする欠点があるね。一方、細線化を行わないでストロークを抽出する方法もいくつか報告されているが、文字が複雑であったり、線幅が太くなった場合に希望通りのストロークが抽出されないことがあ

る。本論文では、統一的な処理が可能であることや、後の処理の容易さを考慮し、細線化処理[46]、幾何学的特徴点抽出という処理方法を採った。

文字パターンは白黒の2値画像で、白地を0、黒地を1とする。細線化処理はヒルディッチの4連結と8連結の混合方式を用いて行った。図6.2.1にその例を示す。同図は64×64のドットパターンをCRTにディスプレイし、写真撮影を行ったものである。また、特徴点の抽出は次のように行う。

まず、文字パターンの左上端から横方向にスキャンし、最初に検出した'1'の点を注目点として、図6.2.2に示す8近傍の点に対して、

$$E(r)=\sum_{l=0}^{7}\left(g(l)-g(l)\cdot g(l+1)\right) \qquad (6\text{-}2\text{-}1)$$

但し、l : mod 7

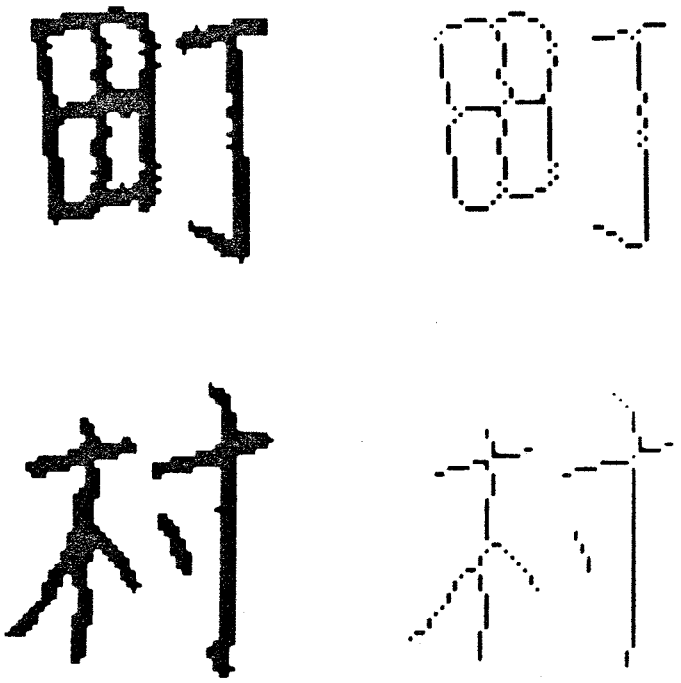
の値を求める。

もし、$E(r)$の値が1, 3, 4の場合、その注目点を特徴点として抽出する。即ち、その場合の注目点は端点か分岐点、あるいは交点のいずれかの特徴点になっている。また、$E(r)$の値が2のときは、図6.2.2に示した8近傍マスクによって次の追跡点に注目点を移し、式(6-2-1)の$E(r)$の値を計算する。もしも$E(r)=2$となる点が$2p+1$個以上続いた場合は、図6.2.3に示すように、注目点と、その注目点のp個前の追跡点、さらにp個前の追跡点とをそれぞれ結んで得られる2本の直線のなす角度αを求める。角度αが、

$$\alpha > \theta \qquad \text{但し、}\theta\text{はいき値} \qquad (6\text{-}2\text{-}2)$$

のとき、注目点からp個前の追跡点をノード(node)2の特徴点、即ち、屈曲点として抽出する。

以上に述べた特徴点抽出アルゴリズムを示したものが、図6.2.4である。特徴点抽出と同時に、各特徴点を連結する線素の方向も検出する。なお、特徴点抽出の為のパラメータ、θ, pの値としては、θが45度、また連結数pの値は、32×32のパターンに対しては3程度、64×64のパターンに対しては5から6程度が適当である。



原パターン　　細線化パターン

図 6・2・1　細線化処理の例

g(3) (i−1, j−1)	g(2) (i−1, j)	g(1) (i−1, j+1)
g(4) (i, j−1)	r (i, j)	g(0) (i, j+1)
g(5) (i+1, j−1)	g(6) (i+1, j)	g(7) (i+1, j+1)

r ： 注目点

図 6・2・2　注目点とその8近傍

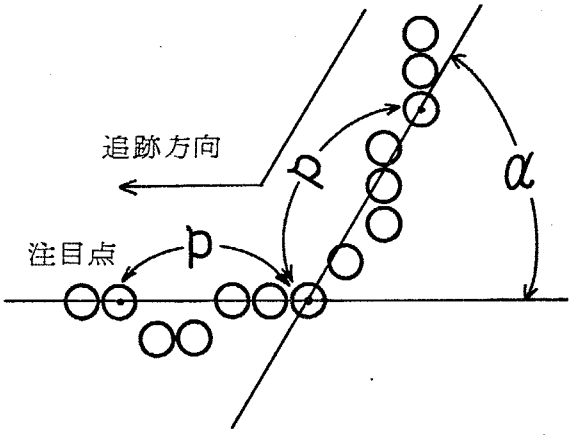


図 6・2・3　角度の測定法

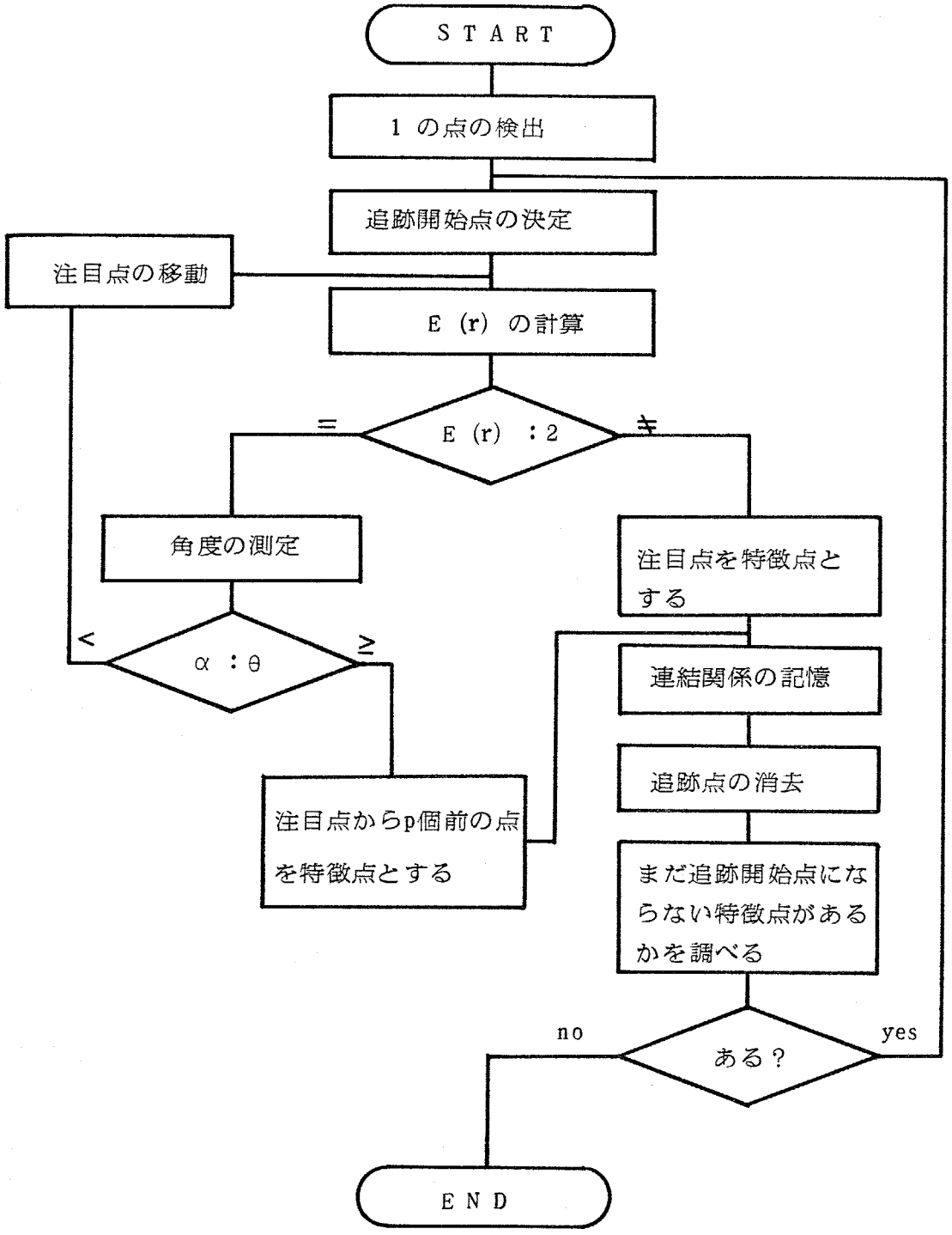


図 6・2・4 特徴点抽出法の概略図

6-2-2 ブロックの構成

抽出された特徴点において、隣り合う2つの特徴点を連結する区間を線素とし、その線素は図6.2.5に示す4種類のいずれかに決定する。方向が逆の線素、を定義した場合、線素の種類を一意に決定することができなくなるため、図6.2.5の4種類とした。この4種類の線素はストロークの方向角を8方向で量子化した場合の各方向の内で、ストロークの出る方向の多いものを選んだものである。

ところで、細線化処理を施すことによって、本来交点であるべき点が図6.2.6のように2つの分岐点として得られることがしばしば生じる。このような場合、2つの分岐点を1個の交点に修正する。今、ある線素 l_i の始点、終点の座標を (x_{is}, y_{is}), (x_{iE}, y_{iE}) とするとき、

始点、終点がいずれも分岐点で、しかも

$$\max\{|x_{is}-x_{iE}|, |y_{is}-y_{iE}|\} \leq d_0$$

であるとき、線素 l_i を消去し、l_i の終点に連結する他の線素を l_i の始点に連結する。但し、d_0 は32×32のパターンでは、文字線幅によって多少の変化はあるものの2から3程度、また、64×64のパターンでは5から6程度が適当である。

特徴点抽出処理によって線素が決定される。この線素を、第三章で定義した連結関係 $\mathbb{C}$ によって合成するが、分岐点や交点においては連結関係 $\mathbb{C}$ の条件を満足する線素を一意に決定できない場合が生じる。そこで、それらの点における線素の合成は、

(1) 同一種類の線素の合成

(2) 合成しようとしている線素から反時計方向に検出され、しかも
連結関係 $\mathbb{C}$ の条件を満たす線素と合成

の順序で行う。同一線素が連続する場合は、それらをまとめて一つの線素とす

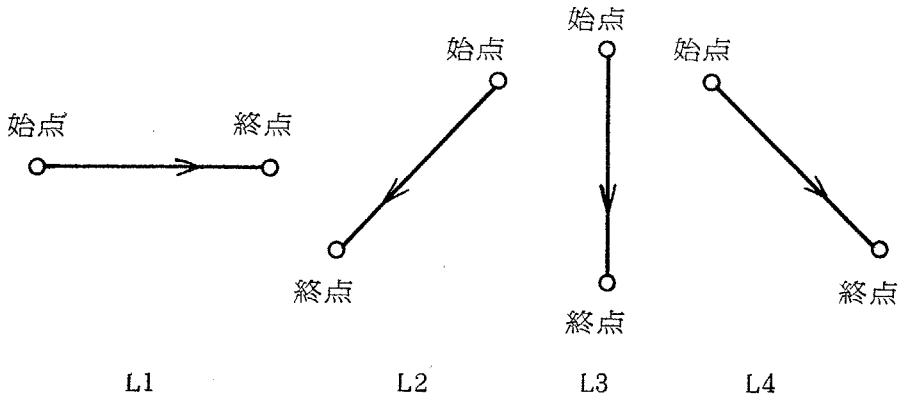


図 6・2・5　四種類の線素

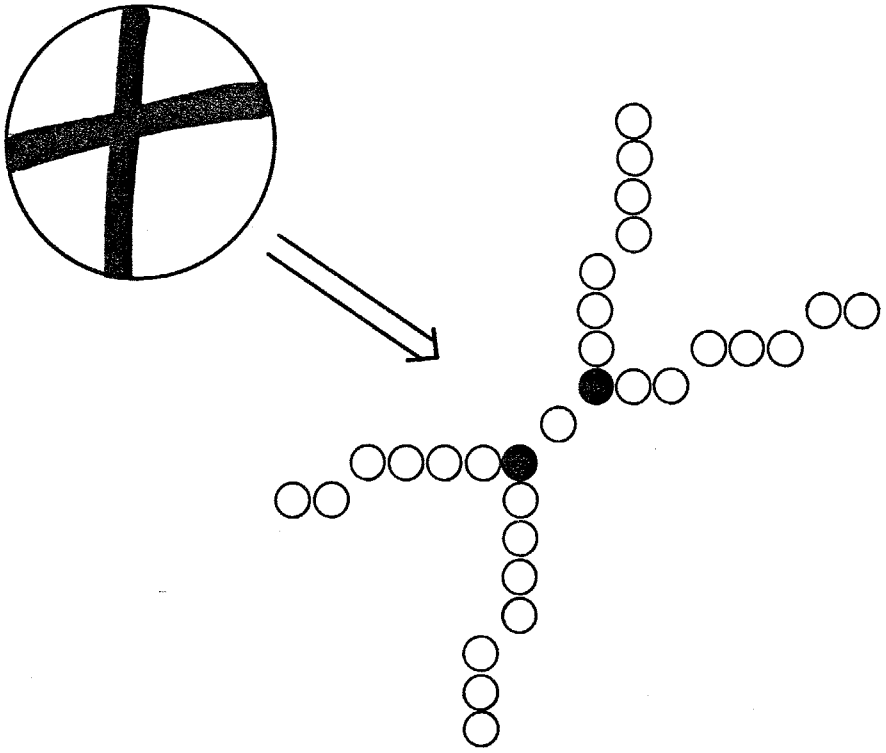


図 6・2・6　細線化処理による交点の変化

る。図6.2.7に線素合成の例を示す。同図(a)が分岐点において同一線素が連続する場合の合成例である。また、図6.2.7(b)の例は合成しようとしている線素 l_i に対して同一線素が存在しないため、(2)の合成法に従って、l_i から反時計方向に最初に検出される l_j と合成を行う場合の例である。

線素の合成によって基本ブロックが構成されるが、この基本ブロックは図6.2.8に示す26種類のうちのいずれかのブロックに決定される。この決定は、基本ブロックの線素表現を利用して行う。基本ブロックの構成後、その基本ブロックの中でクロス関係や近接関係にあるものを、クロスブロックや近接ブロックとして構成する。その構成方法は第三章で述べた方法に従うが、近接ブロックは、手書きの場合の文字の変形を考慮して第二種近接関係のみを対象とした。

図6.2.9に各基本ブロックを構成していく過程と、クロスブロック、近接ブロックを構成して各ブロックの領域を求めた例を示す。また、位置関係は包含関係, 上下関係, 左右関係の3つの位置関係として、第四章で述べた位置関係を決定するパラメータ C_1, C_2 をそれぞれ 0.5, 0.5 として実験を行った。

6-3 符号化とパターンの分類

入力未知文字Kを構成するブロック集合 $\{b\}_K$ を第四章で述べたブロックの分類法に従って分類すれば、その分類に対応した生成規則列が得られる。また、生成規則列と同時にブロックのカテゴリー列が得られる。この生成規則列とブロックのカテゴリー列を利用して符号化を行う。

漢字の構造上の特徴として、複数個の文字が組み合わさって別の文字を構成することが多いということは一般的に知られていることであるが、ブロックの分類におけるステップ2は、この特徴を効果的に利用するための分類段階であった。以後、ステップ2の分類で得られる部分集合は独立な文字として存在す

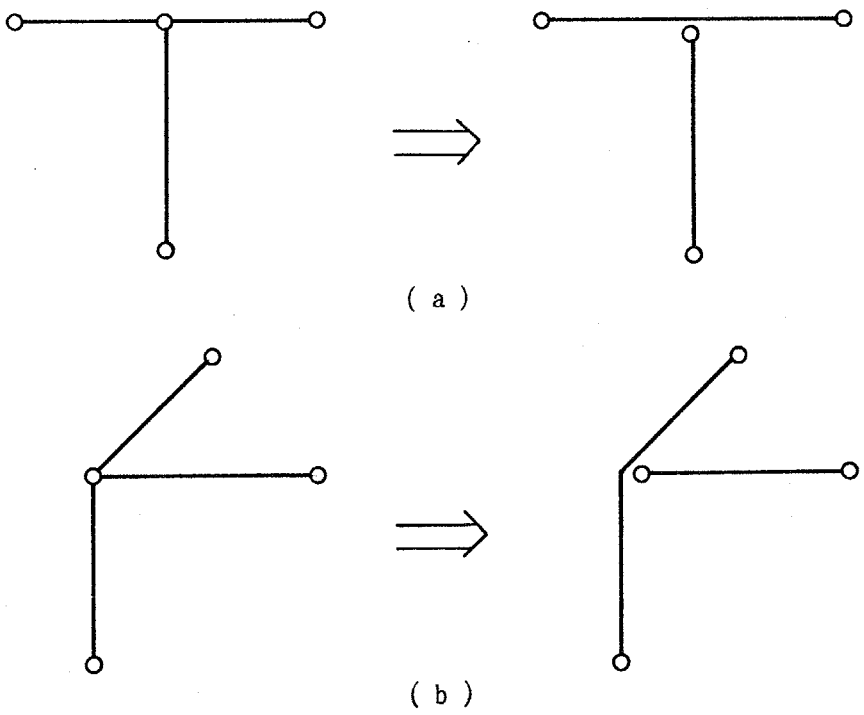


図 6・2・7 線素合成の例

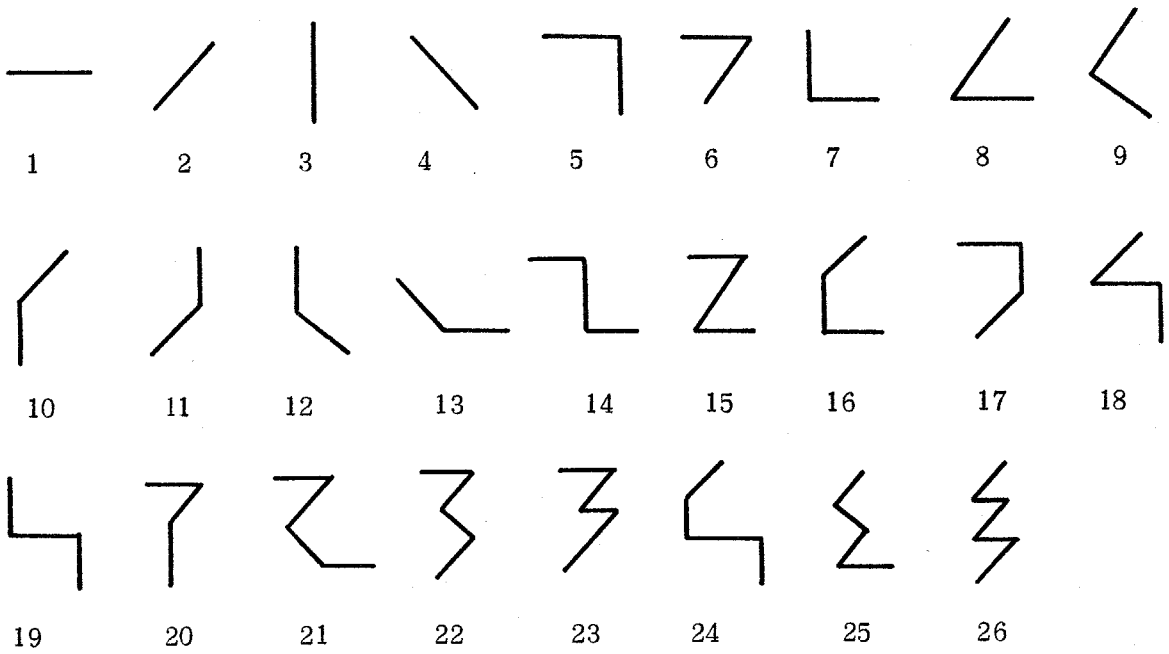
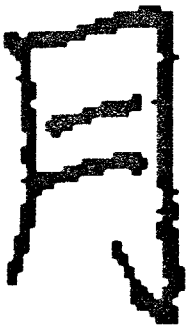
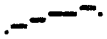


図 6・2・8 基本ブロックの例



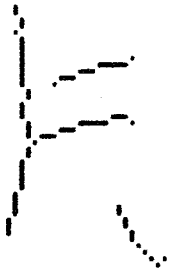
原パターン

細線化パターン



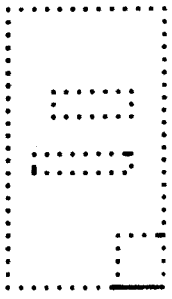
(a)

(b)



(c)

(d)



(e)

ブロックの領域

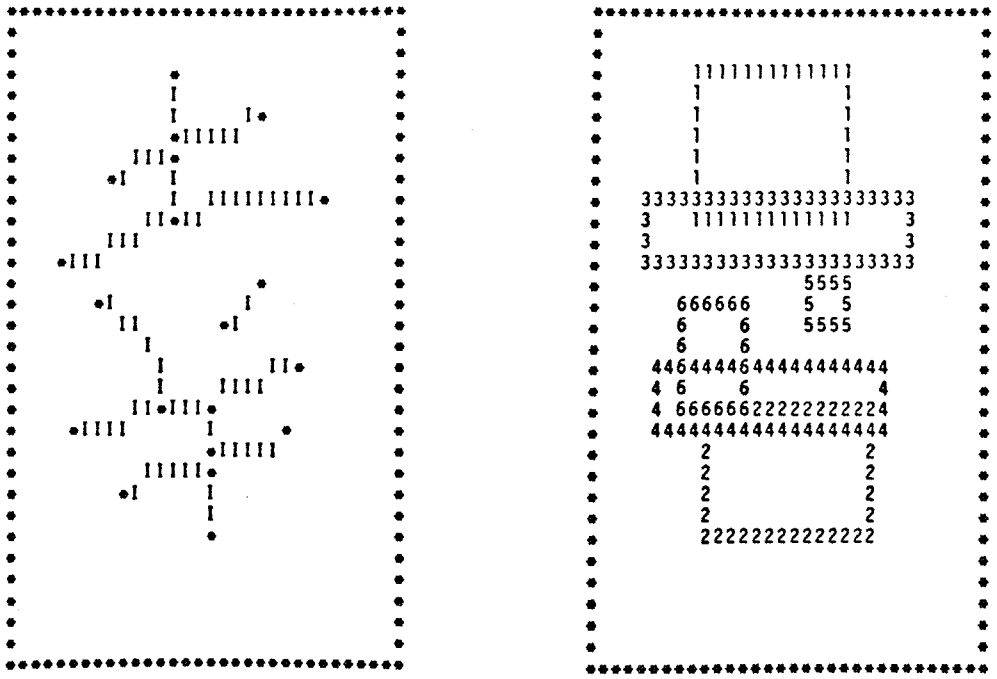
(a) ～ (e) 各ブロックの構成過程

図 6.2.9 ブロックの構成過程の例

ると仮定し、それを基本文字と呼ぶ。そして、ある漢字パターンに対して、ステップ2の分類で複数個の部分集合が得られたとき、その文字は複数個の基本文字が左右関係で組み合さって構成されているものとし、このような文字を複合文字と呼ぶ。小学校教育課程で学習する約1,000字の漢字について調べた結果、約420文字が本論文で定義した複合文字で、基本文字は約670文字存在した。一方、漢字パターンに含まれるブロック数については、前述の約1,000字を調べた結果、最大17個であった。このことから、前章で定義した文法GにおけるMの値を以後17として扱う。

符号化は、入力パターンに対して得られる各生成規則の番号列とブロックのカテゴリー列によって行う。なお、各ブロックの数は、基本ブロックが図6.2.8に示した26種類、近接ブロックが17種類、クロスブロックが56種類である。各ブロックのカテゴリー番号として、基本ブロックを1から26、近接ブロックを101から117、クロスブロックを201から256と決定して各ブロックの種類の区別を行った。

符号化例を図6.3.1から図6.3.8までに示す。図6.3.1から図6.3.6までの例が32×32のパターンに対して符号化を行った例で、各図(a)が細線化パターンから特徴点を抽出した例で、＊印が抽出された特徴点を表わしている。また、各図(b)がブロックを構成してその領域を求めた結果で、各数字で囲まれた部分がそれぞれのブロックの領域を表わし、その数字がブロック番号を示している。この領域の配置例をもとに生成規則番号列とブロックのカテゴリー列を求めた結果を各図の下に示してある。ブロックのカテゴリーの下の括弧の中の数字はブロック番号を表わす数字で、領域を示した数字のブロック番号に対応している。図6.3.7から図6.3.8までが64×64のパターンに対して符号化を行った例であるが、これらの図はCRTディスプレイにドットパターンとして表示を行ったものである。従って、特徴点やブロック番号を示す数字は表示されていない。各ブロックの領域は点線で表示を行っている。

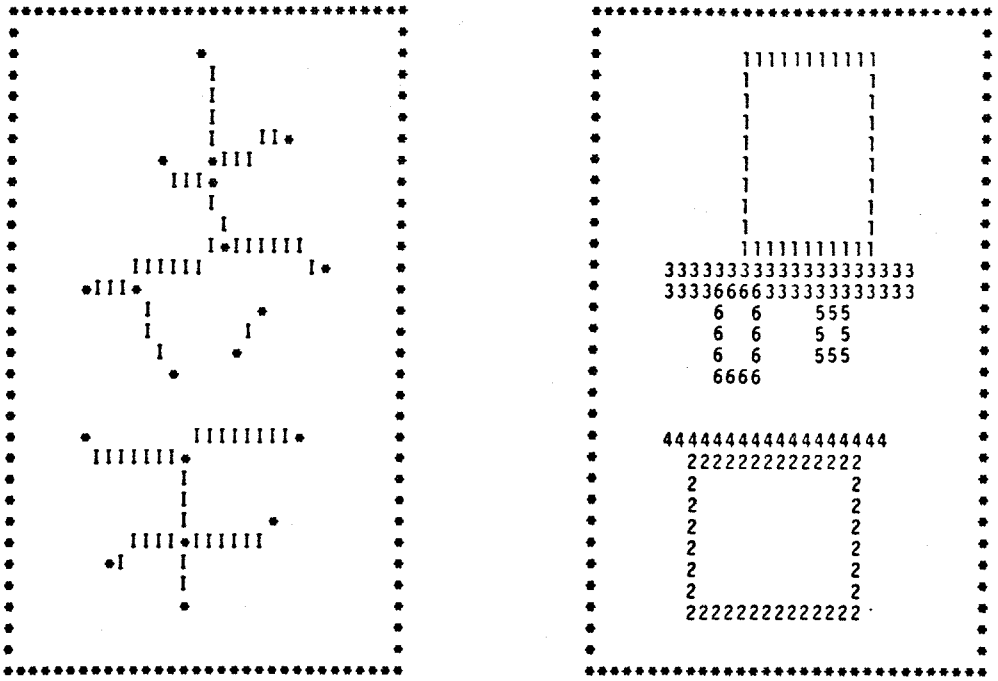


生成規則番号列　(6,51,40,29,19,18)

ブロックのカテゴリー列　(201 , 1 , 4 , 2 , 1 , 201)

(b_1 , b_3 , b_6 , b_5 , b_4 , b_2)

図 6・3・1 '幸' パターンとその符号化例 (1)

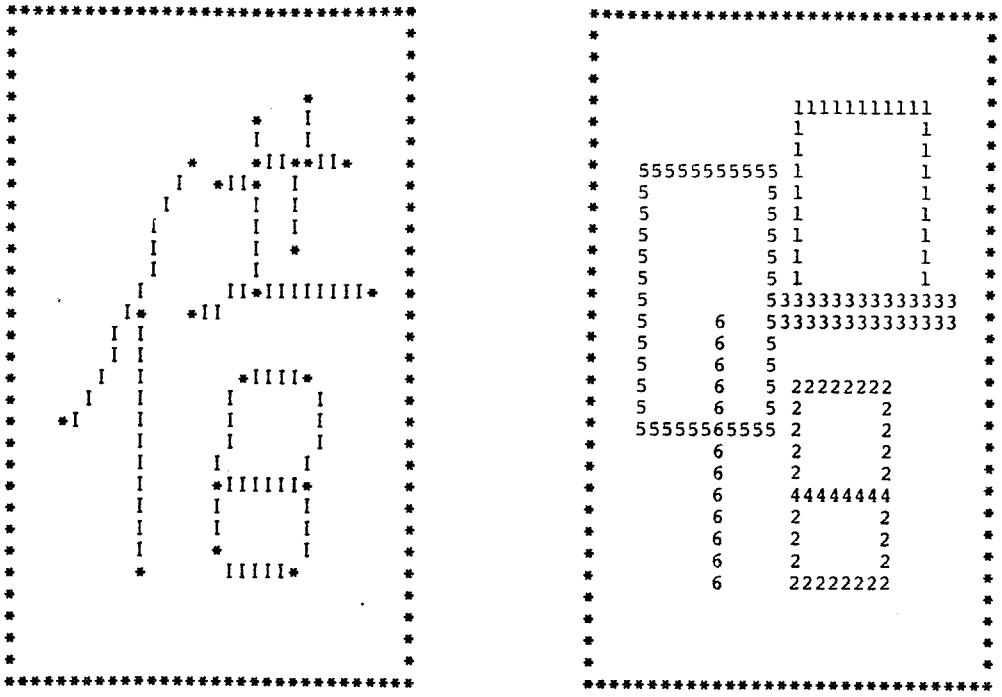


生成規則番号列　(6,51,40,29,19,18)

ブロックのカテゴリー列　(201 , 1 , 4 , 2 , 1 , 201)

(b_1 , b_3 , b_6 , b_5 , b_4 , b_2)

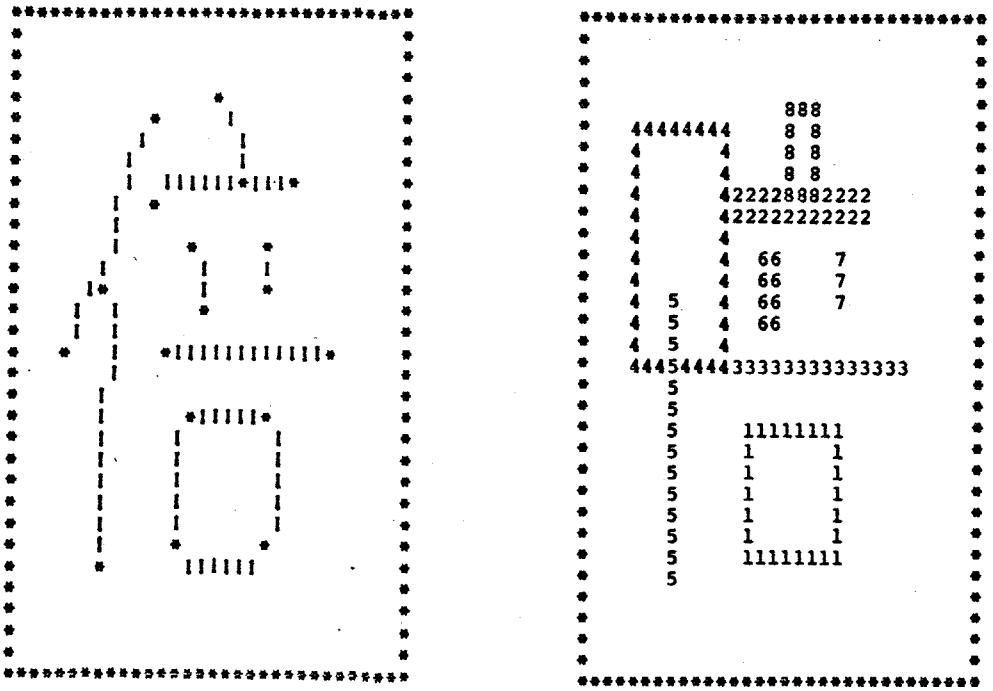
図 6・3・2 '幸' パターンとその符号化例 (2)



生成規則番号列　(6,48) , (2,19) , (4,31,24,20)

ブロックのカテゴリー列　(2　, 3) , (216 , 1 , 105 ,1)
(b_5 ,b_6) , (b_1 ,b_3 ,b_2 ,b_4)

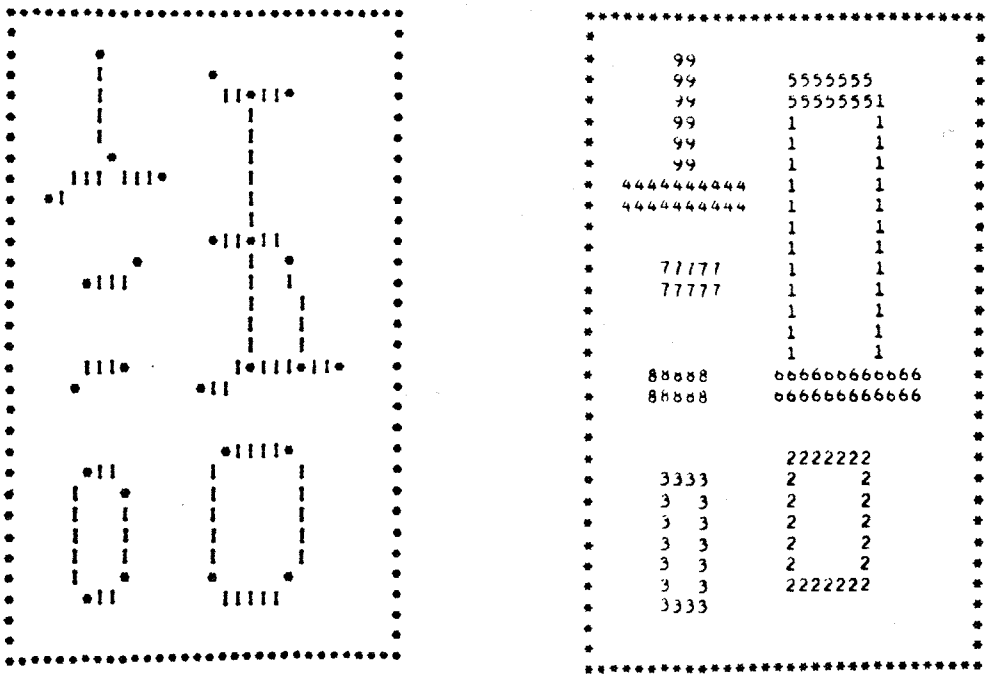
図 6・3。3 '借' パターンとその符号化例



生成規則番号列　(8,76) , (2,19) , (6,51,40,29,19,18)

ブロックのカテゴリー列　(2 , 3) , (3 , 1 , 3 , 3 , 1 , 105)
(b_4 ,b_5) , (b_8 ,b_2 ,b_6 ,b_7 ,b_3 ,b_1)

図 6・3・4 '倍' パターンとその符号化例

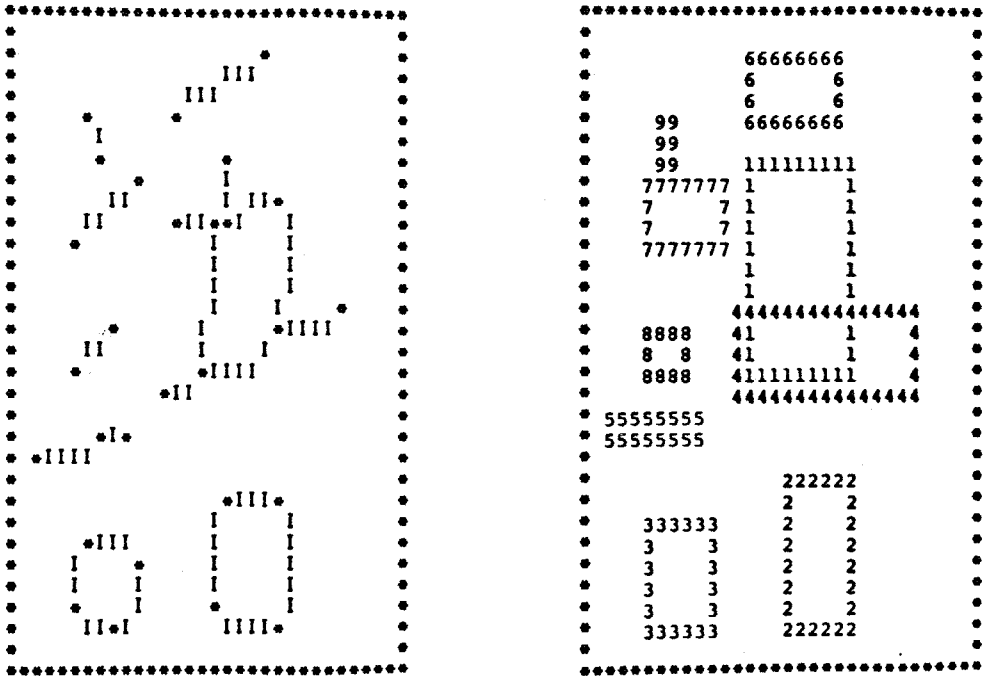


生成規則番号列　(9,87)，(5,40,31,24,19)，(4,31,24,19)

ブロックのカテゴリー列　(3，1，1，1，105)，(1，206，1，105)

(b_9, b_4, b_7, b_8, b_3)，(b_5, b_1, b_6, b_2)

図 6.3.5 '語' パターンとその符号化例 (1)

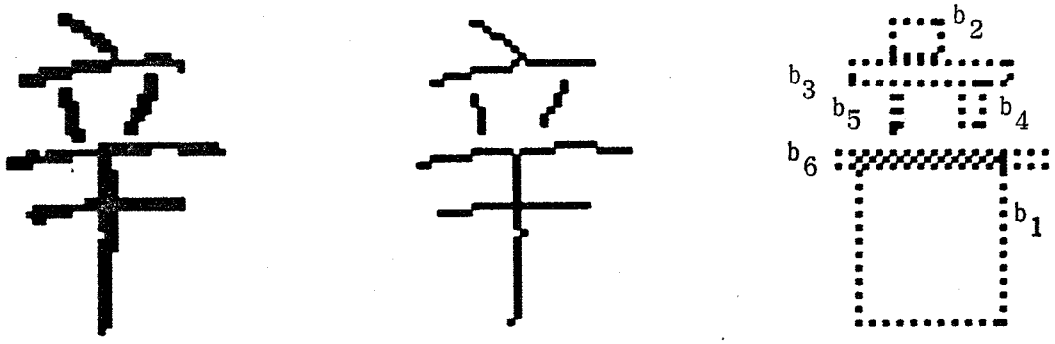


生成規則番号列　(9,87)，(5,40,31,24,19)，(4,31,24,19)

ブロックのカテゴリー列　(4，2，2，1，110)，(2，206，1，105)

(b_9, b_7, b_8, b_5, b_3)，(b_6, b_1, b_4, b_2)

図 6.3.6 '語' パターンとその符号化例 (2)

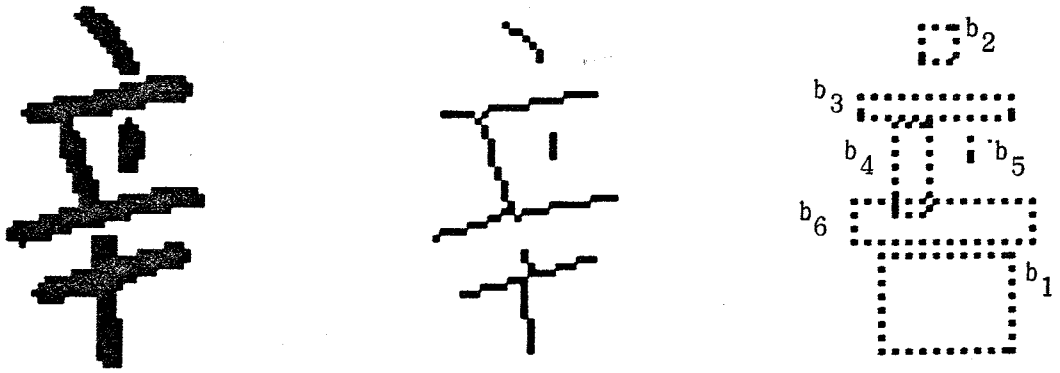


生成規則番号列　(6,51,40,29,19,18)

ブロックのカテゴリー列　(4 , 1 , 3 , 2 , 1 , 201)

(b_2 ,b_3 ,b_5 ,b_4 ,b_6 ,b_1)

図 6.3.7 '辛' パターンの符号化例 (1)



生成規則番号列　(6,51,40,29,19,18)

ブロックのカテゴリー列　(4 ,, 1 , 4 , 3 , 1 , 201)

(b_2 ,b_3 ,b_4 ,b_5 ,b_6 ,b_1)

図 6.3.8 '辛' パターンの符号化例 (2)

符号化された基本文字パターンの集合を基本辞書、複合文字パターンの集合を複合辞書と呼ぶ。各辞書とも能率よく参照するためには、辞書内の文字パターンの分類を行う必要がある。その分類に利用する特徴は、文字の変形に対し安定でなければならない。図6.3.1から図6.3.8までに示した例からもわかるように、位置関係によるブロックの分類過程を符号化した生成規則番号列は、多少の文字変形に対し安定な値となっている。そこで、文字パターンの分類に生成規則番号列を利用する。すなわち、各辞書内の文字パターンの生成規則番号列を同一ブロック数毎に木構造表現することによって文字パターンの分類を行う。例えば、基本辞書におけるブロック数3のパターンの生成規則列を木構造表現すると、図6.3.9のように表わすことができる。同図は、木構造表現の第一段階T_1で3個のカテゴリーに、続く第二段階T_2で9個のカテゴリーに分類されることを示している。又、表6.3.1は基本辞書に含まれる文字パターンの各ブロック数毎のパターン数と、生成規則番号列を木構造表現したときの各段階における文字パターンのカテゴリー数を示したものである。同表において、括弧で囲まれた数字が各ブロック数毎のパターン数を、また、空白部分は次の段階以降カテゴリー数の増加がみられなかったところを表わしている。この分類法は複合文字についても同様に適用することができる。

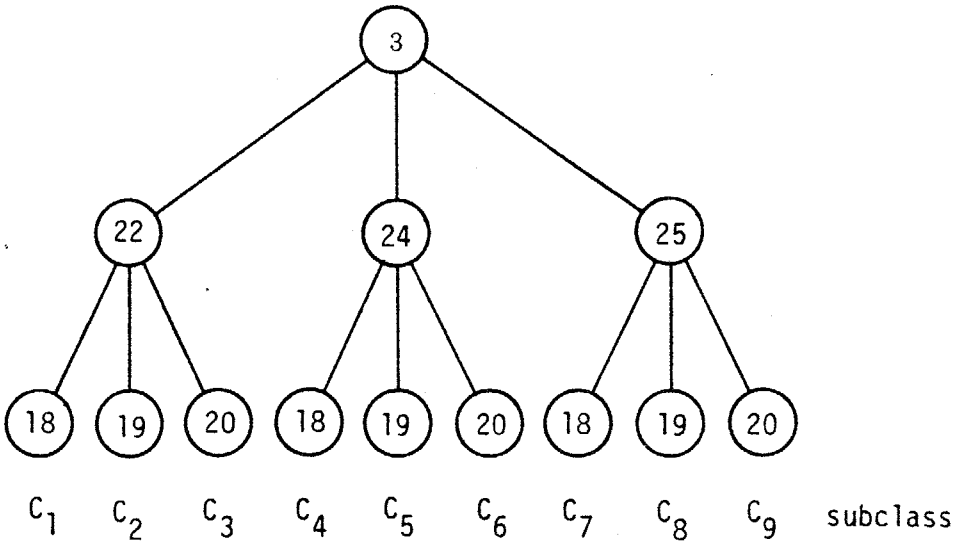


図 6・3・9　ブロック数3のパターンの生成規則番号による分類

表 6・3・1　基本辞書内のブロック数毎のカテゴリー数

ブロック＼段階	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8
1 (26)								
2 (60)	2							
3 (125)	3	9						
4 (106)	5	15	27					
5 (87)	6	19	38	50				
6 (82)	6	18	34	45	51			
7 (81)	6	20	40	55	61	63		
8 (43)	7	18	25	34	38	40	41	
9 (27)	7	13	20	22	22	23	23	24
10 (13)	5	11	12	12	13			
11 (17)	6	13	15	17				
12 (2)	2							
13 (2)	2							

6－4　本符号化法によるパターン識別の検討

これまで述べた手書き漢字パターンの符号化法によるパターン識別の方法について検討する。

本符号化法の特徴を考慮した場合、未知入力パターンの分類、識別を行うためには、入力パターンに対して得られた生成規則番号列によって各辞書内での分類を行い、ブロックのカテゴリー列によって、分類されたある文字カテゴリー内の標準パターンとの類似性を比較するのが適当であると考えられる。

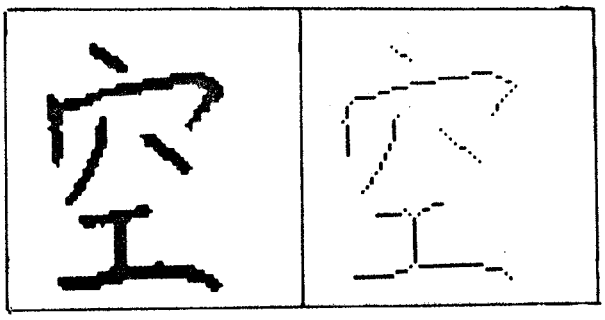
ところで、本論文で定義したブロックには、クロスブロック、近接ブロックそして、そのいずれのブロックをも構成しない基本ブロックの3種類が存在するが、基本ブロックよりも近接ブロックやクロスブロックによって漢字パターンが特徴づけられていると考えられる。従って、このような特徴を生かすことのできる判定法でなければならない。類似性の比較を行う方法は、第二章でも述べたように各種の方法が考えられるが、論理関数的判定法は上述の特徴を有効的に活用できないばかりでなく、手書き文字の変形に対して微力であるため適当ではない。むしろ、2つのパターン間の距離を測定する方法や類似度的手法が適していると考えられる。そこで、次のような、ある重み付きの距離関数を導入してパターン間の類似性の比較、検討を行った。

$$D(A_i,X)=\sum_{j=1}^{n}\frac{\sqrt{a_j\times x_j}}{\sqrt{\sum_{k=1}^{n}a_k^2}\times\sqrt{\sum_{k=1}^{n}x_k^2}}\times|a_j-x_j| \qquad (6\text{-}4\text{-}1)$$

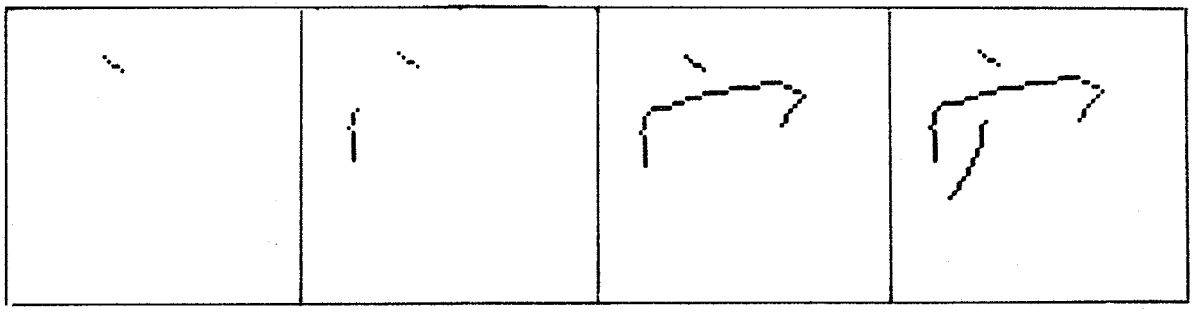
但し、A_i, Xはそれぞれ標準パターン、未知パターンを表わし、各パターンのブロックのカテゴリー列を$(a_1, a_2, \cdots, a_n)$, $(x_1, x_2, \cdots, x_n)$とする。図6.3.1, 図6.3.2, 図6.3.5, 図6.3.6に示したパターンのうちで、例1のパターンを標準パターン、例2のパターンを未知パターンと仮定し、例1のパターンに対して、例2のパターンから得られる生成規則列によって例2のパタ

ーンを分類すると、それぞれ同一の文字カテゴリーに分類される。そこで、式(6-4-1)によって2つのパターン間の距離Dを求めてみると、図6.3.2の例に対しては、D=0.000，図6.3.6の例では、2つの基本文字'言'と'吾'から構成されているが、'言'パターンに対してはD=0.046、'吾'パターンに対してはD=0.000となる。'語'の複合文字として2つの基本文字を合せた場合の距離は0.008と極めて小さな値となる。一方、図6.3.7に示した'辛'パターンと図6.3.8の'辛'パターンとの距離を測ると、D=0.000となるが、この'辛'パターンは生成規則番号列によって分類すると図6.3.1の'幸'パターンと同一の文字カテゴリーに分類される。同様に、図6.3.1の例1のパターンを標準パターンとし、図6.3.7の例1のパターンを未知文字としたときの距離Dの値は0.098となる。従って、適当な距離のいき値を設定することによってパターンの識別が可能となる。式(6-4-1)に示した距離関数やいき値については、多くのデータを処理し、その結果から最適な関数やいき値を設定しなければならない。また、距離を測定する場合、ブロックのカテゴリー番号をどのように定めるかが重要な問題となるが、本論文では実験の便宜上、これまで述べたような番号を設定したもので、必ずしも識別を行う上で最適な値に設定されている訳ではない。

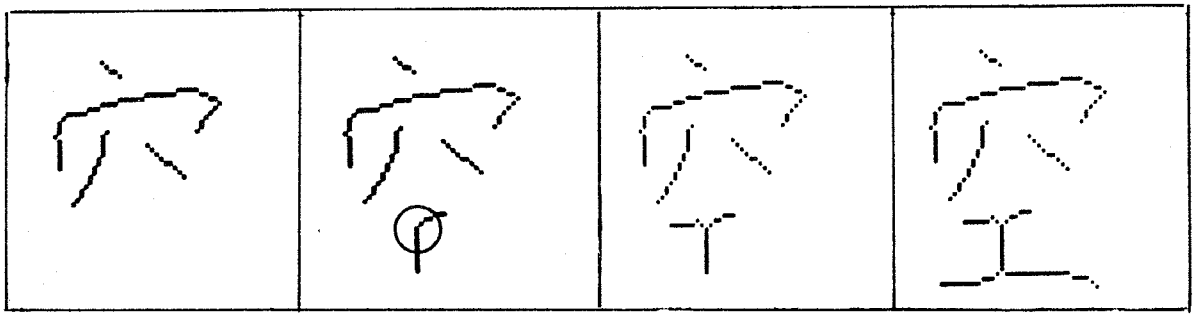
一方、図6.4.1に示した'空'パターンの例では、第六段階において、線素の傾きが原因となって、本来'空'パターンとして得られるべき表6.4.1のような結果に反し、表6.4.2に示したような結果が得られてしまったものである。この場合、生成規則列によって文字の分類を行えば、該当する文字カテゴリーが、前節で述べた基本辞書の中に存在しないためにリジェクトされてしまう。そこで、生成規則列の分類において該当する文字カテゴリーが存在する分類段階までさかのぼることができるような識別機構にしておけば、識別が可能になる。図6.4.1の例の場合、分類の第五段階までさかのぼれば候補文字が属する文字カテゴリーが存在する。その文字カテゴリーの中には、'空'の標準パターンの他にも'音'，'流'，'素'等の標準パターンも含まれているため、これらの



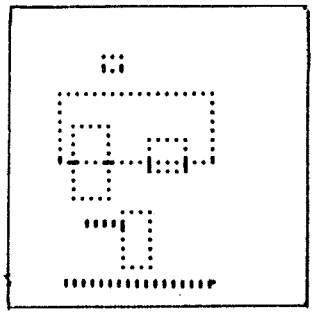
原パターン　　細線化パターン



（1）　（2）　（3）　（4）



（5）　（6）　（7）　（8）



ブロックの領域

（1）～（8）：　ブロックの構成過程

図 6.4.1　空　パターンの例

表 6・4・1　標準　空　パターンに対応する符号化の例

	生成規則	ブロックの分類過程	ブロックのカテゴリー
P_7	$S \rightarrow B_7$	$W_1 = (b_1, b_2, b_3, b_4, b_5, b_6, b_7)$	$b_1 = 108$
P_{64}	$B_7 \rightarrow B_1B_6$ ↓	$W_{11} = (b_2)$, $W_{12} = (b_1)$	$b_2 = 4$
P_{51}	$B_6 \rightarrow B_1B_5$ ↓	$W_{13} = (b_3, b_4)$, $W_{14} = (b_5)$	$b_3 = 2$
P_{36}	$B_5 \rightarrow B_2B_3$ ↓	$W_{15} = (b_6)$, $W_{16} = (bb_7)$	$b_4 = 4$
P_{24}	$B_3 \rightarrow B_1B_2$ ↓	$W_{131} = (b_3)$, $W_{132} = (b_4)$	$b_5 = 1$
P_{19}	$B_2 \rightarrow B_1B_1$ ↓		$b_6 = 3$
P_{18}	$B_2 \rightarrow B_1B_1$ →		$b_7 = 1$
生成規則番号列		7 , 64 , 51 , 36 , 24 , 19 , 18	
ブロックのカテゴリー列		b_2 , b_1 , b_3 , b_4 , b_5 , b_6 , b_7	

表 6・4・2　図 6・4・1の変形　空　パターンに対応する符号化の例

	生成規則	ブロックの分類過程	ブロックのカテゴリー
P_7	$S \rightarrow B_7$	$W_1 = (b_1, b_2, b_3, b_4, b_5, b_6, b_7)$	$b_1 = 108$
P_{64}	$B_7 \rightarrow B_1B_6$ ↓	$W_{11} = (b_2)$, $W_{12} = (b_1)$	$b_2 = 4$
P_{51}	$B_6 \rightarrow B_1B_5$ ↓	$W_{13} = (b_3, b_4)$, $W_{14} = (b_5, b_6)$	$b_3 = 2$
P_{36}	$B_5 \rightarrow B_2B_3$ ↓	$W_{15} = (b_7)$	$b_4 = 4$
P_{22}	$B_3 \rightarrow B_2B_1$ ↓	$W_{131} = (b_3)$, $W_{132} = (b_4)$	$b_5 = 10$
P_{18}	$B_2 \rightarrow B_1B_1$ →	$W_{141} = (b_5)$, $W_{142} = (b_6)$	$b_6 = 1$
P_{18}	$B_2 \rightarrow B_1B_1$ →		$b_7 = 1$
生成規則番号列		7 , 64 , 51 , 36 , 22 , 18 , 18	
ブロックのカテゴリー列		b_2 , b_1 , b_3 , b_4 , b_6 , b_5 , b_7	

パターンに対しても式(6-4-1)に示したような距離関数によって距離を求め、識別判定を行うことになる。

図6.4.1に示した場合のように、同一ブロック数内での変形に対してはある程度の融通性を持っている。しかしながら、ブロック数が異なってしまう程の変形に対しては、現在のままの方法では問題があり、手書き漢字認識へ本符号化法を応用する場合の今後の検討課題の一つである。

第七章 漢字パターンの生成モデル

7-1 漢字パターンの表現と生成について

第五章において、漢字パターンの符号化を行うために、式(4-4-7)で与えられるようなインフィックス形式による表現と、そのポストフィックス表現式との関係、また、ポストフィックス表現式と生成文法との関係について述べた。それらの各表現式、文法等において取扱ってきた位置関係としては、包含関係，上下関係、左右関係の3種類であり、第三章の式(3-4-1)，式(3-4-2)で表わされるような領域の重なり度、あるいは、各ブロックの位置的偏り等の情報の直接的な利用は行っていなかった。パターン認識のための類似性の比較を目的とする手書き漢字パターンの表現法においては、重なり度や位置的偏り具合等の文字の変形を受けやすい情報を利用しない方が、より安定な記述が可能であることを述べた。

一方、パターンの生成を目的とする漢字パターンの表現法においては[49]、包含関係，上下関係，左右関係の単なる位置関係だけでなく、各ブロックの大きさ、あるいは、各ブロック間の連結関係，重なり具合等の情報が与えられれば与えられる程、より正確で漢字らしいパターンの生成が可能である。ところで、あるパターンの生成モデルに対する評価は、そのモデルに与えられる情報の量、そのモデルによって生成される文字パターンの柔軟性、汎用性，あるいは文字らしさ等によって行うことができるが、この章の目的とするところは、生成モデルの良否の議論を行うことではなく、それ以前の、漢字パターンを生成するための基本的な考え方、その実現法、さらに漢字パターン生成モデルの意義等について論じることである。

ところで、第五章で述べたようなポストフィックス形式の表現式を、ある生成文法によって生成される文と見た場合、定義5-3で定義した文法によっては

生成することはできない。M個までのB_1を含む式(5-3-5)のような任意のポストフィックス形式の文を生成するためには、次に述べるような生成文法でなければならない。

$$G_{T2}=(V_N, V_T, P, S)$$

$$V_N=\{B_2, B_3, \cdots, B_M, C_2, C_3, \cdots, C_{M-1}, D_2, D_3, \cdots D_{M-1}, S\}$$

$$V_T=\{B_1, \rightarrow, \downarrow, *\}$$

生成規則集合Pの各要素は

$S \rightarrow B_1$	$B_3 \rightarrow C_2B_1\rightarrow$	$B_4 \rightarrow D_3B_1\downarrow$
$S \rightarrow B_2$	$B_3 \rightarrow D_2B_1\downarrow$	$C_4 \rightarrow B_1B_3\downarrow$
⋮	$C_3 \rightarrow B_1B_2\downarrow$	$C_4 \rightarrow B_1B_3*$
$S \rightarrow B_M$	$C_3 \rightarrow B_1B_2*$	$C_4 \rightarrow D_2B_2\downarrow$
$B_2 \rightarrow B_1B_1\rightarrow$	$C_3 \rightarrow D_2B_1\downarrow$	$C_4 \rightarrow D_3B_1\downarrow$
$B_2 \rightarrow B_1B_1\downarrow$	$D_3 \rightarrow B_1B_2\rightarrow$	$D_4 \rightarrow B_1B_3\rightarrow$
$B_2 \rightarrow B_1B_1*$	$D_3 \rightarrow B_1B_2*$	$D_4 \rightarrow B_1B_3*$
$C_2 \rightarrow B_1B_1\downarrow$	$D_3 \rightarrow C_2B_1\rightarrow$	$D_4 \rightarrow C_2B_2\rightarrow$
$C_2 \rightarrow B_1B_1*$	$B_4 \rightarrow B_1B_3\rightarrow$	$D_4 \rightarrow C_3B_1\rightarrow$
$D_2 \rightarrow B_1B_1\rightarrow$	$B_4 \rightarrow B_1B_3\downarrow$	$B_5 \rightarrow B_1B_4\rightarrow$
$D_2 \rightarrow B_1B_1*$	$B_4 \rightarrow B_1B_3*$	$B_5 \rightarrow B_1B_4\downarrow$
$B_3 \rightarrow B_1B_2\rightarrow$	$B_4 \rightarrow C_2B_2\rightarrow$	⋮ ⋮
$B_3 \rightarrow B_1B_2\downarrow$	$B_4 \rightarrow D_2B_2\downarrow$	$B_M \rightarrow B_{M-1}B_1\rightarrow$
$B_3 \rightarrow B_1B_2*$	$B_4 \rightarrow C_3B_1\rightarrow$	$B_M \rightarrow B_{M-1}B_1\downarrow$

が必要である。

式(7-1-2)で示した生成規則数Nは

$$N = M + \sum_{k=2}^{M}(2k-1) + 2 \times \sum_{k=2}^{M} k$$

$$= 2M^2 - 3$$

となる。式(5-2-7)で与えられる文法G_Tにおける生成規則数に比べ、Mが多く

なる程、2倍近くの生成規則数が必要となるが、M個以下のB_1を含む任意の文を一意に生成することができる。しかし、漢字の表現式に相当する文は、生成文法G_{12}によって生成される言語$L(G_{12})$のごく一部にすぎない。しかも、我々が必要とする漢字の表現式を生成するためには、何らかの情報を与えない限り、G_{12}の中の適用可能な生成規則によって文を生成し、その文が、必要とするパターンの表現式であるかを判定するという非現実的作業となる。そこで、文法G_{12}に与える情報として、適用する生成規則の順序を与えれば、必要とする表現式を一意的に得ることができる。しかしながら、適用する生成規則の順序を与えるということは、もはや生成ではなく、符号化、復号化の単なる変換作業に過ぎない。この観点に立てば、符号化、復号化の過程で形式的に利用する文法が、その符号化、復号化の一意性を保障するものであればよいと考えられる。第五章で述べた漢字パターンの符号化は、生成文法の不完全さとは別に、ブロックの分類法によって一意性が保障されていた。この章で述べる漢字パターンの生成モデルは、表現式からの漢字パターンの復号化に相当するが、その処理過程においては生成文法を利用した復号化の一意性が保障される。

ところで、現在、多くのメーカーによって数多くの漢字入出力機器が製品化されており、印刷機器として考えた場合、時間的、資源的要素などの点から、生成モデルの有用性については多少の疑問がもたれる。しかしながら、3-1節でも述べたように、人工知能的立場に立った場合には多くの興味ある内容を含んでいると考えられる。さらに、これまでのような、漢字単位のコード化においては、「扁」,「旁」,「構」等の構造的な情報を利用した検索を行うためには、ドットパターンの他にそれぞれの情報を記憶させておく必要があり、システムが大規模になる反面、その利用法が限られているという短所があるが、生成モデルを含め、本論文で述べた漢字パターンの表現法はその短所に対する何らかの解決策を与える可能性を含んでいると考えられる。

7-2 ブロックの仮領域とその生成

第三章で定義したブロックの領域は、各ブロックを包含する最小の長方形になっていた。既に書かれた文字に対しては、その文字を構成する各ブロックの大きさから領域を定めることができる。しかしながら、漢字パターンを生成しようとする場合には、何らかの方法によって領域の大きさを決定してやらなければならない。そして、その領域を最大限に使ってブロックを生成することが、これまで述べたブロックとその領域との関係に相当する。従って、文字パターンを生成するためには、そのパターンを構成する各ブロックの領域に相当する長方形を、まず決めなければならない。

第三章で定義したブロックの領域と、第五章で導入した生成文法における各記号 $B_1, B_2, \cdots, B_M$ との関係について考えてみると、ブロックの領域は、ある確定した位置と大きさを持ったものであるのに対し、各記号 B_i に関しては明確な位置や大きさは定まっておらず、ただ、ある集合の中に含まれるブロックの個数を示すものであった。しかしながら、記号 B_i に対応する集合内の i 個のブロックの領域を包含するだけの大きさを属性として持っていると考えることができる。例えば、図7.2.1に示したような実際のブロックの領域の位置関係から、記号 B_1 を使った構造表現式を導けば、

$$B_1 \downarrow ((B_1 \downarrow B_1) \rightarrow B_1) \downarrow B_1 \qquad (7\text{-}2\text{-}1)$$

という表現式が得られる。これに対し、式(7-2-1)で表わされる構造関係を保存した状態で、図7.2.2のような各記号 B_1 に対応する仮想的な領域を想定することができる。このように、構造的な関係を保存した状態で仮想的に考えた領域のことを、以後、仮領域と呼ぶ。一方、ブロックを包含する最小の長方形となる、これまでの領域のことを、仮領域と区別するために実領域と呼ぶ。

ところで、式(7-2-1)の表現式をポストフィックス形式の表現式に変換すると、次のような表現式になる。

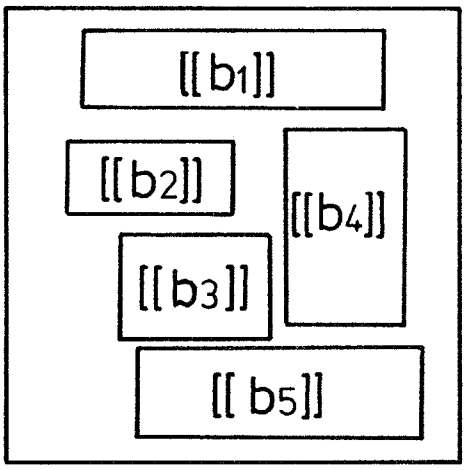


図 7・2・1　ブロックの領域の配置例

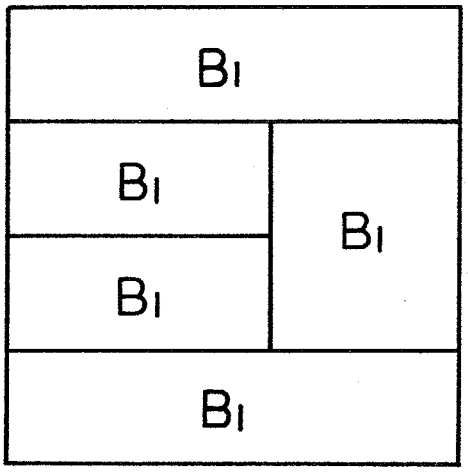


図 7・2・2　図 7・2・1のブロックの仮領域の一例

$B_1B_1B_1\downarrow B_1\blacktriangleright B_1\downarrow\downarrow$ (7-2-2)

このポストフィックス形式の表現式を、前節で述べた文脈自由文法G_{T2}によって最左導出則に従って生成するとすれば、その導出過程は式(7-2-3)の左側に示したようになり、また、各導出過程で用いた生成規則は、生成過程の表現式の右側に示したものとなる。

$S\Rightarrow B_5$	$S\rightarrow B_5$
$\Rightarrow B_1B_4\downarrow$	$B_5\rightarrow B_1B_4\downarrow$
$\Rightarrow B_1D_3B_1\downarrow\downarrow$	$B_4\rightarrow D_3B_1\downarrow$
$\Rightarrow B_1C_2B_1\blacktriangleright B_1\downarrow\downarrow$	$D_3\rightarrow C_2B_1\blacktriangleright$
$\Rightarrow B_1B_1B_1\downarrow B_1\blacktriangleright B_1\downarrow\downarrow$	$C_2\rightarrow B_1B_1\downarrow$

(7-2-3)

式(7-2-3)の導出は、最左導出を行うための生成規則列が情報として与えられて初めて一意的に行われる。この生成過程において、入力された各生成規則の中のどの変数を、次に入力されてくる生成規則で置き換えるかという情報を生成システム側で得ることができるならば、式(7-2-3)で示した各生成規則の中のD_3やC_2の変数をB_3やB_2で置き換えても、式(7-2-3)の表現式を一意に生成することができる。従って、このような生成システムにおいては、前節で述べた生成文法中のD_iやC_iで始まる生成規則は不必要となり、定義5-3の文法G_Tで十分となる。

一方、図7.2.2に示した仮領域は、図7.2.1の実領域を含む適当な大きさにとったものであるが、一度に各ブロックの仮領域を決定しようとすれば、すべてのブロックを考慮して決定していかなければならず、機械によって仮領域を決定する場合には、ブロック数が多くなるに従い困難な作業となる。ところで、式(7-2-3)に示した文生成過程は、1つの変数を2つの変数で置き換えていく操作であるが、この操作と仮領域の決定法とを対応づけて考えると、ある1つの仮領域を分割して2つの仮領域を生成するという操作に相当する。このことから、文生成過程と同様に仮領域生成過程というものが考えられ、ブロックの個数に依らない仮領域の決定法が存在する。しかしながら、式(7-2-3)のよ

うな文生成過程とは異なり．仮領域の決定は単なる記号の置き換えではなく．領域の分割を実際に行う必要がある。さらに．その分割は．必要とする仮領域を得ることのできるようなものでなければならない。そこで．仮領域の分割に関して次に示すような定義を行う。

［定義7-1］

仮領域をωで表わし．２つの仮領域間の位置関係を示す記号→，↓，＊をΔで表わす。このとき．仮領域に関する２つの作用素Ψ，Φを次のように定義する。

$$\Psi\cdot\omega_0\ ;\ \Omega_0 \rightarrow \omega_0 \qquad (7\text{-}2\text{-}4)$$

$$\Phi(a_1,a_2,a_3,\Delta)\cdot\omega_x\ ;\ \omega_x \rightarrow \omega_y\,\omega_z\,\Delta \qquad (7\text{-}2\text{-}5)$$

但し．

ω_0は初期仮領域を表わし．Ω_0はその大きさを表わす。また．式(7-2-5)のa_1，a_2，a_3は正整数である。そして．Δが→，↓のとき．分割方向の長さに関し．ω_xをa_1としたとき．ω_y,ω_zの比がa_2，a_3．ω_yとω_zの重複比が$(a_2+a_3-a_1)$となる分割である。一方．Δが＊であるとき，ω_y,ω_zは．ω_xの(a_2/a_1)倍．$(a_2/a_1)\cdot(a_3/a_1)$倍に縮小される分割である。

定義7-1の式(7-2-5)におけるa_1，a_2，a_3の各値は独立に決定できるものではなく．Δが→，↓のときは．

$$a_1 \le a_2+a_3\ ,\ a_2 \le a_1,\ \ a_3 \le a_1 \qquad (7\text{-}2\text{-}6)$$

でなければならない。また．Δが＊のときは

$$a_1 \ge a_2\ ,\ a_1 \ge a_3 \qquad (7\text{-}2\text{-}7)$$

でなければならない。例えば．図7.2.3に示したような．ある大きさの仮領域ω_xに対して．$\Phi(5,2,3,\rightarrow)$．$\Phi(4,2,3,\downarrow)$．$\Phi(10,8,5,*)$のような作用素を作用させたときのω_y，ω_zは．それぞれ図7.2.4．図7.2.5．

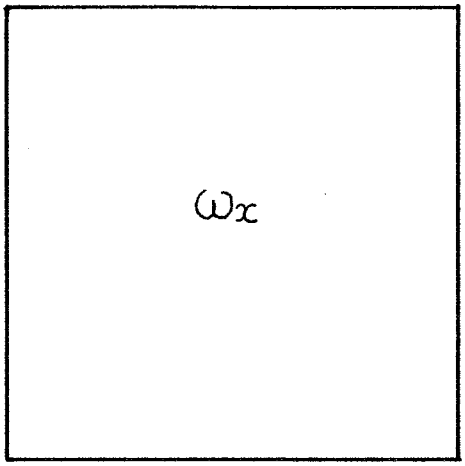


図 7・2・3

仮領域ω_xの例

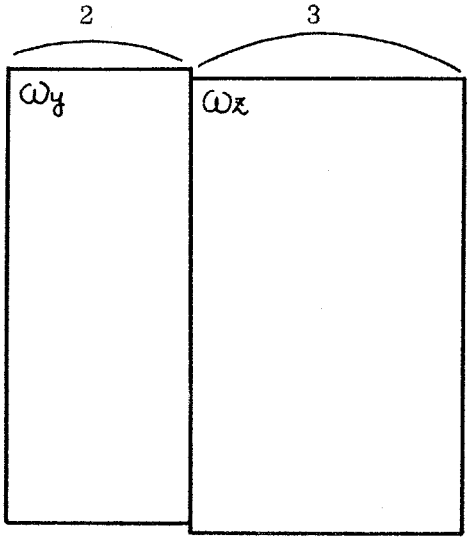


図 7・2・4

Φ(5 , 2 , 3 , →) によるω_xの分割後の仮領域ω_y , ω_zの例

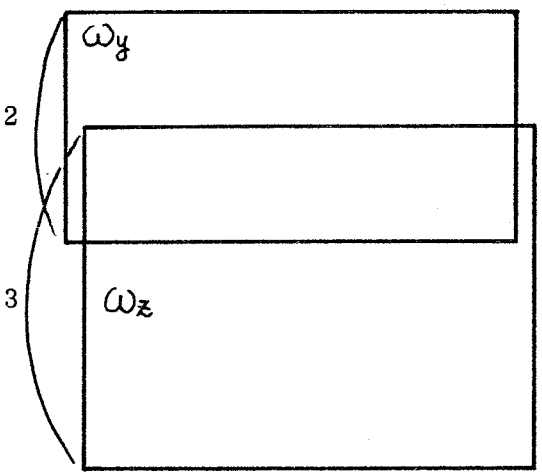


図 7・2・5

Φ(4 , 2 , 3 , ↓) によるω_xの分割後の仮領域ω_y , ω_zの例

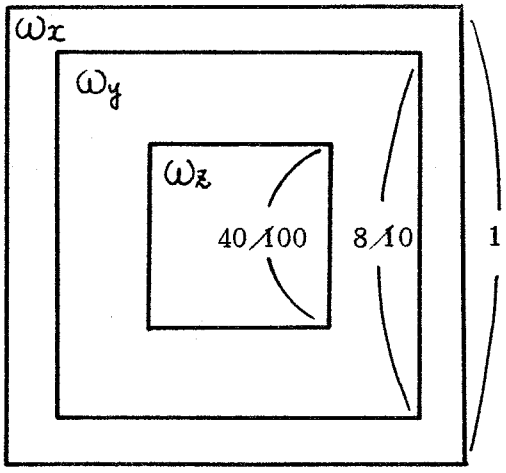


図 7・2・6

Φ(10 , 8 , 5 , *) によるω_xの分割後の仮領域ω_y , ω_zの例

図7.2.6のようになる。

ところで、式(7-2-6)の条件において、a_2をa_1に等しくとった場合について考えてみる。例えば、図7.2.3のω_xに対して、

$\Phi(5, 5, 2, \rightarrow)$

$\Phi(5, 5, 3, \downarrow)$

の作用素を作用させると、図7.2.7, 図7.2.8のようなω_y, ω_zが得られる。このような位置的関係は、第四章で述べた位置関係で言えばいずれも包含関係になってしまうが、この章では、作用素Φで与えられる位置関係とする。

定義7-1の仮領域分割演算子によって得られる仮領域の集合を$\Omega = \{\omega_0, \omega_1, \omega_2, \cdots\}$とする。

〔定義7-2〕

定義5-2で導入した生成文法Grと仮領域集合Ω上で、次のような作用素fを定義する。

$P_i ; S \rightarrow B_i \in P$, $i \leq M$に対し、

$$f(\Psi, P_i) ; S \rightarrow B_i(\omega_0) \qquad (7\text{-}2\text{-}8)$$

$P_i ; B_p \rightarrow B_q B_r \Delta \in P$, $i > M$に対し、

$$f(\Phi(a_1, a_2, a_3, \Delta), P_i) ; B_p(\omega_x) \rightarrow B_q(\omega_y) B_r(\omega_z)\Delta \qquad (7\text{-}2\text{-}9)$$

但し、式(7-2-9)における作用素fは、Φの位置関係Δを生成規則P_iの位置関係と同一のものに決定し、その位置関係に従ってω_y, ω_zの分割を行わせる作用素である。

以上の定義より、各文字パターンの構造表現式を与える生成規則列の各記号B_iに対応する固有な仮領域を決定することができる。

定義7-2における$f(\Psi, P_i)$, $f(\Phi(a_1, a_2, a_3, \Delta), P_i)$を、それぞれ、$(\Omega_0, P_i)$, (a_1, a_2, a_3, P_i)と記す。

定義7-1, 定義7-2と文法Grによって、式(7-2-2)のような形式で、しかも

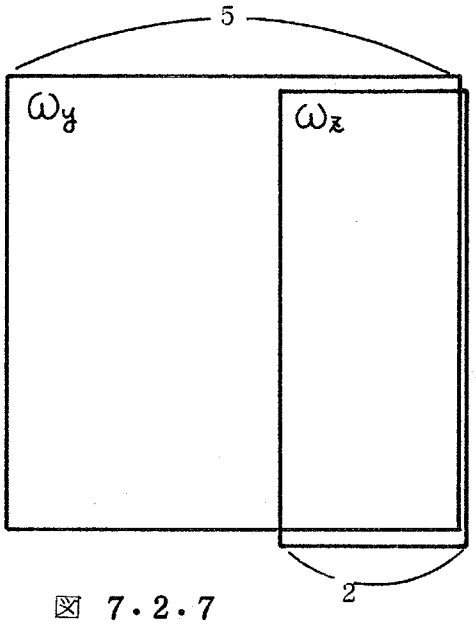


図 7・2・7

Φ(5 , 5 , 2 , →) の結果

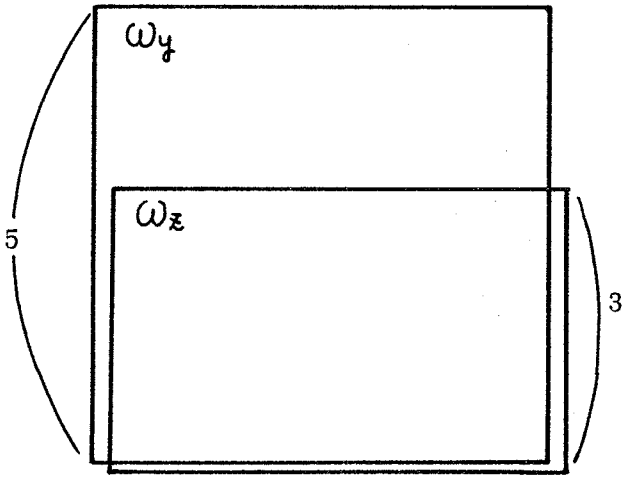


図 7・2・8

Φ(5 , 5 , 3 , ↓) の結果

各記号B_iに対して仮領域が定義されている任意の表現式を生成することができるが、ある漢字パターンの表現式を生成するためには、その表現式を生成するための生成規則を与えてやらなければならない。そこで、その生成規則の与え方に関して次のような規則を定める。

〔規則7-1〕

ある漢字パターンの構造表現式を生成するために文法Gに与える生成規則は、その導出が最左導出になるような順序の生成規則列とする。

定義7-1、定義7-2により、ある漢字パターンの構造表現式の生成過程が、そのパターンを構成するブロックの仮領域の生成過程となっている。従って、文法Gは漢字パターンの仮領域生成システムと見なすことができる。例えば、図7.2.2の仮領域を生成する過程と生成規則を示したものが図7.2.9である。但し、実線で囲まれた部分が生成規則に対応した分割領域で、点線部分が終端記号B_iに対応する仮領域である。図7.2.9の例からも分るように、たとえ導出が最左導出であっても、一般的に終端記号B_iの仮領域の生成が最左導出、即ち上から下、左から右という順序で行われるとは限らない。ブロックの生成を、そのブロックに対応する終端記号B_iの仮領域の生成と対応づけて考えるためには、仮領域の生成も最左導出になっている方が望ましい。そこで、構造表現式の左側に位置する終端記号B_iに対応するブロックから生成するような生成システムを構成する。このような生成システムは2つのスタックを利用することによって容易に構成することができる。即ち、仮領域の最左導出を行うためのスタックとブロックの最左導出を行うためのスタックを必要とする。この生成システムの概要を示したのが図7.2.10である。同図において、Nが仮領域の生成のためのスタック、Aがブロックの生成のためのスタックである。また、b_iは構造表現式の中の左側から数えてi番目の終端記号B_iに対応する実際のブロックを表わしている。一方、同図中の$\omega \Leftarrow b_n$という表現は、ωという領域の中にブロックb_nを生成することを表わすが、ブロックの生成については次節で述べる。

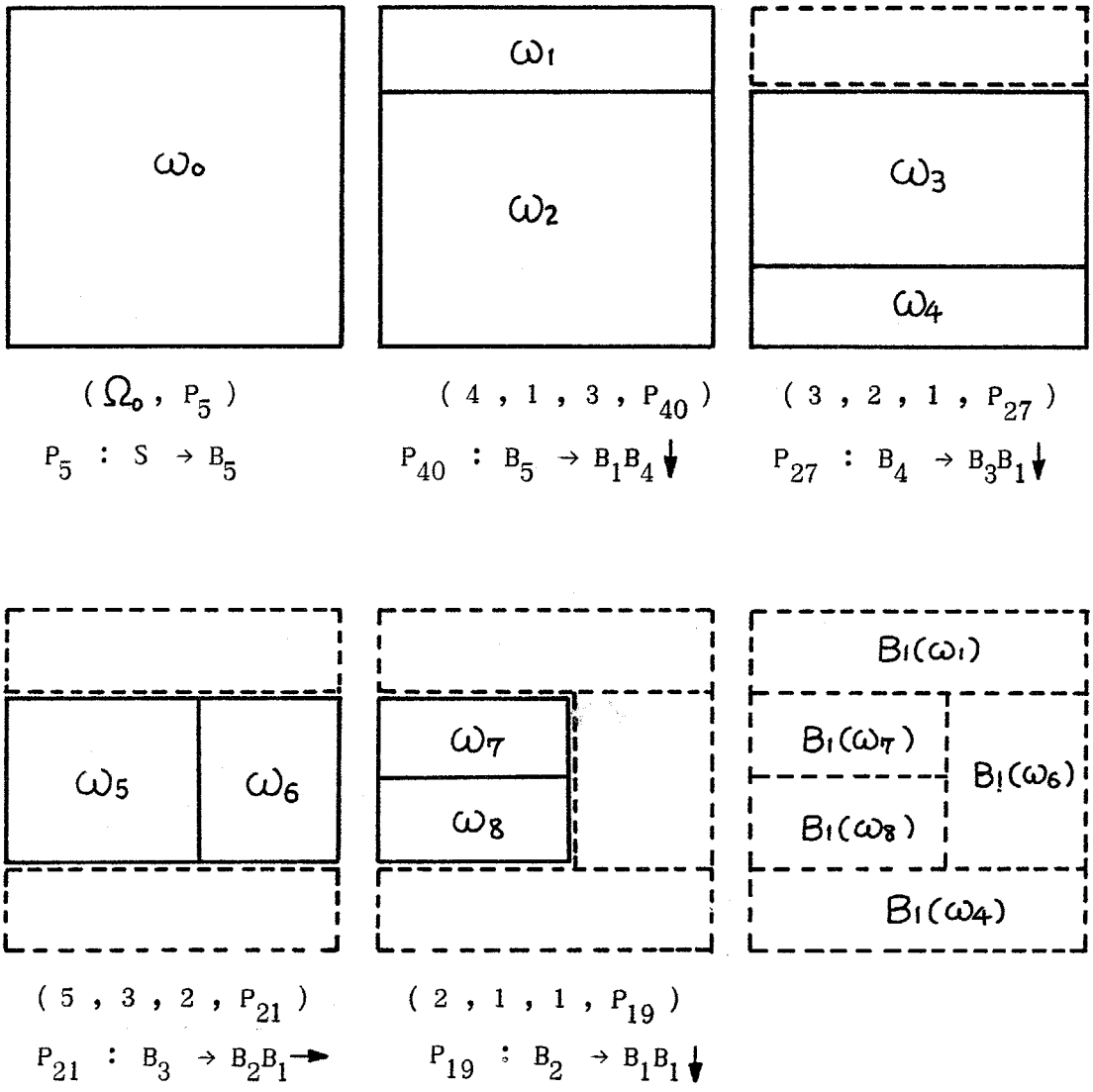


図 7・2・9 生成規則と仮領域の生成過程

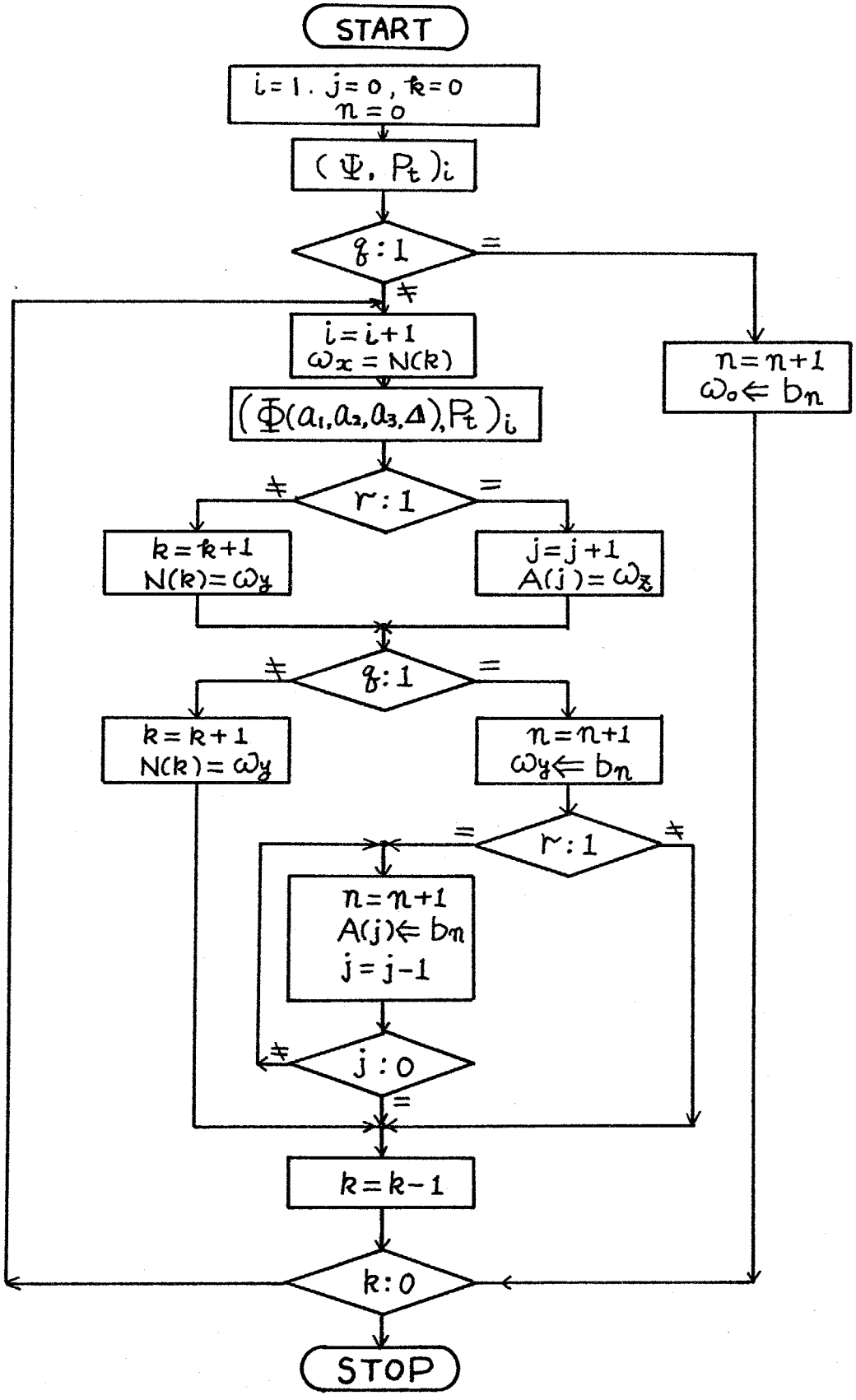


図 7.2.10 漢字パターン生成システムの概要図

7-3 ブロックの生成

7-3-1 ブロックの実領域の決定

図7.2.10に示したアルゴリズムに従って各終端記号B_{11}に対応する仮領域の中にブロックを生成するが、仮領域をそのままブロックの実領域としたのでは、漢字らしくないパターンが生成されてしまう。そこで、仮領域に対する実領域の大きさを次の定義で定めね。

〔定義7-3〕

包含関係以外の位置関係にある終端記号B_{11}に対応する任意の仮領域に対し、図7.3.1に示す9種類の実領域を定義すね。

図7.3.1において、実線で囲まれた部分が仮領域を表わし、一点鎖線で囲まれた部分が実領域を表わす。

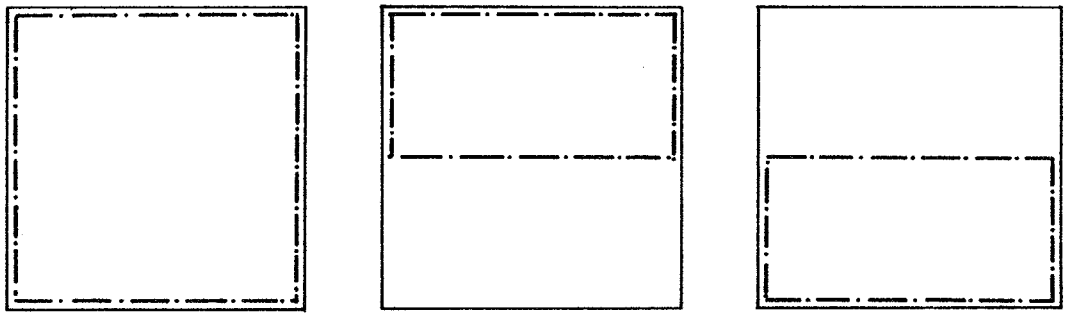
〔定義7-4〕

上下関係、左右関係を分割の対方向とするとき、定義7-3で決定される実領域は、その実領域に対応する仮領域の生成を行った分割位置関係の対方向に対して圧縮の自由度を持つ。但し、包含関係を有するブロックの実領域に関しては、左右方向のみの圧縮を許す。

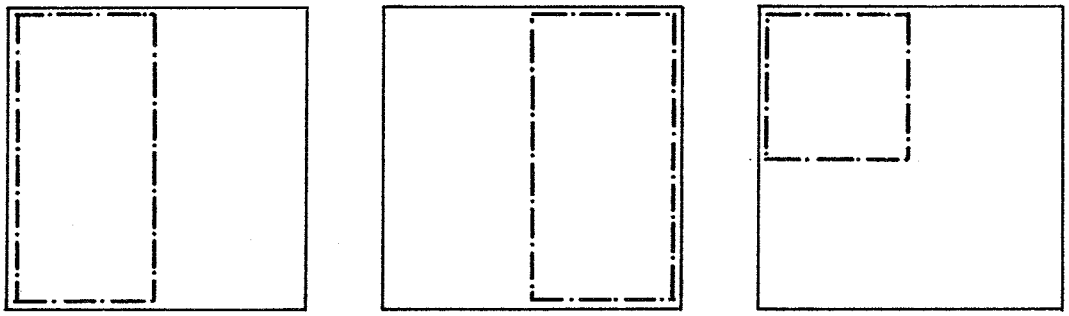
〔規則7-2〕

圧縮係数をα（$0 < \alpha \leq 10$の整数）とすね。そのときの圧縮率は定義7-3で決定した実領域の圧縮方向の長さの$\alpha/10$と定めね。

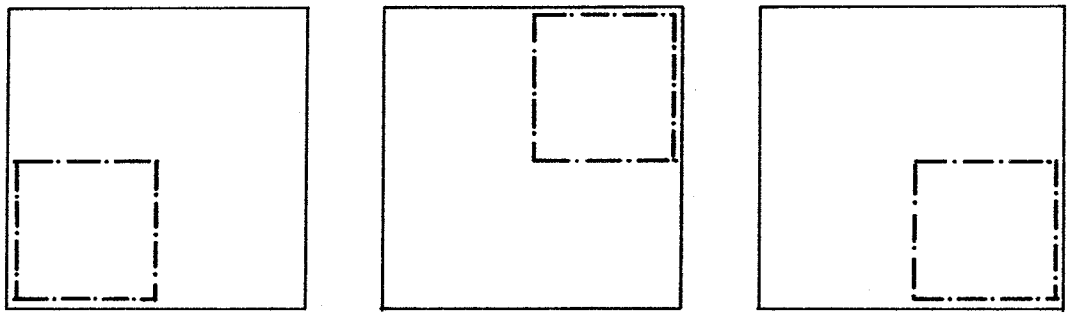
規則7-2は、10段階の圧縮を定めた規則で$\alpha = 10$のときは圧縮を行わず、$\alpha = 5$の場合は分割方向と対の方向のみが1/2に圧縮された実領域が得られね。



TYPE 0　　TYPE 1　　TYPE 2



TYPE 3　　TYPE 4　　TYPE 5



TYPE 6　　TYPE 7　　TYPE 8

実線で囲まれた部分 ・・・ 仮領域

一点鎖線で囲まれた部分・・・ 実領域

図 7・3・1　仮領域に対する実領域の大きさと位置

定義7-3，定義7-4によって，各ブロックの実領域が決定される。そこで，実領域の大きさと圧縮率を与えるパラメータを属性情報として各ブロックに付加する。すなわち，あるブロックbiに対し，

$$b_i(t_i, d_i)$$

と記す。但し，t_iは，b_iの実領域の占める位置を表わすパラメータで，図7.3.1に示した各タイプの番号である。また，d_iは規則7-2で述べた圧縮係数である。

一方，図7.2.10に示した生成システムは，ブロックの生成に関して最左導出になるように構成されているため，ブロックのカテゴリー列は，式(5-3-3)で表わされるパターンの表現式の左側からみた順序どおりに与えることができる。

前節で述べたブロックの仮領域の生成方法と，実領域の決定の仕方について図7.3.2の例をもとに説明する。図7.3.2のブロックの配置例におけるインフィックスの表現式は，

$$b_1 \downarrow (b_2 \rightarrow b_3)$$

となるが，この表現式のポストフィックス形式の構造表現式の生成過程は，

P_3 ; $S \rightarrow B_3$ $\qquad$ B_3

P_{24} ; $B_3 \rightarrow B_1B_2\downarrow$ $\qquad$ $B_1B_2\downarrow$

P_{18} ; $B_2 \rightarrow B_1B_1\rightarrow$ $\qquad$ $B_1B_1B_1\rightarrow\downarrow$

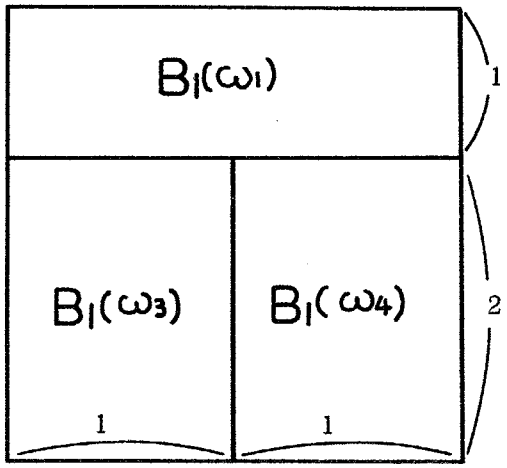
となる。従って，図7.3.2(a)の仮領域の分割比と，同図(b)の実領域の大きさから，図7.2.10に示したパターン生成システムへの入力として

(Ω_0, P_3) $\qquad$ $b_1(0, 5)$

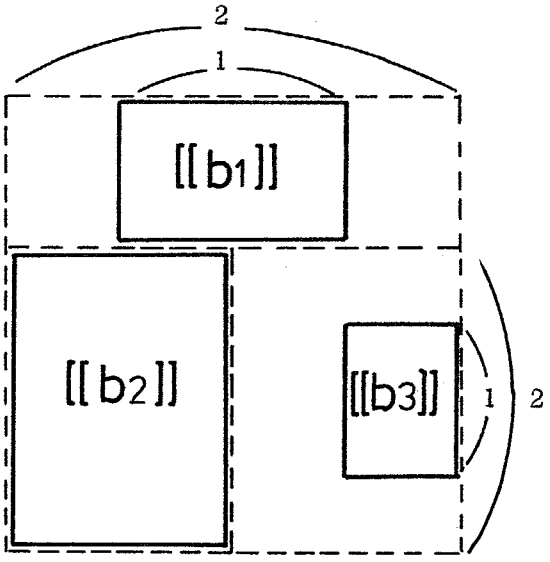
$(3, 1, 2, P_{24})$ $\qquad$ $b_2(0, 10)$

$(2, 1, 1, P_{18})$ $\qquad$ $b_3(4, 5)$

の生成規則列とブロックのカテゴリー列を入力すればよい。



(a) 仮領域の例



(b) 実領域の例

図 7・3・2 仮領域から実領域を決定する例

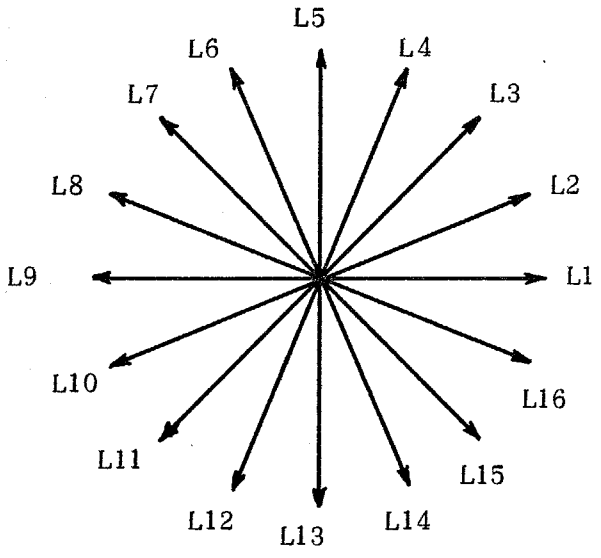
7-3-2 ブロックの生成

ブロックの種類としては、第三章で述べた3種類のブロックを用いる。即ち、基本ブロック、近接ブロック、クロスブロックの3種類である。近接、クロスの両ブロックの構成要素は基本ブロックであり、基本ブロックの生成が基本的な操作となる。

第三章で述べたように、連結関係Cによる線素の合成によって各基本ブロックが構成されるが、本論文のパターン生成モデルで用いる線素として、図7.3.3に示す16種類の線素を用いる。これらの線素は、ストローク方向を16方向に量子化した場合の方向成分のいずれかを持ったものになる。例えば、図7.3.4(a)に示したストロークに対して定めた基本ブロックは、

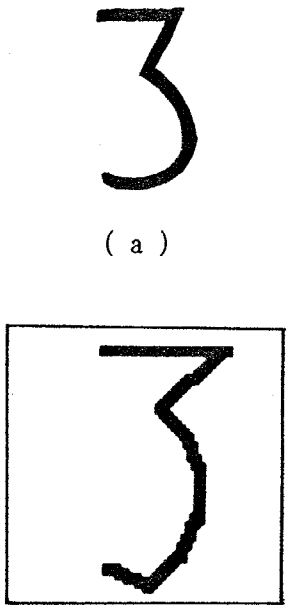
〔L1, L1, L11, L11, L15, L14, L13, L12, L11, L7〕

であり、この基本ブロックを生成した例が図7.3.4(b)である。



矢印の向き ： 線素の方向

図 7・3・3 16種類の線素の例



(a)

(b)

図 7・3・4
ストロークとそれに対応する基本ブロックの一例

漢字らしさということの評価を厳密に行うために何を利用すべきかは明確ではないが、我々が見て判断する限りにおいて漢字らしいというパターンを生成するためには、まず、基本ブロックをどのように構成するかが大きな問題となる。本論文では、漢字を構成する各ストロークを参考にして概念的に設計したもので、必ずしも最適な基本ブロック構成になっているとは限らない。

図7.3.4(b)に示したような基本ブロックの生成は次のような手順で行う。

[ステップ1]

線素表現された基本ブロックに対し、その各線素の合成を行った場合のx方向、y方向のユニット数を求める。

今、ある基本ブロックf_iが$[l_{i1}, l_{i2}, \cdots, l_{in}]$($l_{ik} \in L$)で表現されているとき、x, y方向のユニット数$U_x, U_y$は次のような式によって得られる。

$$U_{x1} = \min_{1 \le j \le n} \{ U_{x1}, \sum_{k=1}^{j} \cos \angle l_{ik} \}, \quad U_{x2} = \max_{1 \le j \le n} \{ U_{x2}, \sum_{k=1}^{j} \cos \angle l_{ik} \}$$

$$U_{y1} = \min_{1 \le j \le n} \{ U_{y1} \sum_{k=1}^{j} \sin \angle l_{ik} \}, \quad U_{y2} = \max_{1 \le j \le n} \{ U_{y2}, \sum_{k=1}^{j} \sin \angle l_{ik} \}$$

$$U_x = U_{x2} - U_{x1}, \quad U_y = U_{y2} - U_{y1}$$

但し、$U_{x1}, U_{x2}, U_{y1}, U_{y2}$の初期値は0で、$\angle l_{ik}$は線素$l_{ik}$の始点からみた$l_{ik}$の方向角である。

[ステップ2]

基本ブロックf_iに割り当てられた実領域のx方向、y方向の長さから、基本ブロックf_iのx方向、y方向の単位ユニット長を求める。

実領域のx方向、y方向の長さをA_x, A_yとするとき、x方向、y方向の単位ユニット長u_x, u_yは、

$$u_x = A_x / U_x, \quad u_y = A_y / U_y$$

で与えられる。

〔ステップ3〕

基本ブロック f_i の始点 head(f_i) を原点として f_i を生成したときに得られる領域の代表点が、与えられた実領域の代表点の座標に一致するように head(f_i) の座標を求める。head(f_i) を原点として f_i を生成したときの領域の代表点の座標 (x_i, y_i) は、

$$x_i = (U_{x1} + U_{x2}) \times u_x / 2.$$

$$y_i = (U_{y1} + U_{y2}) \times u_y / 2$$

で求めることができる。従って、与えられた実領域の代表点の座標を (X_i, Y_i) とすれば、(x_i, y_i) を (X_i, Y_i) に一致させた場合の head(f_i) の座標 (x_{is}, y_{is}) は、

$$x_{is} = x_i - X_i \quad , \quad y_{is} = y_i - Y_i$$

となる。

〔ステップ4〕

ステップ3で得られた始点 (x_{is}, y_{is}) を出発点として f_i を構成する各線素を合成する。そのときの各線素の長さを

$l_{ik} \in \{L1, L10\}$ のとき u_y

$l_{ik} \in \{L5, L13\}$ のとき u_x

$l_{ik} \in L - \{L1, L10, L5, L13\}$ のとき $\max\{u_x, u_y\}$

とする。

以上が基本ブロックの生成法である。一方、近接ブロック、クロスブロックに関しては、第三章で述べたブロックの構造表現式を利用せずに、前節で述べた領域分割操作のみによって基本ブロックとして生成することも可能であるが、パターン生成の処理効率を考慮し、構造表現式を利用する。

生成するブロックが近接ブロックであるとき、その近接ブロックを構成する基本ブロックに対し、近接関係にある点を同一座標とみなしてステップ1，2を行う。次に、いずれかの基本ブロックの始点を原点としてその近接ブロック

を構成したときの領域の代表点が、与えられた実領域の代表点に一致するように、それぞれの基本ブロックの始点を決定する。それぞれの基本ブロックの始点が決まれば、ステップ4に従って各基本ブロックの生成を行う。

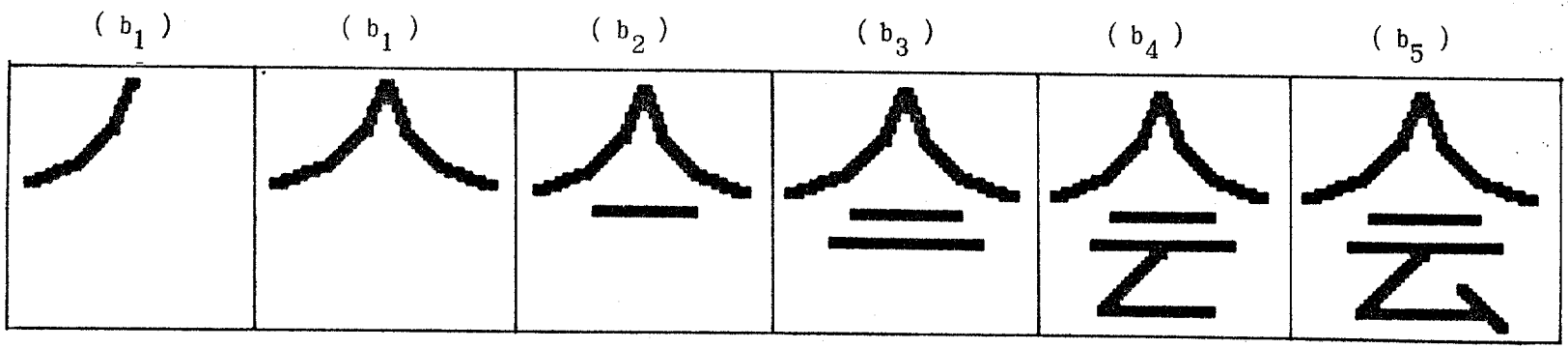
クロスブロックに関しては、交点の存在する線素番号を利用して各基本ブロックを構成する線素のユニット数を求める。即ち、ある線素上に交点が存在しているとき、その交点数だけ線素のユニット数を増す。従って、交点の存在しない線素のユニット数は1，交点がm個存在する線素のユニット数は(m+1)となる。各基本ブロックを構成する線素のユニット数から、そのクロスブロックのx方向，y方向のユニット数をもとめる。そして、幹ブロックの始点を原点としてそのクロスブロックを構成したときの領域の代表点が，与えられた実領域の代表点が一致するように各基本ブロックの始点の座標を決定し、それぞれの基本ブロックを生成する。

'会'という文字パターンの生成過程の例を図7.3.5に示す。このパターンを生成するために、図7.2.10に示した生成システムに入力した情報としては、

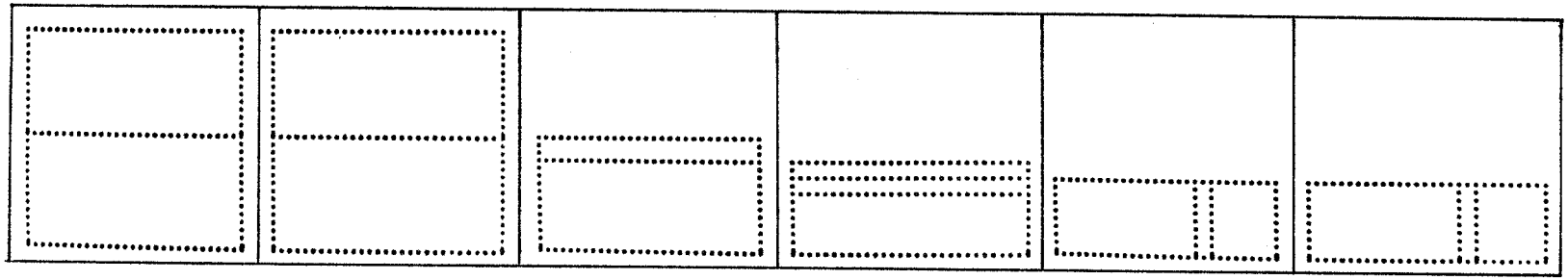
(Ω_0, P_5)　　$b_1 = 65(0, 10)$
$(6, 3, 4, P_{40})$　　$b_2 = 1(0, 4)$
$(8, 3, 6, P_{31})$　　$b_3 = 1(0, 6)$
$(5, 2, 4, P_{24})$　　$b_4 = 21(0, 8)$
$(14, 10, 5, P_{18})$　　$b_5 = 12(3, 6)$

である。但し、Ω_0は60×60の大きさとした。図7.3.5において(B)の各パターンが仮領域生成過程を示し、(A)のパターンがブロックの生成過程を示したものである。ブロックb_1は近接ブロックであるため、仮領域の分割が1回であるのに対し、ブロックの生成は2段階で行われている。また、それぞれのブロックの実領域を示したものが(C)のパターンである。

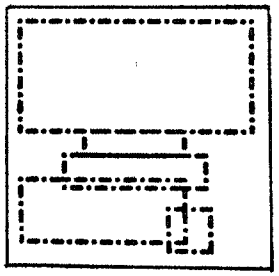
図7.3.6に、'吾'パターンを例にとり、分割比と圧縮係数を変化させた場合の例を示す。(a)のパターンは、圧縮係数を一定にとり分割比を変化させたものであり、(b)のパターンは分割比を一定にして圧縮係数を変化させたもので



(A) ブロックの生成過程

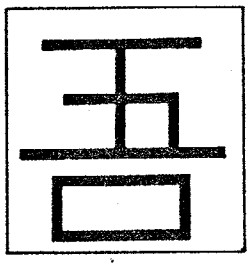


(B) 仮領域の生成過程

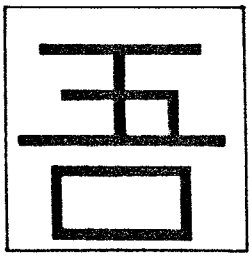


(C) 各ブロックの実領域

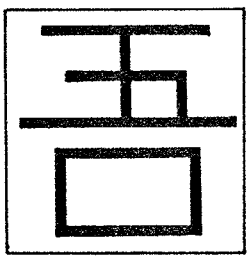
図 7・3・5 会 パターンに対する仮領域とブロックの生成過程



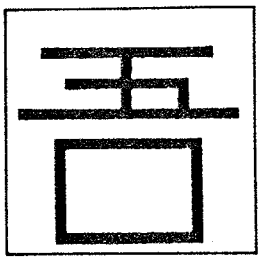
(Ω_o, P_5)
$(8,2,7, P_{40})$
$(7,4,5, P_{31})$
$(5,2,4, P_{24})$
$(4,2,2, P_{19})$



(Ω_o, P_5)
$(7,2,6, P_{40})$
$(12,6,9, P_{31})$
$(9,3,8, P_{24})$
$(8,4,4, P_{19})$



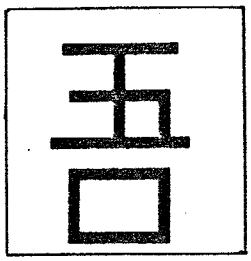
(Ω_o, P_5)
$(14,2,13, P_{40})$
$(13,6,10, P_{31})$
$(10,3,9, P_{24})$
$(9,4,5, P_{19})$



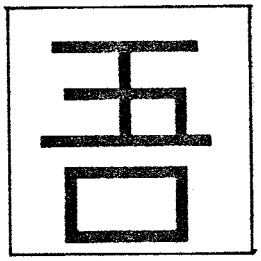
(Ω_o, P_5)
$(7,2,6, P_{40})$
$(12,4,10, P_{31})$
$(10,2,10, P_{24})$
$(10,4,6, P_{19})$

$b_1=1 \quad (0,7)$
$b_2=7 \quad (0,5)$
$b_3=14 \quad (0,5)$
$b_4=1 \quad (0,9)$
$b_5=70 \quad (0,6)$

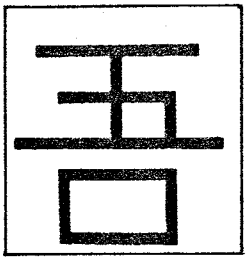
(a)



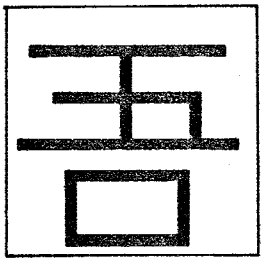
$b_1=1 \quad (0,5)$
$b_2=7 \quad (0,10)$
$b_3=14 \quad (0,4)$
$b_4=1 \quad (0,6)$
$b_5=70 \quad (0,4)$



$b_1=1 \quad (0,6)$
$b_2=7 \quad (0,10)$
$b_3=14 \quad (0,5)$
$b_4=1 \quad (0,7)$
$b_5=70 \quad (0,5)$



$b_1=1 \quad (0,7)$
$b_2=7 \quad (0,10)$
$b_3=14 \quad (0,5)$
$b_4=1 \quad (0,9)$
$b_5=70 \quad (0,5)$



$b_1=1 \quad (0,8)$
$b_2=7 \quad (0,10)$
$b_3=14 \quad (0,6)$
$b_4=1 \quad (0,9)$
$b_5=70 \quad (0,5)$

(Ω_o, P_5)
$(7,2,6, P_{40})$
$(12,6,9, P_{31})$
$(9,3,8, P_{24})$
$(8,4,4, P_{19})$

(b)

図 7・3・6　分割比と圧縮係数を変化させた場合のパターンの変化例

ある。各パターンの下に分割比、および圧縮係数を示した。

図7.3.6の例からもわかるように、仮領域の分割比と実領域の圧縮係数を変化させることによって様々なパターンを生成することができる。逆に、自然な漢字パターンを生成するためには、基本ブロックの設計を最適な設計にすることと共に、各ブロックの大きさを決定する仮領域の分割比、並びに圧縮係数の値をうまく決めてやらなければならない。

同様な手法で生成したパターンの例と実領域の例とを、図7.3.7から図7.3.14に示す。これらのパターンを生成するためのパラメータは、極端に文字らしくないような場合を除くという条件で決定しているため、必ずしも自然な形になっているとは限らない。近接ブロックやクロスブロックを構成する段階で、ストロークの長さや形状に対する自由度が制限されていることも、文字パターンの不自然さの一因になっていると考えられる。

本研究において生成した文字パターンの中の一部の例を図7.3.15に示す。

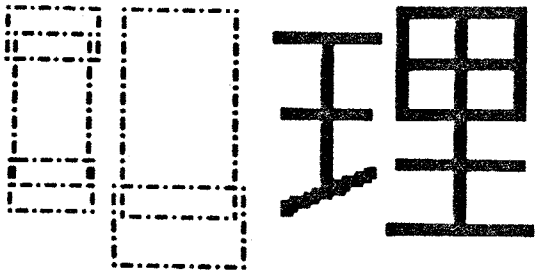


図 7.3.7 理 パターンの例

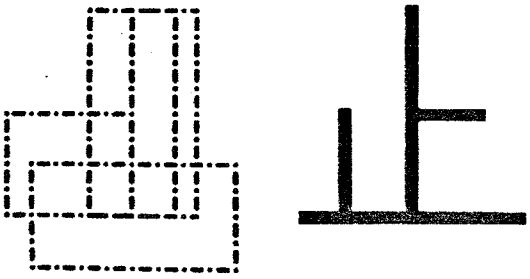


図 7.3.8 止 パターンの例

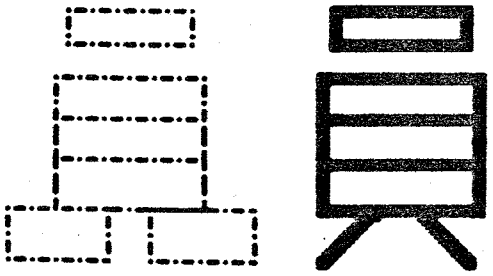


図 7.3.9 員 パターンの例

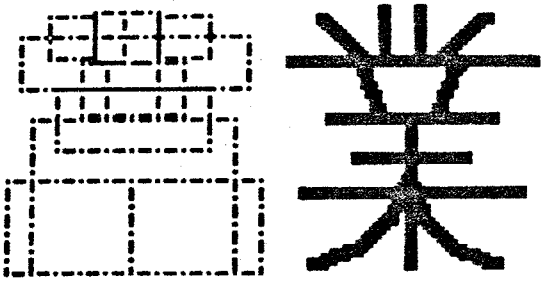


図 7.3.10 業 パターンの例

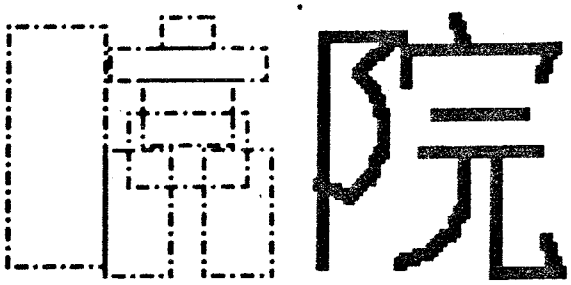


図 7・3・11　院　パターンの例

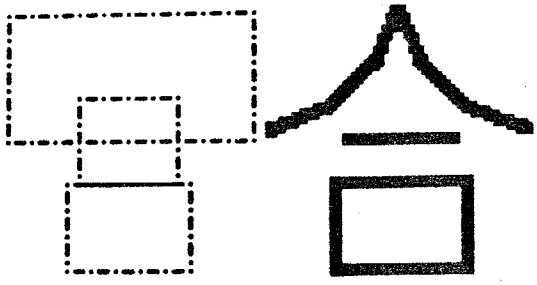


図 7・3・12　合　パターンの例

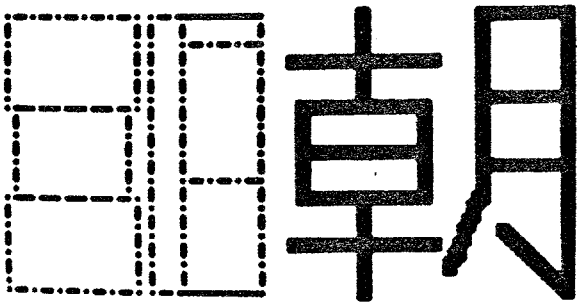


図 7・3・13　朝　パターンの例

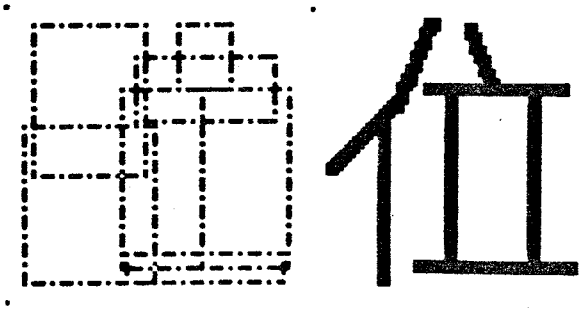


図 7・3・14　位　パターンの例

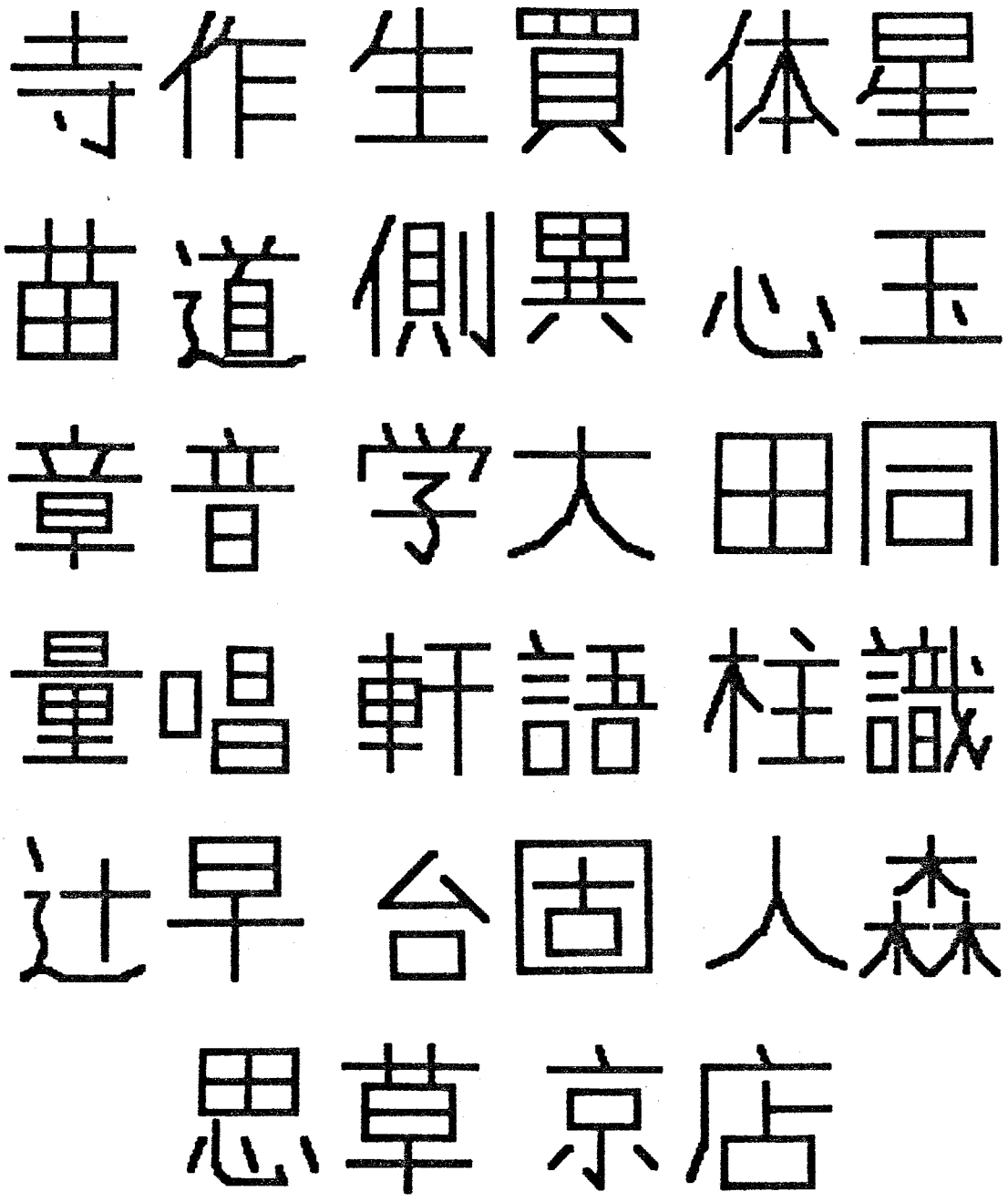


図 7・3・15　生成文字パターンの例

ところで、漢字の書順について考えてみると、上から下、左から右の順序で書くという大法則があるが、この法則にあてはまらない例外的な漢字も多く見られる。一方、本章で述べている漢字パターン生成モデルにおけるブロックの生成は、生成モデルの最左導出規則により、上から下、左から右という順序で行われるが、これは各ブロックの仮領域の決定と同時にブロックを生成することを意味する。この規則とは別に、ブロックの仮領域の生成とブロックの生成とを分離して行えるのも本生成モデルの特徴の一つである。領域分割の定義上、ブロックの仮領域の生成を任意の順序で行うことはできないが、仮領域生成後においては、各ブロックの生成順序を自由に決めることができる。従って、漢字の書順に対応させて各ブロックを生成することが可能となる。例えば、'歩'というパターンを最左導出規則に従って生成すれば、図7.3.16に示したような順序で各ブロックが生成される。これは、'歩'パターンが次式に示されるような表現式になるからである。

$$(b_1 \rightarrow b_2 \rightarrow b_3) \downarrow b_4 \downarrow (b_5 \rightarrow b_6 \rightarrow b_7) \downarrow b_8$$

図7.3.16の生成順序は明らかに'歩'の書順と異なっている。そこで、生成システムに対し、ブロックの生成順序を$b_2, b_3, b_1, b_4, b_6, b_5, b_7, b_8$の順に指示すれば、図7.3.17のように、正しい書順に対応した順序で各ブロックを生成することができる。

漢字の書順ということについて考えた場合、生成された漢字パターンばかりでなく、そのパターンを生成する過程自体が重要な情報となる。本研究で述べた生成モデルは、この書順情報を効果的に利用できるという特徴を持っている。この特徴は従来の生成モデルには見られない特徴であり、現在の日本語情報処理システムでは扱うことのできなかった情報である。

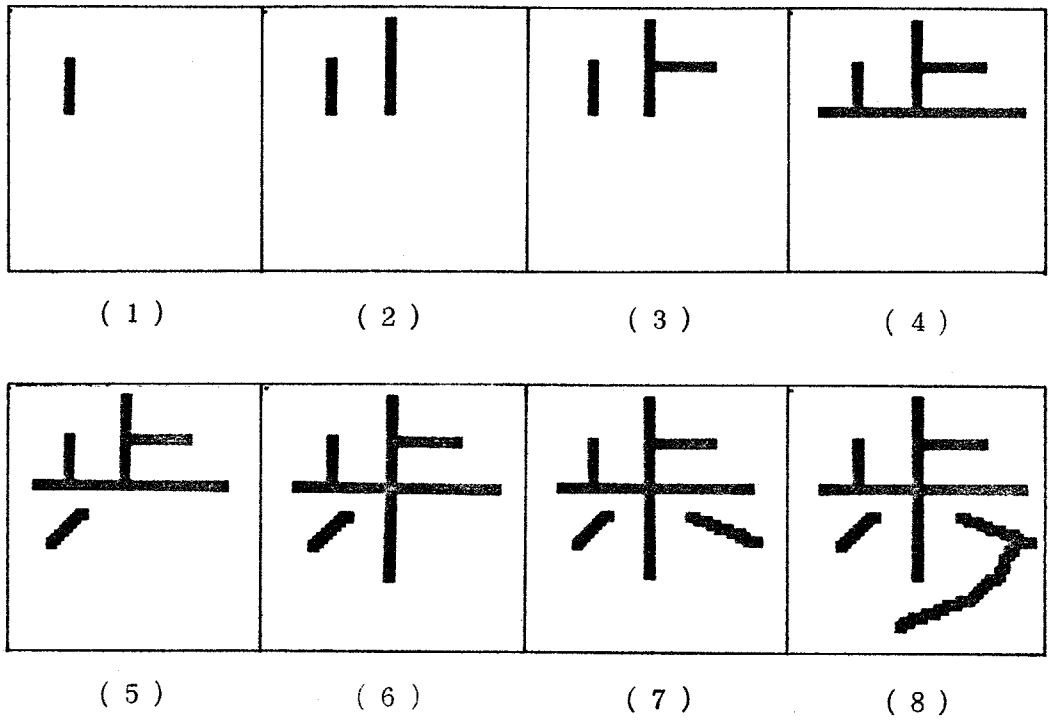


図 7・3・16　最左導出順序によるブロックの生成例

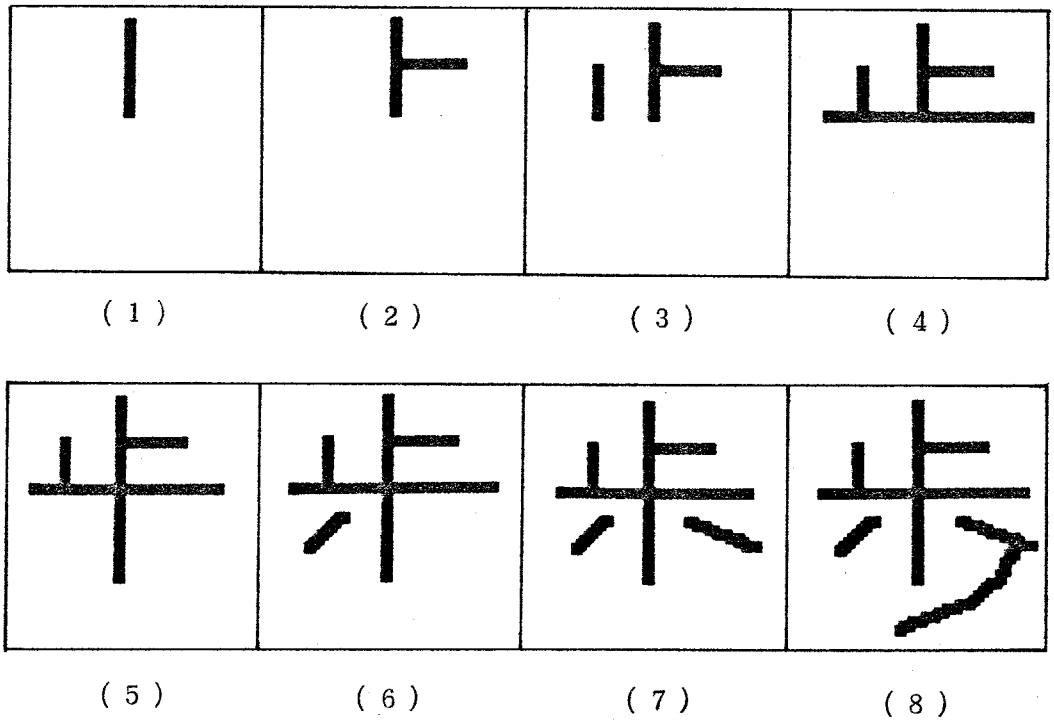


図 7・3・17　正しい書順によるブロックの生成例

7-4 本生成モデルの検討

第五章までに述べた漢字パターンの表現法は、パターンの再現性という点からみれば、可逆性のない表現法であった。表現式にパターンの再構成という可逆性を持たせるために、本章において領域分割のための3つのパラメータと、各ブロックの大きさを規定する2つのパラメータを導入した。そして、これらのパラメータを付加した表現法によって実際にパターンの再構成処理を行ったのが本章の漢字パターンの生成モデルである。従って、本章で述べた生成モデルは、パターンの発生のみを目的としているものではなく、あくまでも漢字パターンの構造的表現式による記述を、より有効的なものにすることを目的としたものである。

ところで、前節において例として示した幾つかの生成パターンを、印刷文字等と比較した場合、かなり形状の異ったパターンがある。これは、パターン生成の際に、生成されたパターンが極度に不自然でない限り、あるいは、連結すべき所が連結している限り、良しとして5つのパラメータを決定したためである。また、各パラメータの値は本研究では試行錯誤的作業によって決定したが、第六章までに述べた漢字パターンの表現法を、標準文字パターンに適用することによって自動的に各パラメータの値を決定することも可能であると考えられる。

各ブロックを構成する線素に関しては、8種類、16種類の場合について行ったが、8種類では限られた字種以外、殆どの文字が不自然になる。本論文に示したパターンはすべて、16種類の線素を用いて生成したものであるが、不自然さは残るものの、殆どの場合16種類の線素で生成可能であると考えられる。また、文字の自然さは、5つのパラメータの決定以前に、基本ブロックの設計の良否によって大きく影響されるが、その基本ブロックの設計を概念的に行っている点も不自然さの原因の一つであると考えられる。さらに、文字の自然さを

評価する尺度が明確でないことも、自然な文字の生成を困難なものとしている。

基本ブロックや領域分割のためのパラメータ、大きさを決めるパラメータ等は、漢字のドットパターン作成の場合と同様に多くの時間と労力を必要とする。しかしながら、一担、決めてしまえば、パターンの大きさに係わらず、常に利用できるという特徴がある。

本生成モデルを日本語情報処理に応用しようとした場合、前章までに述べた漢字の構造表現式と切り離しては考えられない。すなわち、現在の漢字プリンタの速度と印字品質を考慮すれば、本生成モデルによるプリンタとしての利用価値は少ない。むしろ、構造表現式を利用した各種の文字検索、文字合成等の確認作業に、あるいは、漢字プリンタでは不可能な不定形の大きさの文字の印字等に応用することが可能である。さらに、本論文で述べた表現法と簡便な表示装置とを組み合せることによって、図7.3.17に示したような書順に従った生成を行い、漢字習得時の正しい書順の学習における簡易教育機器として利用することなども可能である。

ところで、本生成モデルがすべての文字に対して適用できるか否かは、第五章までに述べた表現法と同様に明確ではない。これを明らかにするためには、すべての漢字パターンに対して構造表現を行い、パターンの再構成を行ってみなければならない。しかしながら、この作業は個人的作業の域を越えたものであり、現時点での議論を、本表現法の適用できる文字パターンに限定せざるを得ない。従って、もし適用できない文字があった場合、どのような修正を行う必要があるかという検討が残されている。

第八章 あとがき

計算機と言語処理との関係について、主にその入出力に視点を置き、年代別あるいは方法論的に現在までの技術や装置について概略的に述べた。本論文でとりあげた方法以外にも、多くの方法が考えられ、また秀れた研究が行われている。

我国では、長年、表意文字である漢字の存在が計算機による日本語情報処理技術進展の障害になっていたが、多くの研究者、技術者の努力によって、漢字を含めた日本語の処理を計算機によって行うことが可能となってきた。

本論文は、このような観点から、漢字に対するより高度な情報処理、あるいは、処理の効率化を行うことを目的の一つとして、漢字パターンの表現法について述べた。従来の表現法が連結関係を重視するために漢字構成上の自然さを欠いているのに対し、本論文で述べた表現法は、我々が漢字に対して持っているイメージにより近い形式の表現法であった。そして、その表現法を計算機で効果的に利用することができるように、形式言語理論の立場から一つの生成文法を定義し、表現式の符号化を行った。さらに、その生成文法によって生成される文と漢字パターンの表現式との関係について論じた。

本論文で述べた表現法の応用の一つとして、手書き漢字パターンの表現について考え、その表現法を利用した手書き漢字パターンの識別法について、幾つかの実験例をもとに検討した。その結果、基本ブロックの構成法、あるいは識別法等について、多くのデータに対する実験を行って検討を加えれば、手書き漢字パターンの認識法として有効な方法になり得るという結論に致った。

本論文で述べた表現法のもう一つの応用として、漢字パターンの生成モデルについて述べた。その生成モデルにおいて利用した表現式は、幾つかのパラメータが付加されてはいるものの、本質的には、第四章、第五章を通じて述べた漢字パターンの表現法において得られる表現式と同一のものであった。一方、

本生成モデルにおいて生成される漢字パターンの自然さは、文字パターンの構成要素である各基本ブロックの設計の如何によって大きく左右されね。しかしながら、この生成モデルの目的は、本論文で述べた表現法によってパターンの生成をも行うことができるということを示すことであり、生成実験に用いた基本ブロックについては、比較的自由に設定した。従って、生成されたパターンが自然さを欠いている点は、ある程度やむを得ない。

以上に述べた手書き漢字パターンの表現法や漢字パターンの生成モデルは、いずれも漢字情報処理の入出力的立場から見た場合の本表現法の応用の一例であるが、これからの日本語情報処理における重要な課題は、構文解析、意味解析、あるいは文脈把握等による日本語文の計算機による理解であね。この場合、表意文字である漢字を計算機がどの程度理解することができるかが重要な鍵となね。この漢字の理解ということを計算機に行わせる上で、本論文で述べた表現法は有効な方法であると考えられね。

最後に、本研究で述べた表現法の特徴についてまとめると、漢字処理システムに対するパターン認識による手書き漢字の入力、システム内部での情報検索、意味解析、さらに、パターンのシステム外部への出力といった諸どの処理を、本論文で述べた表現法によって行えね可能性が十分にあるということであね。しかし、これらの情報処理への応用について十分な考察と検討を行うのには、あまりにも大きな課題であり、この作業を完遂するためには多くの時間と労力が必要であね。

謝辞

本研究を進めるにあたり、終始御指導いただきました安居院猛助教授、および、貴重な助言や討論をしていただきました中嶋正之助手に心から感謝致します。また、安居院研究室の皆様にはいろいろ御協力をしていただきました。合せて感謝致します。

参考文献

(1) 山田博著：コンピューターアーキテクチャ，産業図書 昭和51年

(2) 坂井利之編：文字図形の認識機械 共立出版 昭和42年

(3) 石綿敏雄 "言語学と人工知能" 情報処理 Vol.19, NO.10 P.P.913～920 (1978)

(4) 長尾真 "自然言語の理解" 情報処理 Vol.19, NO.10, P.P952～961 (1978)

(5) 長尾真 "言語情報処理の過去・現在・将来" 情報処理 Vol.19, NO.2 P.P.106～112 (1978)

(6) 長尾、辻井 "自然言語処理プログラム" 情報処理 Vol.18, NO.1 P.P63～75 (1977)

(7) 長尾、辻井、山上、建部 "国語辞書の記憶と日本語文の自動分割" 情報処理 Vol.19, NO.6 P.P.514～521 (1978)

(8) 関英男著：情報理論，オーム社，昭和44年

(9) 安田秀明 "漢字情報処理技術の現状と展望[I]" 電子通信学会誌 Vol.58, NO.7, P.P.754～762

(10) 国立国語研究所："電子計算機による新聞の語彙調査" 国立国語研究所報告 37, P.P.142～ (昭和45-3)

(11) J.E.Hopcroft, J.D. Ullman共著(野崎、木村他訳)：言語理論とオートマトン，サイエンス社 昭和46年

(12) 高橋延匡 "日本語情報処理への期待" 情報処理 Vol.20, NO.1 P.P.44～49 (1979)

(13) 安田秀明 "漢字情報処理技術の現状と展望[I],[II]" 電子通信学会誌 Vol.58, NO.7, NO.8

(14) 牧野、勝部、木澤 "カナ漢字変換の一方法" 情報処理 Vol.18, NO.7 P.P.656～663 (1977)

(15) 松下、山崎、佐藤 "漢字かな混り文変換システム" 情報処理, Vol.15, NO.1, P.P.2～ (1974)

(16) 増田功"オンライン手書き文字認識"電子通信学会誌. Vol.61. NO.2 P.P.125～ (1978)
(17) 長谷川実郎"パターン合成による漢字入出力処理"情報処理 Vol.16 NO.9, P.P.808～817(1975)
(18) 画像電子学会編集委員会"漢字情報処理システム用電子式漢字プリンタおよび写真植字機の仕様一覧"画像電子学会誌 Vol.3, NO.1, P.P.36～49 (1974)
(19) 黒崎悦明"高速漢字プリンタシステムについて"情報処理. Vol16. NO.9 P.P. 802～807(1975)
(20) 長谷川実郎"漢字パターン発生システム"画像電子学会誌 Vol.3, NO.4 P.P.230～242(1974)
(21) 漢字処理研究会：漢字情報処理システム資料集, 日本工業技術センター P.P91～110, 昭和54年
(22) 電気学会特集"パターン認識"電気学会誌 Vol.93, NO.11. P.P.1～88 (1973)
(23) 電子通信学会"文字認識小特集"電子通信学会誌 Vol.61, NO.1 (1978)
(24) 中田和男編：パターン認識とその応用. コロナ社(1978)
(25) 森晃徳, 森俊二, 山本"場の効果法による特徴抽出－閉じ状態の抽出－" 信学論誌 74/5 Vol57-D, NO.5. P.P.308～315(1974)
(26) 山本, 森, 山田"凹凸構造抽出による手書きひらがな文字認識" 信学技報 PRL78-38, P.P.1～10 (1978)
(27) 吉田実"ストローク抽出法による手書き漢字認識システム" 信学総全大 1543 昭和49年
(28) 吉田実"手書き住所の認識" 信学総全大 S10-6 昭和52年
(29) 安居院, 中嶋, 長橋"線と領域を主体とした漢字認識について" 信学総全大S10-7, 昭和52年
(30) 安居院, 中嶋, 長橋"線と領域を主体とした漢字認識に関する研究" 信学技報 PRL77-4 P.P.27～36(1977)

(31) B. Rankin "A linguistic study of the formation of Chinese characters" Ph.D diseration, Univ. of Pennsylvania, Philadelphia, 1965

(32) 宝居院, 中嶋, 長橋 "部分パターンの位置関係を利用した手書き漢字の表現法" 信学論誌 Vol.60-D, NO.12, P.P.1109~1116

(33) 宝居院, 長橋 "漢字パターンの表現法について" 信学技報 PRL77-49 P.P. 1~8 (1978)

(34) 長橋, 宝居院 "手書き漢字パターンの符号化について" 信学論誌 78/11 Vol. J61-D, NO.11, P.P.803~810 (1978)

(35) K. S. Fu : Syntactic Method in Pattern Recognition, New York Academic 1975

(36) K.S. Fu et al., : Syntactic Pattern Recognition Application, New York, Springer, 1977, P.P. 95~123

(37) B. Rankin and S. Siegal. "A Grammar for component combination in Chinese characters" NBS Tech. Note 296, (1966.)

(38) Alan.C. Shaw ; "Parsing of graph-represented pictures" JACM Vol.17, NO.3, July. P.P.453~481 (1970)

(39) 林達也著 : 電子計算機のシステムプログラム, 産報 1970

(40) 電気学会大学講座 : 情報処理工学, 電気学会 昭和48年

(41) 水谷静夫著 : 言語と数学, 森北出版 1973.

(42) C.L. Hamblin, "Translation to and from Polish notation" Comp. J. Vol.3, P.P. 210~213 1962

(43) Stallingo, W. "The Morphology of Chinese characters ; A survey of Models and Applications" Computers and the Humanities, Vol.9 P.P. 13~24, (1975)

(44) 小川, 手塚 ; "漢字の階層表現とその認識" 信学論誌 Vol.57-D, NO.12, P.P. 700~707 (1974)

(45) 内藤，淀川："手書き漢字のストローク密度関数による大分類"
信学技報 PRL 79-3 P.P. 19～28 (1979)

(46) 田村；"図形の細線化についての比較研究"情報処理イメージプロセッシング 1, 1975

(47) T. Agui, H. Nagahashi; "A description method of handprinted Chinese characters" IEEE trans. on PAMI, Vol. PAMI-1, NO. 1, Jan. P.P. 20～24 (1979)

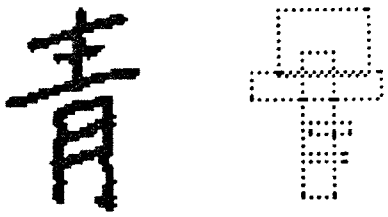
(48) T. Agui, H. Nagahashi: "A coding method of handprinted Chinese characters" IEEE. trans. on PAMI, Vol PAMI-1. NO. 4, Oct. P.P. 333～341 (1979)

(49) Fujimura, O. & Kagaya, R. "Structural patterns of Chinese characters" Annual Bulletin of the Research Institute of Logopedics & Phoniatrics. Univ. of Tokyo, NO. 3, P.P. 131～148 (1969)

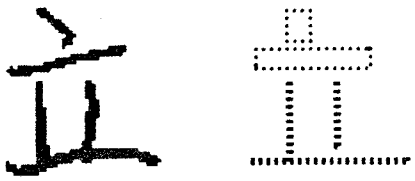
(50) 白川静著：漢字の世界 1, 2, - 中国文化の原点, 平凡社 1976

(51) C.T. Fike 著, 赤木, 加賀美, 鈴木 共訳：科学者のためのPL/1；共立出版 (1973)

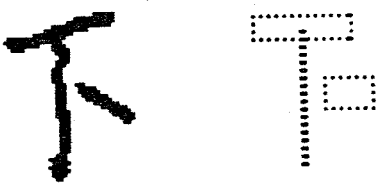
付　録



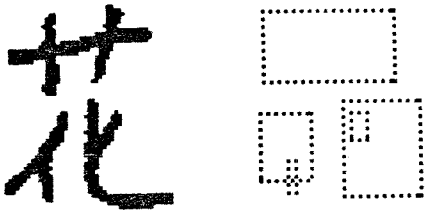
PRODUCTION RULE:
5 40 31 24 19
BLOCK CATEGORY:
218 1 101 1 1



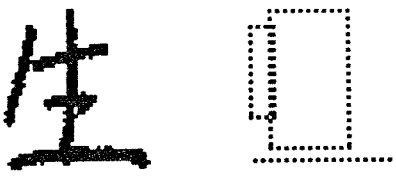
PRODUCTION RULE:
5 40 31 22 18
BLOCK CATEGORY:
4 1 1 3 3



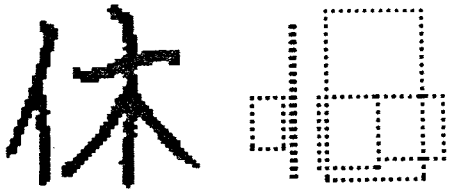
PRODUCTION RULE:
3 24 18
BLOCK CATEGORY:
1 3 4



PRODUCTION RULE:
5 40 28 20 19
BLOCK CATEGORY:
216 7 2 2 3



PRODUCTION RULE:
3 22 18
BLOCK CATEGORY:
1 3 218



PRODUCTION RULE:
5 39 30 24 18
BLOCK CATEGORY:
2 3 209 2 4

手書き漢字パターンの符号化の例

但し、ブロックのカテゴリー列はプログラムの都合上、
ブロックの分類順になっている。

番号	生成規則	番号	生成規則	番号	生成規則
1	$S \rightarrow B_1$	26	$B_4 \rightarrow B_3B_1$→	51	$B_6 \rightarrow B_1B_5$↓
2	$S \rightarrow B_2$	27	$B_4 \rightarrow B_3B_1$↓	52	$B_6 \rightarrow B_1B_5$*
3	$S \rightarrow B_3$	28	$B_4 \rightarrow B_2B_2$→	53	$B_7 \rightarrow B_6B_1$→
4	$S \rightarrow B_4$	29	$B_4 \rightarrow B_2B_2$↓	54	$B_7 \rightarrow B_6B_1$↓
5	$S \rightarrow B_5$	30	$B_4 \rightarrow B_1B_3$→	55	$B_7 \rightarrow B_5B_2$→
6	$S \rightarrow B_6$	31	$B_4 \rightarrow B_1B_3$↓	56	$B_7 \rightarrow B_5B_2$↓
7	$S \rightarrow B_7$	32	$B_4 \rightarrow B_1B_3$*	57	$B_7 \rightarrow B_4B_3$→
8	$S \rightarrow B_8$	33	$B_5 \rightarrow B_4B_1$→	58	$B_7 \rightarrow B_4B_3$↓
9	$S \rightarrow B_9$	34	$B_5 \rightarrow B_4B_1$↓	59	$B_7 \rightarrow B_3B_4$→
10	$S \rightarrow B_{10}$	35	$B_5 \rightarrow B_3B_2$→	60	$B_7 \rightarrow B_3B_4$↓
11	$S \rightarrow B_{11}$	36	$B_5 \rightarrow B_3B_2$↓	61	$B_7 \rightarrow B_2B_5$→
12	$S \rightarrow B_{12}$	37	$B_5 \rightarrow B_2B_3$→	62	$B_7 \rightarrow B_2B_5$↓
13	$S \rightarrow B_{13}$	38	$B_5 \rightarrow B_2B_3$↓	63	$B_7 \rightarrow B_1B_6$→
14	$S \rightarrow B_{14}$	39	$B_5 \rightarrow B_1B_4$→	64	$B_7 \rightarrow B_1B_6$↓
15	$S \rightarrow B_{15}$	40	$B_5 \rightarrow B_1B_4$↓	65	$B_7 \rightarrow B_1B_6$*
16	$S \rightarrow B_{16}$	41	$B_5 \rightarrow B_1B_4$*	66	$B_8 \rightarrow B_7B_1$→
17	$S \rightarrow B_{17}$	42	$B_6 \rightarrow B_5B_1$→	67	$B_8 \rightarrow B_7B_1$↓
18	$B_2 \rightarrow B_1B_1$→	43	$B_6 \rightarrow B_5B_1$↓	68	$B_8 \rightarrow B_6B_2$→
19	$B_2 \rightarrow B_1B_1$↓	44	$B_6 \rightarrow B_4B_2$→	69	$B_8 \rightarrow B_6B_2$↓
20	$B_2 \rightarrow B_1B_1$*	45	$B_6 \rightarrow B_4B_2$↓	70	$B_8 \rightarrow B_5B_3$→
21	$B_3 \rightarrow B_2B_1$→	46	$B_6 \rightarrow B_3B_3$→	71	$B_8 \rightarrow B_5B_3$↓
22	$B_3 \rightarrow B_2B_1$↓	47	$B_6 \rightarrow B_3B_3$↓	72	$B_8 \rightarrow B_4B_4$→
23	$B_3 \rightarrow B_1B_2$→	48	$B_6 \rightarrow B_2B_4$→	73	$B_8 \rightarrow B_4B_4$↓
24	$B_3 \rightarrow B_1B_2$↓	49	$B_6 \rightarrow B_2B_4$↓	74	$B_8 \rightarrow B_3B_5$→
25	$B_3 \rightarrow B_1B_2$*	50	$B_6 \rightarrow B_1B_5$→	75	$B_8 \rightarrow B_3B_5$↓

番号	生成規則	番号	生成規則	番号	生成規則
76	$B_8 \rightarrow B_2B_6\rightarrow$	101	$B_{10} \rightarrow B_8B_2\downarrow$	126	$B_{11} \rightarrow B_6B_5\downarrow$
77	$B_8 \rightarrow B_2B_6\downarrow$	102	$B_{10} \rightarrow B_7B_3\rightarrow$	127	$B_{11} \rightarrow B_5B_6\rightarrow$
78	$B_8 \rightarrow B_1B_7\rightarrow$	103	$B_{10} \rightarrow B_7B_3\downarrow$	128	$B_{11} \rightarrow B_5B_6\downarrow$
79	$B_8 \rightarrow B_1B_7\downarrow$	104	$B_{10} \rightarrow B_6B_4\rightarrow$	129	$B_{11} \rightarrow B_4B_7\rightarrow$
80	$B_8 \rightarrow B_1B_7*$	105	$B_{10} \rightarrow B_6B_4\downarrow$	130	$B_{11} \rightarrow B_4B_7\downarrow$
81	$B_9 \rightarrow B_8B_1\rightarrow$	106	$B_{10} \rightarrow B_5B_5\rightarrow$	131	$B_{11} \rightarrow B_3B_8\rightarrow$
82	$B_9 \rightarrow B_8B_1\downarrow$	107	$B_{10} \rightarrow B_5B_5\downarrow$	132	$B_{11} \rightarrow B_3B_8\downarrow$
83	$B_9 \rightarrow B_7B_2\rightarrow$	108	$B_{10} \rightarrow B_4B_6\rightarrow$	133	$B_{11} \rightarrow B_2B_9\rightarrow$
84	$B_9 \rightarrow B_7B_2\downarrow$	109	$B_{10} \rightarrow B_4B_6\downarrow$	134	$B_{11} \rightarrow B_2B_9\downarrow$
85	$B_9 \rightarrow B_6B_3\rightarrow$	110	$B_{10} \rightarrow B_3B_7\rightarrow$	135	$B_{11} \rightarrow B_1B_{10}\rightarrow$
86	$B_9 \rightarrow B_6B_3\downarrow$	111	$B_{10} \rightarrow B_3B_7\downarrow$	136	$B_{11} \rightarrow B_1B_{10}\downarrow$
87	$B_9 \rightarrow B_5B_4\rightarrow$	112	$B_{10} \rightarrow B_2B_8\rightarrow$	137	$B_{11} \rightarrow B_1B_{10}*$
88	$B_9 \rightarrow B_5B_4\downarrow$	113	$B_{10} \rightarrow B_2B_8\downarrow$	138	$B_{12} \rightarrow B_{11}B_1\rightarrow$
89	$B_9 \rightarrow B_4B_5\rightarrow$	114	$B_{10} \rightarrow B_1B_9\rightarrow$	139	$B_{12} \rightarrow B_{11}B_1\downarrow$
90	$B_9 \rightarrow B_4B_5\downarrow$	115	$B_{10} \rightarrow B_1B_9\downarrow$	140	$B_{12} \rightarrow B_{10}B_2\rightarrow$
91	$B_9 \rightarrow B_3B_6\rightarrow$	116	$B_{10} \rightarrow B_1B_9*$	141	$B_{12} \rightarrow B_{10}B_2\downarrow$
92	$B_9 \rightarrow B_3B_6\downarrow$	117	$B_{11} \rightarrow B_{10}B_1\rightarrow$	142	$B_{12} \rightarrow B_9B_3\rightarrow$
93	$B_9 \rightarrow B_2B_7\rightarrow$	118	$B_{11} \rightarrow B_{10}B_1\downarrow$	143	$B_{12} \rightarrow B_9B_3\downarrow$
94	$B_9 \rightarrow B_2B_7\downarrow$	119	$B_{11} \rightarrow B\ B_2\rightarrow$	144	$B_{12} \rightarrow B_8B_4\rightarrow$
95	$B_9 \rightarrow B_1B_8\rightarrow$	120	$B_{11} \rightarrow B_9B_2\downarrow$	145	$B_{12} \rightarrow B_8B_4\downarrow$
96	$B_9 \rightarrow B_1B_8\downarrow$	121	$B_{11} \rightarrow B_8B_3\rightarrow$	146	$B_{12} \rightarrow B_7B_5\rightarrow$
97	$B_9 \rightarrow B_1B_8*$	122	$B_{11} \rightarrow B_8B_3\downarrow$	147	$B_{12} \rightarrow B_7B_5\downarrow$
98	$B_{10} \rightarrow B_9B_1\rightarrow$	123	$B_{11} \rightarrow B_7B_4\rightarrow$	148	$B_{12} \rightarrow B_6B_6\rightarrow$
99	$B_{10} \rightarrow B_9B_1\downarrow$	124	$B_{11} \rightarrow B_7B_4\downarrow$	149	$B_{12} \rightarrow B_6B_6\downarrow$
100	$B_{10} \rightarrow B_8B_2\rightarrow$	125	$B_{11} \rightarrow B_6B_5\rightarrow$	150	$B_{12} \rightarrow B_5B_7\rightarrow$

番号	生成規則	番号	生成規則	番号	生成規則
151	$B_{12} \rightarrow B_5 B_7 \downarrow$	176	$B_{13} \rightarrow B_5 B_8 \downarrow$	201	$B_{14} \rightarrow B_6 B_8 \downarrow$
152	$B_{12} \rightarrow B_4 B_8 \rightarrow$	177	$B_{13} \rightarrow B_4 B_9 \rightarrow$	202	$B_{14} \rightarrow B_5 B_9 \rightarrow$
153	$B_{12} \rightarrow B_4 B_8 \downarrow$	178	$B_{13} \rightarrow B_4 B_9 \downarrow$	203	$B_{14} \rightarrow B_5 B_9 \downarrow$
154	$B_{12} \rightarrow B_3 B_9 \rightarrow$	179	$B_{13} \rightarrow B_3 B_{10} \rightarrow$	204	$B_{14} \rightarrow B_4 B_{10} \rightarrow$
155	$B_{12} \rightarrow B_3 B_9 \downarrow$	180	$B_{13} \rightarrow B_3 B_{10} \downarrow$	205	$B_{14} \rightarrow B_4 B_{10} \downarrow$
156	$B_{12} \rightarrow B_2 B_{10} \rightarrow$	181	$B_{13} \rightarrow B_2 B_{11} \rightarrow$	206	$B_{14} \rightarrow B_3 B_{11} \rightarrow$
157	$B_{12} \rightarrow B_2 B_{10} \downarrow$	182	$B_{13} \rightarrow B_2 B_{11} \downarrow$	207	$B_{14} \rightarrow B_3 B_{11} \downarrow$
158	$B_{12} \rightarrow B_1 B_{11} \rightarrow$	183	$B_{13} \rightarrow B_1 B_{12} \rightarrow$	208	$B_{14} \rightarrow B_2 B_{12} \rightarrow$
159	$B_{12} \rightarrow B_1 B_{11} \downarrow$	184	$B_{13} \rightarrow B_1 B_{12} \downarrow$	209	$B_{14} \rightarrow B_2 B_{12} \downarrow$
160	$B_{12} \rightarrow B_1 B_{11} *$	185	$B_{13} \rightarrow B_1 B_{12} *$	210	$B_{14} \rightarrow B_1 B_{13} \rightarrow$
161	$B_{13} \rightarrow B_{12} B_1 \rightarrow$	186	$B_{14} \rightarrow B_{13} B_1 \rightarrow$	211	$B_{14} \rightarrow B_1 B_{13} \downarrow$
162	$B_{13} \rightarrow B_{12} B_1 \downarrow$	187	$B_{14} \rightarrow B_{13} B_1 \downarrow$	212	$B_{14} \rightarrow B_1 B_{13} *$
163	$B_{13} \rightarrow B_{11} B_2 \rightarrow$	188	$B_{14} \rightarrow B_{12} B_2 \rightarrow$	213	$B_{15} \rightarrow B_{14} B_1 \rightarrow$
164	$B_{13} \rightarrow B_{11} B_2 \downarrow$	189	$B_{14} \rightarrow B_{12} B_2 \downarrow$	214	$B_{15} \rightarrow B_{14} B_1 \downarrow$
165	$B_{13} \rightarrow B_{10} B_3 \rightarrow$	190	$B_{14} \rightarrow B_{11} B_3 \rightarrow$	215	$B_{15} \rightarrow B_{13} B_2 \rightarrow$
166	$B_{13} \rightarrow B_{10} B_3 \downarrow$	191	$B_{14} \rightarrow B_{11} B_3 \downarrow$	216	$B_{15} \rightarrow B_{13} B_2 \downarrow$
167	$B_{13} \rightarrow B_9 B_4 \rightarrow$	192	$B_{14} \rightarrow B_{10} B_4 \rightarrow$	217	$B_{15} \rightarrow B_{12} B_3 \rightarrow$
168	$B_{13} \rightarrow B_9 B_4 \downarrow$	193	$B_{14} \rightarrow B_{10} B_4 \downarrow$	218	$B_{15} \rightarrow B_{12} B_3 \downarrow$
169	$B_{13} \rightarrow B_8 B_5 \rightarrow$	194	$B_{14} \rightarrow B_9 B_5 \rightarrow$	219	$B_{15} \rightarrow B_{11} B_4 \rightarrow$
170	$B_{13} \rightarrow B_8 B_5 \downarrow$	195	$B_{14} \rightarrow B_9 B_5 \downarrow$	220	$B_{15} \rightarrow B_{11} B_4 \downarrow$
171	$B_{13} \rightarrow B_7 B_6 \rightarrow$	196	$B_{14} \rightarrow B_8 B_6 \rightarrow$	221	$B_{15} \rightarrow B_{10} B_5 \rightarrow$
172	$B_{13} \rightarrow B_7 B_6 \downarrow$	197	$B_{14} \rightarrow B_8 B_6 \downarrow$	222	$B_{15} \rightarrow B_{10} B_5 \downarrow$
173	$B_{13} \rightarrow B_6 B_7 \rightarrow$	198	$B_{14} \rightarrow B_7 B_7 \rightarrow$	223	$B_{15} \rightarrow B_9 B_6 \rightarrow$
174	$B_{13} \rightarrow B_6 B_7 \downarrow$	199	$B_{14} \rightarrow B_7 B_7 \downarrow$	224	$B_{15} \rightarrow B_9 B_6 \downarrow$
175	$B_{13} \rightarrow B_5 B_8 \rightarrow$	200	$B_{14} \rightarrow B_6 B_8 \rightarrow$	225	$B_{15} \rightarrow B_8 B_7 \rightarrow$

番号	生成規則	番号	生成規則	番号	生成規則
226	$B_{15} \rightarrow B_8B_7\downarrow$	251	$B_{16} \rightarrow B_{11}B_5\downarrow$	276	$B_{17} \rightarrow B_{15}B_2\downarrow$
227	$B_{15} \rightarrow B_7B_8\rightarrow$	252	$B_{16} \rightarrow B_{10}B_6\rightarrow$	277	$B_{17} \rightarrow B_{14}B_3\rightarrow$
228	$B_{15} \rightarrow B_7B_8\downarrow$	253	$B_{16} \rightarrow B_{10}B_6\downarrow$	278	$B_{17} \rightarrow B_{14}B_3\downarrow$
229	$B_{15} \rightarrow B_6B_9\rightarrow$	254	$B_{16} \rightarrow B_9B_{17}\rightarrow$	279	$B_{17} \rightarrow B_{13}B_4\rightarrow$
230	$B_{16} \rightarrow B_6B_9\downarrow$	255	$B_{16} \rightarrow B_9B_7\downarrow$	280	$B_{17} \rightarrow B_{13}B_4\downarrow$
231	$B_{15} \rightarrow B_5B_{10}\rightarrow$	256	$B_{16} \rightarrow B_8B_8\rightarrow$	281	$B_{17} \rightarrow B_{12}B_5\rightarrow$
232	$B_{15} \rightarrow B_5B_{10}\downarrow$	257	$B_{16} \rightarrow B_8B_8\downarrow$	282	$B_{17} \rightarrow B_{12}B_5\downarrow$
233	$B_{15} \rightarrow B_4B_{11}\rightarrow$	258	$B_{16} \rightarrow B_7B_9\rightarrow$	283	$B_{17} \rightarrow B_{11}B_6\rightarrow$
234	$B_{15} \rightarrow B_4B_{11}\downarrow$	259	$B_{16} \rightarrow B_7B_9\downarrow$	284	$B_{17} \rightarrow B_{11}B_6\downarrow$
235	$B_{15} \rightarrow B_3B_{12}\rightarrow$	260	$B_{16} \rightarrow B_6B_{10}\rightarrow$	285	$B_{17} \rightarrow B_{10}B_7\rightarrow$
236	$B_{15} \rightarrow B_3B_{12}\downarrow$	261	$B_{16} \rightarrow B_6B_{10}\downarrow$	286	$B_{17} \rightarrow B_{10}B_7\downarrow$
237	$B_{15} \rightarrow B_2B_{13}\rightarrow$	262	$B_{16} \rightarrow B_5B_{11}\rightarrow$	287	$B_{17} \rightarrow B_9B_8\rightarrow$
238	$B_{15} \rightarrow B_2B_{13}\downarrow$	263	$B_{16} \rightarrow B_5B_{11}\downarrow$	288	$B_{17} \rightarrow B_9B_8\downarrow$
239	$B_{15} \rightarrow B_1B_{14}\rightarrow$	264	$B_{16} \rightarrow B_4B_{12}\rightarrow$	289	$B_{17} \rightarrow B_8B_9\rightarrow$
240	$B_{15} \rightarrow B_1B_{14}\downarrow$	265	$B_{16} \rightarrow B_4B_{12}\downarrow$	290	$B_{17} \rightarrow B_8B_9\downarrow$
241	$B_{15} \rightarrow B_1B_{14}*$	266	$B_{16} \rightarrow B_3B_{13}\rightarrow$	291	$B_{17} \rightarrow B_7B_{10}\rightarrow$
242	$B_{16} \rightarrow B_{15}B_1\rightarrow$	267	$B_{16} \rightarrow B_3B_{13}\downarrow$	292	$B_{17} \rightarrow B_7B_{10}\downarrow$
243	$B_{16} \rightarrow B_{15}B_1\downarrow$	268	$B_{16} \rightarrow B_2B_{14}\rightarrow$	293	$B_{17} \rightarrow B_6B_{11}\rightarrow$
244	$B_{16} \rightarrow B_{14}B_2\rightarrow$	269	$B_{16} \rightarrow B_2B_{14}\downarrow$	294	$B_{17} \rightarrow B_6B_{11}\downarrow$
245	$B_{16} \rightarrow B_{14}B_2\downarrow$	270	$B_{16} \rightarrow B_1B_{15}\rightarrow$	295	$B_{17} \rightarrow B_5B_{12}\rightarrow$
246	$B_{16} \rightarrow B_{13}B_3\rightarrow$	271	$B_{16} \rightarrow B_1B_{15}\downarrow$	296	$B_{17} \rightarrow B_5B_{12}\downarrow$
247	$B_{16} \rightarrow B_{13}B_3\downarrow$	272	$B_{16} \rightarrow B_1B_{15}*$	297	$B_{17} \rightarrow B_4B_{13}\rightarrow$
248	$B_{16} \rightarrow B_{12}B_4\rightarrow$	273	$B_{17} \rightarrow B_{16}B_1\rightarrow$	298	$B_{17} \rightarrow B_4B_{13}\downarrow$
249	$B_{16} \rightarrow B_{12}B_4\downarrow$	274	$B_{17} \rightarrow B_{16}B_1\downarrow$	299	$B_{17} \rightarrow B_3B_{14}\rightarrow$
250	$B_{16} \rightarrow B_{11}B_5\rightarrow$	275	$B_{17} \rightarrow B_{15}B_2\rightarrow$	300	$B_{17} \rightarrow B_3B_{14}\downarrow$

番号	生成規則
301	$B_{17} \rightarrow B_2 B_{15}\rightarrow$
302	$B_{17} \rightarrow B_2 B_{15}\downarrow$
303	$B_{17} \rightarrow B_1 B_{16}\rightarrow$
304	$B_{17} \rightarrow B_1 B_{16}\downarrow$
305	$B_{17} \rightarrow B_1 B_{16}*$

発表文献

(1) 安居院、中嶋、長橋、"印刷漢字の情報の分布を調べるための部分提示による心理実験" 電子通信学会全国大会 1170,

(2) 安居院、中嶋、長橋、"線と領域を主体とした漢字認識について"、電子通信学会全国大会シンポジウム、S10-7 6-341～342.（1977）

(3) 安居院、中嶋、長橋、"線と領域を主体とした漢字認識に関する研究"、電子通信学会、パターン認識と学習研究会、PRL 77-4, P.P.27～36（1977）

(4) 安居院、中嶋、長橋、"部分パターンの位置関係を利用した手書き漢字の表現法"、電子通信学会論文誌 Vol. 60-D, NO. 12, P.P. 1109～1116.（1977）

(5) 安居院、長橋、"漢字パターンの表現法について"、電子通信学会、パターン認識と学習研究会、PRL 77-49、P.1～8（1977）

(6) 安居院、長橋、"部分パターン間の位置関係を利用した漢字パターンの一符号化法"、電子通信学会全国大会 1248,（1978）

(7) 長橋、安居院、"手書き漢字パターンの符号化について"、電子通信学会論文誌 Vol. J61-D, NO. 11, P.P. 803～810（1978）

(8) T. Agui, H. Nagahashi; "A description method of handprinted Chinese characters", IEEE trans. on PAMI, Vol. PAMI-1, NO.1 Jan. P.P. 20～24（1979）

(9) 安居院、長橋、"特徴点抽出処理段階における漢字と平仮名文字の特徴の検討"、電子通信学会総合全国大会 1313,（1979.）

(10) 安居院、長橋、"幾何学的特徴量による手書き漢字と平仮名文字の分類について"、電子通信学会情報・システム部門全国大会 155（1979）

(11) T. Agui, H. Nagahashi, "A coding method of handprinted Chinese characters", IEEE. trans. on PAMI, Vol. PAMI-1, NO.4, Oct. P.P. 333～341（1979）