論 文

論文 77-14-5-3

Syntactic法에 의한 한글의 패턴 認識에 關한 硏究

(A Study on the Pattern Recognition of Korean Characters by Syntactic Method)

金 太 均*, 安 居 院 猛**
(Kim, Tae Kyun and Agui, Takeshi)

要 約

本 論文은 Syntactic法에 의한 한글의 認識 algorithm에 對하여 論한 것으로 認識節次는 크게 나누어 前處理, graph化, 分割의 3段階로 構成되어 있다. 前處理過程에서는 Hilditch의 thinning algorithm을 利用하여 入力文字의 骨格패턴을 얻었다. graph化 段階에서는 細線化패턴으로 부터 4種類의 特徵點을 抽出하여 入力패턴을 基本패턴의 構成關係로 나타냈다. 分割段에서는 tree文法에 의한 패턴解析을 수행, 入力패턴을 構成하는 各 字母를 順次的으로 抽出하였다. 本 algorithm의 効用性을 檢討하기 위하여 電子計算機를 利用, 511字의 印刷體 한글에 對하여 認識實驗을 行하였다. 그 結果 約 90%의 正認識率을 얻었다.

Abstract

The syntactic pattern recognition system of Korean characters is composed of three main functional parts; Preprocessing, Graph-representation, and Segmentation. In preprocessing routine, the input pattern has been thinned using the Hilditch's thinning algorithm. The graph-representation is the detection of a number of nodes over the input pattern and codification of branches between nodes by 8 directional components. Next, segmentation routine which has been implemented by top down nondeterministic parsing under the control of tree grammar identifies parts of the graph-represented pattern as basic components of Korean characters. The authors have made sure that this system is effective for recognizing Korean characters through the recognition simulations by digital computer.

1. 序 論

現代社會는 情報化社會라고도 일컬어 지고 있다. 왜냐하면 電子計算機를 비롯한 수많은 情報處理 시스템이 開發되어 各方面에 폭넓게 利用되고 있기 때문이다 그러나 電子計算機等은 주변장치의 技術的인 進步가中央演算裝置에 비해 크게 뒤떨어져 特히 人間과 計算機間의 情報交換役割을 담당하는 入力部分은 現在도 주로 人力에 의존하고 있는 實情이다. 이러한 주변 장치의 非能率性은 방대한 데이터의 迅速, 正確한 處理의 가장 큰 장애가 되고 있어 시급한 機械化가 要求되고 있다. 이 때문에 文字, 圖形의 패턴 認識에 관한 硏究가 活發하게 進行되어 現在特定分野에 있어서는 實用化段階에까지 이르게 되었다. 이러한 추세에 따라 한글 패턴認識시스템의 開發에 관한 硏究도 進行되어 그간 상당한 硏究結果가 發表되었다.[1] [2] [3] [4] 그러나 한글은 認識對象이 되는 文字數가 방대할 뿐아니라 類似文字가 多數存在하는 等 한글 特有의 難點 때문에 풀어 쓰기 한글의 認識에 對한 硏究가 大部分이었다.

풀어쓰기 한글의 認識은 認識對象을 字母에 局限시키기 때문에 比較的 簡單하고 有効한 認識 algorithm의 構成이 可能할 뿐 아니라 認識裝置의 簡略化도 기

* 正會員, 忠南大學校 工業敎育大學
(College of Industrial Education, Chung Nam National Univ.,)
** 非會員, 東京工業大學 工學部 像情報工學硏究 施設

接受日字 : 1977年 9月 24日

할 수 있다. 그러나 이러한 認識 시스템은 現在 거의 使用되지 않는 풀어쓰기 한글을 認識對象으로 하기 때문에 그 効用性이 問題視되며 또한 認識結果인 各 字母로부터 入力文字를 다시 合成해야 하는 難題를 수반하고 있다. 그러므로 궁극적으로는 모아쓰기 한글을 對象으로 하는 識認시스템의 開發이 바람직하다고 생각된다. 이러한 觀點下에서 筆者는 最近 패턴 認識의 諸分野에 널리 應用되고 있는 syntactic法을 利用하여 部分的인 認識對象은 字母에局限시키면서도 모아쓰기 한글의 認識이 可能한 認識 algorithm을 提案한다. 패턴 認識의 한 手法인 syntactic法은 形式言語와 automaton 理論을 應用한 認識法으로 入力패턴을 우선 基本패턴(primitive pattern) 또는 部分패턴(subpattern)으로 細分化한 後, 이들의 構成關係가 패턴生成文法의 文脈과 일치하는가를 檢討하여 文字 또는 圖形을 認識하는 方法이다. 本 認識시스템에서는 한글字母의 生成過程을 多次元패턴 文法의 一種인tree文法[5]으로 定型化한 다음, 字母를 構成하는 基本패턴의 配列關係와 tree文法의 文脈의 一致如否를 檢討하여 入力패턴으로부터 各 字母를 順次的으로 抽出하였다. 本 시스템의 構成은 그림 1에서 나타낸 바와같이 3段階로 되어있다. 以下 各 節에서 各 段階의 處理過程에 對하여 상세하게 論하기로 한다.

그림 1. 시스템의 構成
Fig.1. Structure of the system.

2. 시스템의 構成과 認識 Algorithm

Ⅰ. 前處理(Preprocessing)

現在 部分的으로 實用化되고 있는 自動文字認識시스템은 少數의 文字만을 對象으로 하는 專用機械이다. 그러나 對象文字數가 많고 또한 文字의 構造가 複雜한 경우에는 시스템이 大規模化되기 때문에 一般的으로非經濟的이다. 그러므로 最近에는 文字認識을 전적으로 電子計算機에 의존하고, 專用機械는 다만 計算機入力에 適當한 形態로 入力文字를 變形시키는 방향으로 硏究가 進行되고 있다. 本 硏究에서는 計算機處理에 適當한 量子化패턴(digitized pattern)을 얻기 위해 vidicon 카메라를 利用하여 標本化(sampling)하였다. 量子化패턴은 $P(i,j)=v$ $(1\leq i\leq m,\ 1\leq j\leq n,\ v\leq v_1,v_2,\cdots\cdots v_k\geq 0)$의 $m\times n$行列로, v는 各, 點의 濃淡度를 나타낸다. 本硏究에서는 文字部分을 $P(i,j)=1$, 背景部分은 $P=(i,j)=0$, $m=n=32$의 2値패턴을 入力패턴으로 使用하였다.

그림. 2 (a) 入力패턴 (b) 細線化패턴
Fig.2. (a) input pattern
(b) thinned pattern.

量子化패턴은 一般的으로 各種의 noise를 포함하기 때문에, 計算機處理를 하기 前에 불필요한 noise를 除去하는 前處理過程을 거친다. 本 硏究에 使用된 入力文字는 良質의 것이었기 때문에 前處理過程에서는 細線化作業만을 行하였다. 細線化는 2値패턴으로부터 그 패턴을 構成하는 骨格線을 抽出하는 處理로서, 패턴認識에 많이 應用되고 있다. 細線化處理에는 Hilditch의 algorithm[6]를 使用하였다. 그림 2에 入力패턴과 細線化패턴의 例를 圖示한다.

Ⅱ. Graph化(Graph Representation)

syntactic패턴 認識시스템에서, 入力패턴을 graph 패턴으로 表現할 必要가 있다. 그러므로 graph패턴을 作成하기 위하여 그림 3에 나타낸 4種類의 特徵點, 卽 端點, 屈曲點, 分岐點, 交差點을 細線化 패턴으로부터 抽出하여 graph 패턴의 節點(node)으로, 節點間을 連結하는 線分을 가지(branch)로 하였으며, 이 가지를 入力패턴을 構成하는 基本패턴으로 定義하였다.

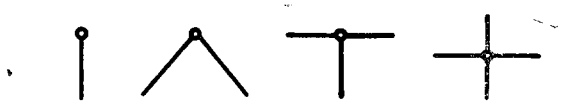


그림 3. 特徵點
Fig. 3. Extracted feature points
(a) end point, (b) break point
(c) touch point, (d) cross point.

graph패턴이 作成되면 다음段인 分割段에서는 tree文法을 適用하여 graph패턴의 構造解析을 하게 되는데 graph패턴을 대신하는 data base로써 graph matrix를 導入하였다. graph matrix를 作成하기 위하여 各節點

에 抽出順으로 番號를 부여함과 동시에 平面座標를 求해, 이를 基準으로 各 節點間의 가지를 8方向의 方向量子로 coding하였다. graph matrix $G(i,j)$, $N\times(N+3)$의 行列로 그 要素는 다음과 같다.

N ; 節點의 總數

n_i ; i番節點

$\phi(1\leqq i\leqq N,\ 1\leqq j\leqq N)$; n_i 와 n_j間에 가지가 存在하지 않을 경우

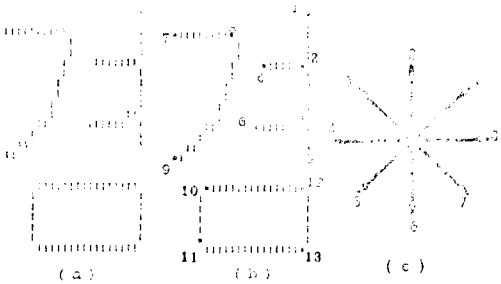
$k,(1\leqq j\leqq N,\ 1\leqq j\leqq N)$; n_i와 n_j間에 가지가 存在하는 경우로 n_i로 부터 n_j에 向하는 方向을 方向量子에 따라 coding한다. $G(i,j)=k$의 경우 $G(j,i)=[k+4]$ [mod.8] $k\in\{0,1,2,\cdots\cdots7\}$이다.

$x(1\leqq i\leqq N,\ j=N+1)$; $1\leqq x\leqq 32$, n_i의 水平座標.

$y(1\leqq i\leqq N,\ j=N+2)$; $1\leqq y\leqq 32$로 n_j의 垂直座標.

$t(1\leqq i\leqq N,\ j=N+3)$; $t\in\{1,2,3,4\}$로 n_i의 形態를 나타낸다. 端點은 1, 屈曲點 2, 分岐點 3, 交差點 4이다.

graph化 以後의 諸般處理過程에서 graph 패턴대신 graph matrix를 使用하므로서 簡單高速의 處理가 可能



NODE NO.	1	2	3	4	5	6	7	8	9	10	11	12	13	x	y	t
1		6												26	4	1
2	2		6	4										27	9	3
3		2			6	4								25	14	3
4		0												19	9	1
5			2											25	27	1
6			0											18	15	1
7							0							5	6	1
8						4		5						14	6	2
9							1							5	16	1
10										6	6			10	21	2
11									2			0		9	32	2
12									4			6		25	21	2
13										4	2			25	27	2

(d)

그림 4 Graph matrix의 作成例

(a) 細線化 패턴, (b) graph 패턴

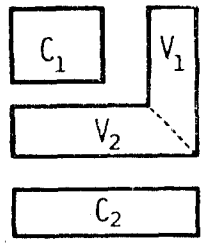
(c) 8方向量子, (d) graph matrix

Fig.4. Example of a graph matrix.

(a) thinned pattern, (b) graph pattern

(c) 8 directional components

(d) graph matrix.



C_1, C_2 全 子 音

V_1; ㅏ, ㅑ, ㅓ, ㅕ, ㅣ
ㅐ, ㅒ, ㅔ, ㅖ

V_2; ㅗ, ㅛ, ㅜ, ㅠ, ㅡ

그림 5 字母의 配列原型

Fig.5. Two dimensional arrangement of each character units.

하였다. 그림 4에 graph패턴 및 graph matrix의 作成例를 圖示한다.

Ⅲ. 分割 Algorithm

分割段에서는 tree文法에 의한 패턴解析을 수행, 入力패턴을 構成하는 各 字母를 graph matrix로 부터 順次的으로 抽出해 낸다. 한글은 最小 2, 最大 7字의 字母로 構成되어 있다. 그러나 認識節次의 簡略化 및 統一을 期하기 위하여 各 字母가 그림 5에 나타낸 原型안에 配列되는 것으로 간주하였다. 때문에, "ㄲ", "ㅖ" 等의 合成字母도 하나의 獨立된 字母로 취급하였다. 그림 5의 C_1에 位置하는 子音의 集合을 편의상 UL (Upper-Left), V_1에 오는 母音의 集合을 UR(Upper-Right), V_2의 集合을 MD(Middle), C_2의 集合을 LW (Lower)로 부르기로 한다. 分割段은 다음과 같은 順序로 進行된다.

step 1 ; UL 및 UR의 抽出을 위한 導出開始節點 (derivation starting node) N_{C1}과 N_{V1}을 決定한다. N개의 節點中 最上左端의 것을 N_{C1}, 最上右端의 것을 N_{V1}으로 擇하였다.

step 2 ; N_{C1}의 形과 N_{C1}에 接觸된 가지의 方向成分等을 利用하여 multilevel decision tree를 作成. 이를 토대로 UL을 다음의 7개 group으로 分類하였다.

group 1=(ㄹ, ㅉ, ㄲ, ㅈ, ㅋ, ㄱ)

group 2=(ㅎ, ㅊ)

group 3=(ㅍ, ㅃ, ㅂ)

group 4=(ㅁ, ㅌ, ㄷ)

group 5=(ㅆ, ㅅ)

group 6=(ㅇ, ㅁ)

group 7=(ㄴ)

step 3 ; group가 決定되면 그 group에 屬하는 文字의 tree文法을 適用하여 N_{C1}으로 부터 導出을 開始한다. 文脈대로 導出이 完了된 文法이 있을 때에는, 그 文法이 生成하는 文字를 UL의 認識結果로서 등록한 後, 抽出된 文字를 構成하는 節點과 가지를 graph matrix로부터 削除한다. 所屬全文法을 適用하여도 導出에 失

敗하였을 경우에는 reject한다.

step 4; N_{V1}이 削除되었는지 그 如否를 檢討한다. 削除되었을 경우 N_{V1}은 이미 抽出된 UL에 포함되는 節點이므로 入力패턴은 母音下置形임을 알 수 있다. 이때에는 MD抽出을 위한 導出開始節點 N_{V2}를 決定한다. N_{V2}는 남은 節點中 最左上端의 것을 擇한다. N_{V1}이 削除되지 않았으면 入力패턴은 母音右置形이다.

step 5; 母音右置形의 경우 N_{V1}의 形態 및 N_{V1}에 接觸된 가지의 方向成分等에 의하여 UR을 다음의 2 group으로 分類한 後, step 3과 同一한 節次를 취한다.

group 8=(ㅐ, ㅒ, ㅖ, ㅑ, ㅘ, ㅓ)

group 9=(ㅔ, ㅚ, ㅣ)

母音下置形의 경우, N_{V2}로 부터 step3과 同一한 節次로 MD를 抽出한다. MD에 屬하는 字母는 다음과 같다.

group 10=(ㅠ, ㅛ, ㅗ, ㅜ, ㅡ)

step 6; $G(i,j)$의 要素가 全部 削除되었는지 그 如否를 檢討한다. $G(i,j)=\phi$이면 抽出된 2字母(UL,UR 혹은 UL, MD)를 認識結果로 各各 output한 後, 認識節次를 終了한다. $G(i,j)\neq\phi$면 入力패턴은 LW를 포함하므로 $G(i,j)=\phi$가 될 때까지 UL의 抽出과 同一한節次로 LW를 抽出한다.

以上 說明한 分割 algorithm에 따른 各字母의 抽出例를 그림 6에 圖示하였다.

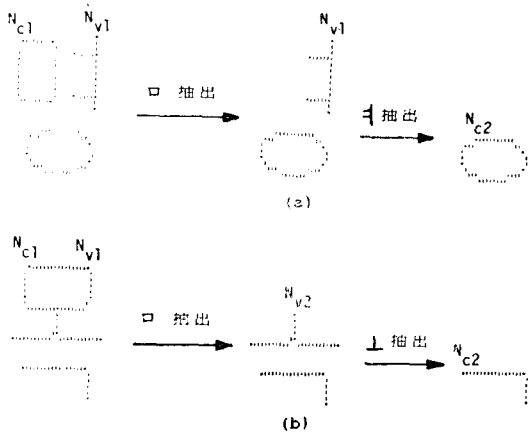


그림 6 各 字母의 抽出過程

(a) 母音右置形 한글, (b) 母音下置形 한글

Fig. 6. Extraction process of each character units

(a) vowel positioned right

(b) vowel positioned lower.

그림 6(b)와 같이 入力패턴을 構成하는 各 字母가 서로 接觸되어 있을 때에는 誤認識 또는 reject가 發生할 수 있으므로 패턴解析에 top down nondeterministic parsing法을 導入하였다. 이것은 parsing의 效率을 희생시키는 대신, 發生이 예상되는 stroke의 變化를 tree 文法에 반영, 試行錯誤的으로 導出을 행하므로서 noise pattern도 認識이 可能토록 한 것이다.

Ⅳ. Tree 文法

自然言語의 model로서 提案된 形式言語(formal language)와 automaton理論이 最近, 패턴認識의 諸分野에 應用되어 그 有效性이 立證되었다[7]. 그러나 이것은 一次元記號列 또는 그 有限集合에 對하여 確立된 理論으로 패턴認識에 應用하기 위해서는 多次元 構成으로 된 패턴構造를 基本패턴의 1次元配列로 나타내야 한다. 이 方法은 이미 確立된 automaton 理論을 直接패턴認識分野에 應用可能하기 때문에 有利한 경우도 있다. 반면에, 이것은 基本패턴間에 一次元的 關係만을 適用하여 入力 패턴을 記述해야하는 難點 때문에 表現可能한 패턴이 制限되고 또 패턴 解析도 複雜해져 패턴 認識의 見地에서 볼 때 有效하다고는 할 수 없다. 그러므로 多次元 패턴의 記述에 편리하고 또한 패턴解析도 비교적 簡單한 tree文法을 導入하여 한글의 各 字母를 記述하였다. 正規 tree文法 G는 다음의 4要素로 構成되어있다.

$G=(V,r,P,S)$

여기서 $V=V_N\cup V_T$, $V_N\cap V_T=\phi$

V_N=非終端設號(nonterminal)의 有限集合.

V_T=終端記號(terminal)의 有限集合.

P=生成規則(production rule)으로 $\Phi\rightarrow\Psi$이다. 여기서 $\Phi\in V_N$이고 Ψ는 V_T에 關한 tree이다.

$r=V_T$의 rank

S=出發記號로 $S\in V_N$이다.

tree는 下向線分에 의해 結合되어 있는 有限個節點의 集合으로 各 節點은 V중 어느 하나에 對應되며 이것을 그 節點의 label이라고 칭한다. tree에는 반드시 下向線分이 流入하지 않는 節點이 오직 하나가 存在하는데 그 節點을 root라고 부른다. root 以外의 各 節點에는 하나의 下向線分이 流入하며 tree 中의 어떠한 節點도 root로 부터 그 節點에 이르는 下向線分列이 存在한다. 또한 label$x$의 節點으로 부터 流出하는 下向線分의 數를 x의 rank라 하며 $r(x)$로 나타낸다. $r(x)=0$인 節點을 tree의 frontier라 하며 root로 부터 發展한 tree는 frontier에서 終結한다.

Ⅴ. Tree 文法에 의한 한글의 記述

Tree文法으로 한글의 生成過程을 定型化하기 위해서

는 우선 文字를 構成하는 基本 패턴의 種類를 選定하여야 한다. 基本패턴으로 複雜한 構造의 것을 擇하면 패턴解析의 節次는 簡單해지나 入力패턴으로부터 複雜한 構造의 基本패턴을 抽出해 내기는 困難하다. 特히 noise를 많이 포함하고 있는 경우 抽出失敗率이 增加하여 誤認識의 原因이 된다. 한편, 簡單한 構造의 패턴을 基本패턴으로 擇하면 抽出失敗率은 減少되나 패턴解析節次가 複雜해진다. 이와같이 相反되는 性質 때문에 基本패턴의 選定은 入力패턴의 構造的 特徵 및 시스템의 效率等을 고려하여 신중히 決定해야 한다.

한글은 各 字母의 構造가 비교적 簡單할 뿐아니라 주로 線形 패턴으로 構成되어 있기 때문에 基本패턴으로서 graph패턴의 가지 卽 8方向線分과 한개의 circle을 選定하였다. tree의 性質上, tree에 의해 記述되는 패턴의 基本 패턴은 二次元的으로 結合되어 있어야 한다. 그러나 合成字母는 2字母의 合成으로 生成되기 때문에 위에서, 選定한 9種類의 基本패턴만으로는 tree 記述이 不可能한 경우가 있다. 이를 解決하기 위하여 擬似連結을 意味하는 3개의 operator를 導入, 基本패턴의 一種으로 使用하였다. 以上 定義한 基本 패턴을 그림 7에 보인다. 그림에서 *jl*, *jr*, *jd*는 →部分에 stroke

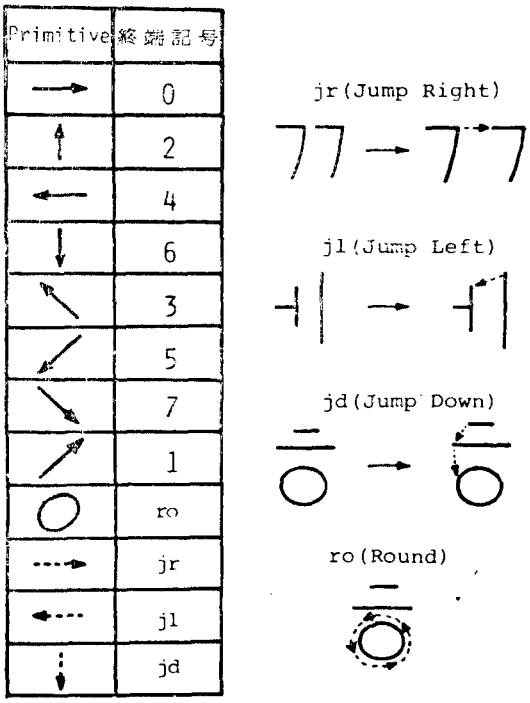


그림 7 한글의 基本패턴

Fig. 7. Primitive pattern for describing the Korean characters.

가 存在한다고 가정하는 擬似連結로 2개 以上의 獨立 패턴으로 構成된 패턴을 하나의 패턴으로 간주한다. 그림 8에 "ㅂ"의 tree文法 및 그로부터 生成되는 tree를 圖示하였다. 그림에서 $는 패턴 記述의 出發點을 나타내는 終端記號로 tree의 root를 나타내는 label이다.

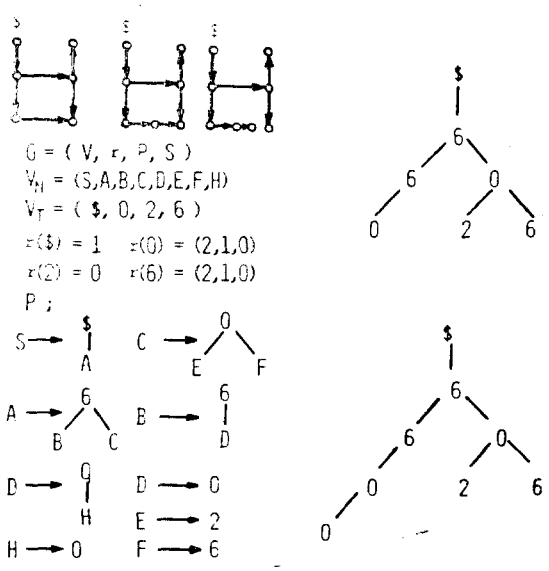


그림 8 Tree 文法에 의한 한글의 記述例

Fig. 8. A description example of a character unit by tree grammar.

Ⅵ. 패턴解析 Algorithm

패턴解析은 graph matrix의 데이터를 使用하여 tree 文法이 生成하는 tree를 直接 作成하기 위한 導出을 行하는 作業이다. 導出에 適用되는 文法은 step 2에서 決定된 任意의 group에 屬하는 n개 ($n \leqq 6$)의 tree 文法으로 文法 $G_k(k=1,2,,n)$에 의해 生成되는 tree를 $T(G_k)$, 또한 $G(i,j)$로 부터 文法 G_k의 文脈대로 $T(Gk)$가 作成된 것을 $G(i,j)G_kT(Gk)$로 나타내기로 한다. $G(i,j)G_kT(Gk)$가 成立되는 k를 求하기 위하여 패턴 解析時 top down nondeterministic parsing을 行하였다.

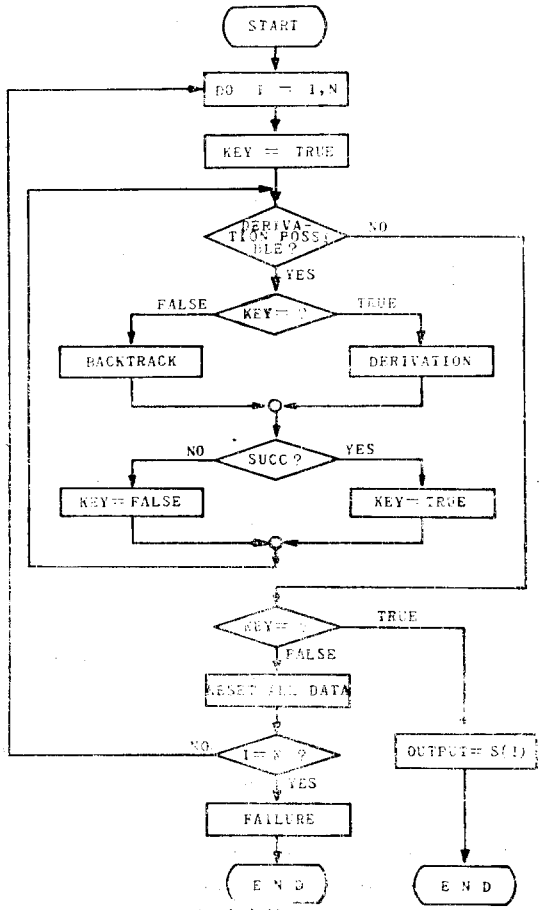
이것은 生成節次와 backtracking 節次로 構成된 algorithm으로 生成節次는 出發記號 S로 부터 시작하여 tree文法의 各 非終端記號를 生成規則에 따라 tree로 置換하므로서 tree文法이 生成하는 tree를 graph matrix로 부터 直接 作成하는 節次이다. 만약 tree로 置換하는데 失敗하였을 때에는 適用可能한 다른 生成規則으로 導出을 行하는 backtracking 節次에 들어 간다. 適用可能한 生成規則을 全部使用하여 backtraking 해도 tree文法이 生成하는 tree의 作成에 失敗하였을때

에는 同一 group에 屬하는 次順의 文法을 適用하여 導出을 行한다. 그림 9의 flow chart는 導出過程의 algorithm을 나타낸다. 한글의 字母中에는 複雜한 構造을 가진 字母의 部分 패턴으로 構成된 字母가 있다. 例를 들면 "ㅐ", "ㅣ"는 "ㅒ"의 部分패턴이다. 그러므로 ㅒ가 入力되었을 때 "ㅐ" 또는 "ㅣ"로서 誤認識될 수 있다. 이러한 誤認識을 방지하고 또한 認識効率을 向上시키기 위해 同一 group所屬文字間에 다음의 두가지基準에 의거, 導出優先關係를 두었다.

(1) 導出長(derivation length)이 긴 文字가 優先한다.

(2) 사용빈도가 많은 文字가 優先한다. 여기서 導出長은 다음과 같이 定義한다.

卽. 패턴 A의 生成 tree $T(G_A)$가 文法$G_A=(V,r,P,S)$에 의해$S\Rightarrow W_1\Rightarrow W_2\Rightarrow\cdots\cdots\Rightarrow W_n\Rightarrow=T(G_A)$의 順으로



N; group 內의 文字數

S(I); I番째 文字

그림 9. 導出過程의 flow chart

Fig. 9. Flow chart of the derivation process.

記號化한 後 Parsing時에 簡單히 引用될 수 있도록 計生成되었을 때 $T(G_A)$의 導出長은 $|T(G_A)|=n$이다. 앞에서 定한 各 group의 文字는 위의 두가지 基準에 의거하여 配列한 것이다.

3. 電子計算機에의한認識實驗結果

本 認識 algorithem의 効用性을 檢討하기 위하여 FORTRAN 프로그램을 作成, FACOM-230 形計算機를 使用하여 印刷體 한글의 認識實驗을 行하였다. 實驗에 使用된 tree文法의 總數는 32個, 文法에 포함된 生成規則은 全部 182個였다. 生成規則은 1個당 8 byte로

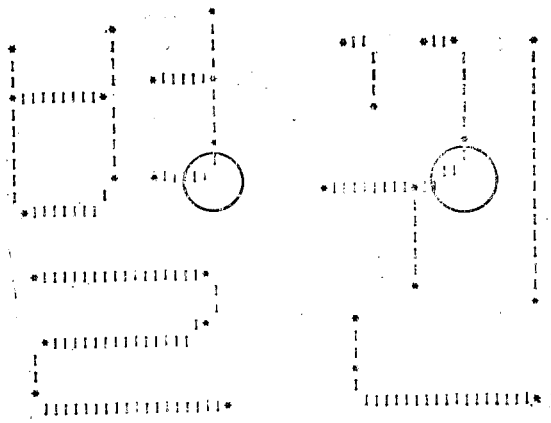


그림 10 短 stroke의 切除例

Fig.10. Example of cut off stroke.

正 認 識	462字	90%
Reject	41字	8%
誤 認 識	8字	2%
計	551字	

表 1. 認識實驗結果

Table 1. Result of the recognition simuatilons.

誤認識文字	構成字母	抽出字母
건	ㄱ, ㅓ, ㄴ	ㄲ, ㅣ, ㄴ
국	ㄱ, ㅜ, ㄱ	ㄹ
든	ㄴ, ㅡ, ㄴ	ㄴ, ㅜ

表 2. 誤認識文字例

Table 2. Examples of miss-recognized characters.

諸 段 階	CPU時間
前 處 理	1-1.5 sec
Graph化	400m sec
分 割	540m sec

表 3 1字당 平均所要時間 CPU時間
Table 3. Average CPU time of each routine for recognizing one character.

記號化한 parsing時에 簡單히 引用될 수 있도록 計算機의 主메모리에 收容하였다. 認識實驗에 使用된 文字는 總 511字로 現在 通用되고 있는 한글의 總數에 비하면 극히 一部分이나 algorithm의 效用性을 檢討하는 데는 충분하였다. 認識實驗의 簡單化를 期하기 위해今回의 實驗에서는 ㅐ, ㅒ等의 合成子音 및 ㅚ, ㅟ 等의 複母音을 포함하는 文字는 認識對象에서 除外하였다. 實驗結果는 表 1에 보인 바와 같다. reject된 文字의 大部分은 細線化패턴의 質이 양호하지 못한 때문으로 特히 短 stroke는 그림 10에 보인 바와 같이 切除되는 경우가 많았다. 誤認識된 文字는 字母間 接觸으로 因한 것이 大部分으로 例를 들면 "국"의 경우 "ㄱ"보다 優先하는 "ㄹ"이 UL의 認識結果로 抽出되었다.

이러한 誤認識은 導出條件을 規定하는 operator를定義하면 거의 修正可能할 것으로 생각된다. 表 2는 誤認識文字의 例, 表3은 各 段階의 實行에 所要된 CPU時間을 나타낸다.

4. 結 論

本 論文은 syntactic法에 의한 한글 認識 algorithm에 대하여 論한 것으로 字母의 生成過程을 多次元패턴文法의 一種인 tree文法으로 定型化한 後, 入力 패턴의 構造가 tree文法의 文脈과 一致하는가를 檢討하여 文字를 認識하였다. 패턴文法으로서 tree文法을 使用한 이유는 tree시스템에 對하여 明確한 理論이 確立되어있다는 點과, tree automaton의 構成이 비교적 簡單한 때문이었다. 本 認識시스템은 top down nondeterministic parsing法을 擇하고 있으므로, 筆記體文字와 같이 noise를 多量 포함하는 패턴을 認識하기 위해서는 發生可能한 stroke의 變化를 全部 tree文字의 生成規則에 반영시켜야 하기 때문에 生成規則의 數가 크게 增加하여 parsing에 必要한 導出回數가 기하급수적으로 增加하는 경향이 있다. 그러나 各 字母의 構造가 비교적 簡單할 뿐 아니라 主로 線形패턴으로 構成된 한글의 認識에는 有效한 認識 시스템임을 電子計算機를 利用한 認識實驗을 通하여 確認하였다. 本 認識 algorithm은 한글과 같이 主로 線形패턴으로 構成된 漢字等의 認識에도 應用可能하리라 생각된다.

參 考 文 獻

(1) 李柱根 ; 한글文字의 computer 조직에 적응하기 위한 特徵抽出에 關한 硏究(Ⅰ) 1969,12. 電子學會誌 Vol.6. No.4. p.198~209

(2) 李柱根 ; 한글文字의 認識에 關한 硏究(Ⅱ). 電子工學會誌, 1970,12. Vol.7. No.3. p.130~136

(3) 姜麟求, 李幸世 ; 한글字體의 特徵抽出의 한 方法. 電子工學會誌. 1969,9. Vol.6. No.2. p.1~5

(4) 이주근, 김홍기 ; 위상회전에 의한 필기체 한글의 자동인식, 전자공학회지, 1976,3. Vol. 13. No.1. p.23~30

(5) K,S,Fu and B, K, Bhargava; Tree system for syntactic pattern recognition, IEEE Trans. comput, Vol. C-22, No.12, pp.1087~1099, Dec 1973.

(6) C,J, Hilditch; Linear skeleton from square cupboard, Machine Intelligence IV, 1966.

(7) K,S,Fu; Syntactic methods in pattern recognition, A. P. Press, New York, 1974.

(8) A,C, shaw; A parsing of graph-representable. pictures, J.A.CM, Vol.17, No.3, pp.453~481, July 1970.

(9) Agui, T and Kim, T,K; Pattern recognition of Korean characters by syntactic method, I.E C.E.J, P77-16, pp.11~20, June 1977.