

Stroke 組合에 의한 筆記體 한글의 표현과 인식

(The Representation and Recognition of Hand-written Hangeul by Stroke Assembly)

金太均*, 安居院猛**, 中嶋正之**

(Taekyun Kim, Takeshi Agui and Masayuki Nakajima)

要 約

본 논문은 문맥자유 속성문법을 적용하여 필기체 한글을 표현하는 방법과, 이를 이용한 2차원 패턴의 구문 해석을 통하여 각 자모를 인식하는 방법에 대하여 논한 것이다. 문법설계를 단순화시키고 생성규칙의 수를 줄이기 위하여 모든 문자는 8개의 stroke 그룹의 조합으로 나타낸다. 구문 해석을 위하여 우선 입력된 문자를 그래프 패턴으로 나타내고 그 다음에는 문법에 의한 도출에 따라 그래프 패턴 위를 이동하는 pointer를 설정하여, 입력문자를 구성하는 각 자모를 순차적으로 추출한다.

본 알고리즘을 사용하여 세사람에 의해 쓰여진 총 384자를 대상으로 인식실험을 수행한 결과 평균 93%의 정 인식을 얻었다.

Abstract

In this paper, it is presented a procedure to recognize hand-written Korean characters by syntax analysis to the graph pattern using the context-free attributed grammars.

Using this algorithm recognition tests have been made for the 384 characters written by three persons, and have obtained 93% of correct recognition rate in average.

I. 서 론

현대사회에서 각종 정보의 발생, 전달의 양은 해가 갈수록 대폭적으로 증가하고 있어, 이것을 인력으로 처리하는 것은 불가능에 가까운 상태가 되고 있다. 더구나 이러한 정보의 대부분은 숫자, 문자, 도형 등으로 이러한 부류의 정보형태를 기계적으로 고속, 정확

하게, 처리, 전달할 수 있는 방법의 확립이 요망되고 있다. 이러한 필요성에 따라 문자의 자동 인식에 관한 연구가 활발히 진행되어 왔는데 그 중에 하나로 한글의 자동인식에 관한 연구도 그간 다수의 연구자에 의해 수행되어 왔다. 이러한 연구는 크게 인쇄체 문자 및 약간의 제한을 가한 필기체 문자의 off-line 인식¹⁻³⁾과 tablet로부터 입력되는 문자의 on-line 인식⁴⁻⁶⁾으로 나눌 수 있다.

한글은 문자수가 극히 많음에도 불구하고 문자구성의 기본 요소가 되는 자모의 수는 24자에 불과하고 또한 자모 및 문자의 생성과정이 계층구조로 되어 있는 점에 착안, 필자는 입력 문자에 대한 구조해석에 의해 각 자모를 순차적으로 추출해내는 방법을 제안하였으며, 구조해석의 수단으로는 구문해석법을 도입하였다.¹⁾ 구문해석적 수법에 의한 문자인식에 있어서, 가장 중

*正會員, 忠南大學校 電子計算機工學科

(Dept. of Computer Eng., Chungnam Univ.)

**非會員, 日本東京工業大學工學部 像情報工學研究施設

(Imaging Science & Eng. Lab., Tokyo Institute of Technology)

接受日字: 1987年 7月 30日

(※ 본 연구는 85년도 전반기 한국과학재단 연구비 지원에 의하여 수행된 연구입니다.)

요한 요소는 인식대상인 문자의 효과적 기술방법에 관한 문제이다. 이 문제를 해결하기 위해 off-line 인식의 경우는 다차원 패턴문법인 tree grammar를 사용하였고^[1] on-line 인식의 경우는 string문법인 PDL^[2] 및 속성문법을 사용하였다.^[3]

그러나, 이들 시스템은 필기체 문자와 같이 변형이 심하거나 또는 각 자모의 상호접촉에 따른 stroke 변형을 내포하고 있는 문자를 기술할 때에는 이러한 모든 변형을 일일이 생성규칙으로 표현해야 하기 때문에 생성규칙의 수가 크게 증가하여 문법의 설계가 복잡해지고 인식율도 떨어지는 단점이 있었다. 이와같이 구문해석적 수법을 도입하는 시스템의 단점을 개선하기 위해 문헌^[4]에서는, stroke의 진행방향 이외에 node의 분기도를 고려하고 있으나 stroke간의 상대위치를 분석하지 않고 있기 때문에 필기체문자와 같이 변형이 심한 문자의 인식에는 한계가 있으며, 문헌^[5]에서는 indexwindow를 도입하여 자음과 모음을 분할하였으나 모음판정시에 중성에 해당하는 자음의 일부가 모음으로 분할되는 사례가 있었다.

본 논문은 구문해석적 방법을 도입하여 필기체 문자를 off-line 인식하는데 있어서 위에 설명한 단점을 개선하기 위하여 다음의 3가지 사항을 제안한다.

첫째, 모든 자모는 속성문법^[6]을 사용하여 8개의 stroke group의 조합으로 나타낸다. 각 그룹은 하나의 대표 stroke와 이 대표 stroke가 자모간 접촉 등에 의해 변형이 된 stroke들로 구성되어 있다. 이와같이 모든 자모를 stroke 그룹의 조합으로만 나타내기 때문에 변형된 문자도 쉽게 생성규칙에 의해 표현이 가능하다.

둘째, 패턴문법으로서 문헌^[4]에서 사용한 속성문법을 2차원 그래프 패턴의 표현에 적합하도록 확장하여 도입한다. 이는 변형이 심한 문자의 인식을 위해서는 구문규칙에 의한 stroke의 진행방향 이외에 stroke의 조합방법 및 각종 속성을 의미규칙을 사용하여 상세히 기술할 필요가 있기 때문이다.

세째, 속성문법은 string 문법인데 반하여 인식대상은 2차원 패턴이기 때문에 인식대상이 string으로 표현되어야 하는 문헌^[4]의 방법으로는 구문해석이 불가능하다. 이문제를 해결하기 위하여 본 논문에서는 생성규칙이 지정하는 방법에 따라 그래프 패턴 위를 이동하는 pointer를 도입, 생성규칙에 의한 도출을 수행할 때마다 pointer의 출발점을 새로히 구하여 그때 그때 pointer를 움직여 나가는 방법을 도입하였다. 도출이 성공적으로 끝난 단계에서 pointer가 전 그래프 패턴을 추적하였으면 도출을 끝내고, 만약 아직도 추적하지 않은 부분이 하나라도 남아 있으면 다른 생성

규칙을 적용하여 동일한 작업을 반복한다. 이 이외에 입력 패턴으로 부터 특징점의 하나인 굴곡점을 추출하는 새로운 방법으로, on-line 한글인식에 응용되었던 SDTS^[7]를 응용하는 방법을 제안한다.

II. 특징점의 추출에 의한 입력패턴의 그래프화

1. 단점, 분기점, 교점의 추출

패턴문법을 사용하여 구조해석을 할 수 있는 형태로 입력패턴을 나타내기 위해 우선 입력된 패턴을 세션화한 다음, 이 세션화 패턴으로 부터 그림 1에 나타낸 바와같은 4 종류의 특징점을 추출한다.

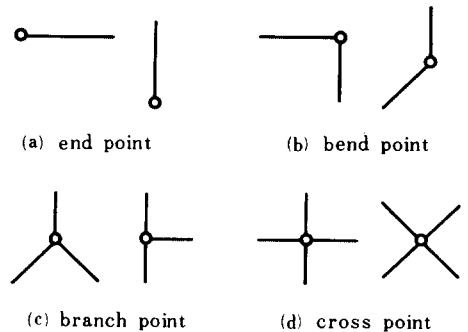


그림 1. 추출된 특징점

Fig. 1. Feature points extracted from pattern.

특징점 추출을 위해 임의의 흑점 P를 중심으로 그림 2에 나타낸 3 * 3의 논리matrix를 설정, 이 matrix내에 있는 흑점의 갯수를 계산한다. 이 값이 0이면 고립점으로 여기서는 잡음으로 간주하여 무시하고, 1이면 단점(end point), 2이면 연결점(connect point), 3이면 분기점(branch point), 그리고 4 이상이면 교점(cross point)에 해당된다. 이와같이 함으로써 단점, 분기점, 교점은 간단히 추출 할 수 있으나 굴곡점(bend point)의 추출은 용이하지 않아 선도형의 굴절각에 의한 방법,^[7] tube형의 sensor에 의한 방법^[8] 등이 제안되어 있다.

m_4	m_3	m_2
m_3	p	m_1
m_4	m_7	m_6

그림 2. 논리 마스크

Fig. 2. Mask using in logical operation.

본 논문에서는 문자선을 구성하는 흑점의 이동방향을 나타내는 방향코드의 다단번역에 의해 연결점 중에서 굴곡점을 추출하는 방법을 제안한다.

2. 굴곡점의 추출

논리 matrix에 의해 추출된 3 종류의 특징점간을 잇는 연결점의 체인을 선이라고 한다면, 입력패턴은 선의 집합으로서 표현 가능하다. 입력패턴을 M라 하고, M을 구성하는 선의 집합을 $\{S\}_M$, 그 요소를 S_i 라 하자. 또한 S_i 를 구성하는 흑점의 집합을 $\{P\}_{s_i}$, 각 점의 진행방향을 나타내는 방향코드의 집합을 $\{C\}_{s_i}$ 로 하면, 각각은 다음과 같이 나타낼 수 있다.

$$\begin{aligned} \{S\}_M &= \{S_1, S_2, \dots, S_i, \dots, S_N\} \\ \{P\}_{s_i} &= \{P_{i1}, P_{i2}, \dots, P_{ij}, \dots, P_{in_i}\} \\ \{C\}_{s_i} &= \{C_{i1}, C_{i2}, \dots, C_{ij}, \dots, C_{in_i}\} \\ C_{ij} &\in \{1, 2, \dots, 8\} \end{aligned}$$

N; 입력문자를 구성하는 선의 총수
 n_i ; S_i 를 구성하는 흑점의 총수

여기서 S_i 의 첫점과 끝점에 해당되는 p_{i1} 과 p_{in_i} 는 이미 추출된 3 종류의 특징점중의 어느 하나이고, 그 이외는 전부 연결 점이다. 또한 $\{C\}_{s_i}$ 는 $\{P\}_{s_i}$ 의 p_{i1} 으로부터 p_{in_i} 까지 각 점을 순차적으로 연결하는 각 선분의 8 방향코드열이다. 또한, p_{ij} 를 방향코드 C_{ij} 의 대응점이라고 부르기로 한다.

S_i 를 구성하는 점들의 이동방향을 나타내고 있는 $\{C\}_{s_i}$ 에는 일반적으로 양자화 화면의 특성 및 세션화 과정에서 발생하는 찌그러짐에 기인하는 잡음성방향코드와 선의 이동방향을 나타내는 데에는 불필요한 방향코드가 다수 포함되어 있다.

본 논문에서는 이러한 잡음과 불필요 성분에 해당되는 연결점을 S_i 로 부터 삭제하여 펜의 이동방향을 나타내는데 필요한 최소한의 점만을 남긴다. 삭제의 방법은 $\{C\}_{s_i}$ 를 구성하는 방향코드열에 대하여 SDTS에 의한 다단번역을 수행하므로써 불필요한 방향코드와 그 대응점들 $\{P\}_{s_i}$ 와 $\{C\}_{s_i}$ 로 부터 순차적으로 제거해 나간다.

번역은 3단에 걸쳐 수행하며, 각 번역단은 특정의 방향코드 열을 추출하여 그것을 다른 방향코드 열로 변환키는 유한변환기(finite transducer)로 자세한 사항은 문헌[5]를 참조하기 바란다.

이와같은 번역과정을 거치면서 $\{P\}_{s_i}$ 로부터 제거되지 않고 최후까지 남은 점의 집합을 $\{P'\}_{s_i}$ 라 하면

$$\{P'\}_{s_i} = \{p'_{i1}, p'_{i2}, \dots, p'_{im_i}, p'_{im_i}\}$$

이다. 여기서 m_i 는 S_i 를 구성하는 특징점의 총수로 $m_i \leq n_i$ 이다. 그리고 $p'_{i1} = p_{i1}$, $p'_{im_i} = p_{in_i}$ 이고 $p'_{i2} \sim p'_{im_i-1}$ 은 모두 굴곡점에 해당된다.

3. 입력패턴의 그래프화

패턴문법에 의한 구문해석을 행하기 위해서는 입력패턴을 기본패턴(primitive)의 조합으로 표현할 필요가 있다. 기본 패턴의 선정은 표현하고자 하는 패턴의 구조에 따라 좌우되는데, 한글의 경우 문자의 성분이 주로 직선이고, 또한 이들 직선성분의 조합형태와 상호위관계가 인식의 중요한 정보원임을 고려하여, 전절에서 추출한 특징점들 간을 잇는 문자선을 직선근사하여 그것을 기본패턴으로 한다. 여기서 각 특징점을 node로 하고, 기본패턴을 branch로 나타내면 입력패턴은 기본패턴의 구성관계로 그래프화 된다. 이와같이 작성된 그래프 패턴은 입력패턴의 구조에 관한 제 정보를 가지고 있으며, 차후의 구문해석 단계에서 입력으로 사용된다. 그림 3에 입력패턴으로 부터 추출된 특징점 및 그로부터 구한 그래프 패턴의 예를 나타낸다.

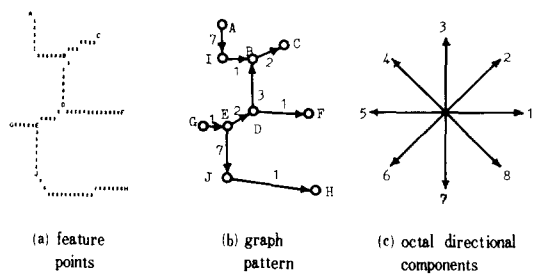


그림 3. 입력 패턴의 그래프화
 Fig. 3. Graph representation of input pattern.

III. 부분패턴 (Subpattern)과 속성 (attributes)

1. 부분패턴

한글은 특정한 구조를 가진 복수개의 부분패턴이 조합되어 하나의 의미를 갖는 문자가 생성된다. 본 논문에서 뜻하는 부분패턴은 전절에서 구한 기본패턴이 1개 이상 결합된 보다 복잡한 구조의 패턴을 말한다.

그림 3(b)의 경우 입력패턴 “논”은 3개의 부분패턴 “ㄴ”, “ㅇ”, “ㄹ”의 집합이고, 각 부분패턴은 각각 3개, 4개, 2개의 기본패턴의 집합으로 구성되어 있음을 알 수 있다.

2. 각종 속성 및 위계관계의 정의

입력패턴 M을 구성하는 부분패턴 중의 어느 하나를 r_1 라 하자. 이 r_1 를 완전히 둘러싸면서, 가로는 X축, 세로는 Y축에 평행인 최소 면적의 직사각형을 부분패턴 r_1 의 영역이라 부르고 $\{r_1\}$ 로 나타낸다. 또한, 이 영역을 X축과 Y축에 투영하여 얻어지는 좌표치의 집

합을 각각 $\{\{r_i\}\}_x, \{\{r_i\}\}_y$, 이 직사각형의 대각선이 서로 교차하는 점의 좌표를 (x_i, y_i) 라 하고 다음의 사항을 정의한다.

[정의1] r_i 영역의 중앙점 (x_i, y_i) 를 부분패턴 r_i 의 대표점이라 부르고 T_i 로 나타낸다. r_i 를 구성하는 첫번째 기본패턴의 시점을 r_i 의 tail, 마지막 기본패턴의 종점을 r_i 의 head라 부르며 각각 $tl(r_i), hd(r_i)$ 와 같이 나타낸다. 또한 $tl(r_i)$ 의 분기도를 r_i 의 시점분기도라 부르고 $D(r_i)$ 와 같이 나타낸다.

[정의2] 부분패턴 r_i 에 대하여 구한 $T_i, \{\{r_i\}\}_x, \{\{r_i\}\}_y, D(r_i)$ 를 각각 속성치라 부른다. 이들 속성치의 집합을 $A(r_i)$ 와 같이 나타내고 r_i 의 속성이라 부른다.

[정의3] 2개의 부분패턴 r_1 와 r_2 에 있어서 r_1 의 $tl(r_1)$ 로부터 r_2 의 $tl(r_2)$ 로 향하는 가상선분의 8방향 코드를 $id(r_1, r_2)$ 로 나타내며, r_1 에 대한 r_2 의 상대위치 속성이라 부른다. 그림 4에 부분패턴 r_1 과 r_2 에 대하여 구한 각 속성치의 예를 나타낸다.

[정의4] 2개의 부분패턴 r_1 와 r_2 가 결합되어 생성되는 보다 복잡한 구조의 부분패턴을 r_{12} , 그 속성을 $A(r_{12})$ 또는 $A(r_{12}, r_1, r_2)$ 라 하며, $A(r_{12})$ 는 r_1 와 r_2 가 동시에 포함되는 영역으로부터 정의 2와 동일한 방법으로 구할 수 있다. 이것을 합성속성이라 부르며 다음과 같이 나타낸다.

$$A(r_{12}) = f(A(r_1), A(r_2)), A(r_1, r_2)$$

여기서 f 는 $A(r_1), A(r_2)$ 를 $A(r_{12})$ 로 사상하는 함수로 구체적으로는 다음에 정의하는 r_1, r_2 간에 존재하는 위치관계이다.

[정의5] 2개의 부분패턴 r_1, r_2 간에 존재하는 위치관계로는 연결관계, 포함관계, 우치(右置)관계, 좌치(左置)관계, 상치(上置)관계, 하치(下置)관계, 근접관계등 7종류가 있다. 여기서 연결, 포함, 우치, 하치 근접관계의 정의는 문헌[6]과 동일하다.

[보제1] r_1 의 대표점 T_i 로부터 r_2 의 대표점 T_j 로 향하는 가상선분의 방향코드가 5이고, 또한 $x_j < \min \{\{r_1\}\}_x$ 가 성립할 때 r_2 는 r_1 의 좌측에 있고, 좌치관계가 존재한다.

[보제2] r_1 의 대표점 T_i 로부터 r_2 의 대표점 T_j 로 향하는 가상선분의 방향코드가 3이고, 또한 $y_j < \min \{\{r_1\}\}_y$ 가 성립할 때, r_2 는 r_1 의 위쪽에 있고, 상치관계가 존재한다.

위와 같이 7종류의 위치관계가 존재하는 2개의 부분패턴의 속성 $A(r_1), A(r_2)$ 로부터 $A(r_{12})$ 로 사상하

는 함수를 각각 $f_c, f_l, f_r, f_u, f_b, f_n$ 라 한다. 그림 4의 경우 부분패턴 r_1 과 r_2 간에는 하치관계, r_1 을 구성하는 기본패턴 “ㅇ”과 “o-o”간에는 연결관계, r_2 를 구성하는 부분패턴 “o-o-o”과 기본패턴 “ㅇ”간에는 상치관계가 존재한다.

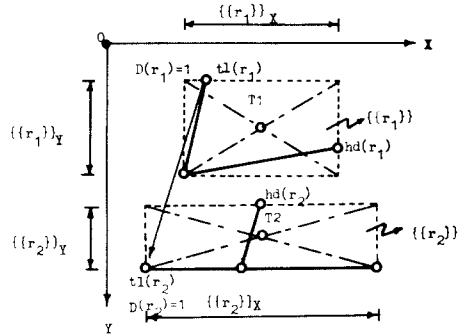


그림 4. 부분 패턴의 속성
Fig. 4. Attributes of subpattern.

IV. 한글 표현 속성문법의 설계

어느 한 문자를 몇개의 부분패턴의 조합에 의해 구성되었다고 생각하였을 경우 그 문자는 부분패턴들의 계층구조로 표현할 수 있다. 한글의 경우는 조립식문자이기 때문에 다른 문자에서는 볼 수 없는 특수한 계층구조로 되어 있다. 따라서, 이 계층구조를 패턴문법으로 기술한다면 한글의 생성과정이 문법으로 명확히 표현되게 된다. 한글의 계층구조는 다수의 샘플패턴을 사용하여 한글의 형태별로 작성한다. 작성된 계층구조는 크게 문자를 구성하는 부분패턴 또는 기본패턴의 종류를 나타내는 부분과 이들 각 패턴들 사이의 위치관계를 나타내는 부분으로 분류된다. 일례로 그림 3에 나타낸 C1·V2·C2형 한글에 대하여 작성한 계층구조를 그림 5에 나타낸다. 그림 5에서 원으로 나타낸 node는 부분패턴을 나타내고, 아래쪽으로 향하는 화살표는 그 부분패턴을 구성하는 또 다른 부분패턴의 종류, 옆쪽으로 향하는 화살표는 두 부분패턴간의 위치관계를 나타낸다.

예를 들면 C1·V2·C2형 한글은 C1(초성자음), V2(중모음), C2(종성자음)으로 구성되는데 C1 아래쪽에 V2가 오고 V2 아래쪽에 C2가 온다. C1 위치에는 “ㄴ”(NU)을 비롯한 각종 자음이 오며 “ㄴ”은 7방향 stroke 그룹 G7과 1방향 stroke 그룹G1이서로 연결되어 생성됨을 나타낸다. 이 stroke 그룹의 계층구조도는 생략하기로 한다.

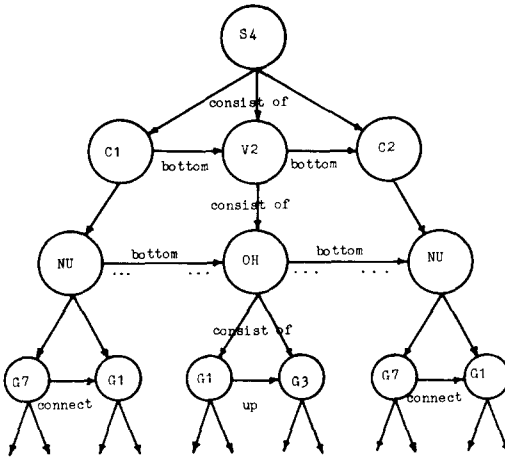


그림 5. 한글의 계층구조
Fig. 5. Hierarchical structure of Korean characters.

이하, 문맥자유속성문법의 정의와 이를 이용하여 그림 5의 계층구조를 기술하는 방법에 대하여 논한다.

[정의6] 문맥자유 속성문법G는 4개 항으로 $G = (VN, VT, P, S)$ 이다.

여기서,

VN; 비종단 기호의 유한집합

VT; 종단기호의 유한집합

$P = (Psy, Pse)$; 생성규칙의 유한집합으로 구문규칙 (Psy)과 의미규칙 (Pse)으로 구성

S; 출발기호로 $S \in VN$

이다. 비종단기호는 부분패턴을 나타내는 기호이고 종단기호는 기본패턴을 나타내는 기호이다. 생성규칙중의 구문규칙은 어느 부분패턴을 구성하는 패턴의 종류를 규정하는 규칙이고, 의미규칙은 이 부분패턴을 구성하는 패턴 사이에 존재하는 위치관계를 규정하는 규칙이다. 구문규칙의 형태는 $X \rightarrow X_1, X_2, \dots, X_i, \dots, X_m$ 으로, $X \in V_N, X_i \in V_N \cup VT$ 이다. 의미규칙의 형태는 $A(X) = f(A(X_1), A(X_2), \dots, A(X_i), \dots, A(X_m))$ 으로 $A(X_i)$ 는 부분패턴 X_i 의 속성이고 $A(X)$ 는 f 의 위치관계가 존재하는 m 개 부분패턴의 조합에 의해 생성되는 보다 복잡한 구조의 부분패턴 X 의 합성 속성을 나타낸다.

그림 5의 계층구조도로부터 속성문법을 작성하는 방법은 다음과 같다. 우선, 계층구조의 각 node에 대하여 label을 할당하여 비종단기호의 집합에 포함시킨다. 특히 정점에 해당하는 node에는 출발기호 S를 부여한다. 임의의 모 node로부터 분기되는 자 node의 종류(아래쪽 화살표)를 표현하는데에는 구문규칙을 사용

하고, 자 node간의 위치관계(옆쪽 화살표)를 표현하는데에는 의미규칙을 사용한다. 각 생성규칙의 좌변은 모node의 기호, 우변은 자 node의 기호를 순차적으로 배열하므로써 간단히 문법의 생성규칙을 작성할 수 있다.

한글은 주로 직선성 stroke로 구성되어 있는데 이러한 stroke가 일정한 규칙에 따라 조합되어 하나의 자모를 구성하고, 이들 자모가 지정된 위치에 복수개 배열되어 하나의 문자가 생성된다. 따라서 이와같은 한글의 구조상의 특징으로 볼 때 stroke의 종류 및 배열 위치등은 한글의 자동인식 측면에서 아주 중요한 정보가 된다. 한글을 구성하는 stroke는, 자모간의 접촉 또는 필기체 본래의 특성, 기타 세선화처리 과정에서 발생하는 잡음 등 때문에 다양각색의 형태로 나타난다. 따라서 동일 카테고리에 속하는 한글이라 할지라도 stroke 변형에 따라 수많은 종류의 문자가 발생하게 되어 이것을 모두 생성규칙으로 표현하고자 한다면 생성 규칙의 수가 크게 늘어나고 복잡해진다. 본 논문에서는 이 문제를 해결하기 위해 한글의 stroke를 진행 방향에 따라 8개의 그룹으로 분류하고, 모든 문자는 이 stroke 그룹의 조합으로 나타낸다. 각 그룹은 진행 방향을 나타내는 대표 stroke와 이것이 각가지 형태로 변형이된 변형 stroke로 구성되어 있다.

그림 6에 자모간 접촉에 의해 7방향 stroke의 형태가 변형되는 예를 나타낸다. 그림 7에는 G1, G7 그룹에 속하는 stroke의 종류를 나타낸다. 생성규칙은 크게 두가지로 분류할 수 있다. 하나는 각 그룹에 속하는 stroke를 발생하는 규칙들이고 다른 하나는 이들 stroke 그룹의 조합에 의해 기술된 각종 한글구조 생성규칙들이다. 이와같이 모든 문자를 stroke 그룹의 조합으로 나타냄으로써 문법의 설계가 간편해지고 또한 생성규칙의 수를 크게 줄일 수 있다.

그림 5의 계층구조에 따라 작성한 문법의 일부를 그림 8에 나타낸다. (1)의 구문규칙과 의미규칙은 $C1 \cdot V1 \cdot C2$ 형 한글의 경우 C1과 V1간에는 위치관계(fr), C1과 C2간에는 하치관계(fb)가 성립함을 나타낸다. (3)

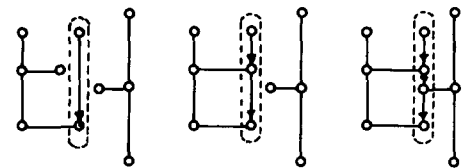


그림 6. 자모간 접촉에 의한 스트로크의 변형
Fig. 6. Stroke transforms by contacts between character components.

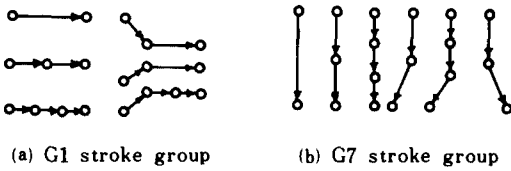


그림 7. 스트로크 그룹의 예
Fig. 7. Example of stroke groups.

$V_n = \{S3, S4, C1, C2, V1, V2, GY, NU, ER, AA, OH, WU, G1, G3, G5, G6, G7, DA, DB\}$
 $V_r = \{1 (\rightarrow), 2 (\nearrow), 8 (\searrow)\}$
P : 구문규칙 의미규칙

(1) S3 → C1 V1 C2 A(C1, V1) = $f_r(A(C1), A(V1))$
 A(C1, C2) = $f_s(A(C1), A(C2))$

(2) S4 → C1 V2 C2 A(C1, V2) = $f_r(A(C1), A(V2))$
 A(C1, C2) = $f_s(A(C1), A(C2))$

(3) C1 → GY | NU | A(C1) = A(GY) | A(NU) |
(4) V1 → ER | AA | A(V1) = A(ER) | A(AA) |
 id(C1, V1) = 1 | 2

(5) V2 → OH | WU | A(V2) = A(OH) | A(WU) |
 id(C1, V2) = 6 | 7

(6) C2 → C1 A(C2) = A(C1), id(C1, C2) = 6 | 7

(7) GY → G1 G6 A(GY) = $f_s(A(G1), A(G6)), D(G1) = 1$

(8) NU → G7 G1 A(NU) = $f_s(A(G7), A(G1)), D(G7) = 1$

(9) ER → G7 G5 A(ER) = $f_s(A(G7), A(G5))$
 id(G7, G5) = 7, D(G7) = 1

(10) AA → G7 G1 A(AA) = $f_s(A(G7), A(G1))$
 id(G7, G1) = 7, D(G7) = 1

(11) OH → C1 G3 A(OH) = $f_s(A(G1), A(G3))$
 id(G1, G3) = 1, D(G1) = 1

(12) WU → G1 G7 A(WU) = $f_s(A(G1), A(G7))$
 id(G1, G7) = 1, D(G1) = 1

(13) G1 → DA DB | DB A(G1) = $f_s(A(DA), A(DB)) | A(DB)$

(14) DA → 1 | 2 A(DA) = A(1) | A(2)

(15) DB → 1 | 1 | 1 A(DB) = $f_s(A(1), A(1)) | A(1)$

그림 8. 한글표현 속성문법
Fig. 8. Attributed grammar to represent korean characters.

의 구문규칙은 C1에 위치하는 자음의 종류로 GY(ㄱ), NU(ㄴ) 등이 있으며, 의미규칙은 GY, NU 등의 속성이 C1으로 상속됨을 나타낸다. (4)의 구문규칙은 V1 위치에 올 수 있는 모음의 종류로 ER(ㅏ), AA(ㅑ) 등이 있으며, 의미규칙은 C1에 대한 V1의 상대 위치속성은 1 또는 2가 됨을 나타낸다. (6)의 구문규칙은 C2에는 C1과 동일한 자음이 위치하는 것을 나타내고 의미규칙은 C1에 대한 C2의 상대 위치속성이 6 또는 7임을 나타낸다. (7)의 구문규칙은 GY(ㄱ)가 G1 그룹과 G6 그룹에 속하는 stroke들로 구성되어 있으며 의미규칙은 G1과 G6 간에는 연결관계(fc)가 성립하고 또한, G1의 시점분기도는 1임을 나타낸다. (13) 이하는

8 종류의 stroke 그룹에 속하는 각 stroke를 생성하는 생성규칙들이 온다. (13)~(15)는 대표 stroke의 방향이 1인 G1 그룹의 구문규칙과 의미규칙으로 G1그룹을 형성하는 각 stroke를 구성하는 부분 패턴의 종류와 연결관계를 규정하고 있다.

V. 속성문법에 의한 구문해석

1. 도출출발 node의 결정

본 논문에서 정의하고 있는 문맥자유 속성문법은 string 문법으로 문헌[6]에서 증명된 바와 같이 입력문자를 string으로 표현가능한 온라인 문자인식에는 유효하게 적용된다. 그러나 string 문법을 사용하여 필기체한글과 같이 2차원 구조의 패턴을 해석하는 데에는 문헌[6]의 알고리즘을 그대로 적용할 수 없다. 본 논문에서는 이 문제를 해결하기 위해 생성규칙이 지정하는 바에 따라 그래프패턴상을 이동하는 pointer를 도입, 생성규칙에 의한 도출을 수행할 때마다 pointer가 새로이 도착 할 node를 결정하여 다음번째 생성규칙을 적용할 때에는 이 node를 pointer의 이동기점으로 사용토록 하는 작업을 추가하므로서 문헌[6]의 구문해석 방법을 적용할 수 있도록 하였다. 새로운 node는 현 node로 부터, 구문규칙이 지정하는 구조와 의미규칙이 정하는 속성을 가진 node가 있는지를 탐색하여 결정하였다.

구문규칙에 의한 도출은 문법설계의 역조작으로, top-down 도출을 수행하면서 도출 tree(계층구조에 해당)를 작성하는 작업이다. 도출 Tree작성에 관한 상세한 사항에 대해서는 문헌[6]을 참조하기 바란다.

계층구조가 성공적으로 작성된 후에는 계층구조의 자모 node의 label를 출력하므로써 입력문자를 구성하는 자모를 인식하게 된다. 이상과 같은 방법으로 입력문자를 인식하는데 있어서 가장 중요한 사항은 각 자모의 추출을 시작할 때마다 최초로 pointer가 위치해야 할 도출 출발 node를 결정하는 문제이다. 이를 위하여 다음에 정의하는 5개의 도출 기준점을 추출한 다음, 그 기준점을 근거로 도출 출발 node를 구하였다.

입력패턴M을 둘러싸는 영역{M}에 대하여 다음의 값을 구한다.

$$x_1 = \min \{ \{M\}_x \}, y_1 = \min \{ \{M\}_y \}$$

$$x_2 = \max \{ \{M\}_x \}, y_2 = \max \{ \{M\}_y \}$$

$$x_3 = 1/2(x_2 - x_1), y_3 = 1/2(y_2 - y_1)$$

[정의7] 5개의 도출 기준점을 Oc1, Ov1, Ov2, Oc2, Oc3라 하고 그 좌표를 각각 (x1, y1), (x2, y1), (x1, y3), (x1, y3), (x3, y3)으로 한다. Oc1은 초성자음(C1그룹)을 추출하기 위한 도출기준점

이고, Ov_1, Ov_2, Oc_2, Oc_3 는 각각 횡모음(V1 그룹), 종모음(V2 그룹), 초성자음(C2 C3 그룹)을 추출하기 위한 도출 기준점이다. 영역 $\{M\}$ 으로 부터 구한 각 도출 기준점을 그림 9에 나타낸다.

각 자모의 도출 기준점을 위와 같이 정의한 이유는 입력문자가 포함되는 영역내에서 각 자모가 배열되는 위치를 고려하여 결정한 것이다.

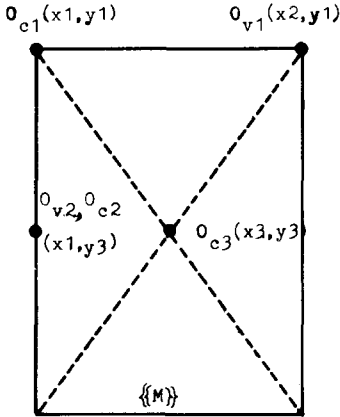


그림 9. 다섯개의 도출기준점
Fig. 9. Five derivation basis points.

2. 각 자모의 분할

도출을 시작하기에 앞서서 우선 C1을 추출하기 위한 도출출발 node $Nc1$ 과 V1추출을 위한 도출 출발 node $Nv1$ 을 구한다. 그 방법은 $Oc1$ 과 $Ov1$ 으로 부터 각각 최단의 거리에 위치하는 node로 한다. $Nc1$ 과 $Nv1$ 이 결정되면 $Nc1$ 을 현 node로 하여 문법의 출발기호 S 로 부터 top-down 최좌도출 (left-most derivation) 을 수행하여 도출 tree를 작성한다. 도출 tree를 작성하면서 구문규칙과 의미규칙이 지시하는 바에 따라 현 node로부터 pointer가 이동해야 할 다음 node를 아래의 수순으로 그래프패턴 상에서 결정한다.

(1): 도출 tree의 label이 비종단 기호의 경우는 현 node로 부터 pointer를 이동시키지 않고 종단기호의 경우에만 종단기호가 지정하는 방향으로 pointer를 이동시킨다. 이때 pointer가 이동되어 새로 도착된 node의 특징을 그 구문규칙에 수반된 의미규칙으로 점검한다.

(2): 새로 도착된 node의 특징이 의미규칙이 지정하는 바와 일치하면 pointer가 통과한 branch는 삭제하고 이동전에 pointer가 위치하였던 node의 분기도를

1만큼 감소시킨다.

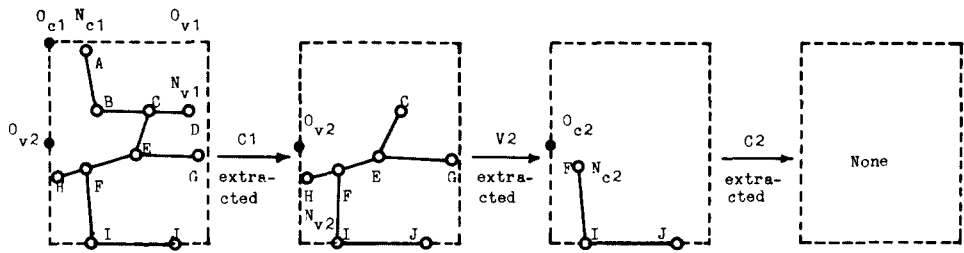
(3): (2)에서 구한 새로운 node를 현 node로 하여(1), (2)의 과정을 반복한다.

(4): (3)의 과정을 수행하는 중에 그래프패턴의 구조가 구문규칙과 의미규칙의 조건에 맞지 않아 pointer의 이동이 불가능한 경우에는 해당 생성규칙을 적용하기 이전의 위치로 pointer를 후퇴시킨후에 다른 구문규칙을 적용하여 (1), (2)의 과정을 반복하는 backtracking 을 수행한다.

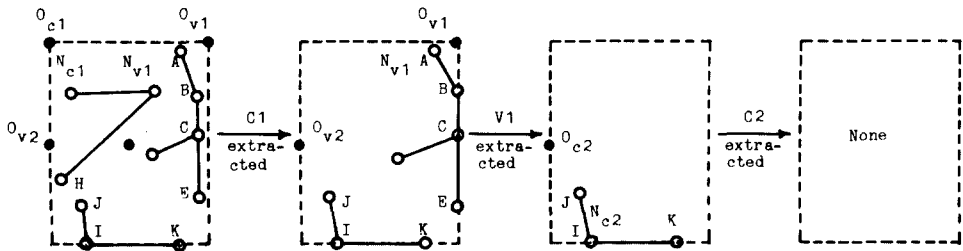
이상과 같은 도출에 의해 초성자음인 C1이 분할된 후에는 횡모음(V1)과 종모음(V2)의 분할을 시작한다. 분할을 시작하기 전에 먼저 주어진 입력문자가 횡모음을 포함하고 있는지를 결정한다. 결정방법은 다음과 같다. 즉 C1의 분할에 들어가기 전에 횡모음추출을 위한 도출 출발 node로 정해 놓은 $Nv1$ 이 C1 추출후에도 그대로 남아 있는지를 검사한다. 만약 $Nv1$ 이 삭제되었을 경우(그림10(a))는 이 $Nv1$ 은 C1에 포함되는 node로 주어진 입력패턴은 횡모음이 없고 종모음만 존재하는 문자임을 알 수 있다. 따라서 이때에는 $Ov2$ 를 기준으로 하여 V2추출을 위한 도출 출발 node $Nv2$ 를 새로 결정한 다음 종모음의 추출을 위한 과정을 위와 동일한 순서로 진행한다. $Nv1$ 이 남아 있을 경우(그림10(b))는 입력패턴은 횡모음을 포함하고 있는 문자임을 알 수 있기 때문에 $Nv1$ 으로 부터 출발하여 V1의 추출을 C1의 경우와 동일한 방법으로 수행한다. V1과 V2가 추출된 후에도 입력패턴의 일부가 남아 있으면 주어진 문자는 종자음도 포함하고 있음을 알 수 있기 때문에 계속해서 C2 및 C3의 추출을 진행한다.

위와 같은 도출을 순차적으로 수행하여 도출 tree의 작성이 완료된 단계에서 그래프패턴의 모든 node와 branch가 삭제되어 있으면 입력문자는 올바르게 인식된 것이다. 이때 입력문자를 구성하는 자모는 도출과정에서 작성한 도출 tree에서 자모가 위치하는 node의 label에 해당하며, 이 label을 최종 인식 결과로서 출력한다.

이상 설명한 바와 같은 구문해석에 따라 입력된 그래프 패턴이 삭제되어 가는 과정을 그림10에 나타낸다. 그림10(a)의 경우A와 D가 각각 $Nc1$ 과 $Nv1$ 로 결정되거나 C1이 추출된 후에는 $Nv1$ 이 삭제되기 때문에 $Ov2$ 에서 최단거리에 위치하는 H가 새로이 $Nv2$ 로 결정되어 V2의 추출을 시작한다. V2가 추출된 후에는 $Oc2$ 에서 최단거리에 위치하는 F가 $Nc2$ 로서 결정되어 C2의 추출에 들어간다. C2가 추출된 후에는 입력패턴이 전부 삭제되어 있으므로 C3의 추출에는 들어가지 않고 최종인식 결과로 “ㄴ,” “ㄴ,” “ㄴ”을 출력하고 도출을 완료한다.



(a) C1 · V2 · C2 type hangeul



(b) C1 · V1 · C2 type hangeul

그림10. 자모의 분리과정

Fig. 10. Segmentation process of each character components.

VI. 인식실험 결과

본 인식알고리즘의 유효성을 확인하기 위하여 인식 실험을 수행하였다. 인식실험에 사용한 문자는 전장에서 정의한 7 종류의 위치관계와 각 자모가 고루 분포하는 128자이다.

이 128자에 대하여 3인의 필자로 하여금 타블렛 위에 설정한 5cm * 5cm 영역내에 또박또박 정자로 쓰도록 하였다. 이 영역내에 쓰여진 문자는 64 * 64 matrix 로 투영하여 입력데이터로 사용하였다. 특징점 추출을 위한 방향 코오드의 번역에 들어가기전에 입력 데이터에 대하여 세선화 처리를 수행하였다. 각 특징점이 추출된 후에는 임의의 분기점을 중심으로 반경 2의 원형시야를 설정하고, 그 안에 다른 분기점이 존재할 경우는 그 2개의 분기점을 동일점으로 간주하여 처리하였다. 이것은 세선화 과정에서 발생하는 패턴의 찌그러짐 현상 때문에 그래프 패턴의 구조가 복잡해지는 것을 가급적 줄이기 위해서이다. 도출 출발 node의 결정에 있어서는 동일 자모라도 필자의 필기습관에 따라서 다른 node가 도출 출발 node로 선정되는 경우가 있다. 예를 들면 그림10(a)의 V2추출을 위한 도출 출발 node는 경우에 따라 H 또는 F가 선정될 수 있다.

출발 node가 다르기 때문에 이 두가지 경우는 문자의 생성과정이 다르게 된다. 따라서 문법의 설계에 있

어서는 이러한 변화를 가급적 모두 흡수할 수 있도록 생성규칙을 작성하였다. 3인이 쓴 총 384자에 대하여 인식실험을 수행한 결과를 표 1에 나타낸다. 오인식 또는 reject된 문자의 예를 그림11에 나타낸다. 그림의 (a), (b), (c)는 세선화에 따른 패턴변형으로 reject된 문자이다. (d), (e)는 수직 stroke의 추출이 잘못되어 "붓"이 "ㄱ", "ㄷ", "ㄴ"로, "식"이 "사", "ㅣ", "ㅡ"로 오인식된 경우이다. (f)는 같은 loop에 속하는 "ㅁ"과 "ㅇ"의 구별에 실패하여 "ㅁ"을 "ㅇ"로서 오인식한 경우이다.

VII. 결 론

인식대상이 되는 문자가 대단히 많은 한글의 경우 결정이론적 방법에 의한 인식방법 보다는 문자의 구성 요소가 24자에 지나지 않고 그 위치도 고정되어 있는

표 1. 인식실험의 결과

Table 1. Result of experimentation.

정 인 식	357 자	93%
리 켜 트	21 자	5.5%
오 인 식	6 자	1.5%
총 계	384 자	100%

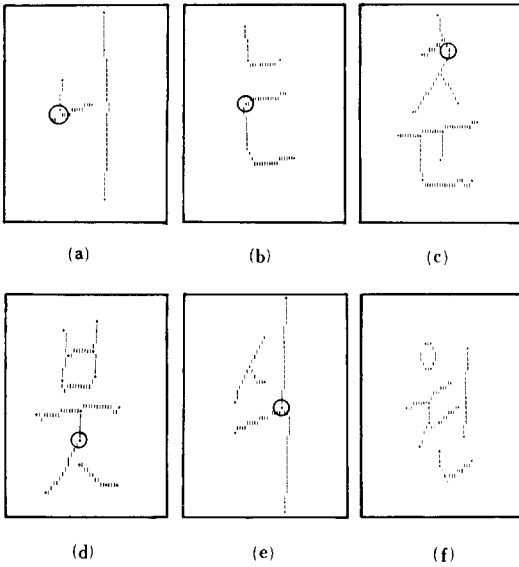


그림 11. 오인식된 문자의 예
Fig. 11. Example of ill-recognized characters.

구조적 특징을 고려, 입력패턴에 대한 구조해석을 통하여 각 자모를 순차로 추출해 하는것이 보다 유효한 인식방법으로 생각된다. 이와같은 입장을 기본으로 하여 각 자모의 배열위치가 일정하고 또한 문자의 생성과정이 계층구조로 되어있는 사실에 착안, 구문 해석적 방법을 사용하여 입력문자에 대한 구조해석을 수행하는 방법을 제안하였다. String 문법인 속성문법을 사용하여 2차원 패턴의 구조해석을 시도하는데 따른 문제는 문법의 생성규칙에 의한 도출에 따라 그래프 패턴상을 이동하는 pointer를 도입함으로써 해결하였다. 문법의 설계를 단순화 시키기 위해 한글의 stroke를 진행방향에 따라 8개의 그룹으로 분류하고 모든 자모는 이들 stroke 그룹의 조합으로 나타냈다. 입력되는 문자는 모두 64*64의 matrix에 수용한 다음 인식

을 수행하고 있기 때문에 본 알고리즘은 on-line, off-line 한글인식 모두에 공히 적용될 수 있는 특징이 있다. 인식 실험을 행한 결과 비교적 양호한 인식율을 얻었지만 실용화 시스템으로 발전시키기 위해서는 여러 정정 구문해석 방법의 도입과 구문해석의 속도를 향상시키는 연구가 필요할 것으로 생각된다.

參 考 文 獻

- [1] Agui, Nakajima and Kim: "A method of recognition and representation of Korean characters by tree grammars", *IEEE Trans. Pattern. & Mach. INtell.* PAMI-1, no. 3, pp. 254-259, 1978.
- [2] B.C. Lee, J.G. Kim and S.K. Yi: "Improvement on Korean characters recognition by resolving ambiguity problem", *Proceedings International Conference on Cybernetics and Society*, pp. 193-197, 1981.
- [3] K.Y. Kim: "New algorithm for recognition of Korean characters," Paper for M.D., Inha University, Korea, 1981.
- [4] W.H. Tsai and K.S. Fu: "Attributed grammar," *IEEE. Trans. SYS, Man & Cyber*, SMC-10, no. 12, pp. 873-885, 1980.
- [5] 김태균, 安居院, 中嶋: "다단번역에 의한 온라인 한글의 인식, 일본전자통신 학회논문지, vol. J 68-D, no. 2, pp. 177-184, 1985.
- [6] 김태균, 安居院, 中嶋: "속성문법에 의한 홀림체 한글의 온라인 인식," 일본전자통신학회논문지, vol. J 68-D, no. 4, pp. 797-804, 1985.
- [7] 安居院, 中嶋: "컴퓨터 화상처리," 산보출판, 1973.
- [8] 小山, 直田, 手塚: "세션화 도형의 굴곡점 검출의 한 방법," 일본전자통신학회 기술연구보고 PRL 80-107, pp. 83-90, 1980.