

# 補強文脈自由文法을 이용한 筆記體 한글 온라인 認識

## (On-Line Recognition of Handwritten Hangeul by Augmented Context Free Grammar)

李 羲 東\*, 金 太 均\*\*

(Hee Dong Lee and Tae Kyun Kim)

### 要 約

본 논문에서는, 보강문맥자유문법을 이용한 필기체 한글의 온라인 인식에 대하여 논한다. 문맥자유문법에 의한 구문해석에서 발생하는 애매성은, 대부분 한글을 구성하는 부분패턴간 위치관계의 기술이 불충분한 것에 기인한다.

실험을 거듭함으로써 애매성을 해소하는 생성규칙 적용조건을 구하였다. 이것을 보강문맥자유 생성규칙의 확장부에 부가함으로써 유연한 구문해석이 가능하였으며 인식률도 높일 수 있었다. 또한, 종래의 구문해석 시스템에 비하여 매우 적은량의 생성규칙으로 시스템구성이 가능하였다. 학습샘플을 빠르게 인식하도록 생성규칙 적용조건을 구한 후, 다양한 모양의 미지의 한글샘플에 대하여 인식실험을 행한결과, 99%의 정인식율을 얻었다.

### Abstract

A method of on-line recognition of Korean characters (Hangeul) by augmented context free grammar is described in this paper. Syntactic analysis with context free grammar often has ambiguity. Insufficient description of relations among Hangeul sub-patterns causes this ambiguity can be determined through repetition of experiments. Flexible syntactic analysis is executed by adapting the condition to the (advice) part of augmented context free grammar. The ratio of correct recognition of this method is more than 99%.

### I. 서 론

고도 정보화사회에서 자연언어 처리의 연구가 각 방면으로 진전되고 있다. 이와 함께 온라인 문자인식은 간단한 문자입력수단의 하나로서 그 필요성이 높아지고 있다. 온라인 문자인식에 의한 입력방식에서는 문자를 직접 쓴다는 자연스러운 동작에 의하여 문서가

입력된다. 또한, 오인식의 발견이나 수정을 즉석에서 대화적으로 행할 수 있는 특징이 있다.

이와같은 특징을 살린, 거의 숙련을 필요로 하지 않는 입력장치로서 온라인 필기체 한자 인식장치가 최근 제용화 되기 시작하였으며, 연구도 활발히 진행되고 있다.<sup>[1]-[6]</sup> 문헌[1], [6]에서는, 흘림체 문자등과 같은 필기체한을 완화한 온라인 한자인식, 문헌[4]에서는, 필순에 의존하지 않는 온라인 문자인식, 문헌[5]에서는 획수에 의존하지 않는 문자인식, 문헌[7]에서는, 흘려 쓴 문자의 온라인 인식등, 각종 온라인문자 인식에 관한 연구가 행하여지고 있으며, 이러한 추세에 따라 한글의 온라인 인식에 관한 연구도 최근에 와서 행하여지고 있다.<sup>[8,9,10]</sup>

한글은 문자의 구성요소가 계층구조로 되어있는 특징

\*正會員, 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)

\*\*正會員, 忠南大學校 電子計算機工學科 (Dept. of Computer Eng., Chungnam Univ.)

接受日字: 1986年 11月 7日

이 있다. 그러므로 한글의 온라인 인식에 있어서는 이러한 특징을 활용하기 쉬운 구문해석적 수법을 이용하는 것이 효과적이다.<sup>[5]</sup> 그러나 종래의 방식<sup>[6]</sup>에서는 문자패턴의 위치관계를 나타내는 오퍼레이터를 비중단기호로 하는 생성규칙을 사용하였기 때문에 생성규칙의 수를 증가시키지 않고 문자패턴을 상세하게 기술하는 것이 곤란하였다. 또한, 인식결과를 정보처리 기기가 직접 이용할 수 있는 형태로 하기 위해서는 구문해석이 끝난 단계에서 얻어지는 구문해석기의 생성언어를 엔트리로 하여, 또 한번 사전찾기를 해야하는 필요가 있다.

본 논문에서는, 보강문맥자유 문법을 이용한 온라인 한글인식 수법을 제안한다. 구체적으로는, 생성규칙의 확장부에, 생성규칙 적용조건으로서 각 자소간의 위치관계를 조사하는 함수를 추가하였다.

학습샘플에 대해서 구문해석을 반복하면서 생성규칙 적용조건을 순차 수정함으로써, 문자패턴을 보다 상세하게 기술하는 것이 가능하였으며, 미지샘플에 대하여 99%의 높은 정인식율을 얻을 수 있었다.

그림 1에 시스템의 인식처리 수준을 나타낸다. 터블렛으로부터 입력되는 좌표 시계열(time sequence)에 대하여, 다단번역에 의한 그룹스트로-크·코-드의 결정과 위치관계를 나타내는 특징량의 추출을 행한다. 이 그룹스트로-크·코-드와 특징량을 이용하여 입력문자를 기술한다. 기술된 입력문자에 대하여, 한글의 일괄생성규칙에 의한 구문해석을 거쳐 입력문자의 인식을 행한다.

구문해석 수행시, 문자를 구성하는 각 자소가 생성

되는 시점에서, 자소명이 결정될 수 있도록, 생성규칙에 자소명을 부가하며, 구문해석에 성공한 생성규칙에 해당되는 자소명을 출력한다. 따라서, 이 방식에서는 인식결과를 자소명의 나열로 얻을 수 있다.

## II. 문자의 기술

### 1. 문자의 입력

문자는 타블렛상의 펜의 움직임에 따른 일련의 x, y 좌표값으로 입력된다. 한 문자가 쓰여지는 타블렛의 영역은 64×64화소로 분할되어 있다. 입력된 x, y 좌표값은, 분할된 각 영역상의 해당위치의 x, y좌표값으로 변환된다. 이와 같이하여 얻어진 좌표점 중에서, 직전의 좌표점과 같은 위치의 점은 무시하고, 전후 두 좌표점이 서로 다른 점만을 샘플점으로 등록한다. 펜다운(pen-down)으로 부터 펜업(pen-up)까지를 하나의 스트로-크로 정의한다.

입력문자를 M, 그 문자를 구성하는 각 스트로-크를 S<sub>i</sub>라고 한다면, 입력문자 M은 다음과 같이 나타낼 수 있다.

$$M = (S_1, S_2, \dots, S_i, \dots, S_N)$$

$$S_i = (P_{i,1}, P_{i,2}, \dots, P_{i,j}, \dots, P_{i,n_i})$$

단,

$$P_{i,j} = (x_{i,j}, y_{i,j}); S_i \text{의 } j\text{번째 샘플점의 좌표}$$

$$P_{i,j} \neq P_{i,j+1}$$

$$i = 1, 2, \dots, N, j = 1, 2, \dots, n_i - 1$$

N; 스트로-크의 총수

n<sub>i</sub>; S<sub>i</sub>의 샘플점수

### 2. 그룹스트로-크·코-드의 결정

한 문자의 입력이 끝나면, 각 스트로-크를 구성하는 좌표 데이터로부터 전후들의 샘플점 간을 잇는 직선요소를, 그림 2에 나타난 8방향 코-드로 양자화

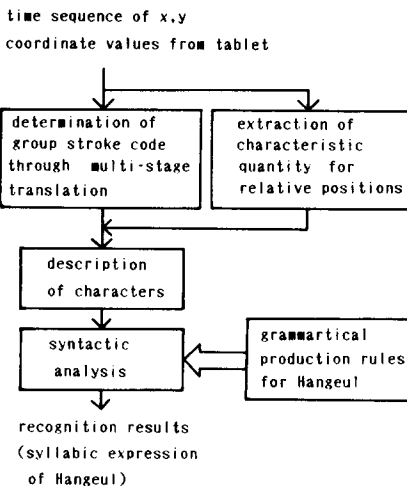


그림 1. 인식처리의 수준  
Fig. 1. Recognition Procedure.

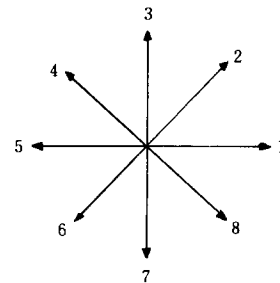


그림 2. 8방향 코-드  
Fig. 2. Eight Directional Codes.

한다. 이와같이 하므로서 입력문자를 구성하는 각 스트로-크는 펜의 움직임에 따른 8 방향 코-드 열로서 표시된다. 이와같이 구한 8 방향 코-드열은, 하나의 스트로-크를 쓰기 시작하여 그 스트로-크가 끝날때까지의 펜의 움직임만을 나타낼 뿐 샘플점간의 거리와는 무관하다. 이 코-드열에 대하여 문맥의존 번역기 (syntax-directed translation schemata)를 사용하여 다단번역<sup>[8]</sup>을 행하므로서 그림 3에 나타낸 바와 같은 그룹 스트로-크·코-드를 결정한다. 이후, 문맥의존 번역기를 단지 번역기라 부르기로 한다.

번역기의 구성에 있어서는 한글의 구조상의 특징, 즉 스트로-크의 길이보다는 펜의 이동방향이 인식에 중요한 정보를 준다는 점을 고려하여, 생성규칙을 설계하였다.

번역은 3 회로 나누어 행하며, 필자들은 이것을 다단번역이라 부르고 있다. 펜의 이동방향을 나타내는 방향코-드에는 필기속도가 늦거나 또는 펜끝의 떨림이 있을 경우, 다수의 불필요한 코-드가 발생한다. 이러한 불필요 코-드는 다단번역을 행할 때 삭제변환 하므로서 없앤다. 그다음, 비직선성 방향코-드는 치환변환하여, 스트로-크의 진행방향을 유지토록 하므로서 직선성을 갖도록 하였다.

불필요한 코-드를 없애고 스트로-크가 직선성을 갖도록 한 후, 다단번역의 최종단계에서는 또한번 삭제변환을 행하여, 스트로-크 방향코-드의 압축과 그룹 스트로-크·코-드를 결정한다.

번역기에 의한 다단번역의 자세한 사항에 대하여서는 문헌<sup>[8]</sup>을 참조하기 바란다.

group stroke codes	basic strokes
rd(right-down)	↘ ↙ ↘ ↙
dr(down-right)	↘ ↙ ↘ ↙
cc(circle)	⊙ ○ ⊙
sw(south-west)	↙
ss(south-south)	
se(south-east)	↘
ee(east-east)	—

그림 3. 기본 스트로-크와 그룹스트로-크·코-드  
Fig. 3. Basic Strokes and Group Stroke Codes.

그림 4는 입력스트로-크로부터 그룹스트로-크·코-드로의 다단번역 과정의 일례를 보인다. 이와같이 구한 그룹스트로-크·코-드는, 스트로-크의 길이에 무관하고 그 진행방향만을 주목한 것으로서, 입력문자의 크기와 필기속도에 의존하지 않는 일종의 정규화된 스트로-크라 볼 수 있다. 이 그룹스트로-크·코-드는 구문해석에서 입력문자의 구조를 나타내는 정보로 사용된다.

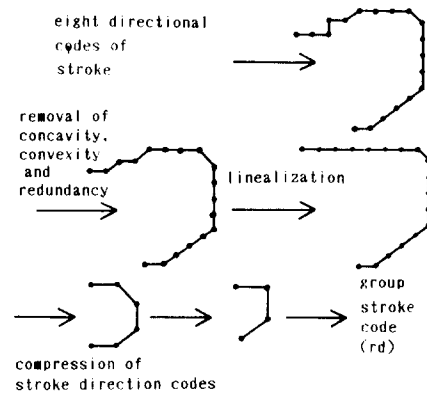


그림 4. 다단 번역에 의한 그룹 스트로-크·코-드의 결정  
Fig. 4. Determination of Group Stroke Codes Through Multi-stage Translations.

3. 상대위치 관계를 나타내는 특징량의 추출

본 인식시스템에서는 번역기의 출력인 그룹스트로-크·코-드열에 대하여 구문해석을 행하므로서 입력문자를 인식한다. 그러나 그룹스트로-크·코-드 열이 동일하게 나타나는 문자의 경우, 그 인식에 애매성이 발생한다. 따라서, 구문해석 때 다른 특징량, 즉 스트로-크간의 위치관계를 설정하여 애매성을 없앤다. 스트로-크간의 위치관계를 명확히 기술하기 위하여 사용하는 특징량을 다음과 같이 정의한다.

먼저 가상 스트로-크의 방향코-드로서  $hh_i$  (head-head<sub>i</sub>)과  $th_i$  (tail-head<sub>i</sub>)를 정의한다.  $hh_i$ 는  $S_i$ 의 시점  $P_{i1}$ 과  $S_{i-1}$ 의 시점  $P_{i-1,1}$  ( $i=1, 2, \dots, N-1$ ) 간을 잇는  $N-1$ 개의 가상 스트로-크의 방향코-드이다.  $th_i$ 는  $S_i$ 의 종점  $P_{iN}$ 과  $S_{i-1}$ 의 시점  $P_{i-1,1}$ 을 잇는  $N-1$ 개의 가상 스트로-크이다. 그 외에  $hp_i$  (head-point),  $tp_i$  (tail-point),  $lx_i$  (left- $x_i$ ),  $rx_i$  (right- $x_i$ ),  $hy_i$  (high- $y_i$ ) 및  $ly_i$  (low- $y_i$ )를 다음과 같이 정의한다.

$$\begin{aligned}
 hp_i &= p_{i,1} \\
 tp_i &= p_{i,n_i} \\
 lx_i &= \min\{x_{i,j} \mid j=1, 2, \dots, n_i\}
 \end{aligned}$$

$$rx_i = \max \{x_{i,j} \mid j = 1, 2, \dots, n_i\}$$

$$hy_i = \min \{y_{i,j} \mid j = 1, 2, \dots, n_i\}$$

$$ly_i = \max \{y_{i,j} \mid j = 1, 2, \dots, n_i\}$$

입력문자「나」에 대한 각 특징량을 그림 5에 나타낸다.

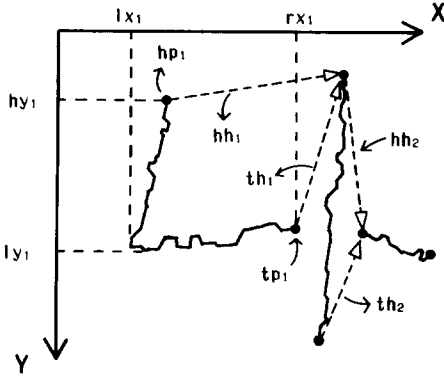


그림 5.  $S_{i,sem}$ 의 구성요소  
Fig. 5. Elements of  $S_{i,sem}$ .

4. 문자의 기술

구문해석의 입력으로 들어가는 입력문자의 최종적인 형태는, 다음의  $M^*$ 과 같다.

$$M^* = (S^*_1, S^*_2, \dots, S^*_i, \dots, S^*_n)$$

$$S^*_i = (S_{i,son}, S_{i,sem})$$

단,  $S^*_i$ 는  $M^*$ 을 구성하는 스트로-크이다.

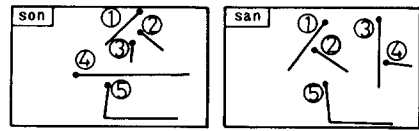
$S_{i,son}$  {그룹 스트로-크 · 코-드}

$S_{i,sem} = (hh_1, th_1, hp_1, tp_1, lx_1, rx_1, hy_1, ly_1)$

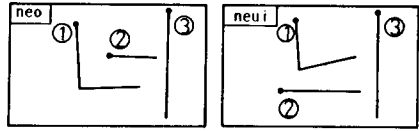
여기서 각 스트로-크  $S^*_i$ 의  $S_{i,sem}$ 은 애매성이 발생하였을 때 이를 해소하는 생성규칙 적용조건을 기술하는 곳에 사용되는 위치관계이다.

$S_{i,sem}$ 을 이용함으로써 인해 애매성이 해소되는 예는 다음과 같다. 즉 그룹스트로-크 · 코-드 열  $S_{i,son}$ 이 동일한 두 문자간에 가상 스트로-크인  $hh_1$ 와  $th_1$ 가 일치하는 경우와 그렇지 않은 경우의 예를 그림 6에 나타낸다. 그림 6의 각 스트로-크의 검은 점은 스트로-크의 head를 나타내고, 원으로 둘러싸인 숫자는 스트로-크의 번호를 나타낸다.

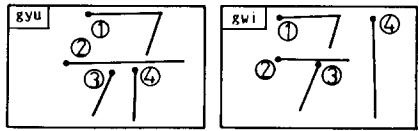
그림 6(a)의 가상 스트로-크 방향코-드  $hh_3$ 와  $th_3$ 의 경우 「손」은 (6, 5)이고, 「산」은 (7, 3)이기 때문에 가상 스트로-크의 방향코-드에 의한 양문자의 구별은 가능하다. 그러나 그림 6(b)의 「너」와 「뇌」의 경우,  $hh_2$ 와  $th_2$ 가 양쪽 모두 (2, 3)이고, 또한 그림 6(c)의 「규」와 「귀」의 경우는  $hh_3$ 와  $th_3$ 가 양쪽 모두 (2, 2)이기 때문에 가상 스트로-크의 방향코-드에 의한 애매성의 해소는 불가능하다.



(a) Son and San (sw se ss ee dr)



(b) Neo and neui (dr ee ss)



(c) Gyu and gwi (rd ee sw ss)

그림 6. 가상 스트로-크에 의해 애매성이 해소되는 경우와 해소되지 않는 경우의 예

Fig. 6. Removal of Ambiguity by Checking Virtual Strokes in Success and Vain.

가상 스트로-크를 이용하여도 애매성이 해소되지 않는 경우에는, 다른 특징량을 이용하여 문자를 구분한다. 예를들면, 그림 6(c)의 경우 「규」의  $hy_3$ 과  $hy_4$ 의 값은 거의 같은 정도이나, 「귀」의 경우는 상당한 차이가 있다. 이와 같은 제한을 보강문맥 자유문법의 생성규칙의 확장부에 기술함으로써 생성규칙의 적용을 제어할 수 있다.

III. 보강문맥자유문법과 인식방법

1. 보강문맥자유문법

본 시스템의 구문해석에서 사용하는 생성규칙의 형식은  $n$ 진 tree의 보강문맥자유문법규칙<sup>[11]</sup>이다. 그림 7에 LISP에 의한 표현형식을 나타낸다. 생성규칙의 (advice)를 확장부, (sem)을 의미해석부라 부르며, 문맥자유문법규칙에 (advice), (sem)을 첨가함으로써 개선된 구문해석을 행할 수 있다. 보강된 생성규칙을 보강문맥 자유문법규칙이라 한다. 다음의 규칙

$$NT_e \rightarrow NT_1 \dots NT_j$$

에서 화살표 우측의 각 비종단기호(nonterminal symbol)  $NT_i (1 \leq i \leq j)$ 에는 확장부(advice)  $i$ 라고 불리우는 프로그램이 부가되어 있다. 이것은  $NT_i$ 에 대한 구문해석이 종료된 단계에서 기동되는 프로그램으로, 이것은 이제까지 수행한 부분적인 구문해석 결과를 검토하여, 이 규칙을 적용하는 구문해석을 계속  $NT_{i+1}$ 로 진행할 것인가를 결정한다. 이렇게 생성규칙에 확장부를

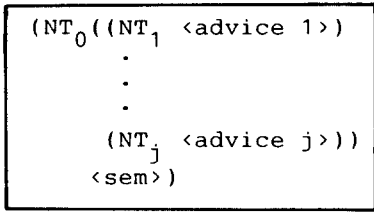


그림 7. 보강문맥자유 문법규칙의 LISP에 의한 표현 형식

Fig. 7. Augmented Context Free Grammar Expressed in LISP.

도입하므로써 생성규칙의 수를 증가시키지 않고 보다 정밀한 문법의 설계가 가능하다.

2. 인식방법

본 시스템에서는, 입력문자를 나타내는 그룹스트로-크·코-드열 및 그 열을 구성하는 각 그룹스트로-크·코-드에 해당하는 패턴의 상호위치 전계가 보강문맥자유문법에 의한 구문해석을 통하여 생성가능 하였을 때, 생성규칙의 의미해석부에 등록되어 있는 자소명이 인식의 결과로서 출력된다.

본 시스템에서 사용하는 구문해석기는 확장 LINGOL<sup>[12]</sup>을 기본으로 한 것으로, 자연언어처리에서 사용되는 구문해석기를 온라인 한글인식에 응용하였다. 확장LINGOL은 bottom-up식 구문해석을 기초로 하여, top-down적인 예측기능을 부가한 쌍방향의 알고리즘을 이용한 것으로, LINGOL<sup>[12]</sup> 해석에 관한 예측 기능과 일본어용 띄어쓰기 기능을 추가하여 확장한 것이다.

IV. 보강문맥자유문법에 의한 구문해석

1. 애매성의 분류

학습 데이터에 대해서 구조정보인 그룹스트로-크·코-드만을 이용하여 구문해석을 하였을 때 발생한 애매성을 조사한 결과, 표1에 나타낸 바와 같이 4종류로 분류할 수 있었다. 표2는 그 예를 나타낸 것으로, 자형란에 있는 두 패턴은 각각 동일한 그룹스트로-크·코-드열을 출력한다. 그룹스트로-크·코-드의 밀줄친 부분은 애매성이 발생한 부분을 나타낸다. 상호

표 1. 애매성의 분류

Table 1. Classification of Ambiguity.

(1) 초성과 중성에 의한 애매성
(2) 중성간의 애매성
(3) 중성과 중성에 의한 애매성
(4) 변형 자소를 허용하였기 때문에 일어나는 애매성

표 2. 애매성의 분류의 예

Table 2. Example of Classification.

	자 형	그룹 스트로-크·코-드열
(1)	털 털	ee dr ee ee ss rd ee dr
(2)	가 고	rd ss ee
(3)	닝 닝	dr ss ee ee cc
(4)	추 추 무 변형 패턴	ss rd ee ee ss

위치관계의 기술에 있어서는, 전 그룹스트로-크·코-드열에 대해서 기술할 필요는 없고, 단지 애매성이 발생하는 부분에만 위치관계를 기술하므로써 이 애매성을 효율적으로 해소할 수 있다. 즉 표2의 그룹스트로-크·코-드열에 대하여 밀줄친 스트링에 대해서만 위치관계를 검토하면 된다.

2. 애매성을 해소하는 생성규칙의 적용조건

PDL(picture description language)은 다차원 구조의 도형을 표현하는 문법으로, 이를 사용하여 다양한 변형을 포함하고 있는 패턴을 기술할 경우, 생성규칙의 수가 크게 증가하는 단점이 있다. 또한 구문해석에 있어서는 서로 이웃하는 전 스트로-크에 대해서, 위치관계를 조사하므로써 애매성을 해소하고 있다.<sup>[1]</sup>

본 수법에서는, 애매성이 발생하는 부분만을 주목하며, 위치관계는 생성규칙 적용조건을 조사하는 함수를 확장부에 도입하므로써 기술한다. 주로 사용되는 문법 적용조건의 예를 표3에 나타낸다. 이 적용조건은 한글의 자소를 구성하는 각 스트로-크간의 위치관계를 나타낸 것으로, 예를들면, Check-t1은 (ee dr ee)의 그룹스트로-크·코-드열을 가지는 입력문자에 대하여,  $0 \leq d_1 \leq 12, 0 \leq d_2 \leq 12$ 의 조건을 생성규칙 적용조건으로 조사하는 함수이다. 서로 인접하는 스트로-크만이 아니고 서로 인접하지 않는 스트로-크간의 상호위치 관계도, 생성규칙 적용조건으로서 패턴의 기술에 사용하고 있다.

3. 구문해석기와 인식결과

입력문자 「만」을 예로하여 본 수법의 인식과정을 설명한다. 그림8에 문자「만」을 기술한 생성규칙을 나타낸다. 여기서, MIEUM, V-A, DIGEUD은 각각 기본자소, 「ㅁ」, 「ㅏ」, 「ㅓ」을 나타내는 비종단기호이다. 생성규칙 (1)은 C1·V1·C2의 구조를 의미하며 C1은 초성, V1은 중성, C2는 중성을 나타낸다. 규칙 (2), (3),

표 3. 주로 쓰여지는 문법규칙 적용조건  
Table 3. Main Restrictions Required by the Grammar.

$d_1$ : distance,  $dc_1$ : eight directional codes

function	shape	restriction on strokes
check-d		$0 \leq d_1 \leq 12$
check-a		$0 \leq d_1 \leq 12$ $dc_1 = 7$ $dc_2 = 3$
check-o		$dc_1 = 5 \text{ or } 6$
check-yu		$0 \leq d_1 \leq 12$
check-wi		$dc_1 = 2 \text{ or } 3$ $dc_2 = 7 \text{ or } 8$
check-tl		$0 \leq d_1 \leq 12$ $0 \leq d_2 \leq 12$
check-k		$0 \leq d_1 \leq 12$ $dc_1 = 2, 3, 4$ or 5

```

(HANGEUL ((C1)
  (V1 check_bat)
  (C2))
  ((interp_HANGEUL
    (interp 'C1)
    (interp 'V1)
    (interp 'C2))) - (1)
(C1 ((MIEUM))
  (interp 'MIEUM)) - (2)
(V1 ((V_A))
  (interp 'V_A)) - (3)
(C2 ((DIGEUD))
  (interp 'DIGEUD)) - (4)
(MIEUM
  ((ss)
  (rd)
  (ee))
  (V_A
  ('m)) - (5)
((ss (check_a))
  ('a)) - (6)
((DIGEUD
  ((ee (check_d))
  (dr))
  ('d)) - (7)
  
```

그림 8. 문법규칙의 예  
Fig. 8. Example of Syntactic Rules.

(4)는 C1 · V1 · C2에 대응하는 각 자소의 종류를 규정하고, 규칙(5), (6), (7)은 실제 각 자소의 스트로-크 구성관계를 나타낸다. 또한 규칙(1)의 (check-bat)은 C2가 C1 · V1 · C2 구조의 종성인가 아닌가를 조사하는 함수이다. 규칙(6)의 (chec-a)는 그룹스트로-크·코-드 ss와 ee와의 사이에 가상 스트로-크의 방향

코-드  $hh_1$ ,  $th_1$ 가 각각 (7,3)인가를 조사하는 함수이다. 구문해석기는 그림 8의 생성규칙을 적용하므로써 그룹스트로-크·코-드열(ss rd ee ss ee ee dr)을 생성가능한 「만」, 「문」, 「단」, 「말」중에서 「만」을 임의적으로 결정한다. 이와 같이 보강문맥자유문법을 도입하므로써 소수의 생성규칙으로 다양한 모양의 한글을 인식할 수 있다.

구문해석기에 입력된  $M^*$ 가 생성규칙에 의해 도출가능하다면, 입력된 미지의 패턴은 생성규칙이 규정하는 한글에 속한다고 판정되며, 이에따라 생성규칙의 의미 해석부가 얽혀져, 입력문자의 자소명이 출력된다. 그림 9는 「만」의 구문해석 과정과 인식결과로서 추출된 자소명을 나타낸 것으로, < >와 [ ]부분은 각각 생성규칙과 자소명에 관계된 기술이다. (interp-HANGEUL (intero 'C1) (interp 'V1) (interp 'C2))는 함수(interp 'C1), (interp 'V1), (interp 'V2)를 평가하는 함수이다. (interp 'C1)은 C1의 의미해석부인 (i (interp 'MIEUM)를 평가하고 (interp 'MIEUM)은 C1의 의미해석부를 평가하여 'm을 결과값으로 가진다. 함수 interp-HANGEUL의 인수를 평가한 결과는 그 자소의 자소명으로 얻어진다. 그 자소명이 미지의 입력 문자를 인식한 결과이다.

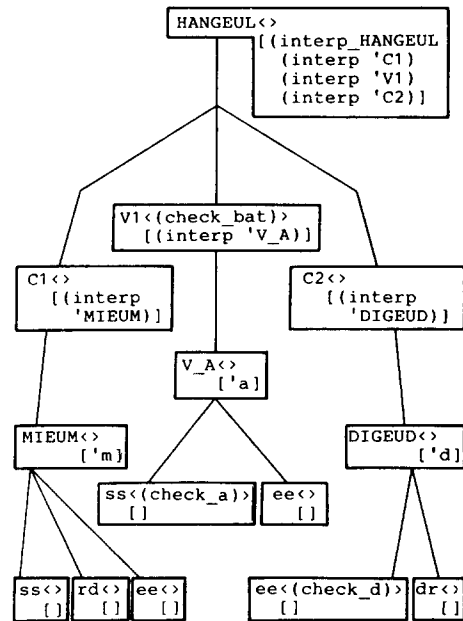


그림 9. 구문 해석기와 인식결과  
Fig. 9. A Parsing Tree and the Result of Recognition.

V. 인식실험

1. 인식실험

인식실험에 사용할 문법을 설계하기 위하여 일반적으로 많이 사용되는 1316자의 한글을 선정하여 학습샘플로 사용하였다. 이 학습샘플을 사용하여 표3에 나타난 바와 같은 애매성이 발생치 않도록 생성규칙 적용조건을 수정을 되풀이 하여 문법을 설계하였다. 또한, 획수 및 필순의 변동이나 문자의 변형등을 가능한 허용하도록 생성규칙을 작성했다.

이상의 방법으로 설계한 생성 규칙을 사용하여 미지 샘플에 대해서, 인식실험을 수행하였다. 인식실험에 사용된 미지의 입력문자는, 학습샘플 1316자 중에서 다양한 구조를 가지는 250자를 선정, 이를 5명으로 하여금 각각 쓰도록 하여 얻은 총 1250자이다. 필기시에는 되도록 정성들여 쓰며, 획과 획은 흘러쓰지 않도록 필기자에게 지시하였다.

문자는 태블릿 상의 3 cm × 3 cm의 테두리 안에 필기하도록 하였다. 문자의 표현정도는 64×64이며, 구문 해석에 사용한 생성규칙의 수는 총 148개였다.

2. 인식 결과와 검토

미지의 입력에 대하여 인식실험을 행한 결과를 표 4에 나타낸다. 정인식을 포함하여 애매성이 발생한 경우는, 생성규칙 적용조건을 좀더 세밀하게 기술함으로써 애매성을 해소할 수 있기 때문에, 보다 높은 인식율을 얻을 수 있다고 생각된다. 정인식을 포함하지 않은 오인식은 발생하지 않았기 때문에, 후처리를 하므로써 애매성의 해소가 가능하리라 생각된다. 그림10에 정인식한 문자의 예를, 그림11에는 리젝트된 문자를 포함하는 오인식의 예를 나타낸다. 오인식의 원인은 다음과 같다.

그림11(a)의 경우, 제2 스트로크의 hebd와 제3 스트로크의 head의 접속부분이 너무 많이 떨어져 있기 때문에, 부분패턴「능」(dr ee cc)을 「ㅎ」(ss ee cc)의 변형패턴으로 인식한 때문이다. 그림11(b)의 경우는 제7, 제8 스트로크에 의해 구성된 부분패턴이「니」



그림10. 정인식의 예  
Fig. 10. Examples of Correct Recognition.

	input pattern	recognition results
(a)		
(b)		
(c)		reject

그림11. 오인식의 예  
Fig. 11. Examples of Ill Recognition.

(ss dr)으로 변역되었다. 그러나, 「니」(ss dr)는 「나」(ss ee)의 변형패턴이기 때문에 정인식되지만, 제7 스트로크-크의 몸체와 제8 스트로크-크의 head가 너무 떨어져 있어, 제8 스트로크-크를 입력문자의 종성으로 판단하였기 때문에 오인식이 발생하였다. 그림11(c)는 제2 스트로크-크의 수직부분과 제3 스트로크-크의 head가 너무 떨어져 있기 때문에 리젝트되었다.

VI 결 론

그룹스트로크·코·드연만을 입력하여 구문해석을 수행할 경우, 애매성이나 리젝트가 빈번히 발생한다. 이를 해소하기 위하여 보강문맥자유문법에서는, 생성규칙 적용조건을 도입하므로써 문자를 구성하는 부분패턴간 및 스트로크간의 위치관계를 세밀하게 기술하였다.

본 논문은, 보강문맥자유문법을 이용한 온라인 한글 인식 방법에 대하여 제안한 것으로, 다음의 특징이 있다.

표 4. 인식실험결과  
Table 4. Results of Recognition Experiments.

정 인 식	99.04 (%)
정인식을 포함하는 애매한 인식	0.48 (%)
오 인 식	0 (%)
리 제 크 트	0.40 (%)
다단 번역에서 그룹 스트로크·코·드 추출의 실패	0.08 (%)

- (1) 동일 그룹스트로-크·코-드열이 생성되는 다수의 문자에 대하여 애매성이 발생하지 않는다.
- (2) 부분패턴간의 위치관계를 상세히 조사하기 때문에 부분패턴 자체의 변형 허용도가 높아진다.
- (3) 생성규칙 적용조건에 의해 구문해석의 실패가 조기에 확인될 수 있으므로, 구문해석에 소요되는 시간이 단축된다.
- (4) 생성규칙 적용조건의 도입으로 종래의 방법에 비하여 생성규칙의 수가 적으며, 인식대상 문자의 제한도 없어졌다.

금후의 연구과제로, 본 수법을 변형이 심한 흘림체 한글의 인식에 적용할 수 있도록 개선할 예정이다.

#### 參 考 文 獻

- [1] “흘림체 문자등 筆記制限을 완화시키는 방향으로 나아가는 on-line 筆記體 漢字認識” 日經エレクトロニクス, 12-05, pp. 115-133 (1983. 12).
- [2] 小高, 若松, 橋本: “on-line 筆記體 文字認識 裝置,” 電子通信 學會論文誌 (D), J65-D, 8, pp. 951-958, 1982. 8.
- [3] 古田, 迫江: “Stack DP Matching에 의한 on-line 筆記體 文字認識,” 日本電子通信學會技術研究報告, PRL 83-29, 1983. 9.
- [4] 小高, 若原, 增田: “筆順에 依存하지 않는 on-line 筆記體 文字認識 알고리즘,” 電子通信學會論文誌 (D), J65-D, 6, pp. 679-686, 1982. 6.
- [5] 若原, 梅田: “選托的 Stroke 결합에 의한 劃數에 依存하지 않는 on-line 文字認識,” 日本電子通信學會技術研究報告, PRL82-14, 1982. 6.
- [6] 若原, 梅田: “Stroke 결합정보를 이용한 on-line 흘림문자 분류법,” 日本電子通信學會技術研究報告, PRL83-30, 1983. 9.
- [7] 藤, 市原: “흘림체 문자를 對象으로한 on-line 文字認識-筆壓情報의 이용가능성에 대하여,” 日本電子通信學會技術研究報告, PRL83-30, 1983. 9.
- [8] 김태균, 安居院, 中嶋: “多段翻譯에 의한 on-line 한글의 認識,” 日本電子通信學會論文誌 (D), J68-D, 2, pp. 177-184, 1985. 2.
- [9] 김태균, 安居院, 中嶋: “屬性文法에 의한 흘림체 한글의 on-line 認識,” 日本電子通信 學會論文誌 (D), J68-D, 4, pp. 797-804, 1985. 4.
- [10] 박중욱, 이주근: “Shape Pattern에 의한 필기체 한글의 인식,” 대한전자공학회지, 제22권, 5호, pp. 1-5, 1985. 9.
- [11] 見: “擴張 LINGOL의 n進 tree로의 擴張,” 日本情報處理學會 計算言語學研究資料, 22-1, 1980.
- [12] 田中, 藤, 元吉: “自然言語를 위한 Programming System-擴張 LINGOL에 대하여-,” 日本電子通信學會論文誌 (D), J60-D, 12, pp. 1061-1068, 1984. 12.
- [13] Pratt, V. R.: “LINGOL-A Progress Report,” 41 JCAI, pp. 422-428, 1975.