

# 한글의 조합 규칙 표현 방법에 관한 연구

이 병 국<sup>†</sup> · 박 윤 범<sup>††</sup> · 이 완 규<sup>†††</sup>

## 요 약

조합형 설계의 한글 폰트 디자인은 한글의 모아쓰기 원리를 이용하여 일관된 모양을 유지하면서, 조합 가능한 활자체를 빠른 시간에 구성할 수 있는 장점이 있다. 그러나 균형감 있는 훌륭한 활자체를 개발하기 위해서는 설계할 자소의 수는 많아지고 조합 방법은 복잡해진다. 본 논문에서는 조합형 설계의 성공적인 결과를 위하여 효율적인 조합 규칙 표현 방법을 제안한다.

## An Effective Representation of Combination Rule for the Hangeul Typeface Design

Byung-Gook Lee<sup>†</sup> · Yun-Beom Park<sup>††</sup> · Woan-Kyu Lee<sup>†††</sup>

## ABSTRACT

The Hangeul font design system by using Hangeul automata is more economical than other system in terms of the time to generate whole Hangeul font set. In order to obtain good quality of Hangeul font by this system, it is known that more Hangeul jamo font primitives and combining rules are required. In this paper, we present an effective combining rules in order to generate Hangeul font set and to provide an integrated capabilities in Hangeul font design environment.

### 1. 서 론

오늘날 정보 산업화 시대의 다양한 요구로 인해 과거와 달리 이제는 한 별의 한글 활자체도 조합 가능한 글자를 모두 갖추어야 하는 시대가 되었다. 특히, 국제 표준 문자 코드(ISO 10646, UNICODE)의 채용에 따르는 운영 체제 및 응용 소프트웨어에서는 한글 11, 172자를 모두 처리할 수 있어야 하며, 이에 대한

활자체의 추가 확보 및 이들을 처리하기 위한 자원의 부담이 커지게 되었다. 이들을 과거처럼 글자 한 자 한 자를 다듬어 내던 설계 방식인 완성형 설계 방식으로는 글자 수를 모두 수용하면서도 다양한 활자체를 신속하게 만들어 내기는 매우 힘들게 되었다. 이를 해결할 수 있는 설계 방식은 한글의 모아쓰기 원리를 활용해서 자소를 조합하여 설계하는 조합형 설계 방식이다.

먼저 조합형 설계 방식에 관하여 알아보자. 조합형 설계는 활자체의 구성을 기준으로 할 때, 기본적으로 네모틀과 탈네모틀의 두 방향으로 나누어 질 수 있고, 중간의 여러 가지 절충 방향도 가능하다. 탈네모틀 조합형 활자체의 기원은 기계적 한계 안에서 한글의 모든 글자를 찍어내고자 했던 한글 기계식 수동

※이 논문은 동서대학교의 1995년도 교내 연구비 지원에 의하여 연구되었음.

† 정 회 원: 동서대학교 시스템공학부 응용수학과 전임강사

†† 정 회 원: 서원대학교 사범대학 수학교육과 전임강사

††† 정 회 원: 한국표준과학연구원 계측공학연구실 선임연구원

논문접수: 1995년 11월 16일, 심사완료: 1996년 7월 4일

락지기에서 비롯되어, 1984년 이상철의 <샘이 깊은 물체>, 1985년 안상수의 <안체>를 필두로 석금호의 <산들체>, 구성희의 <보체>, 한재준의 <한체>, 윤형기의 <윤체>등이 계속 발표되고 있으며 차츰 그 활용 영역도 넓어지고 있다. 네모틀을 조합형 활자체의 설계 자소 수는 초기에는 대체로 가장 단순한 3벌로 설계되었으나 글자 균형의 조정을 위해 점차 자소 수가 증가되는 경향을 보이고 있으며, 그 중에는 탈네모틀과 네모틀의 중간 성격을 띤 활자체들도 많이 개발되고 있다.

네모틀 방향의 조합형 활자체는 완성형 원도의 스케닝에 의해 만들어졌던 완성형 활자체 중에서 사용 빈도가 낮고 원도가 없는 활자들을 보충하는 방법으로 시작되어, 현재에는 활자체 제작의 초기부터 활자체의 기본 자소들을 분류하고 조합 규칙을 설정하여 이들을 수정 보완하면서 개발해 나가는 방법으로 발전하고 있다. 네모틀 조합형 설계의 기본적인 문제점은 자소 수와 조합 규칙이 단순하면 할 수록 네모틀 내의 글자 균형은 좋지 않으며, 반대로 네모틀 안의 글자 균형이 섬세하고 좋을 수록 그 자소의 수와 조합 규칙은 점점 더 복잡해져 가는 것이다. 이러한 조합형 설계의 성공적인 결과를 위해서는, 활자체 설계 전문가의 합리적 조합 규칙 설정 및 자소 설계와 함께 효율적인 활자체 설계 시스템이 뒷받침되어야만 한다.[1]

본 논문에서는 활자체 전문가가 작성한 조합 규칙을 쉽게 표현 및 수정할 수 있고 시스템에서 효율적으로 사용할 수 있는, 조합 규칙 표현 방법과 이들의 조합 알고리즘을 제안하고자 한다. 본 논문은 다음과 같이 구성된다. 2장에서는 한국어 정보처리에 있어 한글의 구조적 조합 특성을 이용한 한글 폰트 개발 사례에 대하여 설명하며, 3장에서는 효율적 조합 규칙 표현 방법과 조합 알고리즘에 대하여 제안한다. 그리고, 4장에서는 전체적인 평가와 앞으로의 연구 방향에 대하여 설명한다.

## 2. 한글 활자체 개발에서의 구조적 조합 특성 이용 사례

한글의 각 자소들은 나타나는 위치에 따라 초성, 중성, 종성으로 나누어지며 이들이 모여 하나의 음절을 이룬다. 이러한 한글의 구조적 조합 특성은 컴퓨

터를 이용한 한국어 정보처리의 여러 분야에서 유용하게 이용된다. 특히, 한글 활자체 개발에서의 이용 사례는 다음과 같다.

안은영외[6]에 의해 기본 획은 여러 개의 그래픽 프리미티브에 의하여 구성되고 이러한 기본 획이 모여 하나의 자소를 형성하고, 이를 초성12벌 중성4벌 종성4벌의 한글 조합 규칙에 따라 모여 하나의 글자를 형성하는 계층적인 방법이 제안되었으며, 이규영외[7]에 의해 특정한 활자체에 대한 날자의 유형 분석 방법과 분석된 유형 정보를 이용하여 초성, 중성, 종성별 날자 폰트를 설계하고, 설계된 473개의 날자 폰트로서, 조합 가능한 모든 한글 음절의 출력이 가능한 알고리즘과 조합형 방식의 폰트 개발 방법이 제안되었다. 그러나, 여기에서는 사용된 조합 규칙과 조합 알고리즘은 기본 자소의 유형 수를 축소 또는 확장하는 경우에 조합 알고리즘이 수정되어야 하는 단점이 있어 다양한 서체를 개발해야 하는 조합형 활자체 설계 시스템에 적용하기에는 문제점이 있다.

## 3. 자소 조합 규칙 표현 방법

한글을 구성하는 자소는 초, 중, 종성에 따라 분류해 보면 표 1의 한글 자소 순서표와 같이 67개의 자소가 얻어지는데 이들을 조합하면 총 11, 172(19\*21\*28)자의 글자가 얻어진다. 따라서 한글의 자소는 적으나 글자의 구성에 따라 자소의 모양이 복잡하게 변하게 된다. 각 자소는 조합되어 글자를 형성할 때 초성, 중성, 종성의 종류에 따라 그 모양, 크기, 위치가 달라지고, 이에 따르는 자소의 여러 가지 모양을 갖추어야 한다. 여기서 자소의 여러 가지 모양을 기본 자소(primitive)라 부르기로 한다.

### 3.1 기본 자소 코딩 방법

기본 자소 코드 값은 자소의 종류 및 여러 가지 모양에 대한 정보가 포함되어야 한다. 이를 위하여 표 1과 같이 전체 자소의 순서를 정하고, 각 기본 자소의 코드 값은 자소의 전체 순서 값에 100(각 자소의 여러 가지 모양 최대 개수)을 곱하고 같은 자소 중에서의 여러 가지 모양에 관한 순서값(설계자가 결정)을 더하여 코딩한다.

여기에서 자소의 전체 순서값 보다는 한글 로마자

<표 1> 한글 자소 순서표

<Table 1> Order Table of Hangui Phoneme

초성 자소	자소 순서	전체 순서	로마자 표기	초성 자소	자소 순서	전체 순서	로마자 표기	초성 자소	자소 순서	전체 순서	로마자 표기
ㄱ	1	1	G	ㅏ	1	20	A	ㄱ	1	41	g
ㄲ	2	2	GG	ㅑ	2	21	AE	ㄲ	2	42	gg
ㄴ	3	3	N	ㅓ	3	22	YA	ㄴ	3	43	gs
ㄷ	4	4	D	ㅕ	4	23	YAE	ㄷ	4	44	n
ㄸ	5	5	DD	ㅗ	5	24	EO	ㄸ	5	45	nj
ㄹ	6	6	L	ㅛ	6	25	E	ㄹ	6	46	nh
ㅁ	7	7	M	ㅜ	7	26	YEO	ㅁ	7	47	d
ㅂ	8	8	B	ㅠ	8	27	YE	ㅂ	8	48	l
ㅃ	9	9	BB	ㅡ	9	28	O	ㅃ	9	49	lg
ㅅ	10	10	S	ㅚ	10	29	WA	ㅅ	10	50	lm
ㅆ	11	11	SS	ㅜ	11	30	WAE	ㅆ	11	51	lb
ㅇ	12	12	"	ㅠ	12	31	OE	ㅇ	12	52	ls
ㅈ	13	13	J	ㅡ	13	32	YO	ㅈ	13	53	lt
ㅉ	14	14	JJ	ㅓ	14	33	U	ㅉ	14	54	lp
ㅊ	15	15	C	ㅕ	15	34	WEO	ㅊ	15	55	lh
ㅋ	16	16	K	ㅗ	16	35	WE	ㅋ	16	56	m
ㆁ	17	17	T	ㅛ	17	36	WI	ㆁ	17	57	b
ㅌ	18	18	P	ㅠ	18	37	YU	ㅌ	18	58	bs
ㅎ	19	19	H	ㅡ	19	38	EU	ㅍ	19	59	s
				ㅣ	20	39	YI	ㅑ	20	60	ss
				ㅣ	21	40	I	ㅓ	21	61	'
								ㅕ	22	62	j
								ㅗ	23	63	c
								ㅛ	24	64	k
								ㅜ	25	65	t
								ㅠ	26	66	p
								ㅡ	27	67	h

표기를 사용하면 가독성이 있어 제작자가 보다 편리하게 작성 및 수정할 수 있다(단, 로마자 표기를 전체 순서 값으로 변환하여 주는 프로그램을 별도로 작성하여야 함). 예를 들어 'ㄱ'의 첫 번째 모양의 기본 자소 코드 값은 101이며 로마자 표기는 G1이다. 또한, 섞임 모임인 경우 제작자에 따라 별도의 자소를 정의하여 디자인할 수 있다. 예를 들면 '과'자인 경우 'ㄱ'과 'ㅏ'의 모임에서 '고'와 'ㅑ'의 모임으로 디자인하는 경우 '고'의 자소를 자소 순서표 뒤에 정의하여 사용할 수 있다. 이는 'ㄱ'과 'ㅏ'로 나누어 디자인할 때보다 규칙성을 찾기 쉽고 경우의 수가 적어 이러한 편법을 사용하기도 한다[8][9].

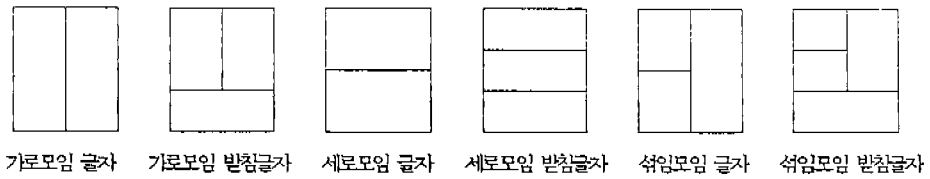
### 3.2 조합 규칙 표현 방법

한글 활자체 한 벌을 제작하기 위하여 한글의 구조적인 특성을 이해하고, 설계자의 의도에 따라 자소들을 종류별로 분류해 나갈 것이다. 한글의 특성상 모음(중성)을 중심으로 자음(초성, 종성)의 위치와 모

양이 확연하게 달라지므로 그림 1과 같이 한글의 6형식으로 분류하여 디자인하는 것이 자연스럽다. 한글은 초성과 중성, 종성의 3별 구조로 모아쓰기를 하는 글자 체계로서, 조합 규칙이란 다음과 같은 특성으로 구성되어져야 한다.

- 첫째, 글자를 이루는 초성과 중성, 중성이 서로 상관 관계를 가지면서 각자의 모양, 즉 기본 자소를 결정하여야 한다.
- 둘째, 설계자가 조합 규칙을 쉽게 수정 가능하도록 표현하여야 한다.
- 셋째, 시스템 구성이 효율적 이어야 한다.

이상과 같은 조합 규칙을 구성하여 개발 도구에서는 각 자소의 기본 자소를 디자인하여 나갈 때 현재에 디팅하고 있는 기본 자소와 관계를 가지는 글자들을 제작자에게 보여 주어 전체적인 균형미와 통일감을 주도록 도와주며, 경우에 따라서는 조합 규칙을



(그림 1) 한글 6형식 분류  
(Fig. 1) Assortment Table by the Forms of Hangul Character

초성	①	ㅍ	ㅑ	ㅓ	ㅕ	ㅗ	ㅛ	ㅜ	ㅠ	ㅡ	ㅝ	ㅞ	ㅟ	ㅠ							
	②	ㅑ	ㅓ	ㅕ	ㅗ	ㅛ	ㅜ	ㅠ	ㅡ	ㅝ	ㅞ	ㅟ	ㅠ								
	③	ㅓ	ㅕ	ㅗ	ㅛ	ㅜ	ㅠ	ㅡ	ㅝ	ㅞ	ㅟ	ㅠ									
	④	ㅕ	ㅗ	ㅛ	ㅜ	ㅠ	ㅡ	ㅝ	ㅞ	ㅟ	ㅠ										
	⑤	ㅗ	ㅛ	ㅜ	ㅠ	ㅡ	ㅝ	ㅞ	ㅟ	ㅠ											
중성	①	ㅏ	ㅑ	ㅓ	ㅕ	ㅗ	ㅛ	ㅜ	ㅠ	ㅡ	ㅝ	ㅞ	ㅟ	ㅠ							
	②	ㅑ	ㅓ	ㅕ	ㅗ	ㅛ	ㅜ	ㅠ	ㅡ	ㅝ	ㅞ	ㅟ	ㅠ								
	③	ㅓ	ㅕ	ㅗ	ㅛ	ㅜ	ㅠ	ㅡ	ㅝ	ㅞ	ㅟ	ㅠ									
	④	ㅕ	ㅗ	ㅛ	ㅜ	ㅠ	ㅡ	ㅝ	ㅞ	ㅟ	ㅠ										
종성	①	ㄱ	ㅋ	ㆁ	ㄴ	ㄷ	ㅌ	ㄹ	ㅍ	ㅑ	ㅓ	ㅕ	ㅗ	ㅛ	ㅜ	ㅠ	ㅡ	ㅝ	ㅞ	ㅟ	ㅠ
	②	ㄴ	ㄷ	ㅌ	ㄹ	ㅍ	ㅑ	ㅓ	ㅕ	ㅗ	ㅛ	ㅜ	ㅠ	ㅡ	ㅝ	ㅞ	ㅟ	ㅠ			

(그림 2) 기본 별수 (초성 5별, 중성 4별, 종성 2별)  
(Fig. 2) The Set of Primitives

초성	중성	종성
① ㅍ ㅑ ㅓ ㅕ ㅗ ㅛ ㅜ ㅠ ㅡ ㅝ ㅞ ㅟ ㅠ	① ㅏ ㅑ ㅓ ㅕ ㅗ ㅛ ㅜ ㅠ ㅡ ㅝ ㅞ ㅟ ㅠ	① ㄱ ㅋ ㆁ ㄴ ㄷ ㅌ ㄹ ㅍ ㅑ ㅓ ㅕ ㅗ ㅛ ㅜ ㅠ ㅡ ㅝ ㅞ ㅟ ㅠ
② ㅑ ㅓ ㅕ ㅗ ㅛ ㅜ ㅠ ㅡ ㅝ ㅞ ㅟ ㅠ	② ㅏ ㅑ ㅓ ㅕ ㅗ ㅛ ㅜ ㅠ ㅡ ㅝ ㅞ ㅟ ㅠ	
③ ㅓ ㅕ ㅗ ㅛ ㅜ ㅠ ㅡ ㅝ ㅞ ㅟ ㅠ	③ ㅏ ㅑ ㅓ ㅕ ㅗ ㅛ ㅜ ㅠ ㅡ ㅝ ㅞ ㅟ ㅠ	
④ ㅕ ㅗ ㅛ ㅜ ㅠ ㅡ ㅝ ㅞ ㅟ ㅠ	④ ㅏ ㅑ ㅓ ㅕ ㅗ ㅛ ㅜ ㅠ ㅡ ㅝ ㅞ ㅟ ㅠ	
⑤ ㅗ ㅛ ㅜ ㅠ ㅡ ㅝ ㅞ ㅟ ㅠ	⑤ ㅏ ㅑ ㅓ ㅕ ㅗ ㅛ ㅜ ㅠ ㅡ ㅝ ㅞ ㅟ ㅠ	
이하 생략		

(그림 3) 가로 모임형 글자의 조합 규칙 예제  
(Fig. 3) The Example of Combination Rule for Vertical Assembly Character

수정하면서 작업할 수 있어야 한다. 또한 전체적인 글자 혹은 제작자가 미리 만들어 놓은 문장을 현재 개발하고 있는 폰트로 디스플레이 하여 영문 및 한문과의 조화도 염두에 두고 제작할 수 있는 환경을 제공하여야 한다.

설계하는 활자체에 따라 조합 규칙은 여러 가지가 있을 수 있으나, 여기에서는 이원규의[8][9]에서 한글 폰트 개발을 위하여 사용한 조합 규칙중 가로 모임형 글자의 일부분을 발췌하여 설명하고자 한다. 먼저 활자체 제작자는 자신의 의도에 따라 초성과 중성의 범수를 그림2와 같이 정할 것이며, 또한 그들의 조합 규칙을 그림 3과 같이 표현할 수 있다.

그림 3의 조합 규칙을 가장 쉽게 생각할 수 있는 표현 방법은 그림 4와 같이 조합 가능한 모든 글자에 관하여 각 자소의 여러 가지 모양에 관한 정보를 가지고 있는 방법으로, 시스템에서 구현하기는 가장 쉬우나 메모리를 많이 차지할 뿐만 아니라 조합 규칙 구성 및 수정이 힘들며, 같은 기본 자소를 가지는 글자들을 보여 주기에는 비효율적인 표현 방법이다. 그러나 조합형 코드와 인덱스가 일대일 대응 관계에 있어 규칙을 빠르게 찾을 수 있고(Index = chosung\_code\*21\*27 + jungsung\_code\*27 + jongsung\_code), 조합 알고리즘도 간단하다.

본 논문에서 제안하는 표현 방법은 그림 3의 조합

	Index	조 성	중 성	중 성
가 각 자 소 간	0	G1 ( 7 첫번째모양, 101)	A1 ( 1첫번째모양, 2001)	NULL
	1	G5 ( 7 다섯번째모양, 105)	A2 ( 1두번째모양, 2002)	g1 ( 1첫번째모양, 4101)
	2	G5 ( 7 다섯번째모양, 105)	A2 ( 1두번째모양, 2002)	gg1( 1첫번째모양, 4201)
	3	G5 ( 7 다섯번째모양, 105)	A2 ( 1두번째모양, 2002)	gs1( 1첫번째모양, 4301)
나	4	G5 ( 7 다섯번째모양, 105)	A3 ( 1세번째모양, 2003)	ni ( 1첫번째모양, 4401)
			NULL	
형	11171	N1 ( 1 첫번째모양, 301)	A1 ( 1첫번째모양, 2001)	NULL
		H5 ( 8 다섯번째모양, 6705)	I4 ( 1네번째모양, 4001)	h2 ( 8두번째모양, 6702)

(그림 4) 가로 모임형 글자의 조합 규칙 표현 방법 1  
(Fig. 4) The First Representation of Combination Rule

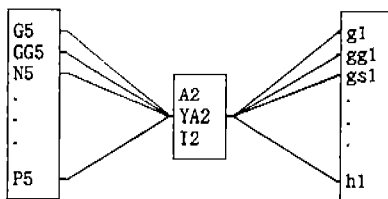
<가로 모임 글자>	
G1 GG1 N1 D1 DD1 L1 M1 B1 BB1 S1 SS1 *1 J1 JJ1 C1 K1 T1 P1 H1 A1 YA1 I1	
G1 GG1 N1 D2 DD2 L2 M1 B1 BB1 S1 SS1 *1 J1 JJ1 C1 K1 T2 P2 H1 E01 YE01	
G3 GG3 N3 D3 DD3 L3 M3 B3 BB3 S3 SS3 *3 J3 JJ3 C3 K3 T3 P3 H3 E02 YE02	
G3 GG3 N3 D4 DD4 L4 M3 B3 BB3 S3 SS3 *3 J3 JJ3 C3 K3 T4 P4 H3 AE1 YAE1	
	E2 YAE2
	E1 YE1
	E1 YE1
	E2 YE2
<가로 모임 받침 글자>	
G5 GG5 N5 D5 DD5 L5 M5 B5 BB5 S5 SS5 *5 J5 JJ5 C5 K5 T5 P5 A2 YA2 I2	
g1 gg1 gs1 nj1 nh1 d1 m1 b1 bs1 *1 j1 k1 p1 l1 lg1 lm1 lb1 ls1 lt1 lh1cl t1 h1	
G5 GG5 N5 D5 DD5 L5 M5 B5 BB5 S5 SS5 *5 J5 JJ5 C5 K5 T5 P5 A3 YA3 I3 n1 s1	
	H5 A3 YA3 I3 n2 s2
	H5 A4 YA4 I4
g1 gg1 gs1 nj1 nh1 d1 m1 b1 bs1 *1 j1 k1 p1 l1 lg1 lm1 lb1 ls1 lt1 lh1cl t1 h1	
이하 생략	

(그림 5) 가로 모임형 글자의 조합 규칙 표현 방법 2  
(Fig. 5) The Second Representation of Combination Rule

규칙을 있는 그대로 표현하는 방법으로, 이를 앞에서 설명한 코딩 방법으로 표기하면 그림 5와 같이된다.

위의 표현 방법은 활자체 설계자의 일반적인 조합 규칙 표현 기법과 동일하여 작성하기가 용이하며, 어떤 활자체 조합규칙도 위와 같은 방법으로 표현할 수 있다. 또한, 같은 기본 자소를 이용하는 글자, 같은 모임형의 글자 등을 즉각적으로 보여줄 수 있게 되어 설계시 많은 도움이 될 것이다. 조합형 활자체 디자인에서 하나의 기본 자소를 디자인 할 때 이와 관계에 있는 다른 자소들과의 조합이 아주 중요한 영향을 미치며, 디자인 시스템에서는 이들을 보여줄 필요가 있다. 현재 에디팅하고 있는 기본 자소 코드 값을 조합 규칙 테이블 배열에서 찾아, 현재 위치에서 앞방향의 내림순으로 가장 작은 코드 값에서 뒷방향의 올

림순으로 가장 큰 값까지의 소행렬을 찾은 다음 이를 초성, 중성 그리고 종성의 세그룹으로 나누어 다른 자소의 관계를 효율적으로 보여줄 수 있다. 만약 현재 에디팅하고 있는 기본 자소가 가로 모임 받침 글자의 “ㅏ” 두 번째 모양(YA2) 일 때 다음 그림 6과 같은 조합으로 글자를 구성한다.



(그림 6) 조합 방법  
(Fig. 6) The Method of Combination

```

조합 알고리즘
조합 규칙 테이블의 행렬 : combination_table
조합 규칙 테이블의 코드값 / 100(division operator) : 기본 자소의 코드값
조합 규칙 테이블의 코드값 % 100(remainder operator) : 기본 자소의 모양값

Input :
chosung_code 초성자소의 코드값           예) 각 =
jungsung_code 중성자소의 코드값         7 (1)
jongsung_code 종성자소의 코드값        ㅏ (1)

Output :
chosung_bul 초성자소의 모양값           ㅏ (1)
jungsung_bul 중성자소의 모양값
jongsung_bul 종성자소의 모양값

WHILE
IF combination_table[i]/100 == chosung_code
chosung_bul = combination_table[i]%100
index i를 현재 검색중인 그룹의 중성코드 시작위치로 옮긴다.
WHILE 현재그룹의 중성에서 (19 < combination_table[i]/100 <= 19+21)
IF combination_table[i]/100 == jungsung_code+19
jungsung_bul = combination_table[i]%100
IF jongsung_code == 0
jongsung_bul = 0
RETURN
ENDIF
index i를 현재 검색중인 그룹의 중성코드 시작위치로 옮긴다.
WHILE 현재그룹의 종성에서 (19+21 < combination_table[i]/100 <= 19+21+27)
IF combination_table[i]/100 == jongsung_code+19+21
jongsung_bul = combination_table[i]%100
RETURN
END IF
END WHILE
END IF
END WHILE
END IF
END WHILE
END WHILE
    
```

(그림 7) 표현 방법 2를 위한 조합 알고리즘  
(Fig. 7) Combination Algorithm for the Second Representation

조합 규칙은 단순 배열(one dimensional array)에 기본 자소의 코드 값들을 저장하며, 이들은 한글 6형식 분류에 따라 6개로 나누어 저장하는 것이 조합 규칙을 찾을 때 효율적이다. 조합 규칙을 표현할 때 전체 자소 순서 값대로 표현하면, 기본 자소 코드의 특성에 따라 올림순으로 배열되어 있는 소행렬(그룹)의 모임으로 나눌 수 있으며, 이들의 그룹 중에서 찾고자 하는 글자의 초성, 중성, 종성의 코드값(기본 자소 코드값/100=자소의 코드값)이 모두 존재하면 이들의 기본 자소 코드 값에서 조합 규칙을 찾아낸다.

#### 4. 결 론

본 논문에서는 제작자가 조합 규칙을 쉽게 작성, 수정할 수 있고 조합형 폰트 디자인 시스템에서 효율적인 표현 방법과 알고리즘을 제안하였다. 제작자는 디자인하고자 하는 한글 활자체의 일정한 모아쓰기 규칙을 본 논문에서 제안한 기본 자소 코딩방법과 표현 방법으로 조합 규칙을 외부 파일로 구성하여 디자인 시스템을 사용하며, 경우에 따라서는 조합 규칙 파일을 수정하면서 조합형 활자체를 디자인 할 수 있다. 기본 자소의 유형 수를 축소 또는 확장하는 경우에도 조합 규칙 파일만 수정하면 된다.

향후 연구 방향은 다음과 같다. 기존에 만들어진 완성형 설계방식의 활자체들은 개발자의 의도에 따라 한 자 한 자씩 개인적인 미적 감각에 의해 다듬어진 글자로서 대개 2,350여자만 개발되어 있고 나머지 활자체들은 개발되어 있지 않다. 이들을 UNICODE 환경에서 사용하기 위하여 나머지 활자체들을 추가하여야만 한다. 그러나 나머지 활자체 8,800여자를 기존의 완성형 활자체 설계 방법으로는 추가하기에는 비효율적이고 많은 시간과 노력이 필요하다. 그리하여 이들 완성형 글꼴 2,300여자의 자소 조합 규칙을 찾아내고, 이를 한글 활자체의 기본적인 조합 규칙과 비교 분석하여 조합 가능한 나머지 활자체를 추가하는 방법에 관한 연구가 계속 이루어져야 하겠다.

#### 참 고 문 헌

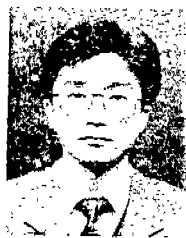
[1] 김 진평, "한글 활자체의 조합형 설계 가능성 연구", 제4회 한글 및 한국어 정보처리 학술 발표

논문집, pp.293-300, 1992.

- [2] 김 진평, "향후 한글 폰트(Font)의 발전 방향", 제4회 한글 및 한국어 정보처리 학술 발표 논문집, pp.17-22, 1992.
- [3] 김 정옥, 최 동혁, 강 식건, 박 건작, 박 규태, "계층 구조 획 추출에 의한 한글 인식", 대한 전자공학회 논문집-B, 제28권 제8호, pp.1-9, 1991.
- [4] 성 태진, 방 승양, "문자 조합 규칙 학습에 의한 온라인 한글 인식", 한국정보과학회 논문지, 제20권 제3호, pp.305-316, 1993.
- [5] 이 규영, 이 상범, "한국어 음운 변동 처리를 위한 효율적인 Rule Base System의 구성", 대한 전자공학회 논문집-B, 제28권 제12호, pp.9-18, 1991.
- [6] 안 은영, 조 형제, "기본 획 합성에 의한 한글 글꼴 생성", 한국정보과학회 논문지, 제21권 제4호, pp.648-658, 1994.
- [7] 이 규영, 김 규식, 이 상범, "한글 음절의 유형 분석 정보에 의한 낱자 폰트의 설계", 대한 전자공학회 논문집-B, 제29권 제9호, pp.17-26, 1992.
- [8] 이 원규의, 한글 폰트 개발, KRISS-92-127-1R, 체신부, 1992.
- [9] 이 원규의, 한글 폰트 개발, KRISS-93-148-1R, 체신부, 1993.
- [10] 정 용주, "한글 문자를 위한 컴퓨터 자형 설계 시스템", 한국정보과학회 논문지, 제11권 제3호, pp.153-163, 1984.
- [11] 김 차종, 강 한중, 이 수연, "다목적 폰트 제작 시스템", 한국정보과학회 논문지, 제16권 제5호, pp.422-433, 1989.
- [12] 이 현표, 양 순성, 황 교철, 이 군하, "한글 문서에서의 낱자 분리 알고리즘", 제1회 한글 및 한국어 정보처리 학술 발표 논문집, pp.203-208, 1989.
- [13] 이 문형, 김 기두, "새로운 한글 코드 Truecode의 개발과 응용", 대한 전자공학회 논문집-B, 제30권 제5호, pp.43-51, 1993.
- [14] 김 성락, 남 시병, 이 상범, "음절 단위를 이용한 한국어 연속 단어 인식", 대한 전자공학회 논문집-B, 제28권 제1호, pp.9-14, 1991.
- [15] 김 형래, 박 인갑, 서 동필, 김 예능, "CFG방법을 이용한 필기체 한글에서의 자소추출과 인식에 관한 연구", 대한 전자공학회 논문집-B, 제29권 제9

호, pp.8-15, 1992.

- [16] 김 중렬, 정 호선, 이 우일, “특정점 추출에 의한 한글 문자 인식 및 전처리용 신경 칩의 설계”, 대한 전자공학회 논문집, 제27권 제6호, pp.114-121, 1990.
- [17] 강 현철, 최 동혁, 이 완주, 박 규태, “원소 변환을 이용한 한글 패턴의 구조 분석”, 대한 전자공학회 논문집, 제26권 제12호, pp.61-69, 1989.
- [18] 한 재준, “한글 세벌체의 우수성 표현”, 제4회 한글 및 한국어 정보처리 학술발표 논문집, pp. 301-308, 1992.
- [19] 이 현주, 박 동인, “한글의 시각적 이미지 다양화에 관한 연구”, 제4회 한글 및 한국어 정보처리 학술발표 논문집, pp.591-599, 1992.



**박 윤 범**

1986년 서울대학교 사범대학 수학교육과 (이학사)  
 1988년 한국과학기술원 응용수학과 (이학석사)  
 1994년 한국과학기술원 수학과 (이학박사)  
 1988년~1991년 (주)신도리코 기술연구소 주임연구원

1995년~현재 서원대학교 사범대학 수학교육과 전임강사

관심분야: CAGD, CAD/CAM, 수치해석



**이 완 규**

1978년 한양대학교 정밀기계과 (공학사)

1988년 일본 고베대학 생산공학과 (공학박사)

1988년~현재 한국표준과학연구원 계측공학연구실

관심분야: 한글정보처리, CAD/CAM, 계측자동화

**이 병 국**

1987년 연세대학교 수학과 졸업 (이학사)

1989년 한국과학기술원 응용수학과 (이학석사)

1993년 한국과학기술원 수학과 (이학박사)

1993년~1995년 (주)데이콤 종합연구소 선임연구원

1995년~현재 동서대학교 시스템공학부 응용수학전공 전임강사

관심분야: 비모수통계학, 한글정보처리, CAGD.

