

# 훈민정음 원리에 기반한 자소형 코드와 최적 폰트 코드의 일치성에 관한 연구

김경욱<sup>o</sup> 변정용  
 동국대학교 컴퓨터멀티미디어학부  
 {kakar36, byunjy}@dongguk.ac.kr

## A study on the Conformity of Jaso-Type Code and Optimum Font Code Based On Hunminjeongeum Principle

Kim Kyoungwook<sup>o</sup> Byun Jeongyong  
 Dept. of Computer and Multimedia, Dongguk University

### 요 약

훈민정음 창제당시의 모든 한글 음절인 약 399억 음절을 표현하기 위해서 자소형 코드만을 입력하여 이를 조합 음절 형태로 구성하고, 최적 폰트 코드를 출력함으로써 모든 한글을 표현할 수 있었다. 하지만, 이러한 최적 폰트 코드는 모든 한글을 표현하는 데에만 치중하였지만, 키보드로부터 입력되는 자소형 코드와 화면에 음절을 구성하며 출력되는 폰트 코드의 일치하지 않는 문제점이 발생하였다. 이를 해결하기 위해서 본 연구에서는 실험을 통하여 입력된 자소형 코드를 파일입출력을 이용하여 파일로 저장 후 이를 본 연구에서의 옛한글 입력기의 폰트 변환 모듈을 사용하여 음절을 표현하여, 자소형 코드와 폰트 코드의 일치함을 보이고자 한다.

키워드: 훈민정음, 자소형 코드, 폰트 코드, 최적폰트

환하는 오토마타를 사용하여 입력 코드를 폰트 코드로 변환하여 음절을 표현하고자 한다.

### 1. 서 론

이전 연구 [1,2,3,4]에서 훈민정음 창제원리에 맞는 자소형 코드기반의 최적 폰트는 유니코드[5] 영역 중에서 사용자 정의 영역(E000~F8FF)의 코드를 할당하여 1,651자를 배치하였다. 이에 훈민정음 창제당시의 모든 한글 음절인 약 399억 음절을 표현할 수가 있었다. 하지만 이러한 최적 폰트 코드는 한글 음절을 표현하는 데에만 중점을 두고 있다. 이는 컴퓨터상에서의 키보드 입력을 통한 입력 코드인 자소형 코드와 음절을 표현하기 위한 최적 폰트 코드 사이에는 코드가 일치하지 않는 문제점이 발생하고 있다.[6,7,8] 즉, 입력 시에는 낱자소인 자소형 코드로 입력이 되지만, 컴퓨터상에서의 표현을 위한 폰트 코드는 사용자 정의 영역에 할당된 최적 폰트 코드가 출력되어 서로 코드가 일치하지 않고 있는 실정이다. 최적 폰트 코드는 자소형 코드로 일치시켜주어야 한다.[9,10] 이는 자소형 코드는 훈민정음 원리에 맞는 낱자소 코드이기 때문이다.

본 논문에서는 이러한 문제점을 해결하기 위해서 입력 코드와 폰트 코드를 일치시키고자 한다. 그러기 위해서 실험을 통하여 폰트 코드가 아닌 입력 코드인 자소형 코드만을 저장 후 이를 표현 시에는 최적 폰트 코드로 변

### 2. 기존 연구

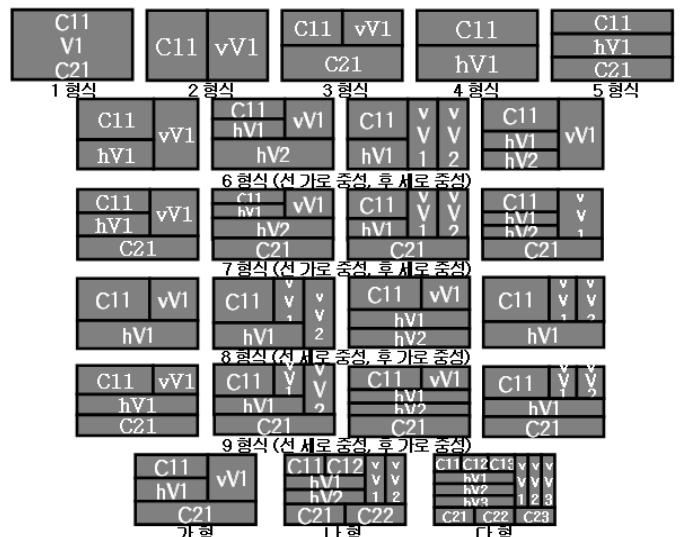


그림 1 한글 음절 9가지 형식과 3가지 형[2,3]

C11 : 초성 1, C12 : 초성 2, C13 : 초성 3  
 hV1 : 가로중성 1, hV2 : 가로중성 2, hV3 : 가로중성 3  
 vV1 : 세로중성 1, vV2 : 세로중성 2, vV3 : 세로중성 3  
 C21 : 중성 1, C22 : 중성 2, C23 : 중성 3

훈민정음 창제당시의 모든 한글을 표현하는 데에 필요한 최적 폰트는 이전 연구에 설계한 그림1과 같이 한글 음절을 구성하기 위해서 9가지 형식과 3가지 형으로 나뉠 수 있다.[2,3]

[표 1] 최적 폰트 코드 총 개수[1]

형식	초성 가	초성 나	초성 다	중성 가	중성 나	중성 다	중성 가	중성 나	중성 다	총 개수
1형식	23	28	29	11	17	23	17	39	64	251
2형식	23	28	29	5	5	5				95
3형식	23	28	29	5	5	5	17	39	64	215
4형식	23	28	29	6	12	18				116
5형식	23	28	29	6	12	18	17	39	64	236
6형식	23	28	29	0	11	34				125
7형식	23	28	29	0	11	34	17	39	64	245
8형식	23	28	29	0	11	33				124
9형식	23	28	29	0	11	33	17	39	64	244
합계										1,651

이에 훈민정음 창제당시의 모든 한글인 약 399억 음절을 표현하기 위한 최적 폰트의 폰트 코드 개수는 음절 구성의 형식과 형을 통하여 총 1,651자가 필요하다.(표 1 참고) 이는 낱자소인 초성, 중성, 종성의 형식과 형에 따른 글꼴 모양만으로 모든 한글을 표현할 수가 있다.[1]

### 3. 자소형 코드와 최적 폰트 코드

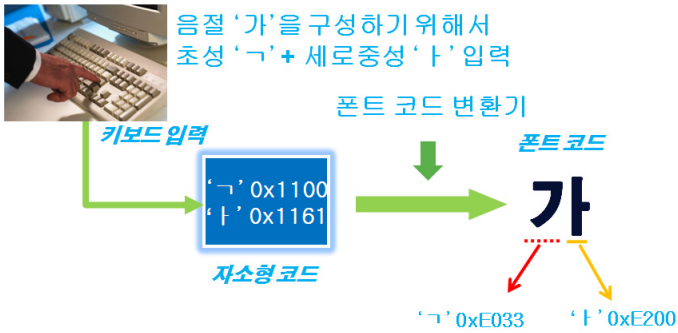


그림 2 자소형 코드와 최적 폰트 코드

키보드 입력 시에 입력되는 자소형 코드는 유니코드의 조합형 코드 중 자모 코드를 제외한 자소 코드의 초성, 중성, 종성이 입력된다. 이에 입력된 자소형 코드를 글꼴로 표현하기 위해서는 폰트 코드로 변환을 하여야 한다. 그러기 위해서는 음절 구성을 위한 오토마타를 통해 한글 음절 9가지 형식과 3가지 형에 따라 형식과 형이 정해지면 이에 대한 폰트 코드 변환기를 통해 폰트 코드로 변환되어 출력된다.

그러나 그림2에 보듯이 음절 '가'를 구성하기 위해서 초성 'ㄱ'과 세로중성 'ㅏ'를 입력하였다. 하지만, 입력된 자

소형 코드는 'ㄱ'은 0x1100, 'ㅏ'는 0x1161 인 반면, 폰트 코드는 'ㄱ'은 0xE033, 'ㅏ'는 0xE200으로 자소형 코드와 최적 폰트 코드가 일치하지 않고 있다. 이를 해결하기 위해서 자소형 코드와 폰트 코드를 일대일 대응시키면 되겠지만, 한글 음절 구성은 풀어쓰기 형태가 아닌 모아쓰기 형태라서 음절 구성 형식과 형에 따라 각 글꼴 모양이 다르다. 그림 3은 각 입력되는 중성에 따라 'ㄱ'의 글꼴 모양이 달라진 모습을 확인할 수 있다. 중성이 세로 중성인지, 가로 중성인지에 각 'ㄱ'의 글꼴 모양이 2가지 형태로 나뉠 수가 있는데, 이에 입력되는 자소형 코드는 하나이지만, 출력되는 폰트 코드는 각각 다르다.[4]



그림 3 중성에 따른 글꼴 모양

이에 자소형 코드는 폰트 코드와의 일대다 관계이다. 그림 4를 보면, 'ㄱ'의 글꼴 모양이 다양하다는 것을 알 수 있다. 이는 모든 한글인 약 399억 음절을 표현하기 위해서는 자소 코드를 이용한 조합 음절을 사용하기 때문이다. 또한, 한 가지 형태만의 글꼴을 만들 경우, 음절 글꼴이 예쁘지 못한 문제점이 있기 때문에 음절 구성의 9가지 형식과 3가지 형에 따라 글꼴 모양을 제작한 것이다. 그러므로 자소형 코드는 하나이지만, 폰트 코드는 다가가 되는 것이다.

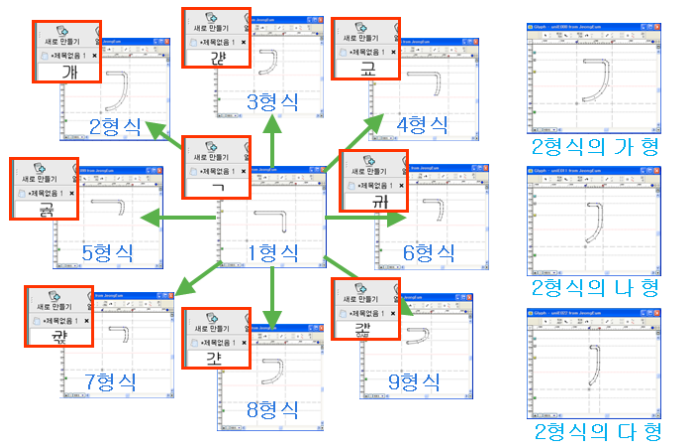


그림 4 음절 구성의 9형식과 3가지 형에 따른 코드

#### 4. 실험 결과

본 연구에서 개발된 옛한글 입력기는 입력 시에는 자소형 코드를 입력받지만, 컴퓨터상에서 표현 시에는 최적 폰트를 적용하기 위해서 음절을 구성할 때마다 폰트 코드로 변환하여 출력되는 입력기이다. 하지만, 입력 시의 자소형 코드와 화면상에 출력되는 폰트 코드가 일치하지 않는 문제점을 해결하기 위해서 본 연구에서는 입력된 자소형 코드를 파일입출력을 통해 파일로 저장하여, 이를 개발된 옛한글 입력기의 오토마타를 통해 최적 폰트 코드로 변환하여 한글 음절을 보이게 한다.

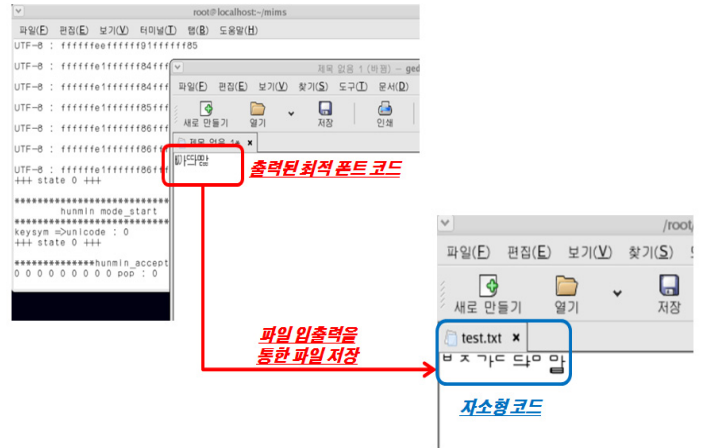


그림 6 입력된 자소형 코드 파일 저장

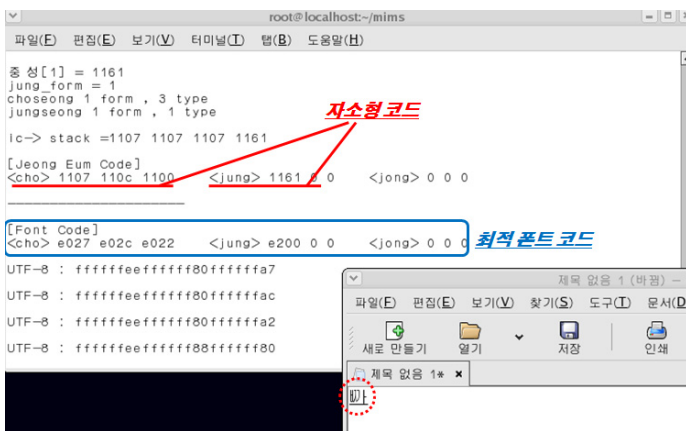


그림 5 자소형 코드와 최적 폰트 코드 실험

그림 5를 보듯이 키보드 입력 시에는 자소형 코드로 입력받으며, 이에 출력되는 코드는 폰트 코드를 통해 글꼴을 표현하고 있다. 이는 각 입력되는 입력 버퍼와 출력되는 출력 버퍼가 별도로 나뉘어져 있다. 즉, 입력 버퍼에 입력된 자소형 코드를 각 음절 구성의 형식과 형에 따라 구분 후 이를 이용하여 폰트 코드로 변환하여 출력하는 방식이다.

실험을 위해서 그림 6과 같이 자소형 코드 입력된 입력 버퍼 값을 파일입출력을 통하여 파일로 저장한다. 그림을 보듯이 출력된 최적 폰트는 모아쓰기 형태로 한글 음절을 구성하고 있지만, 파일로 저장된 자소형 코드는 풀어쓰기 형태로 저장된 것을 확인할 수 있다. 이는 음절 구성인 폰트 코드가 아닌 입력된 낱자소 코드만을 저장하였기 때문이다. 저장된 자소형 코드 파일을 컴퓨터상에서 모아진 음절로 표현하기 위해서는 본 연구의 옛한글 입력기 오토마타(그림 7참고)를 통해 폰트 코드로 변환하여 출력하여야 한다. 입력기 오토마타는 초성, 중성, 종성 상태뿐만이 아닌 조합음절의 음절 구성을 위한 9가지 형식과 3가지 형의 정보도 같이 배정되어 있다.

○ 초성1 'ㄴ' + 초성2 'ㅈ' + 초성3 'ㄱ' + 가로중성1 'ㅏ'

- 입력 버퍼 :

- 초성1 0x1107, 초성2 0x110C, 초성3 0x1100
- 중성1 0x1161, 중성2 0, 중성3 0
- 종성1 0, 종성2 0, 종성3 0

- 출력 버퍼 :

- 초성1 0xE027, 초성2 0xE02C, 초성3 0xE022
- 중성1 0xE200, 중성2 0, 중성3 0
- 종성1 0, 종성2 0, 종성3 0

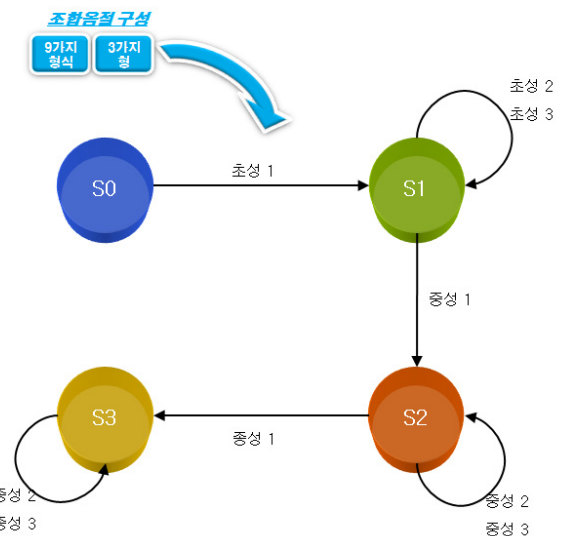


그림 7 자소형 코드의 음절 구성 오토마타

그림 8은 자소형 코드로 저장된 파일을 폰트 코드로 표현하기 위해서 파일입출력을 통해 파일을 읽어서 이를 버퍼에 저장하여, 입력기 오토마타를 통해 각 자소형 코드를 접수기에서 초, 중, 종성을 구분하고, 이를 오토마타를 통해 상태, 형식, 형 등을 배정하여 이를 가지고, 폰트 코드 변환기를 통해 폰트 코드로 변환하여 출력한다.

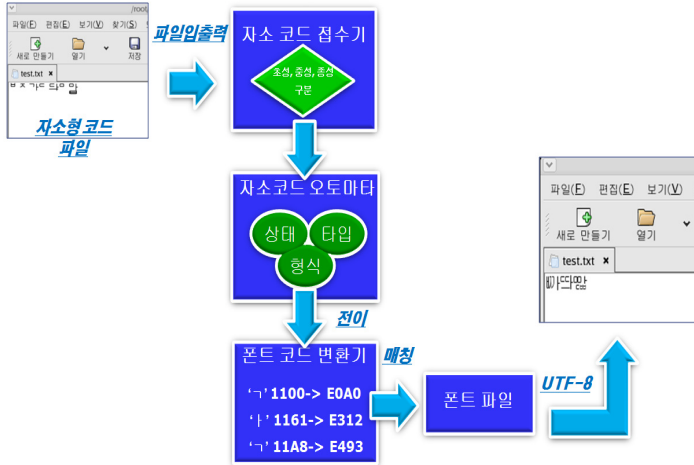


그림 8 자소형 코드를 폰트 코드로 출력

파일입출력을 통한 각 자소형, 폰트 코드는 UTF-8을 이용하여 인코딩되어 출력된다. 그리고 변환된 폰트 코드는 별도의 자소 코드 변환테이블을 통해 자소형 코드로 변환될 수 있다.[3]

## 5. 결론

본 연구에서는 이전 연구의 훈민정음 창제당시의 모든 한글의 약 399억 음절을 표현하는 데에만 치중하는 문제점이 있었다. 이는 입력되는 자소형 코드와 컴퓨터상에 출력되는 폰트 코드의 일치하지 않는 문제점을 해결하는 데에 중점을 두었다. 이는 자소형 코드와 폰트 코드와는 일대다 관계를 가지기 때문에 이를 하나로 통일하여야 한다.

그러기 위해서 실험을 통하여 입력에 대한 코드를 자소형 코드로 파일입출력을 통해 파일로 저장하여 이를 본 연구의 옛한글 입력기의 풀어쓰기 형태로 된 자소형 코드를 모아쓰기 형태인 음절로 표현하기 위해서 폰트 코드 변환 모듈을 사용하여 폰트 코드로 변환하여 음절을 구성하여 보여주고 있다.

그러므로, 훈민정음 원리에 맞는 자소형 코드와 화면에

출력되는 최적 폰트 코드의 일치할 수 있도록 있었다.

추후 연구로는 자소형 코드를 이용하여 최적 폰트에 대한 글꼴 모양을 제작하는 데에 있어서 좀 더 줄일 수 있는 방안을 연구하고, 훈민정음 원리에 기반한 웹 버전의 옛한글 입력기를 개발하여 모든 한글을 표현하고자 한다.

## 참고문헌

- [1] Byun JY, "A Representation of Hangeul Syllables defined in Hunminjeongeum", MMM2007 LNCS4351, SpringerVerlag, 2007
- [2] 변정용, 김경옥 "훈민정음 원리에 기반한 자소형 최적 폰트 개발", 한국정보처리학회 2007년 춘계학술대회, VOL. 14 NO. 01 pp. 0690 ~ 0693, 2007
- [3] Kim KW, Byun JY "An Implementation of Optimized Hangeul Font Set for Multilingual IM Engine", ALPIT2007, VOL. 6 NO. pp. 200~205, 2007
- [4] 김경옥, 변정용 "훈민정음 표현을 위한 최적 폰트 설계", 제20회 한글 및 한국어 정보처리 학술대회, 제20권 제1호, pp.53~57, 2008
- [5] The Unicode Consortium, "The Unicode Standard 4.0", Addison-Wesley, 2003
- [6] 변정용, "국제 문자 부호계에 포함된 한글 부호계의 개선 방안", 98 한국정보과학회 가을 학술발표 논문집 Vol. 25 No.2
- [7] 변정용, "훈민정음 원리의 공학화에 기반한 한글 부호계의 발전방향", 1994 정보과학회지 제12권 8호
- [8] 변정용, "한글 부호계 제정에서 반복되는 시행착오의 고리를 끊으려면", 98 정보처리 제5권 5호
- [9] 변정용, "훈민정음 원리에 따르는 우리글 코드 제정 방안", 96 코리언 컴퓨터처리 학술대회 논문집
- [10] 안대혁, 박영배, "유니코드 환경에서의 올바른 한글 정규화를 위한 수정 방안", 정보과학회지: 소프트웨어 및 응용 제34권 제2호(2007.2.), p169-177