

# A Phonological Process Code in Unicode for the Standardization of Hangeul(Korean Alphabet) Pronunciation

Cheong-Sick Min\*

## Abstract

In this paper, a standardized phonological rule should be embodied for all people who want to use Hangeul beyond nationality. This researcher suggested a phonological process protocol in unicode for the purpose of developing a computer system-based program as one of methods to pronounce Hangeul correctly.

First of all, to solve a problem of standardizing Hangeul pronunciation, it is necessary to develop a phonological process system. In other words, when a mobile phone or a keyboard is used, inputted strings should be changed exactly by an automatic device made according to the phonological rule. For this, a phonological process table is the most important and unicode which digitalizes Hangeul is the most effective. Therefore, the purpose of this study is to complete a phonological process table, suggest an algorithm by the table, and develop a computer program pursuing correct Hangeul pronunciation.

For the methodology of this study, the features of unicode was used to solve a problem of coding each syllable by phonemes after strings were inputted from the phonological process program. This study suggested a protocol that sounds of inputted strings were automatically changed via recombination using a phonological process table. The ultimate purpose of the study was to help pronounce Hangeul correctly according to the pronunciation rule.

▶ Keyword : Phonological processe, Unicode, Standardization of Hangeul pronunciation.

## 1. Introduction

최근 계속되고 있는 한류열풍의 기세와 함께 한국어를 배우고자 하는 외국인들이 늘어나고 한국어 교육기관을 비롯하여 한국어교사 양성기관이 활발히 설립되고 있다. 더구나 대학 내에서도 한국어 교원을 양성하고, 한국어 교원자격증까지 생성되기에 이르러 한국어에 대한 관심은 날로 상승세가 지속되고 있다. 그러나 외국인을 대상으로 ‘한국어에 대한 인식’을 묻는 설문조사 실시한 결과, 응답자의 대부분이 한국어를 공부할 때 ‘발음’이 가장 어렵다고 답했다[1]. 특히 소리가 나는 발음과 표기가 일치하지 않은 관계로 한국어 교육에 어려움을 토로하고 있으며 컴퓨터상에서 한글을 입력하는 데에도 상당한 어려움을 호소하고 있다. 이와 같은 문제점을 해결하기 위해서 ‘한

국인과의 대화 연습’이나 ‘방송매체를 통한 듣기 연습’을 유도하고 음운변동 규칙이 연구되고 있지만 근본적인 해결책이 되고 있지는 않다. 이에 본 연구자는 한국어 발음의 표준화와 함께 표준화된 한글발음규칙 지정과 그 규칙에 의하여 발음할 수 있도록 도와주는 컴퓨터 입출력 장치의 필요성을 갖는다.

한글은 불과 24개의 자모로 구성되어 있으며, 지구상의 언어 중 가장 다양한 소리를 표현 할 수 있을 뿐 아니라 서양 언어와는 달리 한국의 언어는 소리 나는 모든 것을 글자로 바로 표현하는 것이 가능하다. 그러나 받침이 많은 관계로 여러 가지 소리를 낼 수는 있지만 받침이 있는 첫 글자와 한 음절이 만나면 원 글자와는 다르게 발음을 할 수 밖에 없는 현상이 종종 발생한다. 이 부분에서 외국인이 한국어를 할 때 의사 표현에 오류를 발생할 가능성이 많고 외국인들이 한국어를 올바르게 사용하는데 제약 조건이 되고 있다.

• First Author: Cheong-Sick Min, Corresponding Author: Cheong-Sick Min  
\*Cheong-Sick Min (chskmin@hanmail.net), Dept of Korean Language, Dongshin University  
• Received: 2015. 11. 03, Revised: 2015. 11. 12, Accepted: 2015. 11. 19.

따라서 본 연구자는 이 문제를 해결하는 데는 기존의 음운변동 규칙을 토대로 앞글자의 받침과 뒷글자의 초성과의 관계를 모두 코드화 하여 연결된 한글 두 글자에 대하여 음운이 변동하는 결과를 나타내는 컴퓨터 시스템 개발과 함께 입력된 문자열을 음운규칙대로 변동하는 알고리즘에 의한 프로그램이 필요하다고 보고, 입력된 문자열을 자동으로 발음규칙에 의해 출력되는 알고리즘을 제시하고자 한다.

본 연구는 실증적 연구로서 입력된 한글 문자열을 유니코드로 변환하고, 그 코드로부터 초성, 중성 그리고 종성을 분할하여 코드표에 의하여 종성과 초성 등을 변환한 다음, 다시 재조합하여 문자열을 소리 나는 대로 발음하는 문자열로 바꾸는 일련의 과정을 제작하는 기본을 구축하기 위하여 유니코드 상에서의 음운변동표를 음운규칙에 따라 코드화하였고 유니코드를 기반으로 하는 음운변동 프로토콜을 제시하였다. 문자열마다 결과가 단 하나로 결정되어야 하는 관계에 의하여 기계적으로 출력이 가능해야 하기 때문에 본 프로그램은 수학적 함수관계를 적용하였다. 즉, 입력된 한글을 규칙대로 변동하기 위하여 선행 연구들의 결과들을 기준으로 유니코드를 이용하여 모든 경우의 음운변동표를 구축하였으며, 그 표에 의하여 음운이 자동으로 변동하여 출력되는 프로토콜을 제시 하였다.

본 연구자는 앞 글자의 종성과 뒷글자의 초성과의 관계를 규칙에 의하여 코드화 한 다음 자동으로 변동된 문자열을 출력하고자 본문에서 제시한 음운변동표를 이용한 음운변동 구현에 성공하였으며, 이를 토대로 모바일 어플[2] 등의 보급으로 이어져 한국어를 배우려는 모든 이들에게 올바른 한글 발음을 할 수 있도록 도움을 주고자 표준화 하는데 목적을 두고 있다.

이는 한국어를 표준화된 음운규칙에[3] 의하여 발음을 하도록 도움을 주는 입출력 오토마타에 의하여, 한글의 표준화와 국제화를 기대할 수 있을 것이다.

본 고를 통해 입증된 규칙을 기초로 하여 한글 표준화프로그램을 보완 발전시킨다면 이를 수요로 하는 한국어학당을 비롯하여 외국의 한글교육기관의 수요를 충족시킴으로써 한글교육의 저변확대에도 기여할 수 있으며, 한글 관련 프로그램 산업의 발달에 기여함으로써 고도의 가치를 창출할 수 있는 분야가 될 것으로 기대한다. 이와 관련하여 한글프로그램의 생성 후 마케팅전략 등에 대한 논의는 추후 발표하기로 하고, 본고에서는 먼저 한글 음운변동표의 생성에 관한 이론을 중심으로 논의를 전개하고자 한다.

## II. Standardization of Hangu(Korean Alphabet) Pronunciation

한글은 오늘날에는 24자만 쓰인다. 간단한 조합만으로도 소리 나는 모든 말들을 표현할 수 있는 한글이야말로 지극히 과학적인 문자임에 틀림없으며 컴퓨터 등 IT와 접목되면서 편

의성과 우수성 측면에서 새롭게 조명 받고 있다.

따라서 한글코드의 표준화는 항상 필요한 문제였다. 표준이 없는 상태에서 다양한 프로그램이 임의의 코드로 한글을 처리하면서 여러 부작용이 나타났기 때문이다.

### 1. The Unification of Hangu Codes

한글은 20여 년 간 조합형, 완성형, 확장완성형, 유니코드를 비롯해 다양한 처리방식으로 경쟁해왔다. 조합형의 장점은 한글로 표현되는 모든 문자조합을 모두 다 표현할 수 있다는 점이다. 반면에 완성형은 한글 한글자의 음소의 조합으로 보지 않고 상형문자와 같이 하나의 글자를 그림처럼 인식한다. 초기의 완성형은 2350자만을 표현할 수 있어 문 제가 됐다. 이 같은 문제가 발생하자 MS는 1995년 확장 완성형 코드를 선보였다. 기존 완성형 코드를 그대로 사용하고 기존 완성형 코드에서 표현할 수 없었던 한글 8822자를 새로운 영역에 추가시킨 것이다. 이제 확장 완성형이나 조합형 대신 유니코드가 대세로 자리 잡았다. 유니코드는 전 세계 문자코드의 표준화를 위해 업계가 함께 만든 것이다. 한글뿐 아니라, 전 세계의 모든 글자들을 한꺼번에 표현하는 코드방식이다. 나라마다 글자(영어, 한글, 히라가나 따위)를 표현하려면 할당되는 컴퓨터 메모리 공간의 크기가 다른데 이는 국가 간 언어 교환 시에 문제점으로 부각되었으며, 이를 원활하게 하기 위해 문자 1개에 부여되는 값을 16비트로 통일화한 특징이 있기도 하다. 유니코드는 완성형 방식을 따르면서도 조합형의 장점을 수용했다는 점에서 높은 점수를 얻고 있다. 현재 유니코드는 운영체제 Windows XP, 워드 프로세서 한글, MS Word의 기본 문자 코드이자, 인터넷에서 가장 널리 사용 되는 문자 코드이다. 전 세계 문자 체계를 통합하기 위하여 전 세계 대부분의 문자 코드를 수용하였고, 유니코드를 사용하는 컴퓨터 시스템에서 세계 각국의 언어를 사용하고 교환할 수 있도록 만들었다.

모든 정수는 2진법, 10진법 그리고 16진법 등으로 표현 할 수 있는데 각각의 표현을  $(a)_2$ ,  $(a)_{10}$ ,  $(a)_{16}$  등으로 표현한다. 다만 특별한 표현이 없는 숫자는 본 논문에서는 10진법으로 통용하기로 한다. 유니코드의 코드값은 다음과 같다.

$$\text{코드값} = 44032 + (\text{초성값} * 21 * 28) + (\text{중성값} * 28) + \text{종성값}$$

한글의 가장 첫자인 ‘가’를 나타내는  $(AC00)_{16} = 44032$  은 10진법으로 44032이고 마지막자인 ‘힉’을 나타내는 55203 이다. 따라서 유니코드의 한글은 모두 11172개가 등록되어 있다. ISO 표준코드 65536 개에서 11172개를 단지 한글이 점유하고 있는 것이다.

처음 완성형은 규칙성이 없이 만들어 졌지만 유니코드는 완성형 코드에 규칙성을 추가 하였으므로 시작번호가 44032부터 시작 한 것을 제외 하고는 유니코드도 결국 조합형 한글코드의 응용이라 할 수 있다. 전 세계 글자를 모두 통합한 유니코드의 65536 자리 중에서 일본의 강력한 반대에도 불구하고 11172

공간을 한글만으로 확보하였다는 사실은 상당히 중요한 의미를 갖고 있으며 모든 한글 글자가 등록되어 있으므로 전 세계 어디서나 유니코드를 이용한 음운변동 오토마타를 실행 할 수 있는 것이다.

## 2. Constituents Code

$H$ 를 한글 전체 집합이라 하고  $A_0, A_1, A_2$ 를 각각 초성, 중성, 종성집합이라고 하자[4]. 집합  $A_0$ 의 개수를  $|A_0|$ 라고 하면  $|A_0|=19, |A_1|=21, |A_2|=28$ 이다[1-6]. 따라서 한글 전체집합  $H$ 의 개수는  $|H|= 19*21* 28=11172$ 가 된다. 초성집합  $A_0$ 의 구성 원소와 번호는 표 1과 같다.

Table 1. Constituents and numbers of initial consonant set

번호	0	1	2	3	4	5	6	7	8	9
구성원소	ㄱ	ㅋ	ㄴ	ㄷ	ㄸ	ㄹ	ㄴ	ㅁ	ㅂ	ㅃ
번호	10	11	12	13	14	15	16	17	18	
구성원소	ㅅ	ㅇ	ㅈ	ㅊ	ㅌ	ㅋ	ㅍ	ㅑ	ㅓ	

그리고 중성집합  $A_1$ 의 구성원소와 번호는 표 2와 같다.

Table 2. Constituents and numbers of vowel set

번호	0	1	2	3	4	5	6	7	8	9	10
구성원소	ㅏ	ㅑ	ㅓ	ㅕ	ㅗ	ㅛ	ㅜ	ㅠ	ㅡ	ㅚ	ㅜ
번호	11	12	13	14	15	16	17	18	19	20	
구성원소	ㅝ	ㅟ	ㅛ	ㅜ	ㅠ	ㅡ	ㅑ	ㅓ	ㅕ	ㅗ	

마지막으로 종성집합  $A_2$ 의 구성원소와 번호는 표 3과 같다.

Table 3. Constituents and numbers of final consonant set.

번호	0	1	2	3	4	5	6	7	8	9
구성원소		ㄱ	ㅋ	ㄴ	ㄷ	ㄸ	ㄹ	ㄴ	ㅁ	ㅂ
번호	10	11	12	13	14	15	16	17	18	19
구성원소	ㅅ	ㅇ	ㅈ	ㅊ	ㅌ	ㅋ	ㅍ	ㅑ	ㅓ	ㅕ
번호	20	21	22	23	24	25	26	27		
구성원소	ㅝ	ㅟ	ㅛ	ㅜ	ㅠ	ㅡ	ㅑ	ㅓ		

중성집합 에서 첫 번째 칸이 공란인 것은 한글은 받침이 없을 수도 있기 때문이다.

유니코드를 이용하여 한글을 초성, 중성, 종성으로 각각 구별할 수 있다. 유니코드의 한글부분 첫 글자는 ‘가’이고 코드번호는 44032이다. 다음 글자는 ‘각’이며 44033, 마지막 글자인 ‘향’은 55203이다. 55203- 44032=11171이므로 유니코드의 한글은 모두 11172개가 등록되어 있다.

유니코드값은 안드로이드 시스템을 포함한 모바일에서도 제

공되고 있다. 이 점은 한글 발음 자동화를 구축하여 어플을 이용하여 한글을 체계화 하고 한글 발음을 표준화 할 수 있는 상당히 유리한 요인을 갖고 있으며 전 세계 어느 언어도 이런 편리한 구조를 가질 수 없다.

각각의 초성에 대하여 21개의 중성이 있고 각각의 중성 값에는 28개의 중성이 있으므로 초성 하나에 해당되는 글자는 21\*28개이다. 또 유니코드는 44032 부터 한글이 시작 되므로 한글 h가 주어지면 h의 유니코드값을 H 라고 할 때 H에서 44032를 빼고 그 값을 21\*28로 나눈 값의 몫이 초성 순번이다. 더욱이 그 나머지를 중성집합 개수인 28로 나눈 몫이 중성 순번이고 나머지는 중성 순번이다.

예) 한글 h를 ‘뱀’이라고 하면 h의 유니코드값은 48277이고 48277 - 44032 = 4245이므로 ‘뱀’은 유니코드 상에서 한글의 4246번째 글자이다. 4245를 21\* 28로 나눈 몫은 7이고 나머지는 129이다. 따라서 h의 초성은 7번째 음소인 /ㅂ/이다. 더욱이 129를 28로 나눈 몫은 4이고 나머지는 17이므로 h의 중성은 4번째 중성인 /ㅓ/이고 h의 중성은 17번째 음소인 /ㅕ/이다. 코드값과 초성, 중성, 종성과의 관계를 종합하면 다음과 같다.

$$\text{유니코드코드 값} = 44032 + (\text{초성값} * 21 * 28) + (\text{중성값} * 28) + \text{종성값}$$

각각의 문자에 대하여 유니코드값을 알 수 있고 초성, 중성, 종성 값들을 알 수 있다.  $a \in H$  를 한글이라 하면  $p_0(a)$  를 a의 초성,  $p_1(a)$  를 a의 중성,  $p_2(a)$  를 a의 종성 이라고 표기하고 그 계산은 다음과 같다.

$$p_0(a) = ((\text{유니코드값} - 44032) / 28 * 21) \text{의 몫};$$

$$p_1(a) = ((\text{유니코드값} - 44032) / 28) \text{의 몫};$$

$$p_2(a) = ((\text{유니코드값} - 44032) / 28) \text{의 나머지}.$$

유니코드값과 음소들과의 관계는 표 4와 같다.

Table 4. Unicode values and Constituent values

한글	유니코드값	초성	중성	종성
가	44032	0	0	0
각	44033	0	0	1
감	44034	0	0	2
⋮	⋮	⋮	⋮	⋮
뱀	48277	7	4	17
⋮	⋮	⋮	⋮	⋮
를	47456	5	17	8

위의 표4는 한글과 그에 해당되는 유니코드값 그리고 각 음소들의 코드값들이다.

한글이 입력되면 위의 표에서 그 한글을 1열에서 검색하여 바로 옆 숫자인 유니코드값을 알 수 있다[4-8]. 그리고 그 유

니코드값을 이용하여 초성, 중성, 종성 등 각 음소로 분해 할 수 있다.

아래는 한글 각각의 유니코드값에 대하여 초성, 중성, 종성 등을 구하는 컴퓨터사용 가능 수식이다[5].

- 1) 초성값을 구하는 수식:  
QUOTIRNT(유니코드값-44032, 28\*21)
- 2) 중성값을 구하는 수식:  
MOD(QUOTIRNT(유니코드값-44032, 28\*21), 21)
- 3) 종성값을 구하는 수식:  
MOD(유니코드값-44032, 28)

$H$ 를 한글 전체의 집합이라고 하면  $H \times H$ 를 한글집합의 외적[6]이라고 한다. 이것은 한글 두 글자 전체집합이라고 할 수 있다.

이제 함수[1-2],[7]  $T : H \times H \rightarrow H \times H$ 를 음운변동규칙에 의한 함수라 하자.

예)  $T(\text{범}, \text{률}) = (\text{범}, \text{률})$

한글은 과학적으로 이루어져 있기 때문에[6] 연결된 한글 두 글자 중 앞 글자의 중성과 뒷글자의 초성과의 관계만을 작은 함수로서 만든다면 함수의 규모를 대폭 줄일 수 있을 것이다. 즉 음운의 변동은 궁극적으로 중성과의 관계가 드물고 28개 중성과 19개의 초성과의 관계변화만 적용한다면 표기가 가능하므로,  $28 \times 19 = 532$  개의 함수 값만 지정해 주면 모든 음운변동의 출력이 가능하게 될 것이다.

### 3. Phonological process Code

이 절에서는 함수값을 구하기 위하여 음운변동 표를 구성한다.

28개 중성과 19개의 초성과의 관계변화인 함수  $t : A_0 \times A_2 \rightarrow A_0 \times A_2$ 를 표현하기 위하여 다음과 같이 중성과 초성과의 관계를 나타내는 표의 일부를 표 5와 같이 예를 들어 구성하였다.

Table 5. Relation of final consonant and initial consonant

	초성	ㄱ	ㄲ	ㄴ	ㄷ	ㄹ
중성						
받침없음	받침없음	받침없음	받침없음	받침없음	받침없음	받침없음
ㄱ	ㄱ,ㄱ	ㄱ,ㄲ	ㅇ,ㄴ	ㄱ,ㄷ	ㅇ,ㄹ	
ㄲ	ㄱ,ㄱ	ㄱ,ㄲ	ㅇ,ㄴ	ㄱ,ㄷ	ㅇ,ㄹ	
ㄴ	ㄴ,ㄱ	ㄴ,ㄲ	ㄴ,ㄴ	ㄴ,ㄷ	ㄹ,ㄹ	
ㅂ	ㅂ,ㄱ	ㅂ,ㄲ	ㄱ,ㄴ	ㅂ,ㄷ	ㄱ,ㄴ	

예) 표 5에 의하면 (ㅂ, ㄹ)의 좌표값은 (ㄱ, ㄴ)이므로, ‘범률’의 초성과 중성을 좌표값으로 대치하면 ‘범률’로 자동으로 표기되는 것이 가능하게 된다.

표 5를 이용하여 자동으로 음운변동을 실행하기 위하여 코

드화된 표로 바꾸어줄 필요가 있으므로 다음과 같이 코드화된 표 6으로 재구성한다. 1열의 숫자들은 앞글자의 중성의 순서이고 1행의 숫자들은 뒷글자의 초성의 순서이다.

Table 6. Encoding of table 5

	초성	0	1	2	3	5
중성						
0	(0, 0)	(0, 1)	(0, 2)	(0, 3)	(0, 5)	
1	(1, 1)	(1, 1)	(21, 2)	(1, 4)	(21, 2)	
2	(1, 1)	(1, 1)	(21, 2)	(1, 4)	(21, 5)	
4	(4, 0)	(4, 1)	(4, 2)	(4, 3)	(8, 5)	
17	(17, 1)	(17, 1)	(16, 2)	(17, 4)	(16, 2)	

표 6을 이용하여 음운변동을 실행하려면 좌표로의 표기는 실행에 지장이 있으므로 4자리의 단일 정수로 표기하는 것이 컴퓨터 구현에 유리하다. 예를 들면 (16, 2)는 1602로 단일 정수로 표기하면 운용이 간편하고 이 숫자를 각 음소로 바꾸어 줄 수 있다. 앞 두 숫자를 구하려면 4자리 정수를 100으로 나눈 몫으로 계산하면 되고 그 나머지를 뒷글자의 초성으로 구할 수 있다.

예)  $1602 = 16 \times 100 + 2$ 이므로 16은 앞글자의 중성의 순번이고 2는 뒷글자의 초성의 순번이다.

즉 1602는 (ㄱ, ㄴ)을 의미하게 된다.

표 6의 값을 단일 정수로 표를 구성하면 표 7과 같다.

Table 7. Integer value of table 6

	초성	0	1	2	3	5
중성						
0	0	1	2	3	5	
1	101	101	2102	104	2102	
2	101	101	2102	104	2105	
4	400	401	402	403	805	
17	1701	1701	1602	1704	1602	

### 4. The Phonological Process Protocol

한글 두 글자를 입력 받고 그 두 글자 각각에 대하여 초성, 중성, 종성 등 각 음소로 구분하고 앞글자의 중성과 뒷글자의 초성을 위의 음운변동표에 해당하는 각 음소로 바꾸어 한글로 다시 결합하는 과정을 거친다.

아래는 유니코드 상에서의 음운변동 프로토콜을 제시하고 있다.

- 1) 한글 두 글자 ( $a, b$ )를 입력한다.
- 2) 한글들을 유니코드값으로 바꾼다.
- 3) 두 숫자를 초성, 중성, 종성 등으로 각 음소로 분해한다.  
 $((p_0(a), p_1(a), p_2(a)), (p_0(b), p_1(b), p_2(b)))$
- 4) 앞 글자의 중성과 뒷글자의 초성을 음운변동 표에 의한

작은 함수  $t$ 에 의하여 구한다.

$$t(p_2(a), p_0(b)) = (t_1(p_2(a), p_0(b)), t_2(p_2(a), p_0(b)))$$

5)  $(p_0(a), p_1(a), t_1(p_2(a), p_0(b)))$ 와  $(t_2(p_2(a), p_0(b), p_1(b), p_2(b)))$ 를 유니코드로 결합한다.

6) 결합된 숫자를 유니코드값으로 하는 한글을 찾는다.

예)

1) 한글 두 글자 (법, 룰)을 입력한다.

2) ‘법’의 유니코드값은 48277이고 ‘룰’의 유니코드값은 47456이다.

3) 두 글자를 각 음소로 분해하면 ‘법’은 (7, 4, 17)이고 ‘룰’은 (5, 17, 8)이다.

4) 위의 음운변동표에서 17과 5가 만나는 좌표는 (16, 2)이므로,  $t(17, 5) = (16, 2)$ 가 되고, 음운변동 후 앞글자의 종성은 16이고 뒷글자의 초성은 2가 된다.

5) 음운변동 전의 앞글자의 좌표인 (7, 4, 17)의 종성인 17대신 16을, 뒷글자인 (5, 17, 8)의 초성인 5 대신 2를 대체한 (7, 4, 16)과 (2, 17, 8)을 결합하여 유니코드값을 구하면 48276과 45692가 된다. 아래는 음소들을 결합하는 계산방법이다.

$$48276 = 44032 + 7*21*28 + 4 *28 + 16$$

$$45692 = 44032 + 2*21*28 + 17*28 + 8$$

6) 이 코드에 대한 한글은 ‘법’과 ‘룰’이 된다는 것을 알 수 있다.

즉  $T(\text{법}, \text{룰}) = (\text{법}, \text{룰})$ 이 됨을 확인 할 수 있다.

함수  $t$ 의 정의구역 및 공변역의 개수 532개는 사실상 웹상에서 무시할 만한 적은 용량이므로 한글을 제대로 읽는 표준화를 구축하는데 전혀 무리가 없을 것이다.

### III. The Table of Phonological Process Code

#### 1. Preceding Research

본 연구는 한국어 발음교육의 현장실태의 문제점과 학습자 요구분석을 통해 음운변동을 활용한 한국어 발음교육에 대한 실제적 교육 방안에 관한 연구 결과들을 바탕으로 [2-4],[8] 음운변동규칙의 표준화를 위하여 자바와 같은 프로그램 언어에 적용할 수 있는 음운변동 프로토콜을 제시함을 목적으로 한다. 본 연구의 가시적인 결과는 엑셀 프로그램으로 작성하여 구현하는데 성공하였다.

통신규칙의 코드를 작성하기 위하여 현재까지의 선행연구들을 살펴본다. 한국어 발음교육의 연구는 현재까지의 선행연구를 살펴보면 에서 시작되어 1990년대 점차 소폭으로 증가하다가 2000년대에 들어서부터 폭발적인 양적증가를 이루고 있다.

가장 먼저 학계에서는 음운변동 규칙의 교육항목에 대한 기준으로 표기와 발음이 일치하지 않은 것에 대한 음운변동 현상

만을 교수학습내용으로 선정하였는데 그 순서는 다음과 같다.

일곱 끝소리 되기→ 겹자음 줄이기→ 소리 이음→ㅎ 소리 주기→ 된소리 되기→ 콧소리 되기→ 거센소리 되기→ 입천장소리 되기→흐름소리 되기→ 사잇소리 현상.

음절, 말, 자음과 관련된 음운변동 규칙의 교육 항목에 대한 기준으로 표기형과 음성형이 일치하지 않는 경우만 발음교육의 대상으로 하였다. 교육항목은 다음과 같다.

연음 규칙, 겹받침 탈락 규칙, 7종성 규칙, /ㅎ/ 탈락 규칙, 구개음화, 유기음화, 유음화, 비음화, 자음위치 동화(역행동화, 순행동화) 경음화(규칙적, 불규칙적),마찰음화, /ㄹ/의 /ㄴ/되기, 첨가(ㄴ첨가, ㄹ첨가, ㄹ첨가, ㄹ첨가)화. /ㄹ/역행동화 /ㄴ/의 음가 등이 있다.

본 통신규칙에서는 한글 음소들이 결합되는 11172개의 음절들이 모두 유니코드에 등록되어 있으므로 표기형과 음성형이 일치하든 안 하든 모든 경우를 포함하였다. 선행연구는 음운변동 학습에 연구를 시행한 사례가 많았는데 [2-4],[6] 본 규칙에서는 그 사례의 결과들에 의하여 음운변동의 결과를 성립하기 위해 모든 음절들의 변동전체를 코드화하였다. 또 음운변동규칙의 명칭이 제각각 다르게 사용되기도 하지만 통신규칙에서는 코드화가 기본적인 문제이므로 명칭의 통일이 문제가 되지는 않았다. 즉 하나의 코드에 여러 가지 명칭이 있어도 통신규칙의 구축에는 무리가 없었다.

#### 2. Construction of The Phonological Process Table

이장에서는 앞 글자의 종성의 개수 28개와 뒷글자의 초성 19개의 모든 경우 28\*19=532 개의 코드를 모두 작성하기 위하여 선행 학습들의 규칙에 따라 함수 관계를 정립하여 음운변동표를 구성하였다.

1) 앞 글자의 종성이 없는 경우 : 이 경우는 뒷글자의 초성은 표기형과 음성형이 같다. 이것을 코드화 하면 다음과 같다. 표 8의 상단은 뒷글자의 초성이고 하단은 각각에 해당하는 코드이다. 앞글자의 종성이 없는 관계로 음운변동 결과도 앞글자의 종성도 받침이 없고 뒷글자의 초성은 변동전의 초성과 같다.

Table 8. The case of syllables without final consonant

	ㄱ	ㄲ	ㄴ	ㄷ	ㄸ	ㄹ	ㅁ	ㅂ	ㅃ
	0	1	2	3	4	5	6	7	8
ㅅ	ㅆ	ㅇ	ㅈ	ㅊ	ㅋ	ㆁ	ㄷ	ㄹ	ㅎ
9	10	11	12	13	14	15	16	17	18

(예) ‘코드’의 음운변동을 살펴보면 앞글자 ‘코’의 종성은 없으므로 이 경우에 해당되고 뒷글자의 초성은 /ㄷ/이므로 코드번호는 3이다. 4자리 숫자중 앞의 두 수는 0이고

앞글자의 초성은 없고 뒤의 두 수는 3이므로 뒷글자의 종성은 /ㄷ/이므로 ‘코드’의 음운변동은 변화 없이 ‘코드’가 된다.

Table 9. The case of the liaison

중성	ㄱ	ㄲ	ㄴ	ㄷ	ㄸ	ㄹ	ㄺ	ㄻ	ㄼ
○	0	1	109	2	412	2	12	5	800
	ㄹ	ㄹ	ㄹ	ㄹ	ㄹ	ㄹ	ㄹ	ㄹ	ㄹ
○	806	807	809	816	817	5	6	7	1709
	ㅅ	ㅆ	ㅇ	ㅈ	ㅊ	ㅋ	ㅌ	ㅍ	ㅎ
○	9	10	2111	12	14	15	16	17	11

2) 소리 이음: 이 경우는 뒷글자의 초성이 /ㅇ/ 인 경우 앞 글자의 종성을 이어 받는다는 규칙이다. 다만 앞 글자의 종성이 /ㄷ/ 등과 같이 무엇인가로 존재 하느냐 혹은 받침 없이 발음하느냐는 문제는 규칙의 통일문제이며 이 후 권위있는 기관의 결정에 따라 그에 따른 코드 번호만 바꾸어준다. 예를 들면 ‘했음’을 출력할 때 현재는 ‘해썹’과 ‘헨썹’을 혼용하고 있는데, 음운변동이 표준화 되려면 한가지로 통일하여야 하므로 권위있는 기관 혹은 사회의 통념상 한가지로 결정되어야 한다고 사료된다. 본 규칙에서는 앞글자의 종성이 없어지고 뒷글자의 초성을 /ㅇ/ 대신 대체하는 형태로 정리해 보면 표 9와 같다. 예외로 앞 글자의 종성이 /ㅎ/ 일 때는 /ㅎ/ 탈락 규칙이 우선하고 /ㅇ/ 일 때는 종성의 /ㅇ/을 없애지 않는다. 아래의 표는 앞글자의 종성이 다음과 같이 28가지로 변하고 뒷글자의 초성이 /ㅇ/으로 고정되어 있을 때의 코드이다.

예)

(1) ‘앞에’, ‘옆에’는 ‘아페’, ‘여페’ 등으로 코드값은 17이고 앞글자의 종성은 0으로 없으며 뒷글자의 초성은 17번째 초성 /ㅍ/으로 소리이음현상이다. 여기에서 ‘아페’를 ‘압페’로 변동하기로 규칙이 통일 된다면 코드번호를 17에서 1717로 바꾸어 주면 가능한 문제가 될 것이다.

(2) ‘없음’, ‘값은’ 등은 ‘업슴’, ‘갑슴’ 등으로 변동하며 코드값은 1709이고 앞글자의 종성은 17번째 음소인 /ㅂ/이고 뒷글자의 초성은 9번째 음소인 /ㅅ/으로 소리이음현상이다. 복자음은 앞 음소는 존치하고 뒤 음소는 소리이음현상이다. 여기서도 마찬가지로 ‘업슴’ 등을 ‘업썹’ 등으로 변동하기로 통일 한다면 1709를 1710으로만 음운변동표를 변화하면 된다.

(3) ‘좋아’는 ‘조아’로 코드값은 11이므로 /ㅎ/ 탈락현상이다.

일반적으로 음운변동표의 좌표값 하나에 대하여 그에 해당되는 두 음절의 쌍의 개수는  $19 \times 21 \times 21 \times 28 = 234,612$  개이다. 즉 예를 들면, (ㅂ,ㅇ)의 좌표값 1709에 대하여 앞글자 종성 /ㅂ/과 뒷글자 초성 /ㅇ/으로 이루어진 음절의 쌍은 {값아, 값아, 값아, ...값은, ...없음, ..., 값잉} 등으로 234,612개이다. 작은 함수  $t$ 를 사용하는 이유가 여기에 있으며 메모리와 시간의 낭비를 대폭 감축할 수 있다.

모두 1억 개가 넘는 두 음절의 쌍을 모두 작성하기도 힘들지만 작성 하였다고 하더라도 메모리를 너무 많이 차지하기 때문에 메모리 허용량이 적은 모바일의 어플에서 음운변동표를 이용하는 통신규약 없이 음운변동을 실행하는 것은 사실상 불가

능하다. 음운변동을 코드화 하는 이유라고 할 수 있는 이 프로토콜이 가능하게 할 수 있는 근거는 1995년 스위스에서 ISO의 유니코드 배정이 확정되어 11172개의 한글 음절을 모두 등록하게 된 결과라고 할 수 있다. 만약 한 글자라도 등록에서 빠진다면 음운변동 프로토콜은 이론적, 실제적인 동력을 잃을 것이다.

3) 겹받침 줄이기: 겹받침은 한 음절의 종성에 자음이 두 개인 경우인데 겹받침 줄이기는 두 개의 자음 중 하나의 자음만 발음하는 규칙이다. 위의 소리이음 규칙과 중복되는 부분이 있는데, 그것은 뒷글자의 초성이 /ㅇ/ 인 경우는 앞의 자음을 발음한다. /ㅇ/이 아닌 경우, 뒷자음이 발음되는 경우는 ㄹ, ㅁ, ㅂ 이고 나머지 ‘ㄲ, ㄴ, ㄹ, ㄷ, ㄸ, ㄹ, ㅌ, ㅍ, ㅎ’ 경우는 앞자음이 발음된다.

표 10은 앞자음을 발음하는 /ㄲ/과 ‘ㅇ,ㅎ’을 제외하고는 모두 /ㄱ/과 초성들과의 음운변동과 같은 ‘ㄱ,ㄲ, ㄴ,ㅋ,’과 만나는 모든 초성들과의 음운변동 코드이다.

Table 10. The case of final consonants {ㄱ, ㄲ, ㅋ, ㄴ}

	ㄱ	ㄲ	ㄴ	ㄷ	ㄸ	ㄹ	ㄺ	ㄻ	ㄼ	ㅅ
ㄱ	101	101	2102	104	104	2105	2106	108	108	110
ㄲ	101	101	2102	104	104	2105	2106	108	108	110
ㅋ	101	101	2102	104	104	2105	2106	108	108	110
ㄴ	101	101	2102	104	104	2105	2106	108	108	110
	ㅅ	ㅇ	ㅈ	ㅊ	ㅌ	ㅋ	ㅌ	ㅍ	ㅎ	
ㄱ	110	1	113	113	114	115	116	117	15	
ㄲ	110	109	113	113	114	115	116	117	15	
ㅋ	110	15	113	113	114	115	116	117	115	
ㄴ	110	2	113	113	114	115	116	117	15	

다음의 표 11은 뒷자음을 발음하는 /ㄹ/과 만나는 모든 초성들과의 음운변동 코드이다. /ㄹ/과 만나는 초성들과의 음운변동은 /ㅇ/과 /ㄱ/을 제외하고는 모두 /ㄱ/과 초성들과의 음운변동과 같다. 또 /ㅁ/과 만나는 초성들과의 음운변동은 /ㅇ/과 /ㄱ/을 제외하고는 모두 /ㅇ/과 초성들과의 음운변동과 같다.

Table 11. The case of final consonants {ㄹ, ㅁ}

	ㄱ	ㄲ	ㄴ	ㄷ	ㄸ	ㄹ	ㄺ	ㄻ	ㄼ	ㅅ
ㄹ	800	101	2102	104	104	2105	2106	108	108	110
ㅁ	1600	1601	1602	1603	1604	1602	1606	1607	1608	1609
	ㅅ	ㅇ	ㅈ	ㅊ	ㅌ	ㅋ	ㅌ	ㅍ	ㅎ	
ㄹ	110	800	113	113	114	115	116	117	15	
ㅁ	1610	806	1612	1613	1614	1615	1616	1617	1618	

4) /ㄴ/의 /ㄹ/되기 : 특별한 몇 가지 경우를 제외하면 대개의 경우 ‘ㄴ+ㄹ’과 ‘ㄹ+ㄴ’은 ‘ㄹ+ㄹ’로 발음된다. 일단 음운변

동표에는 이런 경우에 /ㄴ/을 /ㄹ/로 발음하고 특별한 몇 가지 경우는 따로 모아서 처리하면 표준화가 가능하다. 더불어 위의 접받침 줄이기의 'ㄴ, ㄹ'은 /ㄴ/으로 발음되는 규칙에 따라 /ㄴ/으로 발음되는 3가지 중성을 초성들과의 만나는 발음의 음운변동코드를 살펴본다. 또 /ㄹ/이 'ㄱ, ㄷ, ㅂ'과 만나는 경우 거센소리 발음효과를 가져온다.

Table 12. The case of /ㄴ/ becomes /ㄹ/

	ㄱ	ㄲ	ㄴ	ㄷ	ㄸ	ㄹ	ㅁ	ㅂ	ㅃ	ㅅ
ㄴ	400	401	402	403	404	805	406	407	408	409
ㄴ	400	401	402	403	404	805	406	407	408	409
ㄴ	415	401	402	416	404	805	406	417	408	409
	ㅆ	ㅇ	ㅈ	ㅉ	ㅊ	ㅋ	ㅌ	ㅍ	ㅎ	
ㄴ	410	2	412	413	414	415	416	417	418	
ㄴ	410	412	412	413	414	415	416	417	414	
ㄴ	410	418	414	413	414	415	416	417	418	

5) 복자음 : 표 13은 'ㄹ, ㄹ, ㄹ, ㄹ, ㄹ'이 초성들과 만나는 발음을 표기하는 음운변동표이다. 복자음들은 위에서 선행연구들에 의해 앞자음인 /ㄹ/로 발음된다.

Table 13. The case of final consonants {ㄹ, ㄹ, ㄹ, ㄹ, ㄹ}

	ㄱ	ㄲ	ㄴ	ㄷ	ㄸ	ㄹ	ㅁ	ㅂ	ㅃ	ㅅ
ㄹ	800	801	805	803	804	805	806	807	808	809
ㄹ	800	801	805	803	804	805	806	807	808	809
ㄹ	800	801	805	803	804	805	806	807	808	809
ㄹ	800	801	805	803	804	805	806	807	808	809
	ㅆ	ㅇ	ㅈ	ㅉ	ㅊ	ㅋ	ㅌ	ㅍ	ㅎ	
ㄹ	810	5	812	813	814	815	816	817	818	
ㄹ	810	807	812	813	814	815	816	817	817	
ㄹ	810	809	812	813	814	815	816	817	818	
ㄹ	810	817	812	813	814	815	816	817	817	
ㄹ	810	816	812	813	814	815	816	817	816	

/ㅃ/은 위의 선행연구대로 앞자음인 /ㅂ/으로 발음된다. /ㅂ/으로 발음되는 두 개의 중성과 초성들과의 발음은 표 14와 같다.

Table 14. The case of final consonants {ㅂ, ㅃ}

	ㄱ	ㄲ	ㄴ	ㄷ	ㄸ	ㄹ	ㅁ	ㅂ	ㅃ	ㅅ
ㅂ	1701	1701	1602	1704	1704	1602	1606	1708	1708	1710
ㅃ	1701	1701	1602	1704	1704	1602	1606	1708	1708	1710
	ㅆ	ㅇ	ㅈ	ㅉ	ㅊ	ㅋ	ㅌ	ㅍ	ㅎ	
ㅂ	1710	6	1713	1713	1715	1715	1716	1717	1717	
ㅃ	1710	1709	1713	1713	1715	1715	1716	1717	1717	

6) 일곱 끝소리 되기: 중성의 발음은 'ㄱ, ㄴ, ㄷ, ㄹ, ㅁ, ㅂ, ㅇ' 이렇게 7가지 자음만으로 발음된다는 것이다. 표기형 중성이 'ㄱ, ㅋ, ㆁ' 들은 /ㄱ/으로 발음되고 'ㅂ, ㅍ' 들은 /ㅂ/으로 'ㄷ, ㅌ, ㅈ, ㅊ, ㅅ, ㅆ, ㅎ' 들은 'ㄷ'으로, 나머지는 표기형 그대로 발음된다.

Table 15. The case of final consonants become /ㄷ/

	ㄱ	ㄲ	ㄴ	ㄷ	ㄸ	ㄹ	ㅁ	ㅂ	ㅃ
ㄷ	701	701	402	704	704	805	1606	708	708
ㅌ	701	701	402	704	704	805	1606	708	708
ㅈ	701	701	402	704	704	805	1606	708	708
ㅊ	701	701	402	704	704	805	1606	708	708
ㅅ	701	701	402	704	704	805	1606	708	708
ㅆ	701	701	402	704	704	805	1606	708	708
ㅎ	701	701	402	704	704	805	1606	708	708
	ㅆ	ㅇ	ㅈ	ㅉ	ㅊ	ㅋ	ㅌ	ㅍ	ㅎ
ㄷ	710	710	12	713	713	714	715	716	717
ㅌ	710	710	14	713	713	714	715	716	717
ㅈ	710	710	12	713	713	714	715	716	717
ㅊ	710	710	14	713	713	714	715	716	717
ㅅ	710	710	9	713	713	714	715	716	717
ㅆ	710	710	10	713	713	714	715	716	717
ㅎ	710	710	11	713	713	714	715	716	717

7) 끝소리 중 그대로 발음되는 /ㅁ/과 /ㅇ/이 초성과 만나는 발음은 표 16과 같다.

Table 16. The case of final consonants {ㅁ, ㅇ}

	ㄱ	ㄲ	ㄴ	ㄷ	ㄸ	ㄹ	ㅁ	ㅂ	ㅃ
ㅁ	1600	1601	1602	1603	1604	1602	1606	1607	1608
ㅇ	2100	2101	2102	2103	2104	2105	2106	2107	2108
	ㅆ	ㅇ	ㅈ	ㅉ	ㅊ	ㅋ	ㅌ	ㅍ	ㅎ
ㅁ	1609	1610	6	1612	1613	1614	1615	1616	1617
ㅇ	2109	2110	2111	2112	2113	2114	2115	2116	2118

## IV. Conclusions

본 논문에서는 모든 음운변동의 결과를 출력하여 한글 발음을 표준화 할 수 있는 유니코드상에서의 음운변동표를 구성하였다. 모든 연결된 문자열은 음운변동표에 의하여 어김없이 변화된 값을 출력할 수 있어서 누구나 한글을 표준화된 규칙대로 읽을 수 있는데 도움이 되도록 하였다. 유니코드상에서의 음운변동코드는 조합형 코드상에서의 음운변동과는 코드배열에서부터 다르다. 이론적이고 학문적인 조합형보다는 경영적 측면에서 볼 때 유니코드가 유리한 측면이 있다.

본 논문은 선행연구들에 의한 결과를 적용하여 유니코드 상에서의 음운변동표를 모두 구축하였다. 추후 권위있는 기관의 발음방식의 결정이 있으면 그 기준에 의하여 음운변동표가 수정될 것이다. 이 표를 이용하여 한국어의 발음이 표준화 될 수 있는 바탕이 될 것으로 사료된다.

하지만, 한글은 규칙대로만 읽을 수 있는 것은 아니고 불규칙한 관계가 있다. 예를 들면 ‘갯잎’은 실제로 읽을 때는 ‘갯닙’이지만 위의 음운변동표에 의하면 ‘깨칩’으로 출력된다. ‘무엇을’의 결과는 ‘무어슬’로 제대로 출력되는 현상과 비교할 사항이다. 전체 문장의 0.1%도 안되는 이런 불규칙 현상도 완벽한 한글발음 표준화를 실현하기 위하여 반드시 해결해야 할 부분이다. 이런 부분도 해결이 모두 가능할 것이며 다음으로는 불규칙 현상의 해결방안으로 과제가 이어갈 것으로 사료 된다.

서두에 언급한 것처럼 외국인들에게 한글수업에 있어 가장 어려움을 겪고 있는 부분이 발음이다. 이에 대한 통일된 프로그램이 완성된다면 한글교육분야에서 상당한 이익을 창출하는 아이템이 될 것으로 판단한다. 이러한 연구계기를 통해 본 연구자는 유니코드 상에서의 음운변동표를 이용하여 음운이 변동된 결과를 구현하는데 성공하였으며 모바일을 이용한 어플 등의 보급으로 한국어를 배우려는 모든 이들에게 올바른 한글 발음을 할 수 있도록 도움을 주는데 목표를 가지고 있다. 본고에서는 이론적인 부분에 한정하여 프로그램 구성원리에 한정하여 살펴보았으나 본 연구를 기초로 한 한글 발음의 표준화프로그램을 통해 생성된 어플 등의 생성과 보급, 시장조사를 통한 상업성여부에 대한 논의는 본 연구의 후속작업으로 지속하고자 한다.

## REFERENCE

- [1] <http://blog.daum.net/siyeong6310/982>
- [2] J. C. Park, "Design and development of mobile-based application for learning Hangeul (Korean alphabet)," Sangmyung University. graduation school of education, pp. 38-57, 2013.
- [3] Heo-Ung, "A Study on the Phonemic process of

Heo-Ung," Woorimal, Vol 16 pp. 3-25, 2005.

- [4] Rudolf Lidl, "Finite Field," Encyclopedia of Mathematics and its Applications, vol 20, pp. 4-20, 1983.
- [5] Seymour Lipschutz, "General Topology," Schaum's Outline Series, pp. 19-22, 1965.
- [6] Sherman K Stein, "Calculus and Analytic Geometry," Third Edition Part 1, McGraw-Hill, pp. 31-38, 1982.
- [7] D. S. Kim, "Excel 2000," 21cbook-press, pp. 214-218, 2001.
- [8] S. J. Jang, "A study on the pronunciation education methods for teaching Korean in phonological process : based on the Korean learners' needs analysis," Graduate school of Konkuk University, pp. 76-104, 2014.

## Authors



Cheong-Sick Min received the B.S., M.S. and Ph.D. degrees in Mathematics from Chonnam National University, Korea, in 1987, 1989 and 1992, respectively.

Dr. Min joined the faculty of the Department of Mathematics at Dongshin University, Naju, Chonnam, Korea, in 1991. He is currently a Professor in the Department of Korean Language, Dongshin University. He is interested in education of Korean language, phonological process of Hangeul(Korean alphabet) pronunciation