

논문 2004-41C1-2-6

## 한국어 음가를 한글 표기로 변환하는 표준규칙 제정

(Establishment of the Korean Standard Vocal Sound into Character Conversion Rule)

이 계 영\*, 임 재 걸\*

(Gye-Young Lee and Jae-Geol Yim)

### 요 약

표기 체계인 한글을 한국어 음가로 변환하는 음운변동 규칙을 역으로 적용하여, 한국어 음가를 한국어를 표기하는 문자 체계인 한글로 전환시키는 규칙을 고안하는 것이 본 연구의 목표이다. 제정된 규칙은 한국어 음성인식에 있어서 매우 귀중한 역할을 담당한다. 일반적인 음성인식 기법은 수회의 학습과정을 통하여 추출된 음성의 표준패턴과 인식 대상으로 입력된 음성을 비교하여 가장 유사한 패턴을 찾는 방법을 사용한다. 이때 표준 음성패턴이 띄어쓰기 단위의 어절이라면 수백만 개의 표준 패턴이 수록되어야 하므로 표준패턴을 위한 방대한 데이터베이스의 구축은 물론 표준패턴과의 비교 회수도 너무 많아져서 실용화가 불가능하다. 이에 대한 대안인 음절 단위 인식의 경우는 인식된 음가가 실제의 한글 표기와 맞지 않으므로, 인식된 결과를 출력할 때에 실제의 한글표기로 변환해 주어야 하는 과제를 안게 된다. 이 과제를 해결하는 과정, 즉 일련의 한국어 음가들을 일련의 한글 표기 문자로 바꾸어 주는 과정에서는 본 논문에서 제안한 표준 한국어 음가 - 표기 문자 변환 규칙을 적용할 수 있을 것이다. 본 논문에서는 새롭게 제안된 표준 한글 음가-표기 문자 변환 규칙을 사용하여 한국어 음가를 한글 표기로 변환하는 시스템을 구현하였다. 그리고, 고안된 규칙의 무결성을 보이기 위하여 표준 발음규칙 30항을 반영하는 데이터 집합을 이용하여 구현된 시스템을 시험하였으며, 그 실험 결과를 제시한다.

### Abstract

The purpose of this paper is to establish the Standard Korean Vocal Sound into Character Conversion Rule (Standard VSCC Rule) by reversely applying the Korean Standard Pronunciation Rule that regulates the way of reading written Hangeul sentences. The Standard VSCC Rule performs a crucially important role in Korean speech recognition. The general method of speech recognition is to find the most similar pattern among the standard voice patterns to the input voice pattern. Each of the standard voice patterns is an average of several sample voice patterns. If the unit of the standard voice pattern is a word, then the number of entries of the standard voice pattern will be greater than a few millions (taking inflection and postpositional particles into account). This many entries require a huge database and an impractically too many comparisons in the process of finding the most similar pattern. Therefore, the unit of the standard voice pattern should be a syllable. In this case, we have to resolve the problem of the difference between the Korean vocal sounds and the writing characters. The process of converting a sequence of Korean vocal sounds into a sequence of characters requires our Standard VSCC Rule. Making use of our Standard VSCC Rule, we have implemented a Korean vocal sounds into Hangeul character conversion system. The Korean Standard Pronunciation Rule consists of 30 items. In order to show soundness and completeness of our Standard VSCC Rule, we have tested the conversion system with various data sets reflecting all the 30 items. The test results will be presented in this paper.

**Keywords:** Standard Pronunciation Rule, Speech Recognition, Vocal Sound into Character Conversion, Hangeul Phonetic Value, Petri Net

\* 정회원, 동국대학교 컴퓨터·멀티미디어학과 (Dept. of Computer & Multimedia, Dongguk Univ.)

※ 본 연구는 한국과학재단 목적기초연구(R05-2001-000-00999-0) 지원으로 수행되었음.

접수일자: 2003년3월14일, 수정완료일: 2004년1월16일

## I. 서론

### 1.1 연구 배경 및 필요성

인간과 기계 사이의 효율적 정보 교류를 위한 대화 수단을 찾는 요구가 점점증하고 있다. 이에 따라 우리 인간에 있어서 가장 자연스러운 대화 수단이면서도 특별한 훈련을 요구하지 않는 음성 언어를 인간과 기계사이의 정보전달 수단으로 사용하고자 하는 연구에 관심이 집중되고 있다. 미국의 경우 1950년대 Bell 연구소의 고립숫자 인식 실험<sup>[1]</sup> 이후, 기계를 통한 자동 음성인식을 위한 많은 노력이 있어 왔으며 DARPA의 지원과 함께 음성언어 처리 분야에 있어 상당한 기술이 축적되어 왔다<sup>[2-6]</sup>. 일본의 경우도 이미 60년대 이후부터 하드웨어 음소 분할기의 구현에 관심을 기울여왔으며, 그 이후에도 음성처리에 관한 연구가 지속적으로 이루어지고 있다<sup>[7]</sup>.

이에 비해서 한국어 기반 음성언어 처리는 80년대에 와서야 관심을 끌기 시작했고, 그 이후 많은 연구가 진행되고 있다<sup>[8-13]</sup>. 그러나 한국어 음성인식 분야에 있어서는 한국어의 특성이 배제된 경우가 많았고, 이에 따라 인식률을 높이는 데에도 많은 한계가 있었다. 따라서, 한국어 음성인식 시스템에 있어서 높은 인식 성능을 얻기 위해서는 한글 표기와 한국어 음가간의 변환 특성을 반영한 언어처리 기술이 필수적이라고 사료된다.

본 논문의 목적은 한국어 음가를 문법에 맞는 한글 표기 언어로 변환하는 규칙, 즉 음가-표기 변환 규칙을 제공하려는 데 있다. 한국어 음가는 한글 표기에 대해 음운변동 규칙이 적용되어 생성된 것이다. 따라서, 한국어 음가에 대해 음운변동 규칙을 역으로 적용시키면 해당 음가를 원래의 한국어 표기언어로 생성할 수 있다.

음운변동 규칙의 역, 즉 음가-표기 변환 규칙을 찾기 위해서는 한글 맞춤법의 음운변동 규칙을 정형화하고 통합한 다음, 가능한 모든 한국어 음가에 대한 생성 규칙을 역으로 추적하는 작업이 필요하다. 본 논문은 음운변동 규칙을 정형화하고 통합하는 방법인 표기-음가 생성규칙에 대한 기존 연구에 대해서 간단히 언급하고, 통합 생성된 규칙으로부터 음가-표기 변환 규칙을 생성하는 방법을 기술한다.

### 1.2 기존 연구결과

문법적으로 정확한 한국어 음성을 합성하기 위하여 표준어 규정의 표준발음법<sup>[14-16]</sup>을 준용할 수 있다. 그리

고, 표기 언어인 한글을 한국어 음가로 변환하여 주는 규칙은 표준발음법을 완전하게 반영하여야 하며 또한 무결해야 하는데, 참고문헌<sup>[17,18]</sup>에서는 한국어 음가생성의 기본규칙으로 준용할 표준발음법을 페트리넷<sup>[19]</sup>으로 모델링하여 표준발음법의 일관성을 검증하였다.

그리고 음운변동 현상을 여러 단계로 나누어 차례로 적용한다든지, 변동된 단어에 대하여 처음부터 다시 변환 작업을 재수행하는 기존의 음가 생성 방법의 문제점을 해결하기 위하여 한번의 테이블 탐색으로 모든 음운변동이 완료되는 한국어 음성합성을 위한 음가변환 테이블을 구현하였다.

이와 같이 표준발음법의 일관성이 검증되어 무결한 표준발음법이 정의된 후에 이를 바탕으로 생성된 표기-음가 변환 테이블은 음가생성을 위하여 입력된 한글 단어나 문장의 자모와 형태소 정보만으로 한국어 표준발음을 쉽게 추출할 수 있게 된다. 이러한 방법은 한국어를 표기하는 문자 체계인 한글을 표준 한국어 음가로 변환할 때 발생할 수 있는 개인간의 해석차를 없애고, 형태소 정보와 자모만으로 정확한 음가변환이 이루어질 수 있도록 도와준다.

한글 맞춤법의 표준발음법은 총 7개 장, 30개 항으로 구성되어 있다. 이 30개 항을 모두 통합하면 통합 표준발음법을 작성할 수 있다. 이를 위하여 30개 항의 각각을 형식 언어로 표현하고 이들을 수학적으로 통합한 연구결과가<sup>[18]</sup>에 소개되었다.

표준발음법 각 항의 형식은 한글 단어를 구성하는 음절과 음절 사이에서 어떠한 조건이 발생하면, 어떻게 발음하라는 형식의 문장으로 구성되어 있으므로, 조건을 입력 장소로 그리고 발음되는 음가를 출력 장소로 하는 페트리넷으로 표현할 수 있다.

표준발음법의 제 9항을 페트리넷으로 표현하는 방법을 예로 들어 설명하면, 제 9항의 경우 ‘ㄱ’이 어말에 사용되면 ‘ㄱ’으로, ‘ㄱ’이 어말에 사용되면 ‘ㄱ’으로, ‘ㄱ’이 자음 앞에서 ‘ㄱ’으로, ‘ㄱ’이 자음 앞에서 ‘ㄱ’으로, ... 등 모두 16개의 규칙으로 나뉘어져 있다.

이를 나타내는 페트리넷은 ‘Iㄱ’, ‘Iㄱ’, ‘Iㅅ’, ‘Iㅅ’, ‘Iㅈ’, ‘Iㅈ’, ‘Iㅈ’, ‘Iㅈ’을 각각 레이블로 하는 8개의 입력 장소와 ‘I어말’과 ‘I자음앞’이라는 레이블을 각각 갖는 두개의 입력 장소를 포함하여 모두 10개의 입력 장소 및 레이블이 각각 ‘OJㄱ’, ‘OJㄱ’, ‘OJㅈ’(J는 종성을 의미함)인 3개의 출력 장소, 그리고 ‘Iㄱ’과 ‘I어말’을 입력으로 하고 ‘OJㄱ’을 출력 장소로 하는 변천 901, ‘Iㄱ’과 ‘I자음앞’을

표 1. 표준발음 규칙 제 9항의 페트리넷 근접 행렬.  
Table 1. The Petri net incidence matrix for the 9th item of 'Standard Hangeul Pronunciation Rule'.

R9 ①	901 ②	902	903	904	905	906	907	908	909	910	911	912	913	914	915	916
[ㄱ]③	-1⑦								-1							
[ㅋ]		-1								-1						
[ㅌ]			-1								-1					
[ㅍ]				-1								-1				
[ㅊ]					-1								-1			
[ㅌ]						-1								-1		
[ㅍ]							-1								-1	
⑥																
[어말] ④	-1⑦	-1	-1	-1	-1	-1	-1	-1								
[자음 앞]									-1	-1	-1	-1	-1	-1	-1	-1
⑥																
[OJ] ⑤	1⑧	1							1	1						
[O]ㄷ			1	1	1	1	1				1	1	1	1	1	
[O]ㅂ								1								1
⑥																

표 2. 통합 처리된 음가생성 규칙 테이블의 일부.  
Table 2. A part of Hangeul vocal sound look-up table obtained by intergration process.

종성/초성		ㄱ	ㄲ	ㄴ	ㄷ	ㄸ	ㄹ	ㅁ	ㅂ	ㅃ	..
ㄱ	초성	ㄱ	ㄲ	ㄴ	ㄷ	ㄸ	ㄹ	ㅁ	ㅂ	ㅃ	..
	종성	ㄱ	ㄲ	ㅇ	ㄱ	ㄲ	ㅇ	ㅇ	ㄱ	ㄲ	..
ㄲ	초성	ㄱ	ㄲ	ㄴ	ㄷ	ㄸ	ㄹ	ㅁ	ㅂ	ㅃ	..
	종성	ㄱ	ㄲ	ㅇ	ㄱ	ㄲ	ㅇ	ㅇ	ㄱ	ㄲ	..
ㄴ	초성	ㄱ	ㄲ	ㄴ	ㄷ	ㄸ	ㄹ	ㅁ	ㅂ	ㅃ	..
	종성	ㄴ	ㄴ	ㄴ	ㄴ	ㄴ	ㄹ	ㄴ	ㄴ	ㄴ	..
ㄷ	초성	ㄱ	ㄲ	ㄴ	ㄷ	ㄸ	ㄹ	ㅁ	ㅂ	ㅃ	..
	종성	ㄴ	ㄴ	ㄴ	ㄴ	ㄴ	ㄴ	ㄴ	ㄴ	ㄴ	..
ㄸ	초성	ㄱ	ㄲ	ㄴ	ㄷ	ㄸ	ㄹ	ㅁ	ㅂ	ㅃ	..
	종성	ㄷ	ㄸ	ㄴ	ㄷ	ㄸ	ㄴ	ㄷ	ㄸ	ㄴ	..

입력으로 하고 'OJ'을 출력 장소로 하는 변천 909, 'I'과 '어말'을 입력으로 하고 'OJ'을 출력 장소로 하는 변천 902, ... 등 16개의 변천으로 구성된다.

제 9항을 나타내는 페트리넷의 근접 행렬은 [표 1]과 같다. [표 1]에서 ①~⑧은 테이블을 설명하기 위한 주

석 표시로 ①은 표준 발음법의 항 번호, ②는 항의 각 소항목을 나타내는 트랜지션이다. ③은 장전 조건 1(입력 플레이스), ④는 장전 조건 2(입력 플레이스2), ⑤는 격발 후 출력(출력 플레이스)이며, ⑥은 각 조건의 구분을 위한 공백이다. 그리고 ⑦은 장전 조건을 만족하는 마킹이며 ⑧은 격발을 나타내는 마킹을 표시한다.

이와 같이 표준발음법의 모든 항을 각각 [표 1] 과 같은 형식으로 표현한 후, 정의 1에서 정의한 '+' 연산을 이용하여 이들을 모두 통합한다. 그 후, 상충과 중복된 항을 제거하고, 규칙의 통합 등을 거쳐 생성된 표의 일부를 [표 2]에서 보였다.

정의 1: 두 개의 페트리넷  $PT1 = (P1, T1, F1)$ 과  $PT2 = (P2, T2, F2)$ 가 주어지면,  
 $PT1 + PT2 = (P3, T3, F3)$ 는

다음과 같이 정의된다.

$P3 = P1 \cup P2$   
 $T3 = T1 \cup T2$   
 $F3 = F1 \cup F2$

## II. 음가생성 규칙 테이블로부터의 음가-표기 변환 규칙의 유도 방안

[표 2]에서 행의 label은 앞 음절의 종성을, 열의 label은 다음 음절의 초성을 나타낸다. 또한 [표 2]의 일반항 [i, j]는 앞 음절의 종성이 i이고 다음 음절의 초성이 j이면 각각의 초성과 종성이 일반항에 표기된 바와 같이 각각 변화함을 보이고 있다. 예를 들어, 일반항 [ㄱ, ㄱ]은 음가 [ㄱ, ㄱ]으로 출력되는데 이는 앞 음절 종성이 'ㄱ'이고 다음 음절 초성이 'ㄱ'이면, 다음 음절 초성은 'ㄱ'으로 앞 음절 종성은 'ㄱ'으로 발음됨을 표현한다. 이에 해당하는 예로 '학교'가 '학꾜'로 발음되는 것을 들 수 있다.

따라서, 이를 역으로 적용하면 인식된 한국어 음가가 앞 음절 종성이 'ㄱ'이고 다음 음절 초성이 'ㄱ'이라면 한글 표기는 앞 음절 종성이 'ㄱ'이고 다음 음절 초성이 'ㄱ'이 될 수 있다는 뜻이다. 그런데 출력 음가항목이 [ㄱ, ㄱ]인 항이 유일하지 않고 [ㄱ, ㄱ]항, [ㄲ, ㄱ]항, [ㄴ, ㄱ]항, [ㄷ, ㄱ]항 등도 역시 출력항의 음가 값이 [ㄱ, ㄱ]이 되므로, 인식된 음가가 [ㄱ, ㄱ]이라면 이 음가를 생성





```

char *J[42]={ " ", "받침없음", "ㄱ", "ㄲ", " ", "ㄴ", "ㄷ", "ㄸ", "ㄹ", "ㄺ", "ㄻ", "ㄼ", "ㄽ", "ㄾ", "ㄿ", "ㅀ", "ㅁ", "ㅂ", "ㅃ", "ㅄ", "ㅅ", "ㅆ", "ㅇ", "ㅈ", "ㅉ", "ㅊ", "ㅋ", "ㅌ", "ㅍ", "ㅎ", "ㄱ어간", "ㄴ어간", "ㄷ어간", "ㄹ어간", "ㅁ어간", "ㅂ어간", "ㅅ어간", "ㅇ어간", "복합어자음", "복합어자음", "관형격사성립", "사이시옷"};
char *C[58]={ "NULL", "NULL", "ㄱ", "ㄲ", "ㄴ", "ㄷ", "ㄸ", "ㄹ", "ㄺ", "ㄻ", "ㄼ", "ㄽ", "ㄾ", "ㄿ", "ㅀ", "ㅁ", "ㅂ", "ㅃ", "ㅄ", "ㅅ", "ㅆ", "ㅇ", "ㅈ", "ㅉ", "ㅊ", "ㅋ", "ㅌ", "ㅍ", "ㅎ", "어말", "모음어미", "ㄱ어미", "ㄷ어미", "ㅅ어미", "ㅈ어미", "모음조사", "이조사", "모음접미사", "이접", "히접", "기접", "야접", "여접", "요접", "유접", "아실형", "애실형", "애실형", "어실형", "에실형", "예실형", "오실형", "와실형", "왜실형", "외실형", "우실형", "위실형", "웨실형", "위실형", "으실형", "의실형", "이실형", "야실형", "여실형", "요실형", "유실형"};

```

그림 3. 초성과 종성을 나타내는 요인.  
 Fig. 3. The elements representing the initial sounds and the ending sounds.

```

종성 받침없음, 초성 ㄱ
(ㄱ, ㅁ), (ㄱ, 모음어미), (ㄱ, 모음조사), (ㄱ, 모음접미사),
(ㄱ, 이접미사), (ㄱ, 이조사), (ㄱ, 야접미사), (ㄱ, 여접미사),
(ㄱ, 요접미사), (ㄱ, 유접미사), (ㄱ, 아실질형태소), (ㄱ, 애실질형태소),
(ㄱ, 애실질형태소), (ㄱ, 어실질형태소), (ㄱ, 에실질형태소), ...중략 ...,
(ㄱ, 유실질형태소), (ㄱ, ㅁ), (ㄱ, 아실질형태소), (ㄱ, 애실질형태소), (ㄱ, 애실질형태소),
(ㄱ, 어실질형태소), (ㄱ, 에실질형태소), ... 중략 ...,
(ㄱ, 의실질형태소), (ㄱ, ㅁ), (ㄱ, 아실질형태소), (ㄱ, 애실질형태소), (ㄱ, 애실질형태소),
(ㄱ, 어실질형태소), (ㄱ, 에실질형태소), ... 중략 ...,
(ㄱ, 외실질형태소)

```

그림 4. 출력된 표기집합의 일부 예.  
 Fig. 4. A sample output character set.

가? 넷째, 다음 음절이 문법 형태소인가? 복합어인가? 등이 있다.

이를 세밀히 분석해보면 표기변환 규칙에서 입력되는 음가에 대한 표기생성 규칙들의 정의에 사용될 '종성'과 '초성'의 표기 형태로 96개의 요인이 발생한다.

[그림 3]의 배열을 사용하여 정의한 \*J[42]의 요인 40개는 받침이 될 수 있는 표기를 정의한 것이고, \*C[58]의 요인 56개는 다음 글자 초성이 될 수 있는 표기와 문법

```

Input : 통합된 표준발음법의 행렬
Output : 한글 표기법 행렬 파일
BEGIN
Step 1 : 종성/초성 번호를 갖는 표기유도 규칙 정의 테이블이 남아 있는 동안
Step 2 : 테이블 로딩(파일명n, 저장될 배열명m)
FOR i=0 TO n의 트랜지션 수
IF i번째 트랜지션 번호와 m의 테이블 번호가 같으면
THEN i의 2차원 행렬 행과 열의 번호를 m에 기록
ELSE i 값을 1씩 증가
Step 3 : IF m의 원소가 없으면
THEN 발생할 수 없는 경우로 정의.
ELSE 다음 표기 변환 규칙 테이블 m을 정의후, Step 2로 이동.
Step 4 : m이 Empty가 될 때까지 m에 저장된 결과를 종성과 초성을 테이블 번호로 하여 출력
END

```

그림 5. 음가변환 테이블의 역변환 알고리즘.  
 Fig. 5. The algorithm of reversely converting the vocal sound look up table.

형태소의 기능에 대해서 정의한 것이다. II장에서 설명한 바와 같이 [그림 2]는 한글 맞춤법의 표준발음 규칙에 따라 통합 생성된 음가 생성 규칙을 역으로 정의한 것이다.

따라서, 종성과 초성의 발음 음가는 [그림 2]의 일반항으로 출현하는 경우만 발생한다. 음가 생성 규칙에 대한 검증은 참고문헌[18]에서 입증한 바 있다. [그림 2]의 항목으로 출현하는 각각에 대하여 해당 표기집합을 모두 유도해 보면 [그림 4]와 같이 작성된다.

즉, [그림 4]는 [그림 3]의 요인에 따라서 입력되는 음가에 대해 자모를 분리하고, 분리한 자모를 입력으로 출력 종성과 출력 초성의 항 번호를 표시한 다음, 배열을 이용하여 정의한 테이블을 참조하여 얻어진 출력 결과를 보인 것이다.

### 3.2 표기변환 규칙 생성

[그림 5]에 보인 음가변환 테이블의 역변환 알고리즘은 통합된 표준발음법 행렬 파일을 로드하여 초성과 종

```
(0) (1) */ㄱ(2)*/ㄴ(3)*/ㄷ(4)*/ㄹ(5)*/ㅁ(6)*/ㅂ(7)*/ㅅ(8)*/ㅇ(9)*/(10)*/ㅈ(11)*/ㅊ(12)*/ㅋ(13)*/ㆁ(14)*/(15)*/ㆂ(16)*/ㆃ(17)*/ㆄ(18)*/ㆅ(19)*/(20)*/(21)*/(22)*/(23)*/(24)*/(25)*/(26)*/(27)*/(28)*/(29)*/(30)*/(31)*/(32)*/(33)*/(34)*/(35)*/(36)*/(37)*/(38)*/(39)*/(40)*/(41)*/(42)*/(43)*/(44)*/(45)*/(46)*/(47)*/(48)*/(49)*/(50)*/(51)*/(52)*/(53)*/(54)*/(55)*/(56)*/(57)*/
```

그림 6. 가로축 인덱스.  
Fig. 6. Indices for columns.

```
받침없음(1)*/ㄱ(2)*/ㄴ(3)*/ㄷ(4)*/ㄹ(5)*/ㅁ(6)*/ㅂ(7)*/ㅅ(8)*/ㅇ(9)*/ㅈ(10)*/ㅊ(11)*/ㅋ(12)*/ㆁ(13)*/ㆂ(14)*/ㆃ(15)*/ㆄ(16)*/ㅁ(17)*/ㅂ(18)*/ㅅ(19)*/ㅇ(20)*/ㅈ(21)*/ㅊ(22)*/ㅋ(23)*/ㆁ(24)*/ㆂ(25)*/ㆃ(26)*/ㆄ(27)*/ㅇ(28)*/ㄴ어간(29)*/ㄷ어간(30)*/ㅁ어간(31)*/ㅂ어간(32)*/ㅅ어간(33)*/ㅇ어간(34)*/한자어간(35)*/용언어간말음(36)*/복합어자음(37)*/복합어자음ㄹ(38)*/관형격ㅅ성립(39)*/사이시옷(40)*/
```

그림 7. 세로축 인덱스.  
Fig. 7. Indices for rows.

성 번호로 정의된 음가생성 규칙 테이블이 모두 처리될 때까지 차례대로 Step 2와 Step 3을 수행한다. 즉, 음가생성 규칙에 정의된 트랜지션 번호와 m의 항 번호가 일치하면 이를 배열에 저장하고 종성과 초성의 항 번호를 이름으로 하는 배열 m이 모두 처리될 때까지 처리한다.

Step 3에서는 음가생성 규칙에서 표기가 검색되지 않는 경우에 대해서 이를 발생할 수 없는 경우라고 정의하고, 종성과 초성 번호를 교대로 증가시켜 가면서 Step 2를 m의 정의가 끝날 때까지 수행케 한다. m의 개수는 음가에 대해서 종성으로 올 수 있는 대표 음가 28개(받침없음 포함)와 초성으로 올 수 있는 대표 음가 18개에 대해서 정의되어 있다. 따라서, 총 532개의 표기변환 규칙이 만들어진다.

참고 문헌[18]에 정의되어 있는 음가생성 규칙을 이용하여 표기변환 규칙을 유도한 결과, 전체 532개 중에서 표기변환 가능 테이블은 총 131개였으며, 트랜지션의 수는 2,208개였다. 그리고 발생할 수 없는 테이블은 401개였다.

[그림 6]의 가로축 인덱스와 [그림 7]의 세로축 인덱

```
//종성 받침없음 0, 초성 ㄱ 0
VTable0_0[100]={1,11},{1,20},{1,21},{1,22},{1,27},{1,28},{1,31},{1,32},{1,33},{1,34},{1,35},{1,36},{1,37},{1,38},{1,39},{1,40},{1,41},{1,42},{1,43},{1,44},{1,45},{1,46},{1,47},{1,48},{1,49},{1,50},{1,51},{1,52},{1,53},{1,54},{1,55},{2,11},{2,35},{2,36},{2,37},{2,38},{2,39},{2,40},{2,41},{2,42},{2,43},{2,44},{2,45},{2,46},{2,47},{2,48},{2,49},{2,50},{3,11},{3,35},{3,36},{3,37},{3,38},{3,39},{3,40},{3,41},{3,42},{3,43},{3,44},{3,45},{3,46},{3,47},{3,48},{3,49},{3,50},{6,31},{9,11},{9,35},{9,36},{9,37},{9,38},{9,39},{9,40},{9,41},{9,42},{9,43},{9,44},{9,45},{9,46},{9,47},{9,48},{9,49},{9,50},{24,11},{24,35},{24,36},{24,37},{24,38},{24,39},{24,40},{24,41},{24,42},{24,43},{24,44},{24,45},{24,46},{24,47},{24,48},{24,49},{24,50}}://100
```

그림 8. 종성없음, 초성 'ㄱ'에 대한 음가표기 규칙 표현.  
Fig. 8. The representation of the case of silence ending sound and 'ㄱ' initial sound.

스의 괄호 내 숫자는 항 번호이며, 종성과 초성에 관한 조합형 코드로 정의되어 있는 번호를 참조하여 트랜지션 번호에 대한 표기를 매핑하게 된다. [그림 8]은 알고리즘 [그림 5]를 이용한 출력 결과인 음가-표기 변환 규칙의 일부로써 종성에 받침이 없고, 다음 초성이 'ㄱ'인 경우에 대한 예를 보인 것이다.

#### IV. 음가-표기 변환기의 구현

본 연구에서 제안하는 '음가-표기 변환 규칙'은 음성 이해 시스템의 구현에 유용하게 사용될 수 있다. 음성 이해는 음성인식과 비슷하지만 같지는 않다. 음성인식은 각종 기기의 음성 명령 인식과 같이 극히 제한된 범위에서 사용되는 데 비하여 음성 이해는 대화체 음성 이해와 같은 모든 범주의 음성을 이해하는 것을 내포한다.

즉, 도메인의 제한이 없는 음성 이해를 위해서는 음소 단위로 인식한 결과를 그대로 표기 체계로 변환할 수 있어야 하는데, 예를 들어 '학표'라는 음성 신호가 'ㅎ,

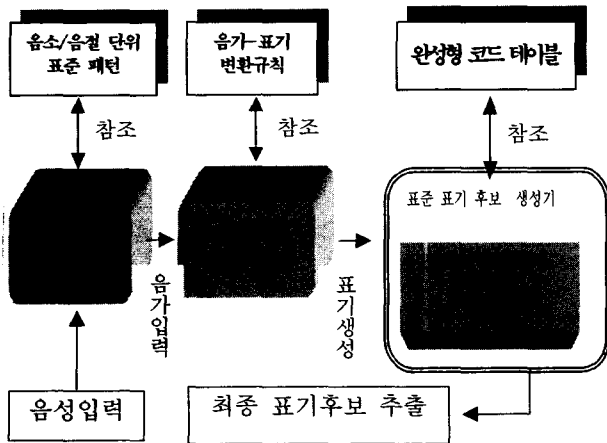


그림 9. 음가 입력에 의한 표기 후보 생성 시스템의 구성도.

Fig. 9. A schematic diagram of the production system outputting candidate characters for input vocal sounds.

ㅏ, ㅑ, ㅓ, ㅕ, ㅗ, ㅛ(ㅇ는 빈글자를 의미함)로 인식된다면, 이 인식 결과를 본 논문에서 제시한 ‘음가-표기 변환 규칙’을 적용함으로써 문법에 맞는 한글 표기를 생성할 수 있다.

즉, 첫음절 음가의 종성과 다음 음절 초성을 ‘음가-표기 변환 규칙’의 좌변에서 찾아 그 규칙의 우변으로부터 한글 표기를 찾으려 한다. 그러나 [그림 1]에 보인 바와 같이 이때 우변에 출현하는 적당한 한글 표기의 후보는 다수인 것이 보통이다. 이 때, 후보 각각에 대한 후처리를 행한 후에 표기 가능한 단어를 사전에서 찾아봄으로써 가장 적당한 후보를 가릴 수 있다.

이와 같은 음가-표기 변환 생성기를 이용하여 표기 후보를 생성하는 시스템의 구성을 도식화 한 것이 [그림 9]이다. [그림 9]에서 양방향 유향 간선은 사전 참조, 유향 간선 아래의 레이블은 자료의 흐름을 표시한다.

4.1 음가-표기 변환 생성기의 구현

음가-표기 변환 생성기의 구현 환경은 변환이 완료된 표기의 출력을 위하여 통합형 한글을 지원하는 Pentium, Windows2000 시스템 상에서 Microsoft Visual C++ 컴파일러 버전 5.0의 C++ 언어로 구현하였다.

[그림 9]에 보인 음가-표기 변환 생성기는 음가-표기 변환 규칙을 참조하여 음성 인식기로부터 인식된 음가를 입력으로 유도할 수 있는 모든 표기집합을 생성하는 시스템이다.

표기집합의 생성은 음가-표기 변환 규칙을 참고하여

```

Input : 자모가 분리된 음소
Output : 표기 집합
int jong, cho;
BEGIN
Step 1: 테이블 로딩(음가-표기 변환규칙 m )
Step 2: 입력된 음소가 Null 일 때까지
        IF 분리된 음소 번호(jong, cho)를 m의
           테이블 번호와 비교
           THEN 테이블에서 표기요인 번호(j, c)를
              배열에 저장
Step 3: 입력 문장이 끝날 때까지 Step1과
        Step2를 반복 처리
Step 4: 배열에 저장된 항 번호와 초성과 종성
        번호를 비교하여 한글 표기로 변환
END
    
```

그림 10. 표기집합 생성 알고리즘.

Fig. 10. The algorithm to produce character set.

표기로 생성될 수 있는 가능한 모든 것이 포함된다. 따라서 표준 표기에 맞지 않는 표기들도 유도되어지며 예를 들어, 입력 음가 ‘학교’에 대해서 유도되어진 표기 가능한 집합은 다음과 같다.

학교, 학교, 학교, 학교, 학교,  
 학교, 학교, 학교, 학교, 학교

[그림 10]은 음가-표기 변환 규칙을 참조하여 입력 음소의 트랜지션 번호와 표기 변환 규칙의 테이블 번호를 비교하여 번호가 같은 테이블의 표기 요인을 모두 출력하는 알고리즘을 보인 것이다.

[그림 10]의 Step 1에서는 ‘음가-표기 변환 규칙을 로드한 다음 Step 2에서 입력된 음가의 자모 번호와 테이블의 항 번호를 비교하여 항 번호가 같은 테이블의 트랜지션을 배열에 저장하고, Step 4에서 배열에 저장된 트랜지션을 표기 종성, 초성의 항 번호를 참조하여 출력한다.

Step 2의 배열에 저장되는 항 번호는 초성과 종성의 번호이며, 프로그램 상에서 전달되는 파라메타들은 항 번호를 참조하여 출력될 때에만 표기로 변환하여 한글 표기로 출력한다.



4. 2 표준 표기 후보 생성기의 구현

[그림 10]의 알고리즘에 의해 생성된 표기 집합들 중, 표준어 표기법에 맞는 표준 표기의 추출을 위하여 비표기 문자 및 비어휘 어절, 중복 유도 어절 등을 제거하는 과정이 필요하다.

즉, 발생할 수 있는 표기 집합들 중 '환'과 같이 일상 생활에 사용되지 않는 문자(또는 표준어 표기에 맞지 않는 문자)가 포함된 표기를 제거하고, 표기 집합들 중 중복 생성된 표기 문자를 제거하여야 한다.

이와 같은 비표기 문자들은 표준 표기집합이라는 어휘 사전을 참조하여 제거하여야 하는데, 인식된 표기가 많을 경우에는 실시간 처리의 어려움이 예상되므로 완성형 코드(KS C 5601)를 참조한 후, 정의되어 있지 않은 문자를 포함하고 있는 어휘를 표기집합에서 제거함으로써 사전을 참조할 최종 표기집합의 수를 감소시킨다.

예를 들어 유도된 표기 집합들 중에 '계회', '계외' 등과 같은 비표기 문자(비완성형 코드)가 포함된 어절(단어)들을 제거하는 과정이 필요한데, "ㄱ + 회"와 "ㅎ + 외"에 관한 완성형 코드를 확인하면, '계'이나 '경'과 같은 문자는 완성형 코드표에 정의되어 있지 않은 문자이므로 이들 단어들이 제거된다. 이와 같이 비표기 및 비어휘 어절을 제거한 후에는 최종적으로 중복 유도된 단어를 표기집합에서 제거시킨다.

[그림 11]의 "학교(중성 ㄱ, 초성 ㄱ)"의 예에서 보이는 것과 같이 표기집합 내에 중복 유도된 표기들이 나타나는데, 이는 중성 'ㄱ'과 초성 'ㄱ'에 대한 유도 규칙 (ㄱ, ㄱ), (ㄱ, ㄱ어미), (ㄱ, ㄱ접미사)에서 표기 중성 'ㄱ', 초성 'ㄱ'이 중복되어 유도되기 때문이다.

[그림 12]는 중복 유도되어진 어절을 제거하는 알고리즘을 보인 것이다. [그림 12]의 Step 1에서 유도된 표기 집합들을 로드하고, Step 2에서 표기집합 중 i번째 단어와 표기집합에 있는 j번째 단어가 같으면 표기집합에 있는 j번째 단어를 제거한다.

단어를 제거한 후에는 j의 값을 차례대로 증가시키고, 표기집합에 대한 검색이 끝나면 i의 값을 증가시켜 표기집합에 있는 단어가 모두 검색될 때까지 반복 수행한다. 이상과 같이 완성형 코드표를 참조하여 비완성형 표기를 포함한 어휘를 제거한 결과와 중복 유도된 표기를 제거한 결과의 일부를 [표 4]에서 보였다.

중성 ㄱ, 초성 ㄱ  
 (ㄱ, ㄱ), (ㄱ, ㄱ), (ㄱ, ㄱ어미), (ㄱ, ㄱ접미사), (ㄱ, ㄱ), (ㄱ, ㄱ), (ㄱ, ㄱ어미), (ㄱ, ㄱ접미사), ( , ㄱ), ( , ㄱ), ( , ㄱ어미), ( , ㄱ접미사), (ㄱ, ㄱ), (ㄱ, ㄱ), (ㄱ, ㄱ어미), (ㄱ, ㄱ접미사), (ㅋ, ㄱ), (ㅋ, ㄱ), (ㅋ, ㄱ어미), (ㅋ, ㄱ접미사), (용언어간말음ㄱ, ㄱ), .....

그림 11. 중성 'ㄱ', 초성 'ㄱ'에 대한 표기 변환.  
 Fig. 11. The character set for the input of initial sound 'ㄱ' and ending sound 'ㄱ'

표 4. 중 제거작업의 효율성을 보이는 표.  
 Table 4. A table to show the effectiveness of the removing steps of our algorithm.

음가 (표기) / 처리 결과	계회기 (계획이)	널파 (널다)	닥파 (닥다)	넉파 (넉과)	학교 (학교)
생성 표기수	106	21	19	22	22
비표기/비어휘 어절 제거후	10	14	13	7	4
중복표기 제거후(최종)	2	4	6	4	2
최초 대비 최종 감소율	98.11	80.95	68.42	81.82	90.91

```

Input : 표기 생성 규칙으로부터 유도된 배열
Output : 중복을 제거한 표기 집합
int i, j = 0;
BEGIN
    Step 1: 표기 집합의 로드
    Step 2: FOR i=0 TO n(표기 집합의 개수)
            i번째 단어를 읽어 표기집합과 비교.
            FOR j=1 TO n(표기 집합의 개수)
                    j번째 단어를 읽어 표기집합과 비교.
                    IF i번째 단어 = j번째 단어
                            THEN j번째 단어를 제거
                                    s = n - 1
                            ELSE : j = j + 1
    Step 3: n = s 로 하고 Step 2로 이동
    Step 4: 중복을 제거한 표기집합 출력
    END
    
```

그림 12. 중복 표기 제거 알고리즘.  
 Fig. 12. The algorithm to remove duplications character.

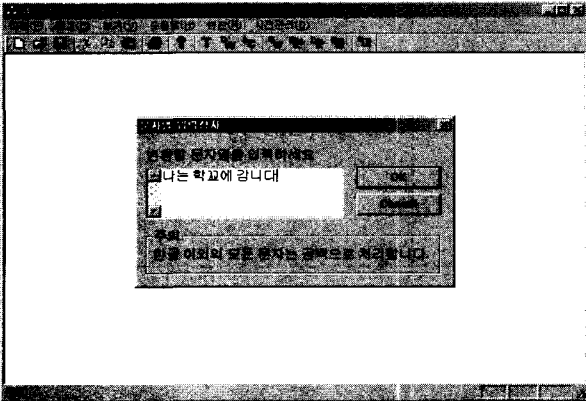


그림 13. 음가 입력용 초기 화면.  
 Fig. 13. The initial screen for keying in symbols of vocal sound.

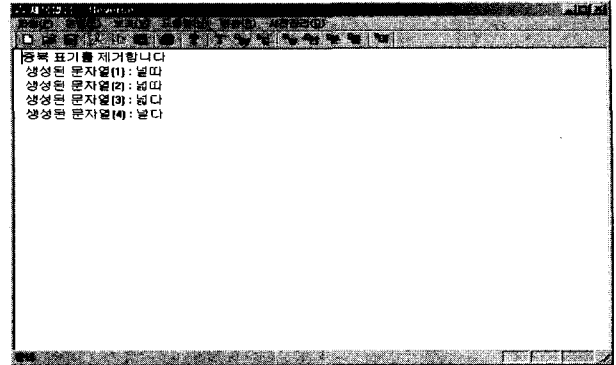


그림 16. 중복 어절 제거후의 최종 후보표기 화면(널따)  
 Fig. 16. The final candidate characters after removing duplications from <Fig. 15>

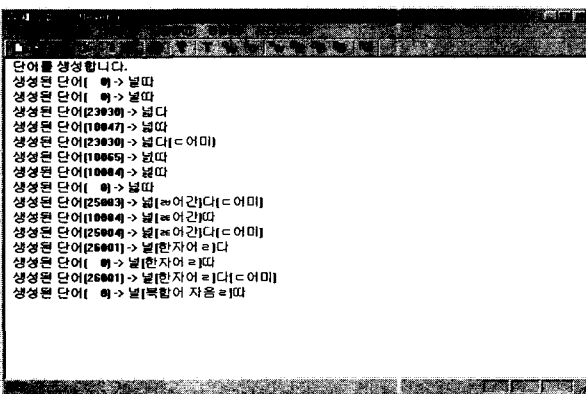


그림 14. 최초 생성 표기집합 화면(널따).  
 Fig. 14. The screen of displaying all the initially produced candidate characters for 널따.

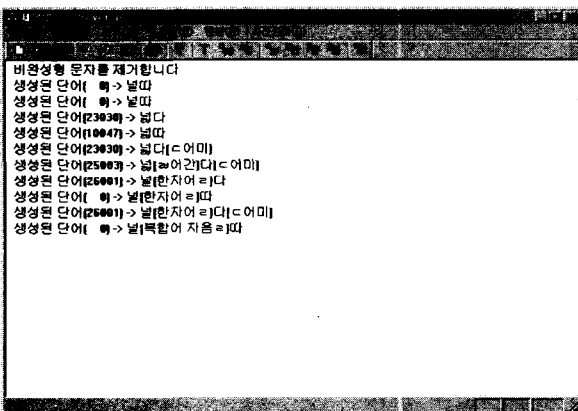


그림 15. 비완성형 문자 제거후 화면(널따).  
 Fig. 15. The result of removing non KSC 5601 characters.

#### 4. 3 한글 표기 후보 출력을 위한 실험 화면

[그림 13]은 구현된 시스템의 음가입력 화면으로 화면 상단에 있는 도구모음 상자의 'T' 키를 입력하여 변환할 음가의 문자열을 입력한다. 도구모음 상자의 'DW' 키는 입력된 음가 문장을 어절(또는 단어) 단위로 분리를 행한 결과를 보여주며, 'DC' 키는 분리된 어절에 대해서 음절 분리와 자모 분리를 행한 결과를 보여준다. 다음의 'GW' 키는 입력된 음가로부터 생성될 수 있는 모든 한글 표기를 표기 변환 규칙을 참조하여 유도한 최초 표기 후보(그림 14)를 보여주며, 'D한' 키는 완성형 코드표를 참조하여 비완성형 문자를 포함하고 있는 어절을 제거한 결과(그림 15)를 보여준다. 다음의 'D문' 키는 문법 형태소 오류를 포함하고 있는 비어휘 어절을 제거한 과정이며, 'D중' 키는 중복 생성된 어절을 제거한 후의 표기 후보(그림 16)를 보여준다.

#### 4. 4 실험 결과

이 실험을 위하여 사용된 음가 데이터의 도메인은 한글 맞춤법 제2부의 표준발음법에 제시된 제 4, 5, 6장의 받침의 발음, 음의 동화, 경음화에 대한 예시어(제9항~제28항) 224개를 대상으로 하였다 그리고, 중복 어절(또는 단어)이 제거된 후의 한글표기 후보에 대한 최종 판정을 위한 구문 해석과 의미 해석 처리는 추후 과제로 미루었음을 밝힌다. [표 5]는 실험을 위해 사용된 표준발음법의 향과 어절수 및 비완성형 표기를 포함한 비어휘 표기와 중복 표기를 제거한 후의 결과를 보인 것이다. 실용적인 음성 이해 시스템은 필연적으로 음절 단위 혹은 음소 단위 인식을 기반으로 하여야 한다. 음절이나



을 제공하는데 있다. 이를 위하여 참고문헌<sup>[18]</sup>에 소개된 통합 발음법 표를 이용하여, 음가-표기 변환 규칙을 제정하였으며 한국어 음성처리(인식)에 있어서 정확도와 유용성을 높이고자 하였다. 음가-표기 변환 규칙과 음가-표기 변환 생성기를 사용하여 유도된 표기 집합들 가운데는 일상생활에 사용되지 않는 표기들이 있다.

이를 제거하고 표준 표기법에 맞는 표기만을 추출하기 위해서 완성형 코드표를 참조하여 비완성형 코드를 포함하고 있는 비표기들 중의 일부를 제거하였고, 표기 변환 규칙에 의해 중복되어 유도되어질 수 있는 표기들 역시 제거하였다. 이를 통하여 한국어 음성인식 분야에서 사용될 수 있는 한글 표준 표기변환과 음성인식 분야의 후처리의 기본 규칙으로 사용될 수 있는 음가-표기 변환 규칙의 유용성을 보였다.

본 논문에서 정의한 방법을 이용하여 음가 생성 규칙에 대해 표기 변환 규칙을 유도한 결과로 총 532개의 음가-표기 변환 규칙 테이블이 생성되었으며, 표기변환 가능 테이블 131개, 발생할 수 없는 경우를 정의한 테이블 401개, 표기변환 테이블에서 표기로 유도될 수 있는 요인 2,028개가 생성되었다.

본 논문은 음성인식 기기를 통하여 추출된 한국어 음가를 표기 가능한 한글로 유도하기 위한 변환 규칙의 생성에 있지만, 비표기 문자 제거와 중복 생성된 표기의 제거 후에도 표준 표기에 맞지 않는 표기가 남아있게 된다.

따라서, 남아있는 비표기 단어 및 어절을 제거하여 정확한 표준 한글 표기로 생성하기 위해서는 형태소 및 구문 분석과 의미 분석을 가능케 하는 구문 및 의미 해석기와의 결합이 필요하며, 자모 단위의 음가 추출이 가능한 음성인식기 연구 분야와의 결합이 앞으로 요구된다.

## 참 고 문 헌

- [1] K. H. Davis, R. Davis and S. Balashek, "Automatic Recognition of Spoken Digits", J. Acoust. Soc.Am., 24(6), 1952.
- [2] F. Jelinek, "Continuous speech recognition by statistical method", Proc. IEEE, vol. 64, pp. 532-556, Apr. 1976.
- [3] C. S. Myers and L. R. Rabiner, "A comparative study of several Dynamic Time Warping Algorithms for Connected Word Recognition", Bell system Tech. J., 60(7) : 1,389-1,409, September 1981.
- [4] S. E. Levinson, "Continuous speech recognition by means of acoustic-phonetic classification obtained from a hidden Markov model," in Proc. ICASSP '87 (Dallas TX), Apr. 1987.
- [5] Mariani, "Recent advances in speech processing", Proc. of Int. Conf. on Acoustics, Speech, and Signal Processing, pp. 429-440, Glasgow, May 1989.
- [6] D. Mansour and B. J. Juang, "A family of discrimination measures based upon projection operation for robust speech recognition", IEEE Trans. on ASSP, Vol. 37, No. 11, pp. 1,659-1,671, 1989.
- [7] K. Kita, Kawabata, and H. Saito. "HMM continuous speech recognition using predictive LR parsing", Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing, Glasgow, Scotland, vol. 2, pp.703-706, 1989.
- [8] 한국과학기술원, 무제한 한국어 음성합성 시스템, 연구보고서, 1990.
- [9] 구명완, 대용량 단어 음성인식 시스템을 위한 화자적응에 관한 연구, 한국과학기술원, 박사학위논문, 1991.
- [10] 김희린, 음성 신호의 부분 정보를 이용한 음성인식 성능 향상, 한국과학기술원, 박사학위논문, 1991.
- [11] 김혜순, 변영태, 이기철, 멀티미디어를 이용한 한국어 발음교육 시스템, 한국정보과학회논문지, 93.12, vol.20, NO1. 12.
- [12] 양진석, 김재범, 이정현, 운율 및 길이 정보를 이용한 무제한 음성 합성기의 설계 및 구현, 한국정보처리학회논문지, 제3권 제5호, 1996.
- [13] S. J. Yun, Y. H. Oh, G. C. Shin, "Improved Lexicon Modeling for Continuous Speech Recognition", International Conference on Acoustics, Speech, and Signal Processing, pp. 1,827-1,830, Munich, Germany, Apr. 1997.
- [14] 문화교육부, 표준어 규정, 문교부 고시 제

88-2호, 1988.

- [15] 이기문 외 9인, 국어 어문 규정집, 대한교과서 주식회사, 1996.
- [16] 서울대학교 사범대학 국어교육연구소, 고등학교 문법, 1996.
- [17] 이계영, 임재걸, 김경징, 페트리넷을 이용한 표준 발음법 분석 시스템 디자인", 한국정보과학회 봄학술발표논문집, pp. 369-371, 1999.
- [18] 이계영, 임재걸, 한국어 음성합성을 위한 음가 변환 테이블 생성, 대한전자공학회논문지, 38CI-5-5, pp. 284-297, 2001.
- [19] T. Murata, "Pertri nets: properties, analysis and applications", Proc. of the IEEE, Vol77. no.4, pp.541-580, April 1989.

---

저 자 소 개

이 계 영(정회원)

제38권 CI편 제5호 참조

임 재 길(정회원)

제38권 CI편 제5호 참조