

한국어 표준발음법의 전산화 및 응용

이계영, 임재걸*†

동국대학교

Gye-Young Lee and Jae-geul Yim. 2003. Computerization and Application of the Korean Standard Pronunciation Rules. *Language and Information 7.2*, 81-101. This paper introduces a computerized version of the Korean Standard Pronunciation Rules that can be used in speech engineering systems such as Korean speech synthesis and recognition systems. For this purpose, we build Petri net models for each item of the Standard Pronunciation Rules, and then integrate them into the sound conversion table. The reversion of the Korean Standard Pronunciation Rules regulates the way of matching sounds into grammatically correct written characters. This paper presents not only the sound conversion table but also the character conversion table obtained by reversely converting the sound conversion table. Making use of these tables, we have implemented a Korean character into a sound system and a Korean sound into the character conversion system, and tested them with various data sets reflecting all the items of the Standard Pronunciation Rules to verify the soundness and completeness of our tables. The test results show that the tables improve the process speed in addition to the soundness and completeness. (Dongguk University)

Key words: 표준발음법(Korean Standard Pronunciation Rules), 음성합성(Speech Synthesis), 음성인식(Speech Recognition), 자연어 처리(Natural Language Processing), 페트리 넷(Petri Net), 한국어 음성처리(Korean Speech Engineering Systems)

1. 머리말

1.1 연구 배경

한글 및 한국어 전산화와 관련하여 최근 한국어 음성합성과 음성인식 분야가 크게 주목받고 있다. 음성합성의 경우에는 한국어를 표기하기 위한 문자체계인 한글을 한국어 표준 음가로 변환하는 과정이 반드시 필요하고, 역으로 음성인식의 경우에는 한국어의 음가를 한국어 표기 체계인 한글로 변환하는 과정이 반드시 필요하다. 이러한 변

* 780-714 경북. 경주시 석장동 707 동국대학교 공학대학 컴퓨터학과, E-mail: lky@dongguk.ac.kr

† 이 연구는 2002학년도 동국대학교 연구년 지원에 의하여 이루어졌음.

환을 규정하는 규범으로 한글 맞춤법 통일안의 표준어 규정 중, 제 2부의 표준발음법(문화교육부 1988)을 이용할 수 있는데, 한국어 음성합성의 경우에는 표준발음법이 그대로 준용되어야 하고, 한국어 음성인식에서는 표준발음법의 역규칙이 준용될 수 있다.

음성인식 연구는 미국의 경우 1950년대 초반부터 고립숫자 음성에 대한 인식 실험을 시작으로 하여 단어 단위의 음성인식, 연속음성인식 등, 기계를 통한 자동 음성인식을 위한 많은 노력이 계속되어 왔으며 음성언어 처리 분야에 상당한 기술이 축적되어 있다. 일본의 경우도 60년대부터 하드웨어적인 음소 분할을 구현할 만큼 큰 관심을 가져 왔고, 그 이후 일본어 정보처리 기술에 대한 기초 연구를 거쳐 상당한 수준의 음성처리 기술을 확보하고 있다(K.H. Davis 1952, D. Klatt 1977, W. Lea 1980, K. Kita 1989, R. Lippmann 1997). 이에 비해서 한국어 기반 음성 언어처리는 80년대에 와서야 관심을 끌기 시작했고, 그 이후 많은 연구가 진행되어 왔지만, 한국어의 특성이 배제된 채, 영어나 일본어에 적용된 음성처리 기술을 그대로 채택한 경우가 많았기 때문에 음성처리를 위한 하드웨어적인 공간성 문제 해결과 인식률을 높이는 데에 많은 한계가 있었다.

한국어의 경우, 곡용과 활용이 자유롭기 때문에 어미 활용이 심하며, 불규칙 현상이나 파생어 및 합성어의 발생, 음운변화 현상이 매우 발달되어 있다는 특징을 갖고 있다. 이런 특징들 중에서 음운현상은 한 음절의 초성과 중성, 중성과 중성, 종성과 다음 음절의 초성이 닿았을 때에 음소 연결에 관한 제약성을 갖고 있으며, 음운이 있는 곳이면 어디서든 일어나게 된다. 또한 형태소의 음소가 놓이는 환경에 따라 다른 음소로 바뀌는 경우는 형태소의 경계에서 발생한다(허웅 1982, 남광우 1989, 이기문 2000). 이와 같은 한국어의 ‘음운변화 현상’은 변동 자체의 성질에 따라 여러 가지 작용으로 나뉘며, 변동이 일어나는 원인으로는 음소 결합의 제약성, 발음 편의를 위한 자연성, 청취 효과에 따른 명확성 등을 들 수 있다.

따라서 한국어를 위한 음성 인식을 제고와 음성 합성음의 자연스러움을 향상시키기 위해서는 음운변화, 운율, 동화현상, 조음결합, 음가변화 등에 대한 한국어 특유의 특징과 구조에 대한 체계적 연구가 있어야 하며, 특히 한국어 음운변화 현상에 대한 특징 규명은 한국어 음성처리의 필수 요소라 할 수 있다. 즉, 한국어 음성처리 시스템에 있어서 높은 성능을 얻기 위해서는 한국어 특성을 반영한 언어처리 기술이 필수적이며 따라서 앞으로 많은 연구가 진행되어야 할 것이다.

1.2 기존 연구 및 연구의 범위

인간 사이의 의사를 전달하는 가장 중요한 수단으로 말과 글이 사용된다. 말이 문자(글)를 음성 언어화하는 기호 처리계라 한다면, 글은 음성 언어(말)를 문자 형태의 표기 체계로 바꾸어주는 인간에 의한 소프트웨어 시스템이라 할 수 있다. 따라서 컴퓨터의 음성인식 기술을 이용하여 추출한 한국어 음가를 그에 대응하는 표기 체계인 한글로 출력시켜 주거나, 입력된 한글 텍스트를 한국어 음가로 변환시켜 음성합성을 시도하는 경우에는 해결해야 할 여러 가지 문제가 나타난다.

본 논문에서는 한국어의 제 특징 중, 음운변화 현상을 효율적으로 처리할 수 있는 시스템을 설계하고 구현하여, 음성인식 시스템에 의해 추출된 한국어 음가를 정확한 한국어 표기 체계인 한글 문자로 변환시켜 주는 과정(음성인식 시스템의 후처리)에서 유용하게 사용될 수 있음을 보이고자 한다. 또한, 입력된 한글 단어나 문장을 한국어의 정확한 음가 정보로 변환시켜 한국어 음성합성 시스템의 전처리 과정에서 유용하게 이용될 수 있음을 보이고자 한다.

기존 연구인 한국과학기술원(1980)에서는 모음 변환, ㄴ첨가, 경음화, ㅎ탈락, 절음 법칙, 연음 법칙, 구개음화, 대표음, 자음 접변 등의 순서를 반복적으로 적용시켜 음운변화를 일으키며, 김혜순(1993)은 한국어 음운변화에 관한 지식을 지식 베이스로 구성하여 전진 추론 방식으로 탐색해 나가면서 어절의 변화가 있는 동안에는 새로운 상태를 만들고, 이 같은 과정을 반복하여 음운변화를 일으키게 하였다.

양진석(1996)은 음운 규칙의 적용 순서를 ㄴ첨가, 구개음화, 음운 축약, ㅎ변환, 자음동화, 경음화, 말음법 및 연음 법칙으로 정하고, 규칙을 적용할 때에는 음운 규칙들이 임의의 어절에 대해 음운 규칙이 하나라도 적용되면 그 어절은 변화되었으므로, 처음부터 다시 음운 규칙을 적용하고, 더 이상의 변화가 없을 때까지 반복하여 적용시키는 구조를 가지고 있다.

다음으로 김재홍(1998)은 한국어 음운변화 과정을 구개음화, 자음 축약, 연음 규칙, 자음 동화, 경음화, 끝소리 규칙의 순서로 적용하여 음운변화를 처리하고 있다. 이와 같이 기존의 연구에서는 표준발음법에 나타나는 음운변화 현상을 여러 단계로 나누어 차례대로 적용시키고, 이때 변동된 단어에 대하여 처음부터 다시 반복하여 변환하는 작업 수행 구조를 가지고 있다. 그리고 음운변화를 수행하는 순서 역시, 증명된 변환 순서가 없이 시스템 구현자에 따라 임의의 순서로 음운변화 현상을 적용시켜, 음운변화 처리의 적용 순서가 바뀔에 따른 음운변화 결과의 정확성 검증이 이루어지지 않은 상태이다.

이러한 방법으로 구현된 음운변화 시스템은 음가 변환의 무결성을 보장할 수 없으며, 음성합성 및 음성인식 시스템의 전처리 및 후처리 결과를 보장하지 못하는 문제가 나타날 수 있다. 따라서 본 논문에서는 자연어로 적시된 표준발음법을 그대로 이용하여 한국어 표기 문자를 한국어 음가로 변환하는 규칙을 하나의 테이블로 통합 생성하고, 이 음가 변환 테이블을 역으로 분석하여 입력된 한국어 음가를 한국어 표기 문자로 역변환하는 테이블을 생성할 수 있음을 보인다. 그리고 생성된 테이블이 한국어 음가와 한국어 표기 문자인 한글 사이에 발생하는 음운변화 현상을 신속히 처리할 수 있는 효율성과 완전성과 무결성 등을 보이고자 한다.

본 논문에서 수행한 '음가 변환 테이블'의 작성 과정은 다음과 같다. 표준발음법의 모든 항을 각각 페트리 넷(T. Murata 1989, 이계영 외 2001)으로 모델링한 다음, 이들을 모두 통합함으로써 표준발음법을 완전히 표현하는 한 개의 통합된 페트리 넷을 생성하고, 이 페트리 넷에 대한 일관성을 분석한다. 일관성 분석의 결과로 나타나는 중복항에 대해서는 두 개의 항 중 하나를 삭제하고, 선택항은 원칙을 취하고 허용항을 삭제하며, 상충항은 적용의 우선순위를 부여하거나 잘못된 정의를 삭제한 다음, 최종

적으로 규칙에 사용된 용어들을 인덱스로 하는 ‘음가 변환 테이블’을 생성한다.

이 테이블은 음가 생성을 위하여 입력된 한글 단어나 문장의 자모와 형태소 정보만으로 한국어 표준 발음 음가를 쉽게 추출할 수 있게 하는데, 이러한 방법은 한국어 표기체계인 한글을 한국어 음가로 변환할 때 발생할 수 있는 연구자 개인간의 해석차를 없애고, 형태소 정보와 자모만으로 정확한 음가 변환이 이루어질 수 있도록 도와준다.

즉, 한국어 음성합성 시스템의 전처리 과정에 적용될 수 있는 음가 변환 과정에 있어서 이 테이블은 1회의 통합 테이블 검색으로 모든 한국어 음가 변환이 완료되므로 검증되지 않은 임의의 순서로 한 단계씩 음가 변환을 시켜나가는 기존의 연구 방법에서 발생할 수 있는 음가 생성의 무결성 입증 문제와 다단계로 처리하는 과정상에서의 시간적 비효율성 문제를 해결할 수 있다.

다음으로 본 논문에서는 ‘음가 변환 테이블’의 모든 엔트리 각각에 대하여 그 엔트리를 지정하는 인덱스들을 찾아 매핑한 다음, 이들을 모두 통합하여 한국어 음가를 한국어 표기(한글)로 변환하는 테이블(이하, ‘문자 변환 테이블’로 기술함)을 생성하는 과정을 소개한다. 이 테이블은 일련의 한국어 음가들을 일련의 한글 문자로 바꾸어 주는 음성인식 시스템의 후처리 과정에서 사용하면 단어 단위는 물론 어절 단위의 한국어 음가를 한글 문자로 변환하는 것이 가능해진다. 생성된 ‘음가 변환 테이블’과 ‘문자 변환 테이블’의 완전성, 무결성, 효율성을 보이기 위하여 이 테이블을 이용한 ‘한국어 표기 - 한국어 음가’ 변환 시스템과 ‘한국어 음가 - 한국어 표기’ 변환 시스템을 구현하고, 실험 결과를 제시한다.

2. 표준발음법의 페트리 넷 모델

2.1 표준발음법의 페트리 넷 모델링

한글 맞춤법 제 1항에는 “한글 맞춤법은 표준어를 소리대로 적되, 어법에 맞도록 함을 원칙으로 한다.”라고 되어 있고, 표준발음법의 제 1항에는 “표준발음법은 표준어의 실제 발음을 따르되, 국어의 전통성과 합리성을 고려하여 정함을 원칙으로 한다.”라고 되어 있다. 즉, 한국어 표기 문자(한글)는 표준어를 대상으로 발음대로 충실하게 적고, 문법에 맞게 적도록 하고 있으며, 한국어를 발음할 때에는 실제 발음과 국어의 제 특질을 반영하라는 의미로써 한국어 표기 문자인 한글과 한국어 발음 사이에서 한글 문자의 내용이 실제 발음과 다르게 나타날 수 있다는 한국어의 특징을 규정하고 있다.

본 논문에서는 표준발음법의 제 4, 5, 6, 7장에 제시된 받침의 발음, 음의 동화, 경음화, 음의 첨가 규정을 정형화된 규칙으로 생성하고자 한다. 그리고 표준발음법이 숙련가 시스템의 규칙들처럼 조건과 결론의 형식을 취하고 있으므로 이러한 지식 표현을 정형화하기 위한 수단으로 사용되는 페트리 넷을 표준발음법의 정형화 도구로 사용한다.

표준발음법은 한국어 표기(한글)의 각 경우에 대하여 발음 음가를 규정하고 있는데, 예를 들어 [표 1]에 보이는 표준발음법 제 10항은 겹받침(ㄱ, ㅋ, ㆁ, ㄷ, ㅌ, ㄹ),

제10항 겹받침 ‘ㄱ’, ‘ㄴ’, ‘ㄹ’, ‘ㄷ’, ‘ㄹ’, ‘ㅂ’은 어말 또는 자음 앞에서 각각 [ㄱ, ㄴ, ㄹ, ㅂ]으로 발음한다.

넋[넉] 넋과[넉과] 앓다[안따] 여덟[여덜] 넓다[널따]
외곶[외곶] 할다[할따] 값[갑] 없다[업:따]

다만, ‘뱌-’은 자음 앞에서 [뱌]으로 발음하고, ‘넙-’은 다음과 같은 경우에 [넙]으로 발음한다.

- (1) 뱌다[뱌:따] 뱌소[뱌:쏘] 뱌지[뱌:찌]
 뱌는[뱌:는 → 뱌:는] 뱌게[뱌:께] 뱌고[뱌:꼬]
(2) 넙-죽하다[넙쭈카다] 넙-둥글다[넙똥글다]

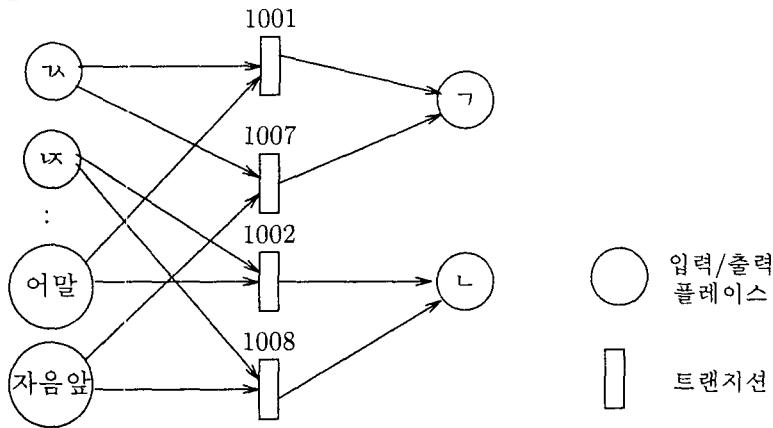
[표 1] 표준발음법 제 10항

ㅂ)이 어말 또는 자음 앞에서 어떻게 발음되는가에 대한 규칙을 제시하고 있다. 이와 같은 조건절을 페트리 넷으로 표현하려면 조건절 자체를 트랜지션으로, 조건부를 이 트랜지션의 입력 플레이스로, 그리고 결론부를 출력 플레이스로 나타내면 된다.

그리고 제 10항은 겹받침에 대한 일반항과 함께 ‘뱌-’과 ‘넙-’에 대한 특수항으로 구분되어 있다. 일반항은 겹받침 ‘ㄱ’, ‘ㄴ’, ‘ㄹ’, ‘ㄷ’, ‘ㄹ’, ‘ㅂ’이 어말에 나타나거나 또는 다음 음절의 초성이 자음이라면 ‘ㄱ’은 ‘ㄱ’으로, ‘ㄴ’은 ‘ㄴ’으로, ‘ㄹ’, ‘ㄷ’, ‘ㅂ’은 ‘ㄹ’로, ‘ㅂ’은 ‘ㅂ’으로 바꾸어 발음하라는 의미이다. 이를 표현하는 페트리 넷은 겹받침 6개, 어말 조건 1개, 다음 음절의 초성 자음 18개로 구성된 25개의 입력 플레이스와 변동될 음가(ㄱ, ㄴ, ㄹ, ㅂ)를 나타내는 4개의 출력 플레이스 및 120개의 트랜지션으로 구성된다.

특수항에 기술된 ‘뱌-’은 자음 앞에서 [뱌-]으로, ‘넙-’은 ‘넙죽하다’와 ‘넙둥글다’의 경우에 각각 [넙쭈카다]와 [넙똥글다]로 발음한다라는 조항이 있다. 따라서 일반항의 규정에 따르면 ‘뱌-’은 [뱌]로 발음되어야 하지만 특수항의 규정에 따라서 [뱌-]으로 발음되어야 하며, 마찬가지로 ‘넙-’의 경우에도 특수항의 조건에 맞으면 [넙]으로 발음하여야 한다. 이와 같은 예외적 사항은 예외 사전으로 구축하여 별도로 처리하며, 나머지 항들은 정의된 용어와 용어의 내용에 따라 각각의 페트리 넷으로 구성한다.

[그림 1]은 표준발음법 제 10항에서 발생하는 120개 트랜지션 중에서 4개(1001, 1002, 1007, 1008)만을 통합하여 페트리 넷으로 표현한 것이다. 그림에서 입력 플레이스에 놓여지는 초기 마킹은 입력되는 문자열의 형태소 분석 정보와 자모 분리 결과에 따라 동적으로 놓여지게 되는데, 예를 들어 입력 문자열로 ‘넋’이라는 글자가 입력되면 형태소 분석과 자모 분리 결과로 ‘어말’임을 인식하고 받침 ‘ㄱ’을 이들에 해당되는 입력 플레이스에 토큰으로 놓여진다. 그러면 트랜지션 1001이 활성화(enable)되어, 트랜지션 1001이 격발되고 출력 플레이스에 ‘ㄱ’이 토큰으로 놓여지게 된다.



[그림 1] 제 10항의 소항목 1001, 1002, 1007, 1008이 통합된 페트리 넷

2.2 페트리 넷 모델의 근접 행렬 변환

[그림 1]과 같이 작성된 페트리 넷 모델을 [표 2]와 같은 근접 행렬(incidence matrix)로 표현하게 되면, 해당 페트리 넷 모델에 대한 분석이 용이해지며 표준발음법의 모든 항에 대한 페트리 넷 모델의 통합도 용이해진다. [표 2]는 페트리 넷으로 표현된 표준발음법 제 10항을 근접 행렬로 변환한 것으로, 입력 플레이스 중 '자음앞'은 18개의 자음으로 확장되어 모델링된다.

R10 ①	1001②	1002	1003	1004	1005	1006	1007	1008	1009	1010	1011	1012
I가 ③	-1⑦						-1					
I나		-1						-1				
I래			-1						-1			
I라				-1						-1		
I레					-1						-1	
I바						-1						-1
⑥												
I어말④	-1⑦	-1	-1	-1	-1	-1						
I자음앞④							-1	-1	-1	-1	-1	-1
⑥												
OJㄱ⑤	+1⑧						+1					
OJㄴ		+1						+1				
OJㄷ			+1	+1	+1				+1	+1	+1	
OJㅂ						+1						+1

[표 2] 표준발음법 제 10항의 페트리 넷 근접 행렬

[표 2]의 근접 행렬에서 ①~⑧은 테이블을 설명하기 위한 주석 표시로, ①은 표준 발음법의 항 번호, ②는 항의 각 소항목을 나타내는 트랜지션 번호, ③은 장전 조건 1(입력 플레이스1), ④는 장전 조건 2(입력 플레이스2), ⑤는 격발 후 출력(출력 플레이스), ⑥은 각 조건을 구분하기 위한 공백, ⑦은 해당 소항목의 조건부(-1), ⑧은 해당 소항목의 결론부(+1)를 의미한다. 표준발음법의 모든 항에 대하여 [표 2]와 같이 완성된 각각의 근접 행렬은 페트리 넷의 통합과 일관성 검증 프로그램의 입력 자료로 사용하기 위하여 완성형 텍스트 형태로 저장된다.

2.3 페트리 넷 모델로 표현된 표준발음법의 일관성 검증

표준발음법에 나타나는 문제점은 크게 세 가지로 나눌 수 있다. 그 첫째는 정의된 두 항의 내용이 상충되는 경우로 제 9항(받침의 발음)과 18항(음의 동화)에서처럼 정의된 내용 중의 일부가 서로 다른 음가를 요구하는 경우로, 예를 들어 제 9항에서는 “받침 ‘ㄱ’이 자음 앞에서 대표음 ‘ㄱ’으로 발음한다”라고 되어 있고, 제 18항에서는 “받침 ‘ㄱ’이 자음인 ‘ㄴ, ㄹ’ 앞에서 ‘ㅇ’으로 발음한다”라고 되어 있다. 이런 경우(상충 항)에는 어느 한쪽에 우선순위를 부여하거나, 해석의 범위를 한정짓거나, 정의를 수정해 주어야 한다.

두 번째는 정의된 두 항의 내용이 중복 정의(중복 항)된 경우로 예를 들어 제 12.3항과 제 18항에서 “받침 ‘ㅎ’과 초성 ‘ㄴ’이 만날 때, 받침 ‘ㅎ’을 ‘ㄴ’으로 발음한다”라고 중복 정의되어 있는 경우이다. 그리고 제 23항과 30.1항의 경우처럼 사이시옷이 아니더라도 ‘ㅅ’은 ‘ㄱ, ㄷ, ㅂ, ㅈ, ㅊ’과 만나면 경음화가 일어나는데, 30.1항에서 사이시옷이라고 범위를 한정지어 같은 내용을 다시 정의한 경우가 이에 해당한다.

세 번째는 원칙과 허용에서 선택의 문제가 발생하는 경우로 제 10항과 제 20항의 “다만”처럼 원칙과 허용을 두어, 두 가지가 모두 표준발음법에 맞는 경우, 원칙과 허용 중 어느 것을 취하고 어느 것을 버릴지 선택해야 하는 경우가 이에 해당한다. 이와 같은 문제를 해결하기 위한 표준발음법의 일관성 검사는 표준발음법의 각 항별로 작성된 각각의 근접 행렬 테이블 파일을 읽어 들여 하나의 테이블로 통합하고, 통합된 결과를 트랜지션 단위로 비교하면서 일관성 여부를 검사하게 되는데, 이를 위한 시스템은 테이블 통합 모듈과 일관성 검사 모듈로 구성된다.

3. 음가 변환 테이블 생성과 실험

표준발음법 제 8항에서 제 30항까지의 각 항에 대하여 [표 2]와 같은 페트리 넷 근접 행렬을 만든 다음, 이들을 모두 통합하는 과정(그림 2)을 거친다. 두 개의 페트리 넷 근접 행렬을 1개의 근접 행렬로 통합한다는 것은 다음과 같이 정의할 수 있으며, 이와 같은 과정을 연속적으로 적용하여 모든 근접 행렬을 하나의 근접 행렬로 통합하게 된다.

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

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

```

입력 : 표준발음법 각 항에 대한 근접 행렬 파일
출력 : 통합된 근접 행렬 파일
파일 x : 각 근접 행렬의 트랜지션 통합후, 저장할 통합 파일명
파일 n : 통합 대상이 되는 근접 행렬의 파일명
파일 m : 통합 대상 근접 행렬을 입력하여 저장할 배열명
정수형 i=0, j=0, k=0
BEGIN
  단계 1 : While 처리되지 않은 테이블이 남아 있는 동안, 단계 2, 3을 수행
  단계 2 : 테이블 로딩(화일명n, 저장될 배열명m)
  단계 3 : FOR i=0 TO n의 트랜지션 수
            n의 i번째 트랜지션과 트랜지션에 연결된 플레이스를 읽어 들임
            FOR j=0 TO x의 트랜지션 수
            x의 j번째 트랜지션과 트랜지션에 연결된 플레이어의 load
            IF i번째 트랜지션의 플레이어와 j번째 트랜지션의 플레이어가
              같으면 THEN
              BEGIN
                통합 근접 테이블 X에 기록(트랜지션, 플레이어)
                k=플레이어의 마지막 트랜지션 위치값
                IF 입력 플레이어 THEN k+1 = -1
                IF 출력 플레이어 THEN k+1 = +1
              END
            ELSE x에 새로운 플레이어 추가후 GOTO 단계 3
  단계 4 : 통합 근접 테이블 출력(화일명x)
END

```

[그림 2] 테이블 통합 알고리즘

[그림 2]의 단계 1은 입력되는 테이블의 마지막 트랜지션이 처리될 때까지 순서대로 단계 2에서 단계 3까지를 반복 수행하라는 것이다. 단계 2는 처리할 대상 파일로부터 하나의 근접 행렬을 읽어 들여 배열에 저장하며, 단계 3은 저장된 배열에서 하나의 트랜지션과 그 트랜지션에 연결된 입력 플레이어 1, 2와 출력 플레이스를 읽어 와서 통합된 테이블에 이미 기록된 플레이어인가를 비교하는 루틴으로 입력 플레이어가 이미 기록된 플레이어이면 그 플레이어가 위치하는 행의 마지막 위치에 트랜지션 번호 및 '-1' 값을 기록하고, 출력 플레이어인 경우에는 '+1' 값을 기록하게 한다.

통합 대상이 되는 근접 행렬 중에서 첫 번째 입력되는 근접 행렬은 단계 3의 ELSE 문에 의하여 배열 x에 모든 플레이스를 추가하게 되며, 두 번째 근접 테이블 부터는 배열 x의 플레이어와 새로 읽어 들인 m의 배열을 비교하여, 새로운 플레이어 나 마킹을 추가시켜 나가면서 각각의 파일로 저장된 근접 행렬을 통합해 나간다. 단계 4는 통합 테이블을 출력시키는 부분이다.

이와 같은 과정을 통하여 페트리 넷 근접 행렬의 모든 항을 통합하면 제 2장 3절에서 기술하였듯이 중복된 항과 상충되는 항이 발생할 수 있다. 즉, 중복 항은 근접 행렬 상에서 입출력이 동일한 두 개의 트랜지션으로 나타나고, 상충 항은 입력이 동일하고 출력은 상이한 두 개의 트랜지션으로 나타난다.

```

입력 : 통합된 근접 행렬 파일
출력 : 검증 결과 출력 파일
배열 Table : 통합된 근접 행렬을 저장할 배열
정수형 n, m=0;
BEGIN
  단계 1 : Table ← 통합된 근접 행렬 X
  단계 2 : FOR m=0 TO Table의 마지막 트랜지션
            Table의 트랜지션m과 연결된 플레이스 선택
  단계 3 : FOR n=1 TO Table의 마지막 트랜지션
            Table의 트랜지션n과 연결된 플레이스 선택
  단계 4 : IF m!=n인 트랜지션m의 플레이스1과 트랜지션n의 플레이스 1이
            같으면
            THEN IF 트랜지션m의 플레이스2와 트랜지션n의 플레이스 2가
                   같으면
                   THEN IF 트랜지션 m, n의 출력 플레이스가 같으면
                          THEN printf("(중복)", 트랜지션m, 트랜지션n)
                          ELSE printf("(상충)", 트랜지션m, 트랜지션n)
  단계 5 : 검증 결과 출력
END

```

[그림 3] 상충 및 중복 검사 알고리즘

[그림 3]은 통합된 테이블을 대상으로 중복 또는 상충이 발생하는 트랜지션을 검사하는 알고리즘으로, 단계 2에서 Table 배열의 첫 번째 트랜지션부터 하나의 트랜지션을 읽어 온 후, 연결된 입력 플레이스와 출력 플레이스를 읽어 들이고, 단계 3에서는 두 번째 트랜지션부터 하나의 트랜지션과 연결된 입력 플레이스와 출력 플레이스를 선택한다.

단계 4에서는 단계 2와 단계 3에서 읽어 들인 트랜지션을 비교하여 입력 플레이스 1과 입력 플레이스 2가 같고 출력 플레이스가 틀리는 경우(상충)에 대한 정의를 하며, 입력 플레이스 1과 입력 플레이스 2, 출력 플레이스가 모두 같으면 중복된 정의로 판별한다. 단계 5에서는 통합된 근접 행렬의 일관성 검증 결과를 출력한다.

[그림 4]는 이와 같은 일관성 검사 출력의 일부를 보인 것으로 실험 결과, 표준발음법의 18항과 9, 10, 11항 및 12항과 9, 11, 13, 14항 그리고, 23항과 10항 등은 동일한 조건하에서 서로 다른 음가 변환이 이루지는 상충 항으로 발견되었으며, 18항과

```

:
(상충) 트랜지션 18003, 트랜지션 9011 : 동일한 조건에 다른 출력을 갖는
규칙임.
TRnum ==>18003 , Place1 IJπ, Place2 IC↳, OutPlace OJ◦
TRnum ==> 9011 , Place1 IJπ, Place2 IC↳, OutPlace OJ↗
/* 18003 : π + ↳ -> ◦ : 닳는 -> 당는(O) */
/* 9011 : π + ↳ -> ↗ : 닳는 -> 닳는(X) */
(상충) 트랜지션 18005, 트랜지션 9029 : 동일한 조건에 다른 출력을 갖는
규칙임.
TRnum ==>18005 , Place1 IJ↻, Place2 IC↳, OutPlace OJ◦
TRnum ==> 9029 , Place1 IJ↻, Place2 IC↳, OutPlace OJ↗
/* 18005 : ↻ + ↳ -> ◦ : 부엌난간 -> 부엌난간(O) */
/* 9029 : ↻ + ↳ -> ◦ : 부엌난간 -> 부엌난간(X) */
<중략>
(중복) 트랜지션 18012, 트랜지션 12023 : 동일 조건, 동일 출력을 갖는 중복
정의임
TRnum ==> 18012, Place1 IJㅎ, Place 2 IC↳, OutPlace OJ↳
TRnum ==> 12023, Place1 IJㅎ, Place 2 IC↳, OutPlace OJ↳
/* 18012 :ㅎ + ↳ -> ↳ : 놓는 -> 논는(O) */
/* 12023 :ㅎ + ↳ -> ↳ : 놓는 -> 논는(X) */
:

```

[그림 4] 일관성 검사의 출력 결과

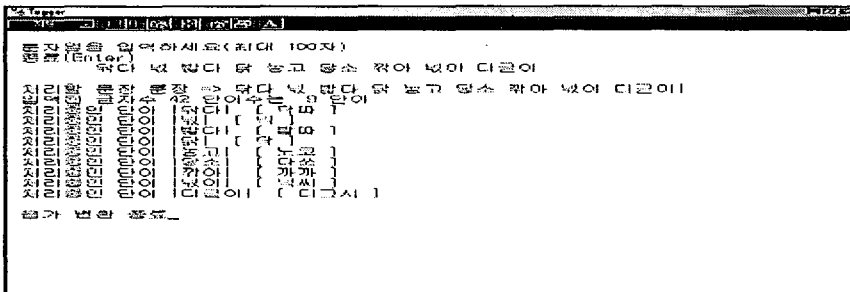
12항 등은 중복 정의된 것으로 발견되었다. 이와 같은 각 항간의 상충과 중복 문제는 각 항에 대해 우선순위를 부여하여 그 정의에 대한 수정이 필요한데, 코멘트 부분의 '(X)'는 제거되는 항을 의미한다.

[표 3]은 [그림 4]의 일관성 검사 결과를 바탕으로 음가 변환 규칙 테이블을 수정하고, 포함 관계에 의하여 복사해야 하는 경우에 대한 모든 처리를 마친 '음가 변환 테이블'의 일부를 보인 것이다.

음가 변환 시스템의 구현 환경은 변환이 완료된 음가를 표기하기 위하여, 통합형 한글을 지원하는 Windows98 시스템 상에서 Microsoft Visual C++ 컴파일러 버전 5.0으로 구현하였으며, [그림 5]는 음가 변환기의 출력 화면을 보인 것이다. 음가 변환기는 입력되는 문자열을 각 어절(또는 단어)별로 분리하여 어절별 음가 변환 결과를 출력한다.

중성/초성		ㄱ	ㅋ	ㄴ	ㄷ	ㄸ	ㄹ	ㄱ	ㅂ	ㅃ	...
ㄱ	초성	ㄱ	ㅋ	ㄴ	ㄷ	ㄸ	ㄹ	ㄱ	ㅂ	ㅃ	
	중성	ㄱ	ㄱ	ㅇ	ㄱ	ㄱ	ㅇ	ㅇ	ㄱ	ㄱ	
ㅋ	초성	ㄱ	ㅋ	ㄴ	ㄷ	ㄸ	ㄹ	ㄱ	ㅂ	ㅃ	
	중성	ㄱ	ㄱ	ㅇ	ㄱ	ㄱ	ㄱ	ㅇ	ㄱ	ㄱ	
ㄴ	초성	ㄱ	ㅋ	ㄴ	ㄷ	ㄸ	ㄹ	ㄱ	ㅂ	ㅃ	
	중성	ㄴ	ㄴ	ㄴ	ㄴ	ㄴ	ㄹ	ㄴ	ㄴ	ㄴ	
ㄷ	초성	ㄱ	ㅋ	ㄴ	ㄷ	ㄸ	ㄹ	ㄱ	ㅂ	ㅃ	
	중성	ㄷ	ㄷ	ㄴ	ㄷ	ㄷ	ㄷ	ㄴ	ㄷ	ㄷ	

[표 3] 통합 처리된 음가 변환 테이블의 일부



[그림 5] 음가 변환기의 출력 화면

4. 문자 변환 테이블 생성과 실험

4.1 ‘한국어 음가 - 한국어 표기’ 변환 규칙의 유도 방안

소리를 문자로 옮겨 적는 작업은 간략 전사(board transcription)와 정밀(narrow transcription) 전사로 나뉜다. 정밀 전사가 세밀한 음성적 차이나 소리의 높낮이, 길이, 쉼 등에 관심을 갖는다면, 간략 전사는 인식된 소리 정보를 중심으로 음운적 변화에 관심을 갖는다. 따라서 본 연구가 음운변화에 중점을 둔 소리와 문자간의 관계 규명에 있으므로 간략 전사에 해당하며, 한국어 음가(소리)를 한국어 표기(문자)로 변환하기 위한 입력 정보로 이미 생성된 [표 3]을 이용하여 전사를 행한 후, 이에 대한 후처리 과정을 통하여 한국어 표기 후보(이하 표기라고 함)를 추출한다.

[표 3]의 행과 열은 입력된 표기의 앞 음절 중성과 다음 음절의 초성을 나타내므로

일반항 테이블 [i, j]는 앞 음절 종성이 i이고 다음 음절 초성이 j이면 각각의 초성과 종성이 표에 적시된 바와 같은 음가로 각각 변화함을 나타낸다. 예를 들어, 일반항 테이블 [ㄱ, ㄱ]은 음가 [ㄱ, ㄱ]으로 출력되는데 이는 앞 음절 종성이 ‘ㄱ’이고 다음 음절 초성이 ‘ㄱ’이면 다음 음절 초성이 ‘ㄱ’으로, 앞 음절 종성은 ‘ㄱ’으로 발음(‘학교’ -> ‘학꾜’)됨을 의미한다.

따라서 이를 역으로 적용하면 인식된 한국어 음가의 앞 음절 종성이 ‘ㄱ’이고 다음 음절 초성이 ‘ㄱ’이라면 한국어 표기는 앞 음절 종성이 ‘ㄱ’이고 다음 음절 초성이 ‘ㄱ’이 될 수도 있다는 뜻이다. 그런데 [ㄱ, ㄱ]으로 출력되는 음가 항목이 유일하지 않고 입력항의 [ㄱ, ㄱ]항, [ㄱ, ㄱ]항, [ㄱ, ㄱ]항, [ㄱ, ㄱ]항 등도 역시 출력항의 음가 값이 [ㄱ, ㄱ]이 되므로, 이 음가를 생성한 입력 표기는 앞 음절 종성과 다음 음절 초성이 [ㄱ, ㄱ], [ㄱ, ㄱ], [ㄱ, ㄱ], [ㄱ, ㄱ], [ㄱ, ㄱ] ... 등이었을 수 있다는 것이다. 즉, 인식된 음가가 [ㄱ, ㄱ]이라면 이에 대한 표기는 다음에 나열된 논리식의 경우 중 한 가지가 된다는 뜻이다.

$$[ㄱ, ㄱ] \rightarrow [ㄱ\wedge\wedge]V[ㄱ\wedge\wedge]V[ㄱ\wedge\wedge]V[ㄱ\wedge\wedge]V[ㄱ\wedge\wedge] \\ V[ㄱ\wedge\wedge]V[ㄱ\wedge\wedge]V[ㄱ\wedge\wedge]V[ㄱ\wedge\wedge]V[ㄱ\wedge\wedge]$$

여기에서 ‘→’은 ‘좌변이면 우변이다’를 의미하며, 좌변의 [ㄱ, ㄱ]은 앞 음절 종성의 음가가 ‘ㄱ’이고 다음 음절의 초성 음가가 ‘ㄱ’임을 나타낸다. 우변의 ‘V’는 ‘혹은’으로 해석하고, 대괄호는 표기 가능한 초성/종성 중의 한 가지로써 대괄호 내 ‘\wedge’의 앞은 앞 음절 종성 표기이고 뒤는 다음 음절의 초성 표기이다. 인식된 음가가 ‘학꾜’인 경우, ‘학’의 종성이 ‘ㄱ’이고 다음 글자인 ‘꾜’의 초성이 ‘ㄱ’이므로 위 식의 조건부인 좌변에 해당한다. 따라서 우변의 결론부를 적용하면 음가인 ‘학꾜’에 대한 표기는 ‘학교, 학교, 학꾜, 학꾜, 학교, 학교, 학꾜, 학꾜, 학교, 혹은 학교’ 중 한 가지임을 알 수 있다.

이들 가능한 표기들 중 단어 사전에 출현하는 단어를 적당한 표기라고 한다면, 경우에 따라서는 적당한 표기가 없을 수도 있으며 혹은 여러 개가 존재할 수도 있다. 적당한 표기가 없는 경우는 인식에 오류가 있었거나 사전이 불완전한 경우인데, 오인식의 경우는 인식 단계에서 해결해야 할 문제이고 사전이 불완전한 경우는 단어사전을 확충함으로써 문제를 해결할 수 있다. 적당한 표기가 다수일 경우가 문제인데, 이 때에는 주위의 문맥을 분석하는 후처리 단계를 거쳐 해결하여야 한다.

[표 3]은 한글 맞춤법의 표준발음 규칙을 망라한 것이다. 따라서 종성과 초성의 발음 음가는 [표 3]의 일반항으로 출현하는 경우 이외에는 발생하지 않는다. 그러므로 [표 3]의 항목으로 출현하는 모든 경우의 각각에 대하여 위의 [ㄱ, ㄱ]의 경우와 같은 방법으로 해당 표기를 모두 찾으면 ‘한국어 음가 - 한국어 표기’ 변환 규칙이 완성될 수 있다. [표 3]에 보인 음가 변환 테이블로부터 ‘한국어 음가 - 한국어 표기’ 변환 규칙을 역으로 변환하는 알고리즘은 [그림 6]과 같다.

이 역변환 알고리즘은 통합된 표준발음법 행렬 파일을 로드한 다음, 음가 생성 규칙에 정의된 트랜지션 번호와 배열 m의 항 번호가 일치하면 이를 배열에 저장하고, 종성과 초성의 항 번호를 이름으로 하는 배열 m이 모두 처리될 때까지 처리하게 된다.

```

입력 : 통합된 표준발음법의 행렬 파일
출력 : 초성과 중성 번호를 테이블 번호로 하는 표기법 행렬 파일
BEGIN
  단계 1 : While 중성과 초성의 번호를 이름으로 하는 표기 유도 규칙 정의
            테이블이 남아 있는 동안
  단계 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
  단계 3 : IF m의 원소가 없으면
            THEN   발생할 수 없는 경우라고 표시
            ELSE   다음 표기 변환 규칙 테이블 m을 정의하고, 단계 2로
                  이동
  단계 4 : While m이 Empty가 될 때까지
            m에 저장된 결과를 중성과 초성을 테이블 번호로 하여 출력
END

```

[그림 6] 음가 변환 테이블의 역변환 알고리즘

단계 3에서는 음가생성 규칙에서 표기가 검색되지 않는 경우에 대해서 이를 발생할 수 없는 경우라고 정의하고, 중성과 초성 번호를 교대로 증가시켜 가면서 단계 2를 m의 정의가 끝날 때까지 수행케 한다.

m의 개수는 한국어 음가에 대해서 중성으로 올 수 있는 경우 28개와 초성으로 올 수 있는 경우 18개에 대해서 정의되어 있다. 따라서 총 504 개의 변환 규칙 테이블이 만들어진다. [그림 7]은 이와 같은 방법으로 완성한 변환 규칙의 일부 예와 2차원 배열로 정의한 규칙 테이블에서 입력되어진 표기 ‘중성’과 ‘초성’에 대하여 한국어 음가로 유도되어 질 수 있는 ‘초성’과 ‘중성’을 정의한 것이다.

4.2 한국어 음가를 한글 표기로 변환하는 생성기의 구현

한국어 음가를 한국어 표기로 변환하는 테이블의 생성을 위하여 문자 변환 생성기를 구현하였다. 구현 환경은 변환이 완료된 표기의 출력을 위하여 통합형 한글을 지원하는 Windows2000 시스템 상에서 Microsoft Visual C++ 컴파일러 버전 5.0을 사용하였다. 구현된 ‘한국어 음가 - 한국어 표기’ 변환 생성 시스템은 음성 인식기를 통하여 인식된 음가를 입력으로 하여 유도될 수 있는 모든 표기 집합을 생성하는 시스템을 의미한다. 따라서 생성되는 표기 집합은 변환 규칙을 참고하여 한글 문자로 생성될 수 있는 가능한 모든 것이 포함되므로, 표준 한국어 표기에 맞지 않는 것들도 유도되어질 수 있다.

<‘한국어 음가 - 한국어 표기’ 변환 규칙의 일부>

[ㄱ,목음]→[ㄱ ㅏ모음어미]V[ㄱ ㅏ모음조사]V[ㄱ ㅏ모음접미사]V[ㄱ ㅏ모음실질형태소]V[ㄱ ㅏ모음실질형태소]V[ㄱ ㅏ모음실질형태소]V[ㄱ ㅏ모음실질형태소]V[ㄱ ㅏ모음실질형태소]V[목음 ㅏ ㄱ].

[ㄴ,목음]→[ㄴ ㅏ모음어미]V[ㄴ ㅏ모음조사]V[ㄴ ㅏ모음접미사]V[ㄴ ㅏ모음실질형태소]V[ㄴ ㅏ모음어미]V[ㄴ ㅏ모음조사]V[ㄴ ㅏ모음접미사]V[ㄴ ㅏ모음실질형태소]V[목음 ㅏ ㄴ].

[ㄷ,목음]→[ㄷ ㅏ모음어미]V[ㄷ ㅏ모음조사]V[ㄷ ㅏ모음접미사]V[ㄷ ㅏ모음실질형태소]V[ㄷ ㅏ모음실질형태소]V[ㅌ ㅏ모음실질형태소]V[ㅌ ㅏ모음실질형태소]V[ㅌ ㅏ모음실질형태소]V[ㅌ ㅏ모음실질형태소]V[ㅌ ㅏ모음실질형태소]V[ㅌ ㅏ모음실질형태소]V[목음 ㅏ ㄷ].

: {중략}

[ㅅ,목음]→[ㅅ ㅏ모음어미]V[ㅅ ㅏ모음조사]V[ㅅ ㅏ모음접미사]V[ㅅ ㅏ이조사]V[ㅅ ㅏ이접미사]V[ㅅ ㅏ모음실질형태소]V[목음 ㅏ ㅅ].

[ㅆ,목음]→[ㅆ ㅏ모음어미]V[ㅆ ㅏ모음조사]V[ㅆ ㅏ모음접미사]V[ㅆ ㅏ이조사]V[ㅆ ㅏ이접미사]V[목음 ㅏ ㅆ].

[ㅇ,목음]→[ㅎ ㅏ모음어미]V[ㅎ ㅏ모음조사]V[ㅎ ㅏ모음접미사]V[ㅎ ㅏ이조사]V[ㅎ ㅏ이접미사].

: {이하 생략}

<‘한국어 음가 - 한국어 표기’ 변환 규칙 테이블의 일부>

Phonetic Table[42][58]={

/* (0) (1) ㄱ(2) ㄴ(3) ㄷ(4) ㅌ(5) ㅍ(6)

/* 받침없음(1)*/{ {0,0},{0,0},{0,0},{0,0},{0,0},{0,0},{0,0}, ...

/* ㄱ(2)*/{ {0,0},{0,0},{2,3},{2,3},{23,4},{2,6},{2,6},

/* ㄴ(3)*/{ {0,0},{0,0},{2,3},{2,3},{23,4},{2,6},{2,6},

/* ㄷ(4)*/{ {0,0},{0,0},{2,3},{2,3},{23,4},{2,6},{2,6},

/* ㄴ(5)*/{ {0,0},{0,0},{5,2},{5,3},{5,4},{5,5},{5,6},

/* ㅌ(6)*/{ {0,0},{0,0},{5,2},{5,3},{5,4},{5,5},{5,6},

/* ㅍ(7)*/{ {0,0},{0,0},{5,17},{5,3},{5,4},{5,18},{5,6},

:

[그림 7] 한국어 음가 - 한국어 표기 변환 규칙의 일부

[그림 8]은 입력 음소의 트랜지션 번호와 변환 규칙(그림 7)의 테이블 번호를 비교하면서 번호가 같은 테이블의 표기 요인을 모두 출력하는 알고리즘을 보인 것이다. 단계 1에서 변환 규칙을 로드한 다음, 단계 2에서는 입력된 음가의 자모 번호와 테이블의 항 번호를 비교하여 항 번호가 같은 테이블의 트랜지션을 배열에 저장하고, 단계 4에서 배열에 저장된 트랜지션을 표기 중성, 초성의 항 번호를 참조하여 출력하게 된

```

입력 : 자모가 분리된 음소(자음)
출력 : 표기 집합
정수형 jong, cho;
BEGIN
  단계 1 : 테이블 로딩(음가-문자 변환 규칙 m)
  단계 2 : WHILE 입력된 음소가 Null일 때까지
            IF 분리된 음소 번호(jong, cho)를 m의 테이블 번호와 비교
            THEN 테이블에서 표기요인 번호(jong, cho)를 배열에 저장
  단계 3 : 입력된 문장이 끝날 때까지 단계 1과 단계 2를 반복하여 처리
  단계 4 : 배열에 저장된 항 번호와 초성과 중성 번호를 비교하여
            한글 표기로 변환하여 출력
END

```

[그림 8] 표기 집합 생성 알고리즘

다. 단계 2의 배열에 저장되는 항 번호는 초성과 중성의 번호이며, 프로그램 상에서 전달되는 파라메타들은 항 번호를 참조하여 출력될 때에만 표기로 변환하여 한글 문자로 출력한다.

```

중성 ㄱ, 초성 ㄲ
( ㄱ, ㄱ), ( ㄱ, ㄲ), ( ㄱ, ㄱ어미), ( ㄱ, ㄱ접미사), ( ㄲ, ㄱ), ( ㄲ, ㄲ), ( ㄲ, ㄱ어미),
( ㄲ, ㄱ접미사), ( ㄴ, ㄱ), ( ㄴ, ㄲ), ( ㄴ, ㄱ어미), ( ㄴ, ㄱ접미사), ( ㄷ, ㄱ), ( ㄷ, ㄲ),
( ㄷ, ㄱ어미), ( ㄷ, ㄱ접미사), ( ㅋ, ㄱ), ( ㅋ, ㄲ), ( ㅋ, ㄱ어미), ( ㅋ, ㄱ접미사),
(용언어간말음ㄷ, ㄲ), .....

```

[표 4] 음가 중성 'ㄱ', 초성 'ㄲ' 에 대한 표기 변환 예

이와 같은 과정을 통하여 생성된 표기 집합들 중에서, 표준어 표기법에 맞는 정확한 표준 표기의 추출을 위해서는 비표기 문자 및 비어휘 어절, 중복 유도 어절 등을 제거하는 과정이 필요하다. 즉, 발생할 수 있는 표기 집합들 중 '합' 같이 일상생활에 사용되지 않는 문자(또는 표준어 표기에 맞지 않는 문자)가 포함된 표기를 제거하고, 중복 생성된 표기들도 제거하여야 한다.

이와 같은 비표기 문자들은 표준 표기 집합이라는 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 표기 집합에서 제거함으로써 사전을 참조할 최종 표기 집합의 수를 감소시킨다.

즉, [표 4]의 학교(중성 ㄱ, 초성 ㄲ)의 예에서 보이는 것과 같이 표기 집합에서 표기들이 중복되어 유도되는 것을 볼 수 있는데, 이는 중성 'ㄱ'과 초성 'ㄲ'에 대한 유도규칙 (ㄱ, ㄱ), (ㄱ, ㄱ어미), (ㄱ, ㄱ접미사)에서 표기 중성 'ㄱ', 초성 'ㄲ'이 중복되어

유도되기 때문이다. [그림 9]는 중복 유도되어진 어절을 제거하는 알고리즘을 보인 것이다.

```

입력 : 표기 생성 규칙으로부터 유도된 배열
출력 : 중복을 제거한 표기 집합
정수형 i, j = 0;
BEGIN
  단계 1 : 표기 집합을 읽어 들임
  단계 2 : FOR i=0 TO n(표기 집합의 개수)
            i번째 단어를 읽어들이 표기집합과 비교.
            FOR j=1 TO n(표기 집합의 개수)
              j번째 단어를 읽어들이 표기집합과 비교
              IF i번째 단어와 j번째 단어가 같으면
                THEN j번째 단어를 제거.
                s = n - 1
                ELSE j 값의 증가
  단계 3 : n = s 로 하고 단계 2로 이동
  단계 4 : 중복을 제거한 표기집합 출력
END

```

[그림 9] 중복 제거 알고리즘

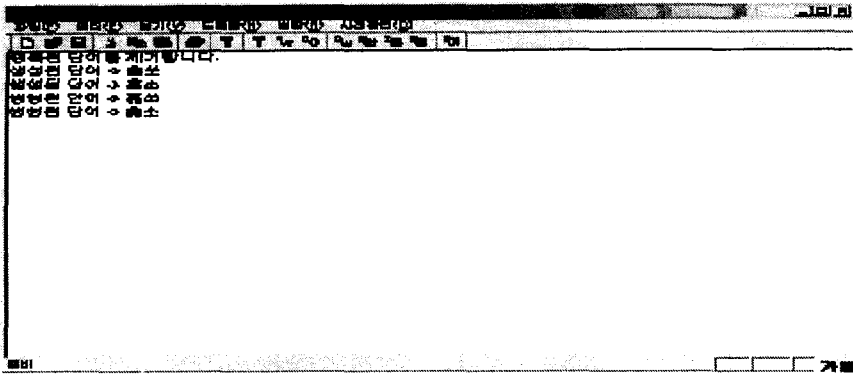
4.3 한글 표기변환 생성기의 실험 화면과 결과

[그림 10]은 구현된 시스템의 음가 입력용 초기 화면이며, [그림 11] 이후의 화면 상단에 있는 도구모음 상자의 기능은 다음과 같다.

- T : 변환할 음가 문자열 입력하기
- D^W : 입력된 음가 문장을 어절(또는 단어) 단위로 분리된 결과 보기
- D^C : 분리된 어절에 대해서 음절 분리와 자모 분리를 행한 결과 보기
- G^W : 입력 음가로부터 생성될 수 있는 한글 표기후보 보기(그림 11)
- D^한 : 비완성형 문자를 포함한 어절을 제거한 결과 보기(그림 12)
- D^문 : 문법 형태소 오류를 포함하고 있는 비어휘 어절을 제거하는 과정
- D^중 : 중복 생성된 어절을 제거한 후의 표기 후보 보기(그림 13)

제 4장을 통하여 ‘한국어 음가 - 한국어 표기’ 변환 생성기에 사용될 변환 규칙의 생성과정을 보였으며, 생성된 변환 규칙의 타당성을 보이기 위하여 표준어 규정의 표준발음법에 나타난 모든 항목에 대하여 다양한 음가를 입력하고 출력 결과를 분석하는 실험을 실시하였다.

이 실험을 위하여 사용된 음가 데이터의 도메인은 한글 맞춤법 제 2부의 표준발음법에 제시된 제 4장(8-16항), 5장(17-22항), 6장(23-28항)의 받침의 발음, 음의 동화, 경음화에 대한 예시어 224개를 대상으로 하였다. 그리고 중복 어절(또는 단어)이 제



[그림 13] 중복 어절 제거후의 최종 후보 표기 화면(홀쏘)

항 번호		9	10	11	12	13	14	15	16	17	18
입력 음가(어절)수		7	5	9	33	10	9	9	6	9	29
표기 가능 후보수	최초 생성 후보	171	87	274	1,354	518	341	1,297	585	781	1,291
	비표기 제거후	83	53	91	366	92	112	157	62	128	320
	중복표기 제거후	36	23	40	185	37	39	115	18	70	285
	최초 생성 후보 대비 최종후보율(%)	21.1	26.4	14.6	13.7	7.1	11.4	8.9	3.1	9.0	22.0
입력 음가당 평균 최종 후보수		5.0	4.6	4.4	5.6	3.7	4.3	12.8	3.0	7.8	9.8
항 번호		19	20	21	22	23	24	25	26	27	28
입력 음가(어절)수		9	12	7	생략	20	8	4	11	8	19
표기 가능 후보수	최초 생성 후보	153	519	251		1,428	203	96	311	1,016	1,179
	비표기 제거후	68	152	77		305	76	51	103	165	313
	중복표기 제거후	56	68	37		203	38	16	28	81	80
	최초 생성 후보 대비 최종후보율(%)	36.6	13.1	14.8		14.2	18.9	16.7	9.0	8.0	6.8
입력 음가당 평균 최종 후보수		6.2	5.7	5.3		10.2	4.8	4.0	2.5	10.1	4.2

[표 5] 실험 항에 대한 최종 표기 후보 추출 결과 요약

표준발음법에 언급된 224개의 실험 대상 음가에 대한 실험 결과를 요약하면, 최초 생성 어절수는 전체 11,855개였으며, 비완성형 코드를 포함한 어절을 비롯한 비어휘 어절을 제거한 후의 어절수는 총 2,774개였다. 그리고 중복 단어를 제거한 후의 최종 표기 후보수는 1,455개였다. 따라서 하나의 음가에 대한 평균 후보 단어는 6.5개 정도였으며, 일반적인 단어 사전과의 매칭만으로도 후보 단어의 수는 더욱 줄어들 수 있음을 알 수 있다.

장	항	실험 결과		
		입력 음가	실제 표기	중복 어절 제거후의 후보 단어
4	9	닥따	닥다	닥따 닥다 닥다 닥따 닥다 닥따
		일따	있다	일따 있다 잇따 있다 잇따 있다 잇따
		빋따	빋다	빋따 빋다 빋따 빋다 빋따 빋다 빋따
		쫘따	쫘다	쫘따 쫘다 쫘따 쫘다 쫘따
5	19	담녁	담력	담녁 담력 담녁 담력 담녁 담력 담력
		침낙	침락	침낙 침락 침낙 침락
		강능	강릉	강능 강릉 각능 각릉 값능 값릉
		항노	항로	항노 항로 학노 학로
	불	협녁	협력	협녁 협력 협녁 협력
입	심니	십리	심니 심리 십니 십리 싶니 싶리	
6	25	넙게	넙게	넙게 넙게 넙게 넙게
		할따	할다	할따 할다 할따 할다
		홀쏘	홀소	홀쏘 홀소 홀쏘 홀소
		뽕찌	뽕지	뽕찌 뽕지 뽕찌 뽕지

[표 6] 실험 결과의 일부

[표 6]은 표준발음법 각 항목에 예시된 음가들을 입력하여 본 시스템이 출력한 일부 결과를 보인 것이다. 실험 데이터로는 표준발음법의 모든 항목이 반영되었으며, 출력 결과에는 찾고자 하는 단어가 반드시 포함되어 있음을 확인하였다. 따라서 본 논문이 제시하는 ‘한국어 음가 - 한국어 표기’ 변환 규칙 생성에 관한 이론의 타당성을 입증할 수 있다.

5. 뱃음말 및 향후 과제

본 논문에서는 한글맞춤법 제 2부에 제시된 ‘표준발음법’을 전산화하여 한국어 음성합성과 음성인식의 전처리 및 후처리 단계에서 사용할 수 있는 방안을 제시하였다. 본 연구에서는 표준발음법의 각 항에 대해서 각각의 페트리 넷 모델을 작성한 다음, 이들을 모두 통합하는 전략을 사용하였으며, 통합된 ‘음가 변환 테이블’을 통하여 입력된 한국어 표기(한글)에 대한 표준 음가를 1회의 검색으로 찾는 방안을 제시하였다.

또한, 표준발음법 각 항에 대한 페트리 넷 근접행렬을 하나의 테이블로 통합하는 과정에서 표준발음법에 내재한 중복과 상충의 문제점을 해결하여 표준발음법의 일관성을 검증하고자 하였다. 이러한 과정을 통하여 한국어 표기 문자에 대한 문법적으로 올바른 발음 음가를 찾을 수 있는 ‘음가 변환 테이블’을 생성하였고, ‘음가 변환 테이블’의 무결성을 확인하기 위한 음가 변환 시스템을 구현하여 다양한 데이터로 실험하였다.

다음으로 ‘음가 변환 테이블’을 이용하여, 입력된 발음 음가에 대한 올바른 한국어 표기 문자를 찾을 수 있는 ‘한국어 음가 - 한국어 표기’ 변환 테이블의 생성에 대한 연구 방법도 제시하였다. 이 변환 테이블의 생성을 위하여, ‘음가 변환 테이블’의 모든 엔트리 각각에 대해 그 엔트리를 지정하는 인덱스들을 찾아 매핑한 다음, 이들을 모두 통합하는 방법을 적용하였다.

본 연구에서 제시된 한국어 음가 변환기는 기존 연구들에서 사용하던, 더 이상의 음운 변동이 없을 때까지 음운변화 규칙을 반복하여 적용하던 방식이 아닌, 한번의 음가 변환 테이블 탐색으로 원하는 음가를 추출할 수 있으므로 더 효율적이라는 것을 보였다. 그리고 본 논문에서 제안한 ‘한국어 음가 - 한국어 표기’ 변환 테이블을 이용하여 발음 음가를 입력으로 받아 문법적으로 올바른 표기 문자를 생성하는 문자 생성 시스템을 구현하였고, 다양한 데이터로 실험한 결과 본 논문이 제공하는 ‘한국어 음가 - 한국어 표기’ 변환 테이블이 완전하고 무결함을 알 수 있었다.

본 논문의 목적은 한국어 음성합성기의 전처리 역할을 할 수 있는 ‘음가 변환 테이블’의 생성과 한국어 음성인식기의 후처리 역할을 할 수 있는 ‘한국어 음가 - 한국어 표기’ 변환 테이블의 생성에 있다. 생성된 테이블들의 인덱스 일부에는 문법 형태소 정보가 사용되기 때문에 정확한 형태소 분석 결과를 생성하는 형태소 분석기와의 결합이 추가로 요구되며, 또한 구문 분석의 결과에 따라 끊어 읽을 어절이 달라지고 발음도 달라지므로 구문 분석기에 대한 연구도 요구된다. 그리고 생성된 2개 테이블과의 결합을 위한 음성인식/합성기 연구 분야, 정확한 표준 표기 후보 생성을 위한 의미 분석기에 대한 연구 분야 및 표준사전 생성 분야와의 협동 연구가 앞으로 요구된다.

<참고문헌>

- 김재홍. 1998. 고품질 한국어 음성합성을 위한 문서음성 변환시스템. 연세대학교, 석사학위 논문.
- 김혜순 외. 1993. 멀티미디어를 이용한 한국어 발음교육 시스템. 한국정보과학회 논문지 20.1.
- 김희린. 1991. 음성신호의 부분정보를 이용한 음성인식 성능 향상, 한국과학기술원. 박사학위논문.
- 남광우. 1989. 한국어의 발음연구[1], 일조각.
- 문화교육부. 1988. 표준어 규정. 문교부 고시 제 88-2호.
- 신지영 외. 2003. 우리말 소리의 체계, 한국문화사.

- 양진석 외. 1996. 운율 및 길이 정보를 이용한 무제한 음성합성기의 설계 및 구현. 한국정보처리학회 논문지 3.5.
- 이계영 외. 2001. 한국어 음성합성을 위한 음가변환 테이블 생성. 대한전자공학회 논문지, 제32권 CI편 제5호, pp. 44-57.
- 이기문 외. 2000. 국어 음운론, 학연사.
- 한국과학기술원. 1980. 무제한 한국어 음성합성 시스템. 연구보고서.
- 허용. 1982. 국어 음운학, 정음사.
- Klatt, D. 1977. Review of the ARPA Speech Understanding Project. *J. Acoust. Soc. Am.* 62.6.
- Davis, K.H., R. Davis and S. Balashek. 1952. Automatic Recognition of Spoken Digits. *J. Acoust. Soc. Am.* 24.6.
- Kita, K., Kawabata and H. Saito. 1989. HMM Continuous Speech Recognition Using Predictive LR Parsing. *Proc. of the IEEE Int. Conference on Acoustics, Speech and Signal Processing 2*, Glasgow, Scotland, pp. 703-706.
- Lippmann, R. 1997. Speech Recognition by Human and Machine. *Speech Communications* 22, pp. 1-14.
- Murata, T. 1989. Pertri Nets: Properties, Analysis and Applications. *Proceeding of the IEEE* 77.4, pp. 541-580, April 1989.
- Lea, W., ed. 1980. *Trends in Speech Recognition*. Prentice Hall.

접수일자: 2003년 5월 20일

게재결정: 2003년 10월 8일