

論文2001-38CI-5-5

한국어 음성 합성을 위한 음가 변환 테이블 생성

(Formation of A Phonetic-Value Look-up Table for Korean Voice Synthesis)

李桂英*, 任在傑*

(Gye-Young Lee and Jae-Geol Yim)

요약

문법적으로 정확한 한국어 음성을 합성하려면 표준어 규정의 '표준 발음법'을 준용해야 한다. 따라서 한국어 음가 합성 시스템에 사용되는 한글을 음성으로 변환하여 주는 규칙은 '표준 발음법'을 완전하게 반영하며 또한 무결해야 한다. 기존의 연구에서는 표준 발음법을 검증없이 적용하여 왔고, 표준 발음법 자체에 모순이 있는가의 여부에 대해서도 체계적인 분석을 위한 시도가 전무하였다. 본 논문에서는 한국어 음가 생성의 기본 규칙으로 준용할 표준 발음법을 페트리 넷으로 모델링하여 표준 발음법의 일관성을 검증하였다. 그리고, 음운 변동 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 음운 변동이 완료되는 한국어 음성 합성을 위한 음가 변환 테이블을 구현하였다.

Abstract

In order to synthesize grammatically correct Korean voices, we have to refer to the "Standard Pronunciation Rules(SPR)" stated in the "Standard Grammar of Korean Language." Therefore, the rules that is used for a Korean-voice-synthesis system to find Korean voices corresponding to a given Korean sentence must completely reflect the SPR and must be sound. However, in the field of computer science they have just used the SPR without proving the completeness and soundness of their rules. In this paper, we construct a Petri net model for each rule of SPR, integrate all the Petri net models to build one big Petri net completely representing SPR, and analyse the Petri net to prove the consistency of it. Then, we transfer the Petri net model into a look-up table for Korean voice. Using this table, we can avoid the drawbacks of existing approaches such as going through several stages or repetitively applying a converting process.

1. 서론

1.1 연구 배경 및 필요성

표기 언어인 한글을 한국어 음성으로 변환하려면 "표준어 규정(1988년, 문교부 고시 제 88-2호)" 중, 제 2부의 "표준 발음법^[1]"을 준용해야 한다. 따라서, 한국어 음가를 생성하는 컴퓨터 시스템에 사용되는 발음법은 '표준 발음법'을 완전히 표현하며 또한 무결해야 한다.

* 正會員, 東國大學校, 컴퓨터情報通信學部

(Dongguk University Dept. of Computer Science)

※ 본 연구는 한국과학재단 목적기초연구(2001-1-303 00-010-2) 지원으로 수행되었음.

接受日字:2000年3月30日, 수정완료일:2001年6月11日

본 논문에서는 표준 발음법의 모든 항을 각각 페트리 넷[2-3]으로 모델링한 다음 이들을 통합함으로써 표준 발음법을 완전히 표현하는 한 개의 커다란 페트리 넷을 생성하고, 이 페트리 넷의 일관성을 분석한다.

그리고, 일관성 분석의 결과로 나타나는 중복항에 대해서는 두 개의 항 중 하나를 삭제하고, 선택항은 원칙을 취하고 허용항을 삭제하며, 상충항은 적용의 우선순위를 부여하거나 잘못된 정의를 삭제하여 한국어 음가 생성을 위한 표준 발음법으로 수정해 나간다.

이와 같이 표준 발음법의 일관성이 검증되어 무결한 표준 발음법이 정의된 후에는 이를 바탕으로 음가 변환 테이블을 생성하게 되며, 이 테이블은 음가 생성을 위하여 입력된 한글 단어나 문장의 자모와 형태소 정보만으로 한국어 표준 발음을 쉽게 추출할 수 있게 된다. 이러한 방법은 표기 언어인 한글을 음성 언어인 한국어 음가로 변환할 때 발생할 수 있는 개인간의 해석 차를 없애고, 형태소 정보와 자모만으로 정확한 음가 변환이 이루어질 수 있도록 도와 준다.

한국어 음성 합성 시스템의 음가 변환 과정에서 검증되지 않은 임의의 순서로 한 단계씩 음가 변환을 시켜나가는 기존의 연구로는 음가 생성의 무결성을 입증할 수 없으며, 임의의 다단계로 처리하는 과정상에서도 비효율성이 발생할 수 있다.

따라서, 검증된 표준 발음법을 적용하여 구현되는 본 논문의 한국어 음가 변환 테이블은 1회의 테이블 검색으로 모든 한국어 음가 변환이 완료되므로, 검증되지 않은 임의의 순서로 한 단계씩 차례로 처리해 나갈 때 발생할 수 있는 음가 변환의 오류와 변환 시간의 문제 등을 극복할 수 있다.

1.2 연구 범위

음성 합성 시스템은 일반적으로(그림 1)의 구조를 갖는다. 음가 변환기는 전처리기의 결과와 형태소 분석기^[4]의 분석 결과, 구문 분석기^[5]의 분석 결과를 입력으로 갖는다.

(그림 1)의 형태소 분석기와 구문 분석기의 구현은 본 연구에서 제외하며, 본 연구에서는 정확한 형태소 분석 결과와 구문 분석 결과에 따라 음가 변환이 정확히 수행되는 한국어 음가 변환 테이블을 구현하는데 중점을 둔다.

그리고, 본 연구에서 생성하는 한국어 음가 변환 테이블은 음가 변환기에서 표기 언어를 음성 언어로 바

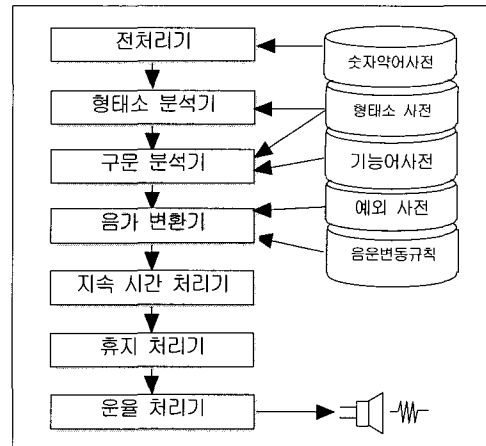


그림 1. 일반적인 음성 합성 시스템의 구조
Fig. 1. A general struture of a voice synthesis system.

꾸어 주는 규칙(음운변동 규칙)에 해당하는 것으로 입력된 한글의 형태소 분석 결과와 구문 분석 결과에 따라서 동일한 문자 표기일지라도 다른 음가로 변환시킬 수 있게 된다.

본 논문은 서론에 이어 2장에서 한국어 음가 생성과 관련된 기존 연구에 대해서 살펴보고 3장에서 표준 발음법의 페트리 넷 모델링에 대하여 기술하며, 4장에서는 표준 발음법의 일관성 검증과 각 항의 통합에 대하여 기술한다. 5장에서는 통합 테이블의 압축과 음가 변환 테이블의 생성에 대하여 설명하고, 6장에서는 각 모듈의 구현과 음가 변환기의 구현 및 각 모듈별 실험 결과와 음가 변환기의 실험 결과를 기술한다.

II. 한국어 음가 생성과 관련된 기존 연구

[6]에서는 한국어 음운 변동 과정을 구개음화, 자음 축약, 연음 규칙, 자음 동화, 경음화, 끝소리 규칙의 순서로 적용하여 음운 변동을 처리하고 있다. [7]에서는 한국어 음운 변동에 관한 지식을 지식 베이스로 구성하여 전진 추론 방식으로 탐색해 나가며 어절의 변화가 있는 동안에는 새로운 상태를 만들고, 이 같은 과정을 반복하여 음운변화를 일으키게 하였다.

[8]에서는 모음 변환, ㄴ첨가, 경음화, ㅎ탈락, 절음 법칙, 연음 법칙, 구개음화, 대표음, 자음 접변 등의 순서를 반복하여 적용시켜 음운 변동을 일으키며, [9]에서는 음운 규칙의 적용 순서를 ㄴ첨가, 구개 음화, 음운 축약, ㅎ변환, 자음 동화, 경음화, 말음법 및 연음 법칙

으로 정하고, 규칙 적용 시에 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.

이와 같이 기존의 연구에서는 표준 발음법에 나타나는 음운 변동 현상을 여러 단계로 나누어 차례대로 적용시키고, 이때 변동된 단어에 대하여 처음부터 다시 변환하는 작업 수행 구조를 가지고 있다. 그리고, 음운 변동을 수행하는 순서 역시, 증명된 변환 순서가 없이 시스템 구현자에 따라 임의의 순서로 음운 변동 현상을 적용시켜, 음운 변동 처리의 적용 순서가 바뀔 때 다른 음운 변동 결과의 정확성 검증이 이루어지지 않은 상태이다.

이러한 방법으로 구현된 음성 합성 시스템은 음가 변환의 무결성을 보장할 수 없으며, 이러한 방법에 의해 구성된 음성 인식 후처리 시스템 역시, 음성 인식의 후처리 결과를 보장하지 못하는 결과가 나타날 수 있다.

Ⅲ. 표준 발음법의 페트리 넷 모델링

3.1 모델링 범위

(1) 모델링 항의 설정

표준 발음법의 일관성 및 무결성을 검증하려면 표준 발음법을 수학적으로 정형화하는 단계를 반드시 선행하여야 한다. 표준 발음법은 마치 숙련가 시스템의 규칙들처럼, 조건과 결론의 형식으로 기술되어 있다. 조건과 결론의 형식으로 기술된 지식을 표현하는 정형화된 방법으로 페트리 넷^[2]이 효율적으로 쓰임으로 본 논문에서는 페트리 넷으로 표준 발음법을 정형화한다. 표준 발음법을 페트리 넷으로 모델링하기 위하여 표준 발음법에 나타나는 항목 중, 모델링에 포함되어야 할 항목과 실제 발음과 직접적인 관련이 없어 모델링에서 제외되는 항 및 그 이유를 <표 1>에 나타내었다.

표준 발음법 정의에서 원칙과 허용이 있어 두 가지 경우 중 하나를 선택해야 할 경우를 선택의 문제라 정의하고, 선택의 문제가 발생할 때에는 원칙을 따르고 허용을 버리는 방법으로 모델링 영역을 결정한다. 그 외에 실제 발음과 직접적인 관련이 없는 부분은 “직접 관련 없음(직·관·없)”으로 분류하여 모델링에서 제외하였으며, 각 항의 정의에 나타나는 예외 사항은 “예외 사전 처리(예·사·처)”로 분류한다.

표 1. 모델링에서 제외된 항과 이유

Table 1. Excluded clauses and reasons.

항번호	요 약	사 유	처 리
1	실제 발음을 따르되 전통성과 합리성 고려	직관없	제 외
2	자음은 19개로 함	직관없	제 외
3	모음 21개로 함	직관없	제 외
4	단모음 발음	직관없	제 외
5	이중모음 발음	선택	제 외
6	단어의 첫 음절에만 긴소리 나타남	직관없	제 외
7	긴소리를 가진 경우에도 짧게 발음해야 하는 경우	예외	제 외
10. 다만(2)	겹받침에 관한 규칙	예외	예·사·처
12.1 불임(2)	붙여 읽을 경우의 발음	선택	제 외
16	한글 자모의 이름	예외	예·사·처
18. 불임	두 단어를 한마디로 발음하는 경우	선택	제 외
20. 다만	다만	예외	예·사·처
21	20 이외의 자음동화 불인정	직관없	제 외
22	'어', '어' 둘 다 허용	선택	제 외
26. 다만	같은 한자가 겹쳐진 단어 된소리 안함	예외	예·사·처
27	관형사형(으)르의 붙여 읽기	선택	제 외
29. 다만 [불임2] 다만	ㄴ첨가 현상에 대한 규칙	예외	예·사·처

표 2. 모델링에서 자음 변화의 표기

Table 2. Notations for Jaum changes in modeling.

구분	변 동 : 예	종성	초성	모델링 표기
자 음	9항(받침 변화) : 첫[첫], 술[술], 녀[녀]	변함	불변	OJ
	12항(종성과 뒤음절 첫소리와 합쳐짐) : 각하[가카], 종단[조 단]	탈락	변함	OC
	12.4(종성 발음하지 않음) : 낱은[나은], 놀아[노아]	탈락	불변	OJN
	14항(뒤엿것만을 뒤음절 첫소리로 옮겨 발음) : 녀이[녀씨], 앓아[안자], 닭을[달글]	변함	변함	OJC →OJ, OC

외 사항은 “예외 사전 처리(예·사·처)”로 분류한다. 근래에는 음의 길이에 대한 구별 의식이 우리말에서 점점 사라져 가는 추세^[10]이므로 소리의 길이 역시 모델링에서 제외하였고, <표 1>의 표기 중 10항 ‘다만’의 (2)번 규정은 “10.다만(2)”의 방법으로 나타내었다. 다른 표기 없이 항 번호만 있는 것은 항 전체가 모델링에서 제외되는 항을 의미한다.

(2) 음가 변화의 모델링

표준 발음법에서 자음이 변하는 경우는 ①중성만 변하는 경우(OJ), ②중성이 탈락하고 초성이 변하는 경우(OC), ③중성만 탈락되는 경우(OJN) ④초성과 중성이 둘 다 변하는 경우(OJC)의 4가지로 나뉜다.

이것을 중성과 초성이 변하는 부분에 따라 OJ와 OC로 구분하면 모델링 표기는 OJ, OC, OJN의 3가지로 표기할 수 있으며, 이때 OJC는 OJ와 OC로 나누어 표기한다. <표 2>는 표준 발음법의 자음 변화 관련 모델링 표기를 요약한 것이다.

3.2 모델링 표기의 정의

(1) 입력 플레이스에 나타나는 표기

- IC : 입력되는 초성 IJ : 입력되는 중성
- ICV : 입력되는 초성과 중성
- IM : 입력되는 문법 형태소
- IMJ : 입력되는 문법 형태소의 중성
- IMCV : 입력되는 문법 형태소의 초성과 중성
- 1 : 입력이 enable됨을 나타내는 표기

(2) 출력 플레이스에 나타나는 표기

- OC : 출력되는 초성 OJ : 출력되는 중성
- OJN : 출력되는 중성을 발음 안함
- +1 : 출력이 enable됨을 나타내는 표기

3.3 표준 발음법의 페트리 넷 모델링

<표 3>의 표준 발음법 제10항은 겹받침에 대한 일반항과 '뱌-'와 '녠-'에 대한 특수항으로 구분된다. 일반항은 'ㄱ'이 어말에 사용되면 'ㄱ'으로, 'ㄲ'이 자음앞에서 'ㄱ'으로, 'ㄴ'이 어말에서 'ㄴ'으로 'ㄴ'이 자음앞에서 'ㄴ'이 됨을 의미한다. 이것을 표현하는 페트리 넷은 25개(겹받침 6, 어말 1, 자음 18)의 입력 플레이스와 4개의 출력 플레이스, 120개의 트랜지션으로 구성된다.

특수항의 예로 「뱌-」은 자음 앞에서 [뱌-]으로, 「녠-」은 「녠죽하다」와 「녠똥글다」의 경우에 각각 [뉼쭈카다]와 [뉼똥글다]로 발음한다」라는 조항이 있다. 따라서, 일반항의 규정에 따르면 「뱌-」은 [뱌-]로 발음될 듯 하지만 특수항의 규정에 따라서 [뱌-]으로 발음되어야 하며, 마찬가지로 「녠-」도 특수항의 조건에 맞으면 [녠]으로 발음하여야 한다. 이 같은 예의 사항은 예외 사전으로 구축하여 별도로 처리하며, 나머지 항들도 정의된 용어와 용어의 내용에 따라 각각의 근접 행렬 표현으로 구성한다.

(그림 2)는 페트리 넷 정의와 격발의 정의^[11]에 따라 10항의 규칙을 각각의 소항목으로 나누어 항목별로 모

표 3. 표준 발음법 제10항

Table 3. Rule 10 of standard pronunciation rule.

제10항 겹받침 'ㄱ', 'ㄴ', 'ㄷ', 'ㄹ', 'ㄺ', 'ㄻ', 'ㄼ'은 어말 또는 자음 앞에서 각각 [ㄱ], [ㄴ], [ㄹ], [ㄻ]으로 발음한다.			
뉼[뉼]	뉼과[뉼과]	얏다[안따]	어뉼[어뉼]
녠다[녠따]	외꺼[외꺼]	얏다[얏따]	꺼[꺼]
얏다[얏따]			
다만, '뱌-'은 자음 앞에서 [뱌]으로 발음하고, '녠-'은 다음과 같은 경우에 [뉼]으로 발음한다.			
(1) 뱌다[뱌따]	뱌소[뱌쏘]	뱌지[뱌찌]	
뱌느[뱌느]	뱌게[뱌게]	뱌고[뱌꼬]	
(2) 뉼 - 죽하다[뉼쭈카다] 뉼 - 똥글다[뉼똥글다]			

델링한 것이다. 10항 내의 각 규칙(소항목)을 페트리 넷의 트랜지션과 사상시킬때, 조건부에 사용된 요인은 해당 트랜지션의 플레이스로 사용하였고, 출력부에 사용된 요인은 해당 트랜지션의 출력 플레이스로 모델링하였다. (그림 2)는 10항의 트랜지션 중에서 1001, 1002와 1007, 1008만을 통합된 페트리 넷으로 나타낸 것이다.

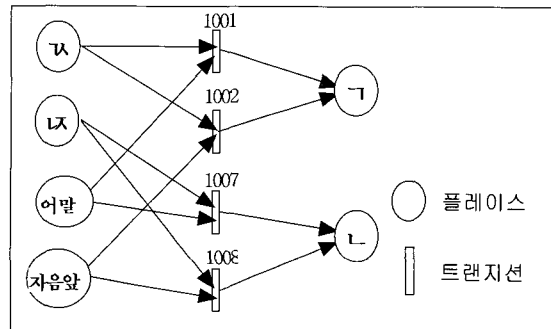


그림 2. 10항의 소항목 1001, 1002, 1007, 1008이 통합된 페트리 넷

Fig. 2. Integrated Petri net of subclauses 1001,1002, 1007,1008.

(그림 2)의 플레이스는 초기 마킹 Mo가 놓여지지 않은 상태이며, 플레이스에 놓여지는 초기 마킹은 입력되는 문자열의 형태소 분석 정보와 자모 분리 결과에 따라 동적으로 놓여지게 된다. 예를 들어, 입력 문자열로 「뉼」이라는 글자가 입력되면 형태소 분석과 자모 분리 결과로 「어말」과 「ㄱ」이 생성되고 이들에 해당되는 플레이스에 토큰이 놓여진다. 그러면 트랜지션 1001이 활성화(enable)되며, 트랜지션 1001이 격발하게 되면 출력 플레이스 「ㄱ」에 토큰이 놓여지게 된다.

3.4 페트리 넷 모델의 근접 행렬 변환

(그림 2)와 같이 작성된 페트리 넷 모델을 통합과 분석을 용이하게 하기 위하여 근접 행렬(incidence matrix)로 표현하였다. 근접 행렬 C는 |P| x |T| 행렬이며, C의 일반항은 다음과 같이 정의한다[2,3].

$$c_{ij}=1 \text{ if } (t_j, p_i) \in F, -1 \text{ if } (p_i, t_j) \in F, 0 \text{ otherwise}$$

<표 4>는 (그림 2)의 페트리 넷을 근접 행렬로 변환한 것으로 이해를 돕기 위하여 10항을 간소화하여 나타내었다. 즉, 실제로는 <표 4>의 입력 플레이스 중 '자음앞'은 18개의 자음으로 확장되어 모델링된다.

표 4. 표준 발음법 10항의 페트리 넷 근접 행렬
Table 4. Petri net incidence matrix of clause 10 of standard korean pronunciation rule.

R10 ①	1001 ②	1002	1003	1004	1005	1006	1007	1008	1009	1010	1011	1012
pa ③	-1⑦						-1⑧					
px ③		-1						-1				
pe ③			-1						-1			
pa ③				-1						-1		
pe ③					-1						-1	
pa ③						-1						-1
⑤												
이만 ④	-1⑦	-1	-1	-1	-1	-1						
자음앞④							-1⑧	-1	-1	-1	-1	-1
⑤												
QJ-1 ③	+1⑧							+1⑥				
QJ- ⑤		+1							+1			
QJ- ⑤			+1	+1	+1				+1	+1	+1	
QJ- ⑤						+1						+1

<표 4>의 근접 행렬에서 ①~⑧은 테이블을 설명하기 위한 주석 표시이며 각 주석이 표시하는 내용은 다음과 같다.

- ① 표준 발음법의 항 번호
- ② 항의 각 소항목을 나타내는 번호(트랜지션)
- ③ 장전 조건 1(입력 플레이스1)
- ④ 장전 조건 2(입력 플레이스2)
- ⑤ 격발 후 출력(출력 플레이스)
- ⑥ 각 조건을 구분하기 위한 공백
- ⑦ 해당 소항목의 조건부
- ⑧ 해당 소항목의 결론부

표준 발음법의 모든 항에 대하여 <표 4>와 같이 완성된 각각의 근접 행렬은 페트리 넷의 통합과 일관성

검증 프로그램의 입력 자료로 사용하기 위하여 완성형 텍스트 형태로 저장된다.

IV. 표준 발음법의 일관성 검증

4.1 표준 발음법에 나타나는 문제점

(1) 정의된 두 항의 내용이 상충되는 경우(상충)

상충이란 9항과 18항에서처럼 정의된 내용 중의 일부가 서로 다른 음가를 요구하는 경우로, 어느 한쪽에 우선 순위를 부여하거나, 해석의 범위를 한정짓거나, 정의를 수정해야 하는 경우에 해당한다.

(2) 정의된 두 항의 내용이 같은 경우(중복)

18항과 12항의 각 항 중 일부가 서로 같은 내용에 대하여 두 번 정의된 경우와 23항과 30.1항의 경우처럼 사이시옷이 아니더라도 'ㅅ'은 'ㄱ, ㄷ, ㅂ, ㅈ'과 만나면 경음화가 일어나는데, 30.1항에서 사이시옷이라고 범위를 한정지어 같은 내용을 다시 정의한 경우가 이에 해당한다^[19].

(3) 원칙과 허용에서 선택의 문제가 발생하는 경우(선택)

10항과 20항의 "다만"처럼 원칙과 허용을 두어 두 가지가 모두 표준 발음법에 맞는 경우, 원칙과 허용 중 어느 것을 취하고 어느 것을 버릴지 선택해야 하는 경우가 이에 해당한다.

4.2 일관성 검증을 위한 구조체와 모듈 구성

(1) 근접 행렬을 위한 구조체

<표 4>에 보인 근접 행렬의 첫 번째 열은 문자열로 된 정보이며, 두 번째 열 이후는 정수형의 +1 또는 -1의 값을 갖게 되므로 한 행의 내용을 표현하기 위하여 다음과 같은 C언어의 구조체를 사용한다.

<표 5>에서 선언된 구조체의 내용을 살펴보면 문자형 변수와 정수형 변수의 2개의 필드가 있다. char Condition[ConditionMax]는 <표 4>의 ③, ④, ⑤의 레이블이 기록되는 부분이다. ConditionMax는 통합될 레이블 문자열의 최대 길이를 의미한다.

int Bit[BitMax]는 레이블의 위치값을 갖는다. 즉, <표 4>의 1012 트랜지션의 '-1' 또는 '+1' 값은 Bit 배열의 12번째 위치에 '-1' 또는 '+1'로 기록된다. BitMax는 각 항의 트랜지션 수를 첨자로 가지며, 입력 플레이스는 항상 초성과 종성으로 구분하여 저장되고, 출력 플레이스는 초성과 종성의 구분 없이 저장된다.

표 5. 근접 행렬을 표현하기 위한 C언어 구조체
Table 5. A structure of C language for incidence matrix representation.

```
struct Rule{
    char Condition[ConditionMax];
    int Bit[BitMax]
} RuleTable[3][MainTableMax];
```

RuleTable[3][MainTableMax]는 하나의 테이블이 저장될 공간을 정의하는 부분이다. 배열의 첫 번째 첨자인 '3'은 근접 행렬이 입력 플레이스 1, 입력 플레이스 2, 출력 플레이스의 세부분으로 나누어져 있어 각각을 따로 저장하기 위해서이며, 두 번째 첨자인 'MainTableMax'는 전체 테이블이 갖는 트랜지션 수의 총합을 첨자로 갖는다.

(2) 일관성 검사 모듈의 구성

표준 발음법 일관성 검사 모듈은 작성된 각각의 근접 행렬 테이블 파일을 읽어들이 하나의 테이블로 통합하고, 통합된 결과를 트랜지션 단위로 비교하면서 일관성을 검사하는 기능을 수행한다. 이 시스템은 기능상으로 테이블 통합 모듈과 일관성 검사 모듈로 구성되어 있다.

테이블 통합 모듈은 각각의 파일로 저장되어 있는 각 항들의 파일을 읽어 들여 각 플레이스를 비교하고, 새로운 플레이스의 추가와 기존의 플레이스에 새로운 '-1' 값 또는 '+1' 값을 추가하는 역할을 하게 된다. 일관성 검사 모듈은 통합된 테이블에서 트랜지션 단위로 비교하면서 중복항과 상충항을 검색하는 기능을 수행한다.

V. 통합 테이블의 압축과 음가 변환 테이블 생성

5.1 통합 테이블의 압축

[11-13]에서 제안한 통합된 표준 발음법 페트리 넷 근접 행렬 테이블에는 음운 변동이 일어나는 항 번호가 트랜지션의 레이블로 기록되어 있다. 여기에서 트랜지션 번호 즉, 음운 변동을 일으키는 표준 발음법의 항 번호는 직접적으로는 음운 변동에 필요 없는 정보이므로, 이 트랜지션 번호는 음운 변동의 규칙에서 삭제할

수 있다.

페트리 넷 모델링에서 트랜지션에 연결되는 입력 플레이스 1과 입력 플레이스 2의 수는 항상 각각 1개씩이며, 출력 플레이스는 초성이 변하는 경우, 초성과 종성이 동시에 변하는 경우가 있으므로 출력 플레이스의 수는 1 또는 2개가 될 수 있다.

따라서, 입력 플레이스 1과 입력 플레이스 2의 모든 레이블에 나타날 수 있는 중복을 제거하고, 출력 플레이스에 나타나는 초성이 변하는 경우와 종성이 변하는 경우에 대하여 하나하나씩 재기록하게 되면 테이블이 압축된다.

표준 발음법 근접 행렬 테이블에서 입력 플레이스 1에 사용되는 레이블은 종성 27개와 초성에 나타날 수 있는 문법 형태소 12개로 총 39개의 입력 플레이스 종류가 있다. 그리고, 입력 플레이스 2에는 초성으로 사용될 수 있는 자음 18개와 초성에 나타날 수 있는 문법 형태소 37개로 총 55개의 입력 플레이스가 있다.

표 6. 압축된 규칙을 위한 C언어 구조체 정의
Table 6. A structure definition in C language for compressed rule.

```
struct Table{
    char Cho[3];
    char Jong[3];
}Tab[39][55];
```

출력 플레이스에는 초성과 종성이 둘 다 나타날 수 있으며, 초성과 종성이 하나만 나타나거나 초성과 종성이 동시에 나타나는 경우도 있으며, 이러한 사항들을 고려하여 정의한 것이 <표 6>이다. 정의된 구조체에서 가로축의 인덱스가 되는 초성과 문법 형태소들은 표준 발음법 페트리 넷 근접 행렬의 입력 플레이스 2에 나타날 수 있는 것으로 <표 7>과 같이 정의되며, 세로축의 인덱스가 되는 종성과 문법 형태소들은 표준 발음법 페트리 넷 근접 행렬의 입력 플레이스 1에 나타날 수 있는 것으로 <표 8>과 같이 정의된다.

<표 7>과 <표 8>에서 나타나는 순서가 인덱스 번호가 되며, <표 7>의 가로축 인덱스에서 'IM모음 조사'는 전체 모음조사에서 'IM이조사'를 제외한 나머지 모음 조사를 의미하고, 'IM모음접미사'는 전체 모음접미사에서 'IM이접미사, IM야접미사, IM요접미사, IM유접미사'를 제외한 나머지 모음접미사를 의미한다.

6.3 압축 모듈

(그림 6)은 통합된 페트리 넷 근접 행렬 파일을 입력으로 받아, 이를 압축된 음가 생성 규칙으로 출력하는 알고리즘이다. Step 1에서는 입력된 통합 근접 테이블을 배열로 읽어 들이며, Step 2에서는 첫번째 트랜지션과 그에 연결된 입력 플레이스 1, 2를 읽어 들인 후, 입

```

Input : 근접 행렬 통합 테이블
Output : 검증 결과

Table 통합된 근접 행렬을 저장할 배열
int n, m=0;

BEGIN
Step1 : Table←통합된 근접 행렬
Step2 : FOR m=0 TO 테이블의 마지막 트랜지션
        테이블의 트랜지션m과 연결된 플레이스 선택
Step3 : FOR n=1 TO 테이블의 마지막 트랜지션
        테이블의 트랜지션n과 연결된 플레이스 선택
Step4 : IF m!=n인 트랜지션m의 플레이스 1과
        트랜지션n의 플레이스 1이 같으면
        THEN IF 트랜지션m의 플레이스2와
        트랜지션n의 플레이스2가 같으면
        THEN IF 트랜지션 m, n의 출력 플레이스가
        같으면
        THEN printf("중복, 트랜지션 m, 트랜지션 n")
        ELSE printf("상충, 트랜지션 m, 트랜지션 번
        호 n")
Step 5 : 검증 결과 출력
END
    
```

그림 5. 상충 및 중복 정의 검사 알고리즘
Fig. 5. The algorithm used to check conflicts and duplications.

력 플레이스 1의 레이블과 <표 8>의 중성 배열 JongIdx를 비교하여 일치되는 JongIdx의 배열 첨자 변수 x에 저장한다. 그리고, 입력 플레이스 2의 레이블과 <표 7>의 초성 배열 ChoIdx와 비교하여 일치되는 ChoIdx 배열의 첨자를 변수 y에 저장한다.

이러한 결과는 <표 6>에 보인 배열 Tab[x][y]의 주소를 결정하는데 사용된다. 출력 플레이스는 최대 2개가 될 수 있으며, 중성 변화를 나타내는 OJ는 Tab[x][y].Jong에, 초성 변화를 나타내는 OC는 Tab[x][y].Cho에 저장한다. Step 3에서는 압축이 완료된 테이블을 출력한다.

6.4 재처리 모듈

(그림 7)은 압축된 테이블을 대상으로 하여 출력된

```

Input : 통합된 표준 발음법 페트리 넷 근접 행렬 파일
Output : 압축된 테이블

Table : 통합된 근접 행렬을 저장할 배열
Tab[][] : 압축된 테이블을 저장할 구조체 배열
int m=0,x=0,y=0;

BEGIN
Step 1 : Table ← 통합된 근접 행렬
Step 2 : FOR m=0 TO 테이블의 마지막 트랜지션
        트랜지션 m과 연결된 입력 플레이스 1과 2 선택
        x ← 입력 플레이스 1의 내용과 일치하는 인덱스 번호
        y ← 입력 플레이스 2의 내용과 일치하는 인덱스 번호
        트랜지션 m과 연결된 출력 플레이스1과 2 선택
        IF 출력 플레이스1 또는 2의 내용이 OC 이면
        THEN Tab[x][y].Cho←플레이스 1 또는 2의 내용
        IF 출력 플레이스1 또는 2의 내용이 OJ이면
        THEN Tab[x][y].Jong←플레이스 1 또는 2의 내용
Step 3 : 압축 결과 출력
END
    
```

그림 6. 통합된 페트리 넷 근접 행렬의 압축 알고리즘
Fig. 6. The algorithm used to compress the integrated Petri net incidence matrix.

```

Input : 압축된 표준 발음법 음가 변환 규칙
Output : 재처리된 음가 변환 규칙
Tab[39][50] : 압축된 테이블
int c,c',j, j'
BEGIN
Step 1 : FOR j=0 To 39(중성의 수)
        FOR c=0 To 55(초성의 수)
        IF Tab[c][j].Cho가 NULL이 아니면
        c' ← Tab[c][j].Cho에 기록된 초성의 인덱스
        번호
        IF Tab[c][j].Jong이 NULL이 아니면
        j' ← Tab[c][j].Jong에 기록된 내용의 인덱스
        번호
Step 2 : IF Tab[c][j].Jong과 Tab[c][j].Cho이 NULL
        이 아니면
        THEN IF Tab[c][j].Jong 과 Tab[c'][j'].Jong
        이 다르면
        THEN print("변경 요청");
        IF Tab[c][j].Cho 와 Tab[c'][j'].Cho가 다르면
        THEN print("변경 요청");
Step 3 : 변화된 초성과 중성이 있으면 Goto Step 1
Step 4 : 재처리 검색 결과 출력
END
    
```

그림 7. 압축된 음가 변환 테이블의 재처리 알고리즘
Fig. 7. The algorithm used to reprocess the compressed phonetic value table.

음운 변동의 결과값이, 다른 음운 변동의 원인이 되는 압축된 Tab[39][55]배열의 첫 번째 초성과 첫 번째 중성 위치의 내용을 읽어와서 c'에는 초성의 인덱스 번호를 저장하고, j'에는 중성의 인덱스 번호를 저장한다. Step 2에서는 Tab[c'][j']와 Tab[c][j]의 내용을 비교하여 내용이 틀릴때에 "변경 요청"이라고 출력한다. Step 3에서는 더이상의 "변경 요청"이 발생하지 않을 때까지 반복하여 검사하며, Step 4에서 검색된 재처리 결과를 출력한다.

6.5 실험 및 결과

음가 변환기의 구현 환경은 변환이 완료된 음가를 표기하기 위하여, 통합형 한글을 지원하는 Windows98 시스템 상에서 Boland C++ 컴파일러 버전 5.0의 C++ 언어로 구현한다.

통합 모듈은 3장에서 설명한 근접 행렬을 입력으로 하여 만들어진다. 입력용으로 사용되는 근접 행렬은 완성형 텍스트로 저장된 화일의 형태이며, 각 항이 별도의 파일로 작성되어 있다. 통합 모듈은 각 항을 읽어 들여 통합 알고리즘에 따라 통합 과정을 수행하여 텍스트 형태의 파일로 출력하며, (그림 8)은 통합 모듈의 출력 결과중 일부를 보인 것이다. 첫 번째 행은 트랜지션의 번호를 나타내며, 첫 번째 열은 플레이스 명을 나타낸다.

일관성 검사 모듈은(그림 8)의 통합 모듈이 출력한 결과를 입력으로 사용하며,(그림 9)는 일관성 검사 모듈의 출력 결과중 일부분을 보인 것이다. 일관성 검사

Place / transition	9001	9002	12001	12002	30012
IJ기	-1				
IJㅋ		-1			
IJㅅ					
	<중략>				
IMJ용언어간말음					
IJㅎ				-1	-1
IJㅁ					
	<중략>				
IMJ복합어자음					
IMJ복합어자음ㄹ					-1
IMJ사이사잇					
IM어말	-1	-1			
IC기				-1	
ICㅋ					
	<중략>				
OJㄴ					+1
OJㄹ					
OJㅁ					
OCㅋ				+1	
OJNㅎ				+1	
OCㅌ				+1	
OCㅊ					

그림 8. 통합 모듈의 출력 결과
Fig. 8. The result of the integration module.

모듈은 통합 테이블에서 하나의 트랜지션을 읽어온 후, 나머지 트랜지션과 차례대로 비교하여 상충항과 중복항이 있는지에 대해서 검사한다.

표준 발음법의 18항과 9, 10, 11항 및 12항과 9, 11, 13, 14항 그리고, 23항과 10항은 동일한 조건하에서 서로 다른 음가 변환이 이루지는 상충항이다. 이와 같은 각 항간의 상충 문제는 각 항에 대해 우선 순위를 부여하거나 그 정의에 대한 수정이 필요하다. 다음의 1) ~ 8)은 일관성 검사 모듈의 출력 결과중, 상충의 실제 예를 몇가지 보인 것이다.

<중략>				
상충4) 트랜지션 18003과 트랜지션 9011이 동일한 조건에 다른 출력을 갖는 규칙입니다;				
TRnum =>18003 , Place1 IJ기, Place2 ICㄴ , OutPlace OJㅇ				
TRnum => 9011 , Place1 IJ기, Place2 ICㄴ , OutPlace OJ기				
상충4) 트랜지션 18004와 트랜지션 9015가 동일한 조건에 다른 출력을 갖는 규칙입니다;				
TRnum =>18004 , Place1 IJ기, Place2 ICㅁ , OutPlace OJㅇ				
TRnum => 9015 , Place1 IJ기, Place2 ICㅁ , OutPlace OJ기				
상충4) 트랜지션 18005와 트랜지션 9029가 동일한 조건에 다른 출력을 갖는 규칙입니다;				
TRnum =>18005 , Place1 IJㅋ, Place2 ICㄴ , OutPlace OJㅇ				
TRnum => 9029 , Place1 IJㅋ, Place2 ICㄴ , OutPlace OJ기				
<중략>				

그림 9. 일관성 검사의 출력 결과
Fig. 9. The result of consistency test.

1) 18항과 9항

항	규칙	예	음가	결과
18003	기+ㄴ=ㅇ	닭는	당는	○
9011	기+ㄴ=기		닭는	×
18005	ㅋ+ㄴ=ㅇ	부엌난간	부엌난간	○
9029	ㅋ+ㄴ=기		부엌난간	×
18013	ㅅ+ㄴ=ㄴ	짚는	진는	○
9047	ㅅ+ㄴ=ㄷ		짚는	×
18015	ㅅ+ㄴ=ㄴ	있는	인는	○
9065	ㅅ+ㄴ=ㄷ		인는	×
18017	ㅈ+ㄴ=ㄴ	맞는	만는	○
9083	ㅈ+ㄴ=ㄷ		맞는	×

항	규칙	예	음가	결과
18019	ㅈ+ㄱ=ㄴ	꽃망울	곶망울	○
9101	ㅈ+ㄱ=ㄷ		곶망울	×
18021	ㅊ+ㄴ=ㄴ	붉는	부는	○
9119	ㅊ+ㄴ=ㄷ		부는	×
18027	ㅍ+ㄴ=ㅁ	앞날	앙날	○
9137	ㅍ+ㄴ=ㅂ		앙날	×
18006	ㅋ+ㄱ=ㅇ	부엌문	부엌문	○
9033	ㅋ+ㄱ=ㄱ		부엌문	×
18014	ㅅ+ㄱ=ㄴ	웃맵시	온맵시	○
9051	ㅅ+ㄱ=ㄷ		온맵시	×
18028	ㅍ+ㄱ=ㅁ	옆막이	엎마기	○
9141	ㅍ+ㄱ=ㅂ		엎마기	×
18019	ㅈ+ㄴ=ㄴ	쫓는	쫓는	○
9101	ㅈ+ㄴ=ㄷ		쫓는	×

우선순위 : 18항이 9항 보다 우선 순위 높음
 수정내용 : 9항의 “자음 앞”을 “ㄴ, ㅁ을 제외한 자음 앞”으로 수정

2) 18항과 10항

항	규칙	예	음가	결과
18007	ㅈ+ㄴ=ㅇ	상는	상는	○
10009	ㅈ+ㄴ=ㄱ		상는	×
18033	ㅃ+ㄴ=ㅁ	값매다	값매다	○
10100	ㅃ+ㄴ=ㅂ		값매다	×
18029	ㅃ+ㄴ=ㅁ	밝는	밤는	○
10045	ㅃ+ㄴ=ㄷ		밤는	×
18034	ㅃ+ㄱ=ㅁ	값매기다	값매기다	○
10104	ㅃ+ㄱ=ㅂ		값매기다	×

우선순위 : 18항이 10항 보다 우선 순위 높음
 수정내용 : 10항의 “자음 앞”을 “ㄴ, ㅁ을 제외한 자음 앞”으로 수정

3) 18항과 11항

항	규칙	예	음가	결과
18005	ㄹ+ㄴ=ㅇ	굵는	굵는	○
11005	ㄹ+ㄴ=ㄱ		굵는	×
18031	ㄹ+ㄴ=ㅁ	옴는	옴는	○
11042	ㄹ+ㄴ=ㅂ		옴는	×
18022	ㄹ+ㄱ=ㅇ	흙만	흙만	○
11008	ㄹ+ㄱ=ㄱ		흙만	×
18030	ㄹ+ㄱ=ㅁ	옴만	옴만	○
11049	ㄹ+ㄱ=ㅂ		옴만	×

우선순위 : 18항이 11항 보다 우선 순위 높음
 수정내용 : 11항의 “자음 앞”을 “ㄴ, ㅁ을 제외한 자음 앞”으로 수정

4) 12항과 9항

항	규칙	예	음가	결과
12015	ㅈ+ㅎ=Nㅈ	낮한때	나탄때	○
9098	ㅈ+ㅎ=ㄷ		나탄때	×

우선순위 : 12항이 9항 보다 우선 순위 높음
 수정내용 : 9항의 “자음 앞”을 “ㅎ을 제외한 자음 앞”으로 수정

5) 12항과 11항

항	규칙	예	음가	결과
12011	ㄹ+ㅎ=ㄹ	흙형	흙형	×
11021	ㄹ+ㅎ=ㄱ		흙경	○

우선순위 : 11항이 12항 보다 우선 순위 높음
 수정내용 : 9항의 “자음 앞”을 “ㅎ을 제외한 자음 앞”으로 수정

6) 12항과 13항

항	규칙	예	음가	결과
12027	ㅎ+모어=Nㅎ	날은	나은	○
13028	ㅎ+모어=ㅎ		나흔	×
12030	ㅎ+모접=Nㅎ	쌀이다	싸이다	○
13042	ㅎ+모접=ㅎ		싸히다	×

우선순위 : 12항이 13항 보다 우선 순위 높음
 수정내용 : 13항의 “홀발침”을 “ㅎ을 제외한 홀발침”으로 수정

7) 12항과 14항

항	규칙	예	음가	결과
12028	ㄴㅎ+모어= Nㅎ	않은	아는	○
14014	ㄴㅎ+모어= ㅎ		안흔	×
12029	ㄹㅎ+모어= Nㅎ	달아	다라	○
14021	ㄹㅎ+모어= ㅎ		달하	×
12031	ㄴㅎ+모접= ㅎ	많아	마나	○
14025	ㄴㅎ+모접= ㅎ		만하	×
12032	ㄹㅎ+모접= Nㅎ	싫어도	시러도	○
14032	ㄹㅎ+모접= ㅎ		실허도	×

우선순위 : 12항이 14항 보다 우선 순위 높음
 수정내용 : 13항의 “홀발침”을 “ㅎ을 제외한 홀발침”으로 수정

8) 23항과 10항

항	규칙	예	음가	결과
23014	ㄹ+ㄱ=ㄴ	넓고	널꼬	×
10043	ㄹ+ㄱ=ㄴ		널꼬	○
23046	ㄹ+ㅅ=ㅆ	넓부	널뿌	○
10050	ㄹ+ㅅ=ㅆ		널뿌	×
23078	ㄹ+ㅈ=ㅉ	널찍	널찍	○
10054	ㄹ+ㅈ=ㅉ		널찍	×
23030	ㄹ+ㄷ=ㄴ	널따	널따	×
10046	ㄹ+ㄷ=ㄴ		널따	○
23062	ㄹ+ㅌ=ㄴ	널씨	널씨	○
10052	ㄹ+ㅌ=ㄴ		널씨	×

우선순위 : 10항이 23항보다 우선순위 높음

수정내용 : 23항의 규정에는 다음 글자 초성이 된 소리로 바뀌는 규칙이 포함되어 있어 23항의 다음 글자 초성이 된소리로 바뀌는 부분만 취함

다음으로 18항과 12항의 일부가 동일한 내용을 두 번 정의하는 중복 정의의 경우에는 두 개의 정의 중 하나를 제거한다.

항	규칙	예	음가	결과
18012	ㅎ + ㄴ = ㄴ	놓는	논는	○
12023	ㅎ + ㄴ = ㄴ		논는	○

우선순위 : 동일한 내용의 트랜지션

수정내용 : 트랜지션 12023 삭제

종성초성	기	끼	니	디	띠	리	미	비	피
ㄱ(초성) (종성)	기	기	기	기	기	기	기	기	기
ㅋ(초성) (종성)	기	기	기	기	기	기	기	기	기
ㄴ(초성) (종성)	기	기	기	기	기	기	기	기	기
ㄷ(초성) (종성)	기	기	기	기	기	기	기	기	기
ㄹ(초성) (종성)	기	기	기	기	기	기	기	기	기
ㄴ(초성) (종성)	기	기	기	기	기	기	기	기	기
ㄷ(초성) (종성)	기	기	기	기	기	기	기	기	기
ㄹ(초성) (종성)	기	기	기	기	기	기	기	기	기

그림 10. 음가 생성 규칙 압축 시스템의 출력 결과
Fig. 10. The result of the compression system of Korean voice look-up table.

(그림 10)은 구현된 시스템의 출력 내용중 일부분으

로 압축된 음가 생성 규칙을 보인 것이다. 예를 들어, 1행-1열의 내용은 종성 'ㄱ' 과 초성 '기'이 만나면 초성이 '기'으로 바뀌고 종성은 변화가 없음을 의미한다.

즉, 음가 생성 단계에서 압축된 테이블의 접근은 형태소 분석 결과와 자모의 종류에 따라서 가로축과 세로축 좌표로 행해지며, 해당 지점에 표기된 내용으로 초성과 종성이 변화된다.

(그림 11)은 음가 변환 규칙 재처리 시스템의 출력 결과 중 일부로서, 테이블의 모든 레이블에 대한 음가 변환 결과를 추적하여 변경이 요청되는 자모를 출력한 것이다. <표 10>은 재처리 모듈의 검색 결과를 요약한 것으로 해당 알고리즘에 따라 여러 번의 변환 과정을 거치게 되는 것도 있으므로 재처리 검색 모듈 처리의 첫 번째 입력으로 사용되었던 종성 및 초성과 최종적으로 재처리가 완료된 종성과 초성을 나타내었다.

(그림 12)는 음가 변환기의 출력 화면을 보인 것이다. 음가 변환기는 입력되는 문자열을 각 단어 별로 분리하여 단어별 음가 변환 결과를 출력한다. (그림 13)은 (그림 11)의 결과를 바탕으로 (그림 10)의 음가 변환 규칙 테이블을 수정하고, 포함 관계에 의하여 복사해야 하는 경우에 대한 모든 처리를 마친 음가 변환 테이블을 보인 것이다.

(그림 13)의 굵은 선으로 그려진 사각형은 시스템의 출력 중 '변경요청'에 의하여 변경된 것이고, 그 이외의 부분(음가 변환 결과, 초성과 종성의 변경 요청이 발생

<중략>	
종성 J _기 , 초성 C _기 :	종성 ㄱ (초성 기) -> 종성, 초성 ㄴ 변경요청
종성 J _끼 , 초성 C _비 :	(종성 기), 초성 비 -> 종성 ㄱ, 초성 ㅂ 종성 ㄱ (초성 비) -> 종성 ㄱ, 초성 ㅂ
<중략>	
종성 J _기 , 초성 C _ㅎ :	종성 ㄱ (초성 ㅎ) -> 종성 N _기 , 초성 ㅋ 변경요청
종성 J _니 , 초성 C _기 :	(종성 ㄴ), 초성 기 -> 종성 ㄱ, 초성 ㄱ 종성 ㄱ (초성 기) -> 종성 ㄱ, 초성 기
종성 J _디 , 초성 C _디 :	(종성 ㄷ), 초성 디 -> 종성 ㄱ, 초성 ㄷ 종성 ㄱ (초성 디) -> 종성 ㄱ, 초성 디
<후략>	

그림 11. 음가 변환 규칙 재처리 시스템의 출력 결과
Fig. 11. An example output of the reprocessing system of Korean voice look-up table.

표 10. 재처리 검색 결과의 요약
Table 10. Summary of reprocess result

입력		재처리전		재처리후		입력		재처리전		재처리후	
종성	초성	종성	초성	종성	초성	종성	초성	종성	초성	종성	초성
ㄱ	ㄹ	ㄱ	ㄹ	ㅇ	ㄴ	ㄱ	ㅎ	ㄱ	ㅎ	ㄱ	ㅋ
ㄱ	ㄹ	ㄱ	ㄹ	ㅇ	ㄴ	ㄴ	ㅎ	ㄱ	ㅎ	ㄱ	ㅋ
ㄴ	ㄹ	ㄱ	ㄹ	ㅇ	ㄴ	ㄹ	ㅎ	ㅂ	ㅎ	ㅂ	ㅍ
ㄹ	ㄹ	ㄱ	ㄹ	ㅇ	ㄴ	ㅅ	ㅎ	ㄷ	ㅎ	ㄷ	ㅌ
ㄹ	ㄹ	ㅂ	ㄹ	ㅇ	ㄴ	ㅅ	ㅎ	ㄷ	ㅎ	ㄷ	ㅌ
ㄴ	ㄹ	ㅂ	ㄴ	ㅇ	ㄴ	ㅅ	ㅎ	ㄷ	ㅎ	ㄷ	ㅌ
ㅅ	ㄹ	ㅂ	ㄴ	ㅇ	ㄴ	ㅅ	ㅎ	ㄷ	ㅎ	ㄷ	ㅌ
ㅋ	ㄹ	ㄱ	ㄹ	ㅇ	ㄴ	ㅅ	ㅎ	ㄷ	ㅎ	ㄷ	ㅌ
ㅌ	ㄹ	ㅂ	ㄴ	ㅇ	ㄴ	ㅅ	ㅎ	ㄷ	ㅎ	ㅂ	ㅍ
ㄹ	ㄴ	ㄹ	ㄴ	ㄹ	ㄴ						

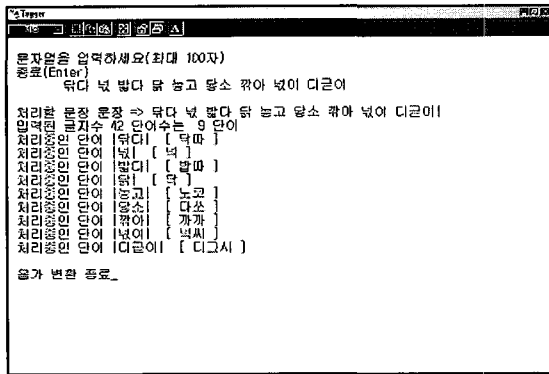


그림 12. 음가 변환기의 출력 결과
Fig. 12. An example output of the Korean voice synthesis system.

종성/초성	ㄱ	ㅋ	ㄴ	ㄷ	ㅌ	ㄹ	ㅂ	ㅍ	ㅃ	ㅅ	ㅆ	ㅈ	ㅊ	ㅅ	ㅆ	ㅈ	ㅊ
ㄱ(초성): (종성):	ㄱ	ㅋ	ㄴ	ㄷ	ㅌ	ㄹ	ㅂ	ㅍ	ㅃ	ㅅ	ㅆ	ㅈ	ㅊ	ㅅ	ㅆ	ㅈ	ㅊ
ㅋ(초성): (종성):	ㄱ	ㅋ	ㄴ	ㄷ	ㅌ	ㄹ	ㅂ	ㅍ	ㅃ	ㅅ	ㅆ	ㅈ	ㅊ	ㅅ	ㅆ	ㅈ	ㅊ
ㄴ(초성): (종성):	ㄱ	ㅋ	ㄴ	ㄷ	ㅌ	ㄹ	ㅂ	ㅍ	ㅃ	ㅅ	ㅆ	ㅈ	ㅊ	ㅅ	ㅆ	ㅈ	ㅊ
ㄷ(초성): (종성):	ㄱ	ㅋ	ㄴ	ㄷ	ㅌ	ㄹ	ㅂ	ㅍ	ㅃ	ㅅ	ㅆ	ㅈ	ㅊ	ㅅ	ㅆ	ㅈ	ㅊ
ㄹ(초성): (종성):	ㄱ	ㅋ	ㄴ	ㄷ	ㅌ	ㄹ	ㅂ	ㅍ	ㅃ	ㅅ	ㅆ	ㅈ	ㅊ	ㅅ	ㅆ	ㅈ	ㅊ
ㅂ(초성): (종성):	ㄱ	ㅋ	ㄴ	ㄷ	ㅌ	ㄹ	ㅂ	ㅍ	ㅃ	ㅅ	ㅆ	ㅈ	ㅊ	ㅅ	ㅆ	ㅈ	ㅊ
ㅍ(초성): (종성):	ㄱ	ㅋ	ㄴ	ㄷ	ㅌ	ㄹ	ㅂ	ㅍ	ㅃ	ㅅ	ㅆ	ㅈ	ㅊ	ㅅ	ㅆ	ㅈ	ㅊ

그림 13. 재처리가 완료된 음가 변환 테이블
Fig. 13. The result of reprocessing.

하지 않아서 해당 부분이 비어 있었던 곳)은 포함 관계를 이용하여 복사한 것이다.

<표 11>은 표준 발음법의 각 항에 대하여 제시된 단어와 실험 결과를 나타낸 것이다. 전체 30항 중 음가 생성에 관련된 364개의 예 전체를 대상으로 실험하였으나, 각 항에 대하여 1개씩의 예만 나타내었다.

표 11. 표준 발음법의 예제 실험 결과
Table 11. The experimental results of standard Korean pronunciation rule on some examples.

항 번호	예시어	표준 발음	실험 결과
9항	닭다	닥따	닥따
10항	넋	넉	넉
11항	닭	닥	닥
12항	놓고	노코	노코
13항	까아	까까	까까
14항	넋이	넉씨	넉씨
15항	발아래	바다래	바다래
16항	디글이	디그시	디그시
17항	벼훅이	벼훅치	벼훅치
18항	먹는	멍는	멍는
19항	담력	담넉	담넉
20항	날로	날로	날로
21항	강기	강기	강기
22항	되어	되어	되어
23항	국밥	국뻬	국뻬
24항	닭고	닥꼬	닥꼬
25항	넋개	넉개	넉개
26항	갈등	갈등	갈등
27항	할것을	할꺼슬	할꺼슬
28항	문고리	문꼬리	문꼬리
29항	숨이불	숨니불	숨니불
30항	넋가	내까	내까

Ⅶ. 결론 및 향후 과제

본 논문에서는 한국어 음성 합성을 위한 정확한 한국어 음가 변환과 음성 인식 분야의 후처리 기본 규칙으로 사용할 수 있는 '표준 발음법'을 페트리 넷으로 모델링할 수 있음을 보였다. 그 결과로서 표준 발음법 상의 각 규칙간에는 선택, 중복, 상충의 문제가 있음이 밝혀졌으며, 이를 배제한 한국어 음가 변환기를 구현하였다.

모델링된 페트리 넷을 대상으로 일관성 검사를 한 결과, 모두 39개의 상충항과 1개의 중복항을 찾아내었으며, 21개의 재처리 항목이 검출되었다. 따라서, 본 논문에서는 검증 결과를 바탕으로 하여 중복되거나 상충되는 항이 없는 무결한 표준 발음법으로의 수정 방향을 제시하였고, 정형화된 음운 변동 규칙을 이용하여 최종 테이블을 작성하였으므로 본 연구에서 작성된 테

이블은 무결성을 유지한다고 할 수 있다.

그리고, 본 논문에서 보인 한국어 음가 변환기는 기존 연구들에서 제시된 더 이상의 음운 변동이 없을 때까지 음운 변동 규칙을 반복하여 적용하던 방식이 아닌, 한번의 음가 변환 테이블 탐색으로 원하는 음가를 추출할 수 있다는 것을 보였다.

본 논문의 목적은 한국어 음성 합성을 위한 음가 변환 테이블 생성에 있다. 생성된 음가 변환 테이블은 형태소 분석기와 구문 분석 결과에 따라 생성되는 음가가 틀려질 수 있다. 따라서, 정확한 형태소 분석 결과를 생성하는 형태소 분석기의 구현이 요구되며, 또한 구문 분석기의 분석 결과에 따라 끊어 읽을 어절이 달라지고 발음도 달라지게 되므로 구문 분석기에 대한 연구도 요구된다.

참 고 문 헌

[1] 문화교육부, “표준어 규정”, 문교부 고시 제 88-2 호, 1988
 [2] T.Murata, “Pertri nets: properties, analysis and applications,” Proceeding of the IEEE, Vol77. no.4, pp.541~580, April 1989.
 [3] 이계영, 임재걸, “한글 받침 발음법의 페트리 넷 표현”, 동국논집 14집, pp. 155~167, 1995
 [4] 강승식, “음절 정보와 복수어 단위 정보를 이용한 한국어 형태소 분석”, 서울대학교 대학원 공학박사 학위논문, 1993
 [5] 서영훈, “의미정보를 이용하는 중심어 주도의 한

국어 파싱”, 서울대학교 대학원, 공학박사 학위논문, 1991.
 [6] 김재홍, “고품질 한국어 음성 합성을 위한 문서 음성 변환시스템”, 연세대학교 대학원, 석사 학위논문, 1998.
 [7] 김혜순, 변영태, 이기철, “멀티미디어를 이용한 한국어 발음 교육 시스템”, 한국 정보과학회 논문지 93.12 vol.20, NO1. 12
 [8] 한국과학기술원, “무제한 한국어 음성 합성 시스템”, 연구보고서, 1990.
 [9] 양진석, 김재범, 이정현 “운율 및 길이 정보를 이용한 무제한 음성 합성기의 설계 및 구현” 한국 정보처리학회 논문지, 제 3권 제 5호, 1996.
 [10] 서울대학교 사범대학 국어 교육 연구소, “고등 학교 문법”, 1996
 [11] 임재걸, 이계영, 김경정, “페트리 넷을 이용한 표준 발음법 분석 시스템 디자인”, 한국정보과학회 봄학술발표논문집, pp. 369~371, 1999.
 [12] 이계영, 임재걸, 김경정, 김규식, “페트리 넷을 이용한 표준 발음법 분석 시스템 구현”, 한국정보처리학회 춘계학술발표논문집, pp. 609~612, 1999
 [13] 이계영, 임재걸, 김경정 “표준 발음법 페트리 넷을 이용한 음가 변환기 설계”, 한국멀티미디어학회 춘계학술발표논문집 제2권1호, pp. 339~344, 1999.
 [14] 이희승, 안병희, “한글 맞춤법 강의”, 신구문화사, 1991
 [15] 이기문 외 9인, “국어 어문 규정집”, 대한교과서 주식회사, 1996.

저 자 소 개



李 桂 英(正會員)

1980년 : 동국대학교 전자계산학과 졸업. 1983년 : 동국대학교 대학원 전자계산학과 공학석사. 1992년 : 단국대학교 전자공학과(컴퓨터공학 전공) 공학박사. 1996~1997년 : Washington State University 방문교수. 1985~현재 : 동국대학교 컴퓨터정보통신학부 교수. <주관심분야> 자연어 처리, 음성 처리, 멀티미디어, 페트리넷 등



任 在 傑(正會員)

1981년 2월 : 동국대학교 전자계산학과 졸업. 1987년 3월 : University of Illinois at Chicago, 컴퓨터 과학 석사. 1990년 12월 : University of Illinois at Chicago, 컴퓨터 과학 박사. 1991-1992년 : 현대전자 과장, 1992~ 현재 : 동국대학교 컴퓨터정보통신학부 교수. <주관심분야> 한국어처리, 숙련가시시스템, 페트리 넷 이론 및 응용, 멀티미디어, 정보 보안 등