페트리넷을 이용한 한글-로마자 표기 변환표 생성에 관한 연구

김 경 징† · 최 영 규† · 이 상 범††

요 약

본 논문에서는 개정된 로마자 표기법에 일치하는 한글의 로마자 표기 생성을 위한 한글-로마자 표기 변환표의 생성에 관한 연구를 수행하였다. 로마자 표기법의 근간이 되는 표준 발음법과 로마자 표기법을 수학적으로 분석하기 위하여 페트리넷 모델을 이용한 자연 언어의 수학적 분석 방법을 도입하였다. 페트리넷 모델을 이용한 분석의 방법으로 한글 로마자 표기 변환 표를 생성하기 위한 방안과 로마자 표기법의 페트리넷 모델링을 통하여 그 실질적인 예를 보여 한국어의 수학적 모델링 방안과 적용방법을 제시한다. 생성된 한글-로마자 표기 변환표를 검증하기 위하여 윈도우 기반 응용 프로그램을 개발하고 로마자 표기 용례사전의 로마자 표기와 응용 프로그램의 결과를 비교하였다.

A Study on Creation of Hangeul-Romanization Conversion Table Using Petri-Nets

Kyoung Jing Kim† · Young Kyoo Choi† · Sang Burm Rhee††

ABSTRACT

In this paper, we proposed the formation of Korean - Roman alphabet notation conversion table for the generation of Korean - Roman alphabet notation that also meets revised Roman alphabet notation. Introduced a mathematical analyzing method of the natural language which used a petrinet model so that a base of Roman alphabet notation analyzed standard pronunciation and Roman alphabet notation to work mathematically. It display the practical example through a petrinet modeling of a plan and Roman alphabet notation to create a Korean Roman alphabet notation conversion table with the method of the analysis that used a petrinet model, and present a mathematical modeling plan and application method of Korean. We developed application program based on window in order to verify a created Korean - Roman alphabet notation conversion table, and compared the result of an application program with Roman alphabet notation of an Roman alphabet notation example dictionary.

키워드 : 로마자 표기법(Roman Alphabet Notation), 표준 발음법(Standard Pronunciation Rule of Hangeul), 한글(Hangeul), 페트리넷(Petrinet), 자연어처리(Natural Language Process)

1. 서 론

2000년 7월 4일 문화관광부에 의해 개정 공포된 「국어의 로마자 표기법」[1](이하 로마자 표기법)은 기존의 매쿤-라이샤워(McCune-Reischauer) 방법에 나타나는 반달표(˘)와 어깻점(′)을 배제하고 로마자만을 이용하여 한글을 로마자로 표기하도록 하는 방법으로 개정되었다. 개정된 방법으로 기존의 매쿤-라이샤워 방법보다는 쉽게 로마자 표기를 쓸 수 있게되었지만, 국어에 전문적인 지식인 없는 일반인들이 한국어를 로마자로 표기할 때 어려움이 많았다[2].

본 논문의 연구 목적은 개정된 로마자 표기법에 맞는 한글-로마자 표기 변환표의 생성과 로마자 표기법의 근간이 되는 표준 발음법[3]과 로마자 표기법의 분석에 목적이 있다. 이를 위하여 페트리넷을 이용한 자연 언어의 수학적 분석방법을 도입하였다. 페트리넷 모델을 이용한 표준 발음법과 로마자 표기법의 모델링 방법과 이를 이용하여 한글-로마자 표기 변환 표를 생성하기 위한 방안을 제시한다.

연구 방법으로서 한글-로마자 표기 변환표 생성을 위한 분석적 접근을 위하여 표준 발음법과 로마자 표기법의 상관 관계를 밝히고, 페트리넷을 이용하여 표준 발음법과 로마자 표기법을 모델링하기 위한 모델링 영역을 설정한다.

※ 이 연구는 2000학년도 단국대학교 대학연구비의 지원으로 연구되었음.
† 준 회 원 : 단국대학교 대학원 전자컴퓨터공학과
†† 종신회원 : 단국대학교 컴퓨터공학전공 교수
논문접수 : 2002년 2월 22일, 심사완료 : 2002년 9월 23일

표준 발음법과 로마자 표기법을 페트리넷을 이용하여 모델링 한 후, 모델링된 결과를 근접행렬로 변환하여 각각을 통합한 후 한글-로마자 표기 변환 표를 생성한다.

표준 발음법과 로마자 표기법의 정의는 기본이 되는 규칙인 '원칙'항과 일정 범위 내에서 규칙의 위반을 허용한 '허용'항으로 나뉘어져 있다[1, 3]. 본 논문에서는 원칙과 허용에서 원칙만을 연구 범위로 설정하였다.

2. 로마자 표기법의 특징과 페트리넷의 특징

2.1 로마자 표기법의 특징

2.1.1 표준 발음법과 로마자 표기법

로마자 표기법 '제 1장 표기의 기본 원칙'의 제 1항은 '국어의 로마자 표기는 국어의 표준 발음법에 따라 적는 것을 원칙으로 한다.'로 정의되어 있다[1, 2, 4, 5]. 이것은 로마자 표기를 구성하는 기본 원칙으로 표준 발음법을 따르는 것을 원칙으로 하고 있음을 밝히고 있다. 그러나 로마자 표기법의 모든 규칙이 표준 발음법과 동일한 것은 아니다. '제 2장 표기 일람'의 '[붙임]'항에서는 '된소리되기는 표기에 반영하지 않는다.'로 정의하고 있다. 제 2장의 붙임은 제 1장 1항의 기본원칙에 위배되나 '된소리되기'는 발음의 편의를 위하여 일어나는 음성 언어에서의 언어 현상이므로 표기에는 반영하지 않은 것으로 사료된다[4, 6-8]. 또한 제 1장 1항의 '[붙임 2]' 항에서는 "'ㄹ'은 모음 앞에서는 'r'로, 자음 앞이나 어말에서는 'l'로 적는다. 단, 'ㄹㄹ'은 'll'로 적는다"라는 붙임 항을 가지고 있다. 이것은 음성 언어의 규칙인 표준 발음법에는 정의 되어있지 않지만, 표기 언어인 한글을 로마자로 표기시에 나타나는 현상에 대한 표기 규칙을 나타낸 항이다.

2.1.2 로마자 표기법의 특징

〈표 1〉 개정 전후의 로마자 표기법 비교

모 음	ㅏ	ㅓ	ㅗ	ㅜ	ㅡ	ㅣ	ㅐ	ㅔ	ㅚ	ㅑ	ㅕ
개정전	a	ŏ	o	u	ŭ	i	ae	e	oe	ya	yŏ
개정안	a	eo	o	u	eu	i	ae	e	oe	ya	yeo
모 음	ㅛ	ㅠ	ㅒ	ㅖ	ㅘ	ㅙ	ㅝ	ㅞ	ㅟ	ㅢ	
개정전	yo	yu	yae	ye	wa	wae	wo	we	wi	ŭi	
개정안	yo	yu	yae	ye	wa	wea	wo	we	wi	ui	
자 음	ㄱ	ㄲ	ㅋ	ㄷ	ㄸ	ㅌ	ㅂ	ㅃ	ㅍ	ㅈ	ㅉ
개정전	k/g	kk	k'	t/d	tt	t'	p/b	pp	p'	ch/j	tch
개정안	g/k	kk	k	d/t	tt	t	b/p	pp	p	j	jj
자 음	ㅊ	ㅅ	ㅆ	ㅎ	ㅁ	ㄴ	ㅇ	ㄹ			
개정전	ch'	s/sh	ss	h	m	n	ng	r/l			
개정안	ch	s	ss	h	m	n	ng	r/l			

<표 1>은 개정전과 개정후의 한글 로마자 표기법에 따른 자모의 변환을 비교한 표이다. 반달표와 어깨점을 삭제하고 국어에 꼭 필요한 'ㄱ, ㄷ, ㅂ, ㅈ'과 'ㅋ, ㅌ, ㅍ, ㅊ'을 구별하도록 하였다[2, 4].

그러나 그 동안 관행적으로 써 오던 인명이나 회사 이름, 단체 이름은 기존 표기대로 계속 쓸 수 있도록 했다. 세계적으로 이미 널리 알려진 '삼성(SAMSUNG)'이나 '현대(HYUNDAI)'와 같은 회사 이름이나 '김(KIM)' 따위의 인명 표기 등 이미 널리 알려진 표기는 기존의 표기법과 개정된 로마자 표기법 둘 다 사용할 수 있다[1, 2].

2.1.3 음운 변동 현상에 관한 연구

로마자 표기법은 한글의 음운 변동 현상에 맞는 로마자 표기를 생성하는 규칙이다. 이를 위해서는 일차적으로 표준 발음법에 준거한 음운 변동 현상이 우선적으로 규명되어야 한다.

김재홍의 연구[9]에서는 음운 변동 과정을 구개음화, 자음축약, 연음규칙, 자음동화, 경음화, 끝소리규칙의 순서로 적용하여 음운 변동을 처리하고 있으며, 김혜순의 연구[10]에서는 한국어 음운 변동에 관한 지식을 지식베이스로 구성하여 전진 추론 방식으로 탐색해 나가며 어절의 변화가 있는 동안에는 새로운 상태를 만들고, 이 같은 과정을 반복하여 음운변화를 일으키게 하였다. 한국과학기술원의 연구[11]에서는 모음변환, ㄴ첨가, 경음화, ㅎ탈락, 절음법칙, 연음법칙, 구개음화, 대표음, 자음접변 등의 순서를 반복하여 적용시켜 음운 변동을 일으키며, 양진석의 연구[12]에서는 음운 규칙의 적용 순서를 ㄴ첨가, 구개음화, 음운축약, ㅎ변환, 자음동화, 경음화, 말음법 및 연음법칙으로 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.

2.2 페트리넷의 특징

2.2.1 페트리넷의 정의

페트리넷은 1962년 Carl A. Petri가 Bonn 대학에 제출한 박사논문 "Communication with Automata"에 처음 소개하였다. 페트리넷은 병행적(Concurrent), 비동기적(Asynchronous), 분산적(Distributed or Decentralized)으로 사상(event)이 발생하는 시스템을 표현하기 위한 수학적 모델이다. 페트리넷이 널리 사용되는 이유는 모델의 구축이 용이하고 구축된 모델을 분석하는 수학적 방법이 널리 연구되었기 때문이다[13, 15].

본 연구에서 모델링 도구로 사용하는 페트리넷은 정점이

장소(place)와 변환점(transition)으로 구성되는 이분 그래프(bipartite graph)이다. 장소는 그래프 상에 원으로, 변환점은 사각형으로 표현된다. 장소와 변환점 혹은 변환점과 장소는 유향간선(arc)으로 연결 될 수 있으며, 장소와 장소 혹은 변환점과 변환점을 연결하는 경우는 없다. 한 변환점에 대하여 입력 장소는 그 변환점으로 들어오는 유향간선의 출발점이 되는 장소들을 의미한다. 출력 장소는 그 변환점으로부터 나가는 유향간선의 끝에 있는 장소들을 의미한다. 유사한 방법으로 한 장소에 대한 입력 변환점과 출력 변환점도 정의된다[13-18].

페트리넷 모델에서 장소는 조건, 데이터, 신호(signal)나 자원 등의 이름을 나타내고 변환점은 조건이 만족되면 일어나는 사건, 데이터가 주어지면 수행되는 시간, 신호에 대한 반작용, 혹은 자원이 충분하면 수행할 수 있는 작업 등을 나타낸다. 변환점에는 토큰(token)이라 불리는 까만 점을 놓을 수 있는데 해당 장소가 표현하는 조건이 만족한 상태를 나타내거나, 신호가 들어온것 혹은 자원이 얼마나 있는가 하는것을 나타내 준다. 페트리넷의 정의는 <표 2>와 같다[13, 15].

〈표 2〉 페트리넷의 정의

페트리넷 PN은 5가지 요소로 구성된다. PN = (P, T, F, W, M_0) : 여기서 P = {$p_1, p_2, \cdots, p_m$}은 장소라는 유한 집합. T = {$t_1, t_2, \cdots, t_m$}은 변환점이라는 유한 집합. F ⊆ (P×T) ∪ (T×P)는 유향간선의 집합. W = F→{1, 2, 3, … }는 유향간선에 대입되는 가중치. M_0 : P→{0, 1, 2, 3, … }는 처음에 각 장소에 놓은 토큰의 수를 표현하며, 초기 마킹이라고 함. 단, P와 T의 교집합은 공집합이고, P와 T의 합집합은 공집합이 아님.

2.2.2 허가와 점화

페트리넷의 초기 상태를 나타는 마킹(marking)을 초기 마킹(intial marking)이라고 하며 M_0로 표기한다. 한 변환점의 입력 장소에 충분한 토큰이 놓여지면 그 변환점이 나타내는 사건이 실제로 수행 될 조건이 충족된 상태를 나타낸다. 충족된 상태란 입력 장소에 놓인 토큰의 개수가 장소와 연결된 유향간선의 가중치 W의 개수와 같거나 많은 상태를 의미한다. 이때 변환점과 연결된 유향간선이 허가(enable) 되었다고 한다. 허가된 유향간선과 연결된 변환점은 점화(firing) 가능하다[13, 15, 17].

실제 사건이 수행되는 것을 페트리넷 상에서 흉내내는 것을 그 변환점의 점화라고 한다. 한 변환점이 점화하면 입력 장소에 있는 토큰을 변환점과 연결된 유향간선의 가중치만큼 소모하고 출력 장소에 연결된 유향간선의 가중치만큼의 토큰을 놓는다. 장소-변환점 망(net)에서 역동적 성질에 대한 모의 실험을 가능하게 하여 주는 변환점 점화의 정의는 <표 3>과 같다[13].

〈표 3〉 점화(firing)의 정의2

1. 하나의 변환점 t에 대하여, 입력 장소 p가 최소한 $w(p, t)$ 만큼의 토큰을 갖고 있으면, t 는 허가되었다고 한다. 2. 허가된 t는 점화될 수도 있고, 아니 될 수도 있다. 3. t의 점화는 $w(p, t)$만큼의 토큰을 각 입력 장소 p에서 제거하고 $w(t, p)$ 만큼의 토큰을 각 출력 장소에 더하여 준다.

2.2.3 근접 행렬의 정의

페트리넷 모델의 통합과 분석을 용이하게 하기 위하여 페트리넷을 근접 행렬(incidence matrix)로 표현한다.

근접 행렬 C는 |P|×|T| 행렬이며, C의 일반항은 다음과 같이 정의한다[13-15].

$$C_{ij} = \begin{cases} 1 & \text{if } (t_j, p_i) \in F \\ -1 & \text{if } (p_i, t_j) \in F \\ 0 & otherwise \end{cases}$$

여기서 P, T는 <표 2>의 정의에 따르는 장소와 변천의 유한집합이며, F는 유향간선의 집합이다.

변환점 t_j가 점화한 후에 장소 p_i에 토큰이 놓인 상태를 1로 표기하고, 장소 p_i에서 변환점 t_j가 점화할 수 있는 조건을 충족하는 토큰이 놓여진 상태를 -1로 표시한다. 그 이외의 상태는 0으로 표시한다. 이와 같은 방법으로 페트리넷 모델을 근접 행렬로 표현할 수 있다.

2.2.4 페트리넷 모델의 근접행렬 변환

페트리넷 모델은 그래픽적인 방법으로 표기한 것이므로 컴퓨터가 처리하기 위해서는 그래픽 이미지에 대한 인식이 필요하다. 따라서 그래픽 이미지가 아닌 컴퓨터가 처리하기에 적합한 형태로의 변형이 필요하다. 근접 행렬의 정의에 따라 페트리넷 모델의 입력 장소와 변환점, 변환점과 출력 장소에 맞추어 근접 행렬로 변환한다. 페트리넷 모델을 근접 행렬로 변환하는 자세한 방법은 3장 2절에서 설명한다.

3. 로마자 표기법의 페트리넷 모델링

3.1 로마자 표기법 모델링

(그림 1)의 로마자 표기법 제 2장 제 2항 붙임1은 파열음에 대한 일반항으로 모음 앞에서의 표기와 자음앞이나 어말에서의 표기를 정하고 있다. 'ㅂ'이 '백암'에서 모음앞에 사용된 'ㅂ'은 'b'로 '호법'에서 어말에서 사용된 'ㅂ'은 'p'로 사용됨을 의미한다. 이것을 표현하는 페트리넷은 22개('ㄱ, ㄷ, ㅂ'3개, 어말 1, 자음 18)의 입력 장소와 6개('g, d, b, k, t, p')의 출력 장소, 57개의 변환점으로 구성된다.

[붙임 1] 'ㄱ, ㄷ, ㅂ'은 모음 앞에서는 'g, d, b'로, 자음 앞이나 어말에서는 'k, t, p'로 적는다.([ ] 안의 발음에 따라 표기함.)

(보기)

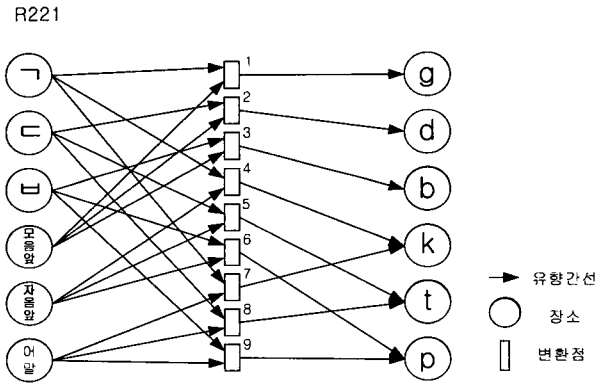
구미 Gumi	영동 Yeongdong	백암 Baegam
옥천 Okcheon	합덕 Hapdeok	호법 Hobeop
월곶[월곧] Wolgot	벚꽃[벋꼳] beotkkot	
한밭[한받] Hanbat		

(그림 1) 로마자 표기법의 예(제 2장 제 2항 [붙임 1])

(그림 2)는 (그림 1)의 로마자 표기법을 페트리넷으로 구성한 것이다. 장소 ㄱ에서 출발하는 3개의 유향간선은 각각의 변환점과 사상되며, 이 각각은 모음앞, 자음앞, 어말에서 출발한 유향간선과 함께 변환점의 입력 유향간선이 된다. 각각의 변환점이 점화 될 조건은 입력 변환점에 토큰이 놓이면 각 유향간선이 허가되고 이때 변환점의 점화 조건이 만족되며, 변환점의 점화으로 출력 장소 g, k에 토큰이 놓이게 된다[16-18].

(그림 2)의 페트리넷 모델에는 초기 마킹 M_0가 놓여지지 않은 상태이다. 장소에 놓여지는 초기 마킹은 입력되는 문자열의 형태소 분석 정보와 자모 분리 결과에 따라 동적으로 놓여진다. 예를 들어 입력 문자열로 "법"이라는 글자가 입력되었을 때 종성 'ㅂ'이 처리되는 과정은 형태소 분석과 자모 분리 결과로 '어말'과 'ㅂ'이 생성되고 각각 해당하는 장소에 초기 마킹 M_0가 놓여진다. 입력 장소 'ㅂ'과 입력 장소 '어말'에 토큰이 놓여지면 변환점 1에 연결된 유향간선이 허가되고 점화하기 위한 조건이 충족되어 변환점 221-9이 점화하면 출력 장소 'p'에 토큰이 놓여진다.

(그림 2)의 페트리넷은 페트리넷의 구성을 설명하기 위하여 간략히 그린 것으로 장소 '자음앞'은 18개의 자음으로 확장되어야 한다.

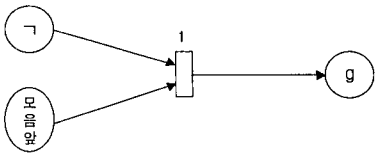


(그림 2) (그림 1)의 페트리넷 모델링

3.2 페트리넷 모델의 근접행렬 변환

3.2.1 로마자 표기법 페트리넷 모델의 근접 행렬 변환

(그림 3)은 ㄱ이 모음앞에서 g가됨을 모델링한 페트리 넷이다. <표 4>는 (그림 1)의 페트리넷을 근접행렬로 변환한 것이다.



(그림 3) (그림 1)의 모델링 일부

〈표 4〉 (그림 3)의 근접 행렬

R 221 ①	1 ②
I ㄱ ③	-1 ⑦
⑥	
I 모음앞 ④	-1
⑥	
Og ⑤	+1 ⑧

<표 4>의 근접 행렬에서 ①~⑧은 테이블을 설명하기 위한 주석 표시이며 각 주석이 표시하는 내용은 다음과 같다. ①의 R221은 로마자 표기법 제 2장, 2항, 붙임 1을 나타내는 항 번호이다. ②의 1은 1번 변환점을 의미하는 번호이다. ③과 ④의 'I ㄱ, I모음앞'에서 I는 입력(Input) 장소를 의미하고 'ㄱ, 모음앞은' 그 입력 장소에 놓일 수 있는 토큰이 'ㄱ'과 '모음앞'임을 의미한다. ⑤의 Og에서 O는 출력(Output) 장소를 의미하고, 'g'는 그 출력이 'g'임을 의미한다. ⑥은 입력장소 1과 입력장소 2를 구분하기 위한 공백이다. ⑦의 '-1'은 입력 장소의 토큰 한 개가 감소함을 의미한다. ⑧의 '+1'은 출력 장소에 하나의 토큰이 증가함을 의미한다.

〈표 5〉 로마자 표기법 2항의 근접 행렬

R 221①	1 ②	2	3	4	5	6	7	8	9
I ㄱ ③	-1 ⑦			-1			-1		
I ㄷ		-1			-1			-1	
I ㅂ			-1			-1			-1
⑥									
I 모음앞	-1	-1	-1						
I 자음앞				-1	-1	-1			
I 어말 ④							-1	-1	-1
⑥									
Og	+1 ⑧								
Od		+1							
Ob ⑤			+1						
Ok				+1			+1		
Ot					+1			+1	
Op						+1			+1

위에서 설명한 방법에 따라 (그림 2)의 페트리넷 모델을 근접 행렬로 변환하면 <표 5>와 같다. <표 5>는 이해를 돕기 위하여 간소화하여 모델링한 것으로 입력 장소 중 '자음앞'은 18개의 자음으로 확장되어야 한다. 다만 여기서는 근접 행렬의 구성을 설명하기 위하여 '자음앞'으로 나타내었다[16-18].

3.3 한글 로마자 변환을 위한 변환 테이블 생성

3.3.1 근접 행렬 통합을 위한 구조체

<표 6>의 근접 행렬의 한 행은 주석에서 설명한 ①과 연속된 ⑦의 구조를 가지게 되고, ①은 문자열로 된 정보이며, ⑦은 정수형의 +1 또는 -1의 값을 갖게 되므로 한 행의 내용을 표현하기 위하여 다음과 같은 C언어의 구조체를 사용한다.

〈표 6〉 근접 행렬을 표현하기 위한 C언어 구조체

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

<표 6>에서 선언된 구조체의 내용을 살펴보면 문자형 변수와 정수형 변수의 2개의 필드가 있다. char Condition [ConditionMax]는 <표 5>의 ③, ④, ⑤의 레이블이 기록되는 부분이다. ConditionMax는 통합될 레이블 문자열의 최대 길이를 첨자로 갖는다.

int Bit[BitMax]는 <표 5>의 ②레이블의 위치의 값을 갖는다. 즉 <표 5>의 9번 변환점의 '-1' 또는 '+1' 값은 Bit 배열의 9번째 위치에 '-1' 또는 '+1'로 기록된다. BitMax는 각 항의 변환점 수를 첨자로 갖는다. 페트리넷으로 모델링된 로마자 표기법은 두 개의 입력 장소를 갖고, 출력 장소는 로마자 표기로 1개의 출력 장소를 갖는다.

RuleTable[3][MainTableMax]는 하나의 테이블이 저장될 공간을 정의하는 부분이다. 배열의 첫 번째 첨자인 '3'은 근접 행렬이 입력 장소1, 입력 장소2, 출력 장소의 3부분으로 나누어져 있어 각각을 따로 저장하기 위해서이며, 두 번째 첨자인 'MainTableMax'는 전체 테이블의 변환점 수의 총합을 첨자로 갖는다.

3.3.2 페트리넷 모델의 통합

로마자 표기법의 각 항은 <표 6>과 같은 형태로 저장되어 있다. 각각의 파일은 하나의 항목만을 모델링한 결과의 근접행렬이므로 이것을 하나로 통합하여야 한다. 통합의 이유는 로마자 변환시 각각의 항에 대하여 모든 변환과정을 수행하여야 하므로 여러 번의 변환 과정을 거쳐야 한다. 하나의 테이블로 통합하면 입력된 단어나 문장에 대하여 형태소 분석기에서 생성한 정보와 자모 정보만으로 로마자 변환을 완료 할 수 있다.

(그림 4)테이블 통합 알고리즘의 Step 1은 입력되는 테이블의 마지막 변환점을 처리할 때까지 순서대로 Step 2에서 4를 수행한다. Step 2에서 처리할 대상 파일로부터 하나의 근접 행렬을 읽어 들여 <표 6>의 구조체 배열에 저장한다. Step 3은 저장된 배열에서 하나의 변환점과 그 변환점에 연결된 입력 장소 1, 2와 출력 장소를 읽어 와서 통합된 테이블에 이미 기록된 장소인지 비교한다. Step 4는 입력 장소의 경우 이미 기록된 장소이면 장소가 위치하는 행의 마지막 위치에 변환점 번호와 '-1'값을 기록하고, 출력 장소인 경우 '+1'을 기록한다. 통합 대상 근접 행렬 중에서 첫 번째 입력되는 근접 행렬은 Step 4의 Else에 의하여 통합된 근접 행렬이 저장되는 배열 x에 모든 장소를 추가하게 된다. 두 번째 근접 테이블부터는 배열 x의 장소와 새로 읽어들인 m의 배열을 비교하여 새로운 장소나 마킹을 추가 시켜 나가며 각각의 파일로 저장된 근접 행렬을 통합해 나간다. Step 5에서 통합된 테이블을 출력한다.

```
Input : 로마자 표기법 근접 행렬 파일
Output : 통합된 로마자 표기법의 근접 행렬 파일
START
Step 1 : 테이블 로딩 ( 화일명n, 저장될 배열명m )
Step 2 : 테이블 비교 ( 배열명m, 배열명m' )
          for i = 0 to m의 변환점 수
           m의 i번째 변환점과 변환점에 연결된 장소를 load
            for j = 0 to m'의 변환점 수
             m'의 j번째 변환점과 변환점에 연결된 장소 load
             i번째 변환점의 장소와 j번째 변환점의 장소 비교
                    일치 : 통합테이블기록 ( 변환점, 장소 )
                    불일치 : 분석 파일에 기록 ( 변환점, 장소 )
Step 3 : 통합테이블기록 ( 변환점, 장소 )
          마지막 변환점 위치 읽어옴
          기록된 장소 이름 비교
            기록된 장소명 : 마지막 변환점 뒤에 기록
            기록되지 않은 장소명 : 장소 명 기록
                                  마지막 변환점 뒤에 기록
Step 4 : 처리되지 않은 테이블이 남아 있으면 Goto Step1
Step 5 : 통합테이블 출력 ( 화일명n )
END
```

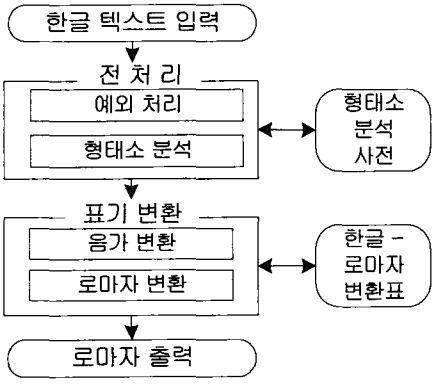
(그림 4) 테이블 통합 알고리즘

4. 구현 및 실험 결과

4.1 구현

4.1.1 전체구조

(그림 5)는 한글-로마자 변환기의 전체 구조를 나타내고 있는 그림이다. 한글 로마자 변환기의 입력은 한글 텍스트만을 가정하였다. 입력된 한글 텍스트는 예외 처리 단계와 형태소 분석의 전처리 단계를 거쳐 표기 변환 모듈로 전달된다.



(그림 5) 한글-로마자 변환 시스템 전체구조

4.1.2 전처리모듈

예외 처리 단계에서는 한글 이외의 영어, 문장 부호 및 완성되지 않은 한글 등을 예외 처리한다. 전처리 모듈의 형태소 분석기는 범용 형태소 분석기의 분석결과보다는 좁은 범위의 분석정보를 생성한다. 본 연구에서 구현하는 형태소 분석기의 기능은 음가변환과 로마자 변환에 이용되는 형태소 분석기를 위한 것으로 한정한다. 형태소 분석항목은 로마자 변환에 사용될 변환 테이블의 인덱스를 생성하기 위하여 요구되는 정보이므로 로마자 변화 테이블에 나타나는 항목들로 형태소 분석기의 분석 항목이 한정된다. 전처리기에 사용된 형태소 분석 사전은 음가 변환 및 로마자 변환과 관련된 어휘에 대한 정보만 실려있다.

형태소 분석 단계에서는 음가 변환 단계에서 필요한 형태소 분석 정보를 생성하여 표기 변환 모듈로 전달된다.

4.1.3 표기변환 모듈

표기 변환 모듈은 음가 변환 단계와 로마자 변환 단계로 구분된다. 한글은 표기와 음가(phonetic value)가 서로 다르기 때문에 표기를 그대로 로마자로 변환시킬 경우 올바른 로마자 표기를 생성할 수 없다.

음가 변환 단계에서는 한글 표준 발음법중 로마자 표기법에 해당하는 영역만을 모델링 하여 얻은 표기-음가 변환 내용과 로마자 표기법을 모델링 하여 얻은 한글-로마자 변환 표가 통합되어 있으므로 이를 이용하여 한글의 음운 변동 현상을 처리하여 입력된 텍스트를 음가로 변환시킨다.

로마자 변환 단계에서는 변환된 음가열을 자모 분해하여 한글-로마자 표기 변환표를 바탕으로 로마자 표기 규칙에 따라 로마자로 변경한다. 이때, 음가 변환과 로마자 변환은 두 개의 모듈이 각각 동작하는 것이 아니고, 통합된 표를 이용하여 인접한 두 음절 사이에서 음가 변환이 필요한 경우는 음가 변환과 동시에 로마자 변환을 수행하고 음가 변환이 필요 없는 경우는 로마자 변환만을 수행하는 것으로 이 같은 과정이 음절과 음절의 경계에서 연속적으로 일어나 표기 변환 과정을 완료한다.

4.2 실험 결과 및 고찰

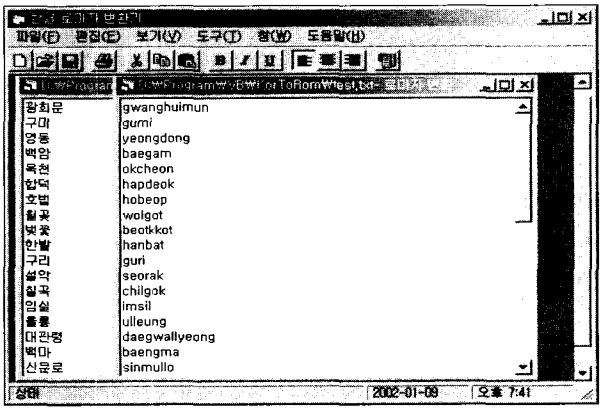
4.2.1 실험 결과

<표 7>은 실험 결과를 요약한 표이다. 10개의 예제 집합에 대하여 각 예제 집합의 변환 결과와 용례 사전의 용례와 비교한 결과 100%의 일치율을 보였다.

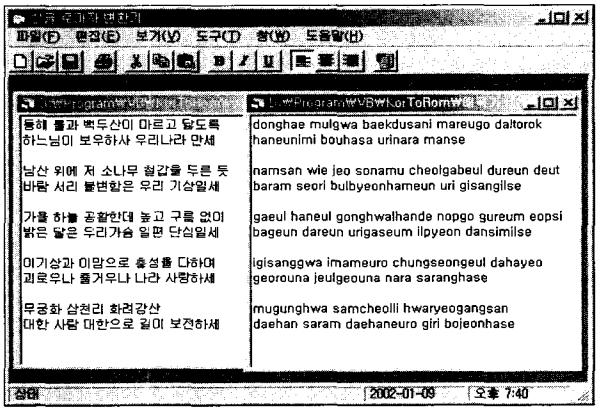
〈표 7〉 한글-로마자 표기 변환 실험 결과

실험 집합	어절수(단어)	음절수(글자)	일치율
표기법 예제	41	93	100%
애국가 가사	48	174	100%
경부선 철도역명	264	604	100%
광역시도명	16	68	100%
전국 군, 구명	253	744	100%
전국 산이름	123	639	100%
전국 강이름	39	112	100%
의, 식, 주생활 용어	35	81	100%
문화유산	42	107	100%
호남선 철도역명	62	141	100%
계	923	2,763	100%

(그림 6)은 로마자 표기법 고시본에 실려있는 예문에 대하여 실험한 결과이다. 모두 41단어 총 93글자에 대하여 실험한 결과 모두 예문의 로마자 표기와 일치함을 나타내었다.



(그림 6) 로마자 표기법 예제 실험



(그림 7) 애국가 실험 결과

(그림 7)은 애국가를 실험한 결과이다. 총 48어절 174자의 변환 결과가 로마자 표기 용례 사전의 로마자 표기와 일치함은 나타내었다.

4.2.2 검 토

본 논문에서는 개정된 한글의 로마자 표기법에 따른 로마자 표기 변환을 수행하는 시스템과 한글-로마자 표기 변환표의 생성을 위한 연구를 수행하였다. 도한 한글-로마자 표기법의 규칙 변환을 위한 자연 언어의 수학적 모델링을 위하여 동적 모델링 도구인 페트리넷을 이용하여 한국어 표준 발음법과 한글 로마자 표기법을 모델링 하여 자연어를 수학적으로 모델링하는 방안을 제시하였다.

표기 언어인 한글을 한국어로 발음하는 규칙을 정의한 한국어 표준 발음법과, 외국인이 한글을 읽는 것을 가정으로 한 국어의 로마자 표기법을 페트리넷으로 모델링하는 방안을 설명하고 예를 들어 보임으로서 자연 언어를 수학적으로 모델링하는 방안을 제시하고 그 실례를 들어 언어 현상을 수학적으로 모델링하는 방안을 검증하였다.

생성된 한글-로마자 표기 변환표를 검증하기 위하여 로마자 표기 용례 사전[5]에서 임의 추출된 10개의 테스트 집합의 923어절 2,763음절에 대하여 변환을 수행한 결과 용례 사전의 용례와 동일한 변환 결과를 나타내었다.

5. 결 론

한글-로마자 표기 변환표를 생성하기 위하여 표준 발음법과 로마자 표기법의 상관관계를 밝히고 페트리넷을 이용하여 표준 발음법과 로마자 표기법을 모델링하는 방안을 제시하였다. 동적인 페트리넷 모델을 정적으로 표현하기 위하여 각 항별 페트리넷 모델을 근접행렬로 변환하는 방안을 제시하였다.

규칙에 기반한 한글-로마자 변환을 위하여 페트리넷을 이용한 표준 발음법과 로마자 표기법의 분석을 통한 한글 로마자 표기 변환표를 생성하고, 생성된 테이블의 검증을 위하여 한글 로마자 표기 변환을 위한 윈도우 기반 응용프로그램을 작성하여 생성된 테이블을 적용하였다.

본 논문에서 작성된 한글-로마자 표기 변환표를 이용하여 한글-로마자 표기 변환표를 이용한 규칙 기반 한글-로마자 변환 시스템의 구현과 검증을 수행하여 10개의 테스트 데이터 집합의 변환을 수행한 결과, 로마자 표기법 용례 사전의 용례와 일치함을 나타내었다.

참 고 문 헌

[1] 문화관광부, "국어 로마자표기법", 문화관광부 고시 제2000-8호, 2000.

[2] 문화관광부, 국립국어연구원, "로마자표기법 이렇게 바뀌었습니다", 문화관광부, 2000.

[3] 문화교육부, "표준어 규정", 문교부 고시 제88-2호, 1988.

[4] 문화관광부, "우리말 바로알기", 문화관광부, 1998.

[5] 문화관공부, 국립국어연구원, "로마자 표기 용례 사전", 동화서적, 2001.

[6] 이희승, 안병희, "한글 맞춤법 강의", 신구문화사, 1991.

[7] 이기문외 9인, "국어 어문 규정집", 대한교과서 주식회사, 1996.

[8] 서울대학교 사범대학 국어 교육 연구소, "고등 학교 문법", 1996.

[9] 김재홍, "고품질 한국어 음성 합성을 위한 문서 음성 변환시스템", 연세대학교 대학원, 석사 학위논문, 1998.

[10] 김혜순, 변영태, 이기철, "멀티미디어를 이용한 한국어 발음 교육 시스템", 한국정보과학회논문지, Vol.20, No.1, 1993.

[11] 한국 과학 기술원, "무제한 한국어 음성 합성 시스템", 연구보고서, 1990.

[12] 양진석, 김재범, 이정현 "운율 및 길이 정보를 이용한 무제한 음성 합성기의 설계 및 구현", 정보처리학회논문지, 제3권 제5호, 1996.

[13] T. Murata, "Pertri nets : properties, analysis and applications," Proc. of the IEEE, Vol.77, No.4, pp.541-580, April, 1989.

[14] 임재걸, 이계영, 이태경, "Petri Net를 이용한 지식 표현 방법", 동국논집, 제14집, pp.169 187, 1995.

[15] 임재걸, 이계영, "페트리네트를 이용한 표준 발음법 표현", 제1회 지능기술 공동학술회의논문 및 작품 요약집, pp.64-70, 1995.

[16] 임재걸, 이계영, 김경징, "페트리넷을 이용한 표준 발음법 분석 시스템 디자인", 한국정보과학회 봄학술발표논문집, pp. 369-371, 1999.

[17] 임재걸, 이계영, 김경징, 김규식, "페트리넷을 이용한 표준 발음법 분석 시스템 구현", 정보처리학회 춘계학술발표논문집, pp.609-612, 1999.

[18] 임재걸, 이계영, 김경징, "표준 발음법 페트리넷을 이용한 음운 변환기 설계", 한국멀티미디어학회 춘계학술발표논문집, 제2권 제1호, pp.339-344, 1999.

김 경 징

e-mail : jjing@dankook.ac.kr

1998년 동국대학교 전자계산학과 졸업 (공학사)

2000년 동국대학교 대학원 전자계산학과 졸업(공학석사)

2000년~현재 단국대학교 대학원 전자컴퓨터공학과(박사과정)

관심분야 : 자연어 처리, 음성 인식/합성

최 영 규

e-mail : young@dankook.ac.kr

1994년 단국대학교 전자공학과 졸업 (공학사)

1997년 단국대학교 대학원 전자공학과 졸업 (공학석사)

2001년 단국대학교 대학원 전자공학과 (공학박사수료)

2000년~2001년 (주)패스싸인

관심분야 : 패턴인식, 인공지능, 멀티미디어 응용

이 상 범

e-mail : sbrhee@dankook.ac.kr

1974년 연세대학교 전자공학과(공학사)

1978년 서울대학교 대학원 전자공학과 (공학석사)

1986년 연세대학교 대학원 전자공학과 (공학박사)

1984년 미국 IOWA대학교 컴퓨터공학과 객원교수

1979년~1999년 단국대학교 전자·컴퓨터공학과 교수

1997년~1999년 단국대학교 교무·연구처장

1997년~현재 단국대학교 멀티미디어산업기술연구소장

2000년~현재 단국대학교 공학부 컴퓨터공학전공 교수

관심분야 : 컴퓨터구조, 패턴인식, 디지털 신호처리