한글 오토마타 개발에 관한 연구

김 인정°, 황 희융
서울대학교 컴퓨터공학과

(A study on the Hangul automata)

Kim Injeong,° Hwang Heeyeung
Seoul Natinal University Computer Engineering

< Abstract >

We have developed a simple and efficient hangul recognizing automata as a part of a graphic project. this automata accepts both English and Korean code from a Korean syllable from a stream of Korean alphabets. The automata then makes a 2 byte Hangul combination code from the recognized syllable and passes that code to the graphic system.

1. 서론

한글은 자음과 모음이 합쳐져서 하나의 음절을 이루는 구조를 가지고 있다. 한글의 최소 단위는 자소로서 24개의 기본 음소가 국문법상의 규칙에 따라 초성, 중성, 종성으로 조합되어 한글 한 음절이 형성되는데 이 음절이 실제적인 발음 단위이면서 의미 단위가 된다. 이와 같은 특성을 지닌 한글을 영어나 일본어처럼 풀어쓰기 형태로 컴퓨터에 구현한다면 이는 간단한 일이겠으나 한글의 특성을 무시한 이 방법은 바람직하지 않다. 따라서 한글의 특성을 그대로 살린 모아쓰기의 측면에서 연구, 개발되어 정착되어야 한다.

한글 처리용 기기들은 구조상 주로 풀어쓰기 형태로 한글을 입력하여 처리하고 있다. 특히 한글의 키보드 입력은 한글 자소를 하나의 건반으로 할당하여 풀어쓰기 형태로 하고 있다. 풀어쓰기 형태로 입력된 한글에 의미를 부여하기 위해 음절을 구성해야 하는데, 이 음절의 구성은 한글 모아쓰기 방법에 의해 이루어 진다.

한글의 모아쓰기는 Right Linear Grammar와 유한 오토마타(Finite Automata) 이론으로 완전히 해결 가능하다는 연구들이 발표되었다. 따라서 본 연구는 기존의 한글 오토마타들를 조사 분석하고 오토마타 이론을 이용하여 간단하고 효율적인 오토마타를 제안하고 이를 구현하였다.

2. 한글 오토마타의 입력 자소의 선정

한글 입력 자소의 선정은 한글 모아쓰기 오토마타의 복잡성과 단순성을 결정하는 요소중 하나로서, 기본 자소 24자에 어떤 자음과 어떤 모음을 추가하느냐에 따라 자소 집합이 변할 수 있다. 기본 자소 집합은 표 1과 같다.

표 1. 한글의 기본 자소

자음	ㄱ,ㄴ,ㄷ,ㄹ,ㅁ,ㅂ,ㅅ,ㅇ,ㅈ,ㅊ,ㅋ,ㅌ,ㅍ,ㅎ
모음	ㅏ, ㅑ, ㅓ, ㅕ, ㅗ, ㅛ, ㅜ, ㅠ, ㅡ, ㅣ

한글은 초성, 중성, 종성으로 구성되어 각 부분은 기본 자소 단독 혹은 자소들의 복합으로 이루어 질 수 있다. 따라서 입력 자소로 기본 자소를 선택할 경우, 초성의 쌍자음과 중성의 복모음, 종성의 복자음, 쌍자음등을 키보드로부터 입력을 여러

번 해야 하고, 음절간의 분절표가 없는 이상 음절간 구분이 모호해지는 경우가 종종 생기는 문제점을 갖고 있다. 따라서 기본 자소 24자 (쌍자음 5자(ㄲ,ㄸ,ㅃ,ㅆ,ㅉ), 수직 복모음 4자(ㅐ, ㅒ, ㅔ, ㅖ)를 합하여 33자를 선정하여 한글 모아쓰기 오토마타의 구성을 간단하게 하도록 하였다.

3. 한글 코드 체계에 따른 오토마타 구성

2 바이트 조합형과 완성형 코드 체계를 비교 분석하고 그에 따른 오토마타 구성을 설명한다.

3.1 완성형과 조합형 코드 체계의 비교 분석

한글 모아쓰기에 한글 코드 체계의 선택은 모아쓰기 오토마타에서 출력되는 코드를 의미하는 것으로 이들에 대해 비교 분석하는 과정이 필요하다. 2 바이트 한글 코드 체계를 조합형과 완성형의 2 가지로 분류할 수 있다. 2 바이트 조합형 한글 코드 체계는 한글 한 글자를 그림 1과 같이 초성, 중성, 종성 5 비트씩 나누어서 각 자소마다 코드 값을 부여한다. 이와 같이 각 자소마다 코드 의미를 부여할 경우에는 모든 한글 음절을 나타낼 수가 있고, 각 음절의 자소를 구분할 수가 있으므로 한글 자연언어 처리시 조사 및 어미 변화 등의 처리가 매우 손 쉽고, 자소 단위의 한글 수정도 가능하다. 또, 한글에 대해 순차 배열할 때에도 매우 용이하게 할 수 있고 한자 영역으로 7,500~8,000자까지 수용할 수 있다.

2 바이트 완성형 코드 체계는 그림 2와 같이 구성되었는데 사용 빈도가 높은 한글 2350자의 음절을 선정해 '가나다' 순으로 정렬된 한글 한 음절 마다 코드를 연속적으로 부여한 것으로 모든 한글 한 음절을 나타낼 수가 없다. 따라서 새로운 한글을 만들 때 사용자 정의 영역에서 별도로 추가해야 하기 때문에 한글 순차 배열시에 어려움이 따른다. 또 각 음절을 자소 단위로 구분할 수 없으므로 자소 단위의 한글 처리나 자연 언어의 처리시에 매우 복잡한 과정을 필요로 하거나 거의 불가능 해진다. 그러나 한글 코드를 보다 압축적으로 표현할 수가 있는 장점을 가지고 있다. 한자의 경우 6,000자 정도의 영역을 가지고 있다.

그림 1. 2 바이트 조합형 코드의 구성

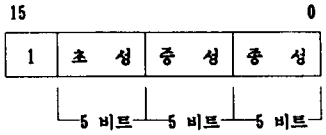
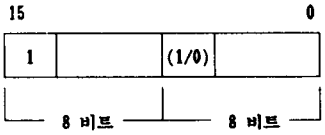


그림 2. 2 바이트 완성형 코드의 구성



두 개의 코드 체계가 모두 영문과 한글의 크기 비율이 1:2 이고 문자 자형의 크기는 조합형일 경우에는 자소별 자형을 가지고 있으면 되지만 완성형일 경우에는 모든 한글에 대한 자형을 가지고 있어야 되므로 조합형에 비해 훨씬 많은 메모리를 필요로 한다.

3.2 한글 코드 발생 방법

조합형의 경우에는 한글 자소 입력에 대해 모아쓰기를 한 것이 바로 사용되는 코드이다. 그러나 완성형의 경우에는 한글 자소 입력시에 모아쓰기를 하여 조합형 코드의 형식으로 색인 값을 만든 다음, 이에 대해 코드 변환 작업을 거친 후에 완성형 코드를 만든다. 따라서 효율과 수행 속도가 매우 뒤떨어 지게 된다. 따라서 본 연구에서 개발한 오토마타는 2 바이트 조합형 코드를 사용한다.

4. 한글 모아쓰기 오토마타의 구현

한글의 초성, 중성, 종성의 3 부분으로 이루어져 있으며, 초성은 자음과 쌍자음, 중성은 단모음과 복모음, 종성은 받침 없는 경우를 포함하여 단자음과 복자음으로 구성 된다. 이들 각 부분은 표 2와 같이 24자의 기본 자모가 합하여져서 이루어진다.

키보드로부터 키가 눌러졌을 때 한글 모드와 영문 모드가 공존하는 가운데 영문 모드시에는 그대로 텍스트 모드를 흉내내면서 시스템에 바로 아스키 코드 값이 보내져서 화면 출력이 되고, 한글 모드

일 경우에는 키로부터의 값들을 계속 조합하여 한 음절이 완성되었을 때 그에 해당하는 2 바이트 조합형 코드를 만들어 시스템에 보내도록하는 입력

표 2. 한글 초성, 중성, 종성의 구성

초성	중성	종성
ㄱ ㄴ ㄷ ㄹ ㅁ ㅂ ㅅ ㅇ ㅈ ㅊ ㅋ ㅌ ㅍ ㅎ ㄲ ㄸ ㅃ ㅆ ㅉ	ㅏ ㅑ ㅓ ㅕ ㅗ ㅛ ㅜ ㅠ ㅡ ㅣ ㅐ ㅒ ㅔ ㅖ ㅘ ㅙ ㅚ ㅝ ㅞ ㅟ ㅢ	Fill ㄱ ㄴ ㄷ ㄹ ㅁ ㅂ ㅅ ㅇ ㅈ ㅊ ㅋ ㅌ ㅍ ㅎ ㄲ ㄸ ㅃ ㅆ ㅉ ㅏ ㅑ ㅓ ㅕ ㅗ ㅛ ㅜ ㅠ ㅡ ㅣ ㅐ ㅒ ㅔ ㅖ ㅘ ㅙ ㅚ ㅝ ㅞ ㅟ ㅢ

용 오토마타로서 그 전체 루틴의 구성은 그림 3에 설명되어 있다. 다시 말하면, 키보드 입력을 키보드 인터럽트 16을 이용하여 그의 스캔코드와 아스키코드 값을 취하여 영문과 한글 모드를 구분하고 한글 모드시 아스키 값을 한글을 위한 입시 코드로 변환시킨다. 변환된 입시 코드를가지고 오토마타 루틴을 호출하여 오토마타 상태로 들어가게 한다. 그림 4에 있는 오토마타 상태 전이도가 한글 모드시에 수행되는 한글 모아쓰기 루틴을 설명한 것이다. 그림 4에서 보는 것처럼 초기 상태 S0를 포함하여 초성, 중성, 종성에 각각 S1, S2, S3 상태를 부여하여 최소화된 오토마타 상태에서 한글 한 음절을 완성하는 것을 보여 준다. 완성된 한글은 상태도에서 *가 이를 의미한다. 또한 이 오토마타에서는 한글 입력 도중 한글이 아닌 키가 눌러졌을때에 특수 문자 처리로서 영문모드와 마찬가지로 동작한다.

5. 결론

앞 절에서 기술한 대로 구성된 이 오토마타는 간단하고 명료하며 효율성을 고려하여 작성하고자 하였다. 따라서 33자의 입력 자소를 가지고 모호함이 없이 한글 2 바이트 코드를 만들어 내고, 오토마타의 상태도를 보면 누구나 알 수 있듯이 아주 간단하고 축소된 형태를 가지고 있음을 볼 수 있다.

이 오토마타는 누구나 쉽게 오토마타를 구성하고 여기에 기능을 추가하여 확장된 오토마타를 구현할 수 있다고 생각된다.

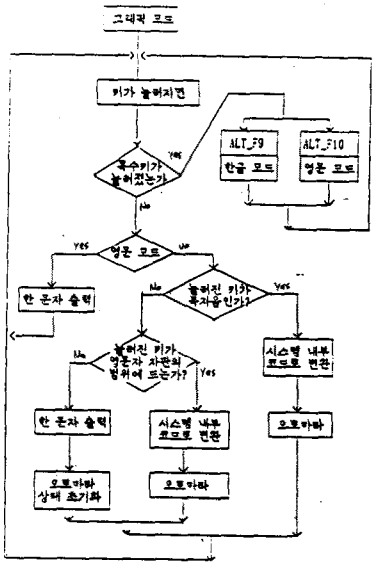


그림 3. 한글 오토마타 전체 흐름도

6. 참고 문헌

[1] 한글 표준 연구소, "한글 정보 처리 표준화 연구", 1988

[2] 마이크로 소프트웨어, "한글 표준 안 코드에 대하여", 1987.11

[3] 퍼스널 컴퓨터, "IBM PC ROM BIOS의 분석", 1989.1

[4] 마이크로 소프트웨어, "한글 도깨비", 1989.4

[5] 퍼스널 컴퓨터, "한글의 표현 방식", 1989.3

[6] IBM, "Techical Reference Personal computer AT"

[7] 김병기, "한글 문자의 지적 처리에 관한 연구", 기계번역

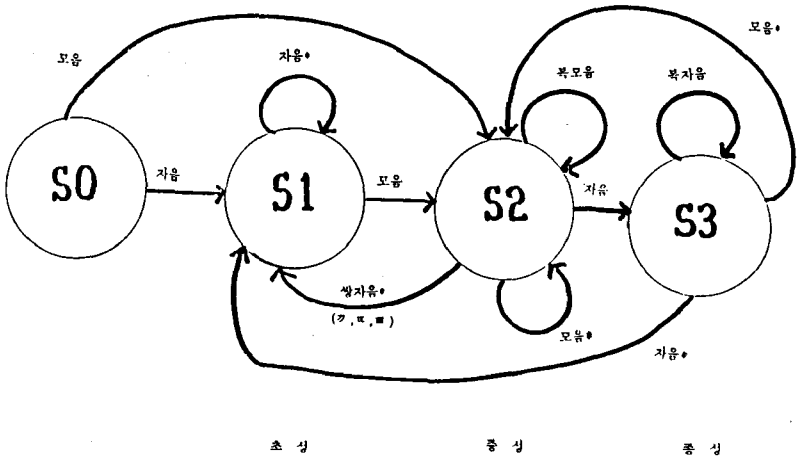


그림 4. 한글 모아쓰기 오토마타 상태도