CS322 프로젝트 #3

20150112 김동성

언어는 Python 3.4.4를 사용하였고 코드 작성에 사용한 IDE의 정보는 다음과 같다.

PyCharm Community Edition 2016.2.3

Build #PC-162.1967.10, built on September 7, 2016

JRE: 1.8.0\_112-release-b343 x86

JVM: OpenJDK Server VM by JetBrains s.r.o

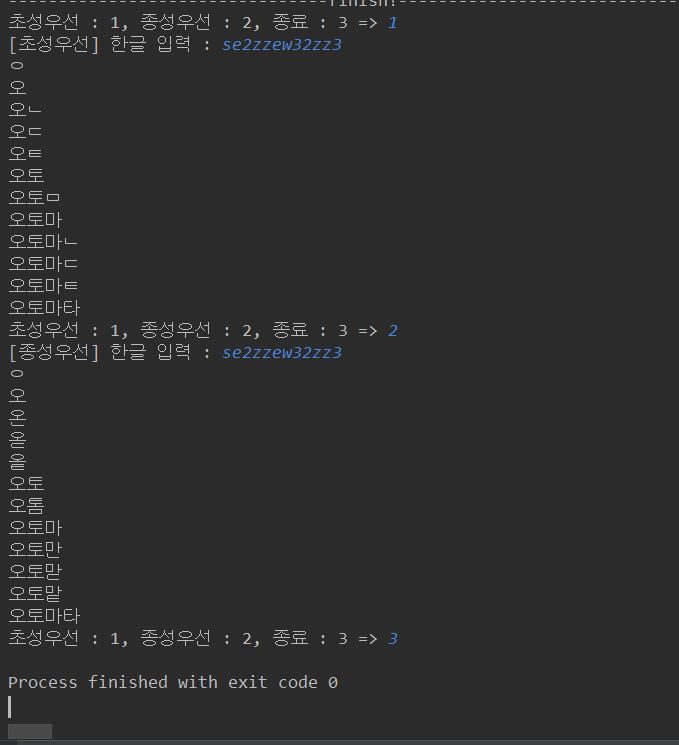
사용한 PLY의 버전은 3.9이다.

**프로젝트 설명**

이 프로젝트는 RE->m-DFA 변환 프로그램(본 프로젝트2)과 mealy machine simulator(예비프로젝트 1-2)을 이용해 3x4 자판을 위한 한글 모아쓰기 오토마타를 만드는 프로젝트이다. 여기서는 KT나랏글 자판의 입력 방식을 이용하였다.

* 한글 입력 방식

1. KT나랏글 자판 입력 방식을 사용했으며, 이 프로그램에서 입력받는 symbol은 123 / qwe / asd / zxc이다. (각각 ㄱㄴㅏ / ㄹㅁㅗ / ㅅㅇㅣ / 획추가 ㅡ 쌍자음)
2. 순환은 허용하지 않는다. 즉, 1333(ㄱㅏㅏㅏ)는 ‘가’가 아니고 잘못된 입력으로 처리된다.
3. Backspace에 해당하는 input symbol은 <이다. Backspace를 해 줄 경우 단모음 또는 단자음만을 제거하며, 출력되는 string의 표기 방식은 한글 입력 전에 선택했던 표기 방식을 그대로 따른다.

한글 string 입출력 방식은 다음과 같이 코드 실행 후 출력하고자 하는 한글 string을 3x4 KT나랏글 자판 형태로 입력한 후, 콘솔 창에 최종 결과가 출력되게 한다. 이 코드에서는 초성우선/종성우선 방식이 잘 작동하는지 확인하기 위해 각 단계를 모두 출력하게 하였다.

**코드 설명**

Line 537 까지의 코드는 lex에서 CHAR 토큰에 해당하는 symbol을 [123qweasdzxc]로 제한한 것을 제외하면 본 프로젝트2에 이용했던 코드와 동일하므로 설명은 생략한다.

* 함수 make\_mealy

RE->m-DFA 변환기를 거쳐 생성된 m-DFA의 정보를 바탕으로 mealy machine을 만들어준다. 그리고 어떤 자모가 출력되었을 때 그것이 **확실히** 초성일 때의 state와 중성일 때의 state를 각각 state\_cho와 state\_jung에 모아서 저장한다.

cf) 확실히 초성인 경우 : “ㄱㅏㄱ”에서 마지막 ㄱ이 입력되었을 때는 이 ㄱ이 종성이될지 초성이 될지 모른다. 그러나 “ㄱㅏㄱㄴ” 에서 ㄴ은 확실히 초성이다.

* 함수 hangul

본 프로젝트 1에서 사용했던 코드와 유사하다. 인자로 string을 받는데, 이 string은 한글 자모로 구성된 스트링이다 (ex. “ㅇㅗㅌ”). 이 string을 구성하는 각각의 자모를 확인하고 모아쓸 수 있는 경우에는 모아쓴 형태를 반환하고 그렇지 않은 경우에는 그대로 반환한다. 단, 모아쓸 수 없는 경우라 함은 어떤 조합이든 모두 처리해준다는 의미가 아니고 올바른 한글 string 입력 과정 중에 나올 수 있는 경우로 제한한다.

Ex. 모아쓸 수 있는 경우 : “ㄱㅗㅏ” -> 과

모아쓸 수 없는 경우 : “ㄱㅗㅏㄹㅇ” -> 괄ㅇ (‘괋’을 입력하는 중에 거쳐가는 과정)

이 때, 인자로 받는 string을 구성하는 자모는 단모음, 단자음 및 ㅐ, ㅔ, ㅒ, ㅖ를 포함한다. 본 프로젝트 1에서는 키보드 자판을 바탕으로 한글 모아쓰기 오토마타를 만들었고, 그 과정에서 쓰인 코드를 재활용한 것이기 때문에 이러한 방식을 그대로 사용하게 되었다. 이외의 겹모음은 각 단모음을 따로 분리한 형태로 들어가게 된다. (Ex. “ㄱㅐ”-> 개, “ㄱㅗㅏ”-> 과)

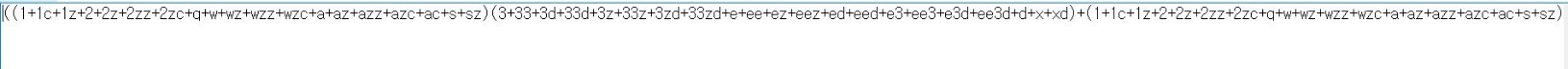
* 함수 output1, output2

각각 초성우선 방식과 종성우선 방식으로 한글 string을 출력해주는 함수이다. 이 함수에 대한 설명은 output1 코드에 주석으로 남겨놓았다. Output2 역시 큰 틀은 동일하므로 output1에 대한 설명으로도 이해하기 충분할 것이다.

설명을 하지 않은 함수들은 별도의 설명이 필요없거나 본 프로젝트 1, 2의 코드를 재활용한 것이므로 설명을 생략한다.

**입출력 설명**

1. 먼저 3x4 KT나랏글 자판을 바탕으로 한 input string을 표현하는 regular expression을 텍스트 파일(re.txt)의 형태로 입력한다.



(1+1c+1z+2+2z+2zz+2zc+q+w+wz+wzz+wzc+a+az+azz+azc+ac+s+sz) : 초성

(3+33+3d+33d+3z+33z+3zd+33zd+e+ee+ez+eez+ed+eed+e3+ee3+e3d+ee3d+d+x+xd) : 중성

(1+1c+1z+1a+2+2z+2zz+2az+2sz+q+q1+qw+qwz+qa+qwzz+qsz+q2zz+w+wz+wzz+wza+a+az+azz+ac+s+sz) : 종성이 될 수 있는 조합

1. RE->m-DFA 변환 결과로 생성된 m-DFA는 re2mdfa.txt 파일로 확인할 수 있다. 다만 make\_mealy에서 m-DFA를 바탕으로 생성하는 mealy machine은 텍스트 파일로 출력하지 않는다.
2. 프로그램이 실행되면, 본 문서 1페이지에 첨부한 사진처럼 콘솔 창에 초성우선, 종성우선 혹은 종료를 선택하도록 하는 문구가 뜬다. 초성우선 또는 종성우선을 선택하면 콘솔 창에 한글 string을 3x4 KT 나랏글 자판 방식으로 입력할 수 있다. 이때, 반드시 최종 결과가 올바른 한글 string또는 올바른 한글 string을 생성하는 도중에 나올 수 있는 형태가 되어야 한다. 종료를 선택하면 프로그램을 종료한다.