

# CS322 프로젝트 #1

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

## 프로젝트 설명

이 프로그램은 mealy machine을 이용해 한글 string을 키보드 자판 상에서 한영변환하여 입력하면 그에 대응되는 한글 string을 출력하는 프로그램이다 (ex. dhxhakxk -> 오토마타). 이 때, 한글 입력방식은 모음 이후 자음이 왔을 때 초성으로 간주하는 초성우선, 가능한 한 종성으로 간주하는 종성우선 두 가지가 있는데 사용자가 1을 입력하면 초성우선, 2를 입력하면 종성우선으로 한글 string을 출력하게 된다. 또한, 3을 입력하면 프로그램을 종료한다. 이 때, 출력은 과정을 확인하기 쉽도록 최종 결과만이 아닌 각 단계에서의 출력 결과를 모두 보이도록 하였다.

아래 스크린샷은 실행 방법의 예시이다.

```
C:\Python34\python.exe C:/Users/m2ucr/PycharmProjects/CS322/main-project#1.py
초성우선 : 1, 종성우선 : 2, 종료 : 3 =>
```

```
C:\Python34\python.exe C:/Users/m2ucr/PycharmProjects/CS322/main-project#1.py
초성우선 : 1, 종성우선 : 2, 종료 : 3 => 1
한글 입력 : goakfr|
```

```
C:\Python34\python.exe C:/Users/m2ucr/PycharmProjects/CS322/main-project#1.py
초성우선 : 1, 종성우선 : 2, 종료 : 3 => 1
한글 입력 : goakfr
ㅎ
해
해ㅁ
해ㅁㅁ
해ㅁㅁㅁ
해ㅁㅁㅁㅁ
|초성우선 : 1, 종성우선 : 2, 종료 : 3 =>
```

## 코드 설명

함수 내 각각의 코드에 대한 설명은 주석으로 남겨놓았으니 여기서는 각각의 클래스 또는 함수가 어떤 역할을 하는지를 간단히 소개하겠다.

- 클래스 Mealy

Mealy machine을 입력 받아 사용할 때 쓰일 클래스이다.

- 함수 make\_mealy

Mealy machine을 입력 받고 클래스 Mealy인 객체를 만들어 그 정보를 저장한다.

이 때 입력 받는 Mealy machine의 정보 형식은 예비 프로젝트 1-2에서와 같은 형식의 txt파일 입력으로 한다. 또한, output function에서 output은 키보드 자판에서 해당 영문자에 해당되는 한글자모이다

- 함수 hangul

인자로 전달된 한글자모 string을 모아쓰기로 바꾸어준다 (ex. 모ㅏㄹ -> 말).

- 함수 output1

초성우선으로 output을 출력할 때 사용되는 함수이다.

- 함수 output2

종성우선으로 output을 출력할 때 사용되는 함수이다.

- 함수 run

출력방법/종료 및 영어 string을 입력 받고 그에 맞는 출력이 나오게 하는 함수이다.

### 입출력 예시

- 초성 우선

```
C:\Python34\python.exe C:/Users/m2ucr/PycharmProjects/CS322/main-project#1.py
초성우선 : 1, 종성우선 : 2, 종료 : 3 => 1
한글 입력 : tjsaudgkrp gkfg<x<<x<<<ekRrhdlTdjTek
ㅅ
ㅅ
ㅅㄴ
ㅅㅇ
ㅅ머
ㅅ머ㅇ
ㅅ명ㅎ
ㅅ명하
ㅅ명하ㄱ
ㅅ명하게
ㅅ명하게
ㅅ명하게 ㅎ
ㅅ명하게 하
ㅅ명하게 하ㅁ
ㅅ명하게 할ㅎ
ㅅ명하게 할
ㅅ명하게 할ㅁ
ㅅ명하게 할
ㅅ명하게 하
ㅅ명하게 하ㅁ
ㅅ명하게 하
ㅅ명하게 ㅎ
ㅅ명하게 ㄷ
ㅅ명하게 다
ㅅ명하게 다ㄱ
ㅅ명하게 닥ㄱ
ㅅ명하게 닥고
ㅅ명하게 닥고ㅇ
ㅅ명하게 닥고이
ㅅ명하게 닥고이ㅅ
ㅅ명하게 닥고있ㅇ
ㅅ명하게 닥고있어
ㅅ명하게 닥고있어ㅅ
ㅅ명하게 닥고있었ㄷ
ㅅ명하게 닥고있었다
```

- 종성 우선

```
C:\Python34\python.exe C:/Users/m2ucr/PycharmProjects/CS322/main-project#1.py
초성우선 : 1, 종성우선 : 2, 종료 : 3 => 2
한글 입력 : tjsaudgkrp gkfg<x<<x<<<ekRrhdLTdjTek
ㅅ
서
선
선ㅁ
선며
선명
선명ㅎ
선명하
선명학
선명하게
선명하게
선명하게 ㅎ
선명하게 하
선명하게 할
선명하게 할
선명하게 할
선명하게 할
선명하게 할
선명하게 하
선명하게 할
선명하게 하
선명하게 ㅎ
선명하게 ㄷ
선명하게 다
선명하게 닥
선명하게 닥ㄱ
선명하게 닥고
선명하게 닥공
선명하게 닥고이
선명하게 닥고있
선명하게 닥고있ㅇ
선명하게 닥고있어
선명하게 닥고있었
선명하게 닥고있었ㄷ
선명하게 닥고있었다
```

## 각 방식의 장단점

기능을 코드로 구현할 때 느꼈던 장단점은 다음과 같다.

- 초성우선

입력하는 낱말이 받침이 없는 경우 초성우선 방식이 효율적이다. 예를 들어, ‘하마’를 입력할 경우, 하 뒤에 ‘ㄹ’이 왔을 때 그냥 뒤에 이어 붙이고 그 다음 ‘ㅏ’가 오면 그대로 ‘ㄹ’과 이으면 된다. 하지만 종성우선일 경우에는 함 이후 ‘ㅏ’가 오면 함에서 하를 분리시키고 ‘ㄹ’과 ‘ㅏ’를 다시 결합시키는 과정을 거쳐야 하기 때문에 작업이 더 많아진다. 반대로 낱말에 받침이 있을 때는 초성우선이 비효율적이게 된다.

- 종성우선

입력하는 낱말이 받침이 있는 경우 종성우선 방식이 효율적이다. 예를 들어, ‘닭장’과 같은 string을 입력할 경우, 모음 뒤에 자음이 오면 받침이 될 수 있는 대로 모두 순서대로 종성에 위치시키고 더 이상 자음이 있어도 받침이 될 수 없으면 다음 초성으로 보내기 때문에 자음이 왔을 때 받침이 될 수 있는지 없는지만 고려하면 된다. 그러나 초성우선일 경우에는 모음 이후 자음이 왔을 때 받침이 되든 아니든 일단 맨 끝으로 보낸 뒤, 그 뒤에 들어오는 자모에 따라 받침을 만들지 아닐지를 따로 판별해주어야 한다. 따라서 종성이 될 수 있는 자음들을 바로바로 종성으로 보내는 종성우선 방식에 비해 초성우선 방식이 비효율적이게 된다. 반대로 낱말에 받침이 없을 때는 종성우선이 비효율적이게 된다.

또한, 종성우선 방식일 경우 어떤 string을 입력 하는 도중에 그 string이 무엇인지 추측이 어려울 수 있다. 예를 들어, ‘오토마타’를 입력할 때 ‘오’까지 입력하면 가능한 string중에 ‘오토마타’가 존재할 수 있지만 ‘올’까지 입력되면 가능한 string은 ‘올’으로 시작하는 string으로 바뀌게 된다. 초성우선 방식으로 입력할 경우 이런 문제가 발생하지 않는다. 물론, 가능한 string의 목록에 종성우선 방식을 고려해 ‘올’까지만 적어도 ‘올’으로 시작하는 string과 ‘오’로 시작하는 string으로 모두 포함시키면 해결이 되겠지만 그럴 경우 string의 개수가 많아져 다소 비효율적이게 된다. 검색창에서 연관 검색어를 출력할 때 이런 점이 문제가 될 수 있을 것이다.