본프로젝트 3 3×4 한글자판을 위한 한글모아쓰기 오토마타

20160389 원종하

1. 프로그램 제작 환경

OS: Windows 10 Home 64bit

언어: Python 3.6.1

PLY(Python Lex-Yacc): ver. 3.10

Notepad++ v7.5.1로 작성, 참고한 천지인 자판은 iOS 11.1.2의 10키 키보드이다.

2. 실행 방법

동일 폴더 내에 RE.txt, eNFA2mDFA.py, RE2eNFA.py, util.py가 있을 때, 3x4KAuto.py를 실행한다.

3. 프로그램 설명

본 프로젝트 2 정규식 to m-DFA 변환기와 본 프로젝트 1 한글오토마타 Mealy/Moore Machine을 응용한 프로그램을 이용하여 3*4 천지인 자판에 맞는 문자열을 입력하면 문자열이 완성되기까지의 과정과, 그 결과가 출력된다.

천지인 자판에 대응되는 입력은 다음과 같다.

1(ㅣ)	2(ㆍ)	3(ㅡ)
4(ㄱ ㅋ | ㄲ)	5(ㄴ ㄹ)	6(ㄷ ㅌ | ㄸ)
7(ㅂ ㅍ | ㅃ)	8(ㅅ ㅎ | ㅆ)	9(ㅈ ㅊ | ㅉ)
*(→, 글자 넘김)	0(ㅇ ㅁ)	#(Backspace)

iOS 10키 키보드의 # 자리에서는 원래 ".,?!"을 입력할 수 있으나, 사용하지 않기에 Backspace 자리로 대체하였다. 또한, 쌍아래아('ㆍㆍ')는 iOS 출력화면에서 ':'와 같은 모습으로 표기되나, 편의성을 위하여 'ㆍㆍ'로 출력한다. 공백은 입력 시 공백을 입력한다.

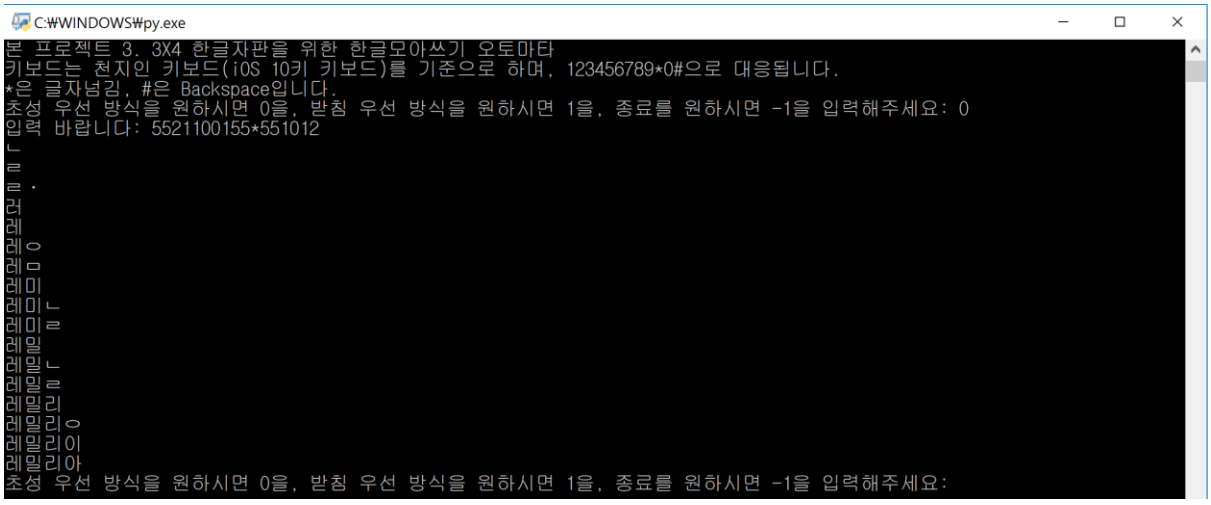
Ex)

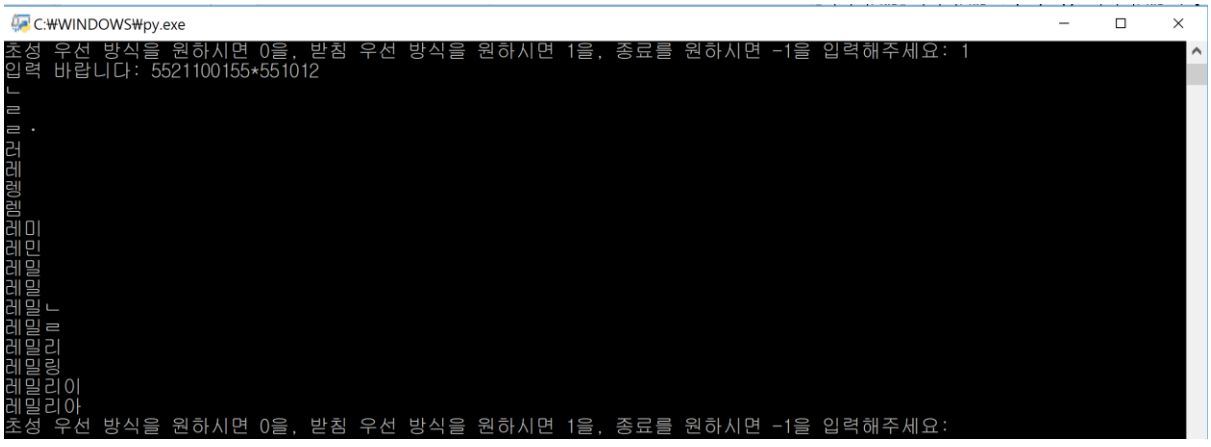
입력: 5521100155*551012 (레밀리아)	
초성 우선	받침 우선
ㄴ	ㄴ
ㄹ	ㄹ

ㄹ・	ㄹ・
러	러
레	레
레ㅇ	렝
레ㅁ	렘
레미	레미
레미ㄴ	레민
레미ㄹ	레밀
레밀	레밀
레밀ㄴ	레밀ㄴ
레밀ㄹ	레밀ㄹ
레밀리	레밀리
레밀리ㅇ	레밀링
레밀리이	레밀리이
레밀리아	레밀리아

입력 전 약간의 설명과 함께, 초성 우선 출력과 받침 우선 출력을 선택할 수 있다.

아래는 위의 예시를 나타낸 스크린샷이다.





기본적으로 글자는 자음 + 모음, 혹은 자음 + 모음 + 자음의 조합이며, 자음이 존재하지 않는 모음만으로 만들어진 글자(ex) ㅠㅠ)는 엄밀히 말해 하나의 한글 글자라 볼 수 없지만, 입력할 수 있기에 오토마타에 포함시켰다. 따라서 이번 프로그램의 정규식은 초성 오토마타를 A, 중성 오토마타를 B, 종성 오토마타를 C라고 했을 때

$$(\mathrm{B} + \mathrm{AB} + \mathrm{ABC})^*$$

가 된다. 따라서 입력되는 정규식은 다음과 같다.

(((1+3)(2+22)(22)*)+((2+22)(22)*(1+3))+3+1+(1(2+22)(22)*1)+((2+22)(22)*11)+(2(22)*31(()+(2(22)*(()+1))))+(322(22)*(1+11))+(3((2(22)*)+())1))+((((4+44+444)(444)*)+((5+55)(55)*)+((6+66+666)(666)*)+((7+77+777)(777)*)+((8+88+888)(888)*)+((9+99+999)(999)*)+((0+00)(00)*))(((1+3)(2+22)(22)*)+((2+22)(22)*(1+3))+3+1+(1(2+22)(22)*1)+((2+22)(22)*11)+(2(22)*31(()+(2(22)*(()+1))))+(322(22)*(1+11))+(3((2(22)*)+())1))+(((4+44+444)(444)*)+((5+55)(55)*)+((6+66+666)(666)*)+((7+77+777)(777)*)+((8+88+888)(888)*)+((9+99+999)(999)*)+((0+00)(00)*))(((1+3)(2+22)(22)*)+((2+22)(22)*(1+3))+3+1+(1(2+22)(22)*1)+((2+22)(22)*11)+(2(22)*31(()+(2(22)*(()+1))))+(322(22)*(1+11))+(3((2(22)*)+())1))(((4+44+444)(444)*)+(4(444)*8(888)*)+((5+55)(55)*)+(5(55)*((9(999)*)+(88(888)*)))+(55(55)*((4(444)*)+(00(00)*)+((7+77)(777)*)+((8+88)(888)*)+(66(666)*)))+((6+66)(666)*)+((7+77)(777)*)+(7(777)*8(888)*)+((8+88+888)(888)*)+((9+99)(999)*)+((0+00)(00)*)))*

이 정규식을 본 프로젝트 2 정규식 to m-DFA 변환기를 이용하여 m-DFA를 생성한다. 이 m-DFA를 바탕으로, Mealy Machine을 생성한다. 생성한 Mealy Machine의 Q, δ function, q_0은 m-DFA와 동일하며 Π는 천지인 자판의 문자(ㄱ, ㄴ, ㄷ, ...) Σ는 m-DFA의 $\Sigma = \{0\sim9\}$에서 $\{*, \#, ' '\}$을 추가했으며, λ function은 정의역은 δ function과 동일하며 치역은 $\delta(q, c)$에 대응되는 천지인 자판 문자이다 $(\in \Pi)$.

기본적인 틀은 본 프로젝트 1 한글오토마타 Mealy/Moore Machine을 바탕으로 하였으나, 그 때와 같이 DFA state transition마다 함수를 지정해 주기에는 어려웠기에, 조건문을 통해 글자를 생성하였다. 조건은 크게 6가지로, Σ에 존재하지 않는 입력을 하였을 때, *(→), #(backspace), 공백을 입력하였을 때, $\delta(s, c)$가 존재할 때, 그 이외의 경우일 때이다. 자세한 설명은 4. 코드 설명과, .py 파일 내의 주석을 참조.

iOS 휴대폰 문자열에서의 backspace는 컴퓨터 문자열에서의 backspace와 사뭇 다른 작동을 하였다. 컴퓨터에서의 backspace 명령은 아직 완성되지 않은 한 글자의 요소들을 다 지울 시 다음 backspace 명령은 한 글자 전체를 모두 지우지만, 휴대폰 문자열에서의 backspace 명령은 *(→), 혹은 공백을 입력하지 않았다면 backspace 명령은 한 글자 전체를 지우지 않고 요소들을 하나씩 지워간다. 예를 들어, '오토마타'까지 입력 후 컴퓨터에서는 backspace를 5번 누르면 글자가 모두 지워지지만, 휴대폰에서는 12번 눌러야 글자가 모두 지워진다. 즉, *(→) 혹은 공백을 입력하기 전에는 아직 버퍼에 존재하는 것이다. 물론, 글자 하나를 완성할 때마다 *(→) 명령을 한다면 backspace를 4번 누를 시 글자가 모두 지워지게 된다. 또한, 'ㅣ', 'ㆍ', 'ㅡ'를 합성해서 만든 모음과 겹자음(겹받침)들은 backspace 명령에 대해 합성 직전으로 돌아간다 (ex) ㅐ→ㅏ, ㅙ→ㅘ, ㄳ→ㄱ). 이 합성 모음은 쌍아래아('ㆍㆍ')를 포함한다. 단, 쌍자음과 같이 같은 키보드를 여러 번 눌러 만들어진 자음은 backspace 명령으로 즉시 삭제된다 (ex) 444#(ㄲ)→()).

4. 코드 설명

1) RE2eNFA.py, eNFA2mDFA.py

변화 부분만 기술한다.

RE2eNFA.py: t_LETTER의 범위를 0~9로 설정, 실행 파트를 RE2eNFA 함수로 대체

eNFA2mDFA.py: Dead state의 이름을 q_n 형태가 아닌 'DEAD'로 설정.

2) util.py

first, **second**, **third** : 각각 초성, 중성, 종성의 string list이다.

vowel, cons: 각각 천지인 자판의 모음, 자음의 string list이다.

ARAEA: 아래아와 쌍아래아가 있는 string list이다. '・'는 '.'로 대체하였다.

CC: 겹받침을 이루는 자음 요소 둘의 third list에서의 index tuple을 key로, 겹받침의 third list에서의 index를 value로 하는 dictionary이다.

MealyM: Mealy Machine class. 6개의 객체변수를 가지고 있다.

메서드	설명
__init__	Qset(Q-set): 상태들의 유한 집합 (set) Sset(Sigma-set): 입력문자들의 유한 집합 (set) Pset(Pi-set): 출력문자들의 유한 집합 (set) ddic(delta-dictionary): 상태변화함수. (qX, sX) tuple을 key로, qY를 value로 하는 dictionary (dict) ldic(lambda-dictionary): 출력함수. (qX, sX) tuple을 key로, pX를 value로 하는 dictionary (dict) qzr(q-zero): 초기 상태 (str) parameter를 그대로 입력한다.

make(buffer, flag): buffer로 string을 만드는 함수. Lettermake를 이용하여 buffer를 string으로 변환하여 반환한다. 글자를 계산하는 기준은 모음의 index이다. 자세한 코드 설명은 주석 참조.

makecheck(buf, flag): buf[:idx]가 한 '글자'로 완성될 때, 최대 idx 값을 찾는 함수. flag가 True일 땐 첫 실행 파트 작업을, False일 땐 반복 실행 파트 작업을 한다. 자세한 코드 설명은 주석 참조.

Lettermake(buffer, flag = 1): buffer로 한 글자, 혹은 한 글자 + 자음 (type: string)을 만드는 함수. buffer의 length, flag(초성우선/받침우선), buffer[n]이 초/중/종성 중 무엇인가 등을 기준으로 생성한다. 자세한 코드 설명은 주석 참조.

c_cycle(base, add): 자음순환함수. 천지인 자판에서는 7개의 기본 자음 자판을 여러 번 입력하는 작업을 통해 비슷한 계열의 자음을 생성하는데, 이를 실행하는 함수이다. base-add가 같은 계열에 존재하지 않는다면, None을 반환한다.

v_change(base, add): 모음합성함수. 천지인 자판에서는 3개의 기본 모음 자판을 통해 모음을 생성하는데, 이를 실행하는 함수이다. base-add로 모음이 생성되지 않는 case라면, None을 반환한다.

erase(buffer): buffer에서 글자 element를 삭제하는 함수. 자음인 경우에는 겹자음(겹받침)은 따로 buffer에 저장했기에 pop하면 되며, 모음의 경우에는 각자 다르다(코드 내의 dictionary v 참조)

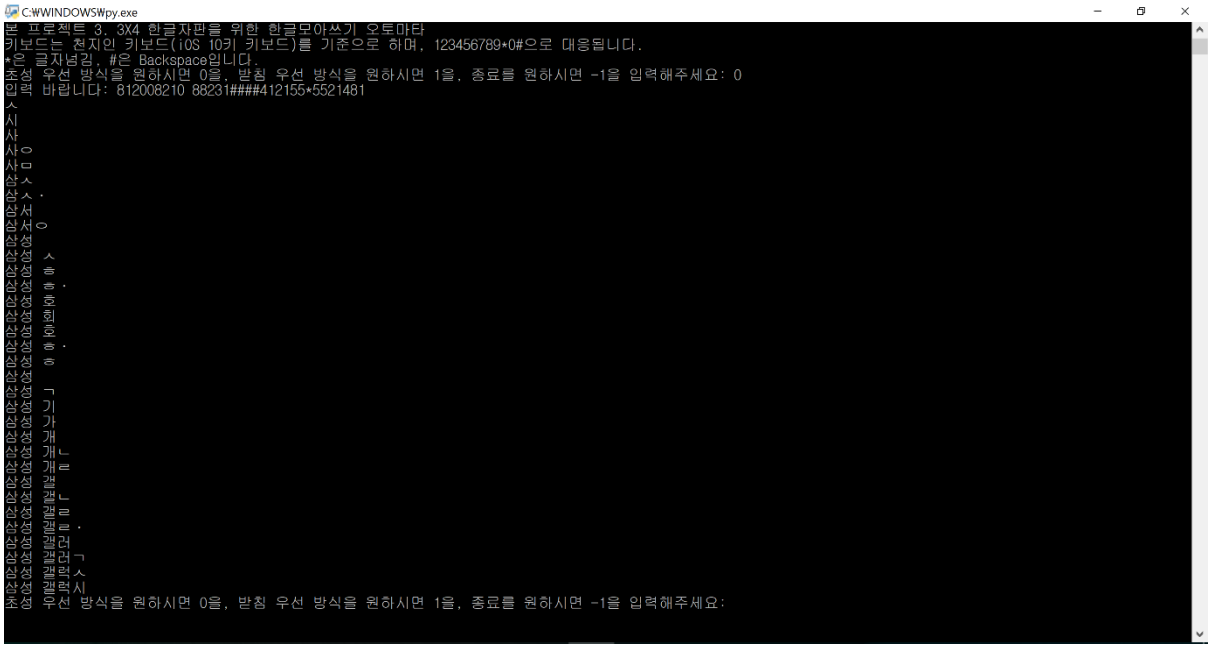
3) 3x4KAuto.py

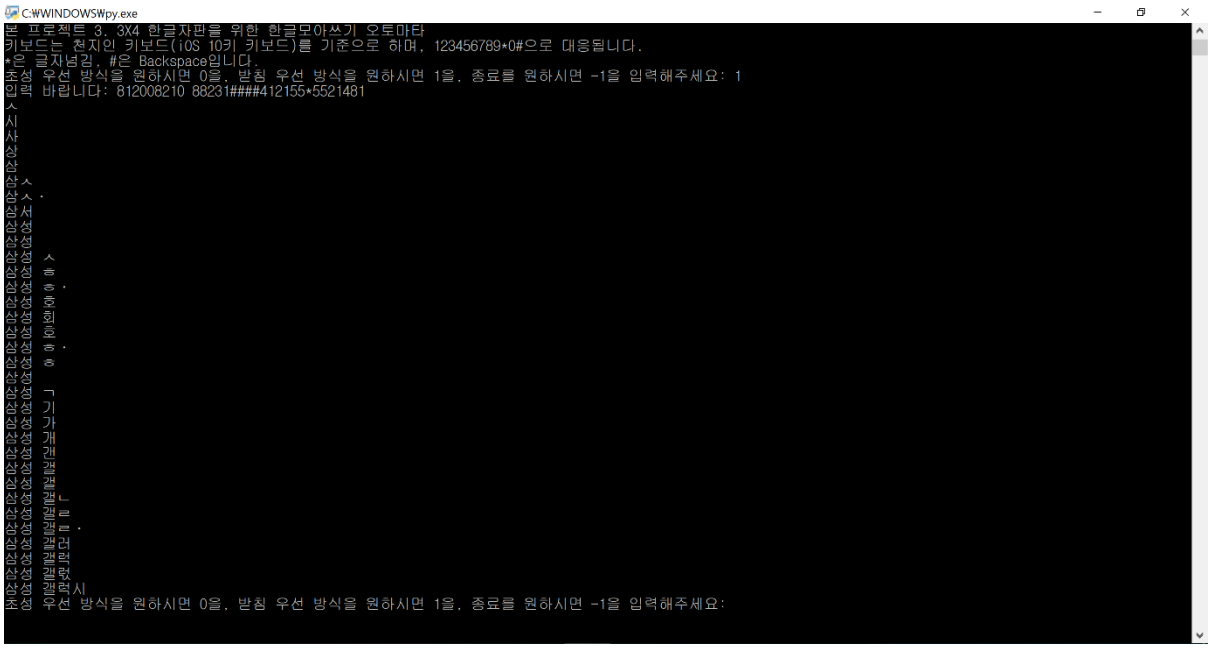
mDFA2MealyM(mDFA): mDFA를 '이 프로그램에 알맞은' Mealy Machine으로 변환하는 함수. Qset, ddic, qzr은 동일하며 Sset은 mDFA의 Sset(0~9)에 *, #, ' '을 추가, ldic은 10개의 천지인 자판이며 ldic은 ddic(state, char)에 대응되는 천지인 자판(PI_list)이다.

Kor3X4Auto(input, MM, flag): 입력된 input과 Mealy Machine을 바탕으로 알맞은 한글 string을 변환하여 반환하는 함수. flag가 0일 때 초성우선방식, 1일 때 받침우선방식이다. 자세한 코드 설명은 주석 참조.

main(): 실행 함수

5. 입력 예시





더 많은 예시 입력은 TXT.txt에 있다.