**본프로젝트 2 정규식 to m-DFA 변환기**

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.

2. 실행 방법

폴더 내의 RE2mDFA.py를 실행한다.

3. 프로그램 설명

정규 입력으로 받아들인 정규식을 lex와 yacc을 통해 파싱하여 abstract syntax tree(이하 AST)로 만든 후, 이를 이용하여 -NFA로 변환한다. 그 후, 예비프로젝트 2-1에서 제작하였던 -NFA to m-DFA 변환 프로그램을 이용하여 최종적으로 m-DFA를 정규 출력한다. 즉, 다음과 같은 과정이 된다.

Input symbol은 알파벳과 숫자로 제한된다.

입력 방식 및 첫 화면은 Figure 1과 같다.

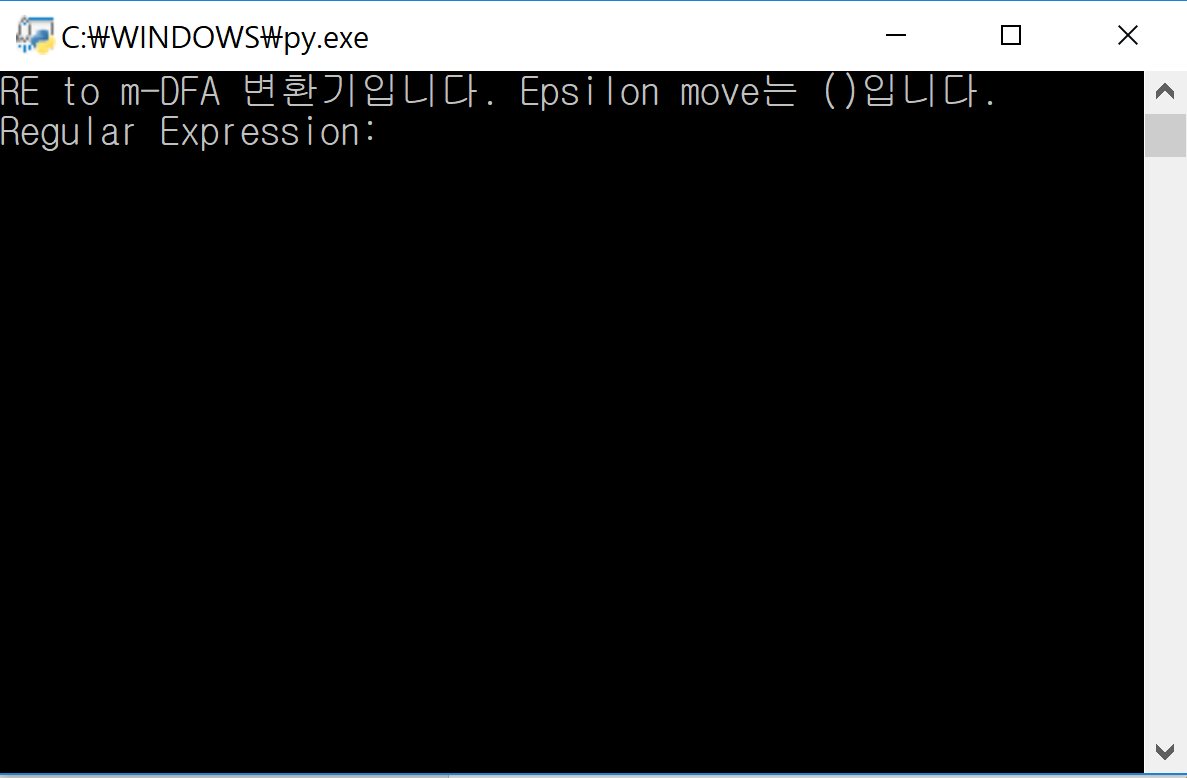


Figure 1

정규식 (0+1)\*011을 예시로 입력하였을 때, 결과는 Figure 2와 같다.

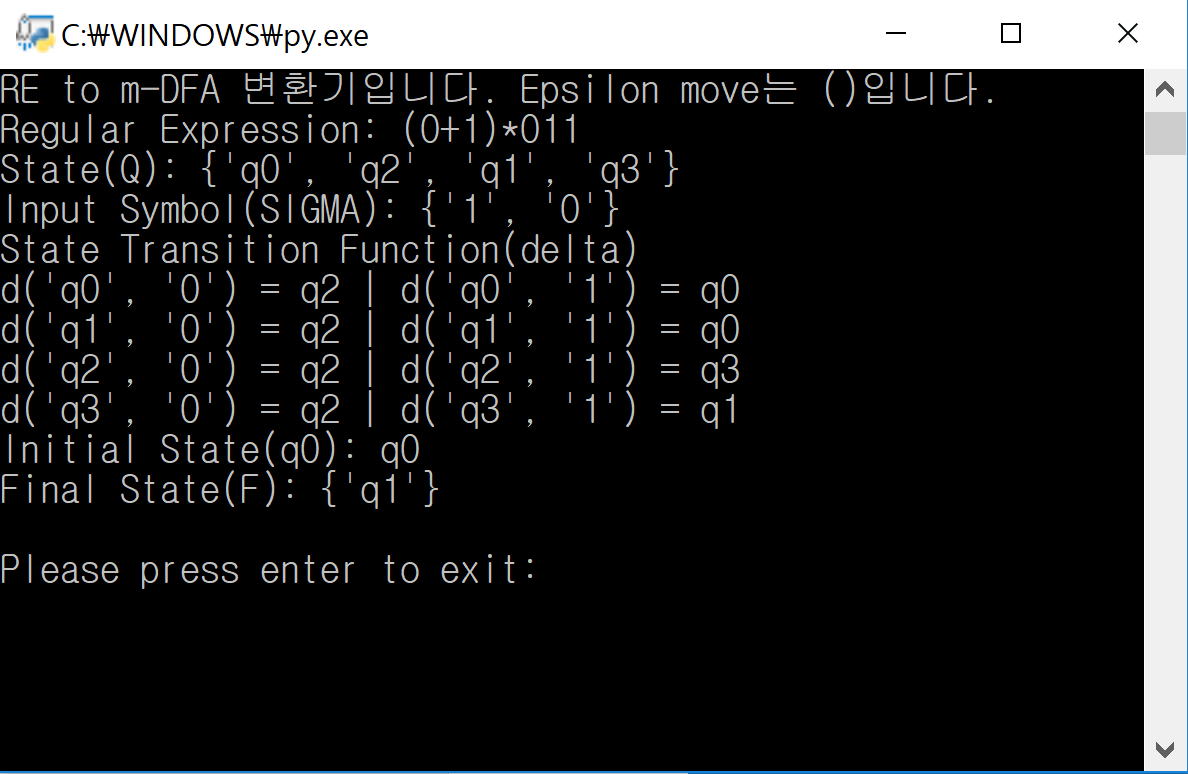
****

Figure 2

에서 은 실행 시마다 달라질 수 있지만, 모두 같은 DFA이다. Enter, 혹은 임의의 string을 입력한 후 enter하면 프로그램이 종료된다.

예비프로젝트 2-1에서 제작하였던 eNFA2mDFA.py 프로그램은 실행 부분을 없앤 것과, class FA의 입력 부분을 조금 더 매끄럽게 바꾼 것과 같은 사소한 변화 이외의 중요한 알고리즘의 변화는 없으므로 생략한다. lex & yacc 부분은 <http://www.dabeaz.com/ply/ply.html>에 나온 예시들을 참조하였고, expression, factor & term은 syntax diagram 표현법을 참조하였다.

4. 코드 설명

1) eNFA2mDFA.py

**FA**: 변화 부분만 기술한다.

|  |  |
| --- | --- |
| 메서드 | 설명 |
| **\_\_init\_\_** | run이 1일 때 입력된 Q, SIG, delta, qz, Fin을 적당한 작업을 통해 Qset, Sset, ddic, qzr, Fset으로 변환하는 것이 아닌, 입력된 Q, SIG, delta, qz, Fin을 그대로 Qset, Sset, ddic, qzr, Fset에 넣어준다. 전 프로그램들 처럼 class FA를 표준 입력(string)을 이용하는 것이 아닌, 직접적인 set과 dictionary를 이용하기 때문이다. 그렇기에, 다른 변환에 사용하는 method는 제거하였다. |
| **printing** | printing 함수를 class FA의 method로 이동시켰다. 전과 변화 없이, class FA를 state(), input symbol(), state transition function(), initial state(), final state() 순서대로 출력하는 method이다. |

**eNFA2mDFA(eNFA)**: DFA2mDFA(eNFA2DFA(eNFA)). 간편한 사용을 위하여 만들었다.

2) RE2mDFA.py

**AST**: AST node class. 객체 변수로 left, right, value가 존재하며, left에는 해당 node의 left child가, right에는 해당 node의 right child가, value에는 해당 노드의 값이 입력된다. Method **\_\_init\_\_**을 통해, value만을 입력하며 left와 right는 None을 기본 값으로 입력한다. Child node는 이후에 처리되기 때문이다.

**parse2AST(parse)**: lex & yacc을 통해 parsing한 list를 AST로 변환하는 함수. 만드는 AST는 이진 트리이며, R + S와 RS는 각각 '+' 또는 "SEREIS"가 root, R이 left child, S가 right child인 tree이며 R\*은 '\*'이 root, R이 left child인 tree가 된다. AST의 말단 노드는 항상 symbol(LETTER), 혹은 e(EPSILON)이다. 이 과정을 recursive하게 반복한다.

**AST2eNFA(root)**: AST2eNFA: AST를 eNFA로 변환하는 함수. Parameter root의 값이 + / SERIES / \* / symbol(epsilon 포함)로 4가지 case가 있으며 교과서 TP 3장 p12 - p13(Figure 3, 4)을 기반으로 프로그래밍하였다. AST의 root부터 recursive하게 반복한다. 자세한 코드 설명은 주석 참조.

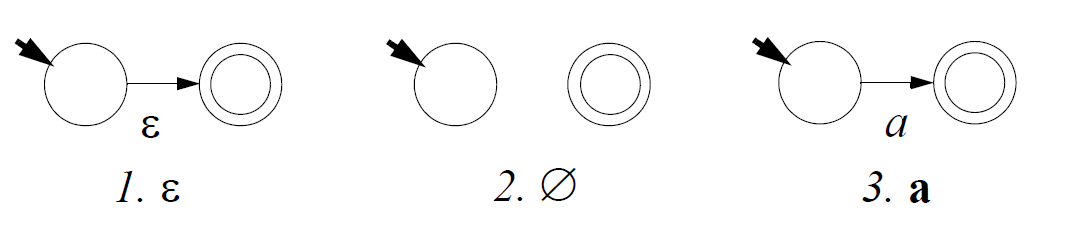


Figure 3

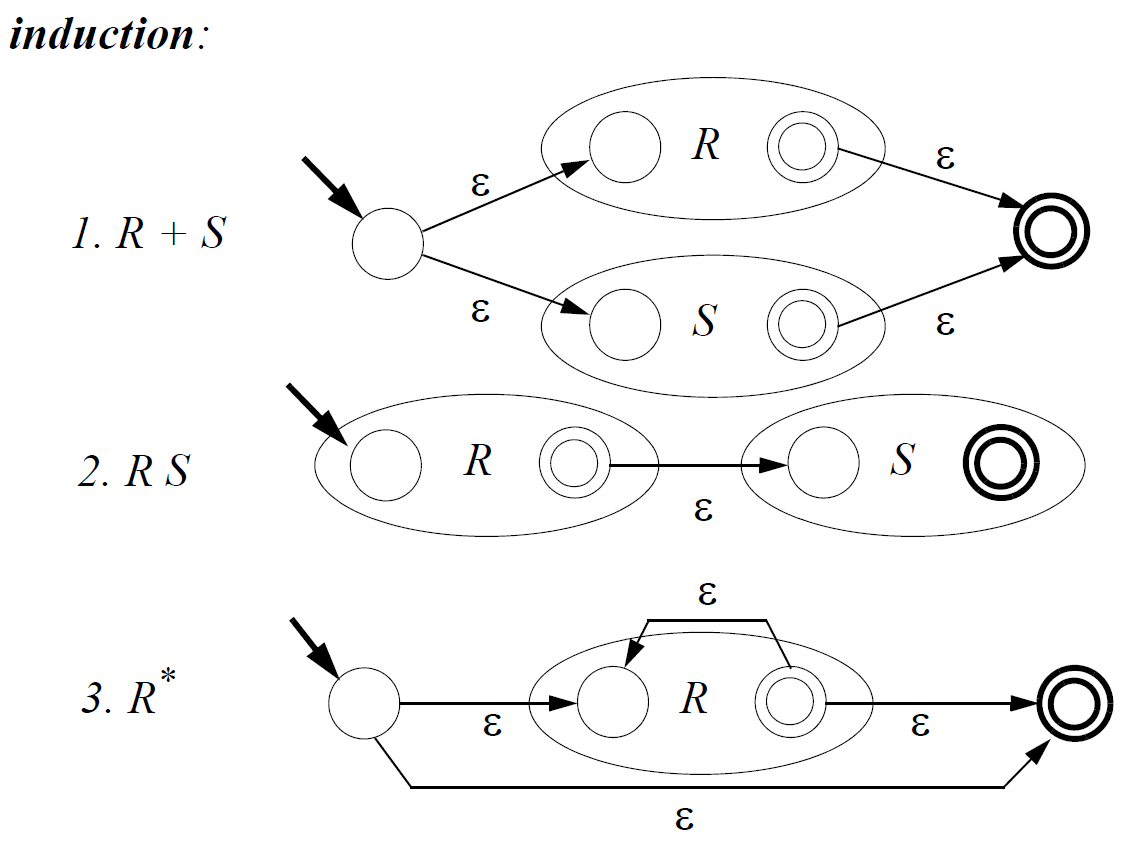
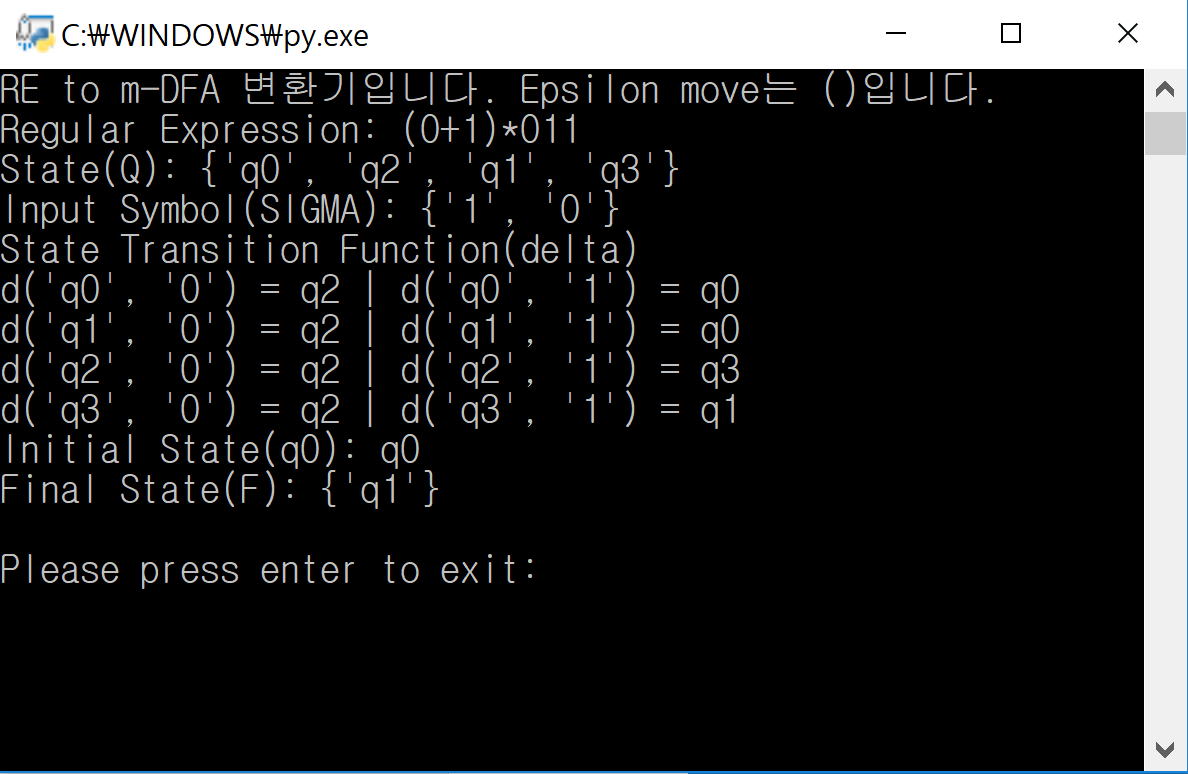


Figure 4

5. 입력 예시

정규식: (0+1)\*011



정규식: (00+1)\*01(11)\* (Homework 3, Q2)

