

예비프로젝트 1-2

20120666 이기문

플랫폼

Python 3.5

소프 파일 구성

P02.py : 메인 파일

mealy.py : Mealy machine 읽기, 기타 함수 정의

실행법

프로젝트 폴더에서 python P02.py 실행

> python P02.py --input (입력 파일 경로) --mealy (Mealy machine 정보 파일 경로) --output (출력 파일 경로) --debug (0|1, 디버깅용 출력 여부)

- 단 P02.py 는 현재 폴더와 한 단계 상위 폴더에 있는 mealy.py 파일을 읽어오므로 주의

프로그램 설명

자세한 내용은 코드 속의 주석 참고. 예비프로젝트 1-1 을 바탕으로 수정하였다.

mealy.read()가 mealy machine 정보를 읽어온다. 예비프로젝트 1-1 의 DFA 정보를 읽어오는 부분을 거의 그대로 재사용하되, mealy machine 이 추가로 가지는 output symbols, output function 정보도 input symbols, transition function 정보와 같은 방식으로 추가했다. 입력 처리 역시 예비프로젝트 1-1 과 거의 동일하나, 각 줄의 마지막에서 final 여부를 확인하는 것 대신, 매 transition 마다 출력할 버퍼에 output symbol 을 쌓는다. 혹시 중간에

transition 이 불가능한 input symbol 이 들어온다면, 기존 버퍼에 상관 없이 "No path exists!"를 출력한다.

```
> python P02.py --input input.txt --output output.txt --debug 1 --mealy mealy.txt
init : a
delta : {( 'c', '1'): 'd', ( 'd', '1'): 'b', ( 'd', '0'): 'c', ( 'a', '0'): 'c', ( 'a', '1'): 'b', ( 'b', '0'): 'c'}
inputSymbols : ['0', '1']
lambda : {( 'c', '1'): 'l', ( 'd', '1'): 'k', ( 'd', '0'): 'i', ( 'a', '0'): 'j', ( 'a', '1'): 'i', ( 'b', '0'): 'k'}
outputSymbols : ['i', 'j', 'k', 'l']
states : ['a', 'b', 'c', 'd']
=====
1
  a to  b by 1
i
=====
10
  a to  b by 1
  b to  c by 0
ik
=====
010
  a to  c by 0
  c to  d by 1
  d to  c by 0
jli
=====
0110
  a to  c by 0
  c to  d by 1
  d to  b by 1
  b to  c by 0
jlk
=====
=====
010101
  a to  c by 0
  c to  d by 1
  d to  c by 0
  c to  d by 1
  d to  c by 0
  c to  d by 1
jlilil
=====
11011
  a to  b by 1
No path exists!
=====
```