

CS322 프로젝트 1-1. DFA 시뮬레이터

개요

한글 오토마타를 구현하기 위한 보조 프로젝트의 일환으로 유한 상태 오토마타(Deterministic Finite Automata: DFA)를 구현하였다.

이론 개요

유한 상태 오토마타 $M_{DFA} = (Q, \Sigma, \delta, q_0, F)$ 는 5가지 요소로 이루어져 있다.

Q : 가능한 상태의 identifier들의 유한집합이다. 서로 다른 상태는 같은 이름을 가질 수 없다.

Σ : 오토마타에 입력할 수 있는 문자들의 집합이다.

δ : 상태 변환 함수 $Q \times \Sigma \rightarrow Q$ 로, 한 상태에서 입력을 받았을 때 이동하는 다음 상태를 지시한다.

q_0 : 오토마타의 초기 상태로, Q 에 속해 있어야 한다.

F : 오토마타에 의해 Accept되는 상태들의 집합으로, Q 의 부분집합이어야 한다.

오토마타 M_{DFA} 는 입력 $s \in \Sigma^*$ 을 받아 $\{True, False\}$ 중 하나를 출력한다. 즉 제한된(정규 언어일 때에 한해) 결정 문제(decision problem)를 풀 수 있는 상태기계이다.

프로그램 설명

project 1-1:

```
testcase/dfa.txt
testcase/input.txt
testcase/output.txt
DFA.py
main.py
```

main.py를 실행하면, DFA.py에 정의된 DFA 클래스를 불러온다. testcase/ 폴더의 dfa.txt를 읽어들이고 DFA를 생성한다. 같은 폴더의 input.txt를 읽어들이고 DFA에 입력하고, 실행한 결과를 output.txt에 저장한다.

개발 언어: python 3.5 (Spyder IDE 사용)

개발 운영체제: Windows 10

DFA.py

DFA 클래스는 아래 세 가지 method를 제공한다.

1. `__init__` (생성자): 5가지 input(DFA의 5요소)를 받아 새 DFA를 생성한다.
2. `step_transit`: 상태와 입력(alphabet)을 받아 그 다음 상태를 리턴한다. Transition function에 직접 접근하지 않도록 메소드를 생성하였다. 두 입력에 대응되는 다음 상태가 없다면 None을 리턴한다.
3. `run`: (alphabet 위의) 스트링을 받아 DFA가 정의하는 언어에 포함되는지를 판별한다. Debug 옵션을 추가하면(debug=True) DFA의 중간 상태들을 점검할 수 있다. 만약 입력이 알파벳의 정의역을 벗어났거나, 혹은 입력과 중간 상태에 대응하는 transition function이 없을 경우에는 false로 간주한다.

main.py

testcase의 dfa, input을 파싱한다.

dfa.txt를 파싱하는 방식은 아래와 같다.

1. 5요소를 설명하는 줄을 기준으로 잘라낸다("State", "Input symbol", ...)
2. 각각의 요소(한 줄, 혹은 여러 줄)를 형식에 알맞게 파싱한다
 - A. 상태, 문자, accept 상태: 반점(comma)을 기준으로 잘라낸다
 - B. 상태 전환 함수: 줄 나눔 문자(\n), 반점을 기준으로 두 번 잘라내어 사전(dict)에 저장한다.
 - C. 시작점: string이므로 별도의 해석이 필요하지 않다.

input.txt는 줄 나눔 문자로만 잘라낸다. 내용이 없는 줄도 입력으로 간주한다(empty string).

testcase/

프로젝트 안내 문서의 '입력 방식 및 입출력 테스트 케이스 설명'과 일치하므로 별도의 기재 안함.

실행 예시

선행 조건

testcase 폴더 안의 dfa.txt, input.txt를 형식에 맞게 수정해주어야 한다.

윈도우 예시(cmd)

```
..> python main.py
```

```
..> notepad testcase/output.py
```

```
...
```

Linux 예시(bash)

```
$ python main.py
```

```
$ cat testcase/output.py
```

```
아니요
```

```
아니요
```

```
네
```

```
...
```