

PL Assignment #03 : Recognizing Tokens 보고서

컴퓨터공학과 3학년

201504280 신윤호

1. 문제 해결 방법

- 전이 그래프를 참조해 볼 수 있다. 초기 상태(state)는 0으로 시작한다. 초기 상태 0에서 Digit이 입력되면 Digit에 대한 승인 트랩 상태인 상태 2로 전이한다. 초기 상태 0에서 '-'가 입력되면 상태 1로 전이한다. 상태 1에서 Digit이 입력되면 상태 2로 전이한다. 초기 상태 0에서 Alpha가 입력되면 Alpha, Digit에 대한 승인 트랩 상태인 상태3으로 전이한다.
- 프로그램의 전체적인 구성은 Token 클래스와 CuteScanner클래스로 구성된다. Token클래스는 토큰의 형태를 정의한다. CuteScanner클래스의 init메소드는 입력받은 문자열을 나누어 리스트에 저장한다. 그리고 반복자를 생성해 next메소드로 리스트의 요소들을 차례로 받을 수 있게 한다.

next_token메소드는 토큰의 타입을 분류하여 반환하는 메소드다. 토큰의 타입을 분류하는 방법은 앞서 언급한 결정적 유한 오토마타의 전이 그래프가 나타내는 바와 동일하게 구현한다. 가령 상태가 1인 경우에 대해서는 if temp_char.isdigit(): state=2 (입력 digit이면 상태2로 전이하라) else : print "ERROR" return None 으로 구현할 수 있다.

tokenize메소드는 next_token메소드에서 토큰을 받아 리스트에 저장한 후, 리스트를 반환하는 함수이다. next_token이 None이 아니라면 append메소드를 사용해 리스트에 토큰을 추가해 준다. 모든 토큰이 추가되면 break로 반복문을 빠져나가고 리스트를 반환한다.

2. 느낀점







이번에 오토마타라는 개념을 처음 접해보게 되었다. 처음에는 과제 문제 자체도 정확히 이해가 되지 않았고 전이 그래프를 보는 방법조차 몰라 헤매었지만, 책과 인터넷을 통해 오토마타의 기본적인 내용을 학습하며 문제를 풀어나가는 과정에서 많은 공부가 되었다. 한편 이번 과제에서도 파이썬에 대해 다양한 것을 배울 수 있었다. 특히 파이썬의 for문은 C의 for문과는 조금 다른 형태인 것을 알 수 있었고, 리스트와 관련해서 append함수를 알게 되었다. 뿐만 아니라 소스를 분석하는 과정에서 iterator개념, strip함수, split함수 등을 배우게 되었다.

3. 테스트 코드 실행 결과

```
def Test_CuteScanner():
    test_cute = CuteScanner("banana 267 h cat -3789 7 y2010")
    test_tokens=test_cute.tokenize()

    print test_tokens
    for token_i in test_tokens:
        print token_i
    print "end"

Test_CuteScanner()
```







D:\Python27\python.exe D:/대학/프로그래밍언어/과제/hw03/201504280_hw03.py
[[ID: banana], [INT: 267], [ID: h], [ID: cat], [INT: -3789], [INT: 7], [ID: y2010]]
[ID: banana]
[INT: 267]
[ID: h]
[ID: cat]
[INT: -3789]
[INT: 7]
[ID: y2010]
end

Process finished with exit code 0