
프로그래밍언어 개론 보고서

[05] Cute17 Scanner

제출일	2017/06/30
학 번	201000287
소 속	일어일문학과
이 름	유다훈

1 과제문제

1 주어진 코드에서 함수 구현

- TransitionMatrix 를 담당하는 enum 초기화
- init_TM() : 전달받은 문자열이 어떠한 타입의 토큰으로 분류할지 검사하는 메소드
- nextToken() : 에러 핸들링 코드 작성
- 과제의 목적
 - 특정 문자열을 입력 받을 경우, 해당 문자열을 나누어서 각 요소를 인식하여 속성 대로 분류하고 출력해주는 Scanner Map 을 이용한 Key, Value 분류를 이용하여 토큰을 분류한다.

2 과제 해결 방법

- 확장된 상태도에 따라 TransitionMatrix 를 구현한다.

```
public enum TokenType {  
    //1) TokenType state number  
    INT(2), ID(3), MINUS(1), PLUS(4), L_PAREN(5), R_PAREN(6), TRUE(8), FALSE(9),  
    , TIMES(10), DIV(11), LT(12), GT(13), EQ(14), APOSTROPHE(15), QUESTION(16),  
  
    // not present in automata  
    DEFINE(-1), LAMBDA(-1), COND(-1), QUOTE(-1), NOT(-1), CAR(-1), CDR(-1), CONS(-1),  
    ATOM_Q(-1), NULL_Q(-1), EQ_Q(-1);  
}
```

```
private enum KeyWord {
    //2) 나머지 키워드 작성
    DEFINE("define", TokenType.DEFINE), LAMBDA("lambda", TokenType.LAMBDA),
    COND("cond", TokenType.COND),
    QUOTE("quote", TokenType.QUOTE),
    NOT("not", TokenType.NOT),
    CAR("car", TokenType.CAR),
    CDR("cdr", TokenType.CDR),
    CONS("cons", TokenType.CONDS),
    ATOM_Q("atom?", TokenType.ATOM_Q),
    NULL_Q("null?", TokenType.NULL_Q),
    EQ_Q("eq?", TokenType.EQ_Q);
}
```

2 init_TM()

```
//4) DFA를 만족하는 Transition Matrix작성하시오.
transM[0]['-'] = 1; // '-'
transM[0]['('] = 5;
transM[0][')'] = 6;
transM[0]['>'] = 6;
transM[0]['#'] = 7;
transM[7]['T'] = 8;
transM[7]['F'] = 9;
transM[3]['?'] = 16;
transM[0]['+'] = 4;
transM[0]['<'] = 12;
transM[0]['='] = 14;
transM[0]['>'] = 13;
transM[0]['*'] = 10;
transM[0]['/'] = 11;
transM[0]['\''] = 15;
```

- 이전 상태에서 더욱 확장한 형태로, 특수문자에 대해서 처리한다.
- while 문을 이용하여 입력받을 토큰이 없을 때까지 토큰분류하는 검사문을 실행한다.

3 nextToken()

```
if (StateNew == -1) {
    //5) 입력된 문자의 상태가 리젝트이므로 에러 메시지 출력 후 리턴함
    System.out.println(String.format("acceptState error %s\n", temp));
    return null;
}
```

- 토큰으로 분류할 수 없는 문자열이 들어왔을 때는 에러 메시지를 출력한다.

3 과제를 해결하면서 느낀 점

- 1 실습 시간 질문을 통하여 enum 형태에 대해서 조금 더 이해할 수 있었다.
- 2 자료구조 Map 을 통하여 key, value 형태의 자료구조를 조금이나마 사용해볼 수 있었다.
- 3 실습을 함에 따라 작성해야하는 코드가 많이 늘어난 것을 알 수 있다.