

## PL Assignment #05 : Cute17 Parser 보고서

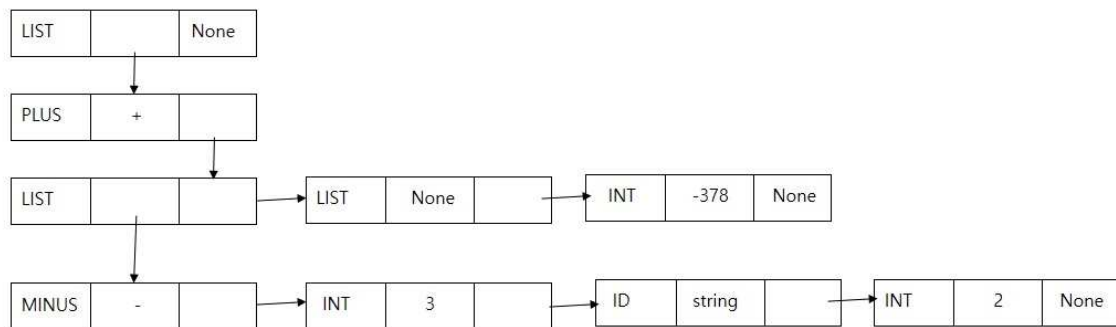
컴퓨터공학과  
201504280 신윤호

### 1. 문제 해결 방법

• 노드의 구조는 

TYPE	VALUE	NEXT
------	-------	------

 형태로 구성되어 있다. 이때 유의할 점은 타입이 LIST인 경우에는 VALUE가 다른 노드를 가리킬 수 있으므로 이 부분에 대한 처리를 고려해야 한다. 이번 과제에서는 `elif self.type is TokenType.List : result = '('+str(self.value)+'')`으로 타입이 LIST인 경우 VALUE에 대해 str이 순환호출 되도록 코드가 이미 주어졌다. 따라서 추가적으로 구현해야 할 부분은 Next에 대한 순환호출을 통해 다음 노드에 대해서 순환적으로 str함수가 적용되도록 하는 것이었다. 이 부분을 `if self.next != None : result = result + ' ' + str(self.next)`으로 구현하였다. 즉, `self.next != None`인 경우에 순환호출이 이루어지는데, 호출이 이루어지며 결과를 result에 누적해서 기록하도록 하였다. 그리고 `self.next`가 None이 되면 순환호출을 빠져나가게 되고 결과인 result를 반환하도록 하였다. 테스트 코드의 방문 과정은 다음 그림과 같다.



• 토큰을 Node로 만드는 `_create_node`함수는 토큰 타입에 따른 적절한 노드를 만들어주는 함수이다. 예를 들어 설명하자면, 토큰의 타입(`token.type`)이 PLUS라면(`CuteType.PLUS`) 노드의 타입을 PLUS로 전달하고, 노드의 value로 토큰의 lexeme을 전달해 노드를 생성하도록 조건을 작성했다.

### 2. 느낀점

`--str--` 문제에서 어려움을 겪었다. List타입인 경우에는 노드를 value로 가진다는 것에 너무 집착해서 애를 먹었다. 순환호출 만들기 전에 경계조건부터 찾으려고 고민하면서 다양한 시도를 하다가 문제에 value로 가는 경우를 처리하는 코드가 있다는 것을 보고 next로 가는 부분을 작성했더니 결과가 나왔다. 결과가 나오고 살펴보니 어차피 List타입은 terminal이 아니기 때문에 List타입 노드가 parsing tree의 단말노드가 되지 않아서 경계조건을 걸 필요가 없었다. 이런 부분을 생각했을 때 이번 과제는 문제를 먼저 이해하지 않고 막무가내로 달려들어서 오히려 고생을 했던 과제인 것 같다.

### 3. 테스트 코드 실행 결과

```
def Test_BasicPaser():
    test_cute = CuteScanner('( + ( - 3 String 2 ) ( ) -378 )')
```

```

test_tokens = test_cute.tokenize()
print test_tokens
test_basic_paser = BasicPaser(test_tokens)
node = test_basic_paser.parse_expr()
print node

```

Test\_BasicPaser()

```

D:\Python27\python.exe D:/대학/프로그래밍언어/과제/hw05/201504280_hw05.py

=== tokenize ===
( + ( - 3 String 2 ) ( ) -378 )
scanning...
[L_PAREN: '(']
[PLUS: '+']
[L_PAREN: '(']
[MINUS: '-']
[INT: '3']
[ID: 'String']
[INT: '2']
[R_PAREN: ')']
[L_PAREN: '(']
[R_PAREN: ')']
[INT: '-378']
[R_PAREN: ')']
[[L_PAREN: '(', [PLUS: '+'], [L_PAREN: '(', [MINUS: '-'], [INT: '3'], [ID: 'String'], [INT: '2'], [R_PAREN: ')'], [L_PAREN: '(', [R_PAREN: ')'], [INT: '-378'], [R_PAREN: ')]]]
([PLUS] ([MINUS] [INT:3] [ID:String] [INT:2]) (None) [INT:-378])

Process finished with exit code 0

```