

PL Assignment #06 : Cute17 Parser 보고서

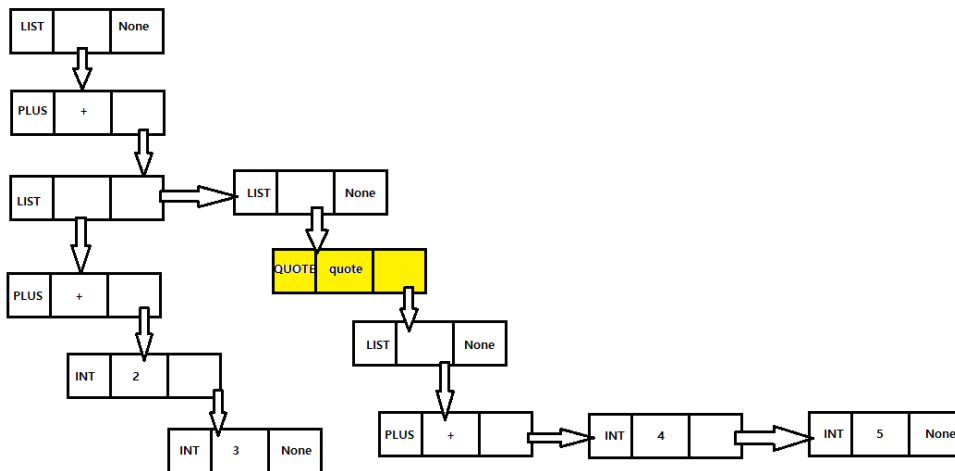
컴퓨터공학과
201504280 신윤호

1. 문제 해결 방법

- `_create_node(self, token)`

조건에 `elif token.type is CuteType.QUOTE or token.type is CuteType.APOSTROPHE`이면 `return Node(TokenType.QUOTE, 'quote')`를 추가하였다. 즉, 메소드에 전달된 token의 타입이 QUOTE 이거나, APOSTROPHE인 경우 QUOTE타입의 노드를 생성하는 코드를 추가하였다. 문제에서 '는 quote의 약자이므로 quote와 동일하게 처리되는 것이라 제시되었기 때문에 APOSTROPHE 토큰이 들어왔을 때에도 QUOTE 토큰이 들어왔을 때와 마찬가지로 quote노드를 생성하도록 한 것이다. 뿐만 아니라 value도 동일하게 갖도록 하기 위해서 두 경우 모두 value를 'quote'로 가지도록 하였다.

"(+ (+ 2 3) (quote (+ 4 5)))"인 경우 내부 구조는 아래 그림과 같다.



- `__str__(self)`

이번 과제에서 str함수는 노드의 타입이 LIST인 경우를 수정하였다. 만약 `self.type`이 LIST이고 `self.value.type`이 QUOTE이면 괄호를 표시하지 않고 그냥 `self.value`에 str함수를 순환 호출한다. 이는 `(quote (expression))` 또는 `'(expression)'`이 있을 때 출력 결과가 `('(expression))'`이 아닌 `'(expression)'`으로 출력되기 하기 위함이다. 한편, LIST타입 노드는 value로 노드를 가질 수 있으므로 LIST타입 노드에 대해 `self.value.type`가 가능하다.

2. 느낀점

이번 과제는 공교롭게도 시험기간이 겹쳐서 시간에 쫓기며 수행했던 것 같다. 그래도 지난번 과제의 개념이 그대로 이어지는 과제라서 다행이었다고 생각한다. 지난번 과제를 수행하면서 코드를 읽을 때 메모해 놓은 것이 이번 과제에 많은 도움이 되었다.

3. 테스트 코드 실행 결과

- 1) "(+ (+ 2 3) (quote (+ 4 5)))" 테스트 결과

```
def Test_BasicPaser():
    #test_cute = CuteScanner('( + ( - 3 String 2 ) ( ) -378 )')
    test_cute = CuteScanner("( + ( + 2 3 ) ( quote ( + 4 5 ) ) )")
    #test_cute = CuteScanner("( quote ( + 3 2 ) )")
    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/hw06/hw06/201504280_hw06.py
=== tokenize ===
( + ( + 2 3 ) ( quote ( + 4 5 ) ) )
scanning...
[L_PAREN: (]
[PLUS: +]
[L_PAREN: (]
[PLUS: +]
[INT: 2]
[INT: 3]
[R_PAREN: )]
[L_PAREN: (]
[QUOTE: quote]
[L_PAREN: (]
[PLUS: +]
[INT: 4]
[INT: 5]
[R_PAREN: )]
[R_PAREN: )]
[R_PAREN: )]
[[L_PAREN: (], [PLUS: +], [L_PAREN: (], [PLUS: +], [INT: 2], [INT: 3], [R_PAREN: )], [L_PAREN: (], [QUOTE: quote], [L_PAREN: (], [PLUS: +], [INT: 4], [INT: 5], [R_PAREN: )], [R_PAREN: )], [R_PAREN: )]]
([PLUS] ([PLUS] [INT:2] [INT:3]) ' ([PLUS] [INT:4] [INT:5]))

Process finished with exit code 0
```

([PLUS] ([PLUS] [INT:2] [INT:3]) ' ([PLUS] [INT:4] [INT:5]))

2) "(' (+ 3 2))" 테스트 결과

```
def Test_BasicPaser():
    #test_cute = CuteScanner('( + ( - 3 String 2 ) ( ) -378 )')
    #test_cute = CuteScanner("( + ( + 2 3 ) ( quote ( + 4 5 ) ) )")
    test_cute = CuteScanner("( ' ( + 3 2 ) )")
    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/hw06/hw06/201504280_hw06.py

```
=== tokenize ===
( ' ( + 3 2 ) )
scanning...
[L_PAREN: (]
[APOSTROPHE: ']
[L_PAREN: (]
[PLUS: +]
[INT: 3]
[INT: 2]
[R_PAREN: )]
[R_PAREN: )]
[[L_PAREN: (], [APOSTROPHE: '], [L_PAREN: (], [PLUS: +], [INT: 3], [INT: 2], [R_PAREN: )], [R_PAREN: )]]
' ([PLUS] [INT:3] [INT:2])
```

Process finished with exit code 0

3) "(+ (+ 2 3) (quote (+ 4 5)))" 내부구조

```
▼ self = {Node} ([PLUS] ([PLUS] [INT:2] [INT:3]) ' ([PLUS] [INT:4] [INT:5]))
  next = {NoneType} None
  type = {int} 5
  ▼ value = {Node} [PLUS] ([PLUS] [INT:2] [INT:3]) ' ([PLUS] [INT:4] [INT:5])
    ▼ next = {Node} ([PLUS] [INT:2] [INT:3]) ' ([PLUS] [INT:4] [INT:5])
      ▼ next = {Node} ' ([PLUS] [INT:4] [INT:5])
        next = {NoneType} None
        type = {int} 5
        ▼ value = {Node} ' ([PLUS] [INT:4] [INT:5])
          ▼ next = {Node} ([PLUS] [INT:4] [INT:5])
            next = {NoneType} None
            type = {int} 5
            ▼ value = {Node} [PLUS] [INT:4] [INT:5]
              ▼ next = {Node} [INT:4] [INT:5]
                ▼ next = {Node} [INT:5]
                  next = {NoneType} None
                  type = {int} 1
                  value = {str} '5'
                  type = {int} 1
                  value = {str} '4'
                  type = {int} 3
                  value = {str} '+'
                  type = {int} 23
                  value = {str} 'quote'
                type = {int} 5
              ▼ value = {Node} [PLUS] [INT:2] [INT:3]
                ▼ next = {Node} [INT:2] [INT:3]
                  ▼ next = {Node} [INT:3]
                    next = {NoneType} None
                    type = {int} 1
                    value = {str} '3'
                    type = {int} 1
                    value = {str} '2'
                    type = {int} 3
                    value = {str} '+'
                  type = {int} 3
                  value = {str} '+'
```

4) "(' (+ 3 2))" 내부구조

```
▼ self = {Node} ' ([PLUS] [INT:3] [INT:2])
  next = {NoneType} None
  type = {int} 5
▼ value = {Node} ' ([PLUS] [INT:3] [INT:2])
  ▼ next = {Node} ([PLUS] [INT:3] [INT:2])
    next = {NoneType} None
    type = {int} 5
    ▼ value = {Node} [PLUS] [INT:3] [INT:2]
      ▼ next = {Node} [INT:3] [INT:2]
        ▼ next = {Node} [INT:2]
          next = {NoneType} None
          type = {int} 1
          value = {str} '2'
          type = {int} 1
          value = {str} '3'
          type = {int} 3
          value = {str} '+'
        type = {int} 23
        value = {str} 'quote'
```