

PL Assignment #07 : Cute17 Built-in Function 구현 보고서

컴퓨터공학과
201504280 신윤호

1. 문제 해결 방법

• cdr(node)

`l_node = run_expr(node.value.next)`를 수행하면 `l_node`에는 quote를 포함한 피연산자가 저장된다. `result=strip_quote(l_node).value`를 수행하면 quote를 제거할 수 있다. `.value`를 붙여주는 이유는 quote뒤의 list노드를 제거하기 위함이다. 예를 들어 설명하자면, 주어진 문장이 `(car '(2 3 4))`라면 `run_expr(node.value.next)`를 수행하면 `'(2 3 4)`가 되고, `strip_quote(l_node)`를 수행하면 `(2, 3, 4)`가 된다. 즉, `strip_quote(l_node).value`를 하면 INT타입 노드 2, 3, 4를 얻을 수 있다. 이것이 `result`에 저장된다.

그 후, `create_new_quote_list`함수를 이용해 `result`를 반환한다. 이 때, `cdr`은 맨 처음 원소를 제외한 나머지 원소들을 반환하는 함수이므로 `create_new_quote_list`의 인자로 `result.next`를 전달해 첫 번째 원소를 제외한다. 또한 `create_new_quote_list`는 두 번째 인자 값으로 `flag`를 전달받는다. `flag`는 기본적으로 `false`로 지정되어 있는데, 이 경우 리스트를 생성하지 않고 원소 하나만 quote로 감싸 반환하게 된다. `cdr`함수는 여러 원소들을 반환하게 되므로 `flag`를 `True`로 전달해 준다.

• cons(node)

`cons` 하나의 원소를 리스트에 포함시키는 함수이다. 그런데 이 때, 원소는 항상 atom인 것이 아니라 list가 원소로 올 수도 있다. 즉 `(cons 1 '(2 3 4))`도 가능하지만 `(cons '(2 3) '(4 5 6))`도 가능하다. 따라서 포함시킬 원소가 list인 경우와 list가 아닌 경우(atom인 경우)로 경우의 수를 나누어서 구현하였다.

먼저 앞서 `cdr`에서 했던 것처럼 `total_node = run_expr(node.value.next)`를 수행하여 피연산자들만 남긴다. 그 후, 만약 `total_node`의 타입이 list가 아닌 경우(atom인 경우) 노드를 하나 생성해서 포함시킬 원소의 값으로 초기화한다. 즉, `Node(total_node.type, total_node.value)`를 한다. 그리고 이를 `head_node`가 가리키도록 한다. 다음으로 `tail_node`는 첫 원소(포함시킬 원소)를 제외한 나머지 원소들을 가리킨다. 이때, `cdr`에서와 마찬가지로 `strip_quote(total_node.next).value`를 수행하여 quote를 제외한 원소들만을 취한다. 그 후 `head_node.next = tail_node`를 가리키도록 하여 원소들을 하나로 합친다. 마지막으로 `if`문을 빠져나가 `create_new_quote_list(head_node, True)`를 수행하여 합쳐진 원소들을 리스트로 만들어 반환한다. 이때 `flag`를 `True`로 전달한 이유는 `cdr`에서 언급한 바와 동일하다.

만약 `total_node`의 타입이 list인 경우에도 `head_node`가 첫 원소(포함시킬 원소)를 가리키는 것은 동일하다. 다만 `head_node`가 atom이 아니라 list인 만큼 quote를 제거하는 함수가 한번 들어간다는 차이점이 있다. 뿐만 아니라 붙일 원소가 atom인 경우에는 원소가 하나이기 때문에 그냥 붙이면 됐지만 붙일 원소가 list인 경우에는 붙일 원소의 맨 끝 원소가 리스트의 맨 처음 요소를 가리키도록 해야 했다. 이는 `Node`객체 메소드 중에서 리스트의 맨 마지막에 노드를 붙여주는 `set_last_next` 라는 메소드가 있어서 이를 이용해 구현했다. 그리고 이 경우에도 마지막으로 `if`문을 빠져나가 `create_new_quote_list(head_node, True)`를 수행해 합쳐진 원소들을 리스트로 만들어 반환한다.

• null_q(node)

먼저 `q_node=node.value.next`를 한다. 이때 `Node`는 List타입 노드이고, 노드의 `value`는 연산자에 해당하는 노드인 `NULL_Q` 타입 노드라고 할 수 있다. 따라서, `node.value.next`는 `null_q`의 quote를 포함한 피연산자에 해당한다. 즉, `q_node`는 `null_q`연산의 quote를 포함한다.

피연산자를 가지게 된다. 가령 (null? '())' 이라면 q_node에는 '()'가 저장된다. 그 뒤, q_node = strip_quote(q_node)를 하여 피연산자에서 quote를 제거한다. 앞서 든 예시에서 q_node에 저장된 '()'가 quote가 제거되고 ()만 남는 것이다. 여기서 q_node의 value가 None이라면 null이므로 True를 반환한다. 그렇지 않은 경우 False를 반환한다.

- atom_q(node)

null_q와 마찬가지로, q_node=node.value.next와 q_node=strip_quote(q_node)를 하면 Quote까지 제거된 피연산자가 q_node에 저장된다. 만약 q_node가 값을 가진 List라면 False를 반환한다. 만약 값을 가지지 않은 List이거나(null list), List가 아니면 True를 반환한다.

2. 느낀점

지금까지 했던 과제들 중에서 가장 어려움을 느꼈다. 일단 주어진 함수 파악하는 것부터 엄청 힘들었다. 특히 run_list함수에 return값이 run_func(op_code_node)(root_node)인데 이런 문법을 처음 봐서 고생했다. 그리고 이번 과제는 보고서 작성하는 것도 쉽지 않았다. 분명히 내가 코딩한 것인데 막상 보고서를 쓰려고 보니 말로 풀어서 쓰는 것이 쉽지 않았다. 그리고 저번부터 Node객체에 set_last_next메소드가 아무데도 쓰이지 않아서 도대체 이게 왜 정의되어 있는 것인지 궁금했는데, 이번 과제에 쓸 수 있어서 궁금증이 풀렸다.

3. 테스트 코드 실행 결과

```
C:\Python27\python.exe "C:/Users/Yoonho/Google 드라이브/대학/프로그래밍언어/과제/hw07/hw07/hw07_Frame.py"
```

```
2
'(2 3)
'(3 4)
'((4 5) 6)
'( )
'(1 2 3 4)
'((2 3) 4 5 6)
'(2)
#T
#F
#F
#T
#F
#T
#F
#F
#T
#F
```

```
Process finished with exit code 0
```