컴파일러개론 Assign 01

과제 제출일자 : 2019/09/07
학번 : 201502085, 이규봉
분반 : 01
과제 : Noo Compiler 구현

문제 해결 방법

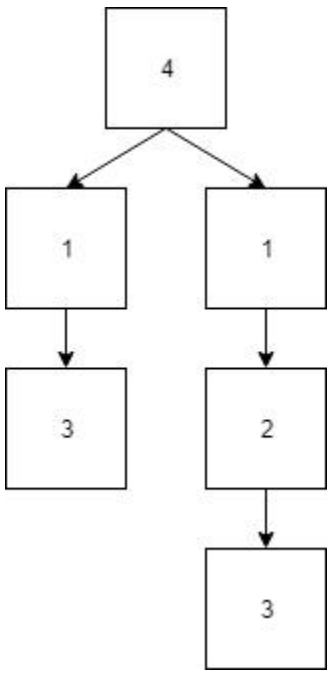
제가 문제를 해결한 방법은 아래와 같습니다.

1. 입력 noo 파일에서 Line을 읽어들입니다. (최대 1줄로 제한)
2. Line을 ‘ 별로 split해 나눈 후 각각의 “ 개수를 원소로 갖는 노드를 생성합니다.
3. 노드를 이어 파싱 트리를 생성합니다. 파싱 트리를 생성할 땐 각 Node가 가지는 “의 개수 (편의를 위해 이를 degree라고 표현하겠습니다.) 에 따라 트리의 구조를 다르게 생성합니다. 그렇게 하기 위해 NooTree 내부에서 TreeIterator라는 이름의 내부 클래스를 사용합니다. 에러 입력을 무시할 경우 각 트리의 리프 노드엔 degree가 3인 노드가 존재합니다. degree가 4인 노드를 만나면 TreeIterator 내부의 retNodeStk 에 이 노드를 집어넣습니다. degree가 5인 노드를 만나면 같은 노드를 두 번 집어넣습니다. (뒤에 3개의 인자가 와야 하므로) 이렇게 함으로써, 리프 노드를 만났을 때 스택이 비어 있지 않다면 스택에서 pop한 노드가 리턴할 노드가 되는 것을 알 수 있습니다.
4. 위처럼 retNodeStk 스택을 이용해 처리한 이유는 degree가 4인 노드나 5인 노드가 중첩되는 케이스를 처리하기 위해서입니다. return할 Node를 1개만 이용해도 degree가 4인 노드나 5인 노드가 중첩되지 않는 케이스를 처리할 수 있지만, 중첩되는 케이스는 처리할 수 없습니다.
5. NooNode의 PrintCstr이란 메서드를 이용해 트리를 재귀적으로 돌며 각 노드들의 degree에 따라 해당하는 C 코드로 변환합니다. Base case는 degree가 3인 노드를 만났을 때입니다.

테스트 코드 실행 결과 (캡쳐)

1. '"""'"'"""'"'"'"'"""

위와 같은 코드는 아래와 같은 트리를 생성합니다.

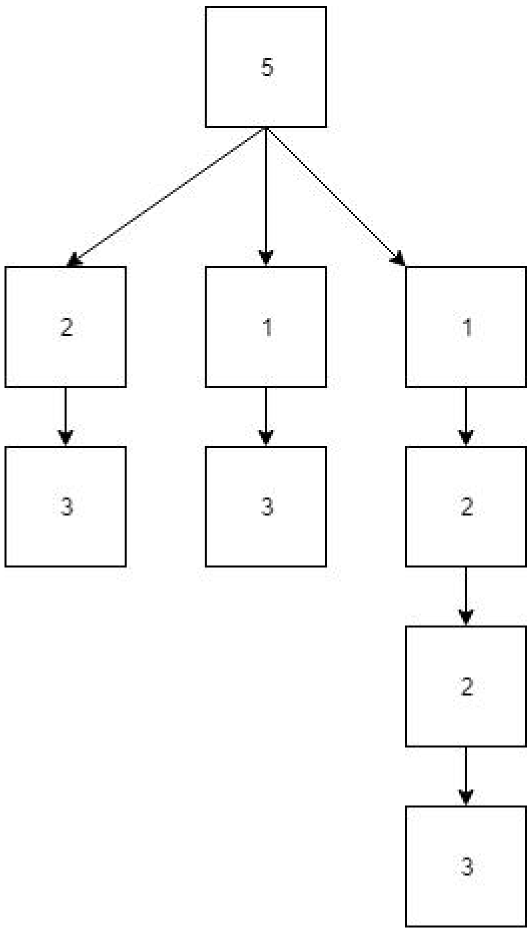


위 트리를 파싱하며 생성하는 C 코드는 아래와 같습니다.

```c
#include <stdio.h>
int main() {
    int r, t1, t2, t3;
    r = 0;
    printf("%d", r);
    r = 0;
    t1 = r;
    r = t1 + 1;
    printf("%d", r);
    return 1;
}
```

2. '"""""""'""""'""""'"""""'""'""""'"""""

위와 같은 코드는 아래와 같은 트리를 생성합니다.

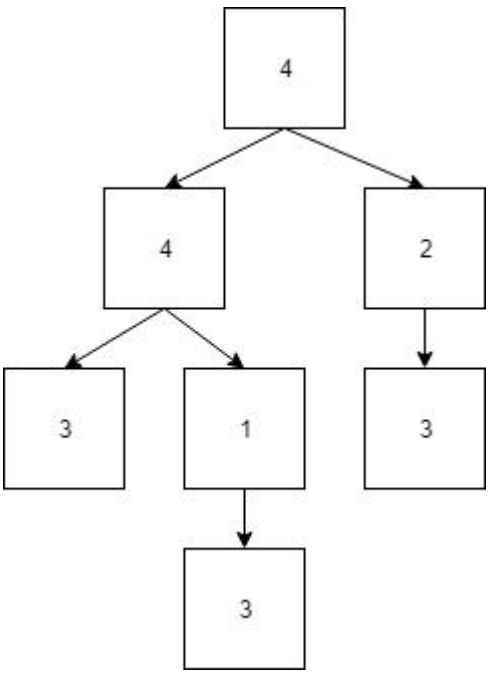


위 트리를 파싱하며 생성하는 C 코드는 아래와 같습니다.

```c
#include <stdio.h>
int main() {
    int r, t1, t2, t3;
    r = 0;
    t1 = r;
    r = t1 + 1;
    t1 = r;
    if(t1 != 0){
    r = 0;
    printf("%d", r);
    }
    else{
    r = 0;
    t1 = r;
    r = t1 + 1;
    t1 = r;
    r = t1 + 1;
    printf("%d", r);
    }
    return 1;
}
```

3. '"""'"""'"""'"""'"'"""'"'"""

위와 같은 코드는 아래와 같은 트리를 생성합니다.

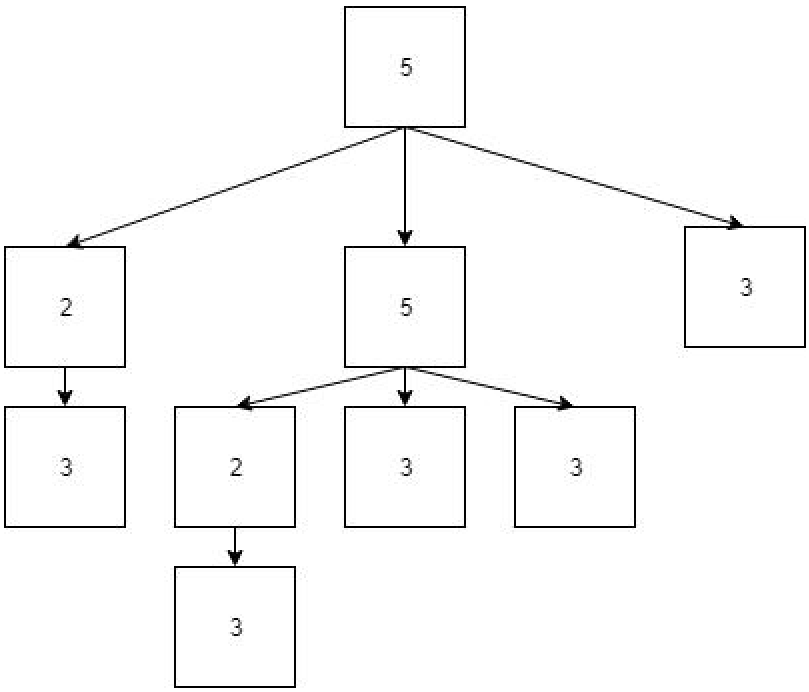


위 트리를 파싱하며 생성하는 C 코드는 아래와 같습니다.
degree가 4인 노드의 자식 노드가 다시 degree가 4인 노드이기 때문에, 3 -> (3 -> 1) -> (3 -> 2) 의 순서로 노드들이 C 코드로 나타나게 됩니다.

```c
#include <stdio.h>
int main() {
    int r, t1, t2, t3;
    r = 0;
    r = 0;
    printf("%d", r);
    r = 0;
    t1 = r;
    r = t1 + 1;
    return 1;
}
```

4. '''

위와 같은 코드는 아래와 같은 트리를 생성합니다.



위 트리를 파싱하며 생성하는 C 코드는 아래와 같습니다.

Y 수식 (root 노드의 두 번째 자식 노드) 에 degree가 5인 노드가 들어 갔기 때문에 if문 내부에 if문이 중첩되었습니다.

```c
#include <stdio.h>
int main() {
    int r, t1, t2, t3;
    r = 0;
    t1 = r;
    r = t1 + 1;
    t1 = r;
    if(t1 != 0){
    r = 0;
    t1 = r;
    r = t1 + 1;
    t1 = r;
    if(t1 != 0){
    r = 0;
    }
    else{
    r = 0;
    }
    }
    else{
    r = 0;
    }
    return 1;
}

```

총 과제 수행에 걸린 시간

어떻게 코드를 짜면 좋을지 생각하는데 2 시간 정도 소요한 것 같습니다. 그리고 트리 구조를 생성하고 읽어들여 C 코드로 바꾸는 방법을 선택하고 자바 코드로 작성하는데 2 ~ 3 시간 정도 소요한 것 같습니다. 그 외 버그 수정 및 디버깅에 3 ~ 4 시간 정도 소요한 것 같습니다. 그 외 보고서 작성에 1 시간 정도 소요되었습니다.