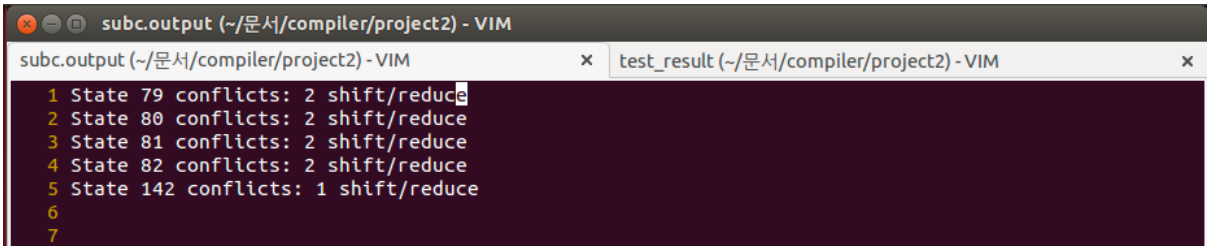
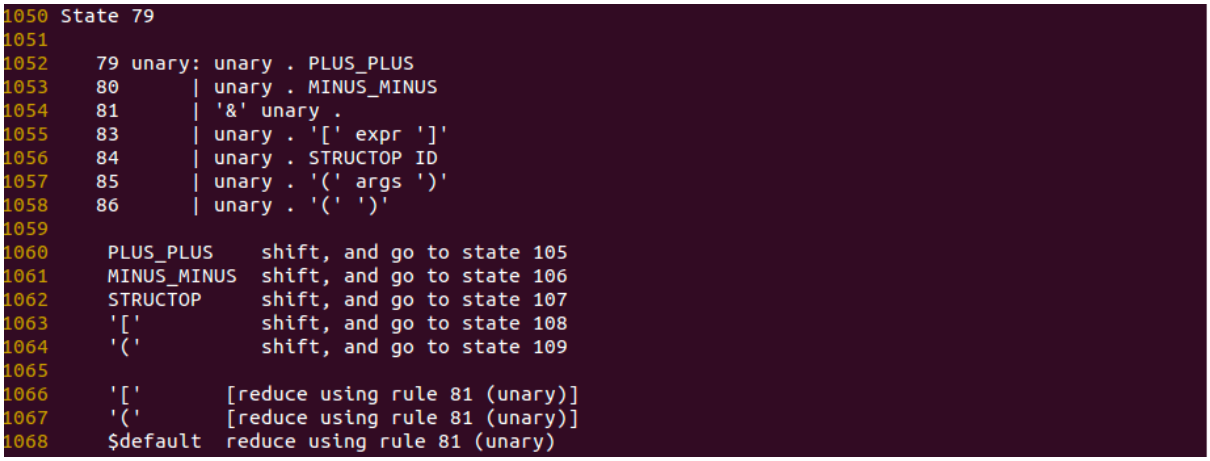
컴파일러의 기초: Project 2 for YACC Programming

2012-11598
민두기

주어진 grammar rule 을 사용하여 컴파일한 결과 subc.output 파일에서 conflicts 가 9 개 존재하는 것을 확인 할 수 있었다.



모두 shift/reduce conflicts 로 해당 state 로 가서 확인한 결과



State 79, 80, 81, 82 의 conflicts 는 !, *, -, & 와 같은 unary operator 가 대괄호나 괄호와 동시에 존재 할 때 생기는 conflicts 였다. 예를들어 &a[3] 이런 경우에 &a 를 unary->& unary 로 reduce 할지 &a[까지 봐서 state 108 로 shift 할지 정해진 규칙이 존재 하지 않아 생기는 conflicts 였다.

이를 해결하기 위해서는 [와 (에 unary 들 보다 더 높은 우선순위를 부여해주면 해결된다. 우선 unary operator 들에게 unary 에 해당하는 우선순위를 부여해주고 그 보다 높은 우선순위로 %right '[' '(' 를 넣어주었다. Right 인 이유는 만약 2-dimensional array 를 지원한다면 오른쪽부터 [ ]를 처리해주어야 하기 때문이다.
또한 '['와 '('는 언제나 높은 우선순위를 가지기 때문에 (ex. 함수, array etc.)이러한 우선순위를 부여해주더라도 전체 grammar 에 영향을 주지 않는다고 판단하였다.
우선순위를 부여해주고 결과를 확인하면 아래 스크린샷과 같이 conflicts 가 해결된 것을 확인 할 수 있다.

```
1043 State 79
1044
1045    79 unary: unary . PLUS_PLUS
1046    80      | unary . MINUS_MINUS
1047    81      | '&' unary .
1048    83      | unary . '[' expr ']'
1049    84      | unary . STRUCTOP ID
1050    85      | unary . '(' args ')'
1051    86      | unary . '(' ')'
1052
1053     PLUS_PLUS    shift, and go to state 105
1054     MINUS_MINUS  shift, and go to state 106
1055     '['          shift, and go to state 107
1056     '('          shift, and go to state 108
1057     STRUCTOP     shift, and go to state 109
1058
1059     $default  reduce using rule 81 (unary)
```

State 142 에 생긴 conflicts 는 Dangling-Else ambiguity 라고 알려진 전형적인 conflicts 였다.

```
2025 State 142
2026
2027    43 stmt: IF '(' test ')' stmt .
2028    44     | IF '(' test ')' stmt . ELSE stmt
2029
2030     ELSE  shift, and go to state 146
2031
2032     ELSE      [reduce using rule 43 (stmt)]
2033     $default  reduce using rule 43 (stmt)
```

IF ELSE 구문을 만났을 때 rule 43 을 사용하여 IF 만 reduce 할지 아니면 ELSE 까지 보고 shift 할 지에 대한 규칙이 존재하지 않아 생기는 shift/reduce conflicts 였다. 예를들어 IF IF ELSE 구문을 만났을 때 IF (IF ELSE) 로 볼 것 인지 IF (IF) ELSE 로 볼 것인지 conflicts 가 존재하게 된다.

이를 해결하기 위해서 ELSE 에 right 우선순위를 부여해주고 %prec 를 사용해 IF 와 IF ELSE 규칙에 %prec ELSE 를 해주었다. 이렇게 해주면 IF 나 IF ELSE 나 같은 우선순위를 가지지만 associativity 가 %right 이기 때문에 IF IF ELSE 같은 구문을 만났을 때 언제나 IF (IF ELSE)로 처리하게 된다. 그 결과 아래 스크린샷처럼 conflicts 가 해결되었다.

```
2010 State 142
2011
2012    43 stmt: IF '(' test ')' stmt .
2013    44     | IF '(' test ')' stmt . ELSE stmt
2014
2015     ELSE  shift, and go to state 146
2016
2017     $default  reduce using rule 43 (stmt)
```