처음 구현은 PDF에 안내되어 있는 대로 그대로 사용하였다. Grammer는 그대로 사용한 채 REDUCE 함수만 적절하게 배치하였고, 연산자들은 처음에 다음과 같이 설정하였다. PDF의 표에 있는 대로 우선순위가 낮은 것부터 associativity에 맞게 분류하였다. 이때 동일한 연산자가 서로 다른 상황에 따라 다양한 associativity와 precedence를 가질 때는 우선순위가 낮은 것을 기준으로 표시하였다. 이미 제시되어 있는 grammer에서 unary -와 *를 명시적으로 !와 같은 우선순위로 만들어주었기 때문에, binary -와 *를 먼저 명시한 것이다.

```
31  %left ','
32  %right '='
33  %left LOGICAL_OR
34  %left LOGICAL_AND
35  %left EQUOP
36  %left RELOP
37  %left '+' '-'
38  %left '*'  '/' '%'
39  %right '!' '&' INCOP DECOP
40  %left STRUCTOP '(' ')' '.'
```

이렇게 했을 때 크게 7개의 conflict warning이 떴지만 종류는 총 2가지였다. 하나는 대괄호와 관련된 것으로 unary 연산자가 배열을 참조하는 연산 앞에 붙어있는 경우로, 원래 선언한 연산자들 중에 대괄호가 없어서 precedence를 판단하지 못한 것이었다.

```
subc.y: warning: shift/reduce conflict on token '[' [-Wcounterexamples]
  Example: '-' unary . '[' expr ']'
  Shift derivation
    unary
    `-> 57: '-' unary
                  `-> 65: unary . '[' expr ']'
  Reduce derivation
    unary
    `-> 65: unary                    '[' expr ']'
              `-> 57: '-' unary .
```

이를 고치기 위해 대괄호 연산자도 연산자 선언에 포함을 시켰고, 배열을 읽는 연산은 그 어떤 연산자보다도 먼저하기 때문에 맨 마지막에 같이 두었다.

```
%left ','
%right '='
%left LOGICAL_OR
%left LOGICAL_AND
%left EQUOP
%left RELOP
%left '+' '-'
%left '*'  '/' '%'
%right '!' '&' INCOP DECOP
%left STRUCTOP '(' ')' '.' '[' ']'
```

2번째는 IF와 IF-ELSE 간에 발생한 것으로, 과제 1에서 나왔던 것처럼 if-if-else 꼴이 나올 때 ambiguous해지는 문제였다.

```
subc.y: warning: shift/reduce conflict on token ELSE [-Wcounterexamples]
  Example: IF '(' expr ')' IF '(' expr ')' stmt . ELSE stmt
  Shift derivation
    stmt
    `-> 32: IF '(' expr ')' stmt
                            `-> 33: IF '(' expr ')' stmt . ELSE stmt
  Reduce derivation
    stmt
    `-> 33: IF '(' expr ')' stmt                        ELSE stmt
                            `-> 32: IF '(' expr ')' stmt .
```

C에서는 else를 언제나 제일 가까운 if에 연산을 시키므로 IF를 IF-ELSE보다 우선권을 주기로 하였고 이에 따라 다음과 같이 만들어서 해결하였다.

```
42  %nonassoc ONLY_IF
43  %nonassoc ELSE
44
```

```
| IF '(' expr ')' stmt %prec ONLY_IF
{
  REDUCE("stmt->IF '(' expr ')' stmt");
}
| IF '(' expr ')' stmt ELSE stmt
{
  REDUCE("stmt->IF '(' expr ')' stmt ELSE stmt");
}
```