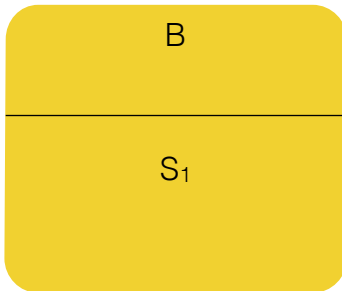


Compiler Scribe Report

Semantic Rule



```

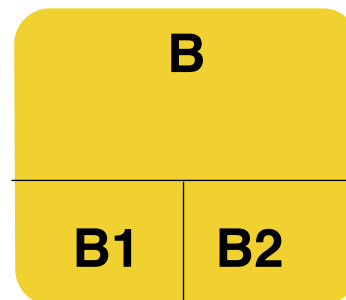
L= new Label()
S.next = L
B.true = new Label()
B.false= S.next
S.code= B.code || Label(B.true) || S1.code
P.code= S.code || Label(L)
id = E.addr
B -> B1||B2 | B1&&B2 | !B1
B -> E1 relop E2
relop -> == | >= | <= | > | <

```

```

if E1 relop E2
    goto B.true
goto B.false
B -> B1 || B2
B1.true = B.True
B1.false= new Label
B.true = B.true
B2.false =B.false

```



```

B.code = B1.code || Label(B1.false) || B2.code
if x>100 || x<150
    x=0

```

`L1 = new Label()`

`S.next = L1`

`L2 = new Label()`

`B.true = L2`

`B.false = S.next`

`B1.true = L2`

`L3 = new Label()`

`B1.false = L3`

`B2.true = L3`

`B2.false = L1`

`B1.code = if (x>100) goto B1.true`

`goto L3`

`B2.code = if (x<150) goto B.true L2`

`goto L1`

`B.code = B1.code || Label(L3) || B2.code`

`= if (x>100) goto L2`

`goto L3`

`= if(x<150) goto L2`

`goto L1`

`S1.code = x = 0`

`S.code = B.code || Label(B.true) || S1.code`

`if(x>100) goto L2`

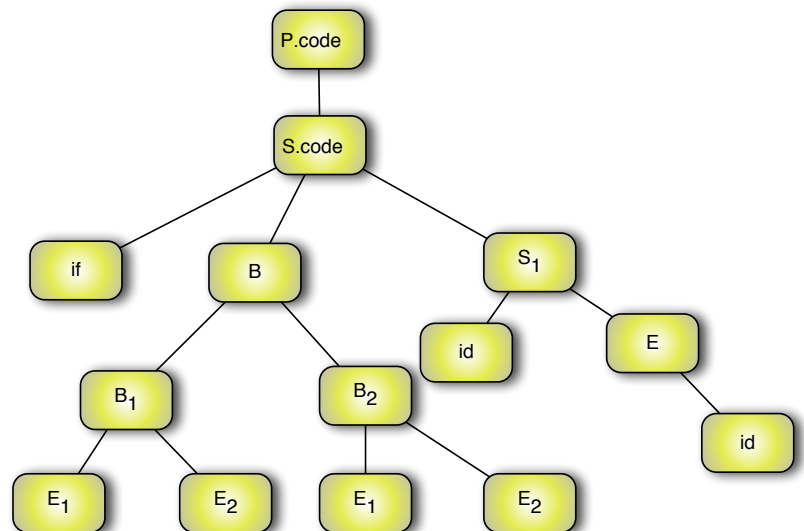
`goto L3`

`L3 : if(x<150) goto L2`

`goto L1`

`L2 : x=0`

`L1 :`



Control Statements

```
if ( x+y >100 && x-y <150)
    x=5
```

$B = B_1 \&\& B_2$

$E = E_1 + E_2$

$B.code = E_1.code \parallel E_2.code$

if $t_1 > t_2$ goto B.true

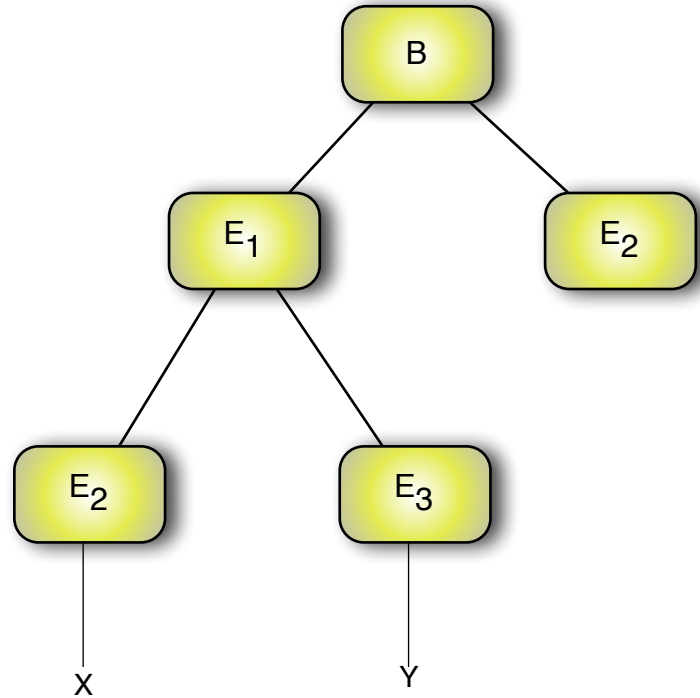
goto B.false

$E_1.code = E_2.code \parallel E_3.code$

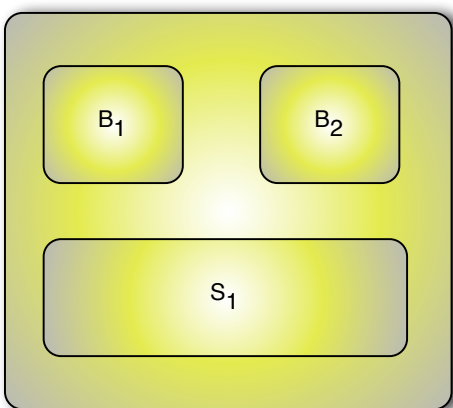
$E_1.addr = t_1$

$E_2.code = t_2 = 100$

$E_2.addr = t_2$



B

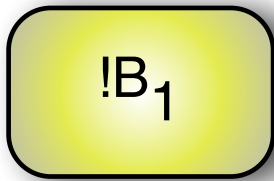


S

```

B1.true = new Label = L3
B1.false = B.false = S.next
B2.true = B.true = L1
B2.false = B.false
B.code = B1.code || Label (B1.true) || B2.code
  
```

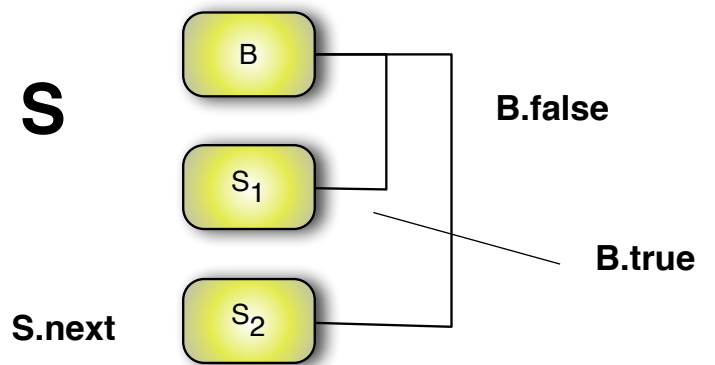
B



B -> !B₁
 B.code = B₁.code
 B₁.false=B.true
 B₁.true=B.false

S -> if (B) S₁
 else S₂

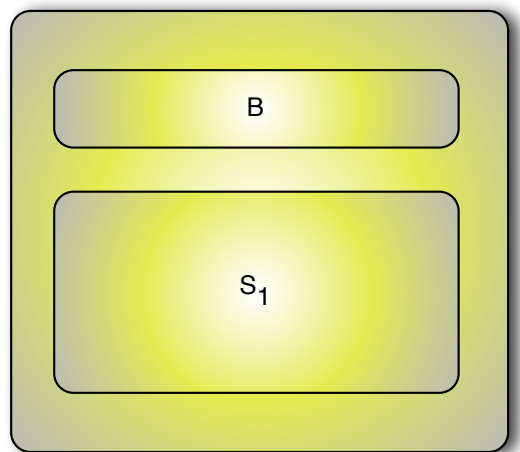
S



B.true= new Label()=L1
 B.false = new Label() =L2
 S₁.next = S₂.next = S.next
 S.code= B.code || Label(B.true) || S₁.code || goto(s,next) || Label(B.false) || S₂.code

S -> while(B) S₁

S



begin= new Label
 B.true = new Label()
 B.false = S.next
 S₁.next=begin
 S.code=Label(begin) || B.code || Label (B.true) || S₁.code || goto (begin)

1. if (x+y > z)
 y=y+1
 else
 y=y-1
2. while (x > y+2)
 y=y+1

Declarations :

P -> D

D -> T id

T -> BC

B -> int | float

C -> e

C -> [num] C

```
int a;  
float b;  
int [10] c;
```