

Control Flow for if, else, while statements.

08 May 2025
Thursday

- We will translate control flow statements such as if-else statements, while statements.
- Translation of control flow statements are tied with Boolean expressions. For writing Boolean expressions, we will need Boolean operators.
- Boolean operators are applied to:
 - Boolean variables (11, & &)
 - Relational expressions with Relational Operators.

$>, <, < >, ==, <=, >=$

Relational Operators.

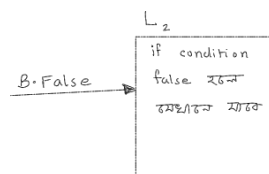
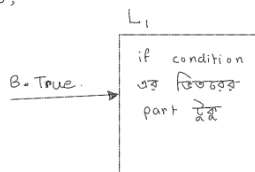
$S \Rightarrow \text{if}(B) S_1$

$S \Rightarrow \text{if}(B) S_1 \text{ else } S_2$

$S \Rightarrow \text{while}(B) S_1$

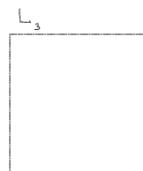
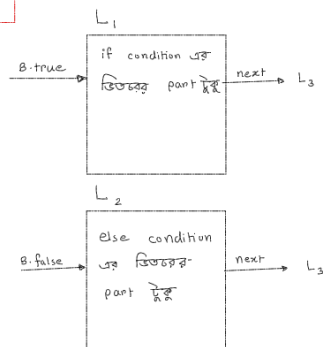
- S हा statement बनता है
- B हा Boolean expression बनता है.

$S \Rightarrow \text{if}(B) S_1$
if () {
..... } statement
}
a = a + 10;
:



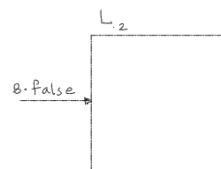
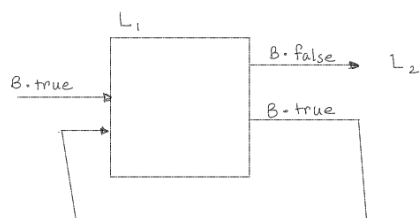
$S \Rightarrow \text{if}(B) S_1 \text{ else } S_2$

if () {
..... }
else {
..... }
}
a = a + 10;
:



$S \Rightarrow \text{while}(B) S_1$

while (a > b) {
..... }
a = a + 10;
:



$S \Rightarrow id = E \quad \{ s.code = E.code \parallel gen(top.get(id.lexeme) '=' E.addr) \}$

$E \rightarrow E_1 + E_2 \quad \{ E.addr = new Temp() \\ E.code = E_1.code \parallel E_2.code \parallel gen(E.addr '=' E_1.addr '+' E_2.addr) \}$

$| - E_1 \quad \{ E.addr = new Temp() \}$

$E.code = E_1.code \parallel gen(E.addr '=' 'minus' E_1.addr) \}$

$| (E_1) \quad \{ E.addr = E_1.addr \}$

$E.code = E_1.code \}$

$| id \quad \{ E.addr = top.get(id.lexeme) \}$

$E.code = '' \}$

$| num \quad \{ E.addr = new Temp() \}$

$E.code = gen(E.addr '=' num.lexval) \}$

$P \rightarrow S \quad \{ s.next = new label() \}$

$P.code = S.code \parallel label(S.next) \}$

$S \rightarrow assign \quad \{ s.code = assign.code \}$

$S \rightarrow if(B) S_1 \quad \{ B.true = new label() \}$

$B.false = S_1.next = S.next$

$S.code = B.code \parallel label(B.true) \parallel S_1.code \}$

$S \rightarrow if(B) S_1 \text{ else } S_2 \quad \{ B.true = new label() \}$

$B.false = new label() \}$

$S_1.next = S_2.next = S.next$

$S.code = B.code \parallel label(B.true) \parallel S_1.code \parallel gen('goto' S.next) \parallel label(B.false) \parallel S_2.code \}$

$S \rightarrow while(B) S_1 \quad \{ begin = new label() \}$

$B.true = new label() \}$

$B.false = S.next$

$S_1.next = begin$

$S.code = label(begin) \parallel B.code \parallel label(B.true) \parallel S_1.code \parallel gen('goto' begin) \}$

$S \rightarrow S_1 S_2 \quad \{ S_1.next = new label() \}$

$S_2.next = S.next$

$S.code = S_1.code \parallel label(S_1.next) \parallel S_2.code \}$

$B \rightarrow B_1 \parallel B_2 \quad \{ B_1.true = B.true \}$

$B_1.false = new label() \}$

$B_2.true = B.true$

$B_2.false = B.false$

$B.code = B_1.code \parallel label(B_1.false) \parallel B_2.code \}$

$B \rightarrow B_1 \&\& B_2 \quad \{ B_1.true = new label() \}$

$B_1.false = B.false$

$B_2.true = B.true$

$B_2.false = B.false$

$B.code = B_1.code \parallel label(B_1.true) \parallel B_2.code \}$

$B \rightarrow !B_1 \quad \{ B_1.true = B.false \}$

$B_1.false = B.true$

$B.code = B_1.code \}$

$B \rightarrow E_1 \text{ rel } E_2 \quad \{ B.code = E_1.code \parallel E_2.code \}$

$\parallel gen('if' E_1.addr \text{ relop } E_2.addr 'goto' B.true) \}$

$\parallel gen('goto' B.false) \}$

$B \rightarrow true \quad \{ B.code = gen('goto' B.true) \}$

$B \rightarrow false \quad \{ B.code = gen('goto' B.false) \}$

■ Draw the syntax tree for the following input string:

if (x+k < 100 || x+k > 200 && x!=y)

x = 0;

y = x + 10;

