ToC

Control Flow

Boolean expression are the building block for influencing the flow of a program. They are manipulated to:

- Alter the flow of control – like in `if` $(E)S$
- Compute logical values

Two different approaches to evaluation:

- Eager
- Lazy

Short-Circuit Code

- Boolean operators ||, && and ! translate into jumps
- The operators do not appear on the code
- The value of a boolean expression is represented by a position in the code sequence

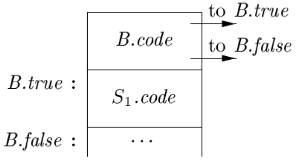
```
if (x < 100 || x > 200 && x != y) x = 0;
```

is translated to

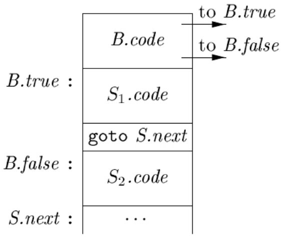
```
        if x < 100 goto L2
        ifFalse x > 200 goto L1
        ifFalse x != y goto L1
L2:     x = 0
L1:     ...
```

Translation using jumping code

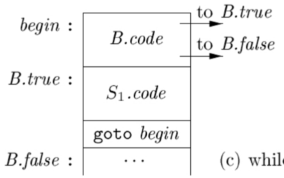
- Statements and Boolean Expressions have a synthesised attribute *code* that is a string containing the translated code
- Boolean Expressions have two inherited attributes *true* and *false* representing labels to which the control flows if the expression is true or false, respectively



(a) if



(b) if-else



(c) while

Control Flow – commands

P	$\rightarrow$	S	S.next=newlabel(), P.code = S.code || label(S.next)
S	$\rightarrow$	**assign**	S.code=**assign**.code

S	$\rightarrow$	**if** (B) S_1	B.true=newlabel(), B.false=S_1.next=S.next
			S.code=B.code||label(B.true)||S_1.code

S	$\rightarrow$	**if** (B) S_1 **else** S_2	B.true=newlabel(), B.false=newlabel()
			S_1.next=S_2.next=S.next
			S.code=B.code||label(B.true)||S_1.code
			||gen('`goto`' S.next)|| label(B.false)|| S_2.code

S	$\rightarrow$	**while** (B) S_1	begin=newlabel(), B.true=newlabel()
			B.false=S.next, S_1.next=begin
			S.code=label(begin)||B.code||label(B.true)||S_1.code
			||gen('`goto`' begin)

S	$\rightarrow$	S_1 S_2	S_1.next=newlabel(), S_2.next=S.next S.code = S_1.code||label(S_1.next)||S_2.code

Control Flow – Boolean expressions

B	$\rightarrow$	$B_1 \| \| B_2$	B_1.true=B.true B_1.false=newlabel() B_2.true = B.true B_2.false=B.false B.code = B_1.code || label(B_1.false)|| B_2.code
B	$\rightarrow$	$B_1 \&\& B_2$	B_1.true=newlabel() B_1.false=B.false B_2.true = B.true B_2.false=B.false B.code = B_1.code || label(B_1.true)|| B_2.code
B	$\rightarrow$	E_1**rel**E_2	B.code = E_1.code||E_2.code || gen('`if`' E_1.addr **rel**.op E_2.addr '`goto`' B.true) || gen('`goto`' B.false)
B	$\rightarrow$	**true**	B.code=gen('`goto`' B.true)
B	$\rightarrow$	**false**	B.code=gen('`goto`' B.false)

Control Flow – commands

Let's translate the following program:

```
if (x != y && x == z) x = y + z;
```