**Compiling If-Then-Else**

[Compiled Factorial Example](http://lara.epfl.ch/w/cc09:compiled_factorial_example)

Assume that translation of c, denoted [ [ c ] ], produces a boolean value on top of the stack

Then:

[[ If (c) sThen else sElse ]] =

[[ c ]]

if\_eq nElse

[[ sThen ]]

goto nAfter

nElse: [[ sElse ]]

nAfter:

**Example**

Translating

public static int test(boolean b) {

if (b)

return 13;

else

return 42;

}

Bytecode:

public static int test(boolean);

Code:

0: iload\_0

1: ifeq 7

4: bipush 13

6: ireturn

7: bipush 42

9: ireturn

**Question**

When generating

if\_eq nAfter

how do we know the offset nAfter?

1. generate symbolic labels
2. compute the values of labels when all instruction sizes are known (note: fixed size of jump instructions, regardless of their destination)
3. emit the code using appropriate values of labels