CMPE 152: Compiler Design

November 7 Lab

Department of Computer Engineering San Jose State University

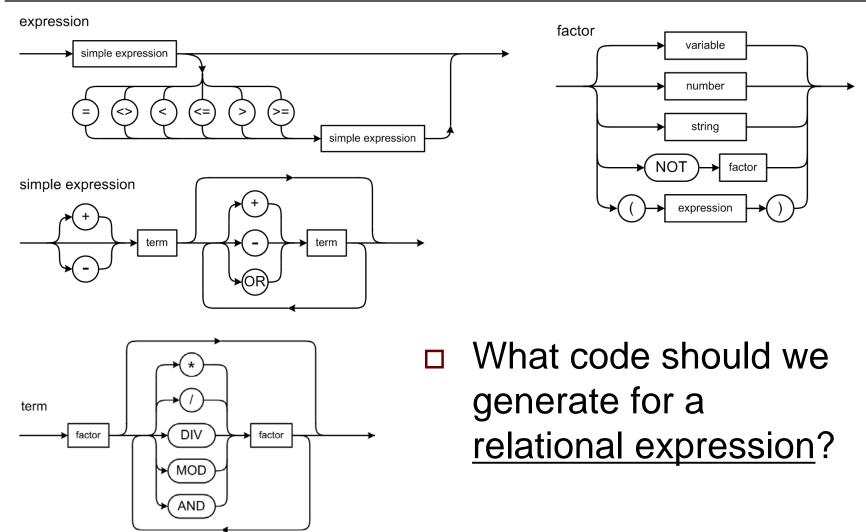


Fall 2017 Instructor: Ron Mak

www.cs.sjsu.edu/~mak



Expression Syntax Diagrams



Relational Expressions

- □ Suppose i and j are local integer variables, and that:
 - i → slot #0
 j → slot #1
- 0 represents false and1 represents true.
- For the expression i < j leave either 0 or 1 on top of the operand stack:</p>

```
L003:
iconst_1 ; push true
L004:
```

Your code generator also needs to emit labels.



Relational Expression Code Template

Code to evaluate the first operand

iload 0

Code to evaluate the second operand

iload 1

Compare-and-branch-if-true to *true-label* instruction

if_icmplt L003

False code iconst 0

goto L004

iconst_0

True code true-label:

L003: iconst_1

next-label: L004:



goto *next-label*

iconst 1

Assignment Statement Code Template

□ The code template for an assignment statement to a <u>local variable</u> <variable> := <expression>



Where *x* is **i**, **1**, **f**, or **d** depending on the <u>type</u> of the computed value of <expression>.

You can generate a shortcut store instruction such as istore_3 (3 is a slot number) whenever possible.



IF Statement Code Templates

Code to evaluate the boolean expression

ifeq next-label

Code for the THEN statement

next-label:

- The code that evaluates the boolean expression leaves either 0 (false) or 1 (true) on top of the operand stack.
 - ifeq branches if [TOS] is 0 (the expression is false)

Code to evaluate the boolean expression

ifeq false-label

Code for the THEN statement

goto next-label false-label:

Code for the ELSE statement

next-label:



Example: IF Statement

Fall 2017: November 7

```
if_icmplt
                                                                  L002
                                                iconst_0
                                                         L003
                                                goto
PROGRAM IfTest;
                                       L002:
                                                iconst 1
VAR
                                       L003:
    i, j, t, f : integer;
                                                ifeq
                                                         L001
                                                sipush
                                                         300
BEGIN {IF statements}
                                                                  iftest/t I
                                                putstatic
                                       L001:
    IF i < j THEN t := 300;
                                                getstatic
                                                                  iftest/i I
                                                getstatic
                                                                  iftest/j I
                                                if_icmpeq
                                                                  L005
    IF i = j THEN t := 200
                                                iconst 0
               ELSE f := -200;
                                                         L006
                                                goto
                                       L005:
END.
                                                iconst 1
                                       L006:
                                                ifeq
                                                         L007
                                                sipush
                                                         200
                                                putstatic
                                                                  iftest/t I
                                                goto
                                                         L004
                                       L007:
                                                sipush
                                                         200
                                                ineg
                                                                  iftest/f I
                                                putstatic
       Computer Engineering Dept.
                                       L004:
```

getstatic

getstatic

iftest/i I

iftest/j I

Looping Statement Code Template

loop-label:

Code for statements before the test

Code to evaluate the boolean test expression

ifne next-label

Code for statements after the test

goto loop-label next-label:

- The code that
 evaluates the
 boolean expression
 leaves either
 0 (false) or 1 (true)
 on top of the
 operand stack.
 - ifne branches if [TOS] is not 0 (the expression value is true)
- There might not be any code before or after the test.



Example: Newton's Square Root Function

```
FUNCTION sqrt(x : real) : real;
VAR
  #1 i : integer;
  #2 root : real;
BEGIN
    i := 0;
    root := x;
    REPEAT
        root := (x/root + root)/2;
        i := i + 1;
    UNTIL i > 10;
    sqrt := root;
END;
```

```
iconst_0
istore_1 ; i := 0
fload_0
fstore_2 ; root := x
```

```
fstore 2
L000:
   fload 0
              ; x
   fload 2
              ; root
   fdiv
              ; /
   fload 2
              ; root
   fadd
              ; +
   fconst 2
              : 2.0
   fdiv
   fstore 2 ; ==> root
   iinc 1 1
              ; i := i + 1;
              ; i
   iload 1
   bipush 10
              ; 10
   if icmpgt L001; if i > 10 goto L001
                   : false
   iconst 0
   goto L002
L001:
    iconst_1
                   ; true
L002:
   ifne L001
             ; if true goto L003
   goto L000
L003:
```

Example: For Statement

```
getstatic
                                                                       fortest/i I
PROGRAM ForTest;
                                                    putstatic
                                                                       fortest/k I
VAR
                                           L001:
     j, k, n : integer;
                                                    getstatic
                                                                       fortest/k I
                                                    iconst 5
BEGIN {FOR statements}
                                                    if_icmpgt
                                                                       L003
                                                    iconst 0
                                                                      This is code
                                                    goto
                                                             L004
                                                                      emitted for a
                                           L003:
     FOR k := 1 \text{ TO } 5 DO BEGIN
                                                    iconst 1
                                                                      general > test.
         n := k;
                                           L004:
                                                                      It can be much
     END;
                                                    ifne
                                                             L002
                                                                      improved!
                                                                       fortest/k I
                                                    getstatic
END.
                                                                       fortest/n I
                                                    putstatic
                                                                       fortest/k I
                                                    getstatic
    Remember that program variables
                                                    iconst_1
  are translated into Jasmin static fields,
                                                    iadd
 and so they have names, not slot numbers.
                                                                       fortest/k I
                                                    putstatic
                                                             L001
                                                    goto
```



CMPE 152: Compiler Design Lab © R. Mak

L002:

SELECT Statement

Code to evaluate the SELECT expression

lookupswitch

For each SELECT value v_i:

*v*_i: *branch-label*_j

default: next-label

LOOKUPSWITCH instruction

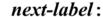
For each branch-labeli:

branch-label_i:

Code for the j th statement

goto next-label

SELECT branch statements





Example: CASE Statement

```
(#0 (#1)
VAR i, j : integer;
                                                           iload 0 ; i
                                                           lookupswitch
   CASE i OF
                                                              100: L010
      100,105: j := 1000;
                                                              105: L010
      200,256,282: j := 2000;
                                                              200: L020
   END
                                                              256: L020
                                                              282: L020
Code to evaluate the SELECT expression
                                                             default: L099
                                                    L010:
     lookupswitch
                                                           sipush 1000
      For each SELECT value vi:
                                LOOKUPSWITCH
                                                           istore 1 ; j := 1000
        v<sub>i</sub>: branch-label<sub>i</sub>
                                instruction
                                                           goto L099
        default: next-label
                                                    L020:
For each branch-labeli:
                                                           sipush 2000
branch-label<sub>i</sub>:
                                                           istore 1 ; j := 2000
                                SELECT branch
                                                           goto L099
     Code for the j <sup>th</sup> statement
                                statements
                                                    L099:
      goto next-label
```



next-label: