

W2 - IPS

Nikolaj Gade (qhp695)

May 2022

Task 1

1) Intermediate code

See appendix A for the full generation.

```
1 LABEL label1
2 t4 := v1
3 t5 := 1
4 IF t4 == t5 THEN label3 ELSE label4
5 LABEL label4
6 t0 := 1
7 place2 := v0
8 place3 := v1
9 t1 := place2 / place3
10 IF t0 < t1 THEN label2 ELSE label3
11 LABEL label2
12 t0 := v1
13 t3 := v0
14 IF t2 < t3 THEN label6 ELSE label7
15 LABEL label5
16 place4 := v0
17 place5 := v1
18 place0 := place4 - place5
19 v0 := place0
20 GOTO label7
21 LABEL label6
22 place6 := v1
23 place7 := v0
24 place1 := place6 - place7
25 v1 := place1
26 GOTO label5
27 LABEL label7
28 GOTO label1
29 LABEL label3
```

2) MIPS code

```
1 label1:
2     mov t4, v1
3     addi t5, R0, 1
4     beq t4, t5, label3
5 label4:
6     addi t0, R0, 1
7     mov place2, v0
8     mov place3, v1
9     div place2, place3
10    mflo t1
11    addi t1, t1, 1
12    slt rd, t0, t1
13    beq rd, R0, label3
14 label2:
15    mov t0, v1
16    mov t3, v0
17    slt rd, t2, t3
18    bne rd, R0, label6
19    j label7
20 label5:
21    mov place4, v0
22    mov place5, v1
23    sub place0, place4, place5
24    mov v0, place0
25    j label7
26 label6:
27    mov place6, v1
28    mov place7, v0
29    sub place1, place6, place7
30    mov v1, place1
31    j label5
32 label7:
33    j label1
34 label3:
```

Task 2

1) $z := x \geq y$

```
1 sub v, y, x
2 slti z, v, 1
```

2) $w := !z$

```
1 slt w, R0, z
```

3) $z := x \geq y, w := !z$

```
1 slt w, x, y
```

Task 3

a)

```
1 bool* y = (bool*)malloc(n*4);
2 int i = 0;
3 while (i < n) {
4     ne = myop(ne, x[i*4]);
5     y[i*4] = ne;
6     i++;
7 }
```

A Intermediate code generation

Code1, $Trans_{Cond}(Cond_0, label12, label13, vtable, ftable)$

!(b == 0) && (1 < a/b)

$Cond_1 \&\& Cond_2$

label14 = newlabel()

code3 = $Trans_{Cond}(Cond_1, label14, label13, vtable, ftable)$

code4 = $Trans_{Cond}(Cond_2, label12, label13, vtable, ftable)$

```
1 code3++
2 [LABEL label14]++
3 code4
```

Code2, $Trans_{Stat}(Stat_1, vtable, ftable)$

if b < a then { a := a - b } else { b := b - a }

if $Cond_3$ then $Stat_2$ else $Stat_3$

label15 = newlabel()

label16 = newlabel()

label17 = newlabel()

code5 = $Trans_{Cond}(Cond_3, label16, label17, vtable, ftable)$

code6 = $Trans_{Stat}(Stat_2, vtable, ftable)$

code7 = $Trans_{Stat}(Stat_3, vtable, ftable)$

```
1 code5++
2 [LABEL label15]++
3 code6++
4 [
5     GOTO label17
6     LABEL label16
7 ]++
8 code7++
9 [
10    GOTO label15
11    LABEL label17
12 ]
```

Code3, $Trans_{Cond}(Cond_1, label14, label13, vtable, ftable)$

!(b == 0)

!($Cond_4$)

code8 = $Trans_{Cond}(Cond_4, label13, label14, vtable, ftable)$

```
1 code8
```

code4, $Trans_{Cond}(Cond_2, label12, label13, vtable, ftable)$

1 < a/b

$Exp_0 < Exp_1$

t0 = *newvar*()

t1 = *newvar*()

code9 = $Trans_{Exp}(Exp_0, vtable, ftable, t0)$

code10 = $Trans_{Exp}(Exp_1, vtable, ftable, t1)$

```
1 code9++  
2 code10++  
3 [IF t0 < t1 THEN label12 ELSE label13]
```

code5, $Trans_{Cond}(Cond_3, label16, label17, vtable, ftable)$

b < **a**

$Exp_2 < Exp_3$

t2 = *newvar*()

t3 = *newvar*()

code11 = $Trans_{Exp}(Exp_2, vtable, ftable, t2)$

code12 = $Trans_{Exp}(Exp_3, vtable, ftable, t3)$

```
1 code11++  
2 code12++  
3 [IF t2 < t3 THEN label16 ELSE label17]
```

code6, $Trans_{Stat}(Stat_2, vtable, ftable)$

a := **a** - **b**

a := Exp_4

place0 = *newvar*()

code13 = $Trans_{Exp}(Exp_4, vtable, ftable, place0)$

```
1 code13++  
2 [v0 := place0]
```

code7, $Trans_{Stat}(Stat_3, vtable, ftable)$

b := **b** - **a**

b := Exp_5

place1 = *newvar*()

code14 = $Trans_{Exp}(Exp_5, vtable, ftable, place1)$

```
1 code14++  
2 [v1 := place1]
```

code8, $Trans_{Cond}(Cond_4, label13, label14, vtable, ftable)$

b == 0

Exp_6 == Exp_7

t4 = newvar()

t5 = newvar()

code15 = $Trans_{Exp}(Exp_6, vtable, ftable, t4)$

code16 = $Trans_{Exp}(Exp_7, vtable, ftable, t5)$

```
1 code15++
2 code16++
3 [IF t4 == t5 THEN label3 ELSE label4]
```

code9, $Trans_{Exp}(Exp_0, vtable, ftable, t0)$

1

1

```
1 [t0 := 1]
```

code10, $Trans_{Exp}(Exp_1, vtable, ftable, t1)$

a/b

Exp_8 / Exp_9

place2 = newvar()

place3 = newvar()

code17 = $Trans_{Exp}(Exp_8, vtable, ftable, place2)$

code18 = $Trans_{Exp}(Exp_9, vtable, ftable, place3)$

```
1 code17++
2 code18++
3 [t1 := place2 / place3]
```

code11, $Trans_{Exp}(Exp_2, vtable, ftable, t2)$

b

b

```
1 [t0 := v1]
```

code12, $Trans_{Exp}(Exp_3, vtable, ftable, \mathbf{t3})$

a

a

1 [t3 := v0]

code13, $Trans_{Exp}(Exp_4, vtable, ftable, \mathbf{place0})$

a - b

$Exp_{10} - Exp_{11}$

place4 = newvar()

place5 = newvar()

code19 = $Trans_{Exp}(Exp_{10}, vtable, ftable, \mathbf{place4})$

code20 = $Trans_{Exp}(Exp_{11}, vtable, ftable, \mathbf{place5})$

1 code19++
2 code20++
3 [place0 := place4 - place5]

code14, $Trans_{Exp}(Exp_5, vtable, ftable, \mathbf{place1})$

b - a

$Exp_{12} - Exp_{13}$

place6 = newvar()

place7 = newvar()

code21 = $Trans_{Exp}(Exp_{12}, vtable, ftable, \mathbf{place6})$

code22 = $Trans_{Exp}(Exp_{13}, vtable, ftable, \mathbf{place7})$

1 code21++
2 code22++
3 [place1 := place6 - place7]

code15, $Trans_{Exp}(Exp_6, vtable, ftable, \mathbf{t4})$

b

b

1 [t4 := v1]

code16, $Trans_{Exp}(Exp_7, vtable, ftable, \mathbf{t5})$

\emptyset

\emptyset

1 [t5 := 1]

code17, $Trans_{Exp}(Exp_8, vtable, ftable, \mathbf{place2})$

a

a

1 [place2 := v0]

code18, $Trans_{Exp}(Exp_9, vtable, ftable, \mathbf{place3})$

b

b

1 [place3 := v1]

code19, $Trans_{Exp}(Exp_{10}, vtable, ftable, \mathbf{place4})$

a

a

1 [place4 := v0]

code20, $Trans_{Exp}(Exp_{11}, vtable, ftable, \mathbf{place5})$

b

b

1 [place5 := v1]

code21, $Trans_{Exp}(Exp_{12}, vtable, ftable, \mathbf{place6})$

b

b

1 [place6 := v1]

code22, $Trans_{Exp}(Exp_{13}, vtable, ftable, place7)$

a

a

1 [place7 := v0]