W3 - IPS

Nikolaj Gade (qhp695)

May 2022

Task 1

1) Intermediate code

See appendix A for the full generation.

```
LABEL label1
t4 := v1
t5 := 1
IF t4 == t5 THEN label3 ELSE label4
LABEL label4
t0 := 1
place2 := v0
place3 := v1
t1 := place2 / place3
IF t0 < t1 THEN label2 ELSE label3
LABEL label2
t0 := v1
t3 := v0
IF t2 < t3 THEN label6 ELSE label7
LABEL label5
place4 := v0
place5 := v1
place0 := place4 - place5
v0 := place0
GOTO label7
LABEL label6
place6 := v1
place7 := v0
place1 := place6 - place7
v1 := place1
GOTO label5
LABEL label7
GOTO label1
LABEL label3
```

2) MIPS code

```
label1:
  mov t4, v1
  addi t5, R0, 1
  beq t4, t5, label3
label4:
  addi t0, R0, 1
  mov place2, v0
  mov place3, v1
  div place2, place3
  mflo t1
  addi t1, t1, 1
  slt rd, t0, t1
  beq rd, R0, label3
label2:
  mov t0, v1
  mov t3, v0
  slt rd, t2, t3
  bne rd, R0, label6
  j label7
label5:
  mov place4, v0
  mov place5, v1
  sub place0, place4, place5
  mov v0, place0
  j label7
label6:
  mov place6, v1
  mov place7, v0
  sub place1, place6, place7
  mov v1, place1
  j label5
label7:
  j label1
label3:
```

Task 2

1) `z := x >= y`

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

2) `w := !z`

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

3) `z := x >= y, w := !z`

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

Task 3

a)

```
int* y = malloc(n);
int   i = 0;
while (i < n) {
  ne = myop(ne, x[i]);
  y[i] = ne;
  i++;
}
```

b)

```
lw    len,  0(x)                // Loads the length of x from the first word in the list

// dynalloc from group project
move  regy, HP                  // Sets y to the start of the non-allocated memory
addi  tmp,  len,  3             // Adds 3 and rounds down, to line up with the 4-byte
    boundary
sra   tmp,  tmp,  2
sll   tmp,  tmp,  2
addi  tmp,  tmp,  4             // Adds space for the length of the array
add   HP,   HP,   tmp           // Moves HP
sw    len,  0(regy)             // Saves the length of the array in the first word

addi  i,    R0,   0             // int i = 0
loop_beg:
  bge   i,    len,  loop_end    // while (i < n) {..
  add   xi,   x,    i           // x[i]
  add   yi,   regy, i           // y[i]
  MIPS.CALL(myop, (ne, xi), ne) // ne = myop(ne, x[i]);
  sw    ne,   0(yi)             // y[i] = ne;
  addi  i,    i,    1           // i++;
  j loop_beg
loop_end:
```

A Intermediate code generation

Code1, $Trans_{Cond}(Cond_0, \texttt{label2}, \texttt{label3}, vtable, ftable)$

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

$Cond_1$ && $Cond_2$

label4 = $newlabel()$

code3 = $Trans_{Cond}(Cond_1, \texttt{label4}, \texttt{label3}, vtable, ftable)$

code4 = $Trans_{Cond}(Cond_2, \texttt{label2}, \texttt{label3}, vtable, ftable)$

```
code3++
[LABEL label4]++
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 $Stat_3$

label5 = $newlabel()$

label6 = $newlabel()$

label7 = $newlabel()$

code5 = $Trans_{Cond}(Cond_3, \texttt{label6}, \texttt{label7}, vtable, ftable)$

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

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

```
code5++
[LABEL label5]++
code6++
[
  GOTO label7
  LABEL label6
]++
code7++
[
  GOTO label5
  LABEL label7
]
```

Code3, $Trans_{Cond}(Cond_1, \texttt{label4}, \texttt{label3}, vtable, ftable)$

`!(b == 0)`

!($Cond_4$)

code8 = $Trans_{Cond}(Cond_4, \texttt{label3}, \texttt{label4}, vtable, ftable)$

```
code8
```

---

code4, $Trans_{Cond}(Cond_2, \texttt{label2}, \texttt{label3}, vtable, ftable)$

`1 < a/b`

Exp_0 `<` Exp_1

`t0` $= newvar()$

`t1` $= newvar()$

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

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

```
code9++
code10++
[IF t0 < t1 THEN label2 ELSE label3]
```

---

code5, $Trans_{Cond}(Cond_3, \texttt{label6}, \texttt{label7}, vtable, ftable)$

`b < a`

Exp_2 `<` Exp_3

`t2` $= newvar()$

`t3` $= newvar()$

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

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

```
code11++
code12++
[IF t2 < t3 THEN label6 ELSE label7]
```

---

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

`a := a - b`

`a :=` Exp_4

`place0` $= newvar()$

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

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

---

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

`b := b - a`

`b :=` Exp_5

`place1` $= newvar()$

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

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

---

code8, $Trans_{Cond}(Cond_4, \texttt{label3}, \texttt{label4}, vtable, ftable)$

`b == 0`

Exp_6 `==` Exp_7

`t4` $= newvar()$

`t5` $= newvar()$

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

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

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

---

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

`1`

`1`

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

---

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

`a/b`

Exp_8 `/` Exp_9

`place2` $= newvar()$

`place3` $= newvar()$

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

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

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

---

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

`b`

`b`

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

---

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

a

a

```
[t3 := v0]
```

---

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

a - b

Exp_{10} - Exp_{11}

place4 $= newvar()$

place5 $= newvar()$

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

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

```
code19++
code20++
[place0 := place4 - place5]
```

---

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

b - a

Exp_{12} - Exp_{13}

place6 $= newvar()$

place7 $= newvar()$

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

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

```
code21++
code22++
[place1 := place6 - place7]
```

---

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

b

b

```
[t4 := v1]
```

---

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

0

0

```
[t5 := 1]
```

---

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

a

a

```
[place2 := v0]
```

---

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

b

b

```
[place3 := v1]
```

---

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

a

a

```
[place4 := v0]
```

---

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

b

b

```
[place5 := v1]
```

---

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

b

b

```
[place6 := v1]
```

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

a

a

```
1 [place7 := v0]
```