Homework 5

posted Mar. 9, due Mar. 22 at the beginning of the class (no late turn-ins will be accepted).

Please put a cover page over the homework. The grade will be written on the second page.

1. (20 pts) Exercise 7.2

   a. Ex(BINOP(+,MEM(a.unEx( )), CONST(5))

   b.
   ```
   Nx(
     MOVE(
       MEM (
         BINOP (+,
                a.unEx(),
                BINOP (*,
                       BINOP(+,
                              i.unEx(),
                             CONST(1)),
                       CONST(w))
                )
         ),
   CONST(0))
   ```

   c.
   ```
   Nx(
   LABEL(s);
   RelCx(<, a.unEx( ), CONST(0)).unCx(t,f);
   LABEL(t);
   MOVE(MEM(a.unEx())), BINOP(+,  MEM(a.unEx()), CONST(1)));
   JUMP(s);
   LABEL(f);
   )
   ```
   Here I use ‘;’ to represent sequence. One can also replace the RelCx(<,…) with a CJUMP as long as the semantics is not changed.

   d. Ex(RelCx(<, a.unEx(), b.unEx()).unEx())

   One can also expand RelCx into a sequence, resulting in the following

   ```
   Ex(
     ESEQ(
       (MOVE(TEMP(r), CONST(0));
        CJUMP(<, a.unEx(), b,unEx(), t, f);
        LABEL(t);
        MOVE(TEMP(r), CONST(1));
        LABEL(f);
   ```

```
            ),
            TEMP(r)
          )
        )

e.  Nx(
    MOVE(MEM(a.unEx( )),
            BINOP(+,x.unEx(), y.unEx())))
f.  Nx(
    RelCx(<, a.unEx(), b.unEx).unCx(t,f);
    LABEL(t);
    MOVE(MEM(c.unEx()), MEM(a.unEx());
    LABEL(f);
    MOVE(MEM(c.unEx()), MEM(b.unEx());
  )
```

2. (20 pts) Short-circuit evaluation of boolean expressions

When an IF condition or a While loop condition is a Boolean expression, e.g. a && b, often the truth value of that expression can be determined without evaluating all components of the expression. For example, if a is false, then we know a && b must be false without even knowing the truth value of b. The branch operation can take place as soon as the branch condition can be determined. This is called short-circuit evaluation of Boolean expression.

Please convert the following While loop into intermediate code using the idea of short-circuit boolean evaluation.

```
while ( (a< d) || (e < f) )  t = t + 1;
```

Introduce labels and temporaries as needed.

Note that the symbol “;” is used to represent these are SEQs of statements.

```
LABEL(loop);
CJUMP(<, a.unEx(), d.unEx(), L1, L2);
LABEL(L2);
CJUMP(<, e.unEx(), f.unEx(), L1, done);
LABEL(L1);
MOVE(TEMP(t), BINOP(+,TEMP( t), CONST(1));
JUMP(NAME(loop));
LABEL(done)
```