University of Cape Town

Department of Computer Science

COURSE : CSC305H

MODULE : COMPILERS

TEST : 2 (SUPP)

DATE : 17 AUGUST 2005

TOTAL MARKS : 35

INSTRUCTIONS TO STUDENTS

- Answer ALL questions

Question One – Symbol Tables and Activation Records [10]

1. Draw the stack of activation records corresponding to the following C-like program when it is at "breakpoint". [4] (Assume static chains and include all parameters).

```
main {
   int x;
   sub SUBA {
      int y;
      sub SUBB {
         int z;
         sub SUBC {
            //breakpoint
         }
         SUBC;
      }
      SUBB;
   }
   SUBA;
}
```

2. Nested subprograms can require saving and restoring of registers used to pass parameters, but this save/restore operation does not always have to be done. Discuss 3 circumstances under which a save/restore of parameter-passing registers is not necessary even though subprograms are nested. [3]

3. Discuss 3 circumstances under which it is necessary to use memory to pass parameters instead of just using registers. [3]

Question Two – Intermediate Representation [10]

1. Discuss 2 disadvantages of intermediate representations. [3]

2. Assuming the IR tree language in the attached pages, convert the following program fragment to an equivalent IR tree. (Assume x/y are stack frame variables at offset k0/k1 from the frame pointer special temporary fp) Provide the final tree and do not use the Nx/Cx/Ex expression types/objects. [4]

```
while (x+1<y) { x=x+1; y=y-1; }
```

3. Convert the following tree into its canonical form by applying transformations from the attached list. Show the result after each transformation. [3]

MOVE (ESEQ (LABEL L1, ESEQ (LABEL L2, TEMP a)), CONST 5)

Question Three – Instruction Selection [8]

1. What is the difference between an optimal and optimum tiling? Give one example of an algorithm in each class, and state what the Big-O complexity of each algorithm is. [4]

2. Using the attached instruction set, apply the Maximal Munch tiling algorithm to the following IR tree. Show the tiled tree and list the instructions generated. [4]

MOVE (MEM (CONST a), MEM (PLUS (CONST b, CONST c)))

Question Four – Register Allocation [7]

1. Use the iterative liveness analysis algorithm to calculate the live-in and live-out sets for each of the following statements in a program. Show succ, use, def, out and in sets. [7]

```
if ( x > 1 )
   then y = x * x;
   else y = ( 1 / x ) * ( 1 / x );
return y+1;
```

Hint: The relevant formulae are:

$$out[n] = \bigcup_{s \in succ[n]} in[s]$$

$$in[n] = use[n] \cup (out[n] - def[n])$$