University of Cape Town

Department of Computer Science

COURSE : CSC305H

MODULE : COMPILERS

TEST : FINAL EXAMINATION

DATE : 2005

TOTAL MARKS : 25

INSTRUCTIONS TO STUDENTS

- Answer ALL questions

Question One – Symbol Tables and Activation Records [6]

What does the scope of a variable refer to? Is it necessary to have more than one scope in a program? Explain why or why not. [3]

It refers to the places where the variable can be used/accessed. [1]

No. A program could be written such that all variables/names are globally accessible. [2]

What is an activation record? Are stack-based activation records always necessary? Discuss why or why not. [3]

An activation record refers to the layout of data needed to support the invocation of a subprogram. [1]

No. If there is no recursion, only one activation record if needed per subprogram and these can be fixed. [2]

Question Two – Intermediate Representation [6]

Assuming the IR tree language in the attached pages, convert the following program fragment to an equivalent IR tree. (Assume a/b/c/d/e are stack frame variables at offset k0/k1/k2/k3/k4 from the frame pointer special temporary fp) Provide the final tree and do not use the Nx/Cx/ Ex expression types/objects. [4]

```
a=b+(c*d)-e;
```

```
MOVE (MEM (BINOP (PLUS,FP,k0)), BINOP (MINUS, BINOP (PLUS,
MEM(BINOP(PLUS,FP,k1)),BINOP(TIMES,MEM(BINOP(PLUS,FP,k2)),MEM(
BINOP(PLUS,FP,k3)))),MEM(BINOP(PLUS,FP,k4)))
```

Marks: PLUS[1], MINUS[1], TIMES[1], MOVE[1]

When canonicalising IR trees (using the transformations in the attached appendix), ESEQ instructions are eventually eliminated – explain how this happens. [2]

Expression transformations move ESEQ nodes to the top of the tree, statement transformations then convert ESEQs at the top level into SEQs.

Question Three – Instruction Selection [8]

In a CISC machine, it may be naïve to assume that every tile in an instruction set has an identical cost. Explain what factors could affect the cost of tiles. [2]

size of the instruction code [1], execution speed [1], memory vs register access [1]

Some machines have fewer registers than others – how is this handled during instruction selection? [2]

Allocate temporaries liberally [1] and leave the register allocation to a later stage [1]

Using the attached instruction set, apply the Dynamic Programming tiling algorithm to the following IR tree. Show the costs at each node in the tiled tree and list the instructions generated. [4]

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

Costs:

ADDI (CONST c) 1

ADDI (CONST b) 1

ADDI (PLUS (CONST b, CONST c)) 2

LOAD (MEM (PLUS (CONST b, CONST c))) 2

ADDI (CONST a) 1

TIMES (TIMES (CONST a, MEM (PLUS (CONST b, CONST c)))) 4

Instructions:

ADDI

LOAD

ADDI

TIMES

Marks: Costs: 2, Tree (nodes in brackets): 1, Instructions: 1

Question Four – Register Allocation [5]

Describe what occurs in each step of the typical simplify-coalesce-freeze-spill-select graph colouring algorithm for allocating registers to temporaries based on an interference graph. [5]

Simplify – non-MOVE-related nodes of degree<K are pushed onto the stack [1]

Coalesce – MOVE-related nodes that do not decrease colourability are merged. [1]

Freeze – MOVE-related nodes are converted to regular nodes by removing MOVE links. [1]

Spill – Node with degree>=K is pushed onto stack and marked as a potential spill. [1]

Select – Nodes are popped off stack and assigned colours. [1]