UNIVERSITY OF EDINBURGH

FACULTY OF SCIENCE AND ENGINEERING

COMPUTER SCIENCE THREE

COMPILING TECHNIQUES

Monday 28 May 2001

09:30 to 11:00

Examiners:

K. Turner (External)

G. Plotkin (Chairman)

INSTRUCTIONS TO CANDIDATES

Answer any TWO questions.

All questions carry equal weight.

1. (a) Describe concisely how an LR(1) parser works. You do not need to explain how it is constructed. Also describe the meaning of an LR(1) item ($X \to \alpha.\beta$). *[7 marks]*

   (b) The following grammar for *lists* of arithmetic expressions separated by + exhibits several conflicts when presented to the Yacc/CUP tool. Explain which conflicts these are, why they arise, and why they cause problems. *[6 marks]*

$$
\begin{array}{lcl}
P & \to & E \mid P{+}E \\
E & \to & \texttt{identifier} \\
E & \to & (E) \\
E & \to & EBE \\
B & \to & + \mid - \mid * \mid /
\end{array}
$$

   (c) Resolve the conflicts using directives, grammar transformations, and/or language alterations. You should use the notation of either Bison, ML-Yacc, or Java-CUP. Minor syntactic mistakes not affecting readability are tolerated. *[6 marks]*

   (d) Define the abstract syntax corresponding to the language fragment described by the grammar for lists of expressions as a C, ML, or Java data structure and augment your grammar with semantic actions which construct the abstract syntax tree for a given input. *[6 marks]*

2. (a) What happens in the *instruction selection* phase of the compilation? [*4 marks*]

(b) Recall that the ix86 instruction

$$\texttt{addi}\ op_1\ op_2$$

adds the operands op_1 and op_2 placing the result into op_2. The operands may be constants, registers, or memory locations of the form *offset*(*register*). At most one of the operands may be a memory location; op_2 must not be a constant. For the purpose of this question we also disallow memory locations as second operands.

Sketch the tree patterns corresponding to this instruction. [*7 marks*]

(c) Define the terms *optimal* and *optimum* solution to the instruction selection problem. [*2 marks*]

(d) Name an algorithm which computes the optimal solution and name an algorithm which computes the optimum solution. Describe *one* of these named algorithms in detail. [*7 marks*]

(e) What is a canonical tree and what is its purpose? [*5 marks*]

3. (a) Explain the difference between caller-save and callee-save registers. Why might a caller-save register sometimes not be saved? [*4 marks*]

(b) The guidelines from a processor manufacturer prescribe once and for all which registers are caller-save and which ones are callee-save. What are the advantages of having such a fixed policy? [*1 mark*]

(c) The table below[1] represents an interference graph. Nodes 1–6 are precoloured (with colours 1-6), and nodes A–H are ordinary (non-precoloured). Every pair of precoloured nodes interferes, and each ordinary node interferes with nodes where there is an X in the table. Assume that register allocation must be done for an 8-register machine.

Colour this graph using *simplify* and *spill*. Record the sequence (stack) of *simplify* and *potential spill* decisions, show which potential spills become actual spills and show the resulting colouring. [*5 marks*]

	1	2	3	4	5	6	A	B	C	D	E	F	G	H
A	X	X	X	X	X	X								
B	X		X	X	X	X								
C			X	X	X	X				X	X	X	X	X
D	X		X	X	X				X		X	X	X	X
E	X		X		X	X			X	X		X	X	X
F	X		X	X		X			X	X	X		X	X
G										X	X	X	X	
H	X			X	X	X			X	X	X	X		

(d) What is the most common reason for two nodes to interfere with each other? [*4 marks*]

(e) How are actual spills performed? [*6 marks*]

(f) Move instructions with identical source and target are obviously redundant and can be eliminated. One particular optimisation during register allocation is based on this observation. Which is it and how does it work? [*5 marks*]

[1]From Andrew Appel: Modern Compiler Implementation in ML, Cambridge University Press, 1998.