COMPUTER SCIENCE TRIPOS Part IB

Wednesday 2 June 1999 1.30 to 4.30

Paper 5

Answer **five** *questions.*

No more than **two** *questions from any one section are to be answered.*

Submit the answers in five **separate** *bundles, each with its own cover sheet. On each cover sheet, write the numbers of* **all** *attempted questions, and circle the number of the question attached.*

Write on **one** *side of the paper only.*

SECTION A

1 Data Structures and Algorithms

Describe, in detail, how the heapsort algorithm works. [10 marks]

Show that the worst-case cost of heapsort is $O(n \log n)$. [6 marks]

Would it be possible to implement a variant of heapsort based on a perfectly balanced ternary structure in which the children of node i are at positions $3i - 1$, $3i$, and $3i + 1$, and if so what would be the advantages and disadvantages of the new method? [4 marks]

2 Computer Design

instruction fetch	decode/ register fetch	execute	memory access	register write back

With reference to the classic RISC pipeline above, explain what a branch delay slot is and why it arises. [5 marks]

Demonstrate how conditional instructions can be used to avoid branches by writing code excerpts to perform the following function using a register-based processor. Comment your code to explain instruction semantics.

```
fun max(a,b) = if a>b then a else b;
```
[5 marks]

To assist with subroutine calls, ARM processors have a branch-with-link instruction and Intel processors have a call instruction. How do these instructions differ from a simple branch? [5 marks]

What is an *interrupt* and how is it similar to a branch-with-link instruction on the ARM? [5 marks]

3 Digital Communication I

Compare packet switching and circuit switching with particular reference to the following issues:

(*a*) how multiplexing is performed in each

(*b*) how addressing is performed in each

(*c*) functions which must be performed by a switch in each case

(*d*) situations in which each is advantageous [15 marks]

"As communication bandwidth becomes less and less expensive, the efficiency of packet switching will become less important than the simplicity of implementation and guarantees offered by circuit switching." Discuss. [5 marks]

4 Computer Graphics and Image Processing

A company wishes to produce a greyscale display with pixels so small that a human will be unable to see the individual pixels under normal viewing conditions.

What is the minimum number of pixels per inch required to achieve this? Please state all of the assumptions that you make in calculating your answer. It may be helpful to know that there are 150 000 cones per square millimetre in the human fovea, and that there are exactly 25.4 millimetres in an inch. [6 marks]

If the pixels could be only black or white, and greyscale was to be achieved by halftoning, then what would the minimum number of pixels per inch be in order that a human could not see the halftone dots? Again, state any assumptions that you make. [2 marks]

The company currently produces a display device with 2-bit greyscale (that is: four different shades of grey). Describe an error-diffusion algorithm which will convert an 8-bit greyscale image into a 2-bit image suitable for display on this device. [Note: the two images must have the same number of pixels.] [7 marks]

Illustrate that your algorithm works using the following test image.

200	40
250	220

[2 marks]

You are asked to design a 4×4 ordered dither matrix. What rules should you follow in the design? [3 marks]

SECTION B

5 Comparative Programming Languages

Discuss what it means for a program written in C to conform to the ANSI Standard and for a compiler to conform to the standard. [4 marks]

Discuss the reasons why a program conforming to a standard may fail to yield exactly the same results when run on different conforming compilers. [10 marks]

Discuss why it is sometimes possible to write apparently simple expressions, such as `9+8/3` in PL/I, that yield unexpected results in languages that have a wide variety of numerical data types. To what extent is it possible to eliminate such problems in future languages? [6 marks]

6 Compiler Construction

Explain a possible implementation method for Java-style or ML-style exceptions and handlers. [8 marks]

Consider a simple arithmetic expression e of abstract syntax:

$$e ::= x \mid n \mid e + e' \mid e - e' \mid e * e' \mid e/e' \mid -e$$

where x ranges over a set of (global) variables, addressable by name, and n ranges over integer constants. Write a procedure in pseudo-code or a language of your choice which takes an expression e and prints (one-per-line) stack-machine instructions of the form

```
pushvar  x
pushnum  n
add          ; pop two items and push their sum
sub          ; pop two items and push their difference
mul          ; pop two items and push their product
div          ; pop two items and push their quotient
neg          ; replace top item with its negation
```

which, when executed, have the net effect of pushing just the value of e onto the stack. Each line of code emitted should contain a comment giving the number of items on the stack after its execution, thus the first `push` and the last instruction would both be commented with “1 item”. [12 marks]

7 Prolog for Artificial Intelligence

N. Wirth's textbook *Algorithms + data structures = programs* (1976) contains the following story.

> I married a widow (call her W) who has a grown-up daughter (D). My father (F), who visited us quite often, fell in love with my step-daughter and married her. Hence my father became my son-in-law and my step-daughter became my mother. Some months later, my wife gave birth to a son (S1), who became the brother-in-law of my father, as well as my uncle. The wife of my father – that is, my step-daughter – also had a son (S2).

Using Prolog, create a list of facts that represents the situation in the above story. [5 marks]

Add rules defining the family relationships (such as father-in-law) described in the story. [5 marks]

Show how a Prolog system would use your program to prove the goal "I am my own grandfather". [10 marks]

8 Databases

Explain the ANSI/SPARC architecture for Data Base Management Systems, and show how it supports data independence. [5 marks]

Describe the relational model of data introduced by E.F. Codd in 1970. [4 marks]

What are its strengths and weaknesses? [7 marks]

What factors have led to its dominant position in the market place today? [4 marks]

SECTION C

9 Semantics of Programming Languages

The commands C of a language for manipulating integer storage locations ℓ are given by

$$\begin{aligned} C \;::=\; & \mathbf{skip} \mid \ell := E \mid C\,;C \mid \mathbf{if}\; B\; \mathbf{then}\; C\; \mathbf{else}\; C \\ & \mid \mathbf{while}\; B\; \mathbf{do}\; C \mid \mathbf{begin\ loc}\, \ell := E\,; C\; \mathbf{end} \end{aligned}$$

where B and E range over boolean and integer expressions respectively, whose precise syntax you do not need to specify, but which include various integer and boolean operations and an operation for reading the contents of a location. The last form of command is for block-structured local state.

You may assume that evaluation of integer expressions E to integer values n has no side-effects on states s and that a suitable evaluation relation of the form $E, s \Downarrow n$ has already been defined; similarly an evaluation relation $B, s \Downarrow b$ for boolean expressions has already been defined. Using these relations, give an operational semantics for commands in the form of an inductively defined relation $C, s \Downarrow s'$ for evaluating a command C in a state s, resulting in a final state s'. Make sure that your definition treats properly the local scope of ℓ in a block $\mathbf{begin\ loc}\, \ell := E\,; C\; \mathbf{end}$. Illustrate this by showing that evaluating $\ell := 0\,; (\mathbf{begin\ loc}\, \ell := 1\,; \mathbf{skip\ end})$ in any state results in a state in which the value stored in ℓ is 0, not 1. [12 marks]

Use the operational semantics to give a definition of *semantic equivalence* for these commands. Prove that if $\ell \neq \ell'$ then $\mathbf{begin\ loc}\, \ell := E\,; \ell' := !\ell\; \mathbf{end}$ and $\ell' := E$ are semantically equivalent for any integer expression E. What happens if $\ell = \ell'$? If C does not involve any occurrences of ℓ, is $\mathbf{begin\ loc}\, \ell := E\,; C\; \mathbf{end}$ always semantically equivalent to C? [8 marks]

10 Logic and Proof

How do Prolog clauses differ from the clauses used by general-purpose resolution theorem provers? [2 marks]

Describe the series of resolutions that is performed by a Prolog interpreter when it is supplied with a program and a set of clauses. To illustrate your answer, explain how the following program executes when presented with the goal $\leftarrow Q(f(f(f(f(a)))))$:

$$Q(a) \leftarrow \tag{1}$$

$$Q(f(a)) \leftarrow \tag{2}$$

$$Q(f(f(x))) \leftarrow Q(f(x)), Q(x) \tag{3}$$

[7 marks]

Considering the program and the goal $\leftarrow Q(f(f(f(f(a)))))$ now as a set of clauses, derive the empty clause using general resolution. (Full credit requires finding the shortest derivation.) [6 marks]

Consider the set of clauses consisting of the program given above and the goal $\leftarrow Q(\underbrace{f(\cdots f(}_{n}(a))\cdots))$. Let $p(n)$ be the number of steps executed by a Prolog interpreter when given those clauses. Let $r(n)$ be the minimum number of steps required to derive the empty clause from those clauses using general resolution. Compare the approximate growth rates of $p(n)$ and $r(n)$ as n increases, and explain any difference you find. [5 marks]

[TURN OVER

11 Complexity Theory

Explain briefly, stating but not proving any relevant results, which of the following statements are true, which are false and which are meaningless in the context of a study of the complexity of computation. [*Each part will be allocated the same weight when marking, but conciseness and clarity of explanation will be important as well as simple factual correctness.*]

(*a*) I can check an integer N to see whether it is prime by doing test division by all the numbers less than it. This involves just under N trial divisions, and division has a polynomial cost. Therefore testing to see whether a number is prime is a problem in the class P.

(*b*) If I am given an integer N and am told in advance that it is composite then I can guess a pair of integers P and Q, multiply them together and check whether their product is N. Multiplication has polynomial cost hence factorising known-composites is in the class NP.

(*c*) The quotient of a pair of n-digit integers can be computed in a time less than $kn^{1.1}$ for some value k which depends on the exact speed of the (ordinary) computer being used.

(*d*) If P is a class of problems, and every instance of P can be converted (efficiently) into an instance of an NP-complete problem Q, and a solution to the corresponding instance of Q lets you (again efficiently) derive a solution to the original instance of P, then P is NP-complete.

(*e*) If P = NP then we can solve the decision version of the Travelling Salesman Problem efficiently on a deterministic computer: i.e. given a graph with weighted edges and an integer k we can find a route visiting each vertex of the graph and having total edge-weight at most k. Because of this we could then solve the minimisation version of the same problem, i.e. find the shortest path through the graph that visits each vertex, and this would still be achievable in polynomial time.

[20 marks]

12 Foundations of Functional Programming

Explain why the Church–Rosser theorem implies that there are λ-terms that are not equal to each other. [2 marks]

Suppose the following reduction rule is added to the λ-calculus:

$$\lambda xy.x \to \lambda xy.y$$

Show that in the resulting calculus, all terms are equal. [3 marks]

Let $A = \lambda xy.y(xxy)$ and $\Theta = AA$. Show that Θ is a fixed-point combinator. [3 marks]

Assume an encoding of lists where $[a_1, \ldots, a_m]$ is represented by

$$\lambda fx.fa_1(fa_2 \cdots (fa_m x) \cdots)$$

Use the fixed-point combinator Θ to obtain a λ-term **rev** such that:

$$\mathbf{rev}[a_1, \ldots, a_m] = [a_m, \ldots, a_1]$$

You may assume a λ-term representation of the booleans and of ordered pairs, but you should define any other terms you require. [12 marks]