COMPUTER SCIENCE TRIPOS Part IB

Monday 1 June 1998 1.30 to 4.30

Paper 3

Answer **five** *questions.*

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.*

1 Compiler Construction

Give *short* answers for each of the following.

(*a*) Give a program which has a different result when interpreted using dynamic scoping from that using static scoping.

(*b*) Give a program (using `let` to declare variables) which runs with no run-time type errors under dynamic typing but is invalid in a static typing regime such as that of Java or C.

(*c*) Compared with `int v[8]`, why might the declaration `int v[`*e*`]` cause less efficient code to be compiled if *e* is not a constant expression, or even if *e* is a large constant?

(*d*) Give two context-free grammars which accept the same strings with one being ambiguous and one non-ambiguous.

(*e*) Give a data-structure type declaration in a language of your choice which might be useful to represent parse trees for the following grammar:

```
E  ->  E + T | E - T | T
T  ->  P ^ T | P
P  ->  ( E ) | n
```

Here `E` is the start symbol and `n` is a terminal symbol representing the result of lexing an integer constant.

[4 marks each]

2 Concurrent Systems

An n-process mutual exclusion algorithm has entry and exit protocols given below. In order to express the algorithm concisely we use a lexicographic "less than" relation on ordered pairs of integers, so that:

$$(a, b) < (c, d) \text{ if } a < c \text{ or if } a = c \text{ and } b < d.$$

Entry protocol for the critical region for process i:

```
$taking[i] := true$;
$ticket[i] := max(ticket[0], ticket[1], \ldots, ticket[n-1]) + 1$;
$taking[i] := false$;
for $j := 0$ to $n - 1$ do
    begin
        while $taking[j]$ do no-op;
        while $ticket[j] \neq 0$ and $(ticket[j], j) < (ticket[i], i)$ do no-op;
    end
```

Exit protocol for the critical region for process i:

```
$ticket[i] := 0$;
```

(*a*) Illustrate fully the operation of the algorithm by showing, for a small value of n, successive values of the arrays *taking* and *ticket* under a variety of concurrent executions. Explain by means of short comments on the values. [14 marks]

(*b*) Is it possible or likely that a value in the array *ticket* might overflow? Why? [2 marks]

(*c*) A RISC processor has an atomic *read-and-clear-memory* instruction. Give pseudo-code for the entry and exit protocols using the instruction. [4 marks]

3 Programming in Java

Describe briefly the facilities provided in Java for describing classes and inheritance. How is an object instantiated, initialised and finalised? [8 marks]

What are abstract classes and interfaces? How and why are they used? [4 marks]

Explain how modifiers are used to control the visibility of identifiers. [4 marks]

Illustrate your answer with examples drawn from the various schemes for handling graphical input in different versions of the Abstract Windowing Toolkit. [4 marks]

4 Continuous Mathematics

Show that for all families of functions which are "self-Fourier" (i.e. equivalent in functional form to their own Fourier transforms), closure of a family under multiplication entails also their closure under convolution, and *vice versa.*

[Hint: closure of a set of functions under an operation means that applying that operation to any member of the set creates a function which is also a member of the set.] [10 marks]

A periodic square wave, which alternates between the constants $+\pi/4$ and $-\pi/4$ with period 2π has the following Fourier series, using all positive odd integers n:

$$f(x) = \sum_{\text{odd } n=1}^{\infty} \frac{1}{n} \sin(nx)$$

Derive from this the Fourier series for a periodic triangular wave, which ramps up and down with slopes $+\pi/4$ and $-\pi/4$ and with period 2π. [3 marks]

Any real-valued function $f(x)$ can be represented as the sum of one function $f_e(x)$ that has even symmetry (it is unchanged after a right–left flip around $x = 0$) so that $f_e(x) = f_e(-x)$, plus one function $f_o(x)$ that has odd symmetry, so that $f_o(x) = -f_o(-x)$. Such a decomposition of any function $f(x)$ into $f_e(x) + f_o(x)$ is illustrated by

$$f_e(x) = \frac{1}{2}f(x) + \frac{1}{2}f(-x)$$

$$f_o(x) = \frac{1}{2}f(x) - \frac{1}{2}f(-x)$$

Use this type of decomposition to explain why the Fourier transform of any real-valued function has *Hermitian symmetry*: its real-part has even symmetry, and its imaginary-part has odd symmetry. Comment on how this redundancy can be exploited to simplify computation of Fourier transforms of real-valued, as opposed to complex-valued, data. [7 marks]

5 Data Structures and Algorithms

Describe

(*a*) how to determine whether or not a point is inside a simple plane closed polygon, paying proper attention to awkward cases; [6 marks]

(*b*) how, with luck, to exclude large numbers of points from the convex hull of a set of points in the plane, with due consideration of what can go wrong; [7 marks]

(*c*) how to compute economically the convex hull of the points that are left after the measures you have described in (*b*) above. [7 marks]

6 Data Structures and Algorithms

What is a *priority queue*? Explain the data structure known as a *heap* and describe how a heap can be implemented using a simple linear block of memory. Assuming that a heap implemented in this way stores N items, describe how it can be viewed as an almost-balanced binary tree. What difference can there be between the greatest and least lengths of paths from the root of the tree to a leaf? [5 marks]

Describe, and estimate the costs of, procedures to

(*a*) find the parent and offspring of a given node; [2 marks]

(*b*) insert a new item into an existing heap; [2 marks]

(*c*) delete the topmost item from a non-empty heap; [2 marks]

(*d*) starting from an array holding N items in arbitrary order, rearrange those items so that they form a heap, taking time less than that which would be needed if the items were just inserted into the heap one after the other. [4 marks]

A *stable* sorting method is one where items whose keys compare as equal will appear in the output in the same order that they appeared in the input list. Would a heap sort based on the algorithms you have documented be stable? Justify your answer. [5 marks]

7 Computer Design

How can a pipeline be used to improve processor performance? [5 marks]

Explain, with reference to the 5-stage pipeline diagram below, how feed-forwarding (also called bypassing) can be used to reduce the effects of load delays but cannot eliminate them.

instruction fetch	register fetch	execute	memory access	register write back

[10 marks]

Most instructions in old accumulator machines read an operand from memory which is then combined with the accumulator. The only exception is usually the `store` instruction which writes the accumulator (without modification) to the memory. If you were to pipeline an accumulator machine, where would the memory access stage go? [5 marks]

8 Operating System Functions

Multimedia applications may be characterised by their requirement for timely completion of their computation or data processing in order to function usefully. Why are traditional operating systems unable to deliver such guarantees? Comment on the features which an operating system should implement in order to support such applications successfully, considering issues of memory management, device access, and scheduling techniques. [20 marks]

9 Computation Theory

What is meant by saying that a model for computation offers unlimited data storage but is restricted to finite logic? [3 marks]

How would you record the *configuration* during computation within such a model? Illustrate your answer by considering both Turing machines and register machines. [5 marks]

Suppose you are given a k-symbol Turing machine having searching states. Show how to represent the transition from the configuration at time t to the configuration at time $t + 1$ by a system of arithmetic equations. Hence show that any Turing machine computation may be simulated by a register machine having a suitable program. [12 marks]

10 Numerical Analysis I

Define *relative error*, *machine epsilon* (*macheps*). [2 marks]

Consider IEEE single-precision arithmetic. How are the 32 bits arranged in terms of sign, exponent and significand? How is the exponent stored? Explain the terms *normalized number*, *denormal number*. What is the *hidden bit* and how is it used? How are negative numbers stored? What does *NaN* stand for? Give an example of an operation that yields a *NaN* value. [6 marks]

Given that $e_{\max} = 127$, show the bit pattern representing each of the following numbers. [Draw lines to separate the sign, exponent and significand. You may use "$0 \ldots 0$" to represent long strings of zeros.]

0
$-\infty$
-1.0
$1.0 + \textit{macheps}$
4.0
$4.0 + \textit{macheps}$
1.125×2^{-31}
a *NaN* value [give one example]
$\hat{x}$, the smallest representable number greater than 2^{16}

[9 marks]

In the last case, what is

(*a*) the value of the least significant bit in the significand of $\hat{x}$, and

(*b*) the relative error if rounding error causes 2^{16} to be stored as $\hat{x}$? [3 marks]