COMPUTER SCIENCE TRIPOS Part IB

Monday 4 June 2001 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.*

You may not start to read the questions printed on the subsequent pages of this question paper until instructed that you may do so by the Invigilator

1 Concurrent Systems

A committed ACID transaction transforms a database from one consistent state to another.

(*a*) Define the property of isolation in a transaction system. [2 marks]

(*b*) The property of isolation may be enforced either at transaction commit or at all times throughout transaction execution. Describe how each of these approaches may be implemented, and the tradeoffs between the approaches, for

(*i*) two-phase locking [5 marks]

(*ii*) timestamp ordering [5 marks]

(*iii*) optimistic concurrency control [8 marks]

2 Further Java

(*a*) Describe the differences and similarities between abstract classes and interfaces in Java. How would you select which kind of definition to use? [5 marks]

(*b*) An enthusiast for programming with *closures* proposes extending Java so that the following method definition would be valid:

```
Closure myCounter (int start) {
  int counter = start;
  return {
    System.out.println (counter ++);
  }
}
```

The programmer intends that no output would be produced when this method is executed, but that it would return an object of a new type, `Closure`, and that invoking `apply()` on that object would cause successive counter values to be printed.

By using an *inner class* definition, show how this example could be re-written as a valid Java program. [5 marks]

(*c*) A common programming mistake is to try to define a class to have more than one superclass. For example a naïve programmer may write

```
class AutoTree extends Thread, BinaryTree {
  ...
}
```

when defining a new kind of data structure which uses an additional thread to keep the tree balanced. Describe *three* ways in which this problem can be resolved to produce (one or more) valid class definitions. State, with a brief justification, which you would use here. [10 marks]

3 Compiler Construction

A *regular grammar* is a grammar whose rules are in one of the two following forms (where A and B are non-terminal symbols and a is a terminal):

$$\begin{aligned} A &\to a \\ A &\to aB \end{aligned}$$

(*a*) Give a regular grammar which generates floating point numbers of exactly the following form:

$$(0|1)^{+}.(0|1)^{*}[e(0|1)^{+}]$$

where "()" indicates grouping, "[ ]" indicates optional item, "ρ^{+}" indicates one or more repetitions of ρ and "ρ^{*}" indicates zero or more repetitions of ρ. [8 marks]

(*b*) Give a non-regular grammar with fewer productions than your answer to (*a*) but which generates the same set of strings. [4 marks]

(*c*) Determine, with justification, for the following grammars

(*i*) whether S generates strings not generated by T; and

(*ii*) whether T generates strings not generated by S.

$$\left\{ \begin{array}{l} S \to aaS \\ S \to Scc \\ S \to d \end{array} \right\} \text{ and } \left\{ \begin{array}{l} T \to aTc \\ T \to d \end{array} \right\}$$

[4 marks]

(*d*) What is the significance for the compilation process of the idea of "strings which can be generated by regular grammars"? Your answer should explain where such a module recognising such strings would appear in a compiler and a possible external interface (functions, variables and/or objects) it might present to the rest of the compiler. [4 marks]

4 Introduction to Security

Which mode (or modes) of operation of the Advanced Encryption Standard (AES) block cipher would you use to protect the following? Give a brief justification for your answers.

(*a*) Inter-bank funds transfers. [4 marks]

(*b*) Email messages. [4 marks]

(*c*) A high-frequency radio modem link. [4 marks]

(*d*) Passwords stored on a local disk. [4 marks]

(*e*) The pulse train from a gearbox sensor to the tachograph in a truck. [4 marks]

5 Data Structures and Algorithms

(*a*) Carefully describe an implementation of quicksort to sort the elements of an integer vector, and state, without proof, its expected and worst case complexity for both time and space in terms of the size of the vector. [7 marks]

(*b*) Describe a more efficient algorithm for the case where it is known that the vector has exactly 10^6 elements uniformly distributed over the range 0 to 10^6. [7 marks]

(*c*) Describe an efficient algorithm to find the median of a set of 10^6 integers where it is known that there are fewer than 100 distinct integers in the set. [6 marks]

6 Computer Design

(*a*) Why does pipelining introduce data hazards? [4 marks]

(*b*) What mechanism may be used to remove some data hazards? [4 marks]

(*c*) Why is it not possible to remove all data hazards? [4 marks]

(*d*) What hardware is required to prevent data hazards from infringing the programmer's model of instruction execution? [4 marks]

(*e*) What is the difference between a data hazard and a control hazard? [4 marks]

7 Operating System Functions

(*a*) What are the key issues with scheduling for shared-memory multiprocessors? [3 marks]

(*b*) *Processor affinity*, *take scheduling* and *gang scheduling* are three techniques used within multiprocessor operating systems.

(*i*) Briefly describe the operation of each. [6 marks]

(*ii*) Which problem does the processor affinity technique seek to overcome? [2 marks]

(*iii*) What problem does the processor affinity technique suffer from, and how could this problem be overcome? [2 marks]

(*iv*) In which circumstances is a gang scheduling approach most appropriate? [2 marks]

(*c*) What additional issues does the virtual memory management system have to address when dealing with shared-memory multiprocessor systems? [5 marks]

8 Continuous Mathematics

(*a*) The MacLaurin series for a continuous, infinitely differentiable function, $f(x)$, is:

$$f(x) = f(0) + \frac{f'(0)}{1!}x + \frac{f''(0)}{2!}x^2 + \frac{f'''(0)}{3!}x^3 + \cdots + \frac{f^{(k)}(0)}{k!}x^k + \cdots$$

Derive the MacLaurin series for each of $\sin(x)$, $\cos(x)$, and e^x. [6 marks]

(*b*) Hence, or otherwise, prove that:

$$e^{i\phi} = \cos\phi + i\sin\phi$$

where $i = \sqrt{-1}$ [3 marks]

(*c*) Prove that the box function, $b(x)$:

$$b(x) = \begin{cases} 1, & |x| \leqslant \frac{1}{2} \\ 0, & \text{otherwise} \end{cases}$$

has the Fourier transform, $B(\nu)$:

$$B(\nu) = \frac{\sin \pi\nu}{\pi\nu}$$

where ν is frequency measured in Hertz (cycles per second). [7 marks]

(*d*) The convolution of $b(x)$ with itself is $t(x)$:

$$t(x) = b(x) * b(x) = \begin{cases} 1 + x, & -1 \leqslant x \leqslant 0 \\ 1 - x, & 0 \leqslant x \leqslant 1 \\ 0, & \text{otherwise} \end{cases}$$

Hence, or otherwise, find the Fourier transform, $T(\nu)$, of $t(x)$. [4 marks]

9 Computation Theory

(*a*) Describe the action of a Turing machine. [4 marks]

(*b*) Define what is meant by a configuration of an N-state, k-symbol Turing machine. [2 marks]

(*c*) Explain briefly how to enumerate all possible Turing machine computations, so that a given computation can be characterised by a single natural number code c. [5 marks]

(*d*) Show that it is not possible to compute the maximum distance travelled by the Turing machine head from its initial position during halting computations as a function of the code c. Any results that you use should be stated clearly. [9 marks]

10 Numerical Analysis I

For IEEE Single Precision $\beta = 2$, $p = 24$, $e_{\text{max}} = +127$, $e_{\text{min}} = -126$. Explain these parameters. How many bits are required to store the *exponent* and the *significand*, respectively? How is the exponent stored? [6 marks]

By means of a table, or otherwise, describe how the following quantities are represented: *zero*, *denormal numbers*, *normalised numbers*, *infinities* and *Not a Number* (*NaN*). [5 marks]

Let ω represent any of the operations $+ - *$ or $/$. Let x be any normalised or denormal number or $\pm\, 0$. Writing n for any *NaN* value, what do the following evaluate to?

(a) $x \;\omega\; n$

(b) $\pm \infty \;\omega\; n$

(c) $x \;\omega\; \pm\; \infty$

(d) $\sqrt{\pm \infty}$

[6 marks]

Let z be the smallest representable positive normalised number. What are the values of the following?

(e) z

(f) the largest representable number smaller than z

(g) the smallest representable positive number

[3 marks]

END OF PAPER