COMPUTER SCIENCE TRIPOS Part Ib

Tuesday 1 June 2004 1.30 to 4.30

Paper 4

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.*

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 Compiler Construction

(*a*) A context-free grammar can be formally defined as a 4-tuple. Give a precise statement of what the components are. [2 marks]

(*b*) Explain the difference between a grammar and the language it generates. [2 marks]

(*c*) Explain what makes a grammar ambiguous, with reference to the grammar which may be commonly expressed as a "rule"

$$E ::= 1 \mid 2 \mid X \mid E + E \mid E * E \mid -E$$

where X is an identifier. [2 marks]

(*d*) For the "rule" in part (*c*), give a formal grammar containing this "rule" and adhering to your definition in part (*a*). [2 marks]

(*e*) Give non-ambiguous grammars each generating the same language as your grammar in part (*d*) for the cases:

(*i*) "$-$" is most tightly binding and "$+$" and "$*$" have equal binding power and associate to the left.

(*ii*) "$-$" is most tightly binding and "$+$" and "$*$" have equal binding power and associate to the right.

(*iii*) "$-$" binds more tightly than "$+$", but less tightly than "$*$", with "$+$" left-associative and "$*$" right-associative so that "$-a + -b * c * d + d$" is interpreted as "$((-a) + (-(b * (c * d)))) + d$".

[2 marks each]

(*f*) Give a simple recursive descent parser for your grammar in part (*e*)(*iii*) above which yields a value of type `ParseTree`. You may assume operations *mkplus*, *mktimes*, *mkneg* acting on type `ParseTree`. [6 marks]

2 Economics and Law

(*a*) What is "strategy" in game theory? [5 marks]

(*b*) Explain the difference between a dominant strategy equilibrium and a Nash equilibrium. [5 marks]

(*c*) Participants in a peer-to-peer file-sharing system can either cooperate (share their files with others) or cheat (try to download from others without making any contribution themselves). Write down a possible payoff matrix for their behaviour, and identify the Nash equilibrium. [5 marks]

(*d*) Is this equilibrium Pareto-efficient, and, if not, what can be done to make it so? [5 marks]

3 Data Structures and Algorithms

(*a*) Describe the structure of splay trees used to represent a set of key–value pairs. [5 marks]

(*b*) Describe how new key–value pairs are added to the tree, how the value associated with a given key can be looked up, and how to delete a pair with a given key. [5 marks]

(*c*) State without proof the attractive properties of splay trees. [4 marks]

(*d*) Describe the ternary tree structure used to hold a dictionary of key–value pairs where the keys are variable-length strings. Illustrate the mechanism by showing the structure after items with keys `MIT`, `SAD`, `MAN`, `APT`, `MUD`, `ADD`, `MAG`, `MINE`, `MIKE`, `MINT`, `AT`, `MATE` and `MINES` have been added in that order to an initially empty ternary tree. [6 marks]

4 Comparative Programming Languages

(*a*) Briefly describe the concept of coroutines as provided in BCPL, and outline the effect of the library functions `createco(f, size)`, `deleteco(cptr)`, `callco(cptr, val)`, and `cowait(val)`. [6 marks]

(*b*) Discuss the relative merits of BCPL coroutines *versus* those of threads such as provided in Java. [6 marks]

(*c*) Outline the overall design and organisation of a BCPL program to perform discrete event simulation using coroutines to implement the simulated activities. Concentrate on the design of the simulation event loop, the organisation of the priority queue and what functions you would provide to simplify the implementation of the activities. It would probably be sensible to adopt a programming style similar to that used in Simula 67. You should hold simulated time as a global (integer) variable. [8 marks]

5 Operating Systems II

(*a*) Most conventional hardware translates virtual addresses to physical addresses using *multi-level page tables* (MPTs):

(*i*) Describe with the aid of a diagram how translation is performed when using MPTs. [3 marks]

(*ii*) What problem(s) with MPTs do *linear page tables* attempt to overcome? How is this achieved? [3 marks]

(*iii*) What problems(s) with MPTs do *inverted page tables* (IPTs) attempt to overcome? How is this achieved? [3 marks]

(*iv*) What problems(s) with IPTs do *hashed page tables* attempt to overcome? How is this achieved? [3 marks]

(*b*) Operating systems often cache part of the contents of the disk(s) in main memory to speed up access. Compare and contrast the way in which this is achieved in (*i*) 4.3 BSD Unix and (*ii*) Windows 2000. [8 marks]

6 Continuous Mathematics

For non-negative integers r and s we have the orthogonality properties

$$\int_0^{2\pi} \cos(rx)\cos(sx)dx = \begin{cases} 2\pi & \text{if } r = s = 0 \\ \pi\delta_{rs} & \text{otherwise} \end{cases}$$

$$\int_0^{2\pi} \sin(rx)\sin(sx)dx = \begin{cases} 0 & \text{if } r = s = 0 \\ \pi\delta_{rs} & \text{otherwise} \end{cases}$$

$$\int_0^{2\pi} \sin(rx)\cos(sx)dx = 0 \qquad \forall \quad r, s$$

where

$$\delta_{rs} = \begin{cases} 1 & \text{if } r = s \\ 0 & \text{otherwise} \end{cases} .$$

(a) Derive expressions for the Fourier coefficients a_0, a_n, b_n $(n = 1, 2, \ldots)$ such that the infinite series

$$\frac{a_0}{2} + \sum_{n=1}^{\infty} \left(a_n \cos(nx) + b_n \sin(nx)\right)$$

is the Fourier series for the function $f(x)$ in an interval of length 2π. [6 marks]

(b) For any fixed integer $N \geq 1$ let

$$S_N(x) = \frac{a_0}{2} + \sum_{n=1}^{N-1} \left(a_n \cos(nx) + b_n \sin(nx)\right)$$

be the Fourier series for $f(x)$ truncated to the first N terms and let

$$S'_N(x) = \frac{a'_0}{2} + \sum_{n=1}^{N-1} \left(a'_n \cos(nx) + b'_n \sin(nx)\right)$$

be any other Fourier series truncated to the first N terms. Show that

$$\int_0^{2\pi} \left(f(x) - S_N(x)\right)\left(S_N(x) - S'_N(x)\right) dx = 0 \,.$$

[8 marks]

(c) Given the function $f(x)$ show that

$$\int_0^{2\pi} \left(f(x) - S'_N(x)\right)^2 dx$$

is minimised by the unique choice $a'_0 = a_0$, $a'_n = a_n$, $b'_n = b_n$ $(n = 1, 2, \ldots)$, that is, the Fourier series gives the best approximation to $f(x)$ using N terms in the sense of minimising the mean-squared error. [6 marks]

7 Numerical Analysis I

(a) The mid-point rule can be expressed in the form

$$I_n = \int_{n-\frac{1}{2}}^{n+\frac{1}{2}} f(x)dx = f(n) + e_n$$

where

$$e_n = f''(\theta_n)/24$$

for some θ_n in the interval $(n-\frac{1}{2}, n+\frac{1}{2})$. Assuming that a formula for $\int f(x)dx$ is known, and using the notation

$$S_{p,q} = \sum_{n=p}^{q} f(n) \ ,$$

describe a method for estimating the sum of a slowly convergent series $S_{1,\infty}$, by summing only the first N terms and estimating the remainder by integration. [5 marks]

(b) Assuming that $f''(x)$ is a positive decreasing function, derive an estimate of the error $|E_N|$ in the method. [5 marks]

(c) Given

$$\int \frac{dx}{(1+x)\sqrt{x}} = 2\tan^{-1}\sqrt{x}$$

illustrate the method by applying it to

$$\sum_{n=1}^{\infty} \frac{1}{(1+n)\sqrt{n}} \ .$$

Verify that $f''(x)$ is positive decreasing for large x, and estimate the integral remainder to be added to $S_{1,N}$. [6 marks]

(d) How large should N be to achieve an absolute error of approximately 2×10^{-15}? [You may assume $N+1 \simeq N$ for this purpose.] [4 marks]

8 Concurrent Systems and Applications

A multi-threaded application is using a long linked list of integers. The list is accessed through `synchronized` methods on a `ListSet` object.

The list itself comprises a chain of `ListNode` objects in ascending numerical order. The chain always starts and ends with special *sentinel* nodes conceptually containing $-\infty$ and $+\infty$ respectively. This simplifies the implementation of operations on the list: they do not have to deal with inserting elements at the very start or at the very end.

(*a*) Sketch the definition of `ListSet` and `ListNode` as Java classes. You need only give appropriate field definitions and the implementation of an `insert` method on `ListSet`. [4 marks]

(*b*) An engineer suggests that, instead of holding a lock on a `ListSet` object, threads only need to lock a pair of `ListNode` objects in the region that they are working.

(*i*) Define methods `lock` and `unlock` for your `ListNode` class to allow a thread to acquire a mutual exclusion lock on a given node. [6 marks]

(*ii*) Show how your insert method could be updated to incorporate the engineer's idea. [8 marks]

(*iii*) Do you think the new implementation will be faster than the original one? Justify your answer. [2 marks]

9 Computation Theory

(*a*) Explain what is meant by the following statements:

(*i*) $f : \mathbb{N} \to \mathbb{N}$ is a *total recursive* (TR) function; [3 marks]

(*ii*) the sequence $\{f_n : \mathbb{N} \to \mathbb{N}\}_{n \in \mathbb{N}}$ of TR functions of a single variable is recursively enumerable. [4 marks]

(*b*) Show that no recursive enumeration can include the set of *all* TR functions of a single variable. [4 marks]

(*c*) Suppose $u(n, x)$ is a recursive enumeration of the sequence of TR functions $f_n(x) = u(n, x)$. Show how to define a sequence $\{g_n : \mathbb{N} \to \mathbb{N}\}$ of TR functions of a single variable such that each g_n is distinct from every function f_n, and also from each g_k for $k \neq n$. [5 marks]

(*d*) Express the sequence $\{g_n\}$ as an explicit recursive enumeration $v(n, x) = g_n(x)$. [4 marks]

END OF PAPER