COMPUTER SCIENCE TRIPOS Part Ib

Thursday 7 June 2001 1.30 to 4.30

Paper 6

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.*

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

SECTION A

1 Data Structures and Algorithms

(*a*) State what is meant by a *directed graph* and a *strongly connected component*. Illustrate your description by giving an example of such a graph with 8 vertices and 12 edges that has three strongly connected components. [5 marks]

(*b*) Describe, in detail, an algorithm to perform a depth-first search over such a graph. Your algorithm should attach the discovery and finishing times to each vertex and leave a representation of the depth-first spanning tree embedded within the graph. [5 marks]

(*c*) Describe an $O(n)$ algorithm to discover all the strongly connected components of a given directed graph and explain why it is correct. You may find it useful to use the concept of the forefather $\phi(v)$ of a vertex v which is the vertex, u, with highest finishing time for which there exists a (possibly zero length) path from v to u. [10 marks]

2 Computer Design

(*a*) What is a *data cache* and what properties of data access does it exploit? [5 marks]

(*b*) What is a *direct mapped cache* and under what conditions will it exhibit poor performance? [5 marks]

(*c*) Under what circumstances might a word of data in main memory be simultaneously held in two separate first-level cache lines? [5 marks]

(*d*) A *translation look aside buffer* is a specialised cache. What does it typically store and why is it often a factor of 1000 smaller than a data cache? [5 marks]

3 Digital Communication I

(*a*) Define the terms *circuit* and *packet* in the context of communication systems. [5 marks]

(*b*) What sort of guarantee does circuit switching provide? [5 marks]

(*c*) What advantages does packet switching provide over circuit switching? [5 marks]

(*d*) Which of *frequency division multiplexing*, *time division multiplexing* and *code division multiplexing* lend themselves to circuit switching? Which to packet switching? Explain why or why not in each case. [5 marks]

4 Computer Graphics and Image Processing

(*a*) Describe the z-buffer polygon scan conversion algorithm. [10 marks]

(*b*) In ray tracing, once we have determined where a ray strikes an object, the illumination at the intersection point can be calculated using the formula:

$$I = I_a k_a + \sum_i I_i k_d (\mathbf{L}_i \cdot \mathbf{N}) + \sum_i I_i k_s (\mathbf{R}_i \cdot \mathbf{V})^n$$

Explain what real effect each of the three terms is trying to model and explain what each of the following symbols means, within the context of this formula:

$$I,\ I_a,\ i,\ I_i,\ k_a,\ k_d,\ k_s, \mathbf{L}_i,\ \mathbf{N},\ \mathbf{R}_i,\ \mathbf{V},\ n$$

[10 marks]

SECTION B

5 Comparative Programming Languages

(*a*) Briefly explain the concept of coroutines as used in BCPL and outline the effect of the library functions `createco(f, size)`, `deleteco(ctpr)`, `callco(cptr, val)` and `cowait(val)`. [10 marks]

(*b*) Outline how you would design a coroutine to merge, in increasing order, two infinite streams of increasing integers supplied by two other coroutines. [5 marks]

(*c*) Briefly outline how you would implement an analogous merging mechanism in an object-oriented language, such as Java, that does not provide a coroutine mechanism. [5 marks]

6 Compiler Construction

(*a*) Describe one possible structure (e.g. ELF) of an object file. Illustrate your answer by considering the form of object file which might result from the following C program.

```
int a = 1, b = -1;
extern int g(int);
extern int c;
int f() { return g(a-b) + c; }
```

It is not necessary to consider the exact instruction sequence, just issues concerning its interaction with the object file format. [10 marks]

(*b*) Describe how a linker takes a sequence of such programs and produces an executable file. [4 marks]

(*c*) Compare and contrast *static* and *dynamic* linking in a system using your object file format. [6 marks]

7 Prolog for Artificial Intelligence

A weighted binary tree can be defined using compound terms in the following way. A node of the tree is represented by the term $n(V, L, R)$, where V stands for the value of the node, and L and R stand for the left and right branches, respectively. A terminal node has the R and L components instantiated to the null list.

Given an input tree T, write a Prolog program that constructs a tree of the same shape as T, but in which the value of each node has been set to the value of the maximum value node in T.

[Note: Maximum marks are available only for programs that perform this task in one recursive descent of the input tree, and which use no more than four clauses.]

[20 marks]

8 Databases

The environmental agency is setting up an SQL database to monitor long-term trends in the climate. Data are collected from observatories of a number of different kinds.

Flood risk is of particular concern. Each water authority measures river levels and rates of flow hourly at major points, and records reservoir levels daily.

In addition, the agency maintains weather stations both inland and at sea. These record precipitation (rainfall etc.), temperature, sunshine, air pressure and wind. Values of new precipitation, temperature, pressure, and wind speed and direction are taken hourly; gusts of over 60 m.p.h. are noted whenever they occur.

Maximum and minimum temperature and pressure, the total number of hours of sunshine and the total precipitation are recorded daily. Inland stations can be grouped by water authority.

By default these primary data will be relegated to archive after 2 years. Selected information is retained permanently in a data warehouse. This serves two purposes. First, it holds monthly summary data consisting of the maximum (and minimum as appropriate) day value for each statistic, together with the monthly totals of sunshine and precipitation. The warehouse also keeps detailed information relating to periods of extreme weather from the relevant observatories, with one or more keywords describing the nature of the incident (*flood*, *blizzard*, *hurricane* etc.) and an optional comment.

Write notes to assist in the design of the schema for the relational data warehouse, including any diagrams that you find helpful. Explain how your design will enable meteorologists to find relevant past records, noting any assumptions that you make about the nature of the data.

[You should not go into unnecessary detail about the structure of the primary database. You may assume that expert meteorologists will select the data for the warehouse.]

[20 marks]

SECTION C

9 Semantics of Programming Languages

Write short notes on *four* of the following five topics.

(a) The relationship between three forms of operational semantics of the Language of Commands (LC) given by

- an evaluation relation $\langle P, s\rangle \Downarrow \langle V, s'\rangle$
- a transition relation $\langle P, s\rangle \to \langle P', s'\rangle$
- a transition relation between the configurations $\langle c, r, s\rangle$ of the SMC-machine

(b) The notion of *semantic equivalence* of LC phrases and its congruence property.

(c) *Call-by-name* and *call-by-value* rules for evaluating function applications in the Language of Functions and Procedures (LFP) and the relationship between the evaluation relations for LFP based upon each of them.

(d) The notion of *bisimilarity* of two configurations in a labelled transition system.

(e) The rules defining the possible labelled transitions of parallel composition ($P_1|P_2$) and restriction ($\nu\, c\,.\, P$) in the Language of Communicating Processes (LCP).

[5 marks each]

10 Foundations of Functional Programming

The following are some concepts that have flourished in the context of functional programming but which have (so far) been less heavily used in main-stream languages even when they have been available:

(a) polymorphic types

(b) type reconstruction

(c) higher-order functions

(d) lazy evaluation

(e) continuations

For *each* case give a brief explanation of the facility referred to, suggest a circumstance in which it might be useful and comment on how immediately relevant to non-functional languages it seems.

[4 marks per part]

11 Logic and Proof

(*a*) In the context of clause-based proof methods, define the notion of *pure literal* and describe what should be done if the set of clauses contains pure literals. [3 marks]

(*b*) Use the Davis–Putnam method to discover whether the following set of clauses is satisfiable. If they are satisfiable, show a satisfying interpretation.

$$\{P, R\} \qquad \{\neg P, \neg R\} \qquad \{P, \neg Q\} \qquad \{\neg Q, R\} \qquad \{\neg P, Q, R\}$$

[6 marks]

(*c*) The three-fingered inhabitants of the planet Triterra build base-3 computers. A Triterran named Randal Tryant has found a way of verifying base-3 combinational logic. His Ordered Ternary Decision Diagrams (OTDDs) are the same as a technology used on planet Earth except that all variables and expressions range over the values 0, 1 and 2 instead of just 0 and 1.

(*i*) Describe how a full ternary decision tree can be reduced to an OTDD without regard for efficiency. [2 marks]

(*ii*) Sketch an efficient algorithm to convert a ternary expression directly to an OTDD without constructing the full decision tree. For a typical ternary connective use modulo-3 multiplication, written as $\otimes$. [6 marks]

(*iii*) Demonstrate your algorithm by applying it to the ternary expression $((i \otimes i) \otimes j) \otimes 2$. [3 marks]

12 Complexity Theory

(*a*) Show that any language that can be accepted by a nondeterministic machine in *time* $f(n)$ can also be decided by a deterministic machine in *space* $O(f(n))$. [4 marks]

(*b*) Show that any language that can be accepted by a nondeterministic machine in *space* $f(n)$ can also be decided by a deterministic machine in *time* $O(c^{(f(n)+\log n)})$, for some constant c. [6 marks]

(*c*) Explain what the above results tell us about the inclusion relationships among the complexity classes:

NL, co-NL, P, NP, PSPACE and NPSPACE

[4 marks]

(*d*) It has been proved that the graph reachability problem is in co-NL. What further inclusions can you derive among the above complexity classes using this fact? Explain your answer. [6 marks]

END OF PAPER