COMPUTER SCIENCE TRIPOS Part Ib

Thursday 3 June 2004 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.*

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*) Describe an efficient algorithm to determine whether two finite line segments in a plane intersect. You may assume that the end points of each line are given as x–y coordinates. [8 marks]

(*b*) Describe in detail an efficient algorithm to find the convex hull of a set of points lying on a plane. Show that the complexity of the Graham scan used in the algorithm is $O(n)$ and that the algorithm as a whole has complexity $O(n \log n)$. [8 marks]

(*c*) Discuss how it is possible to eliminate many of the points before the convex hull algorithm is entered. [4 marks]

2 Computer Design

The ARM processor allows the second operand to be shifted by an arbitrary amount. In order to improve the performance, a six-stage pipeline is proposed with the following stages:

instruction fetch	decode and register fetch	shift operand 2	execute	memory access	register write back

(*a*) What are control hazards and how could they be resolved in the above pipeline? [4 marks]

(*b*) What are data hazards and how could they be resolved in the above pipeline? [4 marks]

(*c*) What are feed-forward paths and where could they be added to the above pipeline to improve performance? [6 marks]

(*d*) Why might a branch instruction result in pipeline bubbles and how many bubbles will appear in the above pipeline as a result of taking a branch instruction? [6 marks]

3 Digital Communication I

(*a*) Define the terms *latency* and *capacity* as applied to communication channels. [2 marks]

(*b*) Is there a strict relation between the two? [1 mark]

(*c*) Show how the latency of a channel can have a direct effect on the capacity of a higher-layer channel which uses it. [10 marks]

(*d*) How can the capacity of the higher-layer channel be improved (keeping the characteristics of the underlying channel unchanged)? [4 marks]

(*e*) In what circumstances might these improvements have only limited benefit? [3 marks]

4 Concurrent Systems and Applications

(*a*) A transaction processing system is using a *write-ahead log* as a mechanism for providing persistent storage. What information must be written to the log

(*i*) when a transaction starts;

(*ii*) when a transaction performs an operation on a persistent object;

(*iii*) when a transaction commits? [2 marks each]

(*b*) Describe how the log can be used to roll-back a transaction that has aborted or become deadlocked. [6 marks]

(*c*) The log can also be used to recover after some kinds of system failure.

(*i*) Describe how the log can be used to recover after a *fail-stop* crash. [2 marks]

(*ii*) What is meant by *checkpointing*? How will using it affect the structure of the log and the recovery procedure after a crash? [6 marks]

SECTION B

5 Computer Graphics and Image Processing

(*a*) Explain why display devices appear to be able to reproduce (almost) all the colours of the spectrum using only red, green and blue light. [4 marks]

(*b*) Describe an algorithm (other than thresholding) which will convert a greyscale image (8 bits per pixel) to a bi-level black and white image (1 bit per pixel), with the same number of pixels, while retaining as much detail as possible. [8 marks]

(*c*) Explain what specular and diffuse reflection are in the real world. State and explain equations for calculating approximations to both in a computer. [8 marks]

6 Databases

(*a*) Define the operators of the core relational algebra. [5 marks]

(*b*) Let R be a relation with schema $(A_1, \ldots, A_n, B_1, \ldots, B_m)$ and S be a relation with schema $(B_1, \ldots, B_m)$. The quotient of R and S, written $R \div S$, is the set of tuples t over attributes $(A_1, \ldots, A_n)$ such that for every tuple s in S, the tuple ts (i.e. the concatenation of tuples t and s) is a member of R. Define the quotient operator using the operators of the core relational algebra. [8 marks]

(*c*) The core relational algebra can be extended with a duplicate elimination operator, and a grouping operator.

(*i*) Define carefully these two operators. [3 marks]

(*ii*) Assuming the grouping operator, show how the duplicate elimination operator is, in fact, unnecessary. [2 marks]

(*iii*) Can the grouping operator be used to define the projection operator? Justify your answer. [2 marks]

7 Artificial Intelligence

In the following, N is a feedforward neural network architecture taking a vector

$$\mathbf{x}^T = (\ x_1 \quad x_2 \quad \cdots \quad x_n\ )$$

of n inputs. The complete collection of weights for the network is denoted $\mathbf{w}$ and the output produced by the network when applied to input $\mathbf{x}$ using weights $\mathbf{w}$ is denoted $N(\mathbf{w}, \mathbf{x})$. The number of outputs is arbitrary. We have a sequence $\mathbf{s}$ of m labelled training examples

$$\mathbf{s} = ((\mathbf{x}_1, \mathbf{l}_1), (\mathbf{x}_2, \mathbf{l}_2), \ldots, (\mathbf{x}_m, \mathbf{l}_m))$$

where the $\mathbf{l}_i$ denote vectors of desired outputs. Let $E(\mathbf{w}; (\mathbf{x}_i, \mathbf{l}_i))$ denote some measure of the error that N makes when applied to the ith labelled training example. Assuming that each node in the network computes a weighted summation of its inputs, followed by an activation function, such that the node j in the network computes a function

$$g\left(w_0^{(j)} + \sum_{i=1}^{k} w_i^{(j)} \mathrm{input}(i)\right)$$

of its k inputs, where g is some activation function, derive in full the backpropagation algorithm for calculating the gradient

$$\frac{\partial E}{\partial \mathbf{w}} = \left(\ \frac{\partial E}{\partial w_1} \quad \frac{\partial E}{\partial w_2} \quad \cdots \quad \frac{\partial E}{\partial w_W}\ \right)^T$$

for the ith labelled example, where $w_1, \ldots, w_W$ denotes the complete collection of W weights in the network.

[20 marks]

8 Compiler Construction

(a) Explain the differences (illustrating each with a small program) between

(i) static and dynamic binding (scoping); [4 marks]

(ii) static and dynamic typing. [2 marks]

(b) Java is sometimes said to be "dynamically typed" in that a variable whose type is (class) C can be assigned a value of (class) D provided that D extends C; conversely a variable of type D can be assigned a value of type C using a cast. By considering storage layouts, explain why the former assignment is always valid and the latter *sometimes* invalid. [4 marks]

(c) A new programming language has the notion of "statically scoped exceptions" in which the program

```
exception foo;
void f()
{   try
    {   void g() { raise foo; }
        try {
          g();
        }
        except (foo) { C2 }
    }
    except (foo) { C1 }
}
```

would execute `C1` rather than `C2` as the former was in scope at the `raise` point. By analogy with statically scoped variables, or otherwise, explain how such exceptions might be implemented on a stack. [10 marks]

SECTION C

9 Logic and Proof

For each of the following statements, briefly justify whether it is true or false. In the following x, y, z are variables, and a, b, c are constants.

(a) Given any propositional logic formula ϕ that is a tautology, converting ϕ to CNF will result in $\mathtt{t}$.

(b) Executing the DPLL method on the clauses

$$\{P, Q, \neg S\} \quad \{\neg P, Q, \neg R\} \quad \{P\} \quad \{\neg Q, R\} \quad \{S, \neg Q\}$$

produces a result without needing any case split steps.

(c) The OBDD corresponding to the propositional logic formula $(P \vee Q) \wedge \neg P$ does not have any decision nodes for the propositional letter P.

(d) Skolemizing the first order logic formula $\exists x(\phi(x))$ results in a logically equivalent formula $\phi(a)$ (where a is a fresh constant).

(e) The Herbrand Universe that is generated from the clauses $\{P(a)\}$, $\{Q(x, b), \neg P(x)\}$ and $\{\neg Q(a, y)\}$ contains two elements.

(f) The two terms $f(x, y, z)$ and $f(g(y, y), g(z, z), g(a, a))$ can be unified.

(g) It is not possible to resolve the clauses $\{P(x)\}$ and $\{\neg P(f(x))\}$ because the *occurs check* prevents the literals being unified.

(h) The clause $\{P(x, x), P(x, a)\}$ can be factored to give the new clause $\{P(x, a)\}$.

(i) The empty clause can be derived from the clauses $\{P(x), P(a)\}$, $\{P(x), \neg P(a)\}$, $\{\neg P(b), Q\}$ and $\{\neg P(c), \neg Q\}$ using resolution.

(j) Because in the modal logic S4 the equivalence $\Box\Box\phi \simeq \Box\phi$ holds for every formula ϕ, it follows that $\Diamond\Diamond\phi \simeq \Diamond\phi$.

[2 marks each]

10 Foundations of Functional Programming

Continuation passing allows lambda calculus and functional languages to describe many forms of sequential control structure. If you write code in your answer to this question you may write it either in lambda-calculus or in an equivalent ML-like syntax, but in the latter case you must not rely on ML's order of evaluation.

(*a*) Using the continuation passing style, show how to model the following impure code without use of `ref` or `:=` but so that the exact sequence of values that `k` takes during a calculation still arise.

```
val k = ref 0;
fun c(n, r) = (
    k := !k + r;
    if r=0 orelse r=n then 1
    else c(n-1,r-1) + c(n-1,r));
```

You may suppose that conditions and basic arithmetic are available to you as existing primitives, so `if`, `orelse` and `+` can all appear in your answer, even though `!`, `:=` and the use of `;` that indicates sequential execution must not. [14 marks]

(*b*) Discuss how exception-handling mechanisms such as ML's `raise` and `handle` can be mapped onto uses of continuations. Construct a small example to illustrate your explanation. [6 marks]

All code you write will be expected to be annotated so that it is easy for a human reader to see what it sets out to achieve: clarity of exposition will be considered much more important than exact syntactic validity in any particular programming language's syntax.

11 Semantics of Programming Languages

L1 has the expression syntax

$$\begin{aligned} e \ ::= \ & n \mid b \mid e_1 \ op \ e_2 \mid \textsf{if } e_1 \textsf{ then } e_2 \textsf{ else } e_3 \\ & \mid \ell := e \mid !\ell \mid \textsf{skip} \mid e_1; e_2 \mid \textsf{while } e_1 \textsf{ do } e_2 \end{aligned}$$

(a) Give the reduction rules for conditionals and while-loops. [3 marks]

(b) Define *semantic equivalence*, $e_1 \simeq^T_\Gamma e_2$, for L1. [4 marks]

(c) For each of the following pairs, state whether they are semantically equivalent; if not, state a nontrivial condition on the subexpressions e, e_1, e_2, e_3 that makes them so, and explain informally why it suffices.

(i) $$l := 3; e \stackrel{?}{\simeq} e; l := 3$$ [3 marks]

(ii) $$e; (\textsf{if } e_1 \textsf{ then } e_2 \textsf{ else } e_3) \stackrel{?}{\simeq} \textsf{if } e_1 \textsf{ then } e; e_2 \textsf{ else } e; e_3$$ [3 marks]

(iii) $$e; (\textsf{if } e_1 \textsf{ then } e_2 \textsf{ else } e_3) \stackrel{?}{\simeq} \textsf{if } e; e_1 \textsf{ then } e_2 \textsf{ else } e_3$$ [3 marks]

(iv) $$\textsf{while } !l \geq 0 \textsf{ do } (e_2; e_3)$$
$$\stackrel{?}{\simeq}$$
$$\textsf{if } !l \geq 0 \textsf{ then } e_2; (\textsf{while } !l \geq 0 \textsf{ do } (e_3; e_2)); e_3 \textsf{ else skip}$$
[4 marks]

12 Complexity Theory

(*a*) Define a one-way function. [4 marks]

(*b*) Explain why the existence of one-way functions would imply that P≠NP. [7 marks]

(*c*) Recall that `Reach` is the problem of deciding, given a graph G a source vertex s and a target vertex t, whether G contains a path from s to t; and `Sat` is the problem of deciding whether a given Boolean formula is satisfiable.

For each of the following statements, state whether it is true or false and justify your answer.

(*i*) If `Reach` is NP-complete then P=NP. [3 marks]

(*ii*) If `Reach` is NP-complete then NP≠PSPACE. [3 marks]

(*iii*) If `Sat` is PSPACE-complete then NP=PSPACE. [3 marks]

END OF PAPER