COMPUTER SCIENCE TRIPOS Part IB

Thursday 5 June 1997 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.*

SECTION A

1 Processor Architecture

The ARM instruction set (implemented by the ARM 7 processor core) contains a branch-with-link instruction (assembler mnemonic `BL`) to perform subroutine calls, but no specific instruction for returning from subroutine calls. How may returns be implemented for both complex and trivial subroutines? [8 marks]

A variant of the ARM 7 processor core can execute the conventional 32-bit ARM instruction set and also the 16-bit "Thumb" instruction set. Code compiled with the Thumb instruction set is typically 30% to 40% denser than code compiled with the 32-bit instruction set.

The ARM 7TDMI has a narrow external datapath (16 bits wide) for use in low-end embedded applications. Internally it has a unified cache with a 32-bit datapath. In what circumstances would Thumb code execute faster than the conventional 32-bit instructions? [12 marks]

2 Computer Architecture

What is the *PCI Local Bus* and when is it used? [5 marks]

What types of signals are defined in its specification? [5 marks]

Characterise operation of buses such as the PCI Local Bus in terms of basic read, basic write, and basic arbitration operations. [10 marks]

3 Digital Communication I

Explain the terms *ARQ protocol* and *window* of an ARQ protocol. [5 marks]

An ARQ protocol uses a window of 1 kbyte. The protocol is used over a link whose capacity is 1 Mbps. In the absence of transmission errors (or any other loss) determine (*a*) for a link delay of 100 μs, and (*b*) for a link delay of 250 ms, the time required to transfer each of the following amounts of information over the link:

1 kbyte, 1 Mbyte and 1 Gbyte [12 marks]

State and explain in which of these cases moving to a larger window size will not significantly improve the transfer time. [3 marks]

4 Computer Graphics and Image Processing

It is convenient to be able to specify colours in terms of a three-dimensional coordinate system. Three such coordinate systems are: RGB, HLS, L*a*b*.

Choose *two* of these three coordinate systems and for *each* of your chosen two:

(*a*) describe what each of the three coordinates represents [2 marks each]

(*b*) describe why the coordinate system is a useful representation of colour [2 marks each]

Draw *either* the first eight one-dimensional Haar basis functions *or* the first eight one-dimensional Walsh–Hadamard basis functions. [4 marks]

Calculate the coefficients of your chosen eight basis functions for the following one-dimensional image data:

12 16 20 24 24 16 8 8 [4 marks]

Explain why, in general, the Haar or Walsh–Hadamard encoded version of an image is preferable to the original image for storage or transmission. [4 marks]

SECTION B

5 Programming in C and C++

Write a declaration of a C++ class that might be used to implement a binary tree with each node able to hold an integer. Your implementation (i.e. the class itself and those bodies which conveniently fit within it) should make it impossible for casual programmers to access the pointer fields that link parts of the tree together except through cleanly specified access functions. Show how you would overload the "+" operator in C++ to provide a neat notation for adding a new item into such a tree. [20 marks]

6 Compiler Construction

Investigate whether the following grammar for regular expressions is SLR(1) by attempting to construct its Action and Goto matrices.

```
S -> R eof

R -> F | R + F

F -> P | F P

P -> x | ( R ) | P *
```

Find all the conflicts, if any, in the two matrices. [20 marks]

7 Prolog for Artificial Intelligence

A binary tree is constructed from binary compound terms $\mathtt{n}(a, b)$ called *nodes*, where components a and b are either nodes or integers. Suppose integer components are restricted to the values 0 and 1.

Write a Prolog program to return a list of all the 0's and a list of all the 1's in a given tree. For example, the goal `enum(n(n(0,1),1),X,Y)` should instantiate `X` to `[0]` and `Y` to `[1,1]`. The program is required to use difference lists. [20 marks]

8 Databases

What particular strengths of the relational model have led to the pre-eminent position that it holds today as a vehicle for database management? [8 marks]

Identify any weaknesses in the model, illustrating your answer by examples. [6 marks]

How might these weaknesses be remedied while retaining the advantages of the model? [6 marks]

SECTION C

9 Foundations of Functional Programming

The λ_I-calculus is a variant of the λ-calculus. The *terms* of the λ_I-calculus, known as λ_I-*terms*, are constructed recursively from a given set V of variables; the λ_I-terms take one of the following forms:

x	variable
$\lambda x.M$	abstraction, where M is a λ_I-term and $x \in \mathrm{FV}(M)$
MN	application, where M and N are λ_I-terms.

The set of free variables $\mathrm{FV}(M)$, $\beta\eta$-equality and $\beta\eta$-reduction are defined in a similar fashion to the corresponding λ-calculus definitions.

(*a*) Define an equality-preserving translation from λ_I-terms to combinators constructed using **I**, **B**, **C** and **S**. Indicate why these combinators are not enough to express the usual λ-calculus. [6 marks]

(*b*) Demonstrate the translation using the λ_I-term $\lambda x.\lambda y.(xMMy)$, where M is the identity function $\lambda z.z$. [3 marks]

(*c*) Define the Church numerals for the usual λ-calculus, and identify those numerals which are not λ_I-terms. [3 marks]

(*d*) By adapting the definition of the Church numerals, define λ_I-terms $\overline{n}$ for each $n \geqslant 0$ such that

$$\overline{n}\,MN = M^n(N) \text{ for } n \geqslant 1 \text{ and arbitrary } \lambda_I\text{-terms } M, N$$

$$\overline{0}\,\overline{m}\,\overline{n} = \overline{n} \text{ for arbitrary } m, n \geqslant 0.$$

Show that these equalities are indeed satisfied. [Hint: the λ_I-term shown in (*b*) may help to resolve a particular case.] [8 marks]

10 Logic and Proof

State the rules on sequents $\Gamma \Rightarrow \Delta$ involving the universal quantifier in first-order predicate calculus. [2 marks]

Give examples to illustrate the need for the side conditions on variable occurrences. [4 marks]

One of the sequent rules for the universal quantifier makes use of the notion of substituting a term for a variable in a formula. Give an example to show what goes wrong if a free variable in the term being substituted becomes bound or "captured" after substitution. [3 marks]

Let the notation $A\langle t/x\rangle$ denote the result of substituting t for *some* occurrence of x in A if no variable in t becomes bound after substitution. Assume the usual first-order sequent calculus rules (including *cut*), together with all sequents of the form

$$\Gamma \;\Rightarrow\; t = t,\; \Delta$$

and

$$\Gamma,\; t_1 = t_2,\; A\langle t_1/x\rangle \;\Rightarrow\; A\langle t_2/x\rangle,\; \Delta$$

where t, t_1 and t_2 range over arbitrary terms, x is a variable and the substitutions with t_1 and t_2 are for the same occurrence of x.

Give an informal argument that these two rules for "=" are sound principles for reasoning about equality. [2 marks]

Prove that:

(*a*) $\Rightarrow\; \forall x\, \forall y\, ((x = y) \to (y = x))$ [3 marks]

(*b*) $\Rightarrow\; \forall x\, \forall y\, \forall z\, ((x = y) \;\wedge\; (y = z) \to (x = z))$ [3 marks]

(*c*) $\Rightarrow\; \forall y\, ((\exists x\, (x = y) \;\wedge\; P(x)) \to\; P(y))$ [3 marks]

11 Complexity Theory

Suppose you had a conventional sequential computer with a special coprocessor which could multiply two n-bit numbers in time proportional to $\log(n)$, even for very large n. Explain how you would implement a fast integer square root program on this system. Comment on the performance you could expect to achieve. [20 marks]

12 Semantics of Programming Languages

A language L of expressions, E, for infinite lists of integers is given by:

$$E ::= \mathbf{ints} \mid \mathbf{from}(n) \mid n : E \mid \mathbf{incr}(E)$$

where n ranges over the integers. A binary relation of evaluation between L-expressions is inductively defined by the following axioms and rule:

$$(\Downarrow_{\text{ints}}) \qquad \mathbf{ints} \Downarrow 0 : \mathbf{incr}(\mathbf{ints}).$$

$$(\Downarrow_{\text{from}}) \qquad \mathbf{from}(n) \Downarrow n : \mathbf{from}(n') \qquad \text{where } n' = n + 1.$$

$$(\Downarrow_{:}) \qquad n : E \Downarrow n : E.$$

$$(\Downarrow_{\text{incr}}) \qquad \frac{E \Downarrow n : E'}{\mathbf{incr}(E) \Downarrow n' : \mathbf{incr}(E')} \qquad \text{where } n' = n + 1.$$

Prove that for every E, there are unique n and E' such that $E \Downarrow n : E'$. [5 marks]

A binary relation $\mathcal{R}$ between L-expressions is called a *bisimulation* if whenever $(E_1, E_2) \in \mathcal{R}$ then $E_1 \Downarrow n : E_1'$ and $E_2 \Downarrow n : E_2'$ hold for some n and some $(E_1', E_2') \in \mathcal{R}$. We write $E_1 \approx E_2$ if E_1 and E_2 are related by some bisimulation.

Prove that $\mathbf{ints} \approx \mathbf{from}(0)$. [5 marks]

Prove that $\approx$ has the congruence property for the language L, i.e. that if $E_1 \approx E_2$, then $E[E_1] \approx E[E_2]$ (where $E[E_1]$ is any L-expression containing an occurrence of E_1 and $E[E_2]$ denotes the result of replacing that occurrence by E_2). [Hint: show that $\approx$ is preserved by the operations $n : -$ and $\mathbf{incr}(-)$, by constructing suitable bisimulations.] [5 marks]

State, with justification, whether or not $\approx$ is an equivalence relation. [5 marks]