COMPUTER SCIENCE TRIPOS Part IB

Wednesday 5 June 1996 1.30 to 4.30

Paper 5

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.*
Write on **one** *side of the paper only.*

SECTION A

1 Processor Architecture

What is a *branch delay slot* and why does it arise? [7 marks]

How can branch delays be avoided? [7 marks]

If a processor exhibited one branch delay slot how would you reorder (and possibly modify) the instructions in the following loop to gain a performance advantage?

```
loop
        ldr r2,r3,#4      % r2=load(r3), r3=r3+4
        add r4,r4,r2      % r4=r4+r2
        add r1,r1,#1      % r1=r1+1
        cmp r1,#10        % compare r1 with constant 10
        bne loop          % branch if not equal to loop
```

[6 marks]

2 Computer Architecture

Write short notes on *each* of the following parameters of cache design:

(*a*) size [4 marks]

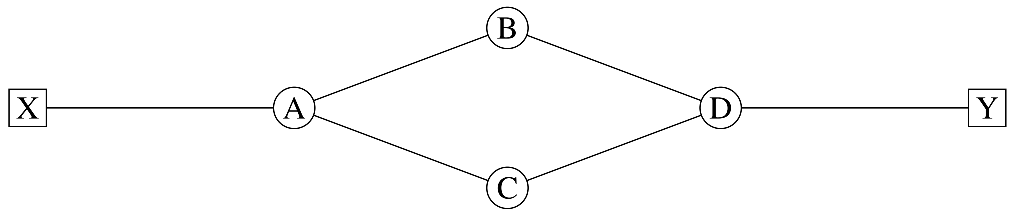
(*b*) mapping function [4 marks]

(*c*) replacement algorithm [4 marks]

(*d*) write policy [4 marks]

(*e*) block size [4 marks]

3 Digital Communication I



Hosts X and Y are communicating through the data network provided by the switches A, B, C and D and the links interconnecting them as shown above. Initially all packets are travelling through switches A, C and D.

(*a*) A packet is corrupted on the link between C and D. Describe the events that take place to recover from the error when

(*i*) an end to end flow and error control protocol is in operation [5 marks]

(*ii*) flow and error control are performed on a hop by hop basis [5 marks]

(*b*) Switch C fails. Describe the events that follow to recover when

(*i*) the network is a datagram network [5 marks]

(*ii*) the network is connection oriented [5 marks]

4 Graphics

Consider the control of detail in a curve represented as a sequence of straight line segments.

Describe Douglas and Peucker's algorithm for removing superfluous points.

[10 marks]

Describe how Overhauser interpolation can be used to introduce additional points.

[10 marks]

SECTION B

5 Programming in C and C++

For *five* of the following C or C++ features write a very short fragment of code (perhaps 2 or 3 lines will suffice in most cases) that illustrates the syntax involved. In each case explain very briefly what your example achieves.

(*a*) preprocessor macros and conditional compilation

(*b*) casts that convert from one pointer type to another

(*c*) C and C++ style comments

(*d*) the declaration of a simple C++ class

(*e*) overloading the operator '`+`'

(*f*) the C `setjmp` function

(*g*) the `switch` statement, including a `default` label

[4 marks each]

6 Compiler Construction

Outline the key features of the design of the part of a compiler that will translate the abstract syntax tree representation of a program into a stack-based intermediate code. Concentrate on those features used in the translation of the following fragment:

```
...
LET i = k
LET j = k
WHILE (i>0) AND (j<100) DO { i := i-1; j := j+2 }
...
```

In particular, concentrate on the mechanism you would choose to deal with

(*a*) the scopes of identifiers [6 marks]

(*b*) the compilation of boolean expressions involving the operators `NOT`, `AND` and `OR` [6 marks]

(*c*) the translation of the `WHILE` command [4 marks]

(*d*) the translation of the two assignments [4 marks]

7 Prolog for Artificial Intelligence

An *ordered integer binary search tree* (or OIBS tree) is either empty or a tuple (T, N, U), where T and U are also OIBS trees and N is an integer. Every node in T has a value less than N, which in turn is less than the value of every node in U.

(*a*) Give two Prolog terms which are suitable for representing an empty OIBS tree and a node in the OIBS tree respectively. [2 marks]

(*b*) Define a Prolog procedure `insert(Item, T, NT)`, where `Item` is an integer being inserted into OIBS tree `T`, producing an OIBS tree `NT`. If `Item` is already present in `T`, then `NT` equals `T`. [9 marks]

(*c*) Define a Prolog procedure `lookup(Item, T)`, where `Item` is to be looked for in OIBS tree `T`. A `lookup` goal will succeed if `Item` is found, or fail otherwise. [9 marks]

8 Databases

Describe the essentials of the ODMG-93 standard for Object Database Management. [7 marks]

To what extent do these proposals conform to the ANSI/SPARC architecture for database management? [3 marks]

Describe how binary relationships can be modelled directly within the ODMG-93 standard. [4 marks]

In what way is it possible to create a representation for n-ary relationships that is similar to that of the relational model? [2 marks]

Explain how these alternative approaches allow a navigational style of data manipulation as well as supporting an extension of SQL. [4 marks]

SECTION C

9 Foundations of Functional Programming

Consider binary trees that are either empty (written **Lf**) or have the form $\mathbf{Br}\ x\,t_1\,t_2$ where t_1 and t_2 are themselves binary trees. Give an encoding of binary trees in the λ-calculus, including functions **isLeaf**, **label**, **left** and **right** satisfying

$$\begin{aligned}
\mathbf{isLeaf}\ \mathbf{Lf} &\twoheadrightarrow \mathbf{true} \\
\mathbf{isLeaf}\,(\mathbf{Br}\ x\,t_1\,t_2) &\twoheadrightarrow \mathbf{false} \\
\mathbf{label}\,(\mathbf{Br}\ x\,t_1\,t_2) &\twoheadrightarrow x \\
\mathbf{left}\,(\mathbf{Br}\ x\,t_1\,t_2) &\twoheadrightarrow t_1 \\
\mathbf{right}\,(\mathbf{Br}\ x\,t_1\,t_2) &\twoheadrightarrow t_2
\end{aligned}$$

If you use encodings of other data structures, state the properties assumed.

[6 marks]

Consider the ML functions f and g defined to satisfy

$$\begin{aligned}
f([], ys) &= ys \\
f(x :: xs, ys) &= f(xs, x :: ys) \\
g([], ys) &= ys \\
g(x :: xs, ys) &= x :: g(xs, ys)
\end{aligned}$$

Using list induction, prove $f(f(xs, []), []) = xs$. [14 marks]

[Hint: generalize this formula, making use of g. You may assume the equation $g(xs, []) = xs$.]

10 Logic and Proof

For each of the following formulæ, construct a proof in the tableau calculus or show that no proof exists.

$((A \to B) \to A) \to A$ [4 marks]

$\forall z\, \exists x\, \forall y\, ((P(y) \to Q(z)) \to (P(x) \to Q(x)))$ [12 marks]

$\Box(A \to B) \wedge \Box(B \to A)$ [4 marks]

Assume S4 modal logic for the last one.

11 Complexity Theory

(*a*) Show that the problem 3-SAT is at least as hard to solve as n-SAT. [5 marks]

(*b*) Show that the task of finding a minimum cost closed circuit in a weighted directed graph (a Travelling Salesman Problem of the minimization variety) is at least as hard as the Hamiltonian Circuit Problem. [5 marks]

(*c*) Show that the class NP-complete is contained in the class P-space. [5 marks]

(*d*) Show that the class P-space is contained in the class EXP-time. [5 marks]

In each case ensure that your answer makes it clear what the problems and classes involved are. Standard results do not need to be proved provided they are clearly stated.

12 Semantics

The abstract syntax of commands in a simple parallel programming language P is given by

$$C ::= \texttt{skip} \mid X := ie \mid C_1 \,; C_2 \mid \texttt{if}\ be\ \texttt{then}\ C_1\ \texttt{else}\ C_2 \mid \texttt{while}\ be\ \texttt{do}\ C \mid C_1 \parallel C_2$$

where *ie*, *be* and X range over the syntactic categories of integer expressions, boolean expressions and program variables, respectively. The intended behaviour of $C_1 \parallel C_2$ is that C_1 and C_2 are executed in parallel until they have both terminated. Hence atomic execution steps from C_1 and C_2 may be arbitrarily interleaved. The other command forms behave as usual.

(*a*) Give a small-step transition semantics for P which derives statements of the form $\langle C, S\rangle \rightarrow \langle C', S'\rangle$, where S and S' are states. You may assume that rules for the evaluation of expressions have already been given.

Comment briefly on your choice of what constitutes an atomic execution step. [9 marks]

(*b*) The binary relation $\sim$ on commands is defined by

$$C_1 \sim C_2 \equiv \forall S, S'.\ \langle C_1, S\rangle \rightarrow^* \langle \texttt{skip}, S'\rangle \iff \langle C_2, S\rangle \rightarrow^* \langle \texttt{skip}, S'\rangle.$$

Show that $\sim$ is *not* a congruence. [5 marks]

(*c*) Assuming that $S(X) = S(Y) = 0$, describe the set of possible execution traces which are derivable in your semantics starting from the configuration $\langle C, S\rangle$, where C is

$$(X := 1) \parallel (\texttt{while}\ X = 0\ \texttt{do}\ Y := Y + 1).$$

Why might one argue that this does not accurately reflect the behaviour of a reasonable implementation of the language? [6 marks]