

COMPUTER SCIENCE TRIPOS Part IB

Wednesday 1 June 1994 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 Introduction to Computer Architecture

The MIPS R2000 is a 32-bit architecture and has three basic instruction formats:

I type	op	rs	rt	immediate		
	6	5	5	16		

J type	op	target address				
	6	26				

R type	op	rs	rt	rd	re	funct
	6	5	5	5	5	6

Discuss the different ways in which addresses may be constructed for

(a) data reference [8 marks]

(b) effecting transfer of control [12 marks]

2 Computer Structures

What are the parameters which characterise the operation of a cache? [8 marks]

Describe three methods of mapping memory blocks to cache blocks. [7 marks]

Would the cache be a help or a hindrance if the CPU were used to handle continuously changing data such as video? [5 marks]

3 Digital Communication I

Define the term *circuit* as used in “circuit switching”. [4 marks]

Sketch the design of the in-band switching function of a circuit switch which switches 4×2 Mbps trunks each supporting 32×64 Kbps channels. [8 marks]

Describe how you would augment this design to allow the set up and clearing of connections. You should invent your own simple protocol for this purpose. [8 marks]

4 Graphics

Discuss sampling artifacts and their effect on image quality on a raster display. [10 marks]

What can be done to reduce or eliminate them? [10 marks]

SECTION B

5 Programming in C

Write a program in C which, given two integer inputs J and K , will output the combinations of J things partitioned into K groups. For example, if $J = 5$ and $K = 3$, the output would be:

(5,0,0)

(4,1,0)

(3,2,0)

(3,1,1)

(2,2,1)

[20 marks]

6 Programming Language Compilation

Suggest how you would represent Lambda-expressions in a form suitable for a simple Lambda-evaluator in which the environment is represented as a linked list of name-value pairs. [5 marks]

Outline the design of the evaluator, paying particular attention to the treatment of bound variables, abstractions and applications. [5 marks]

Augment your evaluator to cope with sufficient expression operators, including a built-in version of the Y operator so that when the evaluator is given the abstract syntax tree for

$$Y (\lambda f. \lambda n. n=0 \rightarrow 1, n*f(n-1)) 5$$

it will yield 120. [10 marks]

7 Concurrent Systems

Semaphores are used to achieve exclusive access to a resource for writing but to allow simultaneous read-only access. *Reader* and *Writer* processes have the following structure:

```

call a procedure to acquire the resource;
use the resource;
call a procedure to release the resource.

```

The procedures to acquire the resource *for reading* and to release it are as shown below. Writers have priority over readers.

```

procedure acquire_for_reading()
  WAIT(CGUARD);
  ar := ar + 1;      # increment the count of active readers
  if aw = 0 then    # if there are no active writers
    begin
      rr := rr + 1; # increment the count of reading readers
      SIGNAL(READ)
    end;
  SIGNAL(CGUARD);
  WAIT(READ);

```

```

procedure release_after_reading()
  WAIT(CGUARD);
  rr := rr - 1;
  ar := ar - 1;
  if rr = 0 then
    while ww < aw do # ww counts "writing writers"
      begin
        ww := ww + 1;
        SIGNAL(WRITE)
      end;
  SIGNAL(CGUARD);

```

```

# note that "writing writers" are assumed to take turns to use
# the resource.

```

What is the function of the operations on the semaphore CGUARD? [3 marks]

Why is the following, apparently simpler, version of `acquire_for_reading()` incorrect?

```

WAIT(CGUARD);
ar := ar + 1;
if aw  $\neq$  0 then WAIT(READ);
rr := rr + 1;
SIGNAL(CGUARD);

```

[3 marks]

What is the function of the operations on the semaphores READ and WRITE? [4 marks]

The procedures to acquire the resource for writing and to release it after writing are similar to those given but, in addition, exclusive access to the resource must be enforced. Outline how this could be programmed. [2 marks]

Write the procedure `release_after_writing()`. [8 marks]

8 Databases

What are the primitives in terms of which a network data model is presented? [4 marks]

Explain how such a data model is specified in the CODASYL Schema Data Definition Language. [6 marks]

How does an application programmer in COBOL gain access to a CODASYL database? [3 marks]

Describe the COBOL programmer's model of the database, illustrating your answer with suitable data manipulation commands. [7 marks]

SECTION C**9 Foundations of Logic Programming**

What are Herbrand models and what are their applications to Computer Science?
[5 marks]

Obtain an Herbrand universe, showing all steps, starting from the following set of axioms:

$$(\exists x y)\neg(x < y) \quad (\forall x)(\exists y)x < y$$

Then take two models such that $<$ denotes “less than” on the natural numbers and obtain two distinct Herbrand models.
[8 marks]

A dual form of Skolemisation, called \forall -Skolemisation, is like ordinary Skolemisation except that it exchanges the rôles of the quantifiers. It eliminates universal quantifiers, instead of existential quantifiers, using Skolem functions.

Prove that \forall -Skolemisation preserves validity and discuss its possible applications.
[7 marks]

10 Foundations of Functional Programming

Describe precisely the meaning and main properties of the equality $M = N$, where M and N are terms of the λ -calculus. [5 marks]

In the following, consider an encoding of lists $[a_1, a_2, \dots, a_m]$ as the λ -term

$$\lambda f x. f a_1 (f a_2 \dots (f a_m x) \dots).$$

Answers should include a brief justification. You may assume λ -encodings of the booleans and ordered pairs.

Define the λ -term **cons** such that

$$\mathbf{cons} a [a_1, \dots, a_m] = [a, a_1, \dots, a_m] \quad [2 \text{ marks}]$$

Define the λ -term **null** such that

$$\mathbf{null} [a_1, \dots, a_m] = \begin{cases} \mathbf{true} & (\text{if } m = 0) \\ \mathbf{false} & (\text{if } m > 0) \end{cases} \quad [3 \text{ marks}]$$

Define the λ -term **append** such that

$$\mathbf{append} [a_1, \dots, a_m] [b_1, \dots, b_n] = [a_1, \dots, a_m, b_1, \dots, b_n] \quad [3 \text{ marks}]$$

Define the λ -terms **hd** and **tl** such that, if $m > 0$,

$$\begin{aligned} \mathbf{hd} [a_1, \dots, a_m] &= a_1 \\ \mathbf{tl} [a_1, \dots, a_m] &= [a_2, \dots, a_m] \end{aligned} \quad [7 \text{ marks}]$$

11 Computation Theory

The Halting Problem for register machines is unsolvable. State, without proof, a precise form of this result. [3 marks]

Let the computation by program p on data d be represented by the natural number k that codes the pair (p, d) . By considering the set $H(k)$ of the HALTING computations represented by codes $k' \leq k$, show that there is an increasing total function $h(k)$ which *grows too fast* to be computable. [6 marks]

Given $h : \mathbb{N} \rightarrow \mathbb{N}$ with the above property

$$\begin{aligned} \text{let } f(k) &= h(k) + k \\ \text{and } g(x) &= \sup\{k : f(k) \leq x\}. \end{aligned}$$

Then $f : \mathbb{N} \rightarrow \mathbb{N}$ is strictly increasing, and $g : \mathbb{N} \rightarrow \mathbb{N}$ satisfies

$$g(f(k)) = k, \quad g(x) < k \quad \text{for all } x < f(k).$$

Show that g *grows too slowly* to be computable in the following sense... given $G : \mathbb{N} \rightarrow \mathbb{N}$ such that

- (a) $\{G(n) : n \in \mathbb{N}\}$ is unbounded
- (b) $G(n) \leq g(n)$ for all $n \in \mathbb{N}$

then $G(n)$ is *not* computable. [11 marks]

12 Software Engineering

Discuss the reasons why substantial software systems often take longer to implement than planned and do not perform as well as expected. [10 marks]

If you were in charge of a large team implementing such a system, how would you ensure that the product was both reliable and finished on time? [10 marks]