COMPUTER SCIENCE TRIPOS Part II

Wednesday 2 June 1993 1.30 to 4.30

Paper 8

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 Digital Signal Processing

Show that an N-point Discrete Fourier Transform (DFT)

$$X(p) = \sum_{n=0}^{N-1} x(n)e^{-j2\pi np/N}$$

may be evaluated in terms of two $\frac{N}{2}$-point DFTs if N is even. [6 marks]

Without giving a detailed mathematical derivation, discuss how this result may be used to give the Fast Fourier Transform algorithm. Discuss the advantages of the algorithm compared with direct evaluation of the DFT. [5 marks]

Discuss briefly the use of window functions in discrete spectrum analysis. [3 marks]

The generalised Hamming window function is defined by

$$\begin{aligned} w(n) &= \alpha - (1-\alpha)\cos(2\pi n/N) \quad \text{for } 0 \leqslant n < N \\ &= 0 \quad \text{otherwise} \\ \text{where} \quad & 0 \leqslant \alpha \leqslant 1. \end{aligned}$$

Obtain an expression for the DFT of this window function. [6 marks]

2 Digital Communication II

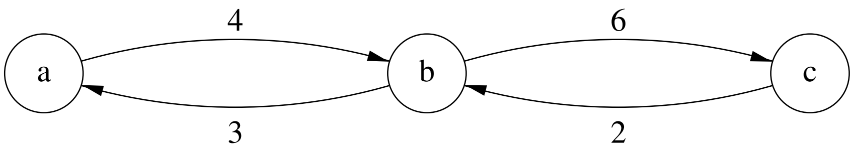
Define the layers of the OSI Reference Model and briefly describe their primary functions. [7 marks]

Using this model, describe the significant differences between the protocols of a traditional file transfer mechanism and a distributed file system. [9 marks]

How is security supported by the two mechanisms? [4 marks]

3 Computer System Modelling

The state diagram for a Markov chain showing transition rates is shown below. Solve for state occupancy probabilities.



[5 marks]

The steady state distribution for the number of jobs in an $M/M/1$ queue, k, is

$$p_k = (1-\rho)\rho^k \qquad k = 0, 1, 2, \ldots$$

where $\rho = \lambda/\mu < 1$. Here λ and μ are the mean arrival rate and mean service rate respectively.

Find the first and second moments of this distribution and hence verify that the variance of the number of the jobs in the system is given by

$$\frac{\rho}{(1-\rho)^2}$$

[10 marks]

What does this result show about the predictability of system performance at high loads? [5 marks]

4 Developments in Technology

EDSAC (1949) was the first practical stored-program computer in operation. Although it had no subroutine jump instruction, a method for using subroutines was devised by David Wheeler. Quote and explain the instruction sequences for subroutine entry and return. [12 marks]

Describe the corresponding instruction sequences that you would use in a modern processor. [8 marks]

SECTION B

5 Designing Interactive Applications

Distinguish the terms *needs analysis* and *requirements analysis*. Provide an example to illustrate the difference. [3 marks]

What is a *strong requirement*? Provide counter-examples and explain why each example is not a strong requirement. [4 marks]

What role does a *functional specification* play in a requirements specification? Give a one-sentence example. [3 marks]

The receptionist at a small research laboratory is required to field incoming messages and make sure that they reach the recipient in a timely manner. Some messages arrive by word of mouth, others by phone, courier, e-mail or FAX. There are about 100 recipients, most of whom are researchers. They spend a large proportion of their time in meetings of one sort or another, some of which are held in offices, the remainder in conference rooms. The receptionist endeavours to avoid interrupting important meetings unnecessarily.

It is proposed to build a system based upon Active Badge technology to improve message handling activities in the laboratory. Each member of staff wears an active badge. An existing Location Server provides client applications with up-to-date information about the location and movements of each active badge.

Sketch out the methods you would employ to establish the users' needs for the proposed system. Describe the categories of information you would pass on to the designer and illustrate each with one or two examples. [10 marks]

6 Optimising Compilers

Many modern architectures have provision for only 32-bit values in registers. However, ANSI C requires code such as

```
extern void g(int);
extern void f(int x)
{   char c = x;
    c += 1;
    g(c);
}
```

to have the effect that a call to `f()` causes `g()` to receive a parameter value as though `c` were stored in memory, i.e. in the range `CHAR_MIN` to `CHAR_MAX`. You may assume that `char` holds 8-bit values and the range is either `-128..127` or `0..255`.

While this example clearly requires range reduction following the incrementation of `c`, it is claimed that this is not always necessary.

One brutal technique to allocate `char` variables to registers is to treat them as `int` variables and achieve ANSI C conformance by range reduction just before each reference in the same manner that load-byte hardware might.

Develop an optimisation technique which might reduce the amount (static or dynamic) of such range reduction in code like:

```
extern char p(int v[100])
{   unsigned char r = 0;
    int i;
    for (i=0; i<100; i++)
        r = (r<<1 ^ r) + v[i];
    return (r & 128) != 0;
}
```

Little credit will be given for mere hand-compilation. [20 marks]

Hints: 1. Consider similarities to live variable analysis.
2. Consider whether register reads or writes occur more often.

7 Artificial Intelligence I

A sliding-tile puzzle consists of three black tiles, three white tiles and an empty space, thus:

B_1	B_2	B_3		W_1	W_2	W_3

There are three legal ways of moving a tile, each with an associated cost:

slide into the adjacent empty location — cost 1

jump over one tile into the empty location — cost 1

jump over two tiles into the empty location — cost 2

The goal is to have all the white tiles to the left of all the black tiles and to achieve this at minimum cost. The final position of the empty space is not important.

(*a*) Represent the problem using the following knowledge representation schemes:

(*i*) production system rules [5 marks]

(*ii*) a semantic network [5 marks]

In one sentence, describe the different emphases of these two schemes. [1 mark]

(*b*) State two possible heuristics to help solve this problem. [2 marks]

(*c*) For a planner to solve this puzzle, what operators (i.e. planning actions) would be needed? [7 marks]

8 Database Topics

Explain the ways in which the data modelling concepts 'IS-A', 'IS-PART-OF' and 'IS-AN-INSTANCE-OF' have been addressed in database design methodology. Your discussion should include the network, relational and functional data models, as well as object-oriented databases. [20 marks]

SECTION C

9 Natural Language Processing

Describe the main differences between formal languages (such as logics or programming languages) and a natural language (such as English). [8 marks]

What different requirements do these differences place on techniques for

(a) syntactically parsing

(b) semantically interpreting

natural, as opposed to formal, languages? [6+6 marks]

10 Semantics

State the principle of fixed point induction. [4 marks]

D is a cpo of 'integer streams': it comes equipped with a continuous function $in : (\mathbb{Z} \times D)_\perp \to D$ that possesses a continuous inverse $out : D \to (\mathbb{Z} \times D)_\perp$. (Thus the composition of in and out in either order is the appropriate identity function.) Moreover D has the property that the identity function $id_D \in (D \to D)$ is the least fixed point of the continuous function $\delta : (D \to D) \to (D \to D)$ which maps $f \in (D \to D)$ to $\delta(f) \in (D \to D)$, where for each $d \in D$

$$\delta(f)(d) = \begin{cases} in([(n, f(x))]) & \text{if } out(d) = [(n, x)] \\ in(\perp) & \text{if } out(d) = \perp \end{cases}$$

Let $mapS : D \to D$ be a continuous function satisfying that for all $d \in D$

$$mapS(d) = \begin{cases} in([(n+1, mapS(x))]) & \text{if } out(d) = [(n, x)] \\ in(\perp) & \text{if } out(d) = \perp \end{cases}$$

Using fixed point induction for δ, show that there is at most one solution $d \in D$ to the equation

$$d = in([(0, mapS(d))])$$

Hint: if d_1 and d_2 are both solutions, consider the property of $f \in (D \to D)$ given by '$f\, mapS = mapS\, f$ and $f(d_1) \sqsubseteq d_2$'. [16 marks]

11 Types

Explain what is meant by the statement 'every closed, typable expression in ML possesses a *principal* type'. Does a similar property hold for the second order lambda calculus $\lambda 2$? [4 marks]

What is meant by a *type scheme* in ML? Give the rules for inductively defining ML type assertions of the form

$$\Gamma \vdash M : \rho$$

where ρ is a type scheme, Γ is a finite function from identifiers to type schemes, and M is an ML expression built up from identifiers using let-expressions, lambda abstractions and function applications. [9 marks]

This fragment of ML is augmented by fixed-point expressions $\mathsf{fix}\, x.M$, which are typed according to the rule

$$\frac{\Gamma, x : \rho \vdash M : \rho}{\Gamma \vdash \mathsf{fix}\, x.M : \rho} \quad \text{(FIX)}$$

(Free occurrences of x in M become bound in $\mathsf{fix}\, x.M$.) Show that the closed expression $\mathsf{fix}\, x.\lambda y.(xx)y$ is typable in this augmented system. Hint: find a type scheme ρ for which $x : \rho \vdash \lambda y.(xx)y : \alpha \to \beta$ holds. [4 marks]

Is the expression typable if the use of the rule (FIX) is restricted by requiring ρ to be a type rather than a type scheme? [3 marks]

12 Concurrency

State the expansion law for observation congruence between CCS agents.

[4 marks]

A counter which can assume any of the natural numbers as its state is specified as a CCS agent by:

$$\begin{aligned} \mathit{Count}_0 &\stackrel{\text{def}}{=} \mathit{inc}.\mathit{Count}_1 + \mathit{zero}.\mathit{Count}_0 \\ \mathit{Count}_n &\stackrel{\text{def}}{=} \mathit{inc}.\mathit{Count}_{n+1} + \mathit{dec}.\mathit{Count}_{n-1} \qquad \text{for } n > 0 \end{aligned}$$

It is required to implement the counter by linking together several copies of an agent C and one agent B, where

$$\begin{aligned} C &\stackrel{\text{def}}{=} \mathit{inc}.(C \frown C) + \mathit{dec}.D \\ D &\stackrel{\text{def}}{=} \overline{d}.C + \overline{z}.B \\ B &\stackrel{\text{def}}{=} \mathit{inc}.(C \frown B) + \mathit{zero}.B \end{aligned}$$

Here the *linking*, $P \frown Q$, of two agents P and Q is an abbreviation for

$$(P[f] \mid Q[g]) \setminus L$$

where

$$L = \{i', z', d'\} \quad \text{and} \quad f(\ell) = \begin{cases} i' & \text{if } \ell = i \\ z' & \text{if } \ell = z \\ d' & \text{if } \ell = d \\ \ell & \text{otherwise} \end{cases} \quad \text{and} \quad g(\ell) = \begin{cases} i' & \text{if } \ell = \mathit{inc} \\ z' & \text{if } \ell = \mathit{zero} \\ d' & \text{if } \ell = \mathit{dec} \\ \ell & \text{otherwise} \end{cases}$$

(*a*) Use the expansion law to prove that $D \frown C \approx C \frown D$ and $D \frown B \approx B \frown B$. (You may assume without proof that $\tau.P \approx P$, for any P.) [5 marks]

(*b*) Letting $C^{(0)}$ stand for B and $C^{(n)}$ (when $n > 0$) stand for

$$\underbrace{(C \frown \cdots (C \frown (C}_{n \text{ copies of } C} \frown B)) \cdots)$$

show that $D \frown C^{(n)} \approx C^{(n)}$. Deduce that $C^{(n)}$ satisfies the defining equations for Count_n up to observation congruence. (You may assume without proof that $B \frown B \approx B$, that observation equivalence is a congruence relation for the operation of linking and that linking is associative up to observation equivalence. State carefully any other general principles you use.) [11 marks]