COMPUTER SCIENCE TRIPOS Part II

Wednesday 7 June 1995 1.30 to 4.30

Paper 8

Answer **five** *questions.*
Submit the answers in five **separate** *bundles each with its own cover sheet.*
Write on **one** *side of the paper only.*

1 Specification and Verification I

Explain the term *partial correctness specification.* Illustrate your answer by giving and explaining the primitive rules of partial correctness for assignment, sequencing and while commands. [5 marks]

Are your rules complete for a language consisting of only these commands? Explain your answer. [3 marks]

Derive a rule for partial correctness statements of the form

```
{P} WHILE K≠E DO C; K:=K+1 OD {Q}
```

where P and Q are arbitrary statements. Give the derivation. [4 marks]

Give a suitable invariant for the loop in the partial correctness statement

```
{T}
Y:=N; X:=1; WHILE X≠N DO Y:=Y×X; X:=X+1 OD
{X=N ∧ Y=N!}
```

[2 marks]

Using your derived rule prove that this partial correctness statement is true. State clearly any mathematical facts you use. [6 marks]

2 Designing Interactive Applications

The figure below shows the controls for a telephone answering machine, designed to meet requirements for low manufacturing cost, efficient operation and ease of learning. The controls include:

(A) a button to *start* recording (of outgoing message) or playback (of outgoing or incoming messages);

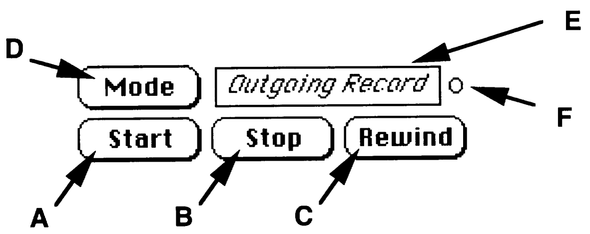
(B) a button to *stop* recording, playback or rewind;

(C) a button to *rewind* the incoming-message tape;

(D) a button to step between the four *modes*: outgoing-message-play, outgoing-message-record, incoming-messages-play and incoming-messages-record (the normal mode for awaiting calls);

(E) a display indicating the current mode;

(F) a light that flashes to show how many messages have been received.



What forms of mental model are likely to be acquired by users of the answering machine, both owners (that is, recipients of calls) and callers? [5 marks]

An additional feature is proposed for the answering machine, allowing the owner to record special outgoing messages, each one to be played when calls are received from an owner-specified telephone number. To record such a message, the owner of the machine sets it in outgoing-message-record mode, keys in the telephone number in question using the attached telephone's push-buttons, and records the message in the normal way, using the *start* and *stop* buttons. Up to 16 special outgoing messages, each for a different incoming telephone number, can be recorded in this manner.

You have been asked to review this proposed new feature, in terms of the owner's ability to learn to use the machine in an error-free manner. What problems can you identify in the feature's design? Propose solutions that are consistent with the original requirements. [15 marks]

3 Digital Communication II

Discuss the relative merits of *shared-media* versus *switch-based* local area networks. [12 marks]

Describe the operation of *cut-through* routing in a switch. What does it say about the switch fabric? [8 marks]

4 Concurrency Theory

Explain how the relation of *observational congruence* ($=$) between CCS agents is defined in terms of observation equivalence ($\approx$). [2 marks]

Say that an agent *can deadlock* if it can perform a sequence of actions to become an agent observationally congruent to 0. For any agent P, show that $P = 0$ if and only if P can do no action. Hence write down a process logic proposition Φ such that P satisfies Φ if and only if P can deadlock. [6 marks]

Let

$$C \stackrel{def}{=} g_0.g_1.p_0.p_1.C \qquad D \stackrel{def}{=} g_1.g_0.p_1.p_0.D$$
$$S_0 \stackrel{def}{=} \overline{g}_0.\overline{p}_0.S_0 \qquad S_1 \stackrel{def}{=} \overline{g}_1.\overline{p}_1.S_1$$

For each of the following agents, determine whether or not it can deadlock:

(a) $(C|C|S_0|S_1) \setminus \{g_0, p_0, g_1, p_1\}$

(b) $(C|D|S_0|S_1) \setminus \{g_0, p_0, g_1, p_1\}$ [5 marks]

Prove that $T \approx 0$, where $T \stackrel{def}{=} \tau.T$. Hence, or otherwise, show that it is possible for an agent that can deadlock to be observationally congruent to one that cannot deadlock. [7 marks]

5 Developments in Technology

Write a short essay on the developments in electronic computers during the 1940s and 1950s. Give attention to the emergence of key ideas about machine architecture in this period, such as conditional branching and program interrupt. [20 marks]

6 Pi Calculus

Show that every term of the π-calculus can be converted, using structural congruence, into the form

$$S = (\nu z_1) \cdots (\nu z_k)(M_1 \mid \cdots \mid M_m \mid !Q_1 \mid \cdots \mid !Q_n)$$

where each M_i is a non-empty choice term. Which rules of structural congruence are *not* needed for the conversion? [10 marks]

Such a form is said to be a *simple system* if no M_i or Q_j contains a composition or a replication. A process P is said to *handle* a name x if it contains a free occurrence of x in some output preaction $\overline{y}\langle\vec{u}\rangle$. A system S as above is said to *uniquely handle* x if at most one of the M_i, and none of the Q_j, handles x.

In the monadic π-calculus, assume S is simple and uniquely handles x, and that $S \longrightarrow S'$ where S' is also simple. Give extra conditions on S which ensure that S' in turn uniquely handles x, and also satisfies the extra conditions. Show by example why your extra conditions are needed, and argue informally that they are sufficient. [10 marks]

7 Optimising Compilers

Explain what is meant when a token α is said to be *live* at some point in a flow graph. [4 marks]

Carefully describe an algorithm to compute the live sets from the *gen*, *kill* and *null* sets at each point in a flow graph, and discuss its efficiency in terms of the number of nodes in the flow graph. [6 marks]

Describe an algorithm to calculate the sets of tokens $C_x(n)$ and $J(n)$ where

$$\alpha \in C_x(n) \iff P(n;\alpha) \equiv x\rho + \rho'$$
$$\alpha \in J(n) \iff P(n;\alpha) \equiv \rho + 1$$

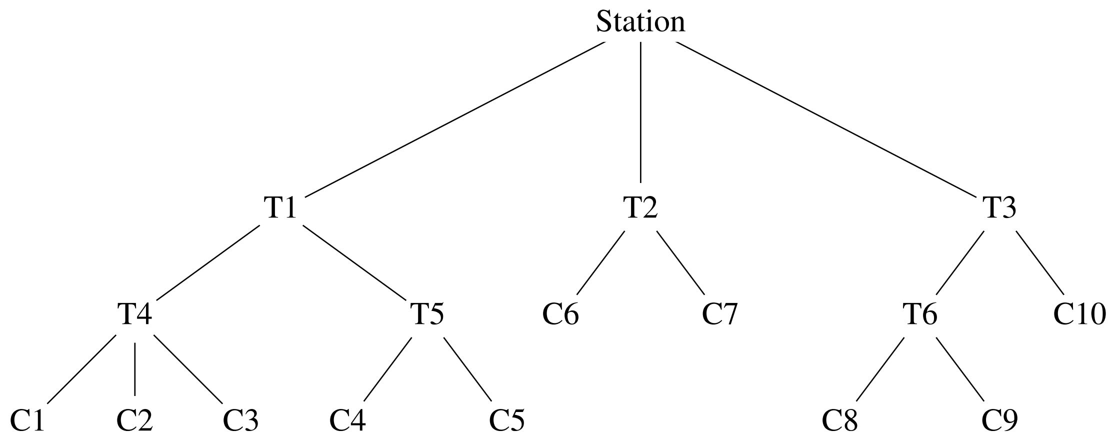
assuming that x, y and z are some permutation of the actions d, r and u (denoting definition, reference and undefinition). [5 marks]

Show how to calculate the set $B_x(n)$ from $C_x(n)$, $C_y(n)$, $C_z(n)$ and $J(n)$ where

$$\alpha \in B_x(n) \iff P(n;\alpha) \equiv x\rho + 1$$ [5 marks]

8 Prolog for Artificial Intelligence

Electricity consumers are supplied with electricity from an electricity generating station. Electricity is distributed from the station to the various consumers through a network of transformers as shown in this diagram:



Nodes C1 to C10 are consumers; nodes T1 to T6 are transformers. Each consumer has a direct connection to only one transformer. Sometimes a transformer may malfunction or need to be taken out of service temporarily.

Devise a data structure in Prolog to represent networks like this. [5 marks]

Write a Prolog procedure to define predicate `supplies` such that goal `supplies(X,Y)` succeeds when there is an electricity supply from node `X` to node `Y` in the network. [10 marks]

Explain how to enhance your representation to permit multiple connections which can be used to make a supply around a transformer which has been taken out of service. [5 marks]

Note: The `supplies` predicate may contain arguments in addition to `X` and `Y` at your discretion.

9 Computational Neuroscience

Neural firing patterns are stochastic: the spike trains emitted by a single neurone contain intrinsic timing randomness resembling a Poisson process, and they create very different response patterns even for apparently identical inputs. Discuss the origins and possible rôles of the stochasticity, including some arguments about why 'noise' may not be a bad thing. Discuss how it might suggest new kinds of computational engine or search strategies. [20 marks]

10 Topics in Artificial Intelligence

Using appropriate mathematical expressions, define the following operations commonly used in computer vision and briefly explain their function and applications:

(*a*) convolution

(*b*) correlation

(*c*) bandpass filtering

(*d*) edge detection by derivative zero-crossings

(*e*) invariant transform

[20 marks]

11 Computer Systems Modelling

Given that in a balanced system with K devices and N customers, the utilisation of each device is given by

$$U = \frac{N}{N + K - 1}$$

derive a formula for the response time in terms of throughput, the number of devices and the average service demand at each device. [6 marks]

A system consists of three types of devices, A, B and C. Customers require service at each type of device but do not care at which particular device they are served. The numbers of each type of device and average service requirements per customer are

	number of devices	average service demand
A	48	48 ms
B	24	24 ms
C	18	18 ms

so that, for example, a customer requires on average 48 ms of service at a type A device.

Give bounds for the system response time at a throughput of 500 customers per second when a scheduling policy ensures that

(a) no device is more than 1.5 times as busy as the average for devices of the same type

(b) no device is more than 1.8 times as busy as the average for devices of the same type

[9 marks]

What can you say about response time if no limit on utilisation skew across devices of the same type is guaranteed? [5 marks]

12 Specification and Verification II

Describe the technique of exhaustive enumeration and discuss its rôle in formal proofs of correctness. [5 marks]

The non-equality of two boolean streams, `a` and `b`, is defined as follows:

$$\texttt{(NotEqual 0 (a,b)} = \texttt{F)} \wedge$$
$$\texttt{(NotEqual(t+1) (a,b)} = \texttt{(}\neg\texttt{(a t} = \texttt{b t)} \rightarrow \texttt{T} \mid \texttt{NotEqual t (a,b)))}$$

Using basic gates and a register with the following behaviour

$$\texttt{REG(in,out)} = \texttt{(}\forall\texttt{t. out t} = \texttt{((t} = \texttt{0)} \rightarrow \texttt{F} \mid \texttt{in(t - 1)))}$$

devise and verify a circuit with inputs `a` and `b` and an output, `out`, which satisfies the following behaviour: $\forall$`t. out t` $=$ `NotEqual t (a,b)` [15 marks]

13 Complexity Theory

Comment on each of the following statements about Computational Complexity. Indicate any places where their wording is not sufficiently precise. For each, decide whether the statement is true, partially true, true but improperly justified, false or just muddled.

(*a*) Given a variant on Quicksort that uses a median-of-three procedure to select pivots, it seems hard to identify exactly what ordering of input data will make the quicksort behave worst. But because there are $N!$ possible different orderings and $N!$ is a bit like 2^N the problem is an NP one.

(*b*) If we could solve the Boolean Satisfiability problem efficiently we could use that to simulate the behaviour of *any* Turing machine and hence solve all other problems efficiently.

(*c*) To keep your secret treasure safe you intend to dig out a series of caves and tunnels forming a maze. You invent a graph for which you know a Hamiltonian circuit (for example, you start by putting in edges to make that circuit and then add lots more to make the graph more complicated). The Hamiltonian circuit problem is known to be NP complete, so given just the graph nobody except you will be able to find the Hamiltonian circuit. You wire up the tunnels so that the treasure can only be reached (safely!) by traversing the Hamiltonian circuit. Being arrogant you then pin details of the graph on your door. NP completeness means that your treasure is almost certainly secure.

You should provide a brief overview of any result, construction or proof that you refer to, but you are not expected to work through the details.

[20 marks]

14 Semantics

The abstract syntax of IMP commands is given by the following grammar:

$$\begin{aligned}Com ::= {} & \texttt{skip} \mid Pvar := Iexp \mid Com\ ;\ Com \mid \\ & \texttt{if}\ Bexp\ \texttt{then}\ Com\ \texttt{else}\ Com \mid \texttt{while}\ Bexp\ \texttt{do}\ Com\end{aligned}$$

where *Iexp* and *Bexp* are syntactic categories of integer and boolean expressions and *Pvar* is a set of program variables. Let *States* be $[Pvar \to \mathbb{Z}]$ and *Cont*, the cpo of *continuations*, be $[States \to A_\perp]$, where A is an unspecified set of program *answers*. A continuation represents what is to be done with the state resulting from the execution of a command in order to return the result of the whole program.

The *continuation semantics* of IMP is defined by giving the meaning $[\![C]\!]$ of each $C \in Com$ as a function which takes a continuation, representing what is to be done when the command has finished, together with a state in which the command is to be executed, and returns an answer:

$$[\![-]\!]\ :\ Com \to (Cont \to (States \to A_\perp)).$$

One clause of the definition of $[\![C]\!]$ is

$$[\![\texttt{skip}]\!]\ k\ S = k(S).$$

Complete the definition of the continuation semantics of IMP commands (expressing their usual behaviour). You may assume that the functions

$$\begin{aligned}[\![-]\!] &: Iexp \to (States \to \mathbb{Z}) \\ [\![-]\!] &: Bexp \to (States \to \mathbb{B}) \quad \text{where } \mathbb{B} = \{true, false\}\end{aligned}$$

have already been defined. [9 marks]

Now add a new command `abort` to *Com* and a new error value *Err* to A. The intended behaviour of `abort` is immediately to terminate the entire program, returning *Err*. Extend the continuation semantics of IMP by giving the definition of $[\![\texttt{abort}]\!]$. [2 marks]

Now add two further new command forms:

$$Com\ ::=\ \ldots \mid \texttt{abort} \mid \texttt{exit} \mid Com\ \texttt{orelse}\ Com$$

The intended behaviour of (C_1 `orelse` C_2) is that it executes exactly like C_1 unless C_1 hits an `exit` command, in which case further execution of C_1 is abandoned and C_2 is executed starting in the state at which C_1 encountered the `exit`. If C_1 does not encounter an `exit` then C_2 is ignored. An `exit` command without an enclosing `orelse` behaves like `abort`.

Give a revised continuation semantics to every command of IMP with `abort`, `exit` and `orelse` which reflects this behaviour and in which the denotation of $C \in Com$ is a function which takes *two* continuations and a state and returns an element of $A_\perp$:

$$[\![-]\!] \; : \; Com \to (Cont \to (Cont \to (States \to A_\perp))).$$

Hint: The first continuation is the ordinary default continuation and the second is the continuation to be applied if the command `exit`s. [9 marks]

15 Numerical Analysis II

A cubic spline $\phi(x)$ is defined over $[a, b]$ with knots x_1, x_2, ... x_n such that $a < x_1$, $x_n < b$. The spline takes the values y_1, y_2, ... y_n at the knots. What continuity conditions are usually imposed on the cubic spline at each knot? [2 marks]

If $d_j = x_{j+1} - x_j$ and $\mu_j = \phi''(x_j)$, the spline has the following formula for $x \in [x_j, x_{j+1}]$

$$\phi(x) = \frac{(x - x_j)y_{j+1} + (x_{j+1} - x)y_j}{d_j} - \frac{(x - x_j)(x_{j+1} - x)\{(d_j + x_{j+1} - x)\mu_j + (d_j + x - x_j)\mu_{j+1}\}}{6d_j}.$$

By differentiating this formula:

(a) find formulae for $\phi'(x_j)$ and $\phi'(x_{j+1})$ for $x \in [x_j, x_{j+1}]$ [4 marks]

(b) verify that $\phi''(x_j) = \mu_j$, $\phi''(x_{j+1}) = \mu_{j+1}$ [4 marks]

(c) deduce the equation which expresses the continuity condition on $\phi'(x)$ at x_j [6 marks]

If the equations derived in part (c) are solved as a simultaneous system, what are the unknowns? If the end conditions specify the spline to be linear in $[a, x_1]$ and $[x_n, b]$ how does this simplify the calculation? State the most important properties of the resulting equations. [4 marks]