COMPUTER SCIENCE TRIPOS Part II

Thursday 5 June 2003 1.30 to 4.30

Paper 9

Answer **five** *questions.*

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.*

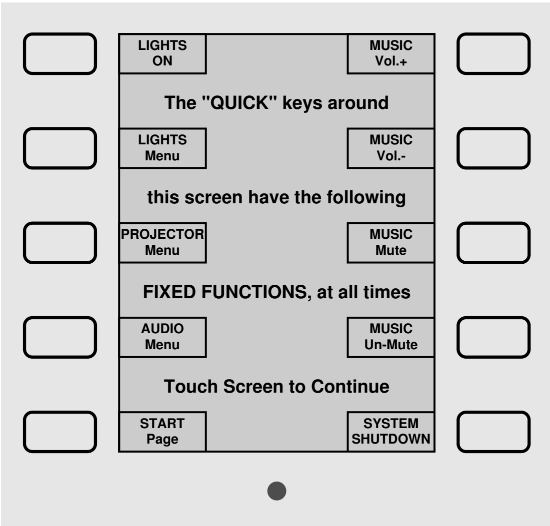
You may not start to read the questions printed on the subsequent pages of this question paper until instructed that you may do so by the Invigilator

1 Human–Computer Interaction

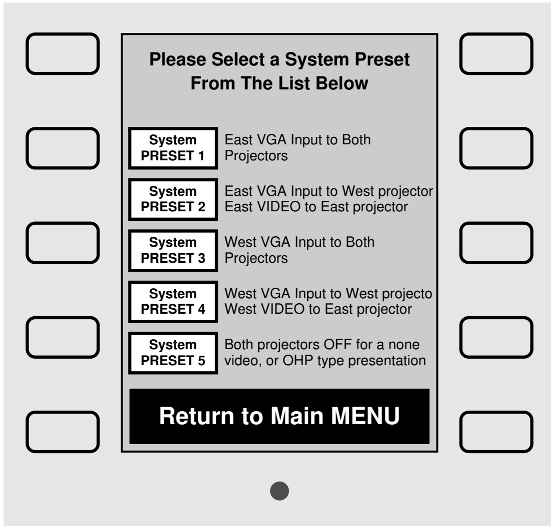
(*a*) A visiting HCI lecturer in Cambridge, when approaching the Gates building, sees the sign “Caution Site Entrance”. What does this illustrate about mental models of the designers and viewers of the sign? [4 marks]

(*b*) On commencing his lecture, he must use the control interface shown below. Describe briefly how to apply *both* a heuristic evaluation *and* cognitive walkthrough to evaluate usability of this interface, identifying from *each* method *two* usability problems. [12 marks]

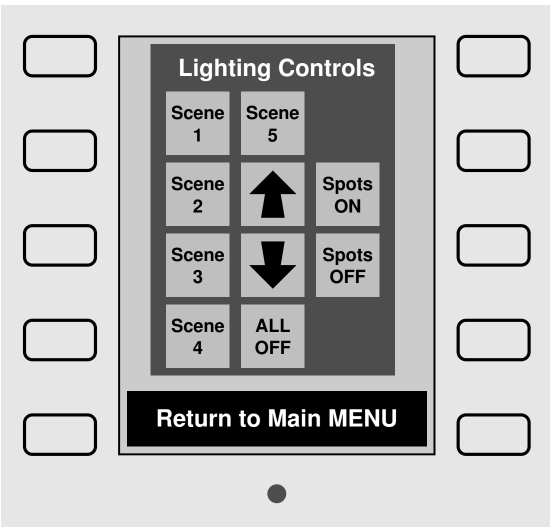
(*c*) Sketch an alternative design for *one* of the screens. Describe a cognitive dimensions trade-off that might occur in this alternative design. [4 marks]

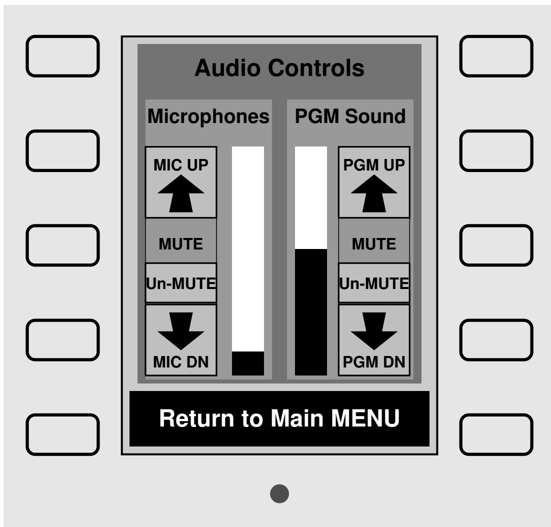


Screen A



Screen B





2 VLSI Design

Explain the operation of ripple carry and carry-skip in a binary adder. [4 marks]

Assuming a delay of τ for each stage of combinational logic (consisting of negation, product and sum), estimate the delays involved in calculating the most significant bit of the sum and the final carry-out of an n-bit adder using

(*a*) ripple carry

(*b*) carry-skip arranged in b blocks of k bits. [4 marks]

What block size minimises the overall delay for a carry-skip adder? [2 marks]

Explain how a variable block size can reduce the overall delay. Compare the delays for a 64-bit adder using fixed and variable block sizes. [4 marks]

An alternative approach is to build a hierarchy of skip units in the style of a carry-lookahead tree. Derive equations for the skip units and estimate the delay in a 64-bit adder arranged as two 4-bit blocks, three groups of four 4-bit blocks (where each group has a higher level skip), and a further two 4-bit blocks. [6 marks]

3 Digital Communication II

In relation to the following headings, compare and contrast the approach taken by BISDN ATM and the Internet Protocol suite:

(*a*) signalling and connection establishment;

(*b*) forwarding/switching;

(*c*) congestion detection and avoidance;

(*d*) Quality of Service;

(*e*) segmentation of large data blocks.

[4 marks each]

4 Distributed Systems

(*a*) Discuss alternative approaches to creating unique names for objects in large-scale distributed systems or applications. [3 marks]

(*b*) Discuss the support for naming and name resolution in the following:

(*i*) the Internet; [5 marks]

(*ii*) object-oriented middleware; [5 marks]

(*iii*) message-oriented middleware; [5 marks]

(*iv*) web-based systems. [2 marks]

5 Advanced Systems Topics

A computer system provides a compare-and-swap operation (CAS) which can be used in the following manner:

```
seen = CAS (address, old, new)
```

It loads the contents of `address`, compares that value against `old` and if it matches stores the value `new` at the same address. All of this is performed atomically and the value loaded from the address is returned as `seen`.

(*a*) Write pseudo-code for a simple spin-lock using `CAS`. [4 marks]

(*b*) Why could this perform poorly on a large multi-processor system? [2 marks]

Consider a singly-linked list of `QNode` objects, each with a boolean field `value` and a reference `next` to its successor (holding `null` at the tail of the queue). A shared location `l` refers to the tail node (or is `null` if the queue is empty).

(*c*) Define the folowing concurrent operations using `CAS`:

```
// Append a new node q to the tail of the list, returning
// the previous tail
QNode pushTail (QNode q);

// Remove q, which must have been at the head of the list,
// returning the new head
QNode popHead (QNode q);
```

[Hint: note that `popHead` only needs to update memory when the queue becomes empty.] [8 marks]

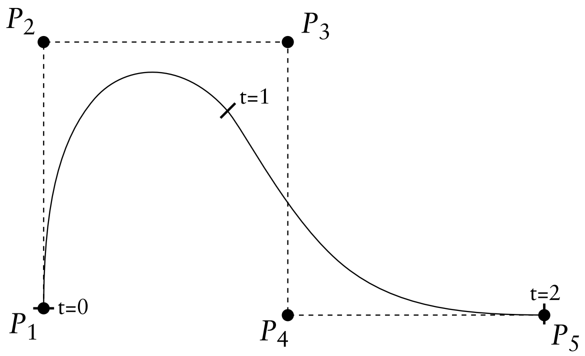
(*d*) Define a *queue-based spin lock* based on these operations. [6 marks]

6 Advanced Graphics

(*a*) (*i*) Derive the quadratic uniform B-spline basis function, $N_{1,3}(t)$, for the knot vector $[1, 2, 3, 4, 5, 6, 7, 8]$. [6 marks]

(*ii*) Explain how $N_{i,3}(t)$ is related to $N_{1,3}(t)$, $i \in \{2, 3, 4, 5\}$. [2 marks]

(*b*) The following picture shows a set of five control points and the B-spline curve generated by the control points and the knot vector $[0, 0, 0, 0, 1, 2, 2, 2, 2]$ with $k = 4$ (a cubic B-spline).

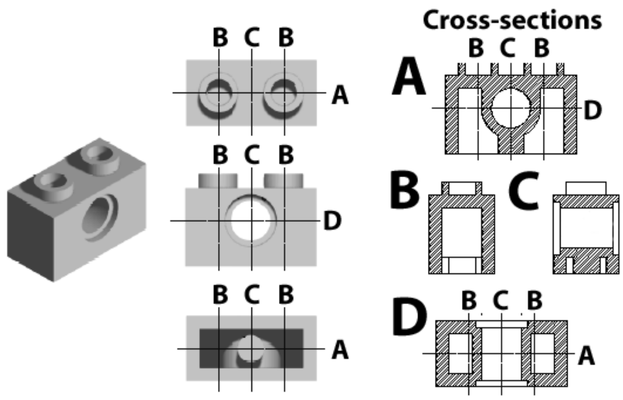


(*i*) Draw a similar diagram, using the same five control points, for the knot vector from part (*a*), $[1, 2, 3, 4, 5, 6, 7, 8]$, defining a quadratic B-spline ($k = 3$). [3 marks]

(*ii*) Draw another diagram, with the same control points, for the knot vector $[1, 2, 3, 4, 4, 5, 6, 7]$, defining a quadratic B-spline ($k = 3$). [3 marks]

(*iii*) What is the continuity of the curve at $t = 4$ in each of the cases in parts (*b*)(*i*) and (*b*)(*ii*)? [2 marks]

(*c*) Show how the following object can be constructed using Constructive Solid Geometry (CSG). You may assume the following primitives: sphere, cylinder, cone, torus, box. [You are expected to describe which primitives are needed and how they are combined but you are not expected to specify accurately all of the parameters of the primitives.]



[4 marks]

7 Optimising Compilers

(*a*) Summarise briefly the principles of *strictness* analysis, including descriptions of:

(*i*) the space of values used for analysis-time representation of a k-argument, 1-result function in the source language;

(*ii*) how a built-in function is given an abstract meaning;

(*iii*) how a recursive user-defined function is given an abstract meaning (it is acceptable to do this part by example);

(*iv*) the machine-level benefit of the associated optimisation.

[8 marks]

(*b*) A problem amenable to similar treatment is that of *escape analysis*. Here we have a call-by-value language with `cons` and the question to be answered is "whether a value containing a cons-node passed as argument to a function *may* be returned ('escape') as part of the function's result".

(*i*) Choose (and state clearly) an appropriate set of abstract values and abstractions of functions to formalise the problem of escape analysis for a simple first-order language with integers and simple integer lists (but not lists of lists). Also give abstract interpretations of `if-then-else`, `+`, `cons`, `hd` and `tl`.
[Hint: to manage this system without using static types, you might best assume that *nil* is treated as 0, and that any type-error (dynamically detected) such as `cons(1,nil)+3`, `tl(3)` and even (because of the 'no-lists-of-lists' rule) `cons(cons(1,nil),nil)` gives result 0.] [8 marks]

(*ii*) Give without proof abstract meanings (resulting from your system) of the following functions:

```
f(x,y,z) = cons(hd(tl(x)), if hd(x) then y else tl(z))
g(x,y) = if x=0 then 0 else cons(hd x, g(tl x, y))
h(x,y) = if x=0 then x else cons(hd x, h(tl x, y))
k(x,y) = if x=0 then y else cons(hd x, k(tl x, y))
```

[4 marks]

8 Artificial Intelligence II

(a) Explain what the terms *ontological commitment* and *epistemological commitment* mean in the context of a language for knowledge representation and reasoning. What are the ontological and epistemological commitments made by propositional logic and by first order logic? [4 marks]

(b) You wish to construct a robotic pet cat for the purposes of entertainment. One purpose of the cat is to scratch valuable objects when the owner is not present. Give a brief general description of *situation calculus* and describe how it might be used for knowledge representation by the robot. Include in your answer one example each of a *frame axiom*, an *effect axiom*, and a *successor-state axiom*, along with example definitions of suitable predicates and functions.
[12 marks]

(c) Give a brief description of the *representational frame problem*, the *inferential frame problem*, the *qualification problem*, and the *ramification problem*.
[4 marks]

9 Database Theory

Assume a database with one relation `parent` consisting of pairs (a, b) where a is a parent of b.

(a) Write a Datalog query which gives the set of pairs (x, y) such that x and y have a common ancestor z and are the same number of generations from z.
[4 marks]

(b) Write a query in Datalog *with stratified negation* which gives the set of pairs (x, y) such that x and y have a common ancestor but not one from which they are the same number of generations distant. You may use the program you defined for part (a). [5 marks]

(c) Prove that the query defined in part (b) above cannot be expressed in Datalog without negation. [7 marks]

(d) For each of queries in parts (a) and (b), give a bound on the running time to evaluate the query on a database with n entries. [4 marks]

10 Types

(*a*) Describe the relation $=_\beta$ of *beta-conversion* between terms of the polymorphic lambda calculus (PLC). How can one decide whether two typeable PLC terms are in this relation? Why does the decision procedure fail for untypeable terms? [8 marks]

(*b*) Let ω be the polymorphic type $\forall \alpha_1((\forall \alpha_2(\alpha_2 \to \alpha_1)) \to \alpha_1)$. Show that there is a closed PLC term I with the following two properties.

(*i*) I has type $\forall \alpha(\alpha \to \omega)$.

(*ii*) If M_1 and M_2 are any closed PLC terms of the same type, τ say, and if $(I\,\tau\,M_1) =_\beta (I\,\tau\,M_2)$, then $M_1 =_\beta M_2$.

[Hint: for property (*ii*), consider the beta-normal forms of $I\,\tau\,M_1\,\alpha\,x$ and $I\,\tau\,M_2\,\alpha\,x$, where α is a type variable and x is a variable.] [12 marks]

11 Numerical Analysis II

(*a*) Explain the term *positive semi-definite matrix*. [1 mark]

(*b*) Let $\mathbf{A}$ and $\mathbf{B}$ be $n \times n$ matrices and let $\mathbf{x}$ be a vector of n elements. State *Schwarz's inequality* for each of the products $\mathbf{AB}$ and $\mathbf{Ax}$. What are the *singular values* of $\mathbf{A}$, and how are they related to the ℓ_2 norm of $\mathbf{A}$? [4 marks]

(*c*) Describe briefly the *singular value decomposition* $\mathbf{A} = \mathbf{UWV}^T$, and how it may be used to solve the linear equations $\mathbf{Ax} = \mathbf{b}$. [4 marks]

(*d*) Let $\hat{\mathbf{x}}$ be an approximate solution of $\mathbf{Ax} = \mathbf{b}$ and write $\mathbf{r} = \mathbf{b} - \mathbf{A}\hat{\mathbf{x}}$, $\mathbf{e} = \mathbf{x} - \hat{\mathbf{x}}$. Find an expression for an upper bound on the relative error $\|\mathbf{e}\| \,/\, \|\mathbf{x}\|$ in terms of computable quantities. Show how this formula may be computed using the singular values of $\mathbf{A}$. [8 marks]

(*e*) Suppose $\mathbf{A}$ is a 5×5 matrix and its singular values are 10^3, 1, 10^{-14}, 10^{-18}, 10^{-30}. If *machine epsilon* $\simeq 10^{-15}$ then choose a suitable *rank* for an approximate solution and form the generalised inverse $\mathbf{W}^+$. [3 marks]

12 Specification and Verification II

The following is an informal specification of a device `D`

> `D` *has a 1-bit input* `in`*, a 1-bit input* `select` *and a 4-bit output* `out`*. The value at* `out` *is* `0000` *if* `select` *is* `0` *and is a word consisting of the values input at* `in` *at the four preceding cycles if* `select` *is* `1`*.*

(*a*) Formalise this informal specification and point out how you have resolved any ambiguities and incompletenesses. [8 marks]

(*b*) Using 1-bit unit-delay elements and multiplexers, design a device that implements your formal specification. Draw a diagram of your design. [8 marks]

(*c*) Outline how you would go about trying to formally verify your design. [4 marks]

13 Natural Language Processing

(*a*) Define the following terms, as they are used in grammar formalisms for natural language:

(*i*) feature structure

(*ii*) feature structure subsumption

(*iii*) feature structure unification

[8 marks]

(*b*) Discuss the advantages and disadvantages for NLP applications of grammar formalisms that use feature structures compared with context free grammars. [12 marks]

14 Denotational Semantics

(*a*) State without details *three* ways to prove properties of the least fixed point $\mathit{fix}(f)$ of a continuous function f on a domain. [3 marks]

(*b*) Let $f : D \to D$ be a continuous function on a domain D. Explain why $\mathit{fix}(f \circ f) = \mathit{fix}(f)$. [3 marks]

(*c*) Let both $f : D \to D$ and $g : D \to D$ be continuous functions on a domain D. Prove

$$\mathit{fix}(f \circ g) = f(\mathit{fix}(g \circ f))$$

by showing

(*i*) $\mathit{fix}(f \circ g) \sqsubseteq f(\mathit{fix}(g \circ f))$, and [4 marks]

(*ii*) $f(\mathit{fix}(g \circ f)) \sqsubseteq \mathit{fix}(f \circ g)$. [10 marks]

[Hint: For the last part, start by expanding the left-hand side as a least upper bound of approximations.]

15 Topics in Concurrency

This question assumes familiarity with the higher-order process language HOPLA, which has prefix types, function types, sum types and recursive types. Subject to suitable typings, HOPLA has transitions $t \xrightarrow{p} t'$ between closed terms t, t' and action p given by the following rules:

$$\frac{t[\mathit{rec}\, x\, t/x] \xrightarrow{p} t'}{\mathit{rec}\, x\, t \xrightarrow{p} t'} \qquad \frac{t_j \xrightarrow{p} t'}{\Sigma_{i \in I}\, t_i \xrightarrow{p} t'}\ j \in I$$

$$\frac{}{.t \xrightarrow{\bullet} t} \qquad \frac{u \xrightarrow{\bullet} u' \quad t[u'/x] \xrightarrow{p} t'}{[u > .x \Rightarrow t] \xrightarrow{p} t'}$$

$$\frac{t[u/x] \xrightarrow{p} t'}{\lambda x\, t \xrightarrow{u \mapsto p} t'} \qquad \frac{t \xrightarrow{u \mapsto p} t'}{t\ u \xrightarrow{p} t'} \qquad \frac{t \xrightarrow{p} t'}{a\ t \xrightarrow{a\, p} t'} \qquad \frac{t \xrightarrow{a\, p} t'}{\pi_a(t) \xrightarrow{p} t'}$$

(a) Write down the typing rule for $[u > .x \Rightarrow t]$. [2 marks]

(b) Let t be a term of type $\mathbb{P}$ with one free variable x of type $\mathbb{Q}$. Say t is *linear in* x iff for any sum of closed terms $\Sigma_{i \in I} u_i$ of type $\mathbb{Q}$

$$t[\Sigma_{i \in I} u_i/x] \ \sim_{\mathbb{P}}\ \Sigma_{i \in I} t[u_i/x].$$

(The relation $\sim_{\mathbb{P}}$ is that of bisimilarity between terms of type $\mathbb{P}$.)

Show from the transition semantics that the following terms, assumed well-typed and to have only x as free variable, are all linear in x:

$$\pi_a(x) \qquad x\ u \qquad [x > .y \Rightarrow u] \qquad (x \text{ is not free in } u).$$

Show that the prefix term $.x$ is not linear in x. (Here you may assume that x has any type that is convenient.) [7 marks]

(c) For u of sum type, let $[u > a.x \Rightarrow t]$ abbreviate $[\pi_a(u) > .x \Rightarrow t]$. Derive a rule for the transitions of $[u > a.x \Rightarrow t]$. [2 marks]

(d) Describe the type you would use to interpret CCS in HOPLA. Write down a HOPLA term that realises the parallel composition of CCS. What is its type? [5 marks]

(e) Write down the HOPLA types you would use to interpret variants of CCS (i) with value passing, and (ii) with process passing. [4 marks]

END OF PAPER