COMPUTER SCIENCE TRIPOS Part IB

Wednesday 6 June 2007 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. 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

STATIONERY REQUIREMENTS
Script paper
Blue cover sheets
Tags

SPECIAL REQUIREMENTS
None

SECTION A

1 Software Engineering

An inertial navigation system is a device that uses gyros and accelerometers to measure changes in its position and orientation. It is typically used in aircraft to provide a backup navigation mechanism for GPS and other radio-based systems, and to provide an altitude reference when flying on autopilot.

You have been hired to adapt such a device to be the main navigation system for an unmanned submersible. Describe the methodology you would use, with particular reference to how you would assure your customer of the device's dependability, and what sort of problems you would watch out for.

[20 marks]

2 Computer Design

(*a*) What is the difference between the *latency* and the *bandwidth* of a communication link? [2 marks]

(*b*) Why are control-flow processors sensitive to memory latency even if the memory bandwidth exceeds what the processor could ever require? [4 marks]

(*c*) How is average memory latency typically reduced? [5 marks]

(*d*) What is the difference between serial and parallel communication and what is the receiver required to do to recover the data in each case? [5 marks]

(*e*) The PCI communication standard for add-on cards for PCs has moved to PCI express (PCIe). PCI uses a parallel bus to communicate to several cards whereas PCIe uses sets of point-to-point serial links for communication. Why are multiple serial links rather than parallel buses preferred for high bandwidth communication? [4 marks]

3 Digital Communication I

Using *four* examples, explain how multiple higher-layer channels can be multiplexed onto a lower-layer channel. In each example consider

(*i*) how the individual higher-layer channels can be recognised;

(*ii*) what the mechanism is for allocation of lower-layer channel resources to the higher-layer channels; and

(*iii*) the characteristics of the higher-layer channels.

[4 × 5 marks]

4 Concurrent Systems and Applications

(*a*) A software engineer is developing computer-aided design software for an architect who works with designs for three different types of building: Contemporary, Victorian, and Roman. When laying out buildings, the architect places windows, doorways, walls and roofs in the appropriate style for the building on which she is working. Explain, including all the Java interfaces and code fragments, how the software can use the Abstract Factory design pattern to construct and maintain its internal data structures representing a building. [6 marks]

(*b*) In terms of software testing, what is *equivalence partitioning*, how can it be done, and how can the partitioned sets be used to select test patterns? [6 marks]

(*c*) What are *decision coverage* and *condition coverage*? [2 marks]

(*d*) Describe Java's Remote Method Invocation (RMI) system, including the means by which objects and servers are named, how clients perform binding, and the purpose of the RMI CodeBase. [6 marks]

SECTION B

5 Computer Graphics and Image Processing

(*a*) Describe, in detail, an algorithm that will draw a one-pixel wide outline of a circle of integer radius centred on the origin. [10 marks]

(*b*) Describe the modifications required to your algorithm in part (*a*) to make it draw a filled circle. [3 marks]

(*c*) Describe the modifications required to your algorithm in part (*a*) to make it draw the outline of a circle centred at arbitrary integer coordinates. [2 marks]

(*d*) Describe the modifications required to your algorithm in part (*c*) to make it draw the outline of a circle centred at arbitrary non-integer coordinates and of non-integer radius. [5 marks]

6 Compiler Construction

(*a*) Describe how a stack is used to implement procedures and functions. [6 marks]

(*b*) Suppose a language allows the creation of pointers. How does this complicate the use of stacks as described in part (*a*)? [2 marks]

(*c*) How does the Java language deal with the problem described in part (*b*)? [2 marks]

(*d*) Consider the following ML-like program containing the function `g` that returns a function as a result.

```
let a = 17 in
let g b = (let h c = a + b + c in h) in
let f1 = g 21 in
let f2 = g 33 in
let v = f1(3) + f2(57) in
...
...
```

Explain carefully how such a program can be compiled. In particular, pay special attention to how the code for the body of the function `h` can access the values of `a`, `b`, and `c`. [10 marks]

7 Concepts in Programming Languages

(*a*) Give an overview of the FORTRAN execution model (or abstract machine) and comment on its merits and drawbacks from the viewpoints of programming, compilation, execution, etc. [5 marks]

(*b*) What is *garbage collection* in the context of programming languages? Comment on its merits/drawbacks and explain the contexts in which it arises, giving examples. [5 marks]

(*c*) Recall that Algol has a primitive static type system. In particular, in Algol 60, the type of a procedure parameter does not include the types of its parameters. Thus, for instance, in the Algol 60 code

```
procedure P( procedure F )
    begin F(true) end ;
```

the types of the parameters of the procedure parameter `F` are not specified.

Explain why this piece of code is statically type correct. Explain also why a call `P(Q)` may produce a run-time type error, and exemplify your answer by exhibiting a declaration for `Q` with this effect.

Why does this problem not arise in SML? [5 marks]

(*d*) Give an implementation of the abstract data type of stacks using the SML module system. [5 marks]

8 Databases

(*a*) The Entity/Relationship model is based around the concepts of *entity*, *attribute*, and *relationship*. Describe how these can be represented in the relational model. [6 marks]

(*b*) Data normalisation is often an important component in database design. Discuss why this is so, and give examples of situations where normalisation is *not* important. [6 marks]

(*c*) Let A and B be disjoint non-empty sets of attributes. Let R be a relation over attributes $A \cup B$ and let S be a relation over attributes B.

Suppose that we want to introduce a new relational operation called *division*, denoted $R \div S$, that will return a relation over attributes A. The relation $R \div S$ is made up of all tuples t such that for all $s \in S$ we have $ts \in R$ (ts is the concatenation of t and s).

Note that in the special case that $R = T \times S$ for some relation T, then $(R \div S) = T$ and $(R \div T) = S$.

In other words, $\div$ can be treated as an inverse to the Cartesian product.

Can we define $R \div S$ in the relational algebra? Prove that your answer is correct. [8 marks]

SECTION C

9 Logic and Proof

(*a*) You have to write a program that accepts a propositional formula ϕ and returns an interpretation that falsifies ϕ. Describe the algorithm you would employ in *each* of the following circumstances:

(*i*) ϕ can be expected to contain many occurrences of the $\leftrightarrow$ connective. [4 marks]

(*ii*) ϕ will be delivered in disjunctive normal form. [4 marks]

(*b*) Let S be a set of propositional clauses, each of which contains a negative literal.

(*i*) Show that applying the resolution procedure to S will never generate the empty clause. [3 marks]

(*ii*) Describe a model that satisfies S. [3 marks]

(*c*) Prove or disprove the sequent $\Box\Diamond(P \lor Q) \Rightarrow (\Box\Diamond P) \lor (\Box\Diamond Q)$ of S4 modal logic. [6 marks]

10 Foundations of Functional Programming

(*a*) Writing as usual $\rightarrow$ for 1-step reduction (i.e. β-η-reduction with α-conversion only used to avoid name clashes) and $\twoheadrightarrow$ for its reflexive-transitive closure, indicate giving reasons (you may merely claim well-known results) whether the following statements are *always*, *never* or *sometimes* true of pure λ-terms L, M and N.

(*i*) if $M = N$ then $M \twoheadrightarrow N$ or $N \twoheadrightarrow M$.

(*ii*) if $M \rightarrow M$ then M is in normal form.

(*iii*) if $L \rightarrow M$ and $L \rightarrow N$ then there exists L' such that $M \rightarrow L'$ and $N \rightarrow L'$

(*iv*) if $L \twoheadrightarrow M$ and $L \twoheadrightarrow N$ then there exists L' such that $M \twoheadrightarrow L'$ and $N \twoheadrightarrow L'$

[2 marks each]

(*b*) Define λ-terms **if**, **true** and **false** that satisfy that **if true** $M\ N = M$ and **if false** $M\ N = N$. [2 marks]

(*c*) Given your definitions in part (*b*) above, indicate giving reasons whether it is *always*, *never* or *sometimes* true that:

(*i*) **if true** $M\ N \twoheadrightarrow_e M$ where $\twoheadrightarrow_e$ represents eager evaluation

(*ii*) **if true** $M\ N \twoheadrightarrow_\ell M$ where $\twoheadrightarrow_\ell$ represents lazy evaluation

[3 marks each]

(*d*) Explain why the β-reduction rule tends not to be used literally for implementing functional programming languages, indicating *two* alternatives. [4 marks]

11 Semantics of Programming Languages

(*a*) Consider the types given by the grammar below.

$$T \ ::= \ \mathsf{unit} \mid T_1 \rightarrow T_2$$

Define the syntax and type system for a pure functional language over these types: a syntax of expressions e for variables, skip, functions, and function application, and typing rules defining a judgement $\Gamma \vdash e{:}T$. State clearly what mathematical objects Γ ranges over, and what the binding is in your language. [5 marks]

(*b*) For each of the following, state whether it is true or false. For the true statements, give examples (instantiations for the existentially quantified variables); for the false statements, give proofs of their negations. For any inductive proofs, include statements of the kind of induction used and the induction hypothesis.

(*i*) $\exists\, \Gamma_1, \Gamma_2, e, T_1, T_2.\ (\Gamma_1 \vdash e{:}T_1) \wedge (\Gamma_2 \vdash e{:}T_2) \wedge (T_1 \neq T_2)$

(*ii*) $\exists\, \Gamma, e, T_1, T_2.\ (\Gamma \vdash e{:}T_1) \wedge (\Gamma \vdash e{:}T_2) \wedge (T_1 \neq T_2)$

(*iii*) $\exists\, \Gamma, e, T.\ \Gamma \vdash e\ e{:}T$

(*iv*) $\exists\, \Gamma, e, T.\ \Gamma \vdash e{:}T$ such that $\Gamma \not\vdash e{:}T$ if the syntax and rules were interpreted concretely, instead of up to alpha equivalence.

[14 marks]

(*c*) Discuss briefly whether alpha equivalence is needed to define type systems for ML-like and Java-like languages. [1 mark]

12 Complexity Theory

(*a*) Give a precise definition of polynomial-time reductions. [2 marks]

(*b*) Give a precise definition of NP-completeness. [3 marks]

(*c*) Let **Subset Sum** denote the following decision problem:

Given a set of positive integers $S = \{v_1, \ldots, v_n\}$ and a number t, determine whether there is a subset of S that sums to exactly t.

(*i*) Explain why **Subset Sum** is in NP. [3 marks]

(*ii*) Describe a polynomial-time reduction from the problem of 3-dimensional matching to **Subset Sum**. [9 marks]

(*iii*) Explain why parts (*i*) and (*ii*) above imply that **Subset Sum** is NP-complete. [3 marks]

END OF PAPER