COMPUTER SCIENCE TRIPOS Part IB

Wednesday 5 June 2002 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

SECTION A

1 Data Structures and Algorithms

You have available a 20 Gbyte disc on which you need to hold an indexed sequential file consisting of variable length records each having a 20 byte key. Records, including the key, are typically 500 bytes long but never exceed 1000 bytes. The total size of all the records is never more than 10 Gbytes.

(*a*) Suggest, in detail, how you would represent this file on the disc. You should choose an organisation that allows

(*i*) efficient insertion of new records,

(*ii*) efficient updating of existing records identified by key, and

(*iii*) efficient inspection of all records in key order.

[14 marks]

(*b*) If the total size of the database is 10 Gbytes, estimate, for your organisation of the file, how many disc transfers would be needed to access a record with a given key, and estimate how many transfers would be required to read the entire database in sequential order. [6 marks]

2 Computer Design

Modern dynamic random access memories (e.g. DRAM, SDRAM and RAMBUS) all support burst mode read and write access.

(*a*) Give an outline of the bus activity for a burst mode read access. [4 marks]

(*b*) Explain the difference between a direct mapped cache and an associative cache. [4 marks]

(*c*) What cache line replacement policies are typically used for a direct mapped cache and a set associative cache? [4 marks]

(*d*) Why are caches able to exploit burst mode accesses, and why is a write buffer often used? [4 marks]

(*e*) What is *bus snooping* and what does it achieve? [4 marks]

3 Digital Communication I

Consider the real-time transport of audio across a network.

(*a*) What are the advantages of digitising the audio? [5 marks]

(*b*) What are the disadvantages and how can they be mitigated? [5 marks]

(*c*) What characteristics of the end-to-end channel across the network would be desirable, and how are these different from those which would be desirable for time-insensitive data? [5 marks]

(*d*) Discuss the applicability of asynchronous and synchronous multiplexing in carrying real-time audio traffic. [5 marks]

4 Concurrent Systems and Applications

(*a*) The `suspend()` and `resume()` methods defined on `java.lang.Thread` can be used to block and unblock a designated thread. Explain why these methods should not be used. [4 marks]

(*b*) Define a Java class `Barrier` that pairs together two different kinds of thread (`A` and `B`) in a concurrent system. It should support two methods, `enterA(Object o)` and `enterB(Object o)`. A call to one `enter` method blocks until a call is made to the other. At that point both invocations continue, `enterA` returning the value passed to `enterB` and *vice versa*.

[8 marks]

(*c*) A number of barbers and customers have to co-ordinate their actions in a barbers' shop. Each barber has a chair in which a customer sits while receiving a haircut and in which that barber dozes when he has no customer. There is plenty of waiting space.

Define Java classes `Barber`, `Customer` and `Shop` to model this situation. The class `Customer` should support two methods:

(*i*) `Barber getHaircut(Shop s)` – request service at `s`, blocking until the customer is served and returning the allocated barber;

(*ii*) `void leaveChair(Barber b)` – signal that `b` can take another customer.

The class `Barber` should support two corresponding methods:

(*i*) `Customer getCustomer(Shop s)` – wait to be allocated a customer at `s`;

(*ii*) `void finishedCustomer(Customer c)` – signal to `c` that the haircut is finished and wait for `c` to leave the chair.

[8 marks]

SECTION B

5 Comparative Programming Languages

This question concerns the representation of parse tree nodes for expressions composed of integer constants, identifiers, and integer operators for addition, subtraction, multiplication and division. In a typeless language, such as BCPL, each node can be implemented as a vector whose first element holds an integer identifying the node operator. The size of the vector and the kinds of value held in the other elements then depends on this node operator.

(*a*) Complete the description of how you would represent such integer expressions in a typeless language. [5 marks]

(*b*) Suggest how you would represent such integer expressions in C and *either* ML *or* Java. [10 marks]

(*c*) Briefly discuss the relative merits of your C data structure compared with that used in the typeless approach. [5 marks]

6 Compiler Construction

(*a*) Assuming a Java type is given to each variable, state a method by which an overloaded operator (such as `+`,`-` etc.) in a Java program can be determined to be an `int`, `real` or other operator. [3 marks]

(*b*) Explain, using pseudo-code as appropriate, how to convert a syntax-tree into stack code such as that used in the JVM. For the purposes of this question, you only need consider trees representing bodies of `void`-returning functions, and these bodies only as consisting of a list of statements of the form `int` x `=` e`;` or $x = e$; where x ranges over variables and e over expressions; expressions contain variables, integer constants, the binary operator `+` and static method invocations. [10 marks]

(*c*) Show how a sequence of simple stack code instructions, such as those used in your answer to part (*b*) above, can be translated into a sequence of instructions for a register-oriented architecture of your choice, for example ARM or Pentium. [7 marks]

7 Prolog for Artificial Intelligence

(*a*) Give a simple definition of the Prolog predicate `dfx` that can perform symbolic differentiation with respect to the variable `x` of expressions composed of integers (e.g. `0`, `1`, ...), symbolic constants (e.g. `a`, `b`, ...), symbolic variables (e.g. `x`, `y`, ...) and the operators `+`, `-` and `*`, for addition, subtraction and multiplication. The first argument of `dfx` is the expression to differentiate and the second argument is the result. Your definition need not perform any simplification of the result. [6 marks]

(*b*) Trace the execution of the call: `dfx(x*x-2, R)`. [2 marks]

(*c*) Now modify your definition so that it simplifies the result by the applications of rewriting rules such as: `1*x`⇒`x` and `x-0`⇒`x`. [8 marks]

(*d*) Discuss to what extent, if any, either of your predicates could be used to integrate an expression. [4 marks]

8 Databases

(*a*) Describe the relational model of data. [6 marks]

(*b*) Explain the following concepts in relational databases:

(*i*) entity integrity constraint; [1 mark]

(*ii*) foreign key and how it can specify a referential integrity constraint between two relations; [4 marks]

(*iii*) semantic integrity constraint. [1 mark]

(*c*) (*i*) What is a functional dependency? [1 mark]

(*ii*) Define Boyce-Codd Normal Form (BCNF). [3 marks]

(*iii*) Define Third Normal Form (3NF). [3 marks]

(*iv*) What is the relationship between BCNF and 3NF? [1 mark]

SECTION C

9 Semantics of Programming Languages

(*a*) The integer expressions `e` of a C-like language take the form `e` ::= `n` | `x` | `x++` | `++x` | `e + e`, where `n` ranges over integer constants and `x` over integer storage variables. The expression `x++` returns the value stored in the integer variable `x` and then increments the stored value by one; whereas `++x` first increments the stored value by one and then returns it. Assuming a left-to-right evaluation order, give an operational semantics for all these expressions, in the form of an evaluation relation $\langle s, \mathtt{e}\rangle \Downarrow \langle s', \mathtt{n}\rangle$, where s, s' range over states which are finite functions from integer storage variables to integers. [5 marks]

(*b*) The commands (statements) `c` of this same language take the form `c` ::= `x = e` | `x += e` | `c;c`. The first form is assignment and the last is sequencing; the command `x += e` evaluates `e`, adds the result to the value stored in `x` and stores the result there. Give an operational semantics for these commands in the form of an evaluation relation $\langle s, \mathtt{c}\rangle \Downarrow s'$ (where s, s' are as above). [4 marks]

(*c*) Define the notion of *semantic equivalence* for these expressions and commands. [3 marks]

(*d*) For each of the following pairs of expressions or commands, state, with justification, whether or not they are semantically equivalent.

(*i*) `++x` and `x++ + 1` [2 marks]

(*ii*) `x = ++x` and `x = x++` [2 marks]

(*iii*) `x = ++x` and `x += 1` [2 marks]

(*iv*) `x += e` and `x = x + e`, for any `e` [2 marks]

10 Foundations of Functional Programming

(*a*) Explain how a lambda-term can be converted into a form that uses only the combinators S and K. [4 marks]

(*b*) Illustrate your method by writing down a lambda term for each of the following functions and then expressing it in terms of just S and K.

(*i*) fun I x = x

(*ii*) fun B f g x = f (g x)

(*iii*) fun C f x y = f y x

(*iv*) fun A x y = y (x x y)

[4 marks each]

11 Logic and Proof

(*a*) For each of the following formulae, state (with justification) whether it is satisfiable, valid or neither:

$$((Q \to R) \to Q) \land \neg Q$$ [2 marks]

$$((P \leftrightarrow Q) \leftrightarrow P) \leftrightarrow Q$$ [2 marks]

$$\exists xy\,[P(x,y) \to \forall xy\,P(x,y)]$$ [3 marks]

$$[\forall x\,(P(x) \to Q(x)) \land \exists x\,P(x)] \to \forall x\,Q(x)$$ [3 marks]

(*b*) Briefly outline the semantics of first-order logic, taking as an example the formula $\forall xy\,f(x,y) = f(y,x)$. [6 marks]

(*c*) Exhibit a model that satisfies both of the following formulae (a is a constant):

$$\forall x\,g(x) \neq a$$

$$\forall xy\,[g(x) = g(y) \to x = y]$$

[4 marks]

12 Complexity Theory

(*a*) Give a precise definition of what is meant by the statement that a decision problem A is *polynomial-time reducible* to a decision problem B. [2 marks]

(*b*) Consider the following three decision problems on graphs.

Connect—the collection of graphs G such that there is a path between any two vertices of G.

Hamilton—the collection of graphs that contain a Hamiltonian cycle.

non-3-colour—the collection of graphs that are not 3-colourable.

For each of the following statements, state whether it is true, false or an unresolved open question. Give a brief justification for your answer.

(*i*) **Connect** is decidable by a polynomial time algorithm.

(*ii*) **Hamilton** is decidable by a polynomial time algorithm.

(*iii*) **non-3-colour** is decidable by a polynomial time algorithm.

(*iv*) **Connect** is polynomial-time reducible to **Hamilton**.

(*v*) **Hamilton** is polynomial-time reducible to **non-3-colour**.

(*vi*) **non-3-colour** is polynomial-time reducible to **Connect**.

[3 marks each]

END OF PAPER