COMPUTER SCIENCE TRIPOS Part II (General)
DIPLOMA IN COMPUTER SCIENCE

Wednesday 4 June 2003 1.30 to 4.30

Paper 12 (Paper 3 of Diploma in Computer Science)

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 Data Structures and Algorithms

(*a*) A million singleton sets each containing a distinct integer are to be successively combined by calls of `union(`S_1`, `S_2`)`. The result represents the union of the two disjoint sets represented by S_1 and S_2. Interspersed among these calls are several calls of `inSameSet` where `inSameSet(`m`, `n`)` yields `true` if and only if m and n are integers now in the same set. Describe in detail how you would implement `union` and `inSameSet` assuming they will be called about one million and five million times, respectively. Explain why your solution is efficient. [10 marks]

(*b*) Describe in detail an implementation of Kruskal's algorithm for finding a minimum cost spanning tree of an undirected graph with positive integer costs on the edges that uses your version of `union` and `inSameSet`. [5 marks]

(*c*) Explain why the spanning tree is unique if all the edge costs are distinct. [5 marks]

2 Computer Design

(*a*) Name and describe *three* reasons for a cache miss. [6 marks]

(*b*) For *each* reason, suggest a technique for reducing the number of misses. [6 marks]

(*c*) Why might it be advantageous to use a set-associative cache instead of a fully associative one? [4 marks]

(*d*) Describe *two* techniques for reducing the miss penalty. [4 marks]

3 Digital Communication I

(*a*) Define the terms *capacity* and *latency* as applied to a communications channel. [4 marks]

(*b*) How can variable latency cause problems? You may wish to consider

(*i*) XON/XOFF flow control;

(*ii*) streaming media;

(*iii*) protocol timeouts. [6 marks]

(*c*) Describe the operation of a simple ARQ protocol with a window of a single packet. [4 marks]

(*d*) A simple ARQ scheme is used to provide reliable transport over a link where 80% of packets other than short acknowledgements experience a 1 ms delay, 10% experience a 10 ms delay, and 10% are lost. Acknowledgements always experience a 1 ms delay and are never lost. What would be the expected throughput in packets/sec if the timeout was

(*i*) 10 ms?

(*ii*) 12 ms?

Assume that the transmitter always has information to send and that transmission time is negligible.

It may be helpful to note that

$$\sum_{i=0}^{\infty} i\,x^i = \frac{x}{(1-x)^2}$$

[6 marks]

4 Distributed Systems

(*a*) You are to design a component of a distributed system which takes action on the arrival of an alarm event from another component.

Discuss the design issues, relating to the characteristics of distributed systems, of the component and its communication. [4 marks]

(*b*) You are to design a service that takes in streams of messages from distributed sources and notifies its clients when certain specified patterns of messages occur.

Discuss the design issues associated with supporting the following operators for constructing message patterns, where A and B represent messages:

(*i*) A OR B;

(*ii*) A AND B (unordered pairs);

(*iii*) A BEFORE B (ordered pairs).

[16 marks]

5 Computer Graphics and Image Processing

(*a*) Describe the A-buffer polygon scan conversion algorithm using 4×4 sub-pixels in each pixel. [10 marks]

(*b*) It is possible to represent continuous tone greyscale images using just black ink on white paper because of limitations in the human visual system. Explain how and why. [4 marks]

(*c*) Describe an algorithm which, given a greyscale image, will produce a black and white (bi-level) image of four times the resolution in each dimension which provides a good approximation to the greyscale image. [6 marks]

6 Compiler Construction

(*a*) A Java static method is defined in `class C` by

```
class C {
  public static int f(int x, int y) { int z = x; ...; return x+y*z;
                                    }
}
```

where '`...`' represents commands the details of which are not important to this question. It is called in an expression e of the form

```
  f(f(1,2), f(3,4))
```

Give JVM (or other stack machine) code corresponding to the expression e and explain how this is derived from the syntax tree for e. [6 marks]

(*b*) Explain how the body of `f` above is mapped into JVM (or other stack machine) code, explaining the rôle of the registers FP and SP (precise details are not important, but their rôle should be well explained). You may write '`...`' for the translation of the '`...`' in `f`. [6 marks]

(*c*) Consider the Java class definitions:

```
class A {
  public int a1, a2;
  public void m() { println("I am an A with " + a1 + " and " + a2);
                  }
}
class B extends A {
  public int b1, b2;
  public void m() { println("I am a B with " + a1 + " and " + a2 +
                            " also with " + b1 + " and " + b2);
                  }
}
```

Describe the run-time storage layout for objects of `class A` and for those of `class B`, particularly noting the size and offsets of members and how a *cast* of an object of type `class B` to one of `class A` can be achieved.

Explain how calls to `m()` work, particularly in code like:

```
  public static void g(B x) { h(x); }
  public static void h(A x) { x.m(); }
```

[8 marks]

 [TURN OVER

7 Artificial Intelligence I

The following Prolog relation appends a list A to a list B to give a list C.

```
append([],Y,Y).
append([H|T],Y,[H|Z]) :- append(T,Y,Z).
```

(*a*) Using the append relation, write a Prolog predicate insert(X,Y,Z) that is true if X can be inserted into a list Y to give a list Z. Your relation should be capable of using backtracking to generate all lists obtained from Y by inserting X at some point, using a query such as:

```
insert(c,[a,b],Z).
```

to obtain Z=[c,a,b], Z=[a,c,b], and Z=[a,b,c] and it should generate each possibility exactly once. [5 marks]

(*b*) Using the insert relation, write a Prolog predicate perm(X,Y) that is true if a list Y is a permutation of a list X. Your predicate should respond to a query such as

```
perm([a,b,c],Y)
```

by using backtracking to generate all permutations of the given list. [6 marks]

(*c*) We have a list of events [e1,e2,...,en]. A partial order can be expressed in Prolog by stating

```
before(e3,e4).
before(e1,e5).
```

and so on, where before(a,b) says that event a must happen before event b (although not necessarily immediately before). No ordering constraints are imposed other than those stated using before.

Given a list of events, a *linearisation* of the list is any ordering of its events for which none of the before constraints are broken. Given the example above and the list [e1,e2,e3,e4,e5], one valid linearisation would be [e3,e1,e2,e5,e4]. However, [e4,e2,e1,e5,e3] is not a valid linearisation because the first before constraint does not hold.

Using the perm predicate or otherwise, and assuming that your Prolog program contains before constraints in the format suggested above, write a Prolog predicate po(X,Y) that is true if Y is a valid linearisation of the events in the list X. Your relation should be capable of using backtracking to generate all valid linearisations as a result of a query of the form

```
po([e1,e2,e3,e4,e5],Y).
```

[9 marks]

8 Databases

(*a*) (*i*) Define the operators in the core relational algebra. [5 marks]

(*ii*) Define the domain relational calculus. [4 marks]

(*iii*) Show how the relational algebra can be encoded in the domain relational calculus. [3 marks]

(*b*) A *constraint* can be expressed using relational algebra. For example, $R = \emptyset$ specifies the constraint that relation R must be empty, and $(R \cup S) \subseteq T$ specifies that every tuple in the union of R and S must be in T.

Consider the following schema.

RockStar($\overline{\texttt{name}}$, address, gender, birthday)
RockManager($\overline{\texttt{managername}}$, starname)

(*i*) Give a constraint to express that rock stars must be either male or female. [1 mark]

(*ii*) Give a constraint to express the referential integrity constraint between the `RockStar` and `RockManager` relations. (Note: `starname` is intended to be a foreign key.) [3 marks]

(*iii*) Give a constraint to express the functional dependency `name`→`address` for the `RockStar` relation. [4 marks]

9 Numerical Analysis II

(*a*) Let n_+ be the number of positive real roots of a polynomial $p_n(x)$. Let c be the number of changes of sign when the coefficients are taken in order. State *Descartes' rule of signs.* [2 marks]

(*b*) If $p_3(x) = x^3 + 13x^2 + 54x + 72$ what does the rule say about the polynomials $p_3(x)$, $p_3(-x)$? [2 marks]

(*c*) Using Descartes' rule, a simple search technique and factorisation, find all the real and complex roots of

$$q_5(x) = x^5 + 5x^4 + 32x^3 + 160x^2 + 256x + 1280.$$

[7 marks]

(*d*) The Newton–Raphson formula $\tilde{x} = x - f(x)/f'(x)$ can often be used to find real roots of a polynomial. However, a user of numerical software reports failure to find an accurate root of

$$f(x) = 3x^4 - 28x^3 + 24x^2 + 144x + 432$$

for $x > 3$ using Newton–Raphson. The user has tried different starting values but gets either a floating-point exception or failure to converge. Using Descartes' rule on $f(x)$ and $f'(x)$ investigate the cause of these numerical problems. [9 marks]

10 Introduction to Functional Programming

(*a*) Give a definition of an ML datatype `bool_exp` to describe Boolean expressions built up from named variables using Boolean operations of conjunction, disjunction and negation:

For example, the Boolean expression $((A \vee B) \wedge \neg C) \wedge D$ would be given by

```
Conj(Conj(Disj(Var "A",Var "B"),Neg (Var "C")),Var "D")
```

[4 marks]

(*b*) Write an ML function `variables` which takes an argument `e` of type `bool_exp` and returns a value of type `string list` which lists all variables occurring in `e`. [8 marks]

(*c*) Write an ML function `eval` which takes two arguments—`e` of type `bool_exp` and `a` of type `(string * bool) list` giving a value for each variable—and returns the value of the expression `e` under the assignment `a`. [8 marks]

11 Natural Language Processing

(*a*) Define the following terms in morphology:

(*i*) morpheme

(*ii*) affix

[4 marks]

(*b*) In English morphology, 'y' maps to 'ie' when preceded by a consonant and followed by the affix 's'. Give a finite state transducer that implements this spelling rule, explaining the notation that you use. Your transducer should accept the following pairings:
party/party, parties/party ˆ s, partying/party ˆ ing

It should reject:
partys/party ˆ s, toies/toy ˆ s

[12 marks]

(*c*) The 'y' to 'ie' mapping also applies when the affix is 'ed'. Briefly discuss how this might be handled, taking into account that the morphology system should accept partied/party ˆ ed and not partieed/party ˆ ed. [4 marks]

12 Complexity Theory

If $A \subseteq \Sigma_1^*$ and $B \subseteq \Sigma_2^*$ are two languages over the alphabets Σ_1 and Σ_2 respectively, we write $A \leq_P B$ to denote that A is polynomial-time reducible to B.

(*a*) Give a precise definition of $\leq_P$ [2 marks]

(*b*) Is the relation $\leq_P$ on languages:

(*i*) reflexive?

(*ii*) symmetric?

(*iii*) transitive?

Give a proof for your answer in each case. [9 marks]

(*c*) If Σ is an alphabet, show that if P = NP then every language $L \subseteq \Sigma^*$ in NP is NP-complete except $\emptyset$ and Σ^*. Why are these two exceptions not NP-complete? [9 marks]

END OF PAPER