COMPUTER SCIENCE TRIPOS Part IB

Wednesday 3 June 1998 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.*

Write on **one** *side of the paper only.*

SECTION A

1 Structured Hardware Design

Describe individually or explain the differences between an FPGA (field programmable gate array) and a PAL (programmable array logic) device. [10 marks]

Outline an implementation of a synchronous counter in both styles of programmable logic. [6 marks]

What are the primary performance limitations when building large counters in each style of logic? [4 marks]

2 Computer Design

Early computers (and early microprocessors) were accumulator machines. RISC computers replaced the accumulator with a register file.

(*a*) What is a register file and why is it preferable to an accumulator? Illustrate your answer by writing a loop to calculate factorial of 10 for an accumulator and a RISC processor (you may invent instruction sets and assume that a multiply instruction is available). [12 marks]

(*b*) Why is the Intel x86 processor family often referred to as being an extended accumulator machine? [4 marks]

(*c*) The Intel x86 `LOOP` instruction decrements the `CX` register and, if the result is not zero, jumps to a given label. Why is a compiler likely to find it hard to exploit this instruction, especially for nested loops? [4 marks]

3 Digital Communication I

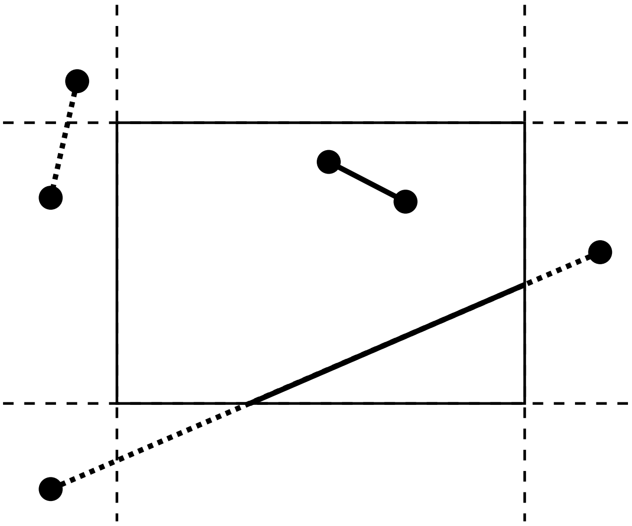
How can packet loss occur in a network? [5 marks]

Outline a way in which packet loss can be reduced. Can it be eliminated completely? [5 marks]

How does an ARQ system deal with packet loss? [5 marks]

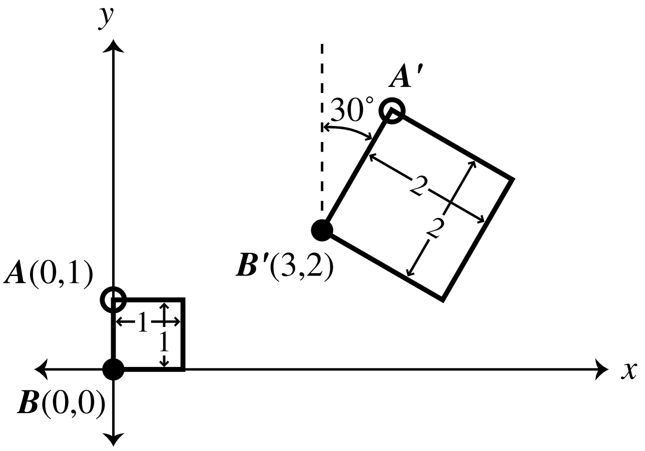
"An ARQ implementation should keep as much data in transit as the receiver is willing to receive." Discuss. [5 marks]

4 Computer Graphics and Image Processing



Describe an algorithm for clipping a line against a rectangle. [9 marks]

Show that it works using the above three examples. [3 marks]



The above diagram shows a complicated 2D transformation applied to a unit square. The overall transformation can be described in terms of a number of simpler transformations. Describe each of these simple transformations and give a matrix representation of each using homogeneous coordinates. [6 marks]

Use the matrices from the previous part to find the (x, y) coordinates of point A', the image of point A under the overall transformation. [2 marks]

SECTION B

5 Introduction to Security

Some banks issue their Automatic Teller Machine (ATM) card customers with a randomly selected personal indentification number (PIN). Others issue their customers with an initial PIN only, and let the customers choose their own PIN the first time they use the card in an ATM. Describe the advantages and disadvantages of these approaches. [5 marks]

Again, some banks compute the customer PIN by encrypting the account number using DES and a key known only to their central systems and ATMs, taking the first four hex digits of the result, replacing the digits A, . . . , F with 0, . . . , 5 respectively, and finally, if the first digit of the result is 0, replacing it with a 1. What is the probability that a criminal can get the PIN right given three guesses? [5 marks]

Yet other banks have used DES, and a key known only to their central systems and ATMs, to encrypt the PIN (whether randomly generated or customer selected); they then write the result on the magnetic strip on the customer's card, so that the ATM can verify it without reference to the central system. Describe the disadvantages of this arrangement. [5 marks]

In order to prevent attacks based on manipulating magnetic strips, banks in some countries have moved to using smart cards. What effect would you expect such a move to have on the incidence of card-based fraud? [5 marks]

6 Compiler Construction

Explain how a parse-tree representation of a program may be converted into a stack-based intermediate language giving sketches of code to translate expressions, assignments and the `if-then-else` command; you should also explain how occurrences of a variable in an expression or assignment are translated.

The program may be assumed to conform to the following syntax:

```
E  ->  n | x | E + E | f(E,E)
D  ->  let f(x,x) = {Dseq; Cseq; E} | let x = E
C  ->  x := E; | if E then C else C
Cseq -> C | C Cseq
Dseq -> D | D Dseq
```

with start symbol `Dseq`. Here `n` corresponds to integer constants, `x` corresponds to identifiers used as variable names and `f` corresponds to identifiers used as function names (you may assume these are disjoint). The function declaration construct has the effect of defining a function which, when called, makes declarations, performs commands and then returns the result of its expression; note that therefore functions may be defined within functions, but the above restriction on identifiers means that they cannot be returned as results. [20 marks]

7 Prolog for Artificial Intelligence

Write Prolog programs that define the following predicates. Your programs should ensure that backtracking does not produce spurious alternative solutions.

(*a*) The *n*th element of a list: nth(X,N,L) instantiates X to the Nth element of list L. Assume that list elements are numbered increasing from 1. [4 marks]

(*b*) The last element of a list: last(X,L) instantiates X to the last element of list L. [4 marks]

(*c*) Remove an element from a list: remove(X,L,M) instantiates M to a list containing all the elements of list L except for every occurrence of term X. [6 marks]

(*d*) Substitute one element for another: subst(L,X,Y,M) instantiates M to a list containing all the elements of list L except that every occurrence of term X in L is replaced by term Y in M. [6 marks]

8 Databases

The relational model of data was introduced in the early 1970s in a sequence of papers by E.F. Codd. This model proposes a tabular view of data, with a simple Data Manipulation Language (DML) based on relational algebra or relational calculus. Briefly explain the essential features of the model and its DML. [6 marks]

Later work by Codd and others addressed weaknesses in the expressive power of the relational model. For each of the following give an example to show how the weakness arises, and explain an approach proposed to resolve the difficulty:

(*a*) computing the transitive closure [4 marks]

(*b*) manipulating collections of records [4 marks]

(*c*) handling entity specialisation [6 marks]

In case (*c*) you should outline the proposals of *either* the Object-Oriented Database System Manifesto *or* the Third Generation Database System Manifesto.

SECTION C

9 Foundations of Functional Programming

State the connection between the β-equality relation ($=_\beta$) and the β-reduction relation on λ-terms. Prove that $=_\beta$ is non-trivial, in the sense that there exist λ-terms M and N such that $M \neq_\beta N$. [4 marks]

Compare the call-by-name and call-by-value reduction strategies, giving examples to illustrate that

(*a*) sometimes the call-by-name strategy gives fewer reductions than the call-by-value strategy, and *vice versa*;

(*b*) the call-by-name strategy terminates when the normal form exists, whereas the call-by-value strategy need not.

[6 marks]

Given the λ-term $(\lambda x.xI)(\lambda y.(\lambda z.zzzz)(yt))$ where I is $\lambda u.u$, display reduction paths arising from the call-by-name and call-by-value reduction strategies. Also, find the reduction path which consists of the fewest reduction steps and comment on your answer. [10 marks]

10 Logic and Proof

Construct an ordered binary decision diagram (OBDD) for the formula

$$[(P \to Q) \land (\neg R \lor \neg Q)] \to \neg R,$$

showing each step carefully. What does the OBDD tell us about whether the formula is (*a*) valid, (*b*) satisfiable and (*c*) inconsistent? [10 marks]

Attempt to prove the above formula using the sequent calculus until either it is proved or the proof cannot be continued. [4 marks]

Design a method for determining whether a propositional formula is inconsistent. The method should work by examining the formula's disjunctive normal form. Demonstrate your method by applying it to the formula

$$\neg[(P \land Q) \lor (Q \to P)].$$

[6 marks]

11 Complexity Theory

(*a*) Describe the problem 3-SAT. [2 marks]

(*b*) Show how any instance of the seemingly more general problem n-SAT can be reduced to an equivalent one where each term has *exactly* three literals in it. Estimate how much larger the reduced problem would be than the original one. [4 marks]

(*c*) A certain computation using a non-deterministic Turing machine completes in T time-steps. The Turing machine has k states and uses an alphabet of N symbols. A major theorem underpinning the concept of NP-completeness is based on a conversion of a description of such computations to boolean formulae which characterise them.

Explain how, in such a reduction, boolean variables may be used to describe states that the Turing machine might be in. Show how to derive those components of the boolean formula that relate just to the way in which the Turing machine moves its read–write head. Your explanation should be sufficiently complete and carefully explained that it could be used as a specification of a program that would perform that part of the translation from Turing machine descriptions to boolean formulae. You should not attempt to explain the rest of the boolean formula or how it fits into a complete proof or program. [11 marks]

(*d*) In terms of T, k and N, about how many symbols does it take to write the boolean expression you generate? [3 marks]

12 Semantics of Programming Languages

An abstract machine for evaluating closed terms of the untyped lambda calculus has configurations which are non-empty lists of closed terms. Its transitions are of two forms:

$(\overrightarrow{\text{app}})$ $\qquad (M_1\, M_2) :: L \to M_1 :: M_2 :: L$

$(\overrightarrow{\text{abs}})$ $\qquad \lambda\, x\, (M_1) :: M_2 :: L \to M_1[M_2/x] :: L$

where :: denotes list concatenation and $M_1[M_2/x]$ denotes the result of substituting M_2 for all free occurrences of the variable x in M_1. Let $\Downarrow$ be the binary relation between closed terms inductively defined by the following axioms and rules:

$(\Downarrow_{\text{abs}})$ $\qquad \lambda\, x\, (M) \Downarrow \lambda\, x\, (M)$

$(\Downarrow_{\text{app}})$ $\qquad \dfrac{M_1 \Downarrow \lambda\, x\, (M_2) \quad M_2[M_3/x] \Downarrow \lambda\, x\, (M_4)}{M_1\, M_3 \Downarrow \lambda\, x\, (M_4)}.$

(*a*) Prove by Rule Induction that if $M_1 \Downarrow \lambda\, x\, (M_2)$ holds, then so does $M_1 :: L \to^* \lambda\, x\, M_2 :: L$, where $\to^*$ denotes the reflexive-transitive closure of the transition relation $\to$. [5 marks]

(*b*) Prove by Mathematical Induction on n that if $(\ldots((M[M_0/x]\, M_1)M_2)\ldots)M_n \Downarrow \lambda\, x\, (M')$, then $(\ldots((((\lambda\, x\, (M))M_0)M_1)M_2)\ldots)M_n \Downarrow \lambda\, x\, (M')$. [5 marks]

(*c*) Given a configuration $M :: L$, let $M@L$ denote the closed term defined by induction on the length of the list L by:
$M@\mathbf{nil} \overset{\text{def}}{=} M$ and $M@(M' :: L) \overset{\text{def}}{=} (M\, M')@L$. Using (*b*), show by case analysis for $\to$ that if $M_1 :: L_1 \to M_2 :: L_2$ and $M_2@L_2 \Downarrow \lambda\, x\, (M')$ hold, then so does $M_1@L_1 \Downarrow \lambda\, x\, (M')$. [5 marks]

(*d*) Deduce from (*a*) and (*c*) that $M_1 \Downarrow \lambda\, x\, (M_2)$ holds if and only if $M_1 :: \mathbf{nil} \to^* \lambda\, x\, (M_2) :: \mathbf{nil}$ does. [5 marks]