COMPUTER SCIENCE TRIPOS Part IB

Tuesday 3 June 2008 1.30 to 4.30

PAPER 4

*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

STATIONERY REQUIREMENTS
Script paper
Blue cover sheets
Tags

SPECIAL REQUIREMENTS
None

1 Concurrent Systems and Applications

(*a*) Describe the syntax of the `synchronized` keyword in Java and explain the effect on the runtime behaviour of a program. [5 marks]

(*b*) Compare and contrast the approaches of using a single mutex to guard access to an entire data structure and using individual mutexes on each unit of storage within the data structure. [5 marks]

(*c*) Consider a queue data structure, based on a linked list. The operations `pushTail` and `popHead` are to be provided and it should be possible to execute both concurrently whenever doing so would be safe (but not when it would be unsafe). Provide a Java implementation of the data structure and concurrency control mechanism, including the methods to push new items on the tail of the queue and to pop items from the head. [10 marks]

2 Algorithms II

(*a*) Briefly describe the Dijkstra algorithm for finding shortest paths from a single source and explain why it cannot be used on graphs with negative edge weights. [Pseudocode is not required.] [4 marks]

(*b*) Describe and explain in detail the Johnson algorithm that finds all-pairs shortest-paths by repeatedly applying Dijkstra to each vertex, even in graphs with negative edge weights. [Pseudocode is not required but all phases of the algorithm must be clearly explained.] [7 marks]

(*c*) Some people wonder why it would not be simpler to reweight edges by adding a sufficiently large constant K to each edge weight so as to make all the weights positive. Prove that this cannot work. [2 marks]

(*d*) In Johnson's algorithm, why do we introduce a new vertex s? Could we not use, instead of a new vertex, one of the vertices of the original graph? Either prove that we can or provide a counterexample. [7 marks]

3 Compiler Construction

(*a*) Given the following program fragment in C or Java

```
static int a = 3;
static int f(int x, int y) { int z = a+x; ...; return ... }
```

explain how the four variables (`a`, `x`, `y` and `z`) are accessed from within `f` at the instruction level for an architecture such as MIPS, ARM or x86. Pay particular attention as to how and when storage is allocated for these variables, and which system components of a standard compile-link-and-execute model are involved in selecting the instruction and determining the run-time address calculation. Your answer should briefly explain what occurs when `f` makes a recursive call to itself. [8 marks]

(*b*) Suppose we extend the language to allow nested function definitions:

```
static int a = 3;
static int f(int x) {
  static int g(int y) { int z = a+x; ...; return ... }
  return g(7);
}
```

(*i*) Complete the definition of `g` in such a way that, during execution of a call to `f`, the difference between the addresses allocated to `x` and `y` varies. [2 marks]

(*ii*) Explain a possible run-time mechanism that allows all four variables to be accessed from inside `g`. [6 marks]

(*c*) Suppose function-valued variables are added to the language, e.g.:

```
funvar double v(double);
v = (...) ? sin : cos;
```

and a pointer to code or to data is represented as a 32-bit value. How many bits might naturally be used to hold `v`? Justify your answer, emphasising the way in which it might differ according to whether the language is as given in part (*a*) or part (*b*) above. [4 marks]

4 Mathematical Methods for Computer Science

(*a*) Consider a *simple random walk*, S_n, defined by $S_0 = a$ and $S_n = S_{n-1} + X_n$ for $n \geq 1$ where the random variables X_i $(i = 1, 2, \ldots)$ are independent and identically distributed with $P(X_i = 1) = p$ and $P(X_i = -1) = 1 - p$ for some constant p with $0 \leq p \leq 1$.

(*i*) Find $E(S_n)$ and $Var(S_n)$ in terms of a, n and p. [4 marks]

(*ii*) Use the *central limit theorem* to derive an approximate expression for $P(S_n > k)$ for large n. You may leave your answer expressed in terms of the distribution function $\Phi(x) = P(Z \leq x)$ where Z is a standard Normal random variable with zero mean and unit variance. [6 marks]

(*b*) Consider the *Gambler's ruin problem* defined as in part (*a*) but with the addition of absorbing barriers at 0 and N where N is some positive integer. Derive an expression for the probability of ruin (that is, being absorbed at the zero barrier) when starting at position $S_0 = a$ for each $a = 0, 1, \ldots, N$ in the two cases

(*i*) $p \neq \frac{1}{2}$ [5 marks]

(*ii*) $p = \frac{1}{2}$. [5 marks]

5 Logic and Proof

(*a*) State (with justification) whether the following formula is satisfiable, valid or neither. Note that a and b are constants.

$$\Big[\forall x\,[q(x) \to r(x)] \wedge \neg r(a) \wedge \forall x\,[\neg r(x) \wedge \neg q(a) \to p(x) \vee q(x)]\Big] \to p(b) \vee r(b)$$

[13 marks]

(*b*) Attempt to prove the formula $[\exists x\,\forall y\,R(x, y)] \to \exists x\,\forall z\,R(x, f(z))$ by resolution, with brief explanations of each step, including the conversion to clause form. [4 marks]

(*c*) Give a model for the following set of clauses, or prove that none exists.

$$\{\neg R(x, y), \neg R(y, x)\}$$

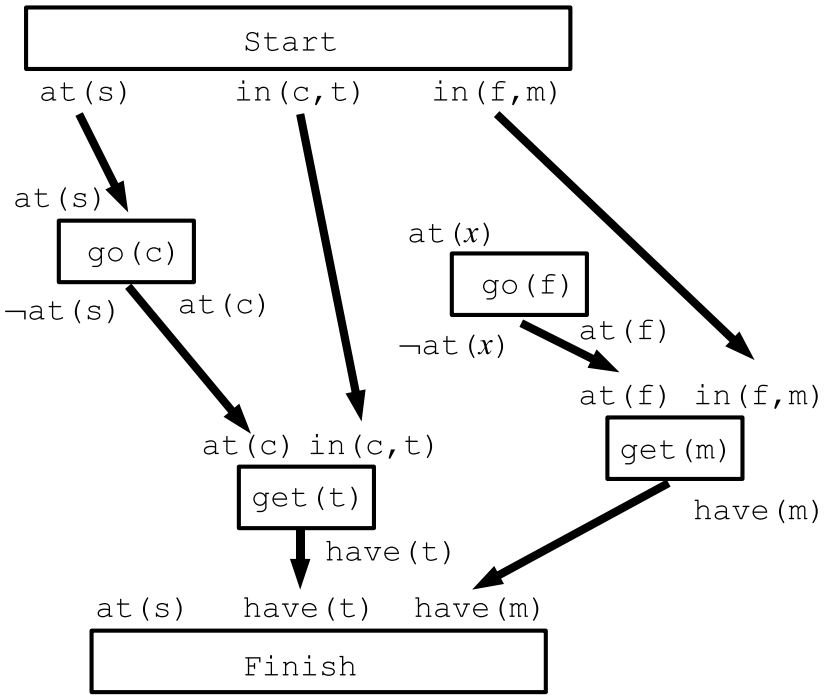
$$\{R(x, f(x))\}$$

$$\{\neg R(x, y), \neg R(y, z), R(x, z)\}$$

[3 marks]

6 Artificial Intelligence I

A brilliant student has finished his exams and is making a well-deserved cup of tea. He is confused, however, and is trying to use the *partial order planning* algorithm to solve part of the problem. Using the abbreviations `f` for "fridge", `c` for "cupboard", `s` for "sink", `m` for "milk" and `t` for "tea", his start state is $\{\texttt{at}(\texttt{s}), \texttt{in}(\texttt{c}, \texttt{t}), \texttt{in}(\texttt{f}, \texttt{m})\}$. Using x and y to denote variables, he has two actions. The first action is $\texttt{get}(y)$ having preconditions $\texttt{at}(x)$ and $\texttt{in}(x, y)$, and effect $\texttt{have}(y)$. The second action is $\texttt{go}(y)$ having precondition $\texttt{at}(x)$ and effects $\neg\texttt{at}(x)$ and $\texttt{at}(y)$. His goal is $\{\texttt{at}(\texttt{s}), \texttt{have}(\texttt{t}), \texttt{have}(\texttt{m})\}$. So far he has made the following attempt at finding a plan:



In this diagram, arrows denote causal links.

(a) Can the $\texttt{at}(x)$ precondition on $\texttt{go}(\texttt{f})$ be achieved by adding an ordering constraint and causal link from `Start` to $\texttt{go}(\texttt{f})$, and perhaps one or more further ordering constraints, in such a way that the plan remains valid? Explain your answer. [4 marks]

(b) Describe a method, different from any suggested in part (a), by which the $\texttt{at}(x)$ precondition on $\texttt{go}(\texttt{f})$ can be achieved in such a way that the plan remains valid. [8 marks]

(c) Describe a way in which the plan can be completed after making the addition you have described in part (b). [8 marks]

7 Introduction to Security

(*a*) The following files are shown by an `ls -l` command on a typical Unix system:

```
-r-xr-sr-x  1 charlie acct    70483 2008-01-04 22:53 accounting
-r--rw----  1 alice   acct   139008 2008-05-13 14:53 accounts
-rwxr-xr-x  1 system  system 230482 1997-04-27 22:53 editor
-rw-r--r--  1 alice   users    7072 2008-06-01 22:53 cv.txt
-r--r-----  1 bob     gurus   19341 2008-06-03 13:29 exam
-r--r-----  1 alice   gurus    6316 2008-06-03 16:25 solutions
```

Unix users `alice` and `bob` are both members of only the group `users`, while `charlie` is a member of only the group `gurus`. Application `editor` allows users to read and write files of arbitrary name and change their permissions, whereas application `accounting` only allows users to append data records to the file `accounts`. Draw up an access control matrix with subjects {`alice`, `bob`, `charlie`} and objects {`accounts`, `cv.txt`, `exam`, `solutions`} that shows for each combination of subject and object whether the subject will, in principle, be able to read (R), (over)write (W), or at least append records (A) to the respective object. [9 marks]

(*b*) A C program uses the line

```
buf = (char *) malloc((n+7) >> 3);
```

in order to allocate an $\lceil \frac{\mathtt{n}}{8} \rceil$-bytes long memory buffer, large enough to receive `n` bits of data, where `n` is an unsigned integer type.

(*i*) How could this line represent a security vulnerability? [2 marks]

(*ii*) Modify the expression that forms the argument of the `malloc()` call to avoid this vulnerability without changing its normal behaviour. [3 marks]

(*c*) Name *three* types of covert channels that could be used to circumvent a mandatory access control mechanism in an operating system that labels files with confidentiality levels and give a brief example for each. [6 marks]

8 Prolog

The Prolog predicate `perm(+In,-Out)` generates all permutations of the input list `In`. A programmer implements `perm/2` as follows:

```
perm([],[]).
perm(L,[H|T]) :- take(L,H,R), perm(R,T).
```

The predicate `take(+L,-E,-R)` removes one element (`E`) from the input list `L` and unifies `R` with the remainder of `L`. Thus, the list `R` has one element fewer than `L`.

(*a*) Consider the `perm/2` predicate:

(*i*) Explain briefly in words the operation of the `perm/2` predicate. [3 marks]

(*ii*) Provide an implementation of the `take/3` predicate. [4 marks]

(*iii*) Give the complete sequence of answers (in the correct order) generated by `perm([1,2,3],A)`. [3 marks]

(*b*) A student attempts to invoke the query `perm(A,[1,2,3])`.

(*i*) Explain what happens and why. [5 marks]

(*ii*) Implement a predicate `sameLength/2` which is true if the two parameters are lists of the same length. [2 marks]

(*iii*) Using `sameLength/2`, or otherwise, provide an implementation of `safePerm/2` which generates permutations regardless of the order in which the parameters are provided: both `safePerm(+In,-Out)` and `safePerm(-Out,+In)` should generate all permutations of `In`. The order in which these permutations are generated is not important. [3 marks]

END OF PAPER