COMPUTER SCIENCE TRIPOS Part IB

Monday 6 June 2011 1.30 to 4.30

COMPUTER SCIENCE Paper 3

*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
Approved calculator permitted

1 Algorithms II

The pseudocode below is a first attempt at a recursive algorithm to enumerate all the paths from source to sink, in the context of a maximum flow problem.

```
0 def allPaths(graph, source, sink):
1       # Each path is a list of vertices from source to sink, eg [2, 4, 7]
2       # The result is a list of paths, eg [[2, 4, 7], [2, 7]], initially empty
3     result = []
4
5     if source == sink:
6         result.append([source])
7     else:
8         for v in graph.verticesAdjacentTo(source):
9             for path in allPaths(graph, v, sink):
10                  # Reject paths that revisit the source, else infinite loops
11                if source not in path:
12                    result.append([source] + path)
13    return result
```

(*a*) Point out all the bugs you can find, highlighting the failures with test cases. [5 marks]

(*b*) Correct all the bugs you found, clearly explaining your fixes. Rewrite a corrected and clearly commented version of the pseudocode. [10 marks]

(*c*) Provide a correctness proof for your new version. [5 marks]

2 Algorithms II

(a) Explain, with prose and no more than 15 lines of pseudocode, the "Jarvis's march" algorithm for finding the convex hull of a set of points. Marks awarded for: clarity of explanation, legibility of pseudocode, correctness of algorithm. [5 marks]

(b) Consider the following claim: "The cross-product trick can be used as a plug-in substitution for the comparison operation between two arguments that gives a three-way result (namely $<$, $>$ or $=$). Therefore, given an arbitrary set of 2D vectors all starting from the origin, I can sort them in order of increasing polar angle, from 0 to 2π, without computing any angles, merely by applying a standard sort algorithm with the comparison operation replaced by the cross-product trick".

(i) Explain the "cross-product trick". [3 marks]

(ii) Prove the correctness of the above claim or refute it with a clearly explained graphical or numerical counterexample. [7 marks]

(c) Traditionally, Jarvis's march handles the left side and the right side of the hull separately. Clearly explain why a single pass would not work as desired and then describe a variant implementation to handle the problem in one pass, highlighting its advantages and disadvantages.

[Hint: see part (b)(ii).] [5 marks]

3 Programming in C and C++

In this question, where appropriate, you may use a short fragment of code to complement your explanation.

(*a*) What is the difference between declaration and definition?

(*b*) Describe the layout of the memory components: Dynamic Memory Allocation, Data Segment, Code Segment and Stack. You may use an illustration as part of your explanation.

(*c*) Using an example, explain what *stack unwinding* is in C++.

(*d*) How may I use template meta programming to inline recursive functions?

(*e*) Why did the designers of the C++ language decide to make an empty class 1 byte in length?

[4 marks each]

4 Compiler Construction

(*a*) In a stack-based runtime system, what problem does the *static link method* attempt to solve, and how does it work? [4 marks]

(*b*) Can static linking be used to implement a language with first-class functions? If yes, then explain how. If no, give an example and explain how static linking fails. [6 marks]

(*c*) Explain how exceptions (ML-like `raise` and `handle`) could be implemented with a stack-oriented machine. [5 marks]

(*d*) A program may evaluate to an exception that has been raised all the way to the top-level and never handled. Discuss how you might modify your implementation in part (*c*) to dump debugging information when such top-level exceptions are raised. The debugging information should include some description of the state of the computation *just before the top-level exception was raised*. [5 marks]

5 Compiler Construction

Consider a simple grammar for arithmetic expressions:

$$E ::= n \mid x \mid -E \mid E \ + \ E \mid (\ E\ )$$

with n ranging over integer constants and x ranging over variables. We want to compile expressions to code for a simple stack-based machine with the following instruction set.

instruction	meaning
`pushvar` k	push the value of the k-th variable on top of stack
`push` n	push n on top of stack
`add`	replace the top two stack items with their sum
`neg`	replace the top stack item with its negation

For this problem, we will not worry about how variables are bound to values nor how abstract syntax trees are produced.

(a) How will your compiler generate code from expressions of the form $-E$?
[4 marks]

(b) How will your compiler generate code from expressions of the form $E_1 \ + \ E_2$?
[4 marks]

(c) What code will your compiler generate for the expression `-(-x + (17 + y))`?
[4 marks]

(d) Suppose we now want to extend the language of integer expressions with multiplication

$$E ::= \cdots \mid E \ * \ E$$

but we cannot extend the machine with an instruction for multiplication.

Can you implement this extended language directly with the machine instruction set presented above? If not, suggest a *minimal* extension to the instruction set that allows for the implementation of multiplication using the addition from the instruction set. Explain the semantics of your extensions and how you would use them to implement multiplication. [8 marks]

6 Concepts in Programming Languages

(*a*) Give **one** difference and **one** similarity between the programming languages:

(*i*) Algol and SIMULA [2 marks]

(*ii*) LISP and Smalltalk [2 marks]

(*b*) What is the type of the expression `fn f => fn x => f(f(x))` inferred by the SML interpreter? Explain your answer. [6 marks]

(*c*) Give an example in the SML Modules language of two distinct signatures, say `IN` and `OUT`, and of a functor that takes structures matching `IN` to produce structures matching `OUT`. [6 marks]

(*d*) Comment on the mechanism for parameter-passing in the programming language Scala. [4 marks]

You may wish to consider the following two code samples.

```
def whileLoop( cond: => Boolean )( comm: => Unit )
  { if( cond ) comm; whileLoop( cond )( comm ) }

def qsort[T]( xs: Array[T] )( implicit o: Ord[T] ): Array[T]
  = if( xs.length <= 1 ) xs else
    { val pivot = xs( xs.length/2 )
      Array.concat
        ( qsort( xs filter (x => x.lt(x,pivot)) ) ,
          xs filter (x => x == pivot )            ,
          qsort( xs filter (x => x.lt(pivot,x)) ) ) }
```

7 Further Java

In this question you will need to fill in missing parts of a Java program. You may ignore any exception handling and will not be penalised for minor syntactic errors.

You are provided with a class `Eval`:

```
public class Eval {
  public static int f(Record r) { ... }
}
```

(*a*) Add another method `Integer maxf(Iterator<Record> it)` to the class `Eval`. Your method should return the maximum value computed by `f` for every `Record` returned by the iterator or `null` if there are no records available. The relevant portion of the `Iterator` interface is as follows:

```
interface Iterator<T> {
  // return true if there are more values available
  public boolean hasNext();

  // return the available value and advance to the next one
  public T next();
}
```

[5 marks]

(*b*) Complete the methods `run()` and `join()` in the following abstract class. You may add additional fields or methods if you wish.

```
abstract class Joinable implements Runnable {
  abstract void exec();
  final public void run() {
    // ... call the exec() method ...
  }
  void join() throws InterruptedException {
    // block the calling thread until exec() completes in run()
  }
}
```

[7 marks]

(*c*) Provide a method `Integer parmaxf(Iterator<Record> it, int n)` which is functionally equivalent to `Eval.maxf`, except that it should create `n` parallel threads of execution to speed up the calculation of the result. You may assume that `Iterator<Record>` is thread-safe. You may find it helpful to subclass the `Joinable` class. [8 marks]

8 Prolog

(*a*) Give the result and any variable bindings that occur from making each of the following (independent) queries:

(*i*) `A=3` [1 mark]

(*ii*) `A is 4` [1 mark]

(*iii*) `A<5` [1 mark]

(*iv*) `A=6,not(A=6)` [1 mark]

(*b*) Consider the following clauses:

```
a(1).
a(a).
b(3).
b(a).
c(A,B) :- b(B),!,a(A).
c(X,_) :- a(X),b(X).
```

(*i*) List all the solutions to the query `c(A,B)` in order, giving any binding of variables that occurs. [2 marks]

(*ii*) List all the solutions to the query `c(X,1)` in order, giving any binding of variables that occurs. [2 marks]

(*c*) A binary tree has nodes whose values are non-empty lists. A tree node is represented using a term that takes the form `Left-SomeList+Right` or `nil` (note that parentheses can be used to enforce precedence). For example, the following term would be a valid tree:

```
nil-[root,1,2]+(nil-[child,4]+nil)
```

(*i*) Write a predicate `preorder(+Tree,-ValueList)` that unifies `ValueList` with a list containing all of the tree nodes' values from a pre-order tree walk (i.e. emit node value, then left subtree, then right subtree). The predicate should fail if any of the tree nodes' values are not of the correct form. For the above example tree, `preorder/2` would unify `ValueList` with `[[root,1,2],[child,4]]`. You may assume that `append/3` has been defined already. [6 marks]

(*ii*) Now write a predicate `preorderdl/2` that behaves exactly like your `preorder/2` predicate, but in its implementation makes use of difference lists, instead of querying `append/3`. [6 marks]

9 Software Engineering

(*a*) Describe the waterfall model for software development and list **three** of its advantages for software development. [5 marks]

(*b*) When discussing system development Fred Brooks says, "plan to throw one away – you will anyway". What disadvantages of the waterfall model is he referring to? Outline an alternative software development model that deals with these disadvantages. [7 marks]

(*c*) You work for a large "social networking" company which has recently introduced a one-to-one chat mechanism, promising that they will never censor conversations. Users are now reporting that their friends' computers are being compromised by malicious software. When users click on links within messages sent by this malicious software, their machine is also compromised, and spreads the infection still further. A crisis meeting has decided that the chat software must be modified to block this "worm" behaviour.

As manager of this project, how will you approach the development, how will you estimate how long the task will take, and how will you establish that your solution is safe to deploy? [8 marks]

END OF PAPER