COMPUTER SCIENCE TRIPOS Part Ib

Tuesday 31 May 1994 1.30 to 4.30

Paper 4

Answer **five** *questions.*
Submit the answers in five **separate** *bundles each with its own cover sheet.*
Write on **one** *side of the paper only.*

1 Further Modula-3

Arrays in Modula-3 can be indexed only by ordinal types, which means that, for example, it is not possible to have an array indexed by `TEXT` strings. One way of overcoming this restriction is to use *hash tables*.

Write a generic interface, `Table`, to manage tables of values of type `Value.T` indexed by values of type `Key.T`. A `Table.T` should be an opaque object with methods

`init` to initialise the hash table to a particular size, and

`put` and `get` to store and recover values,

all having appropriate signatures. Use exceptions to indicate duplicate or missing keys. [6 marks]

Sketch a generic implementation of the `Table` module giving a concrete revelation of `T` and providing appropriate default methods. It is not necessary to protect the data structure against concurrent access. [8 marks]

What constraints do your interface and implementation impose on interfaces supplied for `Key` and `Value` when instantiating `Table`? [2 marks]

Show how `Table` could be instantiated to provide an array of integers indexed by text strings, and indicate how this might be used to manage a (write-once) telephone directory. [4 marks]

2 Programming Language Compilation

Carefully describe how to construct the precedence matrix for a given grammar, illustrating your answer by investigating whether the following is a precedence grammar.

```
S -> a P Q b  |  c Q P d
P -> X
Q -> Y
X -> p  |  X x Y
Y -> q  |  y Y z
```

[10 marks]

Explain why replacing the productions

```
S -> a P Q b  |  c Q P d
```

by

```
S -> a X Y b  |  c Y X d
```

would introduce conflicts in the precedence matrix. [2 marks]

Give a detailed description of a parsing algorithm that uses a precedence matrix. [8 marks]

3 Formal Languages and Automata

Explain what is meant by a *context-free* language. [5 marks]

Show that the union of two context-free languages (over the same alphabet Σ) is again context-free. [5 marks]

Consider the language L over the alphabet $\{a, b, c\}$ consisting of all strings of the form $a^\ell b^m c^n$, where $\ell, m, n > 0$ and either $\ell = m$ or $m = n$. Is L context-free? Is it a regular language? Justify your answers, stating carefully any well-known results that you use. [10 marks]

4 Operating System Functions

Describe the use of a *table of pointers* and a *table of extents* to store the list of disc blocks which compose a file. Pay particular attention to how such data structures can be designed to enable the efficient support of very large files, and give estimates of their performance for finding the location of a block chosen at random. [14 marks]

Describe how such data structures would be used in the implementation of a *log-structured* file system with consideration of the impact of garbage collection. [6 marks]

5 UNIX Case Study

For the UNIX operating system, define

(a) process [2 marks]

(b) the execution environment of a process [3 marks]

(c) the `fork` system call [3 marks]

Explain how a command line of the form given below is implemented:

```
command <argument-list> &
```
[9 marks]

Outline the essential difference in the implementation of command lines of the form given below:

```
command1 <arg1-list> & command2 <arg2-list>
command1 <arg1-list> | command2 <arg2-list>
```
[3 marks]

6 Data Structures and Algorithms

Explain how it is possible to construct *red/black* search trees, using one extra bit per node, so that the trees remain reasonably balanced. [13 marks]

How out of balance can your tree get and what are the worst case costs for adding a new item and finding an existing one? [7 marks]

7 Data Structures and Algorithms

Suppose that you have been provided with a procedure that can, given n items, find the one that would come at position $n/3$ if the items were sorted into ascending order. Further suppose that when this procedure is called it always costs exactly $10n$ comparisons. Explain how this would allow you to implement a variant on Quicksort exhibiting guaranteed good computing time. [7 marks]

Estimate very roughly (but justify your estimate) how much data would be needed before you could be certain that the worst case for a simple implementation of Quicksort would involve more comparisons than the worst case of your new algorithm. [10 marks]

Would you ever expect anybody to want to use your new method in practical applications? [3 marks]

8 Graphics

Describe the method of Douglas & Pücker for approximating one line chain by another to within some tolerance. Are there any special cases? [20 marks]

9 Numerical Analysis I

With reference to a decimal floating-point implementation with 4-digit precision ($\beta = 10$, $p = 4$), describe the two most common methods of rounding. (Use 1.2345 and 1.2375 as examples.) Which method is unbiased? [3 marks]

What do you understand by the terms *machine epsilon*, and *guard digit*? [4 marks]

Suppose the largest representable floating-point number is about 10^{50}, and consider evaluation of $\sqrt{x^2 - y^2}$. How would you compute the result? (Use $x \simeq 5.10^{40}$, $y \simeq 3.10^{40}$ as an example.) How could your method also improve accuracy on some machines? [3 marks]

A programmer writes $(x+y)+z$ but a compiler evaluates the right-hand side in the form $x + (y + z)$. Explain how this could be harmful in floating-point arithmetic (a) when x, y and z are large, and (b) when x, y and z are numbers of moderate size. Which of these two problems would be more likely to occur in practice: (a) or (b)? [3 marks]

Explain the term NaN as used in IEEE arithmetic. Roughly, how many NaN values are there in IEEE single precision? Consider an *operation* to be any one of $+ - * /$. Give examples of (a) an operation that yields a NaN value when neither of its arguments is a NaN, (b) an operation with finite arguments that yields $+\infty$, (c) an operation with an argument $+\infty$ that yields a finite result. [5 marks]

What two rules govern operations where at least one argument is a NaN value? [2 marks]

10 Structured Hardware Design

For each of the following entities which are held in the design database of a digital CAD/CAE system, describe their function and the attribute values or data structures that the system must maintain for each instance. Include both the simulation phase and the mask or PCB generation phase of the design process.

(a) a signal net, including provision for back annotation of layout delays [10 marks]

(b) a stable master clock oscillator [3 marks]

(c) a tri-state buffer, with output load dependent delay [4 marks]

(d) an artificial stimulus source, used to model an existing external interface component [3 marks]