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

Tuesday 31 May 1994 1.30 to 4.30

Paper 11 (Paper 2 of Diploma in Computer Science)

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 Prolog

Consider the following problem to be solved using a Prolog program:

Given a closed planar polygon chain represented as a list of n vertices

$$\texttt{[v(}x_1\texttt{,}y_1\texttt{), v(}x_2\texttt{,}y_2\texttt{), ..., v(}x_n\texttt{,}y_n\texttt{)]}$$

compute the area of the enclosed polygon, and the orientation of the chain. The area is computed by the line integral $1/2 \int x \, \mathrm{d}y - y \, \mathrm{d}x$ where the integral is over the polygon chain. A naïve solution is given by the following program, which defines the predicate `area`. The goal `area(Chain,Area)` succeeds when `Chain` is the list of vertices, and the magnitude of `Area` is the area of the polygon bounded by the chain. The sign of `Area` is positive if the orientation of the polygon is anticlockwise and negative if it is clockwise:

```
area([X],0).
area([v(X1,Y1),v(X2,Y2)|VS],Area):-
          area([v(X2,Y2)|VS],Temp),
          Area is Temp + (X1 * Y2 - Y1 * X2) / 2.
```

Explain how vertices are processed by this procedure. [4 marks]

Why does this program execute inefficiently? [3 marks]

Write an alternative definition that is tail-recursive (iterative) and makes use of accumulator variables. [10 marks]

Explain why your alternative definition executes more efficiently. [3 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

Give an algorithm which selects the *median* value from an array of n values explaining how the possible execution time depends on n. (You should clarify the word "possible".) [10 marks]

Describe two tests. The first should determine whether two line segments intersect. The second should determine whether a given (half-infinite) straight line from a point (x, y) in the direction (u, v) [thus the points on the line are $(x + \mu u, y + \mu v)$ for $\mu \geqslant 0$] intersects a given polygon *other than at a vertex.* [10 marks]

7 Distributed Systems

Explain the importance of ensuring the timeliness of messages in cryptographic protocols. [8 marks]

Give a careful account of the main techniques used for ensuring timeliness, including any environmental assumptions. Under what circumstances is it proper to use time values when there is no clock synchronisation? [12 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 Discrete Mathematics

Let $(\mathbb{N}, \leqslant)$ be the natural numbers under the usual ordering. Assuming that $(\mathbb{N}, \leqslant)$ is well-ordered, prove that the Cartesian product $(\mathbb{N} \times \mathbb{N})$ is well-ordered under the derived lexicographical ordering. [6 marks]

State the *Principle of Well-Ordered Induction.* [3 marks]

Define inductively f: $(\mathbb{N} \times \mathbb{N}) \to \mathbb{N}$ as follows:

$$\begin{cases} f(0,y) & = \; y+1 \\ f(x+1,0) & = \; f(x,1) \\ f(x+1,y+1) & = \; f(x,f(x+1,y)) \end{cases}$$

Show that f is defined for all pairs (x, y). [2 marks]

Prove that for all $y \in \mathbb{N}$:

$$\begin{cases} f(2,y) & = \; 2y+3 \\ f(3,y) & = \; 2^{y+3}-3 \end{cases}$$

[9 marks]

11 Introduction to Functional Programming

Could functional programming techniques find commercial applications in the near future? Discuss in a well-structured and coherent essay. Support your reasoning with facts about functional programming and a variety of computer applications.

[20 marks]

12 The MIPS Instruction Set

Summarise *five* of the following six subjects concerning the MIPS R2000 in *no more than* 150 words each (diagrams are allowed and do not count to the word limit):

(*a*) instruction formats

(*b*) fetch–execute cycle

(*c*) delayed load

(*d*) delayed branch

(*e*) *byte sex* or *endianness*

(*f*) C function call interface under MIPS Ultrix, DEC's version of UNIX

[4 marks each]