

COMPUTER SCIENCE TRIPOS Part IA

Tuesday 6 June 1995 1.30 to 4.30

Paper 1

Answer **two** questions from Section A, and **one** question from **each** of sections B, C, D and E.

Submit the answers in six 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 Discrete Mathematics

Take the identity $(1 + \alpha)^r(1 + \alpha)^s = (1 + \alpha)^{r+s}$ and use generating functions to derive Vandermonde's convolution:

$$\sum_{k \geq 0} \binom{r}{k} \binom{s}{r-k} = \binom{r+s}{r}$$

Reach the same result by considering how many ways there are to choose r people from r men and s women. [10 marks]

2 Programming in Modula-3

What is meant by the *signature* of a procedure in Modula-3?

Describe the different parts of a signature and explain how arguments may be supplied when a procedure is invoked.

[10 marks]

3 Foundations of Computer Science

An ML list can be considered to be a set if it has no repeated elements, e.g., `[4,2,3]` is a set but `[4,2,4,3]` is not.

Write an ML function `intersect` to compute the set-theoretic intersection of two lists that satisfy this property of being a set. (The intersection of two sets is the set of elements that appear in both sets.) Your function must also satisfy conditions (a)–(c) below:

- (a) The result list has no repeated elements;
- (b) The number of `cons (::)` operations performed does not exceed the number of elements in the result list;
- (c) The elements of the result list appear in the same order as they do in the first argument list.

You may assume the existence of `[]` (`nil`) and `::` (`cons`). All other functions over lists must be defined by you. Little credit will be given for answers that do not satisfy conditions (a)–(c).

Write an ML function `subtract` that given two lists satisfying the property of being a set, returns a list consisting of those elements of the first list that do not appear in the second list. Your `subtract` function must satisfy conditions (a)–(c) above.

Write an ML function `union` that given two lists satisfying the property of being a set, returns a list consisting of those elements that appear in one or other or both of the lists. Your `union` function must satisfy conditions (a) and (b) above and (d) below:

- (d) Elements from the first argument list appear before any others in the result and in the same order as they appear in the first argument.

State the most general type of your `union` function.

[10 marks]

4 System Design

Describe a typical address map for a Von Neuman computer. Explain how a group of four adjacent bytes are typically used to represent

- (a) a positive integer;
- (b) a signed integer;
- (c) a floating point number.

[10 marks]

SECTION B

5 Foundations of Computer Science

Describe and compare the call-by-value, call-by-name, and call-by-need evaluation strategies for functional programming languages.

The ML function `butlast` removes the last element from a non-empty list:

```
exception Butlast;
fun butlast [] = raise Butlast
  | butlast [_] = []
  | butlast (x::xs) = x::(butlast xs);
```

Show how the evaluation of `butlast [[1,2], [], [3], [4,5]]` proceeds in ML.

Write an iterative version of `butlast` (i.e. one in which the recursive function calls are tail recursive). You may assume the existence of the `append (@)` function.

State with justification the time complexity of your function.

An ML data type of lazy lists can be defined by:

```
datatype 'a lazy_list = Nil | Cons of unit -> 'a * 'a lazy_list;
```

An ‘infinite’ list of increasing integers can be generated by the function `infinite` below:

```
fun infinite n = Cons (fn () => (n, infinite(n+1)));
```

Write a version of `butlast` for lazy lists which terminates when applied to an infinite lazy list such as `infinite(0)`.

Can an iterative version of this function be written that still terminates on infinite lazy lists? Explain your reasoning.

[20 marks]

6 Foundations of Computer Science

Consider the following ML declarations:

```
datatype bit = one | zero;
datatype cardinal = Cardinal of bit list;
exception Result_would_be_negative;
```

Define functions to add, subtract and multiply `cardinals`, viewing them as representations of integers stored as in binary with the least significant bit of the value first in the list. Thus for instance the number eleven has the value 1011 in binary and so would be represented (note the ordering of the bits) by the structure

```
Cardinal [one, one, zero, one]
```

Using Big-O notation state the time complexity of your add and multiply functions.

[20 marks]

SECTION C

7 Discrete Mathematics

Define a bipartite graph, a matching in a bipartite graph, and an augmenting path relative to a matching. Show that all matchings other than the largest have an augmenting path, and sketch how to look for one.

Indicate how to reduce a bipartite matching problem to a maximum network flow problem.

[20 marks]

8 Discrete Mathematics

If S_i , ($0 \leq i < k$) are finite sets, give without proof a formula for $|\bigcup_i S_i|$ in terms of the sizes of S_i and the intersections of sets of S_i 's.

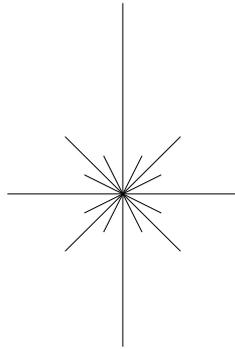
Use this to yield an explicit formula for the number of surjections from a set A to a set B .

[20 marks]

SECTION D

9 Programming in Modula-3

A Modula-3 program to draw compass roses is needed:



In this example the line pointing North is 5 units long, that pointing South is 4 units, East and West 3 units, NE, SE, SW and NW 2 units and NNE, ENE, ESE, SSE, SSW, WSW, WNW and NNW 1 unit. In general if the line pointing North is n units long there will be 2^{n-1} radial lines in the complete rose.

A library routine for drawing lines is provided — its signature is:

```
PROCEDURE Line(angle:REAL; length:CARDINAL)
```

Write two versions of the rose-drawing procedure, one using recursion and one using iteration. In the latter case it may be useful to write an auxilliary procedure that calculates the highest power of two that divides exactly into a given number. The only argument to the top-level rose-drawing procedures should be the length of the North line.

[20 marks]

10 Programming in Modula-3

The following Modula-3 program uses exceptions. What does it print on its standard output? Explain your answer.

```

MODULE Main;
IMPORT Fmt, IO;
EXCEPTION Bad;
VAR j: CARDINAL := 0;

PROCEDURE P (i: CARDINAL): CARDINAL RAISES {Bad} =
  BEGIN
    IF i = 0 THEN RAISE Bad END;
    RETURN j + 100 DIV i;
  END P;

PROCEDURE Q (i: CARDINAL): CARDINAL RAISES {Bad} =
  BEGIN
    TRY
      RETURN P (i);
    FINALLY
      j := j + i + 1;
    END;
  END Q;

VAR k: CARDINAL := 0;
BEGIN
  TRY
    k := Q (1); k := k + Q (0);  (**)
  EXCEPT
    Bad => IO.Put ("Bad Raised\n");
  END;
  IO.Put ("k = " & Fmt.Int (k) & "\n" &
    "j = " & Fmt.Int (j) & "\n");
END Main.

```

Rewrite procedures P, Q and the main program so that they perform the same calculations and have the same side-effects without using exceptions.

If the line marked (**) is replaced by the line:

```
k := Q (1) + Q (0);
```

what can you say about the output from the program?

[20 marks]

SECTION E

11 System Design

An application program in wide use contains a set of printable strings which are error messages and other helpful messages to the user. The program prints a selection of these messages each time it is run. Describe the route these messages have typically taken from the point where the programmer textually enters them, to the time they are displayed by the running application on an output device.

[20 marks]

12 System Design

Personal computers and workstations contain monitor and boot programs in non-volatile memory. Explain the functions that the code in this memory can provide, including the facilities that could be available at the following times:

- (a) power-on reset;
- (b) system self test;
- (c) kernel boot;
- (d) normal operation;
- (e) boot failure;
- (f) interactive debugging.

[20 marks]