UNIVERSITY OF LONDON
(University College London)

B.Sc. DEGREE 1992

COMPUTER SCIENCE B330: FUNCTIONAL PROGRAMMING

Answer THREE Questions.

The Use of Electronic Calculators: is NOT Permitted.

1. (a) What is a higher-order function? Give an example of a higher-order function definition in SML and how it might be applied. Illustrate the behaviour of your function by providing a hand-evaluation of the example application.

[5]

(b) Explain, with examples, the two terms "currying" and "partial application".

[4]

(c) Discuss the behaviour of SML's strong type system and how it may be used to aid the debugging of an SML program.

[11]

(d) Explain the difference between tail-recursive, accumulative-recursive and stack-recursive functions. Why are tail-recursive functions of special interest to implementors of functional languages?

[5]

[Total 25]

[TURN OVER]

2. (a) What is the most general type of each of the following three SML definitions?

```
fun f a b c = if (a b) then c else (b :: c)

fun map f nil = nil
|   map f (front :: rest) = (f front) :: map f rest

val doublemap = (map map)
```

[8]

(b) Given the following function definitions:

```
fun member (nil :''a list, _ :''a) = false :bool
|   member (front :: rest, item)
     = (item = front)
       orelse
       member (rest, item)

fun reduce f def nil
     = def
|   reduce f def (front :: rest)
     = f front (reduce f def rest)

fun accumulate f acc nil
     = acc
|   accumulate f acc (front :: rest)
     = accumulate f (f acc front) rest
```

Write two curried versions of `member` using `reduce` and `accumulate` respectively; compare and contrast these two definitions, paying particular attention to their types.

[17]

[Total 25]

[TURN OVER]

3. Compare and contrast the implementation of reference counting garbage collection with two other common garbage collection mechanisms. Your discussion should make special reference to the following points:

   memory overheads
   limited-width reference counts
   full-width reference counts
   Hughes's scheme for the collection of cycles

[Total 25]

4. Given the following definition of a sorted TREE:

   A TREE is sorted if either (i) it is empty, or (ii) it contains nodes (each of which has a value and a left sub-tree and a right sub-tree) where all the elements of a node's left sub-tree have a value that is less than the node value itself and all the elements of the node's right sub-tree have a value that is not less than the node's value.

   (a) Give an SML datatype definition for a TREE of polymorphic items.

   [3]

   (b) Explain why it would be unwise to attempt to represent a tree of arbitrary depth by a list of lists.

   [4]

   (c) Write a function to remove an element from a sorted tree and return a tree that is still sorted. Your function should have type

   `(''a -> ''a -> bool) -> ''a -> ''a TREE -> ''a TREE`

   where the first argument is a function which implements the ordering relationship and the second argument gives the value to be deleted. The result should be a TREE which is reasonably balanced.

   [13]

   (d) Give the function definition for a function `maptree` which provides similar functionality to `map` but works on a TREE rather than on a list.

   [5]

[Total 25]

[TURN OVER]

5. (a) The fixpoint combinator Y may be defined in SML as:

```
fun Y f = f (Y f)
```

Explain why the above definition cannot be used by an SML programmer. Show how Y may theoretically be used to transform recursive function definitions to non-recursive form. Do *not* discuss mutual recursion.

[6]

(b) Given the combinator definitions:

```
fun S f g x = f x (g x)

fun B f g x = f (g x)

fun W h x = h x x

val E = S (B W B) (B W B)
```

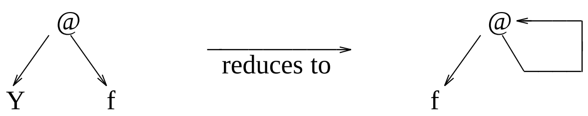
show that E is equivalent to the Y fixpoint combinator, by illustrating that successive transformations of the graph representation of `(E f)`, for any `f`, produces a graph equivalent to `(f (E f))`. At each stage, indicate which of the nodes in the graph are the same as the previous stage, which have been updated, and which are new.

[8]

(c) In fact the combinator E given above is not a valid SML definition. Explain why the definition of E gives rise to an error in SML.

[5]

(d) The Y combinator has the graph reduction rule:



Explain why the Y combinator is superior to the E combinator. Does Y have any disadvantages?

[6]

[Total 25]

[END OF PAPER]