Department of Computer Science
University College London

Cover Sheet for Examination Paper to be sat in May 2004

COMP3C11: Functional Programming

Time allowed 2.5 hours

Calculators are allowed

Answer THREE questions

Checked by First Examiner: Date:

Approved by External Examiner: Date:

Answer any THREE questions

Marks for each part of each question are indicated in square brackets.

Calculators are permitted

1. (a) What is a Higher Order Function? Give three examples to illustrate your answer.

[6 marks]

(b) Explain how a Curried Function can be partially applied. Give an example to illustrate your answer. [4 marks]

(c) Sometimes a function may appear to be applied to too many arguments. Explain how this can occur without any error – i.e. where the function and its application are correctly typed and there is no compile-time or run-time error. Give an example to illustrate your answer. [6 marks]

(d) You are given the following function definitions, where `number` is the name of a type synonym:

`one :: number` `one f x = f x`	`two :: number` `two f x = f (f x)`

`three :: number`
`three f x = f (f (f x))`
	`operator a b = h`
	`  where`
	`  h c x = a c (b c x)`

i) Give the most general type synonym definition for `number`. [6 marks]

ii) Give the polymorphic type of the function `operator`. [5 marks]

iii) Explain the operation of the function `operator` (in the context of the other definitions given above) by giving the evaluation steps of a simple application such as `(operator one two)`. Then explain in general terms what the function `operator` does (for example, could it be given a more descriptive name?). [6 marks]

[Total 33 Marks]

2. (a) Explain, in words, how the recursive type [*] could be defined. Your definition should itself be recursive. [3 marks]

(b) Either explain what the following function does, or explain what is wrong with it:

```
f [] = 0

f ('1' : rest) = 1 : (f rest)

f (2 : rest) = 2 : (f rest)

f (x : rest) = f rest
```

[3 marks]

(c) Give a Miranda algebraic type definition for a new type called `numchar` with two legal values, one of which contains a number and the other contains a character.

[3 marks]

(d) Give a Miranda algebraic type definition for a binary tree called `numchartree` that contains one data value (only) in each node, and where that data value can contain either a number or a character (but no other type).

[3 marks]

(e) Give the Miranda function definition, including its type, for a function called `insertnum` which inserts a number into a `numchartree` such that the tree contains only numbers and such that the numbers in the tree are *sorted*; i.e. for each node containing value **`v`**, all the numbers held in its right subtree are greater than or equal to **`v`**, and all the numbers held in its left subtree are less than **`v`**.

[9 marks]

(f) Give the Miranda function definition, including its type, for a function called `deletenum` which deletes a number from a `numchartree` such that the `numchartree` contains only numbers and such that the numbers in the tree are sorted (as defined in part (e) of this question) both before and after the deletion. Give definitions for any auxiliary functions that you require (except functions that exist in the Miranda standard environment).

[12 marks]
[Total 33 Marks]

3. (a) What is the most general type (i.e. using polymorphic types as generally as possible according to the context of the definitions) of the following three Miranda definitions?

```
f a b c = c, if (a b)
        = (b : c), otherwise


map f [] = []
map f (x : xs) = (f x) : (map f xs)


doublemap = map map
```

[10 marks]

(b) Given the following function definitions:

```
member :: [*] -> * -> bool
member [] y = False
member (x:xs) y = (x=y) \/ (member xs y)


foldr :: (*->**->**)->**->[*]->**
foldr f def [] = def
foldr f def (x:xs) = f x (foldr f def xs)


foldl :: (** -> * -> **) -> ** -> [*] -> **
foldl f acc [] = acc
foldl f acc (x:xs) = foldl f (f acc x) xs
```

(i) Write two new definitions for the function `member`, each of which has the same type `([*]->*->bool)` and functionality as the original function `member`, the first using `foldr` and the second using `foldl` in the function body. [15 marks]

(ii) Compare and contrast your definitions, paying particular attention to their types and to the ability to interchange foldr and foldl. [8 marks]

[Total 33 Marks]

4. You are told that a novel memory allocator preferentially allocates large blocks at the top end of available memory and small blocks at the bottom end of available memory. In all other respects the allocator is conventional. The allocator uses a single free list in a heap memory of size ***H*** bytes (see Figure 1).

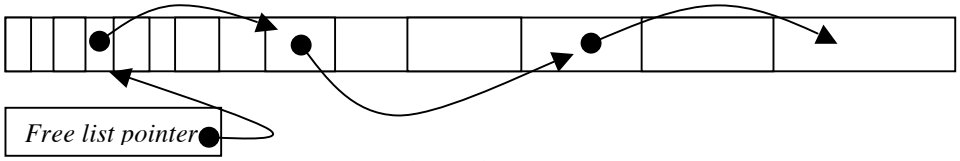

Figure 1

(a) How many blocks of memory (and of what size) are there on the free list when the program starts (before any allocation)? [5 marks]

(b) Give the details of how the very first allocation could be achieved if the requested block size is (i) small, and (ii) large. Give a diagram to illustrate the heap and the free list both before and after allocation in each case. [9 marks]

(c) If the free list is double-linked (see Figure 2), and each block's header contains its size, what is the size of a block header and what is the minimum size of an allocatable block? [5 marks]

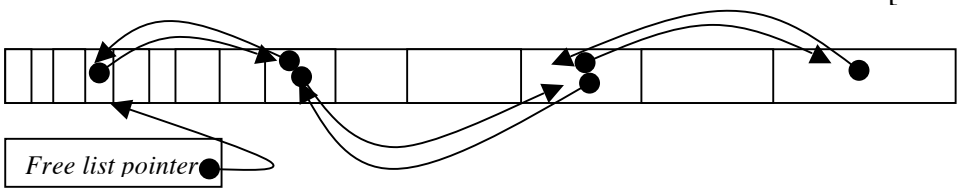

Figure 2

(d) You are told that the allocator maintains two pointers – one to the start of the double-linked free list and one to the end of the double-linked free list. Give an allocation algorithm using these two pointers which would, for multiple successive block allocation requests, place small blocks at the bottom end and large blocks at the top end of available memory. State any assumptions you make. [7 marks]

(e) Give a sequence of allocation requests and requests to free previously-allocated blocks that could defeat the intentions of this allocator, thereby causing small and large blocks to be intermingled. [7 marks]

[Total 33 Marks]

5. (a) State briefly what garbage collection is and why it is necessary for both Miranda and Java. Give a pictorial example of the creation of garbage in a graph reduction system. [8 marks]

(b) Describe briefly the operation of three different garbage-collection techniques and compare their advantages and disadvantages. [14 marks]

(c) What is fragmentation and how can it be cured? Your explanation should make reference to the three garbage collectors described in your answer to part (b) of this question. [11 marks]

[Total 33 Marks]

END OF PAPER