UNIVERSITY OF EDINBURGH

FACULTY OF SCIENCE AND ENGINEERING

COMPUTER SCIENCE THREE

PROGRAMMING METHODOLOGY

Saturday 3 June 2000

09:30 to 11:00

Examiners:

Prof. K. Turner (External)

Prof. G. Brebner (Chairman)

INSTRUCTIONS TO CANDIDATES

Answer any TWO questions.

All questions carry equal weight.

1. A manufacturer of office equipment is developing a new range of electronic office accessories for the busy executive. One of these is an electronic appointment diary. This will support the basic functions of a diary (entering, changing and deleting appointments and displaying the schedule for a day or week) as well as allowing other diary owners to enquire about free slots via a wireless link and to "pencil in" appointments. Statistical analysis of time spent in different kinds of activities will be supported, as well as backup and restoration of data. The leather carrying case has already been designed but all other details are open for discussion. The firm is determined to make the diary so useful that it will quickly become a "must have" accessory.

   (a) Identify at least three different rôles that a user of the diary may play and give a brief summary of their likely needs in each rôle. [*5 marks*]

   (b) Choose one of the rôles for users you outlined above and provide a brief description of an interface that would meet their needs. [*4 marks*]

   (c) Choose one of the functions in your chosen interface and provide a detailed specification of this function. [*5 marks*]

   (d) What attributes do you think would be important in the requirements specification of this system? Justify your choice and suggest how you might go about measuring them. [*4 marks*]

   (e) At the beginning of the project the project manager must decide on the lifecycle model to use. Outline two lifecycle models and compare their suitability for developing this system. [*7 marks*]

2. It is planned to introduce a new style of Pelican crossing near schools in an effort to improve road safety for children. The system incorporates sensors to detect vehicle location and movement and the presence of children in the crossing, plus a powerful laser capable of instantly vapourizing vehicles in the vicinity of the crossing if necessary to prevent accidents. You are responsible for the validation and verification of this system.

   A critical subsystem is described by the following signature:

```
signature ZAP =
  sig
    datatype Signal = GREENMAN | REDMAN
    type State
    val init : State
    val signal : State -> Signal
    val zap : State -> bool
    val update : State * bool * int * bool -> State
  end
```

   The state of the subsystem consists of the state of the laser (on or off) and the state of the crossing indicator. The next state is produced by the *update* function. This takes the current state as input together with the current value of the children-in-crossing sensor, the distance from the nearest vehicle to the crossing, and whether or not that vehicle is moving. The state of the crossing signal changes every 100 applications of the *update* function and the laser is triggered (*zap* applied to the current state is *true*) whenever there is a moving vehicle in the immediate vicinity of the crossing while the "little green man" is lit and children are on the crossing.

   (a) Explain the difference between verification and validation, and between static and dynamic methods. What is a test oracle? [*6 marks*]

   (b) Describe a technique for generating black-box tests and illustrate the approach using the above subsystem. Give an example test set. [*9 marks*]

   (c) Suppose you had access to the implementation code for this subsystem. Explain how you would go about performing structural testing. Contrast this with black-box testing. [*6 marks*]

   (d) Describe one technique for static verification and indicate what information you would require to apply the technique to this subsystem. [*4 marks*]

3. (a) Depth-first search for an element in a tree that satisfies a given property (expressed as a function from elements to bool) can be expressed as follows:

```
datatype 'a tree = Leaf | Node of 'a tree * 'a * 'a tree
fun find(p,Leaf) = NONE
  | find(p,Node(t,a,t')) = if p a then SOME a
                           else case find(p,t) of SOME a => SOME a
                                                | NONE => find(p,t')
```

which uses the built-in type

```
datatype 'a option = NONE | SOME of 'a
```

Prove that this function satisfies the property

```
forall (p,t,a) => (find(p,t) = SOME a) implies (p a andalso member(a,t))
```

where member is the function

```
fun member(a,Leaf) = false
  | member(a,Node(t,a',t')) =
        (a==a') orelse member(a,t) orelse member(a,t')
```

Hint: Recall the induction principle for trees of this kind:

$$\phi(\texttt{Leaf}) \land (\,\forall t_1, v, t_2. \phi(t_1) \land \phi(t_2) \Rightarrow \phi(\texttt{Node}(t_1, v, t_2))\,) \Rightarrow \forall t. \phi(t)$$

[9 marks]

(b) A *labelled directed graph* consists of a set of nodes, each containing a value (its *label*), and a set of edges, each connecting one node (its *source*) to another (its *target*). Trees like those in part (a) can be regarded as labelled directed graphs in which there is a distinguished node (the *root*) and exactly one path from the node to every other node.

Suppose that you are provided with the following types, constants and functions for (finite) labelled directed graphs:

```
type 'a node
type 'a edge
val source : 'a edge -> 'a node
val target : 'a edge -> 'a node
val label : 'a node -> 'a
type 'a graph
val empty : 'a graph
val addnode : 'a node * 'a graph -> 'a graph
val isnode : 'a node * 'a graph -> bool
  (* check if a given node is in a graph *)
val addedge : 'a edge * 'a graph -> 'a graph
```

Give EML axioms that specify the following additional functions:

```
val isedge : 'a edge * 'a graph -> bool
  (* check if a given edge is in a graph *)
val ispath : 'a node * 'a node * 'a graph -> bool
  (* check if there is a path from one node to another in a graph *)
val iscycle : 'a graph -> bool
  (* check if there is a cycle in a graph *)
```

[8 marks]

QUESTION CONTINUES ON NEXT PAGE

(c) Write EML specifications for depth-first search and breadth-first search from a designated start node in a labelled directed graph for a node with a label that satisfies a given property. Your specification should use the functions from part (b). Do not specify the search *algorithms* but rather the following *properties* that they must satisfy: breadth-first search will find a node having a label with the given property if there is a path from the start node to such a node; depth-first search will find a node having a label with the given property if there is a path from the start node to such a node and the graph contains no cycles. [*8 marks*]