CSM205 The Construction of Formal Specifications of Programs

Answer any 3 questions

Important:

- Unless otherwise specified all specifications are to be formally expressed using Z.
- Do not forget to introduce any new types you will need in your specification.
- Give a brief informal description of any mathematics written.

1. Consider the following informal description of the game Cover:

- Cover is a two-player game played on a circular table.
- Players take turns to place a one cent coin on the table such that the coin is completely on the table and does not overlap with previously placed coins.
- The first player unable to place a coin (with the restrictions just mentioned) loses the game.

(a) Specify the state of the game using Z.
(b) Specify the operations to place a coin and to check whether the game is over.
(c) A game is a sequence of game states which is possible according to the rules of the game. Specify the set of all possible games of Cover.
(d) A deterministic strategy is a function which, given the state of a game, returns a move to play. Specify the type of a deterministic strategy and define the function which, given a deterministic strategy \( f \), returns the set of all Cover games in which player one played according to strategy \( f \).
(e) A deterministic strategy is said to be total for player one if, whenever player one plays according to the strategy, whatever player two plays leaves the game in a state in which the strategy is determined. Specify the set of Cover strategies which are total for player one.

(f) A deterministic strategy \( f \) (total for player one) is said to be a winning strategy for player one if, for any game in which player one plays according to strategy \( f \), player one always wins, no matter what player two plays. Specify this set of strategies for \( \text{Cover} \).

2. The file allocation table (FAT) of an operating system is a database of entries, each of which has an associated filename and physical location on the disk. For simplicity no files may have the same filename (with no explicit concept of a directory structure).

\[
\text{[FAT - ENTRY, FILENAME, LOCATION]}
\]

| name : FAT - ENTRY \rightarrow FILENAME |
| loc : FAT - ENTRY \rightarrow LOCATION |

(a) An abstract specification of the FAT simply consists in storing a collection of entries. Specify the state \( \text{FAT}_a \) of such a table, its initialisation \( \text{Init}_a \) and the two operations \( \text{Add}_a \) and \( \text{Exists}_a \) which add an entry and check for the existence of a file with a certain filename respectively.

(b) A more concrete implementation stores the entries in a sequence. Specify the state \( \text{FAT}_s \), the initialisation \( \text{Init}_s \) and the operations \( \text{Add}_s \) and \( \text{Exists}_s \) which perform the equivalent operations as in part (a).

(c) State the proof obligations and give an outline of the proofs that the specification in (b) is a refinement of that in (a).

(d) Given a hash function \( \eta \) which maps filenames to natural numbers less than a constant \( N \), we can store the FAT entries by maintaining a number of lists of entries \( \text{fat}_0, \text{fat}_1, \ldots, \text{fat}_{N-1} \). \( \text{fat}_i \) is the list of entries whose filename is mapped by \( \eta \) to value \( i \). Formalise this implementation and give a retrieve schema which can be used to prove that it is a refinement of the specification given in (b).

3. (a) A graph over a set \( X \) consists of subset of \( X \) called the vertices of the graph and a set of pairs of vertices called edges of the graph.

Formally define the type (not schema type) \( \text{graph} \ X \) representing graphs over set \( X \).
(b) A path of a graph $G$ over set $X$ is a sequence of vertices of $G$ such that any two consequent vertices in the path are an edge of $G$.

Define the function which, given a graph $G$, returns the set of all paths in $G$.

(c) A museum consists of a set of interconnected rooms and a number of exhibits which are placed in the rooms. The museum also has an entrance hall and an exit (which are rooms).

Define the state of a museum using graphs.

(d) The museum requires terminals which, given a set of exhibits, outputs a path from the museum entrance to the exit which visits all the input exhibits. Furthermore, no shorter path which also visits the given exhibits is to be possible.

Specify this operation.

(e) Note that more than one such shortest path may be possible. Specify an alternative operation which, if more than one shortest path is possible, it outputs the one which visits most distinct rooms.

(f) State the proof obligations to show that the operation in question (d) is a refinement of the one in (e).

4. Consider the following definitions which will be used to express properties of timed systems:

$$\begin{align*}
SWITCH & ::= \text{ON} \mid \text{OFF} \\
\text{TIMED } X & ::= \text{seq } X
\end{align*}$$

A TIMED $X$ type stores a sequence of values of $X$, which will be interpreted as values of data sampled every time unit.

These types will be used to specify temporal safety requirements of a nuclear reactor water reservoir. The signature of the state of the reservoir is shown below:

$$\begin{align*}
\text{HeavyWater} \\
\text{waterleaver} : \text{TIMED } \mathbb{R} \\
\text{overflow} : \text{TIMED } \text{SWITCH} \\
\text{underflow} : \text{TIMED } \text{SWITCH} \\
\text{alarm} : \text{TIMED } \text{SWITCH} \\
\vdots
\end{align*}$$

Specify the following properties in Z:
(a) overflow is ON if and only if the water lever is higher than 10 units, while the underflow indicator is ON if and only if the water lever falls below 5.

(b) The alarm is turned on if there is an underflow.

(c) The alarm is also turned on if there has been an overflow during these past $\epsilon$ time units.

(d) Generalise the previous requirement by formally defining the meaning of $A \xrightarrow{n} B$, informally meaning that, if $A$ was ON for these past $n$ time units, then $B$ must also be ON.

(e) Once the alarm is turned on, it is to remain on for at least $\delta$ time units.

(f) Generalise the previous stability requirement by defining the meaning of $S(A, n)$, informally interpreted meaning that once $A$ becomes ON it must remain so for at least $n$ time units.

5. Consider the following specification of a sock drawer:

Socks can be either blue or red:

\[
\text{COLOUR ::= BLUE | RED}
\]

A sock drawer consists of a function which, given a colour returns the number of socks of that colour in the drawer:

\[
\text{SockDrawer} \\
\text{drawer} : \text{COLOUR} \rightarrow \mathbb{N}
\]

The owner can also hold a number of blue socks and a number of red socks:

\[
\text{Owner} \\
\text{owner} : \text{COLOUR} \rightarrow \mathbb{N}
\]

We start with 20 red socks and 20 blue socks in the drawer and none in the owner’s hands:

\[
\text{Start} \\
\text{Owner'} \\
\text{Drawer'} \\
\text{owner'} = \text{COLOUR} \times \{0\} \\
\text{drawer'} = \lambda c : \text{COLOUR} \cdot 20
\]
Pulling a random sock out of the drawer moves a blue or red sock from the drawer to the owner:

\[
\text{PullBlueSock} \quad \Delta \text{Owner} \quad \Delta \text{Drawer} \\
\text{owner}' = \text{owner} \oplus \{ \text{BLUE} \mapsto \text{owner} (\text{BLUE}) - 1 \} \\
\text{drawer}' = \text{drawer} \oplus \{ \text{BLUE} \mapsto \text{drawer} (\text{BLUE}) + 1 \}
\]

\[
\text{PullRedSock} \quad \Delta \text{Owner} \quad \Delta \text{Drawer} \\
\text{owner}' = \text{owner} \oplus \{ \text{RED} \mapsto \text{owner} (\text{RED}) - 1 \} \\
\text{drawer}' = \text{drawer} \oplus \{ \text{RED} \mapsto \text{drawer} (\text{RED}) + 1 \}
\]

\[\text{PullSock} \equiv \text{PullRedSock} \lor \text{PullBlueSock} \]

(a) Specify the property that just after starting, pulling two socks in succession can result in having a pair of the same colour.

(b) Specify the property that just after starting, pulling three socks in succession always results in having a pair of the same colour.

(c) Prove this last property.

(d) Calculate the schema precondition of \text{PullRedSock}.

(e) State the schema precondition of \text{PullSock}.

(f) State the proof obligations required to show that \text{PullRedSock} is a refinement of \text{PullSock}.