Summer Examinations 2016

CSY200816N

Module Title              Formal Specification of Software Systems
Level                    Five
Time Allowed             Two hours

Instructions to students:

• Enter your student number **not** your name on all answer books.
• Answer question 1 from **Section A** and two questions from **Section B**.
• All questions are equally weighted. Where a question has more than one part the division of marks is stated.
• You are not required to answer each question in a separate answer book.
• Unless explicitly required, there is no need to consider failure of preconditions in Z schemas which specify operations.
• Any symbolic expressions written by candidates should be accompanied by appropriate explanatory annotation in **plain English** otherwise marks may be withheld.

No. of Pages             8
No. of Questions         5
Section A

Answer question 1

Question 1

Provide answers to the following as succinctly as possible:

a. Define the following sets as comprehensions:
   
   i. \{0, 2, 4, 6, 8, 10, 12, 14, 16\} (2 marks)
   ii. \{5, 10, 15, 20\} (2 marks)

b. Enumerate the following sets:
   
   i. \{n: |n < 8 \& n^2\} (2 marks)
   ii. \{n: |n \& 4 \& n + 1\} \& \{1, 2, 3\} (2 marks)

c. Given the following declarations:

   \(x, y, z: |\); \(s: \text{STUDENT}; c: \text{COURSE}\);
   registered\_students: \& \text{STUDENT} ;
   on\_course: \& \text{STUDENT} \& \text{COURSE}

   Say whether the following notations are well-formed.

   i. \# \text{registered\_students} = x (1 mark)
   ii. \(s \sqsubset \text{registered\_students}\) (1 mark)
   iii. \(\text{on\_course}_\& = \text{on\_course} \& (s,c)\) (1 mark)
   iv. \((s,c) = \text{on\_course}\) (1 mark)
   v. \(s \sqsubset \text{on\_course} c\) (1 mark)
   vi. \# \text{registered\_students} \sqsubset \) (1 mark)

d. If \(A = \{x, y, z\}, B = \{54\}\) and \(C = \{9, 8, 7\}\) write out:

   i. \(A \times C\) (2 marks)
   ii. \(\Box A\) (2 marks)

e. Given the relations \(X = \{(1, 5), (3, 5), (4, 6)\}\) and \(Y = \{(5, 2), (6, 3)\}\)
   evaluate:

   i. \(X \sqcap Y\) (2 marks)
   ii. \(X \sqcap Y \& Y\) (2 marks)
   iii. \(X \sqcap Y \& X\) (2 marks)
   iv. \(Y \sqcap X\) (2 marks)

Question 1 continues overleaf
f. If \( S = \{ 1, 2, 3 \} \) and \( R = \{ (1, 1), (1, 5), (1, 9), (3, 1), (9, 5), (9, 12), (10, 11) \} \), find:

i. \( \{ 1 \} \circ R \)  
   2 marks

ii. \( \text{ran}\{ S \circ R \} \)  
    2 marks

iii. \( R\downarrow \)  
     2 marks

iv. \( R\leftrightarrow\{ S \} \)  
    2 marks

\[ \text{g. Using a truth table prove De Morgan’s Law, } (P \lor Q) \equiv \neg P \land \neg Q. \]  
(6 marks)

\[ \text{h. Given the following sets, } A = \{ 1, 2, 3 \}, B = \{ 1, 2, 3, 4 \}, C = \{ 3, 4, 5 \}, \text{ determine:} \]

i. \( A \propto B \)  
   2 marks

ii. \( A \propto B \propto C \)  
    2 marks

iii. \( A \setminus C \)  
    2 marks

iv. \( A \propto B \propto C \)  
   2 marks

v. \( (A \propto B) \setminus (B \propto C) \)  
   2 marks

Total: 50 marks
Section B

Answer two questions.

Question 2.

A system is to be developed that will record population data for a small country. For each resident of the country, the system must record the name and age of the person. It has been proposed that the system will be modelled using the total function has_age, where has_age: PERSON $\otimes$ $\mathbb{N}$.

A provisional state schema for the system has been developed below:

- Population_DB
  - has_age : PERSON $\rightarrow$ $\mathbb{N}$
  - living_residents : $\mathbb{P}$ PERSON
  - living_residents = dom has_age

1. Explain with the aid of a suitable diagram why has_age has been modelled as a total function, rather than a simple relation or a partial function. (5 marks)

2. Construct suitable schemas to represent the following essential system operations:

   a. Resident_Born – A schema to add a new resident to the system. (The system should be provided with the newly born person as the only operation input) (5 marks)

   b. Resident_Gets_Older – A schema to increment the age of a resident by one year. (The system should be provided with the person as the only operation input) (5 marks)

   c. Resident_Dies – A schema to remove a dead resident from the system. (The system should be provided with the dead person as the only operation input) (5 marks)

   d. Total_Residents – A schema to provide the total number of residents currently recorded within the system. (The system should provide a single numeric output for the total population) (5 marks)
Question 3.

An aircraft has a fixed capacity and it is required to record the number of people aboard the aircraft at any time. The aircraft seats are numbered and passengers enter the aircraft and are assigned seats sequentially (e.g. first person is allocated seat 1, the second person would be allocated seat 2, etc.)

The basic types needed for the specification of the system in Z, have been defined as:-

**[PERSON]**

-- The set of all possible persons who may, at some time, board the aircraft.

**capacity:**

-- The assigned capacity of the aircraft at any given time.

Assuming a sequence will be used to store the passengers and their allocated seats, a provisional state schema for the system has been defined as:-

```
Seq_Aircraft

passengers : seq PERSON

# passengers ≤ capacity
(∀ i, j : N | i ∈ dom passengers ∧ j ∈ dom passengers
• i ≠ j ⇒ passengers i ≠ passengers j)
```

1. Provide a suitable plain English translation of the two predicates recorded in the Seq_Aircraft schema. (4 marks)

2. Suggest an approach, using a specific type of sequence, which could significantly reduce the complexity of the originally drafted state schema. Include the revised schema in your answer. (4 marks)
3. Construct suitable schemas to represent the following essential system operations:

   a. The Initial_State schema of the system showing the aircraft as empty.  
      (4 marks)

   b. A Seq.Board schema to allow a passenger to board the aircraft, where the passenger is added to the next sequentially numbered available seat.  
      (5 marks)

   c. A Seq.Disembark schema that allows a passenger to leave the aircraft from any given seat.  
      (6 marks)

   d. A Seq.Number schema to provide the total number of passengers currently recorded within the system. (The system should provide a single numeric output for the total population)  
      (4 marks)

You do not have to include system output messages or consider error violations. Suitable English narrative should be included for all symbolic expression.

Total: 25 marks

Question 4.

A simple game is based upon the following:

A pond may contain any number of fish up to and including a specified maximum number. Conceptually, an angler is fishing the pond with a rod and line. Any fish caught by the angler must be placed in a net which is suspended in the pond.

Suppose the given set [FISH] represents all possible fish that may ever be in the pond and we declare a global:

\[ \text{maxFish} : \mathbb{N} \]

to represent the maximum number of fish that the pond can contain at any particular time. Assuming that we are only interested in the pond, the net and the relationship between them, create, based on two sets pond and net:

   a. A schema, Fishing, describing the state of the system.  
      (5 marks)

   b. A schema, Catch-One, for the operation whereby one fish is caught and placed in the net.  
      (5 marks)
c. A schema, **Free_Some**, for the operation whereby one or more fish are removed from the net and returned to the pond.  

(5 marks)

d. A schema, **Add_Some**, for the operation whereby a number of new fish are added to the pond.  

(6 marks)

e. A schema, **How_Many_Free**, for the operation which reports the number of fish currently free in the pond.  

(4 marks)

You do not have to include system output messages or consider error violations. Suitable English narrative should be included for all symbolic expression.

**Total: 25 marks**

**Question 5.**

A concert promoter sells tickets for a variety of musical events. Currently customers have to register with the company before tickets for any event can be purchased.

When the promoter secures a new event for promotion, the event is added to the list of promoted events. Once an event is on the promoted events list, it can then have tickets assigned to the event. Event tickets can then be purchased by registered customers. (Customers can buy more than one ticket for any given event, but obviously, once a ticket is sold, it may not be resold to another customer)

As a requirements engineer you have been asked to investigate and subsequently specify the following key system functionality:

- Register a customer
- Add an event
- Add a ticket for an event
- Allow a customer to purchase a ticket
- Allow a customer to cancel a ticket purchase
- Find out what tickets are assigned to an event
- Find out how many tickets are assigned to a given event
- Find out what tickets are available for an event (e.g. not purchased by a customer)
- Find out how many tickets are available for a given event
- Find out which customers have purchased tickets for a given event.

Once registered, customer records are never removed from the system, irrespective of whether a customer purchases tickets or not.

Events and their associated tickets are also never removed from the system, regardless of whether they are sold or not.
Based on the above specification, the following types have been defined for the system:

[PERSON] -- the set of all possible persons
[EVENT] -- the set of all possible events
[TICKET] -- the set of all possible tickets

customers: ∈ PERSON -- the set of registered customers
promoted_event: ∈ EVENT -- the set of promoted events
event_tickets: ∈ EVENT -- the set of event tickets for promoted events
purchased_tickets: PERSON → TICKET -- the tickets purchased by customers
available_tickets: EVENT → TICKET -- the tickets available for a given event

RESPONSE::= success | event_made_available | already_an_available_event | not_an_available_event | customer_added | etc

A draft state scheme for the system is also defined below:

<table>
<thead>
<tr>
<th>Events_Management_System</th>
</tr>
</thead>
<tbody>
<tr>
<td>customers: ∈ PERSON</td>
</tr>
<tr>
<td>promoted_event: ∈ EVENT</td>
</tr>
<tr>
<td>event_tickets: ∈ EVENT</td>
</tr>
<tr>
<td>purchased_tickets: PERSON → TICKET</td>
</tr>
<tr>
<td>available_tickets: EVENT → TICKET</td>
</tr>
</tbody>
</table>

1. Explain the meaning of the three predicates number 1-3 in the state schema above. (3 marks)

2. Construct suitable schemas to represent the following essential system operations:
   a. An Add_Ticket_For_An_Event schema, where a single ticket is added to the set of event_tickets and this ticket simultaneously added to the set of available_tickets for the event. (10 marks)
   b. A Purchase_A_Ticket schema, that allows a registered customer to purchase an available ticket for a valid event. (12 marks)

You do not have to include system output messages or consider error violations. Suitable English narrative should be included for all symbolic expression.

Total: 25 marks