Examination Period 3: 2016/17

CSY200817N

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

Instructions to students:
- Enter your student number not your name on all answer books.
- Answer three questions: one compulsory question from Section A and two from Section B.
- Where a question has more than one part the division of marks is stated.
- 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 all parts of Section A.

Question 1

Provide answers to the following as succinctly as possible:

a. If \( A = \{1, 2, 3, 4\} \), \( B = \{2, 4, 6, 8, 19\} \) and \( C = \{2, 3, 5, 7\} \) write down the sets:

i. \( A \cup B \cup C \)  
   (2 marks)

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

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

b. Enumerate the following sets:

i. \( \{n: n_1 | n < 8 \cdot n+2\} \)  
   (2 marks)

ii. \( \{n: n_1 | n \leq 4\} \cap \{1,2,3\} \cap \{n: n_1 | n = 2\} \)  
   (2 marks)

c. Given the declarations:

\( x, y, z : \mathbb{N}; a : \text{AUTHOR}; b : \text{BOOK}; \)
\( \text{on_shelves : P BOOK; novelists : P AUTHOR} \)

Say whether the following notations are well-formed

i. \( x > y \)  
   (1 mark)

ii. \( x \in \text{on_shelves} \)  
   (1 mark)

iii. \( a \in \text{on_shelves} \)  
   (1 mark)

iv. \( a \in \text{novelists} \)  
   (1 mark)

v. \( \text{on_shelves} \subseteq \text{novelists} \)  
   (1 mark)

vi. \( \text{on_shelves} \subseteq \text{Book} \)  
   (1 mark)

vii. \( \{\text{on_shelves}, \text{novelists}\} \)  
   (1 mark)

viii. \( \{a, b\} \)  
   (1 mark)

ix. \( \#x + y \)  
   (1 mark)

x. \( \#\text{on_shelves} = y \)  
   (1 mark)

d. If \( A = \{1, 2\} \), \( B = \{2, 1\} \) and \( C = \{9, 8, 7\} \) write out:

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

ii. \( P C \)  
   (2 marks)

iii. \( P A \)  
   (2 marks)

iv. \( A \times C \)  
   (2 marks)
e. Given the relations $X = \{(1, 1), (2, 4), (3, 9)\}$ and $Y = \{(1, 1), (4, 8), (9, 18)\}$ evaluate:

v. $X \circ Y$ (2 marks)
vi. $X \circ Y \circ Y^{-1}$ (2 marks)
vii. $Y \circ X$ (2 marks)
viii. $X^{-1} \circ Y^{-1}$ (2 marks)

f. If $S = \{1, 2\}$ and $R = \{(1, 1), (1, 5), (1, 10), (10, 11)\}$, find:

i. $\{1\} \triangleleft R$ (2 marks)
ii. $\text{ran}\{S \triangleleft R\}$ (2 marks)
iii. $R^{-1}$ (2 marks)
iv. $R\{\{S\}\}$ (2 marks)

g. Create a truth table to evaluate the following:

$P \text{ XOR } Q \iff (P \lor Q) \land \neg (P \land Q)$ (6 marks)

Total: 50 marks
Section B

Answer two out of four questions

Question 2

A stack is a data structure into which elements can be added ("pushed") and from which elements can be removed ("popped"). A stack is such that the next element to be popped is always the one most recently pushed (it is a last-in-first-out structure). If we suppose that the elements stored on the stack are of some generic type X, we could attempt to model a stack using a sequence.

```
[ X ]             any type
|bound : N         capacity of the stack

Stack[X]

s : seq X
#s ≤ bound
```

Based on the description and partial draft Z specification of the Stack above:

a. Create an Initial State schema called “int_state”. (3 marks)

b. Formulate a schema called “push” that pushes a new generic object x? into the stack. Ensure that you include appropriate English annotation. You need not consider precondition errors at this stage. (5 marks)

c. Create a schema called “pop” that pops a generic object x! off the stack. Ensure that you include appropriate English annotation. You need not consider precondition errors at this stage. (5 marks)

d. Formulate a schema called “push & pop” that simultaneously pushes and pops an object onto the stack resulting effectively in a null operation. Ensure that you include appropriate English annotation. You need not consider precondition errors at this stage. (7 marks)

e. Formulate a schema called “how_big” that returns the total number of objects currently in the stack. Ensure that you include appropriate English annotation. You need not consider precondition errors at this stage. (5 marks)

Total: 25 marks
Question 3

A warehouse holds stocks of various items carried by an organisation. A computer system records the level of all items carried. The computer system also records the withdrawal of items from stock and the delivery of stock. Occasionally a completely new item will be carried by the warehouse.

If [ITEM] is the set of all items that could possibly ever be carried by the warehouse then the Warehouse schema might describe the system state:

```
Warehouse
  carried : P ITEM
  level : ITEM → N
  dom level = carried
```

a. Write down an Initialise schema which shows that, initially, there are no items carried by the warehouse. (4 marks)

b. For a quantity of an item to be withdrawn, the item must be carried and there must be enough stock. Write down a Withdraw schema which withdraws a quantity q? of an item i? from the warehouse (ignore violation of preconditions). (7 marks)

c. If only deliveries for carried items are accepted, write down a Deliver schema for the operation which delivers a quantity q? of an item i? to the warehouse (ignore violation of preconditions). There is no upper limit on stock held. (7 marks)

d. A new item must not initially be carried and will initially have a stock level of zero. Write a schema CarryNewItem which adds a new item to the warehouse (ignore violation of preconditions). (7 marks)

Total: 25 marks
**Question 4**

As a key member of a software engineering company you have been asked to review the potential benefits and limitations of utilising formal methods for some, or all, of the software engineering projects that your company undertakes.

Write a short report, for the managing director of your company, to include the following essential information:

- **a.** Details of the different formal notations and CASE tool support currently available.  
  (5 marks)

- **b.** The software project characteristics that would make the project suitable, or not suitable, for specification development via formal methods.  
  (5 marks)

- **c.** The potential impact of using formal methods on project cost and development life cycle.  
  (5 marks)

- **d.** Highlight and debunk some of the myths associated with the use of formal methods.  
  (5 marks)

- **e.** The potential benefits of using formal methods over natural language specifications.  
  (5 marks)

**Total: 25 marks**
Question 5

You have been provided with the following partial requirements specification document for an Events Ticket Management system.

“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 once a ticket is sold, it may not be resold to another customer)

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 specification supplied, the following preliminary attempt at a formal specification of the system has been provided:

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

RESPONSE ::= success | already_a_customer | already_an_available_event | ticket_already_exist | not_a_promoted_event | ... etc.

-- the set of all system output messages

<table>
<thead>
<tr>
<th>Ticket_Sales</th>
</tr>
</thead>
</table>
| customers: \( \subseteq \) PERSON -- the set of registered customers
| promoted_event: \( \subseteq \) EVENT -- the set of promoted events
| events_tickets: \( \subseteq \) TICKET -- the set of event tickets for promoted events
| purchased_tickets: PERSON \( \leftrightarrow \) TICKET -- the tickets purchased by customers
| available_tickets: EVENT \( \leftrightarrow \) TICKET -- the tickets available for a given event

1. dom purchased_tickets \( \subseteq \) customers
2. dom available_tickets \( \subseteq \) promoted_event
3. ran available_tickets = events_tickets
```
a. Based on the system state schema **Ticket_Sales**, produce a suitable diagram to illustrate the relationship between each of the declared schema features. (4 marks)

b. Using plain English, clearly state the meaning of the three constraining predicates specified in the system state schema (labelled (1), (2) and (3)). (6 marks)

c. Based on the partial requirements specification document provided, make suggestions as to the most likely system state changing operations (e.g. $\Delta$**Ticket_Sales**) that would need to be specified as schemas in order to complete the specification. (5 marks)

d. Construct suitable Z schema definitions for each of the following two essential business enquiry operations (ensure that you include suitable English annotation for each):

i. **Tickets_Assigned**, an enquiry that returns the set of tickets assigned to a given event; (5 marks)

ii. **Tickets_Available**, an enquiry that returns the set of tickets that are available for a given event (i.e. tickets not currently purchased by a customer); (5 marks)

Total: 25 marks