### Summer Examinations 2015

**CSY200815N**

<table>
<thead>
<tr>
<th>Module Title</th>
<th>Formal Specifications of Software Systems</th>
</tr>
</thead>
<tbody>
<tr>
<td>Level</td>
<td>Five</td>
</tr>
<tr>
<td>Time Allowed</td>
<td>Two hours</td>
</tr>
</tbody>
</table>

**Instructions to students:**

- Enter your student number **not** your name on all answer books.
- Answer **three** out of **five** questions.
- 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 booklet.
- 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.

<table>
<thead>
<tr>
<th>No. of Pages</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. of Questions</td>
<td>5</td>
</tr>
</tbody>
</table>

Page 1 of 9
Answer three out of five questions.

Question 1

Consider the system description and a partial draft Z specification Programmer Project Allocation system, as provided below.

Each programmer employed within the organisation is uniquely identified by a programmer-number. The name of any programmer may be derived from their programmer-number. Each project is, similarly, uniquely identified by a project-code from which the project name may be derived.

Agreed basic types for the system are:

\[ [\text{PROG\_NUM}] \quad \text{the set of all possible programmer numbers} \]
\[ [\text{PROG\_NAME}] \quad \text{the set of all possible programmer names} \]
\[ [\text{PROJ\_CODE}] \quad \text{the set of all possible project codes} \]
\[ [\text{PROJ\_NAME}] \quad \text{the set of all possible project names} \]

and the state schema is Project_Programmer where:

\[
\begin{align*}
\text{Project\_Programmer} & : \text{programmer} : [\text{PROG\_NUM} \rightarrow [\text{PROG\_NAME}]
\text{project} : [\text{PROJ\_CODE} \rightarrow [\text{PROJ\_NAME}]
\text{assignment} : [\text{PROG\_NUM} \rightarrow [\text{PROJ\_CODE}]
\end{align*}
\]

\[
\begin{align*}
\text{dom} \text{assignment} & \subseteq \text{dom} \text{programmer} \\
\text{ran} \text{assignment} & \subseteq \text{dom} \text{project}
\end{align*}
\]

(Note that the symbol \( \rightarrow \) indicates an injective (1:1) partial function)

The admission of a new programmer to the organisation may be expressed by the following schema:

\[
\begin{align*}
\text{Add\_Prog} & : \text{Project\_Programmer}
\Delta \text{Project\_Programmer} \\
\text{prognames}' : [\text{PROJ\_NAME}]
\text{programns}' : [\text{PROG\_NUM}]
\text{dom} \text{programmer} \cap \{ \text{programn}' \} = \{ \}
\text{programn}' = \text{programn} \uplus \{(\text{programn}', \text{prognames}')\}
\text{project}' = \text{project}
\text{assignment}' = \text{assignment}
\end{align*}
\]
a. Use **plain** English to explain why the three sets programmer, project and assignment in the declaration part of the Project_Programmer schema have been modelled as they have in the state schema.  

(9 marks)

b. Use **plain** English to explain the combined meaning of the two conjuncts in the **predicate** part of the Project_Programmer schema.  

(4 marks)

c. For the Add_Prog schema above,

i. Use **plain** English to explain the meaning of the first conjunct in the **predicate** part of the schema.  

(2 marks)

ii. Write a different, logically equivalent, symbolic form for this first conjunct.  

(2 marks)

d. Use **plain** English to explain the second conjunct in the **predicate** part of the Add_Prog schema and suggest another way of expressing the same thing without using $\circ$. Why do you think that $\circ$ was used?  

(4 marks)

e. Assuming that FEEDBACK is a free-type where:

```
FEEDBACK ::= OK | Unknown Programmer | Unknown Project | Already Assigned
```

Create a Z schema Total_Add_Programmer to define the total operation which caters for the possible violation of the precondition in the schema Add_Prog.  

(12 marks)

Total: 33 marks
Question 2

A college provides a multi-user computer system for its students and staff. All staff and students must register with the college’s IT Services unit before they are allowed access to the computer system. To use the system, each registered user must log-in. At any given time a registered user will either be logged-in or not logged-in (it is not possible for a user to be logged-in more than once concurrently). A registered user can only log-in if there is an available machine, as the college has a finite number of machines.

A partial draft Z specification of the proposed system has been provided below:

\[
\text{[PERSON]} \rightarrow \text{the set of all uniquely identifiable persons}
\]

\[
\text{OUTPUT: } := \text{success} \mid \text{already_a_user} \mid \text{not_a_registered_user} \\
\text{already_logged_in} \mid \text{no_capacity}
\]

\[\rightarrow \text{the set of all system output messages}\]

\[
\text{Computer_Management} \\
\begin{array}{ll}
\text{users} : \mathbb{P} \text{PERSON} & (1) \\
\text{logged_in} : \mathbb{P} \text{PERSON} & (2) \\
\text{capacity} : \mathbb{N} \\
\text{capacity} < 0 \\
\end{array}
\]

\[
\text{logged_in} \subseteq \text{users} & (3) \\
\#\text{logged_in} \leq \text{capacity} & (4)
\]

(1) \text{the set of all registered users} \\
(2) \text{the set of users who are currently logged-in} \\
(3) \text{you must be a registered user to be logged-in} \\
(4) \text{the number of users logged-in can never exceed capacity}

Based on the system state schema provided, define suitable Z schemas for each of the following operations (on page 5). Ensure that you conform to all notation standards and include suitable English annotation for all parts of the schemas you produce.
a. The initial state of the system schema called **Int_State.** (3 marks)

b. A schema used to register a new user called **Reg_New_User.**
   You should include a suitable system output message but need not include precondition violation within this schema. (6 marks)

c. A log-on schema for an existing registered user called **Log_On.** You should include a suitable system output message but need not include precondition violation within this schema. (7 marks)

d. A schema to deal with registration precondition violation called **Reg_Error.** (4 marks)

e. A schema to deal with log-on precondition violation called **Log_On_Error.** (9 marks)

f. A total schema to represent the complete register a new user operation called **Total_Register_User.** (2 marks)

g. A total schema to represent the complete log-on operation called **Total_Log_On.** (2 marks)

**Total: 33 marks**
Question 3

J.P. Bowen and M.G. Hinchey are credited for writing one of the seminal papers on the application of formal methods to software systems, called the ‘Ten Commandments of Formal Methods’. In this paper they addressed several issues including:

a. The type of software projects suitable for formalisation.  
   (5 marks)

b. The implications of using Formal Methods on software development costs and project life-cycle.  
   (6 marks)

c. Why traditional software development methods should not be abandoned completely in favour of formal methods.  
   (5 marks)

d. Why the formal methods shouldn’t be used as an excuse to accept compromised software testing strategies.  
   (6 marks)

e. The potential benefits of reusable specification elements, for use within subsequent software projects.  
   (5 marks)

f. How the problems evident within natural language specifications can be overcome when formal methods are deployed.  
   (6 marks)

Write a short report expanding on each of the ideas outlined above (a-f). Ensure that you provide evidence of your understanding of each of the statements and where possible provide examples of actual software developments that have exemplified the points being made within your report.

Total: 33 marks
Question 4

As part of a larger system, a data processing utility stores numeric data in three different ways:

1. Integers stored in the sequence they are received by the tool.
2. Unique Integers stored in a non-sequenced collection.
3. Repeated Integers stored in a non-sequenced collection.

Each of the three stores has a fixed storage capacity.

To model this part of the system the following axiomatic schema and system state schema have been developed:

```
container_1_capacity : N
container_2_capacity : N
container_3_capacity : N

container_1_capacity > 0
container_2_capacity > 0
container_3_capacity > 0

Data_Store
container_1 : seq N
container_2 : P N
container_3 : bag N

#container_1 ≤ container_1_capacity
#container_2 ≤ container_2_capacity
size container_3 ≤ container_3Capacity

RESPONSE:: = success|not_an_integer | already_in_set | at_capacity
| multiple_errors  -- The set of system responses defined thus far.
```

a. The existing model stores integer data values in three distinct ways, using a sequence, a set and a bag. Outline how the structures of each of these data stores functions and highlight the fundamental differences of each. Illustrate your answers by providing suitable examples of where each of the structures would commonly be used as part of larger scale system specification.

(9 marks)
b. Based on the description and partial draft Z specification of the data processing utility store provided above:

i. Create a schema called **Add_To_End_Of_Sequence** that adds valid integers to the end of the sequence container (`container_1`). Ensure that you include a suitable operation output message (taken from the freetype RESPONSE) and appropriate English annotation. You need not consider precondition errors at this stage.  
   (5 marks)

ii. Define a schema called **Add_To_Set** that adds valid integers to the set container (`container_2`). Ensure that you include a suitable operation output message (taken from the freetype RESPONSE) and appropriate English annotation. You need not consider precondition errors at this stage.  
   (5 marks)

iii. Formulate a schema called **Add_To_Bag** that adds valid integers to the bag container (`container_3`). Ensure that you include a suitable operation output message (taken from the freetype RESPONSE) and appropriate English annotation. You need not consider precondition errors at this stage.  
   (5 marks)

iv. For the **Add_To_End_Of_Sequence** schema you developed for question 3(a) above, create an additional schema called **Add_To_End_Of_Sequence_Error**, that deals with precondition violations and provides suitable output error messages (taken from the freetype RESPONSE) and appropriate English annotation.  
   (7 marks)

v. Using the schemas you developed in question 3(a) and 3(d) create a **Total_Add_To_End_Of_Sequence** schema.  
   (2 marks)

**Total: 33 marks**
Question 5

Suppose an organisation has an internal telephone network. A database is created which holds details of people and associated extension number(s). Some people may have more than one extension and some extensions may be shared by more than one person; so, we might have:

[PERSON] the set of all people who might part of the database at some time.

[PHONE] the set of all possible telephone extension numbers that might be allocated at some time.

If we use telephones to act as an identifier expressing the relation between people and their extension numbers, we have (from the comments above):

\[ \text{telephones} : \text{PERSON} \leftrightarrow \text{PHONE} \]

The state of the database might be specified by:

\[
\begin{array}{l}
\text{Phone_DB} \\
\text{members : P PERSON} \quad \text{-- set of all people in the organization} \\
\text{telephones : PERSON} \leftrightarrow \text{PHONE} \\
\text{dom telephones} \subseteq \text{members} \\
\end{array}
\]

Based on the specification provided thus far, attempt to complete the following essential specification using standard Z Notation (Do not consider precondition violations at this stage):

a. An Add_Entry schema that allows a new Phone_DB entry for a given member, for a new phone extension entry.

   (9 marks)

b. A Remove_Entry schema that allows an existing Phone_DB entry for a given member to have the telephone extension mapping removed.

   (9 marks)

c. Use schema composition to show how the operation of someone changing their extension can be composed from the Remove_Entry and Add_Entry operations.

   (15 marks)

Total: 33 marks

End of Paper