SOEN331/W: Introduction to Formal Methods for Software Engineering

Assignment 4 on Z Specifications

Instructor: Dr. Ormandjieva

In groups of **2** students
Weight: 5% of the overall grade.
Date posted: April 4th, 2017.
Date due: April 11th by midnight.

Submission instructions: submit electronically (EAS) as "Theory Assignment 4", one submission per team. List the names and the IDs of the team members on the first page of your submission.

Part 1. Consider the following specification for a simple Invoicing Orders System:

- *OrderID* is a set of order identifiers. Identifiers are unique.
- The status *OrderState* of an order may be "pending" or (alternatively) "invoiced".
- *Product* is the set of products which can potentially be held in stock
- *Stock* schema contains the quantities of each of the products in stock

Stock

$$stock : \text{bag}\, Product$$

- Here we define *Order* as a bag of products and assume that the customer decides to allow orders of one or more products for extra flexibility, but not empty orders (i.e., an order for no products).

$$Order == \{order : \text{bag}\, Product \mid order \neq \varnothing\}$$

- The status of orders can be modelled as a function from an identifying *OrderId* to their state (pending or invoiced). State components *orderStatus* and orders are packaged into an *OrderInvoices* schema with appropriate type information:

$$
\begin{array}{|l}
\textit{OrderInvoices} \\
\hline
orders : OrderId \pfun Order \\
orderStatus : OrderId \pfun OrderState \\
\hline
\dom orders = \dom orderStatus \\
\hline
\end{array}
$$

- Abstract state space *State* includes the *Stock* and *OrderInvoices* together with a further state component *newids*. It is defines as follows:

$$
\begin{array}{|l}
\textit{State} \\
\hline
Stock \\
OrderInvoices \\
newids : \mathbb{P}\, OrderId \\
\hline
\dom orders \cap newids = \varnothing \\
\hline
\end{array}
$$

- We define a set of possible reports from operations
  Report ::= order not pending | not enough stock | no more ids

- Constraints that apply for all operations on the system:
  - If any new identifiers from the set *newids* are used for orders (and hence their status) these are no longer available for use by any subsequent operation. Thus they are removed from the set of new identifiers.
  - The above constraint is specified using $\Delta State$ schema convention as shown below:

$$
\begin{array}{|l}
\Delta \textit{State} \\
\hline
State \\
State' \\
\hline
newids' = newids \setminus \dom orders' \\
\hline
\end{array}
$$

Part 2. Your assignment:

Write the following operations using Z schemas.

1. **Initialization operation *InitState.*** Initialize *stock*, *orders* to empty sets and *newids* to *OrderId* set.

2. ***NewOrder* Operation.** This operation handles new orders.

   Preconditions for ***NewOrder*** operation are:
   - An *order*? must be provided as an input and a valid new identifier *id!* (from the set *newids*) is output by the operation.

Postconditions for ***NewOrder*** operation are:

- A new order is added to the set of orders with order ID *id!* and products *order?*
- the status of the newly added order *order*? with order ID *id!* has to be *pending*
- The stock itself is unaffected by the operation above.

3. ***InvoiceOrder*** **Operation.** This operation invoices pending orders.

   These are the ***InvoiceOrder*** operation's **preconditions** that must be satisfied for an order to be successfully invoiced:

   - An input *id?* is required to specify which order is to be invoiced. *id?* has to be a valid existing order identifier.
   - there must be enough stock available to fulfil the order (hint: use a Z *bag* operator to specify this requirement), and
   - the status must be pending.

Postconditions for ***InvoiceOrder*** operation are:

- the ordered items are not in stock after the operation (hint: use a Z *bag* operator to specify this requirement).
- The status of the order with *id?* is updated to "*invoiced*" (use overriding Z operator)
- The orders themselves are unaffected by the operation above.

4. For **error case *InvoiceError*** where the order state is not pending, assume the state is not to change. Return an error report message *rep!* "*order not pending"*

5. For **error case S*tockError*** where there is no enough stock available for the order, assume the state is not to change. Return an error report rep! "*not enough stock"*

6. Specify **total operation *InvoiceOrderOp***

Instructions:

1) Use Z/EVES tool to write the Z specifications for the Invoicing Orders System.
2) Your solution should include:
   - Z types and schemas listed in Part 1 of this assignment: *Stock, Order, OrderInvoices, State, Report, OrderState*
   - Your Z schemas *InitState, NewOrder, InvoiceOrder, InvoiceError, StockError, InvoiceOrderOp*
3) Submit (zipped) your solution file .zed and a pdf file of your specifications electronically (EAS) as "Theory Assignment 4", one submission per team. List the names and the IDs of the team members on the first page of your submission.