COMP3601 Software Specification (2004/1)
Assignment (20%)

School of Information Technology and Electrical Engineering*,
The University of Queensland

Due 9am Tuesday 25th May 2004

Specifying a system for managing items for hire in a hire shop

The system to be specified for this assignment is for managing items for hire. You need to keep track of the customers, what items (and number of items) each customer has on hire and the return date for each item.
The Hire machine contains the following deferred sets.

SETS *ITEMS* **;** *CUSTOMERS*
where
ITEMS represents the set of all possible items that can be hired;
CUSTOMERS represents the set of all possible customers.

We also have several constants, one of which is *maxitems* and is the maximum number of items a customer can hire at any given time. The other constants will be discussed later.
The state variables (and their types) are given below.

VARIABLES
today , *stock* , *hasHired*
INVARIANT
$today \in \mathbb{N} \land$
$stock \in ITEMS \pfun \mathbb{N} \land$
$hasHired \in (CUSTOMERS \times \mathbb{N}) \pfun (\mathsf{dom} (stock) \pfun \mathbb{N})$

today is the current day.
stock keeps track of the number of instances of each item owned by the store.
hasHired keeps track of the items currently on hire. It is a function that maps customer, return date pairs to functions from items to quantities.
So, for example, the *hasHired* function
$\{(Fred \mapsto 124) \mapsto \{ladder \mapsto 3,\ spraygun \mapsto 1\},\ (Fred \mapsto 130) \mapsto \{coversheet \mapsto 4\},\ (Mary \mapsto 126) \mapsto \{ladder \mapsto 2\}\}$ }
says that *Fred* has 3 ladders and one spraygun on hire to be returned on day 124 and 4 coversheets to be returned on day 130. *Mary* has 2 ladders to be returned on day 126.

You will find that you will need to count the number of instances of items on hire and to do this you will need to use **SIGMA** ($\sum$).
The expression $\sum\ zz\ .\ (P \mid E)$ calculates the value of E for each zz that satisfies P and adds them together.
So, for example,

$\sum\ zz\ .\ (zz \in \mathbb{N} \wedge zz < 5 \mid zz) = 1 + 2 + 3 + 4 = 10$

$\sum\ zz\ .\ (zz \in \mathbb{N} \wedge zz < 5 \mid zz * zz) = 1 + 4 + 9 + 16 = 30$

$\sum\ zz\ .\ (zz \in \mathbb{N} \wedge false \mid \mathrm{zz}) = 0$

Your task is to add definitions for some constants, add constraints to the invariant and define some operations. **NOTE:** For the operations you should take care to add preconditions so that the operations preserve the invariant.
To aid in the definition of the constants we define the following auxilary (constant) function that adds all the instances of all the items in an item bag (a function of type $ITEMS \pfun \mathbb{N}_1$ that records the number of instances of items).

$bag\_total \in (ITEMS \pfun \mathbb{N}_1) \fun \mathbb{N} \wedge$
$\forall\ bb\ .\ (\ bb \in ITEMS \pfun \mathbb{N}_1 \Rightarrow$
$\quad bag\_total\ (\ bb\ ) = \sum\ zz\ .\ (\ zz \in \mathsf{dom}\ (\ bb\ )\ \mid\ bb\ (\ zz\ )\ )\ )$

To aid in defining the operations you need to define the following three constants:

1. $customer\_items \in (\ (\ (\ CUSTOMERS \times \mathbb{N}\ ) \pfun (\ ITEMS \pfun \mathbb{N}_1\ )\ ) \times CUSTOMERS\ ) \fun \mathbb{N}$

   that takes a hiring function and a customer and calculates the total number of item instances hired to the customer.

2. $item\_total \in (\ (\ (\ CUSTOMERS \times \mathbb{N}\ ) \pfun (\ ITEMS \pfun \mathbb{N}_1\ )\ ) \times ITEMS\ ) \fun \mathbb{N}$

   that takes a hiring function and an item and calculates the total number of instances of that item currently on hire.

3. $bag\_union \in (\ ITEMS \pfun \mathbb{N}_1\ ) \times (\ ITEMS \pfun \mathbb{N}_1\ ) \fun (\ ITEMS \pfun \mathbb{N}_1\ )$

   that takes two item bags and merges them into a single bag.

   For example,
   $bag\_union(\{\ ladder \mapsto 2, spraygun \mapsto 1\}, \{\ coversheet \mapsto 3, ladder \mapsto 3\ \})$
   $= \{\ ladder \mapsto 5, spraygun \mapsto 1, coversheet \mapsto 3\}$

Invariant: The following constraints need to be added to the invariant. Note that the above constants may be helpful.

- No customer can have more than *maxitems* instances of items on hire. (In the example earlier *Fred* has 8 items on hire and *Mary* has 2.)
- The total number of instances of a given item on hire can not exceed the number owned by the hire shop.

The following operations are required. Note that you should choose a precondition for each operation that preserves the invariant after the operation.

- **hire** that takes an item, a customer, the number of instances of the item, and the duration of the hire and hires the supplied number of instances of the item to the supplied customer for the supplied duration.
- **overdue** that outputs all the overdue items as a subset of the *hasHired* function where the return date is less than *today*.

Doing the assignment:

Download the tar'ed machine `Hire.tar.gz`.
Untar this in the directory where you keep your B machines.
Add your name and student number to the comment section at the beginning.
Edit this machine to produce your answer for the assignment. To aid in marking please do not move things around.
Use the BTool to check the types and proof obligations.
When you are ready to submit your assignment you need to generate a postscript version. To do this do the following

```
cd TEX
dvips -o Hire.mch.ps Hire.mch.dvi
gv Hire.mch.ps       (to check if it looks OK)
```

NOTE: you need to generate the dvi file using Document Mark-Up in the BTool.
The postscript file you generated will be part of the submission.

See over for submission details.

Submitting your assignment:

You must submit your completed assignment electronically through the website:
`http://submit.itee.uq.edu.au`
Please read
`http://submit.itee.uq.edu.au/student-guide.pdf`
for information on using electronic submission.

You may submit your assignment multiple times before the deadline - only the last submission will be marked.
A penalty of 20% per day will apply for late assignments.

You are required to submit 2 files:
`Hire.mch.ps`
`Hire.mch`

You MUST use these names for your files - the first one is the postscript you generated using the instructions given earlier. The second is the machine source.
NOTE: Make sure you get the `Hire.mch` in the CFG directory and NOT from the SRC directory. We will use this file in the BTool if we find problems with your postscript.

Marking Criteria (20 marks total)

customer_items definition	4 marks
item_total definition	4 marks
bag_union definition	4 mark
Invariant	3 marks
hire operation	3 marks
overdue operation	2 marks

Plagiarism and Collusion

The school policies on plagiarism and collusion apply and full details are available at http://www.itee.uq.edu.au/about/student-misconduct.jsp.