T-79.5303 Safety Critical Systems

Home Assignment 2006

Teemu Tynjälä

April 6, 2006

Ilkka’s questions

(Neil Storey, Chapter 15.2 An explosive chemical plant - please email herttua@eurolock.org if you need a copy of the report)

1. List potential hazards of the basic nitrator process

2. Explain functions of safety components and their relation to hazards

Teemu’s questions - 1

A machine **Access.mch** is provided on the course web page with this assignment.

Introduce into the invariant of *Access* the condition that no user should have access to more than 6 printers. Which operations are not consistent with your revised invariant? Strengthen the preconditions of any such operations.

In your answer, submit the new invariant clause, and the changed operations.

You may find the inconsistent operations simply by observations. If you want to use ProB to help, you can change the set sizes in the **Preferences -> Animation Preferences** menu. Pick the "Size of unspecified sets in SETS section" and experiment with different values. This helps you to find the invariant violations.

Teemu Tynjälä

Teemu’s questions - 2

Still dealing with the Access machine..

Add an operation **exchange** to Access, which switches the printers associated with two users. Submit the B code for the new operation (it’s a good idea to simulate the whole machine with this change, but only the new operation code is required to be submitted).

Teemu’s questions - 3

Still dealing with the Access machine..

Add an operation **maintenance** to Access, which removes one printer from the relation *access*, and associates all of the corresponding users with an alternative printer. Submit the B code for the new operation.

Teemu Tynjälä

Teemus's questions - 4

A car park has 640 parking spaces. Give an abstract machine which specifies a system to control cars entering the car park. It should keep track of the number of cars currently in the car park, and should provide three operations:

- **enter**, which records the entry of a new car. This should occur only when the car park is not full;
- **leave**, which records the exit of a car from the car park;
- $nn \longleftarrow$ **query**, which outputs the number of cars currently in the car park.

Submit the complete B machine for this question. It's a good idea to model check your answer for 1000 steps to make sure it's OK.

Teemu Tynjälä

Teemu’s questions - 5

What is the weakest precondition required for the following statement to establish a state in which $category \neq medium$?

```
CASE (i + 2)/3 OF
EITHER 0 THEN category := small
OR 1 THEN category := small
OR 2 THEN category := medium
OR 3 THEN category := medium
ELSE category := large
END
```

Teemu's questions - 6

What does the following statement achieve?

ANY a **WHERE** $a \in \mathbb{N}_1 \wedge a \leq 5$ **THEN** $total := total + a$ **END**

What is the weakest precondition required for it to guarantee the postcondition $(total > 8)$?

($\mathbb{N}_1$ set means the natural numbers starting with 1)

Teemu's questions - 7

What does the following statement achieve?

ANY s **WHERE** $s \subseteq 1..N \wedge card(s) \leq 3$ **THEN** $myset := s$ **END**

What is the weakest precondition required to guarantee the postcondition that the sum of the elements in the set $myset$ is less than 40? I.e. $\Sigma i.\ (i \in myset | i) < 40$

Teemu's questions - 8

In **ProB/Machines** directory, there is one called **Priorityqueue.mch**. This may be used as an inspiration to solve the following:

a) Create a MonotonicQueue.mch machine that allows only increasing sequences. (i.e. each element of a sequence is greater than the one before it). Think what the correct invariant and preconditions for operations should be. Hint: the insert operation for your answer should be simpler than in PriorityQueue..

b) Extend your machine in such a way that you have two variables, queue and rev_queue, which contain the same items in the opposite order. I.e. appending an item to the queue means that the same item has to be prepended to rev_queue

c) Write the invariant saying that the sum $queue(xx) + rev\_queue(xx)$ stays constant for all indexes xx. Hint: the invariant in PriorityQueue.mch is a very good example for this...

Submit a single B machine that contains answers to all 3 parts. If you don't get some parts, tell in your answer which part(s) you solved.

Teemu Tynjälä

Due Dates + Submission format

- You have until midnight on May 5 to return the assignments.

- Make an electronic submission (*.doc, *.ps or *.pdf – B machines sent separately as text files) and mail your answers to Ilkka and myself to addresses teemu.tynjala@nokia.com and herttua@eurolock.org