ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

J-R. Abrial, T.S. Hoang and M. Schmalz

DEPLOY Block Course

(ETH Zurich 9-11 April, 2008)

Exercise Sheet 1: Requirement Document and Introduction to the Rodin Platform

Description of Work

The first aim of this exercise is to come up with a clean requirement document for a simple access control system of a building. You are given the informal descriptions of the system as follows. It is quite clumsy and poorly written (on purpose).

The second aim of this exercise is to be familiar with the Modeling interface and the Interactive Proving interface of the Rodin Platform. The example that we choose to use in this exercise is the development of search programs in the slides "A Summary of the Event-B Modeling Notation".

The last exercise to develop an algorithm to find the maximum number within an array is intended to be a challenge exercise.

An Access Control System

Here is an informal description of the system.

> The system consists of the following parts:
>
> - a set of rooms,
> - among them, a special room called "hallway".
> - Rooms are connected by doors.
> - The hallway is connected to all rooms.
> - A person will be authorized to be in certain rooms.

Out of the above description, you need to produce a clean requirement document. You are suggested to use the taxonomy of requirements:

- EQP for equipment,
- FUN for functional,
- SAF for safety properties.

For this exercise, you should work in group of 2 or 3 people.

Development of Search Algorithms

Proving the Remaining Obligations

In the development "search" that was distributed to you along with the summary slides, there are some proof obligations that are not discharged. Use the interactive proving facility of the Rodin Platform to discharge those obligations.

- Model **m_0b**: *search/act1/FIS*.
- Model **m_1a**: *search/grd1/WD*, *progress/grd1/WD* and *progress/inv2/INV*.
- Model **m_1b**: *search/grd1/WD*, *progress/grd1/WD*, *progress/inv2/INV* and *progress/VAR*.

A useful point to know here is how to reuse proofs (or rather part of the proof) by "Copy/Paste".

- Right click on the proof tree node where you want to copy the proof sub-tree starting from that node and choose "Copy".
- Right click on the proof tree node where you want to re-use the copied proof tree on and choose "Paste".

A Different Search Algorithm

This is the exercise from the last slide (slide 33) of the set of slides "A Summary of the Event-B Modeling Notation".

Create a new refinement of **m_0a** or **m_0b** in order to obtain the following final program:

$i, j := 1, n + 1$;	**initialisation**
WHILE $f(j-1) \neq v$ DO	
$j := j - 1$	**progress**
END ;	
$i := j - 1$	**search**

Write down on paper the following:

- guard strengthening proof obligations if any, and
- simulation proof obligations if any

for the event *search* of the new machine. Compare the result with the proof obligations generated by the Rodin Platform.

Finding the Maximum Number of An Array (Challenge Exercise)

In this exercise, we develop a program to find the maximum number of an array. Assume that we have an array "a" which has "n" elements, with the indexes starting from 1 to n. The algorithm to find the maximum number within an array (or rather the index of the maximum number) is as follows.

We use to indexes x and y where x starting from 1 and y is initially n. At each step of the algorithm, we do the following:

- if $x = y$ then x is the index of the maximum number.

- otherwise, i.e. $x \neq y$, we compare the value of a at indexes x and y and do the following:
  - if $a(x) < a(y)$, increase x by 1.
  - otherwise, decrease y by 1.

Formalize the above algorithm in Event-B and establish the sequential program out of it. Do not forget to prove the convergence of new events in a refinement.