Project 1

Metode formale în ingineria software 2018-2019
Formal Methods in Software Engineering 2018-2019

Deadline: Wednesday, November 28, 16:00. An archive including the detailed description of the solution (pdf file) and a folder proj1/ with source code will be uploaded using the link https://www.dropbox.com/request/Kk3tOrA7AqOOwkQJvrPQ The pdf file will include also the instructions how to test the source code.

Exercise 1 Consider the following problem PRJ1:
Input: A integer n.
Ouput: The smallest integer m that is greater than n and for which there is a positive integer > 1 that divides both n and m.

1. Specify PRJ1 as a pair (precondition, postcondition).
2. Write a CinK program solving PRJ1.
3. Find the appropriate invariant(s) and use the algorithm 2pt() from the file tableaux.maude [1] to compute the proof tableaux of the program and its specification. Do not forget to annotate the program with the invariant(s).
4. Identify in the resulted proof tableaux the implications that must be proved and prove them manually or using the Web interface of Z3 [2]. In the later case, the pdf file will include the interaction with the Web tool, and the archive will include a file with the Z3 code. If a proof of an implication fails, explain why it fails and how the error can be corrected.
5. Show, for this example, how the valid proof tableaux can be used to prove the partial correctness of the program w.r.t. the specification.

Exercise 2 The goal is to add unidimensional array to the language CinK, used in classes and implemented in Maude [1]. Here is an example how the arrays should be declared and used in programs:

```
int a[5];
int i;
i = 2;
a[1] = 7;
a[i*2] = 5 + a[i+1];
```

1. Add the required statements to the file cink-syntax.maude to handle unidimensional arrays in programs. Write several programs and show that they are correctly parsed.

2. Modify the files state.maude and cink-bigstep (or cink-smallstep) in order to give semantics to arrays. For the array values use the module INTARRAY defined in the file array.maude. An array variable will be initialized with the constant zeros: $\texttt{a} \mapsto \texttt{zeros}$. For changing an array component use the operation update: $\texttt{a} \mapsto \texttt{update}(\texttt{a}, 1, 7)$, and for reading an array component use the operation sel: $5 + \texttt{sel}(\texttt{a}, 3)$. Use the examples from the first item to show that the programs are correctly executed.

3. Find a counter example showing that the following axiom is not sound:

$$\overline{(\!|Q[E_2/A[E_1]]|\!)\ A\texttt{[}E_1\texttt{]=}E_2\texttt{;}\ (\!|Q|\!)}$$

4. Argue why the following axiom is sound:

$$\overline{(\!|Q[\texttt{update}(A, E_1, E_2)]|\!)\ A\texttt{[}E_1\texttt{]=}E_2\texttt{;}\ (\!|Q|\!)}$$

   where $\texttt{update}(A, E_1, E_2)$ is similar to the operation with the same name given in array.maud, but here defined as an expression in the (extended) language definition. Give a big-step semantics for the new expression $\texttt{update}(A, E_1, E_2)$.

Exercise 3

1. Add new statements to the file tableaux.maude such that the algorithm 2pt() can compute the proof tableaux for programs with arrays. Show, using the test programs, that new definition works correctly.

2. The same questions as for Exercise 1, but for the following problem:
   Input: A sequence $(a_0, a_1, \ldots, a_{n-1})$ of n integers.
   Ouput: 1 if there is an i such that a_i is zero, 0 otherwise.

References

[1] Maude framework for FMSE course: `http://fmse.info.uaic.ro:8088/dorel.lucanu/fmse-maude/`

[2] the Web intrface for Z3: `https://rise4fun.com/z3`

Prof. dr. Dorel Lucanu