Homework 2

Metode formale în ingineria software 2021-2022
Formal Methods in Software Engineering 2021-2022

Deadline: Friday, January 14, 2022, 22:00.

General rules

The answer will be included into a subfolder HW2 on Google drive (in the same folder with seminar/laboratory work). The subfolder HW1 will include:
– a pdf file containing a detailed description of the solution for each exercise and instructions how to test the source code (if any);
– a Google doc file including the source code.
It is strongly recommended to include in the pdf descriptions all the steps performed to build the solution, what problems have been encountered and how they have been solved.
Solutions including only the source code will not be considered.

Remark

The homework is an individual task. The collaborations are not allowed; solutions with a high level of similarity will be rejected.
Sending the homework implies that you have signed the following
Sworn declaration
I hereby declare, under oath, that this homework has been my independent work and has not been aided with any prohibited means. I declare, to the best of my knowledge and belief, that all passages taken from published and unpublished sources or documents have been reproduced whether as original, slightly changed or in thought, have been mentioned as such at the corresponding places of the thesis, by citation, where the extent of the original quotes is indicated.

Exercise 1 The goal of this exercise is to see how Why3 applies the weakest-precondition-based calculus for computing the verification condition (see Floyd-Hoare Logic presentation, slide 56/90, `https://drive.google.com/file/d/1bbO1Uof9QaOL98fO0saecTqLLurnBAf2/view`). It can be used a local installation (`http://why3.lri.fr/`) or the online tool (`http://why3.lri.fr/try/`) for Why3. To explain a rule, use a a function with following template:

let $\langle name \rangle$ ($\langle params \rangle$) : $\langle result - type \rangle$
ensures Q
=
$\langle body \rangle$

where $\langle body \rangle$ should describe a statement as follows:

1. for an assignment $X = E$;, $\langle body \rangle$ will return E;
2. for `if` E A `else` B, $\langle body \rangle$ will use `if-else` for computing the returned result by the assignment A or B;
3. for `while` E A, $\langle body \rangle$ will include a `while` statement together with the invariant specification.

For each rule, display the verification condition and explain it by identifying each component of the rule.

Exercise 2 This is similar to the previous exercise, but it asks for explaining in the same way how the verification condition is computed for a function contract, using the rule described in Contracts presentation, slides 17-19 (`https://drive.google.com/file/d/1wwn1Iq5CG8cLlGR-RXH43b0_qus4Fn44/view`)

Exercise 3 The goal of this exercise is to synthesize a loop program that preserves the loop invariant $y = x * x \wedge z = y * x \wedge x \leq n$ and the variant $n - x$. The only allowed arithmetic operation is addition (without multiplications!). Use Why3 or Frama-C to show that loop program indeed preserves the invariant. The program will have the structure

$\langle initial - values - assignment \rangle$
while $\langle condition \rangle$˜
$\langle invariant - specification \rangle$
$\langle while - body \rangle$

Exercise 4 The goal of this exercise is to use Frama-C (`https://frama-c.com/index.html`) for specifying and correctness proving of a C program. First task is to install Frama-C. Then chose a C program from the section "C Programs on Arrays" from the site
`https://www.tutorialgateway.org/c-programming-examples/` and reorganize it into three functions: one for reading the input, one for writing the output, and one for the processing part. The processing function must include at least one loop statement. Write an appropriate ACSL specification for the processing function and use Frama-C to check if the program is correct w.r.t. the specification.

Prof. dr. Dorel Lucanu