Secure and Dependable Systems
Jacobs University Bremen
Dr. Florian Rabe

Homework 3
given: 2017-02-28

Homework 3

You have to submit your solutions as announced in the lecture.
Unless mentioned otherwise, all problems are due 2017-03-09, before the lecture.
There will be no deadline extensions unless mentioned otherwise in the lecture.

Problem 3.1 *Formal Systems: Practice*

Points: 3

Consider the partial implementation of a formal system in the course repository. It already includes most algorithms and features discussed in the notes.

Extend this implementation with product types. That requires changing the data types, parser, printer, checker, and interpreter.

Problem 3.2 *Formal Systems: Theory*

Points: 4

Consider the type theory and programming language developed in the lecture notes.

1. Give two meaningful programs as terms over the programming language developed in the lecture notes.

   The details of the programs are not essential but one program each should consist of (at least):

   - a recursive function declaration,
   - a function declaration that contains a while-loop that makes assignments to a mutable variable

   For example, you can implement the factorial function once recursively and once using a while-loop.

2. Give the derivations for the well-formedness of your programs in all detail.

Note: The typing derivations are trees that tend to grow quickly and can be painful to write down. For example, you may:

- write them manually on paper and submit a photograph or scan, or
- write them in LaTeX, or
- adapt the Checker.scala class to print out the derivation in some format (svg, LaTeX, . . .).

Problem 3.3 *Formal Systems: Theory*

Points: 3

Add disjoint-union types to the type theory by giving new

- productions to the grammar,
- typing rules,
- evaluation rules.

The disjoint-union type $A + B$ implements the disjoint union of the two types A and B. Its terms are $inj_1(a)$ for $a \in A$ and $inj_2(b)$ for $b \in B$. To work with a term $u : A+B$, we use case-distinction $cases(u, f, g) : C$ where $f : A \to C$ and $g : B \to C$.