Secure and Dependable Systems
Jacobs University Bremen
Dr. Florian Rabe

Homework 4
given: 2017-03-20

Homework 4

You have to submit your solutions as announced in the lecture.
Unless mentioned otherwise, all problems are due 2017-03-30, before the lecture.
There will be no deadline extensions unless mentioned otherwise in the lecture.

Problem 4.1 *Verification* Points: 4

Choose two of the following functions, and give appropriate function specifications, loop invariants, and termination orderings for them.

```
fun $EuclideanAlgorithm(m : int, n : int) : int$ =
    $x := m$
    $y := n$
    while $x \neq y$
        if $x < y$
            $y := y - x$
        else
            $x := x - y$
    return $x$
```

```
fun $factorial(n : \mathbb{N}) : \mathbb{N}$ =
    $product := 1$
    $factor := 1$
    while $factor \leq n$
        $product := product \cdot factor$
        $factor := factor + 1$
    return $product$
```

```
fun $linearFibonacci(n : \mathbb{N}) : \mathbb{N}$ =
    if $n \leq 1$
        $n$
    else
        $prev := 0$
        $current := 1$
        $i = 1$
        while $i < n$
            $next := current + prev$
            $prev := current$
            $current := next$
            $i := i + 1$
        return $current$
```

```
fun $revertImmutable[A](x : List[A]) : List[A]$ =
    $rest := x$
    $rev := []$
    while $rest \neq []$
        $rev := cons(rest.head, rev)$
        $rest := rest.tail$
    return $rev$
```

Problem 4.2 *Dynamic Logic: Practice* Points: 3

Install either Why3 or KeY (see the links in the lecture notes).

Write a simple function that contains a while-loop in it. Annotate it with pre/postcondition and loop invariant and use the tool to verify that it meets the specification.

Submit a reasonable combination of screen shots, shell logs, system output etc. that demonstrates you completed the task.

You may use any example that is already part of the available documentation or tutorials. But you have to prove that you actually installed the system and ran the verification. For example, you can copy an example from the tutorial, rename the function to your name, and then run the verification.

Problem 4.3 *Dynamic Logic: Theory* Points: 3

Prove the soundness of the rules for if and while.