Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

D. Basin and P. Müller

Formal Methods and Functional Programming

Exercise Sheet 7: Motivation and Induction

Submission deadline: April 26th, 2010

Note that the tutors for exercise groups will be different for the second half of the course. Please have a look at the course web page (`http://www.infsec.ethz.ch/education/ss10/fmfp/`) to see who is your tutor. However, the times and rooms of exercise sessions remain the same. For questions about the new exercise groups, please contact Alex Summers (`alexander.summers@inf.ethz.ch`).

Please submit your solution before **9:15am** on the submission date specified above. Solutions can be submitted via e-mail or by using the boxes to the left of **RZ F1**. Make sure that the first page always contains your name, the exercise sheet number as well as your tutor's name and the weekday (Tuesday or Wednesday) of your exercise group. Don't forget to staple your pages if you submit more than one page.

Assignment 1

Task: Write a program - in your preferred imperative programming language - that computes $\lfloor \sqrt[m]{n} \rfloor$, for integers $m > 0$ and $n \geq 0$. Assume that the variables `x` and `y` initially store the integers m and n, respectively. The result of the computation should be finally stored in the variable `z`. Give textual arguments justifying the correctness of your program.

The notation $\lfloor x \rfloor$ denotes the floor function applied to x, i.e., a real number x will be rounded down to the nearest integer.
More precisely: $\forall x \in \mathbb{R} : n = \lfloor x \rfloor \in \mathbb{N} \Rightarrow n \leq x < n + 1$.

You should restrict yourself to using only the basic constructs of an imperative programming language (loops and conditionals) and to the basic arithmetic operations (addition, substraction, multiplication, division) on integers.

Hint: You might want to first look at the simplified problem where $m = 2$ and then generalize your solution to $m > 0$.

Assignment 2

Strong vs. weak induction

Remember that there are two main types of induction on the natural numbers (you have already seen and used these in the first half of the course):

- When using **weak induction** we first prove the base case ($n = 0$) and for the induction step ($n + 1$) we assume that the statement holds for case n.
- When using **strong induction** we do not prove a separate base case. Instead, we prove $P(n)$ for an arbitrary n, using the assumption $\forall m < n : P(m)$.

The assumption in both types of induction is called the *induction hypothesis*.

Coin coverage

Task: Given a magical wallet with an unlimited supply of 5 Rappen and 3 Rappen coins, prove that you are able to pay any amount equal to or bigger than 8 Rappen.

Give arguments for your choice(s) of induction (weak or strong).

Assignment 3

Consider the following definition (*Peano numerals*) of the natural numbers and the addition operation defined on them:

```
data Nat  =  Zero
          |  Succ Nat
```

```
plus Zero n      = n
plus (Succ m) n  = Succ (plus m n)
```

Task: Prove the associativity of `plus`, i.e. prove the statement
$\forall r, s, t \in$ `Nat:  plus r (plus s t) = plus (plus r s) t`.

Assignment 4 - Headache of the week

Task: Prove the commutativity of the `plus` operation as defined in assignment 3, i.e., prove the statement $\forall m, n \in$ `Nat:  plus m n = plus n m`.

A major factor for success is to carefully structure your proof: make it explicit what you may assume, what you want to show and on which variable(s) you perform induction.