Semantics and Verification of Software (SS15)
apl. Prof. Dr. Thomas Noll
Dr. Federico Olmedo Christoph Matheja

Lehrstuhl für Informatik 2 Softwaremodellierung und Verifikation
RWTH AACHEN UNIVERSITY

Exercise Sheet 1

Due date: April 21st. You can hand in your solutions at the start of the exercise class.

Exercise 1 (Recursion and Structural Induction) — 67.5%

Consider the set of arithmetical expressions AExp given by grammar

$$a ::= z \mid x \mid a - a \mid a + a \mid a * a \ .$$

Here z ranges over the set of integers $\mathbb{Z}$ and x over the set of program variables Var.

(a) [7.5%] Give a recursive definition of function $\mathtt{fv} : \mathsf{AExp} \to \mathcal{P}(\mathsf{Var})$ that computes the set of variables of an arithmetical expression. For instance, we should have

$$\mathtt{fv}(5 * y + x * (4 + 3y)) = \{x, y\} \ .$$

(b) [7.5%] Give a recursive definition of the *textual substitution* operator $a\,[x := a']$ that replaces every ocurrence of variable x in expression a with expression a'. For example, we should have

$$(x + x * y)\,[x := 3 + z] = (3 + z) + (3 + z) * y \ .$$

(c) [7.5%] Give a recursive definition of function $\mathtt{occ} : \mathsf{AExp} \times \mathsf{Var} \to \mathbb{N}$ that counts the number of occurrences of a variable within an arithmetic expression. For instance, we should have

$$\mathtt{occ}(x + x * y, x) = 2 \ .$$

(d) [20%] Show by induction on the structure of a that

$$\mathtt{fv}(a\,[x := a']) \subseteq (\mathtt{fv}(a) \setminus \{x\}) \cup \mathtt{fv}(a') \ .$$

(e) Consider the recursive function $\mathtt{length} : \mathsf{AExp} \to \mathbb{N}$ defined by clauses

$$\mathtt{length}(z) = \mathtt{length}(x) = 1$$
$$\mathtt{length}(a_1 \oplus a_2) = 1 + \mathtt{length}(a_1) + \mathtt{length}(a_2) \quad \text{for } \oplus \in \{-, +, *\} \ .$$

i) [5%] Determine $\mathtt{length}(a\,[x := a'])$ in terms of $\mathtt{occ}(a, x)$, $\mathtt{length}(a)$ and $\mathtt{length}(a')$.

ii) [20%] Prove that your proposed formula in (i) is correct.

Exercise 2 (Programming Language WHILE) — 32.5%

(a) [25%] Write a program that computes the n-th Fibonacci number[1] and stores it in variable y. (Recall that the programming language presented in the lecture features neither recursion nor arrays.)

(b) [7.5%] Depict the flow diagram for the above program.

[1] The n-th Fibonacci number f_n is defined as follows: $f_0 = f_1 = 1$, $f_{n+2} = f_{n+1} + f_n$ for all $n \geq 0$.