Exercise Sheet 1

Due date: October 25th. You can hand in your solutions at the start of the exercise class.

Task 1: Recursion and Structural Induction (7 points)

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) 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

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

(b) Give a recursive definition of function $\texttt{occ} : \mathsf{AExp} \times \mathsf{var} \to \mathbb{N}$ that counts the number of occurrences of a variable within an arithmetic expression. For instance, we should have

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

(c) Show by induction on the structure of a that

$$\texttt{FV}(a\left[x := a'\right]) \subseteq (\texttt{FV}(a) \setminus \{x\}) \cup \texttt{FV}(a') \ .$$

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

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

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

(ii) Prove that your proposed formula in (i) is correct.

Task 2: The Programming Language WHILE (2+1 points)

(a) 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) 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 $n \geq 0$.