universität innsbruck

Program Verification	SS 2021	LVA 703083+703084
Sheet 8		Deadline: May 11, 2021, 8am

- Prepare your solutions on paper.
- Marking an exercise in OLAT means that a significant part of that exercise has been treated.
- Upload your Haskell files and your paper solution in OLAT, the latter as one PDF.

Exercise 1 *Implementation of Pattern Disjointness* 4 p.

The aim is to implement a function that tests a program on pattern disjointness. In the negative case a term should be returned, that is matched by two different rules.

1. Implement a function to rename variables in a term. You can assume that the input is a linear term, since all lhss of functional programs must be linear. (2 points)
2. Implement the decision procedure. Note that the template file already contains the unification algorithm from the lecture. (2 points)

Exercise 2 *Correctness of Implementation of Unification* 12 p.

Study the proof given on slides 4/36–40

1. Perform the proof of case 3, i.e., where the arguments are $(f(ts), x) : u$ and v. (3 points)
2. Write down the three missing sub-cases of case 4. (1 point)
3. Write down a proof for the sub-case corresponding to an occurs check. (3 points)
4. Complete the proof of the sub-case of case 4 that was handled in the lecture (slide 4/39), i.e., show $\mathit{set}\ ((x, t) : v') \in NF(\rightsquigarrow)$ (5 points)

Exercise 3 *Lexicographic Combinations* 4 p.

On slide 4/40, it was argued that termination holds because of a lexicographic measure. In this exercise we want to be a bit more formal about this aspect by showing that taking lexicographic combinations is a valid technique for termination proving.
In detail: A binary relation $\succ$ over some set A is strongly normalizing, if and only if there does not exist an infinite sequence of the form

$$a_0 \succ a_1 \succ a_2 \succ a_3 \succ \dots.$$

Given n binary relations $\succ_1, \dots, \succ_n$ over sets $A_1, \dots, A_n$, we define their lexicographic combination $\succ_{lex}$ as a binary relation over $A_1 \times \ldots \times A_n$ as follows: a lexicographic decrease happens, if for some position i, the element at position i decreases w.r.t. $\succ_i$, the elements before position i are unchanged, and there is no restriction on the elements after position i. This can be made formal via the following inference rule:

$$\frac{1 \leq i \leq n \quad a_i \succ_i b_i}{(a_1, \dots, a_{i-1}, a_i, \dots, a_n) \succ_{lex} (a_1, \dots, a_{i-1}, b_i, \dots, b_n)}$$

Prove that whenever all $\succ_i$ are strongly normalizing for $1 \leq i \leq n$, then so is $\succ_{lex}$. (4 points)