- Prepare your solutions on paper.
- Mark the exercises in OLAT before the deadline.
- Marking an exercise means that a significant part of that exercise has been treated.

Exercise 1 *Pattern Completeness* **7 p.**

Consider the algorithm for pattern completeness on slide 4/36.
The output of the algorithm is just a Boolean, i.e., the result is either $\bot$ (not pattern complete) or $\varnothing$ (pattern complete).
Note that the fully expanded semantics of completeness of a set of pattern problems P is as follows:

$$P \text{ is complete iff } \forall pp \in P.\ \forall \gamma : \mathcal{V} \to \mathcal{T}(\mathcal{C}).\ \underbrace{\exists mp \in pp.\ \exists \sigma.\ \forall (\ell, t) \in mp.\ \ell\sigma = t\gamma}_{=:\varphi(pp,\gamma)}$$

Hence, if P is not pattern complete there must be some witness pattern problem $pp \in P$ and witness substitution $\sigma : \mathcal{V} \to \mathcal{T}(\mathcal{C})$ such that $\varphi(pp, \gamma)$ is not satisfied.

- Modify the algorithm for pattern completeness in a way that witnesses can be obtained instead of just returning $\bot$ for incomplete P.
- Illustrate the modified algorithm on the example input on slide 4/38.
- You do NOT have to prove correctness of the modified algorithm.

Hint: only $\rightharpoonup\!\!\!\!\!\rightharpoonup$ needs to be modified, $\rightharpoonup$ does not need to be altered.

Exercise 2 *Strong Normalization of Lexicographic Combinations* **7 p.**

1. On slide 4/32, 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.

   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 $SN(\succ_i)$ for all $1 \leq i \leq n$, then also $SN(\succ_{lex})$ is satisfied. (4 points)

2. Find a lexicographic combination of strongly normalizing relations such that all rules of the pattern completeness algorithm on slide 4/36 decrease w.r.t. that combination. You can inline and merge rules as on slide 4/39, i.e., you only need to consider the (inlined) rules of (decompose), (match), (clash, merged with remove-mp), (success, merged with remove-pp), (failure), and (instantiate).

   Here, you may assume that $\succ_1$ is defined as the relation on slide 4/43 such that $P \succ_1 P'$ whenever P is instantiated to P', and $P \succeq_1 P'$ for all other (inlined) rules. (3 points)

Exercise 3 *Innermost Evaluation* **2 p.**

Can you figure out why in the $\overset{i}{\hookrightarrow}$-definition, $NF(\hookrightarrow)$ was used instead of $NF(\overset{i}{\hookrightarrow})$?

Exercise 4 *Termination Analysis on Paper* **4 p.**

Write your favourite sorting algorithm as functional program and try to prove termination via the subterm criterion. If the proof is not completed, indicate which dependency pairs remain.
Of course, here you also have to define a function for comparing natural numbers and other auxiliary functions. But you can assume that there are already datatypes Nat, List and Bool.