In order to be admitted to the exam, you have to successfully submit your homework every week, except for 2 weeks. A successful submission is one where you get at least 1 point.

**Handin** Please submit this homework until Thursday, December 20, either via email to Philipp Schuster (philipp.schuster@uni-tuebingen.de) before 12:00, or on paper at the beginning of the lab.

**Groups** You can work in groups of up to 2 people. Please include the names and Matrikelnummern of all group members in your submission.

**Points** For each of the Tasks you get between 0 and 2 points for a total of 6 points. You get:
- 1 point, if your submission shows that you tried to solve the task.
- 2 points, if your submission is mostly correct.

**Task 1: Join and Meet**

Consider the following set of types:
\[
\langle \text{type}\rangle ::= \text{Top} | \langle \text{type}\rangle \rightarrow \langle \text{type}\rangle | \{l_i : \langle \text{type}\rangle_i\}
\]

Consider the following rules for the subtyping relation:

\[
\begin{align*}
\text{S-Top} & : S <: \text{Top} \\
\text{S-Arrow} & : T_1 <: S_1 \quad T_2 <: S_2 \\
& \quad \Rightarrow S_1 \rightarrow S_2 <: T_1 \rightarrow T_2 \\
\text{S-RCD} & : \{l_i | i \in 1..n\} \subseteq \{k_j | j \in 1..m\} \\
& \quad \text{for each } k_j = l_i, S_j <: T_i \\
& \quad \Rightarrow \{k_j : S_j j \in 1..m\} <: \{l_i : T_i i \in 1..n\}
\end{align*}
\]

For each of the following pairs of types, what is their Join and what is their Meet?

1. \{a : \text{Top}\} and \text{Top}
2. \{a : \text{Top}, b : \text{Top}\} \rightarrow \{x : \text{Top}\} \rightarrow \{z : \text{Top}\} and \{a : \text{Top}\} \rightarrow \{y : \text{Top}\} \rightarrow \{z : \text{Top}\}
3. \{\{a : \text{Top}, b : \text{Top}\} \rightarrow \{x : \text{Top}\}\} \rightarrow \{z : \text{Top}\} and \{\{a : \text{Top}\} \rightarrow \{y : \text{Top}\}\} \rightarrow \{z : \text{Top}\}
Task 2: Reflexivity of the subtyping relation

Show that for the subtyping relation from Task 1 we have the following property: For all $T \in \text{type}$, we have $T \leq T$.

Task 3: Type checker for a language with subtyping

Implement the algorithmic type checker for the language from the lecture in a programming language of your choice. Do not try to write a parser but assume a representation of terms in memory. Examples for a representation of terms and types in Haskell and Java are on the website. Optionally, extend your implementation with support for an if-then-else construct in which case you would have to compute Joins and Meets.