Lambda-Calculus and Type Theory
ISR 2024
Obergurgl, Austria
Herman Geuvers & Niels van der Weide
Radboud University Nijmegen NL
Exercises Day 6

Lecture 12. Principal types: a functional programmers' view on type theory

1. (a) Determine the most general unifier of $(\alpha{\to}\beta){\to}\gamma$ and $\alpha{\to}\beta{\to}\gamma$.
   (b) Determine the most general unifier of $(\alpha{\to}\beta){\to}\gamma$ and $\gamma{\to}\alpha{\to}\beta$

2. Compute the principal type of $\mathbf{S} := \lambda x.\lambda y.\lambda z.x\, z(y\, z)$.

3. Consider the following two terms

   - $\lambda x.x\,(\lambda y.y\,(\lambda z.x))$
   - $\lambda x.x\,(\lambda y.x\,(\lambda z.z))$

   For each of these terms, compute its principal type, if it exists. (Give the end result and show your computation; if the term has no principal type, show how your computation yields 'fail'.)

4. For each of the following two terms, compute its principal type, if it exists.

   - $\lambda x.(\lambda y.x\,(x\, y))\,(\lambda u\, v.u)$
   - $\lambda y.(\lambda x.x\,(x\, y))\,(\lambda u\, v.u)$

   Give the end result and show your computation; if the term has no principal type, show how your computation yields 'fail'.