# Sémantique des Langages de Programmation (SemLP)
# TD n° 8 : Type Inference – Program Transformations

## Type Unification

**Exercice 1 : (LN : 196)**

Compute, if possible, by the method of unification the *most general types* for the following $\lambda$-terms :

$$\lambda x.\lambda y.\lambda z.xz(yz) \qquad \lambda x.\lambda y.x(yx) \qquad \lambda k.(k(\lambda x.\lambda h.hx))$$

**Exercice 2 : (LN : 200)**

Apply the reduction of unification to type inference to the types $t_1$, $t_2$ and $(t_1 \to t_2)$ relatively to the set of (type) variables $\{t_1, t_2\}$.

**Exercice 3 : (LN : 202)**

Find the $U_{A,B}$ $\lambda$-terms of Proposition 201 in the LN to the following types :

**1.** $A = t_1$ and $B = t_2$, and

**2.** $A = t_1$ and $B = t_1 \to t_2$

## Program transformations

**Exercice 4 : (LN : 208)**

Write down a simplified CPS transformation for a monadic call-by-value $\lambda$-calculus without let-definitions and tuples. Then apply the CPS transformation to show that it is possible to simulate the call-by-value $\lambda$-calculus in the call-by-name $\lambda$-calculus.

**Exercice 5 : (LN : 213)**

Define a closure conversion transformation that applies directly to the source language rather than to the CPS, value named form.

**Exercice 6 : (LN : 215)**

Apply the hoisting transformation to the terms resulting from the closure conversion of exercise 5.