Sémantique des Langages de Programmation (SemLP)
TD n° 7 : Typed λ-calculus

Exercice 1 : (LN : 179)
Show that if $x_1 : A_1, \ldots, x_n : A_n \vdash M : B$ is derivable then $(A_1 \to \cdots (A_n \to B) \cdots)$ is a *tautology* of propositional logic where we interpret $\to$ as implication and atomic types as propositional variables. Conclude that there are types A which are *not inhabited*, *i.e.*, there is no (closed) λ-term M such that $\emptyset \vdash M : A$.

Exercice 2 : (LN : 180)
Show that there is no λ-term M such that : $\emptyset \vdash M : (b \to b) \to b$. Write $A \to b$ as $\neg A$. Show that there are λ-terms N_1 and N_2 such that :

$$\emptyset \vdash N_1 : A \to (\neg\neg A) \ , \quad \emptyset \vdash N_2 : (\neg\neg\neg A) \to (\neg A) \ .$$

On the other hand, there are tautologies which are not inhabited ! For instance, consider : $A \equiv ((t \to s) \to t) \to t$. Show that there is no λ-term M in normal form such that $\emptyset \vdash M : A$ is derivable. This is enough because later we will show that all typable λ-terms normalize to a λ-term of the same type. For another example, show that there is no λ-term M in normal form such that $\emptyset \vdash M : \neg\neg t \to t$ is derivable (the intuitionistic/constructive negation is not involutive !).

Exercice 3 : (LN : 183)
Suppose we reconsider the *non-logical extension* of the simply typed λ-calculus with a *basic type nat*, *constants* Z, S, Y, and with the following fixed-point rule :

$$C[\mathsf{Y}M] \to C[M(\mathsf{Y}M)] \ .$$

Let a *program* be a closed typable λ-term of type *nat* and let a *value* be a λ-term of the shape $(\mathsf{S} \cdots (\mathsf{SZ}) \cdots)$. Show that if P is a program in *normal form* (cannot reduce) then P is a *value*.

Exercice 4 : (LN : 200) recursive types
Assume a *recursively defined type* t satisfying the equation $t = t \to b$ and suppose we add a rule for typing up to type equality :

$$\frac{\Gamma \vdash M : A \quad A = B}{\Gamma \vdash M : B} \ .$$

Show that in this case the following λ-term (Curry's fixed point combinator) is typable (*e.g.*, in Curry's style) :

$$Y \equiv \lambda f.(\lambda x.f(xx))(\lambda x.f(xx)) \ .$$

Are the λ-terms typable in this system terminating ?