Sémantique des Langages de Programmation (SemLP)
TD n° 4 : λ-calculus

Exercice 1 : β-normal forms – (LN : 104)

Let NF be the smallest set of λ-terms such that :

$$\frac{M_i \in NF \quad i = 1, \ldots, k}{\lambda x_1 \ldots x_n . x M_1 \ldots M_k \in NF}\ .$$

Show that NF is exactly the set of λ-terms in *β-normal form*.

Exercice 2 : Curry FP – (LN : 105)

We define $Y \equiv \lambda f.\ \Delta_f \Delta_f$ with $\Delta_f \equiv \lambda x.\ f(xx)$. Show that

$$YM =_\beta M(YM)$$

Exercice 3 : Turing FP – (LN : 106)

We define $Y_T \equiv (\lambda xy.y(xxy))(\lambda xy.y(xxy))$. Show that $Y_T f$ is not only convertible to, but *reduces to*, $f(Y_T f)$.

Exercice 4 :

Recall the definition of parallel β-reduction given in the lecture notes :

$$\frac{}{M \Rightarrow M} \qquad \frac{M \Rightarrow M' \qquad N \Rightarrow N'}{(\lambda x.M)N \Rightarrow [N'/x]M'} \qquad \frac{M \Rightarrow M' \qquad N \Rightarrow N'}{MN \Rightarrow M'N'} \qquad \frac{M \Rightarrow M'}{(\lambda x.M) \Rightarrow (\lambda x.M')}$$

Let $M \equiv (\lambda x.Ix)(II)$ where $I \equiv (\lambda z.z)$. Say what is the minimum number of reductions required to reduce M to I. Justify your answer.

Exercice 5 : Church Numerals

Recall the definition of *Church Numerals* of the lecture notes :

$$\underline{n} \equiv \lambda f.\lambda x.\ (\underbrace{f \ldots (f}_{n \text{ times}}\, x) \ldots)$$

1. Show semi-formally that the following functions are correct encodings of their natural numbers counterparts :

$$\begin{array}{rcll} A & \equiv & \lambda n.\lambda m.\lambda f.\lambda x.\ (nf)(mfx) & \text{(addition)} \\ S & \equiv & \lambda n.\ A\ n\ \underline{1} & \text{(successor)} \\ M & \equiv & \lambda n.\lambda m.\lambda x.\ n(mx) & \text{(product)} \end{array}$$

2. Consider now the encoding of *Booleans* :

$$T \equiv \lambda x.\lambda y.\ x \quad \textsf{(true)} \qquad F \equiv \lambda x.\lambda y.\ y \quad \textsf{(false)}$$

Show that the following term encodes the standard if-then-else construct :

$$C \equiv \lambda x.\lambda y.\lambda z.\ xyz \quad \textsf{(if-then-else)}$$

3. Do the same for the following encoding of pairs and projections :

$$\begin{array}{rcll} P & \equiv & \lambda x.\lambda y.\lambda z.\ zxy & \text{(pairs)} \\ P_1 & \equiv & \lambda p.\ p\ (\lambda x.\lambda y.\ x) & \text{(first projection)} \\ P_2 & \equiv & \lambda p.\ p\ (\lambda x.\lambda y.\ y) & \text{(second projection)} \end{array}$$