λ-Calcul et Logique Informatique

Guillaume Bury
bury@lsv.ens-cachan.fr

Exercice 1 — Connecteurs logiques dans le système $\mathcal{F}$
On rappelle les règles de typage pour la nouvelle construction du système $\mathcal{F}$, la quantification du second ordre. On note en majuscule les variables de type, et on suppose que la variable X n'apparaît pas libre dans Γ.

$$\frac{\Gamma \vdash u : F}{\Gamma \vdash \lambda X.\ u : \forall X.\ F} \qquad \frac{\Gamma \vdash u : \forall X.\ F}{\Gamma \vdash u\ T : F[X := T]}$$

1. On pose $\bot \stackrel{\text{def}}{=} \forall X.\ X$. Démontrer $\bot \Rightarrow P$ pour un type / une formule quelconque P.
2. On pose $A \wedge B \stackrel{\text{def}}{=} \forall X.\ (A \Rightarrow B \Rightarrow X) \Rightarrow X$. Montrer que cet encodage rend admissibles les règles usuelles de la conjonction, en donnant les encodages correspondants pour les constructions de paire et projections :

$$\frac{\Gamma \vdash u : A \quad \Gamma \vdash v : B}{\Gamma \vdash \langle u, v \rangle : A \wedge B} \qquad \frac{\Gamma \vdash u : A_1 \wedge A_2}{\Gamma \vdash \pi_i(u) : A_i}$$

3. On pose $A \vee B \stackrel{\text{def}}{=} \forall X.\ (A \Rightarrow X) \Rightarrow (B \Rightarrow X) \Rightarrow X$. Dériver les règles usuelles :

$$\frac{\Gamma \vdash u : A_i}{\Gamma \vdash \iota_i(u) : A_1 \vee A_2} \qquad \frac{\Gamma \vdash u : A \vee B \quad \Gamma, x_1 : A \vdash v_1 : C \quad \Gamma, x_2 : B \vdash v_2 : C}{\Gamma \vdash \text{case}(u, x_1.v_1, x_2.v_2) : C}$$

4. Proposer un encodage de $\exists X.\ F$ qui permette de dériver les règles suivantes (où on suppose que X n'apparaît pas dans Γ et P).

$$\frac{\Gamma \vdash u : F[X := T]}{\Gamma \vdash \langle\langle T, u \rangle\rangle : \exists X.\ F} \qquad \frac{\Gamma \vdash u : \exists X.\ F \quad \Gamma, x : F \vdash v : P}{\Gamma \vdash \text{CASE}(u, x.v) : P}$$

Remarque : on pourrait aussi vérifier que les réductions attendues (par exemple, $\text{case}(\iota_i(u), x_1.v_1, x_2.v_2) \to v_i[x_i := u]$) sont simulées par nos encodages.

Exercice 2 — Types de données en système $\mathcal{F}$
Nous allons maintenant exploiter le polymorphisme du système $\mathcal{F}$ pour pouvoir typer les encodages vus au TD 2 (booléens, paires, entiers, listes. . .).

1. Les entiers naturels peuvent être définis comme les termes générés par la signature $\{\ \mathsf{z} : \mathsf{N},\ \mathsf{s} : \mathsf{N} \Rightarrow \mathsf{N}\ \}$. En système $\mathcal{F}$, le type N ainsi donné se code en $\forall X.\ X \Rightarrow (X \Rightarrow X) \Rightarrow X$. Définir l'interprétation de z et s selon cet encodage.
2. Montrer que tout terme clos $u : \mathsf{N}$ est β-équivalent à un (unique) entier de Church $\lambda X.\lambda z.\lambda s.\ s^n\ z$. On écrira alors $[u]^{-1} = n$.

   On pourra s'appuyer sur la caractérisation des formes normales en système $\mathcal{F}$: elles s'écrivent $\lambda \mathcal{V}_1 \ldots \lambda \mathcal{V}_m.\ x\ u_1 \ldots u_n$ où les $\mathcal{V}$ sont des variables de terme ou de type.

   Remarque : si on avait pris $\forall X.\ (X \Rightarrow X) \Rightarrow X \Rightarrow X$ on aurait eu besoin de la $\beta\eta$-équivalence.
3. Coder la multiplication par deux sur les entiers, comme une fonction $\mathsf{double} : \mathsf{N} \Rightarrow \mathsf{N}$. Enoncer et prouver sa correction.
4. On considère un type arbitraire T donné par un ensemble fini de constructeurs $(\mathsf{C}_i)_{1 \leq i \leq n}$ où chaque constructeur a un type de la forme $A_1 \Rightarrow \ldots \Rightarrow A_n \Rightarrow T \Rightarrow \ldots \Rightarrow T \Rightarrow T$ où les A_i sont déja des types de $\mathcal{F}$. Expliquer comment on peut encoder T et ses constructeurs en système $\mathcal{F}$.
5. On définit le type B par trois constructeurs : $\mathsf{b} : \mathsf{B}$, $\mathsf{i} : \mathsf{B} \Rightarrow \mathsf{B}$ et $\mathsf{o} : \mathsf{B} \Rightarrow \mathsf{B}$. Donner l'encodage en système $\mathcal{F}$ du type B et de ses constructeurs.
6. Coder l'injection $\mathsf{b2n} : \mathsf{B} \Rightarrow \mathsf{N}$ définie par les équations suivantes :

   $$\mathsf{b2n}(\mathsf{b}) = \mathsf{s}\,\mathsf{z} \quad \mathsf{b2n}(\mathsf{i}(x)) = \mathsf{s}(\mathsf{double}(\mathsf{b2n}(x))) \quad \mathsf{b2n}(\mathsf{o}(x)) = \mathsf{double}(\mathsf{b2n}(x))$$

   Que représente le type B via cette traduction ?
7. Pour les plus courageux, coder l'addition sur B : donner une fonction $\mathsf{bplus} : \mathsf{B} \Rightarrow \mathsf{B} \Rightarrow \mathsf{B}$ de telle sorte que pour tous termes x et y on a $\mathsf{plus}\ (\mathsf{b2n}\ x)\ (\mathsf{b2n}\ y) = \mathsf{b2n}\ (\mathsf{bplus}\ x\ y)$.