

λ-Calcul et Logique Informatique

David Baelde
baelde@lsv.ens-cachan.fr

Exercice 1 — Paradoxe de Russell

On formalise (une partie de) la théorie naïve des ensemble en ajoutant à la logique du premier ordre les constructions suivantes :

- les termes du premier-ordre $\{ x \mid F \}$ représentant intuitivement un ensemble défini par compréhension ;
- les formules $t \in s$ représentant intuitivement l'appartenance ;
- les termes de preuve I_\in et E_\in pour introduire et éliminer les types \in ;
- les règles de typage suivantes :

$$\frac{\Gamma \vdash u : F[x := t]}{\Gamma \vdash I_\in(u) : t \in \{ x \mid F \}} \quad \frac{\Gamma \vdash u : t \in \{ x \mid F \}}{\Gamma \vdash E_\in(u) : F[x := t]}$$

- et enfin la nouvelle réduction $E_\in(I_\in(u)) \rightarrow u$.

Nous allons formaliser le paradoxe de Russell (aussi appelé paradoxe du menteur) dans ce système, et ainsi montrer son incohérence. Pour cela, on pose $S := \{ x \mid \neg(x \in x) \}$.

1. Donner un terme de type $(S \in S) \Rightarrow \neg(S \in S)$.
2. En déduire un terme de type $S \in S$.
3. En déduire un terme de type \perp .
4. Ce terme vous rappelle-t-il quelquechose ? réduisez-le si besoin.

Exercice 2 — 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 .

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

1. On pose $\perp \stackrel{\text{def}}{=} \forall X. X$. Démontrer $\perp \Rightarrow P$ 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 :

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

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

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

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).

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

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

Exercice 3 — 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 $\{z : \mathbf{N}, s : \mathbf{N} \Rightarrow \mathbf{N}\}$. En système \mathcal{F} , le type \mathbf{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 $u : \mathbf{N}$ 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 $\lambda \mathcal{V}_1 \dots \lambda \mathcal{V}_m. x u_1 \dots u_n$ 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 double : $\mathbf{N} \Rightarrow \mathbf{N}$. Envisager différentes façons d'énoncer et prouver sa correction.

4. On considère un type arbitraire T donné par un ensemble fini de constructeurs $(C_i)_{1 \leq i \leq n}$ où chaque constructeur a un type de la forme $A_1 \Rightarrow \dots \Rightarrow A_n \Rightarrow T \Rightarrow \dots \Rightarrow T \Rightarrow T$ où les A_i sont déjà des types de \mathcal{F} . Expliquer comment on peut encoder T et ses constructeurs en système \mathcal{F} .

5. On définit le type \mathbf{B} par trois constructeurs : $b : \mathbf{B}$, $i : \mathbf{B} \Rightarrow \mathbf{B}$ et $o : \mathbf{B} \Rightarrow \mathbf{B}$. Donner l'encodage en système \mathcal{F} du type \mathbf{B} et de ses constructeurs.

6. Coder l'injection $\text{b2n} : \mathbf{B} \Rightarrow \mathbf{N}$ définie par les équations suivantes :

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

Que représente le type \mathbf{B} via cette traduction ?

7. Pour les plus courageux, coder l'addition sur \mathbf{B} : donner une fonction $\text{bplus} : \mathbf{B} \Rightarrow \mathbf{B} \Rightarrow \mathbf{B}$ de telle sorte que pour tous termes x et y on a $\text{plus}(\text{b2n } x) (\text{b2n } y) = \text{b2n}(\text{bplus } x \ y)$.