TD Système F, types de données

E. Lozes

31 mars 2009

Exercice 1 – Rappel système F_2

On rappelle les règles d'inférence de types propres au système F_2 (ajouter Ax, $\Rightarrow I$, $\Rightarrow E$ pour tout avoir) :

$$\frac{\Gamma \vdash u : \forall X.F}{\Gamma \vdash uF' : F[X := F']}(\forall E) \qquad \frac{\Gamma \vdash u : F \quad X \notin v(\Gamma)}{\Gamma \vdash \Lambda X.u : \forall X.F}(\forall I)$$

1. On définit $\perp$ comme $\forall X.X$. Donner un terme preuve de $\forall X.\perp \Rightarrow X$.
2. Proposer un encodage du type $F \wedge F'$ (on pourra se souvenir de l'encodage de la paire $\langle u, v \rangle$ dans le lambda-calcul pur). Donner un terme preuve de $(A \Rightarrow B \Rightarrow C) \Rightarrow (A \wedge B) \Rightarrow C$. Montrer que les règles $\wedge E_1$, $\wedge E_2$ et $\wedge I$ sont admissibles en système F_2 si on adopte cet encodage.
3. On se propose d'encoder $A \vee B$ de sorte que le terme $i_1 u$, preuve de $A \vee B$ lorsque u est une preuve de A, est défini par $i_1 u = \Lambda P.\lambda k_1, k_2.k_1 u$. Retrouver la définition de $A \vee B$. Proposer un encodage de $\mathsf{case}\ u \left\{ \begin{array}{c} i_1 x \rightarrow v_1 \\ i_2 x \rightarrow v_2 \end{array} \right\}$, vérifier le typage et les réductions associées.

Exercice 2 – Les types de donnée

1. On définit le type Bool comme $\forall X.X \Rightarrow X \Rightarrow X$. Définir les termes true, false, et $\mathsf{ifthenelse}$ de type Bool et $\forall X.\mathsf{Bool} \Rightarrow X \Rightarrow X \Rightarrow X$ respectivement.
2. Rappeler le type des entiers de Church en système F.
3. On appelle $\Sigma = \{f_i : \tau_i\}_{i=1..n}$ une X-signature si tout type τ_i du symbole de fonction f_i est de la forme $U_1 \Rightarrow .. \Rightarrow U_n \Rightarrow X \Rightarrow .. \Rightarrow X$, où les U_i sont des types clos (sans variable libres). On définit le type de donnée τ_Σ de la signature Σ comme $\tau_\Sigma = \forall X.\tau_1 \Rightarrow ..\tau_n \Rightarrow X$. Montrer que les entiers, les booléens, la paire, et la somme ont le type de donnée de signatures que l'on commentera.
4. Définir de même le type de donnée des listes d'entiers, ainsi que les constructeurs de listes.
5. Généraliser, et montrer que tout terme de la Σ-algèbre libre (resp. tout symbole de fonction f_i) peut être représenté par un lambda terme du type de donnée τ_Σ (resp. de type $\tau_i[\tau_\Sigma/X]$).

6. Un type de données ne contient (presque) que des termes codants des termes libres. Vérifier, dans le cas des entiers, qu'un λ-terme normalisé de type Nat est un entier de Church $\Lambda X \lambda x, f.f^n(x)$ (on pourra raisonner sur la forme normale de tête).

Exercice 3 – Types existentiels

On veut montrer que les types existentiels $\exists X.F$ sont admissibles dans le système F_2, au sens où on sait donner un encodage de $\exists X.F$ et des règles :

$$\frac{\Gamma \vdash V[X := U]}{\Gamma \vdash \exists X.V}(I\exists) \qquad \frac{\Gamma \vdash \exists X.V \quad \Gamma, V \vdash W \quad X \notin \mathsf{fv}(\Gamma, W)}{\Gamma \vdash W}(E\exists)$$

On propose d'encoder $\exists X.F$ en $\forall Y.(\forall X.F \Rightarrow Y) \Rightarrow Y$. Montrer que les règles ci-dessus sont admissibles au sens où on peut donner des termes $\langle U, v\rangle$ de type $\exists X.V$ et $\nabla X, x.w$ de type $(\exists X.V) \Rightarrow W$, pour certaines hypothèses de typage des sous-termes v et w que l'on précisera. Interpréter l'élimination des coupures pour $\exists$ sur le redex $(\nabla X, x.w)\langle U, v\rangle$.