Le lambda calcul vu comme monade initiale

Julianna Zsidó

19. juin 2006

But : Théorème

lamda calcul pur

Le lambda calcul pur est l'objet intial dans la catégorie des monades exponentielles.

lambda calcul simplement typé

Le lambda calcul simplement typé est l'objet initial dans la catégorie des monades exponentielles typées.

Étude du lambda calcul

lambda calcul simplement typé

- ensemble des types $\mathcal{T} : * | \mathcal{T} \Rightarrow \mathcal{T}$
- ensemble des variables typés $\Gamma_X = ((x_i : t_i))_{x_i \in X}$ où $t_i \in \mathcal{T}$
- var : $\dfrac{(x : t) \in \Gamma_X}{\Gamma_X \vdash x : t}$
- app : $\dfrac{\Gamma_X \vdash M : u \Rightarrow t, \Gamma_X \vdash N : u}{\Gamma_X \vdash \mathrm{app}(M, N) : t}$
- $\mathrm{app1}_u$: $\dfrac{\Gamma_X \vdash M : u \Rightarrow t}{\Gamma_{X \cup (\infty_X : u)} \vdash \mathrm{app1}(M) : t}$
- abs_u : $\dfrac{\Gamma_{X \cup (\infty_X : u)} \vdash M : t}{\Gamma_X \vdash \mathrm{abs}(M) : u \Rightarrow t}$

lambda calcul simplement typé

- la substitution : $M[x \leftarrow N]$ où $x : t$ est libre dans $M : u$ et de même type que $N : t$
- β–réduction : $\dfrac{\Gamma_{X \cup (\infty_X : u)} \vdash M : t, \Gamma_X \vdash N : u}{\Gamma_X \vdash \mathrm{app}(\mathrm{abs}(M), N) \rightarrow_\beta M[\infty_X \leftarrow N] : t}$
- β–réduction (2) : $\dfrac{\Gamma_{X \cup (\infty_X : u)} \vdash M : t}{\Gamma_{X \cup (\infty_X : u)} \vdash \mathrm{app1}(\mathrm{abs}(M)) \rightarrow_\beta M : t}$
- η–réduction : $\dfrac{\Gamma_X \vdash M : u \Rightarrow t}{\Gamma_X \vdash \mathrm{abs}(\mathrm{app}(M, \mathrm{var}(\infty_X))) \rightarrow_\eta M : u \Rightarrow t}$
- η–réduction (2) : $\dfrac{\Gamma_X \vdash M : u \Rightarrow t}{\Gamma_X \vdash \mathrm{abs}(\mathrm{app1}(M)) \rightarrow_\eta M : u \Rightarrow t}$

Structures catégorielles

- *Monade* sur une catégorie $\mathcal{C}$: (R, η, μ) où $R : \mathcal{C} \to \mathcal{C}$, $\eta : \mathrm{Id} \stackrel{.}{\to} R$ et $\mu : R \circ R \stackrel{.}{\to} R$
- *Module* sur une monade R : (M, σ) où $M : \mathcal{C} \to \mathcal{C}$ et $\sigma : M \circ R \stackrel{.}{\to} M$
- *Morphisme de modules* $F : T \stackrel{.}{\to} S$ compatible avec σ
- *Module derivé (typé)*
- *Monade exponentielle (typée)*

Structures catégorielles

lambda calcul pur

- $\mathcal{C} = \mathrm{Ens}$
- *Module derivé* : $M'X := M(X^*) = M(X \cup \infty_X)$
- *Monade exponentielle* : (R, α) où R est une monade et $\alpha : R \stackrel{\cdot}{\to} R'$ un isomorphisme de modules

lambda calcul simplement typé

- $\mathcal{C} = (\mathrm{Ens} \downarrow \mathcal{T})$, $X \to \mathcal{T} \in Ob\mathcal{C}$, $X \to Y \in Mor\mathcal{C}$
- *Module derivé typé* : $\partial_u M(X \to \mathcal{T}) := M(X^* \to \mathcal{T})$
- *u–Module* : $M^u(X \to \mathcal{T}) := M(X \to \mathcal{T} \to \mathcal{T})$
- *Monade exponentielle typée* : monade R munie pour chaque type u d'un isomorphisme de modules $R^u \stackrel{\cdot}{\to} \partial_u R$

Construction des monades exponentielles

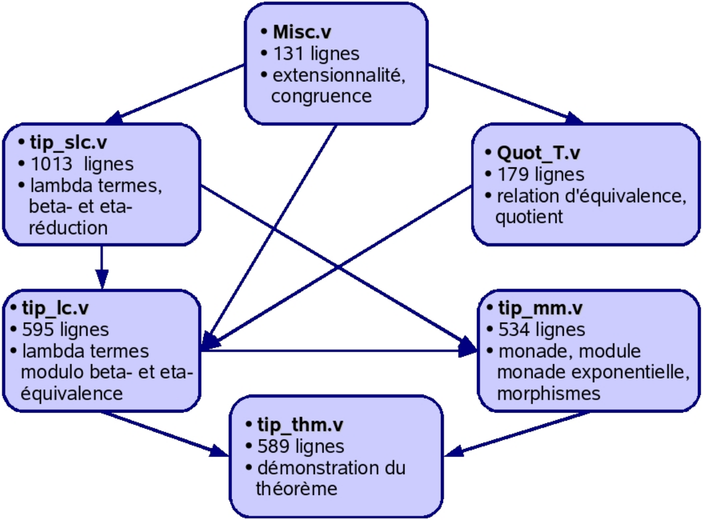
lambda calcul pur

LC est une monade exponentielle : $((\mathsf{LC}, \mathsf{var}, \mathsf{subst}), \mathsf{app1})$

lambda calcul simplement typé

ST est une monade exponentielle typée :
$((\mathsf{ST}, \mathsf{var}, \mathsf{subst}), (\mathsf{app1}_u)_{u \in \mathcal{T}})$
L'inverse de $\mathsf{app1}_u$ est abs_u ;β– et η–équivalence le vérifient.

Coq


• Misc.v
• 131 lignes
• extensionnalité, congruence
• tip_slc.v
• 1013 lignes
• lambda termes, beta- et eta- réduction
• Quot_T.v
• 179 lignes
• relation d'équivalence, quotient
• tip_lc.v
• 595 lignes
• lambda termes modulo beta- et eta- équivalence
• tip_mm.v
• 534 lignes
• monade, module monade exponentielle, morphismes
• tip_thm.v
• 589 lignes
• démonstration du théorème