Projet de DEA 2002/2003

Cours *Calcul des Constructions Inductives*

Présentation

L'objectif du projet est d'étudier diverses notions de réalisabilité pour les calculs des prédicats intuitionniste et classique. À titre d'exemple on comparera la réalisabilité de l'axiome de choix.

Les projets sont individuels. Toute question est à adresser à l'une des personnes suivantes :

Bruno.Barras@inria.fr, Christine.Paulin@lri.fr, Hugo.Herbelin@inria.fr.

Le calcul des prédicats intuitionniste IPC^ω

Les termes du calcul des prédicats d'ordre fonctionnel supérieur IPC^ω sont les λ-termes non typés (on note cet ensemble Λ).

$$t, u \quad ::= \quad x \mid \lambda x.t \mid tu$$

Les formules de IPC^ω sont

$$A, B \quad ::= \quad t = u \mid A \to A \mid \forall x.A \mid \exists x.A$$

Les règles d'inférence de IPC^ω sont définies par

$$\frac{}{\Gamma, A \vdash A} \qquad \frac{t =_\beta u}{\Gamma \vdash t = u} \qquad \frac{\Gamma \vdash t = u \qquad \Gamma \vdash A[t/x]}{\Gamma \vdash A[u/x]}$$

$$\frac{\Gamma, A \vdash B}{\Gamma \vdash A \to B} \qquad \frac{\Gamma \vdash A \to B \qquad \Gamma \vdash A}{\Gamma \vdash B}$$

$$\frac{\Gamma \vdash A \qquad x \text{ pas libre dans } \Gamma}{\Gamma \vdash \forall x.A} \qquad \frac{\Gamma \vdash \forall x.A}{\Gamma \vdash A[t/x]}$$

$$\frac{\Gamma \vdash A[t/x]}{\Gamma \vdash \exists x.A} \qquad \frac{\Gamma, A \vdash C \qquad \Gamma \vdash \exists x.A \qquad x \text{ pas libre dans } \Gamma}{\Gamma \vdash C}$$

où la substitution est sans capture de variable et les Γ des ensembles de formules[1].

Pour la gestion des variables liées et de la substitution, on pourra (en particulier) se référer à la contribution CoC-in-Coq [?]. Pour les critères de terminaison non structurels, on pourra éventuellement consulter le fichier `Logic/Wf.v` de la bibliothèque standard de Coq.

Réalisabilité dans IPC^ω

On réalise IPC^ω par les termes du λ-calcul (non typé) modulo β-conversion. On définit (t, b) comme $\lambda f.ftb$, $\pi_1(b)$ comme $b(\lambda xy.x)$ et $\pi_2(b)$ comme $b(\lambda xy.y)$. On définit $a \Vdash A$ de la manière suivante :

$$\begin{array}{rll} b \Vdash t = u & \text{ssi} & t =_\beta u \\ b \Vdash A \to B & \text{ssi} & \text{pour tout } a \Vdash A, \text{ on a } ba \Vdash B \\ a \Vdash \forall x.A & \text{ssi} & \text{pour tout } \lambda\text{-terme } t, \text{ on a } at \Vdash A[t/x] \\ a \Vdash \exists x.A & \text{ssi} & \pi_2(a) \Vdash A[\pi_1(a)/x] \end{array}$$

[1] On pourra prendre alternativement des suites de formules et ajouter une règle de contraction

1) Définir le prédicat $t =_\beta u$ en Coq.
2) Justifier la bonne fondation de la définition de $a \Vdash A$ et définir le prédicat $a \Vdash A$ en Coq.
3) Donner une preuve en Coq que l'axiome du choix sous sa forme fonctionnelle.

$$\forall x \exists y.\phi(x,y) \ \rightarrow\ \exists f \forall x \phi(x, fx)$$

est réalisable

4) Montrer que les formules (négatives) de la forme $\neg A$ (on pose $\neg A \equiv A \rightarrow \lambda xy.x = \lambda xy.y$) sont pré-réalisables, c'est-à-dire qu'il existe a tel que s'il existe b qui réalise $\neg A$ alors a aussi réalise $\neg A$.

5) Donner une preuve en Coq que le principe d'indépendance des prémisses pour les formules négatives (x n'est pas libre dans A) est réalisable.

$$(\neg A \rightarrow \exists x.\phi(x)) \ \rightarrow\ \exists x(\neg A \rightarrow \phi(x))$$

6) Formaliser IPC^ω et montrer que $A_1, ...A_n \vdash A$ entraîne l'existence de b (ayant $x_1, ..., x_n$ comme variables libres) tel que pour tout $a_1 \Vdash A_1$, ... $a_n \Vdash A_n$, on a $b[a_1/x_1]...[a_n/x_n] \Vdash A$.

Remarque : on pourra alternativement choisir comme ensemble de réalisateurs un sous-ensemble des λ-termes du méta-langage (c'est-à-dire Coq lui-même). Typiquement, ce sera la réunion des types du CCI engendrés par la grammaire

$T \quad ::= \quad$ `unit` $\mid T$`->`$T \mid \Lambda$`->`$T \mid \Lambda * T$

La β-conversion sera alors implicite.

Le calcul des prédicats classique PC^ω

On considère maintenant la version classique PC^ω de IPC^ω. Le système de preuve est inchangé à ceci près que l'on autorise plusieurs formules à droite du symbole $\vdash$.

$$\frac{}{\Gamma, A \vdash A, \Delta} \qquad \frac{t =_\beta u}{\Gamma \vdash t = u, \Delta} \qquad \frac{\Gamma \vdash t = u, \Delta \qquad \Gamma \vdash A[t/x], \Delta}{\Gamma \vdash A[u/x], \Delta}$$

$$\frac{\Gamma, A \vdash B, \Delta}{\Gamma \vdash A \rightarrow B, \Delta} \qquad \frac{\Gamma \vdash A \rightarrow B, \Delta \qquad \Gamma \vdash A, \Delta}{\Gamma \vdash B, \Delta}$$

$$\frac{\Gamma \vdash A, \Delta \qquad x \text{ pas libre dans } \Gamma, \Delta}{\Gamma \vdash \forall x.A, \Delta} \qquad \frac{\Gamma \vdash \forall x.A, \Delta}{\Gamma \vdash A[t/x], \Delta}$$

$$\frac{\Gamma \vdash A[t/x], \Delta}{\Gamma \vdash \exists x.A, \Delta} \qquad \frac{\Gamma, A \vdash \Delta \qquad \Gamma \vdash \exists x.A, \Delta \qquad x \text{ pas libre dans } \Gamma, \Delta}{\Gamma \vdash \Delta}$$

Réalisabilité classique dans PC^ω

La réalisabilité classique [2] met en jeu 3 types de réalisateurs. Les *termes* réalisent les conclusions, les *continuations* (ou *piles*) réalisent les hypothèses et les *commandes* réalisent l'absurde.

On utilise le calcul classique de [3] où les termes, continuations et commandes sont définies par induction mutuelle (les variables de ce calcul sont notées z, z', ... par opposition aux variables du λ-calcul dont parle la logique qui sont notées x, y, x', ...).

$$\begin{array}{lcll} a & ::= & z \mid \lambda z.a \mid \lambda x.a \mid \mu\alpha.c \mid (t, a) & \text{termes} \\ e & ::= & \alpha \mid a \cdot e \mid t \cdot e \mid (x, z) \mapsto c & \text{continuations} \\ c & ::= & a \star e & \text{commandes} \end{array}$$

La réalisabilité classique se définit relativement à un ensemble donné $\perp\!\!\!\perp$ de "réalisateurs de commandes". Plutôt que de gérer explicitement l'évaluation dans la définition de la réalisabilité, on exige que $\perp\!\!\!\perp$ soit clos par réduction inverse, où les règles de réduction sont :

$$\begin{array}{rcll} (\lambda z.a) \star (b \cdot e) & \to & a[b/z] \star e & (\beta_1) \\ (\lambda x.a) \star (t \cdot e) & \to & a[t/x] \star e & (\beta_2) \\ (t,a) \star ((x,z) \mapsto c) & \to & c[t/x][a/z] & (\mathit{paire}) \\ (\mu\alpha.c) \star e & \to & c[e/\alpha] & (\mu) \end{array}$$

On définit mutuellement la réalisabilité d'une hypothèse A par une continuation e, noté $e : A \Vdash$, et la réalisabilité d'une conclusion A par un terme a, noté $a :\Vdash A$, de la manière suivante

$$\begin{array}{rll} e : t = u \Vdash & \text{ssi} & t \neq_\beta u \\ a \cdot e : A \to B \Vdash & \text{ssi} & a :\Vdash A \text{ et } e : B \Vdash \\ t \cdot e : \forall x.A \Vdash & \text{ssi} & e : A[t/x] \Vdash \\ e : \exists x.A \Vdash & \text{ssi} & \text{pour tout } t, \text{ pour tout } a :\Vdash A[t/x], \text{ on a } (t,a) \star e \in \perp\!\!\!\perp \\ & & \\ a :\Vdash A & \text{ssi} & \text{pour tout } e : A \Vdash, \text{ on a } a \star e \in \perp\!\!\!\perp \end{array}$$

Noter que si on prend $\perp\!\!\!\perp = \emptyset$ alors, la réalisabilité classique coïncide avec la vérité : une conclusion est ou bien réalisable par n'importe quel terme ou bien réalisable par aucun terme.

On peut montrer par induction que $A_1, ..., A_m \vdash B_1, ..., B_n$ entraîne l'existence d'une commande c tel que pour tout $a_1 :\Vdash A_1$, ... $a_m :\Vdash A_m$, $e_1 : B_1 \Vdash$, ... $e_m : B_n \Vdash$, on a $c[a_1/z_1]...[a_m/z_m][e_1/\alpha_1]...[e_m/\alpha_m] \in \perp\!\!\!\perp$.

En particulier, si on définit $\perp\!\!\!\perp$ comme l'ensemble des commandes normalisables, alors on peut démontrer que les réalisateurs tant d'une conclusion que d'une hypothèse sont normalisables et l'on obtient un théorème de normalisation.

La formule $t = t$ (même si $\perp\!\!\!\perp \neq \emptyset$) est réalisée par n'importe quel terme, tandis que $\perp \equiv (\lambda x \lambda y.x = \lambda x \lambda y.y)$ n'est réalisée que par les termes de la forme $\mu\alpha.c$ avec $c \in \perp\!\!\!\perp$.

En réalisabilité classique, la formule $\forall t, u.\neg\neg t = u \to t = u$ est réalisée. Un réalisateur est

$$d = \lambda x, y.\lambda z.\mu\alpha.(z \star ((\lambda z'.\mu\beta.(z' \star \alpha)) \cdot \gamma))$$

car, si $(t \cdot u \cdot a \cdot e) : (\forall t, u.\neg\neg t = u \to t = u \Vdash)$ avec $a :\Vdash \neg\neg t = u$ et $e : t = u \Vdash$, alors

$$d \star (t \cdot u \cdot a \cdot e) \stackrel{*}{\to} a \star ((\lambda z'.\mu\beta.(z' \star e)) \cdot \gamma)$$

.

Comme $(\lambda z'.\mu\beta.(z' \star e)) \star a'' \cdot e'' \stackrel{*}{\to} a'' \star e$ quelque soit $a'' :\Vdash t = u$ et $e'' : \perp \Vdash$ et que $a'' :\Vdash t = u$ entraîne $a'' \star e \in \perp\!\!\!\perp$, on a $(\lambda z'.\mu\beta.z' \star e) :\Vdash \neg t = u$.

Comme $a :\Vdash \neg\neg t = u$, on obtient $a \star (\lambda z'.\mu\beta.z' \star e) \cdot \gamma \in \perp\!\!\!\perp$. D'où le résultat car $\perp\!\!\!\perp$ est clos par réduction inverse.

1) Démontrer formellement en Coq que $d :\Vdash \forall t, u.\neg\neg t = u \to t = u$.

2) Montrer formellement que $\vdash A$ implique l'existence d'un b tel que $b :\Vdash A$.

3) Montrer informellement que l'axiome du choix n'est pas réalisable.

4) On change maintenant la réalisabilité des quantificateurs comme suit

$$\begin{array}{rll} e : t = u \Vdash & \text{ssi} & t \neq_\beta u \\ a \cdot e : A \to B \Vdash & \text{ssi} & a :\Vdash A \text{ et } e : B \Vdash \\ e : \forall x.A \Vdash & \text{ssi} & \text{il existe } t \text{ tel que } e : A[t/x] \Vdash \\ e : \exists x.A \Vdash & \text{ssi} & \text{pour tout } t \text{ on a } e : A[t/x] \Vdash \\ & & \\ a :\Vdash A & \text{ssi} & \text{pour tout } e : A \Vdash, \text{ on a } a \star e \in \perp\!\!\!\perp \end{array}$$

Montrer informellement que l'axiome du choix est alors réalisable (à supposer que l'axiome du choix soit vrai dans le méta-language).

Références

[1] B. Barras, *A formalisation of the Calculus of Construction*, contribution Coq, http ://coq.inria.fr/contribs/coq-in-coq.html, 1997.

[2] M. Parigot, *$\lambda\mu$-calculus : An Algorithmic interpretation of classical natural deduction*, LPAR'92.

[3] J.-L. Krivine, *Typed lambda-calculus in classical Zermelo-Fraenkel set theory*, Arch. Math. Logic 40 (2001) 3, 189-205 (cf aussi http ://www.pps.jussieu.fr/ krivine/articles/lc2000.pdf).

[4] P.-L. Curien, H. Herbelin, *The duality of computation*, ICFP'00.