Décidabilité de l'Égalité dans les Catégories avec Familles

Samuel Mimram

Datavetenskap
Chalmers tekniska högskola
Göteborg, Sverige

10 mai 2004

- Se fonde sur des théories élaborées et étudiées : λ-calcul avec types dépendants, catégories
  - Cette marge est trop petite
- Très syntaxisant
  - Restons calme
- L'intérêt est surtout dans les détails techniques
  - Attardons-nous sur des points précis
- Les motivations sont principalement métaphysiques
  - Donnons les principales caractéristiques plutôt que les définitions formelles

Définition

Une catégorie est définie par:

- objets : A, B, C, $\ldots$
- morphismes (flèches) : $f : A \to B$, $\ldots$
- identité : $\mathrm{id}_A : A \to A$
- composée : $A \xrightarrow{f} B \xrightarrow{g} C : g \circ f : A \to C$

Tout se passe bien :

- $\mathrm{id} \circ f = f \circ \mathrm{id} = f$
- $h \circ (g \circ f) = (h \circ g) \circ f$

Quelques exemples

- **Set** : ensembles + fonctions entre
- **Grp** : groupes + homomorphismes de groupes
  - $(\mathcal{E}, \star, 1)$
    - associatif : $a \star (b \star c) = (a \star b) \star c$
    - élément neutre : $a \star 1 = 1 \star a = a$
    - inverse : $\forall a, \exists b, a \star b = b \star a = 1$ (unique)
  - morphismes : $\varphi : (\mathcal{E}, \star, 1_\star) \to (\mathcal{F}, \cdot, 1_.)$
    - $\varphi(a \star b) = \varphi(a) \star \varphi(b)$
    - $\varphi(a^{-1}) = \varphi(a)^{-1}$ et $\varphi(1_\star) = 1_.$ (inutile ici)
- **Cat** : categories + foncteurs
- types + λ-termes

Quelques exemples

- **Set** : ensembles + fonctions entre
- **Grp** : groupes + homomorphismes de groupes
- **Cat** : categories + foncteurs
- types + λ-termes
  - $\mathrm{id} : \lambda x.x : A \to A$
  - $\circ$: application
    $M : B \to C \qquad N : A \to B \qquad N \circ M \equiv \lambda x.M(Nx) : A \to C$

Le produit

Illustration de l'abstraction

$$a \in A \qquad b \in B \qquad (a,b) \in A \times B$$

$$\begin{aligned} \pi_A(a,b) &= a \\ \pi_A(a,b) &= b \end{aligned}$$

$$\begin{tikzcd} A \times B \arrow[r, "\pi_B"] \arrow[d, "\pi_A"'] & B \\ A & \end{tikzcd}$$

Le produit

Illustration de l'abstraction

$a \in A \qquad b \in B \qquad (a, b) \in A \times B$

$$\begin{aligned} \pi_A(a, b) &= a \\ \pi_A(a, b) &= b \end{aligned}$$

$$\begin{tikzcd}
Q \arrow[dr, dashed, "\langle f, g \rangle"] \arrow[drr, bend left, "g"] \arrow[ddr, bend right, "f"'] & & \\
& P \arrow[r, "\pi_B"] \arrow[d, "\pi_A"] & B \\
& A &
\end{tikzcd}$$

On veut montrer la décidabilité de l'égalité dans la théorie des catégories avec familles

- Équivalence de la théorie des cwf et des LF (λ-calcul avec types dépendants)
- Décidabilité de l'égalité dans LF

Motivations :

- Sémantique catégorique
- Résultat de base de décidabilité

- Classes syntaxiques des objets manipulés :
  - termes : $M, N \quad ::= \quad x \quad | \quad \lambda x.M \quad | \quad \mathrm{App}_{\Pi x:A.B}(M, N)$
  - types : $A, B \quad ::= \quad \star \quad | \quad \mathrm{El}\ M \quad | \quad \Pi x : A.B$
  - contextes : $\Gamma \quad ::= \quad \diamond \quad | \quad \Gamma, x : A$
- *Types dépendants*
  - Générateur de listes de 0 le longueur n :

$$\begin{array}{rcl} \Pi n : \mathrm{Nat} & . & \mathrm{List}(n) \\ n & \mapsto & [0; 0; \ldots; 0] \end{array}$$

- Règles de bonne formation (présentation propositionnelle : $\Gamma \vdash$, $\Gamma \vdash A$, $\Gamma \vdash M : A$, etc.)

- Classes syntaxiques des objets manipulés :
  - termes : $M, N \quad ::= \quad x \quad | \quad \lambda x.M \quad | \quad \mathrm{App}_{\Pi x:A.B}(M, N)$
  - types : $A, B \quad ::= \quad \star \quad | \quad \mathsf{El}\ M \quad | \quad \Pi x : A.B$
  - contextes : $\Gamma \quad ::= \quad \diamond \quad | \quad \Gamma, x : A$
- *Types dépendants*
- Règles de bonne formation (présentation propositionnelle : $\Gamma \vdash$, $\Gamma \vdash A$, $\Gamma \vdash M : A$, etc.)
  - contexte : liste de (variable, type), où un type peut dépendre de ce qui précède
  - $$\frac{\Gamma \vdash \qquad \Gamma \vdash A \qquad \Gamma, x : A \vdash B \qquad \Gamma, x : A \vdash M : B}{\Gamma \vdash \lambda x.M \ : \ \Pi x{:}A.B}$$

- Classes syntaxiques des objets manipulés :
  - termes : $M, N \quad ::= \quad x \quad | \quad \lambda x.M \quad | \quad \mathrm{App}_{\Pi x:A.B}(M, N)$
  - types : $A, B \quad ::= \quad \star \quad | \quad \mathsf{El}\ M \quad | \quad \Pi x : A.B$
  - contextes : $\Gamma \quad ::= \quad \diamond \quad | \quad \Gamma, x : A$
- *Types dépendants*
- Règles de bonne formation (présentation propositionnelle : $\Gamma \vdash$, $\Gamma \vdash A$, $\Gamma \vdash M : A$, etc.)
- Égalité typée ($\Gamma \vdash A = B$, $\Gamma \vdash M = N : A$)
  - Exemple :
$$2 \neq 0 \quad : \quad \mathbb{Z}$$
    mais
$$2 = 0 \quad : \quad \mathbb{Z}/2\mathbb{Z}$$

- Classes syntaxiques des objets manipulés :
  - termes : $M, N \quad ::= \quad x \quad | \quad \lambda x.M \quad | \quad \mathrm{App}_{\Pi x:A.B}(M, N)$
  - types : $A, B \quad ::= \quad \star \quad | \quad \mathsf{El}\ M \quad | \quad \Pi x : A.B$
  - contextes : $\Gamma \quad ::= \quad \diamond \quad | \quad \Gamma, x : A$
- *Types dépendants*
- Règles de bonne formation (présentation propositionnelle : $\Gamma \vdash$, $\Gamma \vdash A$, $\Gamma \vdash M : A$, etc.)
- Égalité typée ($\Gamma \vdash A = B$, $\Gamma \vdash M = N : A$)
  - Cette égalité est essentiellement la $\beta\eta$-equivalence sur les termes typables :
    - β : $(\lambda x.M)\, N = M\, [N/x]$
    - η : $M = \lambda x.Mx$

- Catégorie des contextes **C**
- Termes : foncteur $(\mathrm{Term}, \mathrm{Type}) : \mathbf{C}^{\mathrm{op}} \to \mathbf{Fam}$
  - contexte dans **LF** : $[x_1 : A_1; x_2 : A_2; \ldots; x_n : A_n]$
  - type = ensemble de termes / type dépendant = famille
  - *famille* : ensemble d'ensembles indexé par un ensemble : $(B(x))_{x \in A}$
- Combinateurs :
  - p : début du contexte $(\to [x_1 : A_1; x_2 : A_2; \ldots; x_{n-1} : A_{n-1}])$
  - q : dernier élément du contexte $(\to x_n)$
  - , : ajout d'un élément au contexte
  - λ, App, $\Pi(A, B)$: abstraction, application, type produit
  - Star, Elem : cf. El
- Substitution : morphismes de **C**
  - $\langle \gamma, M \rangle$: ajout d'une substitution
- Règles de bonne formation

$$[\![\lambda xy.x]\!] \equiv \lambda(\lambda(q[p]))$$

Explication de $[\![\lambda xy.x]\!] \equiv \lambda(\lambda(q[p]))$

$$\dfrac{\dfrac{x : A, y : B \vdash x : A}{x : A \vdash \lambda y.x : B \to A}}{\diamond \vdash \lambda x.\lambda y.x : A \to B \to A} \qquad \dfrac{\dfrac{A, B \vdash q[p] : A}{A \vdash \lambda(q[p]) : B \to A}}{\diamond \vdash \lambda(\lambda(q[p])) : A \to B \to A}$$

- Pas besoin de variables

Intérêt de la sémantique catégorique

- Permet de comprendre la notion de catégorie
- Pas d'α-conversion ni même de notion de variable
  - on veut : $\lambda xy.x \equiv \lambda yz.y$
  - ($\sim$ De Bruijn)
- La substitution n'est pas une meta-opération
  - $\sim$ substitutions explicites
- Définissable dans **Gat**
  - Modèle de la théorie des types dépendants internalisable

Ce qu'on veut d'une théorie équationnelle :

- Des sortes (types) et des termes dépendants construits par des symboles donnés
- Une égalité « naturelle »
- Exprimé à l'aide de règles de dérivation
- Dans quoi définir **Gat** ?

L'exemple des entiers naturels

Définition de $(\mathbb{N}, +)$:

- $\diamond \vdash \mathrm{Nat}$
- $\diamond \vdash \mathrm{O} : \mathrm{Nat}$
- $n : \mathrm{Nat} \vdash \mathrm{S}(n) : \mathrm{Nat}$
- $n : \mathrm{Nat}, m : \mathrm{Nat} \vdash \mathrm{Add}(n, m) : \mathrm{Nat}$
- $$\begin{array}{rcl} n : \mathrm{Nat} & \vdash & \mathrm{Add}(\mathrm{O}, n) = \mathrm{O} \\ n : \mathrm{Nat}, m : \mathrm{Nat} & \vdash & \mathrm{Add}(\mathrm{S}(n), m) = \mathrm{Add}(n, \mathrm{S}(m)) \end{array}$$

On peut déduire :

- $\diamond \vdash \mathrm{S}(\mathrm{O}) : \mathrm{Nat} \qquad \diamond \vdash \mathrm{Add}(\mathrm{O}, \mathrm{O}) = \mathrm{O}$
- Si $n = n'$ alors $\mathrm{S}(n) = \mathrm{S}(n')$

- Interprétation des classes de termes syntaxiques (contextes, termes, types) dans des ensembles (ou autre) qui respecte les axiomes de bonne formation
  - e.g. : $\mathcal{I}(\mathrm{Nat}) \equiv \mathbb{N} \quad \mathcal{I}(\mathrm{O}) \equiv 0 \quad \mathcal{I}(\mathrm{S}(n)) \equiv \mathcal{I}(n) + 1$ etc.
- Tout ce qui est vrai dans la théorie est vrai dans les modèles
- Les termes de la théorie forment un modèle (*term-model*)
- (on veut que le term-model soit initial dans la catégorie des modèles)

Très intuitif quand on compare de loin les théories, beaucoup plus subtil en pratique.

Deux approches sont possibles :

- Sémantique (catégorie des modèles) : plus « propre », on veut avant tout parler de la catégorie des modèles
- Syntaxique : plus « simple », se fait avec deux interprétations ($\sim$ réécritures)

- Termes :

$$
\begin{aligned}
[\![x_i]\!]^{\Gamma}_{\mathsf{Cwf}_{\mathsf{LF}}} &\equiv \begin{cases} q_{[\![A_n]\!]^{\Gamma'}_{\mathsf{Cwf}_{\mathsf{LF}}}} & \text{if } i = n \\ [\![x_i]\!]^{\Gamma'}_{\mathsf{Cwf}_{\mathsf{LF}}}\left[p_{[\![A_n]\!]^{\Gamma'}_{\mathsf{Cwf}_{\mathsf{LF}}}}\right] & \text{else} \end{cases} \\
[\![\lambda x_A.M]\!]^{\Gamma}_{\mathsf{Cwf}_{\mathsf{LF}}} &\equiv \lambda_A\left([\![M]\!]^{\Gamma,x:A}_{\mathsf{Cwf}_{\mathsf{LF}}}\right) \\
[\![\mathrm{App}_{\Pi x:A.B}(M,N)]\!]^{\Gamma}_{\mathsf{Cwf}_{\mathsf{LF}}} &\equiv \mathrm{App}_{\Pi\left([\![A]\!]^{\Gamma}_{\mathsf{Cwf}_{\mathsf{LF}}},[\![B]\!]^{\Gamma,x:A}_{\mathsf{Cwf}_{\mathsf{LF}}}\right)}\left([\![M]\!]^{\Gamma}_{\mathsf{Cwf}_{\mathsf{LF}}},[\![N]\!]^{\Gamma}_{\mathsf{Cwf}_{\mathsf{LF}}}\right)
\end{aligned}
$$

- Même chose pour les contextes et les types

- Interprétation partielle ($\rightarrow$ égalité de Kleene) à cause des indices (e.g. $\mathrm{App}_{\Pi x:A.B}(M, N)$)
  - $a \cong b$: a est défini ssi b l'est et dans ce cas $a = b$
  - interprétation p ou q dans le contexte vide ?

- Interprétation partielle ($\to$ égalité de Kleene) à cause des indices (e.g. $\mathrm{App}_{\Pi x:A.B}(M,N)$)
- Principale difficulté : mettre en relation les substitutions (sémantique / syntaxique)
  - en λ : $(\lambda x.\lambda y.x)\,M = (\lambda y.x)\,[M/x] \equiv \lambda y.M$

- Interprétation partielle ($\rightarrow$ égalité de Kleene) à cause des indices (e.g. $\mathrm{App}_{\Pi x:A.B}(M, N)$)
- Principale difficulté : mettre en relation les substitutions (sémantique / syntaxique)
  - en λ : $(\lambda x.\lambda y.x)\, M = (\lambda y.x)\,[M/x] \equiv \lambda y.M$
  - cwfs :

$$
\begin{array}{rcl}
\mathrm{App}\,(\lambda\,(\lambda\,(q\,[p]))\,, M) & = & \lambda\,(q\,[p])\,[\langle \mathrm{id}, M\rangle] \\
& = & \lambda\,(q\,[p]\,[\langle\langle \mathrm{id}, M\rangle \circ p, q\rangle]) \\
& = & \lambda\,(q\,[p \circ \langle\langle \mathrm{id}, M\rangle \circ p, q\rangle]) \\
& = & \lambda\,(q\,[\langle \mathrm{id}, M\rangle \circ p]) \\
& = & \lambda\,(q\,[\langle \mathrm{id}, M\rangle]\,[p]) \\
& = & \lambda\,(M\,[p])
\end{array}
$$

- Typable : normalisable : on sait décider de la β-convertibilité grâce à Church-Rosser (il suffit de réduire à la forme normale)
- Problème : l'égalité est ici la $\beta\eta$-conversion
- Pour se ramener à la β-convertibilité on utilise l'*incarnation* ($\eta_A(M)$) qui pré-calcule les η-expansions en se fondant sur le type :
  - $\eta_\star(M) = \eta_{\mathrm{El}\ N}(M) = M$
  - $\eta_{\Pi x:A.B}(M) = \lambda z.\eta_{B[\eta_A(z)/x]}(M\ \eta_A(z))$
- On veut prouver (en gros) : $M = N : A \Leftrightarrow \eta_A(M) =_\beta \eta_A(N)$
- Tout est syntaxique et pourtant la preuve passe par des per-modèles par ce qu'on a pas : $\eta_A(MN) = \eta_A(M)\eta_A(N)$
  - Il reste des choses à comprendre

- Les cwfs sont définissables dans **Gat**
- **Cwf** et **Gat** sont équivalentes
- On peut représenter la théories des types à l'intérieur d'elle-même

- Devrait être prouvable en COQ
  - Utilisation de setoïdes (ensemble + relation d'équiv)
    - égalité intentionnelle (COQ) : on peut prouver l'égalité
    - extensionnelle (catégoires) : $\forall x, f(x) = g(x)$
- Très rébarbatif $\rightarrow$ automatisable ?