Sémantiques dénotationnelles du Calcul des Constructions

Alexandre Miquel

22 novembre 2000

Séminaire du Projet LEMME
INRIA Sophia-Antipolis

Introduction

Le Calcul des Constructions: formalisme à la base su système Coq

Un formalisme riche dans lequel on peut exprimer
des programmes, des spécifications, des preuves de correction.

Utilisé pour le développement de logiciels critiques
$\Rightarrow$ Nécessité d'une sémantique bien comprise

Deux grandes familles de sémantiques:

Sémantique opérationnelle $\Rightarrow$ insiste sur le procédé de calcul (intensionnel)
(adaptée à l'étude des langages de programmation)

Sémantique dénotationnelle $\Rightarrow$ insiste sur le résultat du calcul (extensionnel)
(adaptée à l'étude logique des systèmes de preuves)

Le Calcul des Constructions (CC)

[T. Coquand, G. Huet]

Syntaxe:

$$\begin{array}{rcll} M, N, T, U & ::= & x & \text{(variable)} \\ & | & \textsf{Prop},\ \textsf{Type} & \text{(sortes)} \\ & | & \Pi x{:}T\,.\,U & \text{(produit dépendant)} \\ & | & \lambda x{:}T\,.\,M & \text{(abstraction)} \\ & | & (M\ N) & \text{(application)} \end{array}$$

[**Notation:** $T \to U \ \equiv\ \Pi x{:}T\,.\,U$ lorsque $x \notin FV(U)$]

Règle de β-réduction: $((\lambda x{:}T\,.\,M)\ N) \to_\beta M\{x := N\}$

Propriétés:
- Confluence de la β-réduction
- Préservation du typage par la β-réduction

Cohérence et normalisation forte

le Calcul des Constructions est:
- cohérent (pas de preuve de $\Pi A:\mathsf{Prop}\,.\,A$)
- fortement normalisant (les calculs terminent)

La preuve de cohérence usuelle repose sur la preuve de normalisation forte (NF)

Iconvénients:
- preuve de NF difficile (mélange syntaxe et sémantique)
- fait intervenir un modèle caché [T. Altenkirch, B. Werner]
- ne permet pas d'établir la cohérence d'axiomes supplémentaires

$\Rightarrow$ Introduire un modèle explicite pour prouver la cohérence de manière directe

- cohérence d'axiomes supplémentaires (ex: tiers-exclus)
- idées réutilisables pour la NF: modèle $\rightsquigarrow$ prémodèle [G. Dowek, B. Werner]

Première partie

Le modèle classique

Le dictionnaire

[T. Coquand]

Remarque: Les termes (clos en forme normale) de CC sont soit:

- des fonctions $\lambda x : T \,.\, M$
- des types fonctionnels $\Pi x : T \,.\, U$
- des sortes (types des types) Prop, Type

Théorie des Types	Théorie des Ensembles
fonction (λ)	fonction ($\mapsto$)
type	ensemble
type fonctionnel	ensemble de fonctions
sorte (type de types)	ensemble d'ensembles
$M : T$	$[\![M]\!] \in [\![T]\!]$
relation de typage	relation d'appartenance

Schéma général de l'interprétation

$$
\begin{array}{rcl}
[\![\lambda x : T \,.\, M]\!] & = & (x_{\in [\![T]\!]} \mapsto [\![M]\!]) \\
[\![\Pi x : T \,.\, U]\!] & = & \prod_{x \in [\![T]\!]} [\![U]\!] \\
[\![\mathsf{Prop}]\!], [\![\mathsf{Type}]\!] & = & ???
\end{array}
$$

Remarque: Ce schéma est moins trivial qu'il n'en a l'air:

- Théorie des Types: fonction = programme (codage intensionnel)

$$\mathsf{succ} \quad = \quad \lambda n \,.\, n+1 \quad = \quad \lambda n \,.\, \lambda f \,.\, \lambda x \,.\, (f\ (n\ f\ x))$$

- Théorie des Ensembles: fonction = graphe (codage extensionnel)

$$\mathsf{succ} \quad = \quad (n_{\in \mathbb{N}} \mapsto n+1) \quad = \quad \{(0,1);\ (1,2);\ (2,3);\ldots\}$$

Propriétés de clôture des sortes

- Formation d'un type dans Prop:

$$\frac{\vdash T : \mathsf{Prop} \quad x : T \vdash U : \mathsf{Prop}}{\vdash \Pi x\!:\!T\,.\,U : \mathsf{Prop}} \qquad \frac{\vdash T : \mathsf{Type} \quad x : T \vdash U : \mathsf{Prop}}{\vdash \Pi x\!:\!T\,.\,U : \mathsf{Prop}}\,(*)$$

- Formation d'un type dans Type:

$$\frac{}{\vdash \mathsf{Prop} : \mathsf{Type}} \qquad \frac{\vdash T : \mathsf{Prop}}{\vdash T : \mathsf{Type}} \qquad \frac{\vdash T : \mathsf{Type} \quad x : T \vdash U : \mathsf{Type}}{\vdash \Pi x\!:\!T\,.\,U : \mathsf{Type}}$$

Difficulté: circularité liée à l'imprédicativité de Prop $(*)$

Exemple: $\Pi A\!:\!\mathsf{Prop}\,.\,A \to A$ est de type Prop!

Modélisation de l'imprédicativité (Prop)

$$\frac{\vdash T : s \quad x : T \vdash U : \mathsf{Prop}}{\vdash \Pi x : T \, . \, U : \mathsf{Prop}} \; s \in \{\mathsf{Prop}; \; \mathsf{Type}\}$$

Traduction ensembliste: Pour toute famille d'ensembles $(U_x)_{x \in T}$, on a

$$(\forall\, x \in T \quad Prop(U_x)) \quad \Rightarrow \quad Prop\left(\prod_{x \in T} U_x\right)$$

Problème: Comment définir le prédicat $Prop(X)$?

Une solution simple: $Prop(X) \quad \equiv \quad X$ a au plus un élément

$\Rightarrow$ Sémantique de la proof-irrelevance

Sémantique de la proof-irrelevance

$$\textbf{Proposition} \quad = \quad \text{ensemble vide ("faux") ou singleton ("vrai")}$$

Déf: $[\![\mathsf{Prop}]\!] \;=\; \{\varnothing;\ \{\mathsf{prf}\}\}$ $\qquad$ (prf = ensemble représentant toutes les preuves)

Lemme: — Soient T un ensemble et $(U_x)_{x\in T}$ une famille d'ensembles indicée par T. Si $U_x \in [\![\mathsf{Prop}]\!]$ pour tout $x \in T$, alors

$$\prod_{x\in T} U_x \;=\; \begin{cases} \varnothing & \text{si } U_x = \varnothing \text{ pour au moins un } x \in T \\ \{(x_{\in T} \mapsto \mathsf{prf})\} & \text{si } U_x = \{\mathsf{prf}\} \text{ pour tout } x \in T \end{cases}$$

$\Rightarrow$ On identifiera prf à toutes les fonctions constantes $(x_{\in T} \mapsto \mathsf{prf})$.

Interprétation de la sorte Type

$$\frac{}{\vdash \mathsf{Prop} : \mathsf{Type}} \qquad \frac{\vdash T : \mathsf{Prop}}{\vdash T : \mathsf{Type}} \qquad \frac{\vdash T : \mathsf{Type} \quad x : T \vdash U : \mathsf{Type}}{\vdash \Pi x : T \, . \, U : \mathsf{Type}}$$

Déf: $[\![\mathsf{Type}]\!] = HF$ (ensemble des ensembles héréditairement finis)

En supposant que $\mathsf{prf} \in HF$, on a :

- $\emptyset \in HF$ et $\{\mathsf{prf}\} \in HF$, donc $[\![\mathsf{Prop}]\!] \subset HF$
- $[\![\mathsf{Prop}]\!] = \{\emptyset; \{\mathsf{prf}\}\} \in HF$
- HF est clos par les produits $\prod_{x \in T} U_x$

Modèle proof-irrelevant: $\boxed{\mathcal{M} = HF \cup \{HF\}}$

Interprétation des termes

Interprétation $[\![M]\!]_\rho$ paramétrée par une valuation $\rho \in (\mathrm{Var} \to \mathcal{M})$.

$$
\begin{array}{rcl}
[\![x]\!]_\rho & = & \rho(x) \\
[\![\mathsf{Prop}]\!]_\rho & = & \{\varnothing;\ \{\mathsf{prf}\}\} \\
[\![\mathsf{Type}]\!]_\rho & = & HF \\
[\![\Pi x : T \,.\, U]\!]_\rho & = & \prod_{\alpha \in [\![T]\!]_\rho} [\![U]\!]_{(\rho; x \leftarrow \alpha)} \\
[\![\lambda x : T \,.\, M]\!]_\rho & = & \left(\alpha_{\in [\![T]\!]_\rho} \mapsto [\![M]\!]_{(\rho; x \leftarrow \alpha)}\right) \\
[\![(M\ N)]\!]_\rho & = & [\![M]\!]_\rho([\![N]\!]_\rho)
\end{array}
$$

Valuations adaptées à Γ: $[\![\Gamma]\!] = \left\{\rho;\quad \forall (x : T) \in \Gamma \quad \rho(x) \in [\![T]\!]_\rho\right\}$

Validité: $\boxed{\text{Si } \ \Gamma \vdash M : T \ \text{ et } \ \rho \in [\![\Gamma]\!], \ \text{alors } \ [\![M]\!]_\rho \in [\![T]\!]_\rho}$

Quelques dénotations de termes clos

$$
\begin{array}{rcll}
[\![\Pi A : \mathsf{Prop}\,.\,A]\!] & = & \varnothing & \text{(Cohérence de CC)} \\
[\![\lambda A, B : \mathsf{Prop}\,.\,A \wedge B]\!] & = & \text{table de vérité du ``et''} & \\
[\![\lambda A, B : \mathsf{Prop}\,.\,A \vee B]\!] & = & \text{table de vérité du ``ou''} & \\
\ldots & & \ldots & \\
[\![\Pi A : \mathsf{Prop}\,.\,A \vee \neg A]\!] & = & \{\mathsf{prf}\} & \text{(Cohérence du tiers-exclus dans } \mathsf{Prop}\text{)} \\
[\![0 \neq_{\mathsf{nat}} 1]\!] & = & \varnothing & (0 \neq 1 \text{ n'est pas prouvable dans CC)} \\
[\![0 =_{\mathsf{nat}} 1]\!] & = & \{\mathsf{prf}\} & \text{(car } 0 = 1 \text{ dans le modèle!)}
\end{array}
$$

Mais aussi:

- extensionnalité propositionnelle $(A \Leftrightarrow B) \quad \Rightarrow \quad (A =\!=_{\mathsf{Prop}} B)$
- extensionnalité fonctionnelle $(\forall x\ f(x) = g(x)) \quad \Rightarrow \quad f = g$
- proof-irrelevance, axiome du choix, axiome K de Streicher, etc.

Modèle classique: le bilan

- modèle minimal (dénombrable: même "taille" que la syntaxe)
- montre la cohérence de CC (+ beaucoup d'axiomes) à peu de frais
- disparition du calcul $\Rightarrow$ plus de problème de terminaison!

Mais surtout: montre les limitations de CC (dans sa version de base)

- impossible de distinguer les preuves d'une même proposition
- pas de vraie structure infinie (dans le modèle, tous les types sont finis)

$\Rightarrow$ Suggère naturellement des extensions de la syntaxe

- ▶ une nouvelle sorte **Set** pour pallier les limitations de **Prop**
- ▶ une hiérarchie d'univers Type_i au dessus de la sorte **Type**
- ▶ des types inductifs primitifs pour encoder les structures de données

Deuxième partie

Set, Type_2, Type_3 et C^{ie}

La hiérarchie d'univers prédicatifs

Famille infinie d'univers prédicatifs: $\mathsf{Type}_1 (= \mathsf{Type}), \mathsf{Type}_2, \ldots, \mathsf{Type}_i, \ldots$

$$\frac{}{\vdash \mathsf{Type}_{i-1} : \mathsf{Type}_i} \qquad \frac{\vdash T : \mathsf{Type}_{i-1}}{\vdash T : \mathsf{Type}_i} \qquad \frac{\vdash T : \mathsf{Type}_i \quad x : T \vdash U : \mathsf{Type}_i}{\vdash \Pi x : T \,.\, U : \mathsf{Type}_i}$$

Idée: Construire une famille d'ensembles $(\mathcal{U}_i)_{i \geq 1}$ telle que

- $\mathcal{U}_{i-1} \in \mathcal{U}_i$
- $\mathcal{U}_{i-1} \subset \mathcal{U}_i$
- $\mathcal{U}_i$ est clos par les produits $\prod_{x \in T} U_x$

Remarque: on conserve $[\![\mathsf{Prop}]\!] = \{\varnothing;\ \{\mathsf{prf}\}\}$ et $[\![\mathsf{Type}_1]\!] = HF = \mathcal{U}_1$.

$\Rightarrow$ Mais pour $\mathcal{U}_2$, $\mathcal{U}_3$, etc., on utilise des cardinaux inaccessibles

Cardinaux inaccessibles et ZF-modèles

Un ZF-modèle (ou univers) est un ensemble $\mathcal{U}$ tel que:

1. $\mathcal{U}$ est transitif $\quad (i.e.\ X \in \mathcal{U} \;\Rightarrow\; X \subset \mathcal{U})$
2. $\mathbb{N} \in \mathcal{U}$
3. $X \in \mathcal{U} \;\Rightarrow\; \mathfrak{P}(X) \in \mathcal{U}$
4. $\forall (Y_x)_{x\in X} \quad X \in \mathcal{U},\ Y_x \in \mathcal{U} \;\Rightarrow\; \bigcup_{x\in X} Y_x \in \mathcal{U}$

$\boxed{\text{Cardinal inaccessible } \lambda \quad \leftrightarrow \quad \text{ZF-modèle } \mathcal{U}_\lambda}$ avec $\lambda < \mu \;\Rightarrow\; \mathcal{U}_\lambda \in \mathcal{U}_\mu$

D'où: $[\![\mathsf{Type}_1]\!] = HF, \quad [\![\mathsf{Type}_2]\!] = \mathcal{U}_{\lambda_1}, \quad [\![\mathsf{Type}_3]\!] = \mathcal{U}_{\lambda_2}, \quad$ etc.

$\Rightarrow$ suppose l'existence d'une infinité de cardinaux inaccessibles (Axiome!)

Faut-il avoir peur des cardinaux inaccessibles ?

$\Rightarrow$ Un moyen simple et standard pour étendre la puissance théorique de ZF
Utilisé déjà chez Tarski pour construire de grands modèles

Question: Peut-on s'en passer ?

- Pour CC + **Type**$_i$ (sans **Set** ni inductifs): Oui [P.-A. Melliès, B. Werner]

$\Rightarrow$ Construction du modèle **beaucoup** plus difficile!

- Pour le Calcul des Constructions Inductives (Coq): Non [P. Azcel, B. Werner]

$\Rightarrow$ Chaque univers de Coq a (pratiquement) la puissance d'un ZF-modèle!

Set: le faux-jumeau de Prop

Deux univers imprédicatifs Prop et Set (de type Type_1):

$$\frac{\vdash T : s \quad x : T \vdash U : \mathsf{Prop}}{\vdash \Pi x : T \,.\, U : \mathsf{Prop}} \qquad\qquad \frac{\vdash T : s \quad x : T \vdash U : \mathsf{Set}}{\vdash \Pi x : T \,.\, U : \mathsf{Set}}$$

Différence: élimination forte = élimination d'un inductif de Set vers Type_i

```
<[_:nat]Prop>Cases n of O => True | (S _) => False end.
```

(Set) (Type)

(Équivaut à supposer que $0 \neq 1$ dans **nat** : Set)

– Interprétation de Prop (presque) inchangée
– Nécessite de nouveaux outils pour interpréter Set (ω-sets, PER)

Réalisabilité et ω-sets

ω-set: $X = (|X|, \Vdash_X)$ avec:

1. X ensemble quelconque (support)
2. $\Vdash_X \subset \omega \times |X|$ (relation de réalisabilité) t.q: $\forall x \in |X| \quad \exists n \in \omega \quad n \Vdash_X x$

Produit d'une famille de ω-sets: X ω-set, $(Y_x)_{x \in |X|}$ famille de ω-sets.
On définit le ω-set produit $\Pi(x \in X; Y_x)$ par:

$$|\Pi(x \in X; Y_x)| \equiv \left\{ f \in \prod_{x \in |X|} |Y_x|; \quad f \text{ a au moins un réalisateur } n \right\}$$

$$n \Vdash_{\Pi(x \in X; Y_x)} f \equiv \forall p, x \quad p \Vdash_X x \Rightarrow n \cdot p \Vdash_{Y_x} f(x)$$

Notation: $n \cdot p \equiv$ "n-ième fonction récursive appliquée à p"

PER et ω-sets modestes

X ω-set modeste si: $\quad \forall\, n, x, x' \quad n \Vdash_X x \quad \text{et} \quad n \Vdash_X x' \quad \Rightarrow \quad x = x'$
($i.e.$ chaque fonction récursive n réalise au plus un élément du support)

PER $\quad = \quad$ Relation d'équivalence partielle sur ω ($i.e.$ symétrique + transitive)

À une équivalence de catégories près: $\quad \boxed{\omega\text{-set modeste} \quad \simeq \quad \text{PER}}$

Propriété: $(Y_x)_{x \in |X|}$ famille de ω-sets.

$$(\forall\, x \in |X| \quad Y_x\ \omega\text{-set modeste}) \quad \Rightarrow \quad \Pi(x \in X ; Y_x)\ \omega\text{-set modeste}$$

$\Rightarrow$ La catégorie des ω-sets modestes (représentée par l'ensemble des PER) est le bon candidat pour interpréter **Set**.

Interprétation de CC dans la catégorie des ω-sets

On interprète:

- les types par des ω-sets;
- **Set** par l'ensemble des PER
- **Prop** par $\{\varnothing;\ \omega \times \omega\}$ (PER vide/PER plein)
- **Type**$_i$ par l'ensemble des ω-sets $\in \mathcal{U}_{\lambda_i}$

Remarques:

- $[\![\Pi A : \mathsf{Prop}\,.\,A]\!] = [\![\Pi A : \mathsf{Set}\,.\,A]\!] = \varnothing$ (Cohérence)
- dans le modèle, **Prop** est inclus dans **Set**;
- élimination forte validée par le modèle ($0 \neq 1$ dans **Set**)
- l'interprétation de **Set** ne comporte que des fonctions calculables
- s'étend facilement aux types inductifs: λ-sets [T. Altenkirch, B. Werner]

Troisième partie

Domaines

Sous-typage

$\Rightarrow$ Enrichir le système de types avec un jugement de sous-typage $\Gamma \vdash A \leq A'$

$$\frac{\vdash M : A \quad \vdash A \leq A'}{\vdash M : A'}\ (\textsc{Sub})$$

- Déjà une forme faible de sous-typage dans CC + Univers:

$$\vdash \mathsf{Prop} \leq \mathsf{Type}_1 \qquad \vdash \mathsf{Type}_i \leq \mathsf{Type}_{i+1}$$

- Polymorphisme à la ML: une forme particulière de sous-typage:

$$\vdash (\forall\, \alpha.\ \alpha \to \alpha) \leq \mathsf{int} \to \mathsf{int}$$

Question: peut-on interpréter simplement le sous-typage par une inclusion ?

Les limites de l'interprétation ensembliste

Sous typage dans les types fonctionnels:

$$\frac{\vdash B \leq B'}{\vdash A \to B \leq A \to B'}\ (\textsc{Cov}) \qquad \frac{\vdash A' \leq A}{\vdash A \to B \leq A' \to B}\ (\textsc{Contrav})$$

Problème: dans la théorie des ensembles

$$A \to \mathbb{N} \;\subset\; A \to \mathbb{Z} \quad (\textsc{Cov}) \qquad \textbf{mais} \qquad \mathbb{Z} \to B \;\not\subset\; \mathbb{N} \to B$$

Solutions: on peut interpréter le sous-typage

– à l'aide de coercions: modèles basés sur la théorie des catégories [M. Hoffman]
– sans coercions: modèles basés sur la théorie des domaines [A. Miquel]

La théorie des domaines

Motivation: construire un modèle non-syntaxique du λ-calcul [D. Scott]

Domaine de Scott $=$ ensemble ordonné & espace topologique (structure hybride)

$\Rightarrow$ basé sur la notion de fonction continue

Théorème. [Scott] — *Il existe un domaine D^ω tel que $D^\omega \simeq (D^\omega \to D^\omega)$.*

- Jadis à la mode pour la sémantique des langages fonctionnels
- Peu de littérature concernant les systèmes de types
  (modèles davantage ensemblistes/catégoriques)

$\Rightarrow$ modèle du système F dans les espaces cohérents [J.-Y. Girard, P. Taylor]
$\rightsquigarrow$ découverte de la logique linéaire

Modèles basés sur les espaces cohérents

Motivation: sémantique du Calcul des Constructions implicite [A. Miquel, 1998]

$=$ CC $+$ λ à la Curry $+$ lieur de types intersections ($\forall$) $+$ sous-typage

- Construction d'un modèle dans les espaces cohérents

[A. Miquel. A Model for Impredicative Type Systems, Universes, Intersection Types and Subtyping, Logic In Computer Science 2000]

Possibilités d'interprétation étendues:

- abstraction à la Curry
- sous-typage sans coercions
- polymorphisme à la ML
- imprédicativité triviale/non-triviale (K-espaces cohérents)

K-espaces cohérents

Une alternative à l'utilisation des ω-sets/PER

Idée: les PER sont des domaines cachés

$\Rightarrow$ Révéler leur structure de domaine dans le cadre des espaces cohérents

Intérêt: permet d'interpréter **Prop** et **Set** de manière modulaire

Construction du modèle paramétrée par un modèle K du λ-calcul

Suivant le choix du paramètre, on peut interpréter dans **Set**:

– des fonctions non calculables (ex: le test d'arrêt)
– la théorie classique des réels (*i.e.* non-constructive)
– la théorie des ensembles ZF

Conclusion

Preuves de cohérence relative

$\Rightarrow$ deux points de vue complémentaires:

- preuve de cohérence d'une théorie A dans une théorie B
- transformation syntaxique exprimable dans l'arithmétique

Un regard nouveau sur la syntaxe

- objets calculatoires/non-calculatoires
- sous-typage, annotations de type

Suggère des extensions de la syntaxe (Coq)

- extension du sous-typage (contravariance, inductifs)
- allègement du test de conversion
- proof-irrelevance dans le test de conversion