Calcul des Constructions Inductives
D.E.A. Programmation: Sémantique, Preuves et Langages
2003-2004

Calcul des Constructions Inductives

Christine Paulin-Mohring, Benjamin Werner,

Bruno Barras, Hugo Herbelin,

Jean-Christophe Filliâtre, Claude Marché

Table des matières

6 Sémantique du Calcul des Constructions Inductives **3**
- 6.1 Le Calcul des Constructions pur (CC) 3
  - 6.1.1 Puissance logique 3
  - 6.1.2 Puissance calculatoire 4
  - 6.1.3 Extensions incohérentes du Calcul des Constructions 4
- 6.2 Le Calcul des Constructions Étendu (ECC) 5
  - 6.2.1 Encodage de l'arithmétique 5
  - 6.2.2 Encodage de la théorie des ensembles de Zermelo 5
  - 6.2.3 Puissance logique 6
- 6.3 Calcul des Constructions Inductives (CCI) 6
  - 6.3.1 Modèles des types inductifs 6
- 6.4 Extensions cohérentes et incohérentes du Calcul des Constructions Inductives . . 6

Chapitre 6

Sémantique du Calcul des Constructions Inductives

6.1 Le Calcul des Constructions pur (CC)

La théorie du Calcul des Constructions pur se ramène essentiellement à celle du système F_ω par effacement des dépendances en les preuves dans les types (cf chapitre sur l'extraction).

Tous deux correspondent à une logique très faible, puisqu'en interprétant `Prop` de manière booléenne, toute formule admet un modèle fini. En revanche, tous deux correspondent à un langage de fonctions très riche puisqu'ils contiennent toutes les fonctions prouvablement totales dans l'arithmétique d'ordre supérieur.

6.1.1 Puissance logique

Le CC admet un modèle booléen, dans lequel `Prop` est interprété par l'ensemble a deux élément $\mathcal{B} = \{\mathsf{true}, \mathsf{false}\}$. L'imprédicativité se résume alors à une quantification sur $\mathcal{B}$. Les types de `Type` sont construits récursivement à partir de $\mathcal{B}$ et le produit fonctionnel. Dans le cas d'un produit $A \to T$ avec A une proposition et T un type, l'interprétation est celle de T[1]. L'interprétation s'applique à tous les jugements qui ne sont pas des jugements de preuve (on note X, P, Q et T, U pour les objets et types du niveau `Type` et x, t, u et A, B pour les objets et types[2] du niveau `Prop`).

$$\begin{array}{l}
[\![\mathtt{Prop}]\!] = \{\mathsf{true}, \mathsf{false}\} \\
[\![\forall x : A.U]\!] = [\![U]\!] \\
[\![\forall X : T.U]\!] = [\![T]\!] \to [\![U]\!] \\
[\![\forall X : T.B]\!]_\rho = \mathsf{true} \text{ ssi } [\![B]\!]_{\rho, X:=V} = \mathsf{true} \text{ pour tout } V \in [\![T]\!] \\
[\![\forall x : A.B]\!]_\rho = \mathsf{true} \text{ ssi } [\![A]\!]_\rho = \mathsf{false} \text{ ou } [\![B]\!]_\rho = \mathsf{true} \\
[\![\lambda x : A.P]\!]_\rho = [\![P]\!]_\rho \\
[\![\lambda X : T.P]\!]_\rho = V \in [\![T]\!] \mapsto [\![P]\!]_{\rho, X:=V} \\
[\![Pu]\!]_\rho = [\![P]\!]_\rho \\
[\![PQ]\!]_\rho = [\![P]\!]_\rho([\![Q]\!]_\rho) \\
[\![X]\!]_\rho = \rho(X)
\end{array}$$

La règle de conversion est validée par le modèle et on a $\vdash t : A$ qui implique $[\![A]\!] = \mathsf{true}$, d'où la cohérence du CC puisque $[\![\forall A : \mathtt{Prop}.A]\!] = \mathsf{false}$.

[1] Le modèle booléen standard de CC différencie selon que A est prouvable ou pas (cf par exemple [MW03]). Ce raffinement, qui force à considérer des interprétations partiellement définies n'est pas nécessaire pour notre analyse. Autrement dit, on se restreint à un modèle ayant le grain du modèle booléen de F_ω

[2] qui sont incidemment aussi des objets du niveau `Type`

Le modèle est localement fini et $[\![A]\!] = \mathtt{true}$ est décidable puisque les quantifications ne portent que sur des espaces fonctionnels engendrés à partir de $\mathcal{B}$ qui est fini et d'un nombre fini de construction par produit. En particulier la cohérence de CC est prouvable dans l'arithmétique.

Dans ce modèle, tous les entiers (dans le type $\forall A.(A \to A) \to (A \to A)$) sont identifiés. On ne peut donc prouver $0 \neq 1$. Les axiomes de Peano ne sont donc pas dérivables, en accord avec l'existence d'une preuve de cohérence dans l'arithmétique.

Question ouverte: complétion de F_ω (et plus généralement de CC dans son modèle booléen standard) vis à vis du modèle booléen. Quels axiomes ajouter pour obtenir $\models A \to \vdash A$ (candidats: $\forall A : \mathtt{Prop}.A = \mathsf{True} \vee A = \mathsf{False}$ (complétude propositionnelle) et $\forall f, g : (\forall x : T.U).(\forall u : T.fu =_{Uu} gu) \to f = g$ (extensionalité)) ?

Remarque: l'indiscernabilité des preuves (« proof-irrelevance »), la complétude propositionnelle et le tiers-exclu sont tous les trois validés par le modèle booléen..

6.1.2 Puissance calculatoire

Le CC admet aussi un modèle par réalisabilité qui respecte le contenu intensionnel des fonctions (et des preuves). Un tel type de modèle permet de prouver la normalisation forte de CC et F_ω, donc la cohérence de l'arithmétique d'ordre supérieur et, a fortiori, de l'arithmétique. En particulier, le modèle par réalisabilité montre la cohérence de CC + $(0 \neq n+1)$ + injectivité du successeur + induction, c'est-à-dire de CC + les axiomes de Peano.

Tous les programmes prouvablement terminant dans l'arithmétique d'ordre supérieur sont dérivables dans F_ω et donc dans le CC [Gir72].

Schématiquement, le modèle par réalisabilité interprète les propositions comme des ensembles de λ-termes clos par β-expansion avec la propriété que $t \in [\![A \to B]\!]$ ssi $tu \in [\![B]\!]$ pour tout $u \in [\![A]\!]$ (propriété dite de réductibilité). L'interprétation des types de `Type` est alors la même que pour le modèle booléen standard du CC : le produit du CC est interprété comme le produit ensembliste à ceci près que le domaine de base, `Prop`, est maintenant un ensemble de parties de λ-termes.

6.1.3 Extensions incohérentes du Calcul des Constructions

Systèmes $U-$, U et `Type:Type`

Les systèmes $U-$ et U (dus à J.-Y. Girard [Gir72]) sont en fait des extensions du système F_ω. Alors que la couche `Type` de F_ω correspond à un λ-calcul simplement typé bati sur la constante `Prop`, la couche `Type` des systèmes $U-$ et U correspond à un λ-calcul polymorphe (le système F) incluant `Prop` comme type de base.

Le système `Type : Type` (énoncé par P. Martin-Löf) est le système ou `Prop` et `Type` sont confondus et où tous les produits sont permis. En particulier, `Type : Type` est un objet terminal de la catégorie des PTS : tout PTS se plonge dans `Type : Type`.

Ces trois systèmes sont incohérents. Une des manières de dériver une incohérence est de construire dans $U-$ (et donc aussi dans les 2 autres systèmes) un type $Univ$: `Type` et une injection $i : \forall A : \mathtt{Type}.(A \to A \to \mathtt{Prop}) \to (A \to \mathtt{Prop}) \to Univ$[3].

En spécialisant i à $Univ$, on peut plonger dans $Univ$ toute relation $R : Univ \to Univ \to \mathtt{Prop}$ définie sur un domaine $D : Univ \to \mathtt{Prop}$. Par ce fait, on peut dériver le paradoxe de Burali-Forti (cf Girard [Gir72, Coq86]), le paradoxe de Reynold-Coquand (cf Coquand [Coq94]), ou plus généralement encoder la théorie naïve des ensembles de Cantor et en particulier le paradoxe de Russell (cf Miquel [Miq01]).

[3] Typiquement il suffit de prendre $Univ = \forall A : \mathtt{Type}.(A \to A \to \mathtt{Prop}) \to (A \to \mathtt{Prop})$.

6.2 Le Calcul des Constructions Étendu (ECC)

Le Calcul des Constructions Étendu est une extension du CC avec une hiérarchie dénombrable d'univers $\mathtt{Type} = \mathtt{Type}_1{:}\mathtt{Type}_2{:}\mathtt{Type}_3$... Les produits sur cette hiérarchie sont prédicatifs : si $T : \mathtt{Type}_i$ et $U(X) : \mathtt{Type}_j$ alors $\Pi X : T.U(X) : \mathtt{Type}_{max(i,j)}$. La sorte Prop reste imprédicative : si $T : \mathtt{Type}_i$ et $U(X) : \mathtt{Prop}$ alors $\Pi X : T.U(X) : \mathtt{Prop}$.

De plus, ECC introduit une relation de sous-typage vérifiant $\mathtt{Prop} \subset \mathtt{Type}_1 \subset \mathtt{Type}_2 \subset \mathtt{Type}_3$... Concrètement, la relation de conversion est remplacée par une relation de sous-typage définie par

$$\mathtt{Type}_i \subset \mathtt{Type}_j \text{ ssi } i \leq j$$

$$\mathtt{Prop} \subset \mathtt{Type}_i$$

$$\Pi X : T.U(X) \subset \Pi X : T'.U'(X) \text{ ssi } T =_\beta T' \text{ et } U(X) \subset U(X')$$

$$T \subset T' \text{ ssi } T =_\beta T' \text{ sinon}$$

La règle de conversion est remplacée en conséquence par

$$\frac{\Gamma \vdash t : T \qquad \Gamma \vdash T' : s \qquad T \subset T'}{\Gamma \vdash t : T'}$$

ECC correspond à la partie sans inductifs et sans Set du Calcul des Constructions Inductives[4].

6.2.1 Encodage de l'arithmétique

Dans ECC, on peut définir l'ensemble des entiers naturels au niveau $\mathtt{Type}_2$

$$\begin{array}{l} Nat : \mathtt{Type}_2 := \forall X : \mathtt{Type}_1.X \to (X \to X) \to X \\ 0 : Nat := \lambda X f a.a \\ s : Nat \to Nat := \lambda n X f a.n X f (f a) \end{array}$$

puis on peut définir le prédicat caractéristique des entiers naturels par

$$IsNat := \lambda n : Nat.\forall P : Nat \to \mathtt{Prop}.P(0) \to (\forall m : Nat.P(m) \to P(s(m))) \to P(n)$$

On peut alors prouver que $0 \neq s(n)$ et que s est injectif sur tout entier de $IsNat$, ce qui donne l'arithmétique de Peano.

Ainsi, CC avec juste un niveau supplémentaire de Type contient l'arithmétique[5].

6.2.2 Encodage de la théorie des ensembles de Zermelo

En utilisant un codage des ensembles sous forme de graphe pointé, Miquel [Miq01] a pu plonger la théorie des ensembles de Zermelo (qui est beaucoup plus faible que ZF) dans ECC avec trois niveaux de Type[6].

[4] La version originale de ECC est due à Luo [Luo90] et intègrait une construction primitive pour les types existentiels (Σ-types). Une version de ECC avec sous-typage restreint apparaît aussi dans la thèse de T. Coquand [Coq85] sous le nom de Calcul des Constructions Généralisé.

[5] En fait, les dépendances en les preuves ne sont pas nécessaires pour encoder l'arithmétique. Ainsi, F_ω avec un niveau supplémentaire convient tout aussi bien.

[6] Comme pour l'encodage de l'arithmétique, F_ω avec deux niveaux supplémentaires suffit pour encoder la théorie des ensembles de Zermelo.

6.2.3 Puissance logique

Nous allons faire une analyse de la force logique de ECC en termes ensemblistes, c'est-à-dire en termes de profondeur d'imbrication des ensembles que l'on peut construire dans ECC [7].

On vient de voir que $\mathtt{Type}_2$ contient l'ensemble des entiers naturels (ω) ce qui correspond en théorie des ensembles à un ensemble de profondeur ω. Par ailleurs, la construction de chaque $\mathtt{Type}_{i+1}$ permet au pire d'itèrer ω fois le produit fonctionnel à partir de $\mathtt{Type}_i$. Comme, en termes ensemblistes, une fonction de $\Pi x : A.B(x)$ est un graphe consistant en une partie du produit $A \times \cup_{x \in A} B(x)$, l'ensemble de fonctions $\Pi x : A.B(x)$ se construit en ajoutant un niveau d'imbrication au maximum des niveaux de A et des $B(x)$. Les types de $\mathtt{Type}_{i+1}$ ont donc un nombre fini de niveaux d'imbrications en plus du niveau d'imbrication de $\mathtt{Type}_i$. En passant à la limite, le type $\mathtt{Type}_{i+1}$ lui-même a donc ω niveaux d'imbrication en plus que $\mathtt{Type}_i$. On arrive de la sorte à établir que $\mathtt{Type}_i$ contient des ensembles de niveaux d'imbrication ensemblistes $\omega.i$. En passant à la limite une nouvelle fois, on établit que ECC doit contenir des ensembles de niveau d'imbrication au moins ω^2[8].

Par ailleurs, P.-A. Melliès et B. Werner [MW98] ont exhibé un modèle de cette cardinalité montrant que la force ordinale de ECC était bien V_{ω^2} (itération ω^2 des parties).

6.3 Calcul des Constructions Inductives (CCI)

6.3.1 Modèles des types inductifs

L'interprétation d'un type inductif peut se faire en introduisant une nouvelle constante de type et de nouvelles constantes de types...

6.4 Extensions cohérentes et incohérentes du Calcul des Constructions Inductives

On considère le Calcul des Constructions Inductives avec Set et Prop.

Dans le cas où Set est imprédicatif, il n'existe pas de modèle complet du CCI. Les deux approximations les plus proches sont

- le modèle de B. Werner qui ne prend en compte ni Prop ni la hiérarchie d'univers [Wer94]
- le modèle d'A. Miquel qui distingue une sorte Prop interprétée classiquement et une sorte Set interprétée par réalisabilité mais qui ne prend pas en compte les types inductifs[Miq01]

Dans le cas d'un Set prédicatif, il n'existe pas non plus explicitement de modèle, mais il est communément admis que le modèle booléen avec Set interprété comme un espace de produits ensemblistes fonctionne.

Indiscernabilité des preuves

L'indiscernabilité des preuves s'exprime par

$$\forall A : \mathtt{Prop}.\forall p, q : A.p = q$$

Indépendant dans CC : non prouvable (pas de preuve close), validé par le modèle booléen. Incohérent dans CCI en remplaçant Prop par Set (car $0 \neq 1$ dans Set).

[7] En théorie des ensembles, un niveau supplémentaire d'imbrication des ensembles correspond à une application de l'axiome des parties. On démarre de l'ensemble vide dont la profondeur est nulle. Le passage à la limite se fait avec l'axiome de l'union.

[8] ce qui correspond en théorie des ensembles à la cardinalité V_{ω^2}.

Logique classique

La logique classique peut être exprimée par exemple par le tiers-exclu.

$$\forall A : \texttt{Prop}.A \vee \neg A$$

Indépendant dans CC et ECC : non prouvable (pas de preuve close), validé par le modèle booléen.

Incohérent dans CCI si la disjonction est dans `Set` : de part la dérivabilité de l'axiome du choix (somme forte/élimination dépendante) et de $\texttt{true} \neq \texttt{false}$ dans `Set`, on peut injecter `Prop` dans `bool` (une « rétraction »), puis encoder le système $U-$ et dériver l'absurde.

Remarque: $\forall A : \texttt{Set}.A \vee \neg A$ est supposé cohérent. Ce qui permet vraiment de montrer une incohérence, c'est que la disjonction soit dans `Set`.

Dans CCI, la logique classique entraîne l'indiscernabilité des preuves (la preuve nécessite l'élimination dépendante des inductifs de `Prop` et l'imprédicativité, cf Barbanera et Berardi [BB96]).

Extensionnalité propositionnelle et complétude propositionnelle

La complétude propositionnelle ($\forall A : \texttt{Prop}.A = \texttt{True} \vee A = \texttt{False}$) est équivalente dans CC à la conjonction de la logique classique et de l'extensionalité propositionnelle ($\forall A, B : \texttt{Prop}.(A \leftrightarrow B) \rightarrow A = B$).

Dans CCI, l'extensionalité propositionnelle entraîne l'indiscernabilité des preuves (la preuve nécessite l'élimination dépendante des inductifs de `Prop`).

Axiome du choix

L'axiome du choix (forme fonctionnelle) est dérivable si l'existentielle (notée alors comme un Σ-type) est dans `Set` ou dans `Type` :

$$(\forall x : X.\Sigma y : Y.P(x, y)) \rightarrow \Sigma f : X \rightarrow Y.\forall x : X.P(x, fx)$$

Si l'existentielle est dans `Prop` et Y dans `Set`, l'axiome est incohérent en présence de la logique classique dans `Prop`, car alors, on peut injecter `Prop` dans `bool` et encoder le système $U-$.

$$\forall X : \texttt{Type}.\forall A : Set.(\forall x : X.\exists a : A.P(x, y)) \rightarrow \exists f : X \rightarrow A.\forall x : X.P(x, fx)$$

En fait, une forme beaucoup plus faible de l'axiome du choix, à savoir l'axiome de description suffit à obtenir cette contradiction.

Remarque: Si Y est dans `Type`, du fait du sous-typage $\texttt{Set} \subset \texttt{Type}$, la contradiction avec la logique classique persiste.

L'axiome de description

L'axiome de description (aussi appelé « principe des descriptions définies », ou « axiome du choix unique ») s'exprime par

$$(\forall a : A.\exists! b : B.P(a, b)) \rightarrow \exists f : A \rightarrow B.\forall a : A.P(a, fa)$$

Si l'existentielle est dans `Set` ou `Type`, l'axiome est dérivable. Dans `Prop` (et avec B dans `Set` ou `Type`), il entraîne l'existence d'une rétraction de `Prop` vers `bool:Set`. Ainsi, l'axiome est incohérent avec la logique classique dans `Prop` pour le CCI avec `Set` imprédicatif (car alors on peut simuler le système $U-$). Si `Set` est prédicatif, il reste cohérent avec la logique classique (la rétraction s'interprète comme un oracle qui décide la validité de toute proposition).

On peut aussi s'intéresser aux interactions avec la logique classique calculatoire, c'est-à-dire basée sur une réalisation par l'opérateur `call-cc` de Scheme ou SML ou l'opérateur μ de Parigot. Une telle interprétation calculatoire de la logique classique est incompatible avec l'axiome de description même avec `Set` prédicatif.

Axiome du choix relationnel

L'axiome du choix sous sa forme relationnelle s'exprime par

$$(\forall a : A.\exists b : B.P(a, b)) \quad \rightarrow \exists P'.\forall a : A.\exists b : B.(P(a, b) \wedge P'(a, b) \wedge \forall b' : B.P'(a, b') \rightarrow b = b'$$

L'axiome du choix relationnel + l'axiome de description est équivalent à l'axiome du choix fonctionnel.

Autant l'axiome de description est problèmatique avec la logique classique, autant on ne connaît pas de contradiction à avoir simultanément l'axiome du choix relationnel et la logique classique.

Extensionnalité des prédicats et axiome du choix

L'extensionnalité des prédicats + l'axiome du choix (même sous une forme relationnelle qui n'implique pas l'axiome de description) entraîne la logique classique !! C'est à l'origine un résultat de Diaconescu pour la théorie des ensembles qui a été adapté à la théorie des types [Dia75, LW99].

Extensionnalité fonctionnelle

Non prouvable. Cohérente ?

Condition de garde

Sans condition de garde, on peut directement dériver l'absurde:

```
Fixpoint Paradox (u:unit) : False := Paradox u.
```

Condition de positivité

Sans condition de positivité, on peut facilement dériver l'absurde:

```
Inductive A : Prop := intro : (A->False)->A.
Definition Paradox : False := (fun (H:A->False) => H (intro H)).
```

Élimination forte sur un ensemble large

...

Bibliographie

[BB96] F. Barbanera and S. Berardi. Proof-irrelevance out of excluded-middle and choice in the calculus of constructions. *Journal of Functional Programming*, 6(3):519–525, 1996.

[Coq85] Th. Coquand. *Une Théorie des Constructions*. PhD thesis, Université Paris 7, January 1985.

[Coq86] Th. Coquand. An analysis of girard's paradox. In *Symposium on Logic in Computer Science*, Cambridge, MA, 1986. IEEE Computer Society Press.

[Coq94] Thierry Coquand. A new paradox in type theory. In *9th International Conference on Logic, Methodology, and Philosophy of Science*, pages 555–570, 1994.

[Dia75] R. Diaconescu. Axiom of choice and complementation. In *Proceedings of AMS*, volume 51, pages 176–178, 1975.

[Gir72] J.-Y. Girard. Interprétation fonctionnelle et élimination des coupures de l'arithmétique d'ordre supérieur, 1972.

[Luo90] Z. Luo. *An Extended Calculus of Constructions*. PhD thesis, University of Edinburgh, 1990.

[LW99] Samuel Lacas and Benjamin Werner. Which choices imply the excluded middle? manuscript, 1999.

[Miq01] Alexandre Miquel. *Le calcul des constructions implicite : syntaxe et sémantique*. PhD thesis, Université Paris 7, December 2001.

[MW98] Paul-André Melliès and Benjamin Werner. A generic normalisation proof for pure type systems. In Eduardo Giménez and Christine Paulin-Mohring, editors, *Types for Proofs and Programs, International Workshop TYPES'96, Aussois, France, December 15-19, 1996, Selected Papers*, volume 1512 of *Lecture Notes in Computer Science*, pages 254–276. Springer, 1998.

[MW03] Alexandre Miquel and Benjamin Werner. The not so simple proof-irrelevant model of cc. In Herman Geuvers and Freek Wiedijk, editors, *Types for Proofs and Programs, Second International Workshop, TYPES 2002, Berg en Dal, The Netherlands, April 24-28, 2002, Selected Papers*, volume 2646 of *Lecture Notes in Computer Science*, pages 240–258. Springer, 2003.

[Wer94] B. Werner. *Une théorie des constructions inductives*. Thèse de doctorat, Université Paris 7, 1994.