Sous-typage coercitif en présence de réductions non-standard dans un système aux types dépendants

Lionel Marie-Magdeleine

sous la direction de Sergei Soloviev

Université Paul Sabatier, IRIT, équipe ACADIE

Soutenance de Thèse
11 décembre 2009

Objectifs

Le but de cette étude est :

- d'enrichir une théorie de type d'une réduction non standard concernant les fonctions entre types finis.
- d'étudier les propriétés de ce système étendu
- explorer les propriétés de ce système enrichi en plus par la notion de sous-typage

Sommaire

Sommaire

Sommaire

Sommaire

Sommaire

Sommaire

Sommaire

Le système UTT

UTT est une théorie de types déclarée dans le "Logical Framework" LF comprenant :

- des types dépendants,
- des types inductifs,
- un univers imprédicatif des propositions,
- un univers prédicatif de types.

"Logical Framework" *LF*

Les termes de LF sont de la forme suivante :
Type, $El(A)$, $(x : K)K'$, $[x : K]k'$, $f(k)$,
où les occurrences libres de la variable x sont liées par les opérateurs $(x : K)$ et $[x : K]$ respectivement.

Il y a cinq formes de jugement dans LF :

- $\Gamma \vdash$ **valid** signifie Γ est un contexte valide,
- $\Gamma \vdash K$**kind**, signifie que K est une sorte,
- $\Gamma \vdash k : K$ signifie que k est un objet de sorte K,
- $\Gamma \vdash k = k' : K$ signifie que k et k' sont des objets égaux de sorte K,
- $\Gamma \vdash K = K'$ signifie que K et K' sont des sortes égales.

Remarque : Nous employons le mot "sorte" pour désigner les "kinds" de LF.

"Logical Framework" *LF*

Les types inductifs

Pour un contexte valide Γ et $\bar{\Theta}$ une famille de schémas dans Γpar rapport à X notée $SCH_{\Gamma;X}(\bar{\Theta})$, les constantes suivantes sont définies :

- $M^X[\bar{\Theta}]$: $\quad$ **Type**
- $\iota_i^X[\bar{\Theta}]$: $\quad \Theta_i(M^X[\bar{\Theta}]) \quad (1 \leq i \leq n)$
- $E^X[\bar{\Theta}]$: $\quad (C : (M^X[\bar{\Theta}])\mathbf{Type})\ (f_1 : \Theta_1^{\circ}[M^X[\bar{\Theta}], C, \iota_1^X[\bar{\Theta}]]) \ldots$ $(f_n : \Theta_n^{\circ}[M^X[\bar{\Theta}], C, \iota_n^X[\bar{\Theta}]])\ (z : M^X[\bar{\Theta}])C(z)$

ainsi que les règles d'égalité associées :

$$E^X[\bar{\Theta}](C, \bar{f}, \iota_i^X[\bar{\Theta}](\bar{a})) =$$
$$f_i(\bar{a}, \Phi_1^{\natural}[M^X[\bar{\Theta}], C, E^X[\bar{\Theta}](C, \bar{f}), a_1], \ldots, \Phi_k^{\natural}[M^X[\bar{\Theta}], C, E^X[\bar{\Theta}](C, \bar{f}), a_k]) :$$
$$C(\iota_i^X[\bar{\Theta}](\bar{a})).$$

"Logical Framework" *LF*

Les types inductifs

Ex : les entiers naturels

- $N :$ **Type**
- $0 :$ N
- $succ :$ $(N)N$
- $Rec_N :$ $(C : (N)\mathbf{Type})$
  $(c : C(0))$
  $(f : (x : N)(C(x))C(succ(x)))$
  $(n : N)C(n)$

Nous avons les règles d'égalité associées :

- $Rec_N(C, c, f, 0) = c : C(0)$
- $Rec_N(C, c, f, succ(n)) = f(n, Rec_N(C, c, f, n)) : C(succ(n))$

- le type N est défini par $N =_{df} M^X[\bar{\Theta}_N]$,
- où $\bar{\Theta}_N \equiv \langle X, (X)X \rangle$,
- le premier constructeur est $0 =_{df} \iota_1^X[\bar{\Theta}_N]$,
- le second constructeur est $succ =_{df} \iota_2^X[\bar{\Theta}_N]$,
- l'opérateur de récursion est $Rec_n \equiv E^X[\bar{\Theta}_N]$.

Univers imprédicatifs des propositions

La logique interne de *UTT* consiste en un univers *Prop* des propositions et le type de leurs preuves.

- $Prop : \mathbf{Type}$
- $\mathbf{Prf} : (Prop)\mathbf{Type}$
- $\forall : (A : \mathbf{Type})((A)Prop)Prop$
- $\Lambda : (A : \mathbf{Type})(P : (A)Prop)((x : A)\mathbf{Prf}(P(x)))\mathbf{Prf}(\forall(A, P))$
- $\mathbf{E}_{\forall} : (A : \mathbf{Type})(P : (A)Prop)(R : \mathbf{Prf}(\forall(A, P)))Prop)((g : (x : A)\mathbf{Prf}(P(x)))\mathbf{Prf}(R(\Lambda(A, P, g))))(z : \mathbf{Prf}(\forall(A, P)))\mathbf{Prf}(R(z))$
- On a l'égalité calculatoire suivante : $\mathbf{E}_{\forall}(A, P, R, f, \Lambda(A, P, g)) = f(g) : \mathbf{Prf}(R(\Lambda(A, Pg)))$.

Univers prédicatif

L'univers prédicatif contient les noms des types, et permet de quantifier sur les types.

- $\mathbf{U} : \mathbf{Type}$
- $\mathbf{T} : (\mathbf{U})\mathbf{Type}$ ($\mathbf{T}$ transforme le nom d'un type en le type correspondant)
- $prop : \mathbf{U}$ ($prop$ est le nom du type $Prop$)
- $prf : (Prop)\mathbf{U}$ ($prf(A)$ donne le nom de $Prf(A)$)
- Les règles d'égalité associées sont les suivantes :

$$\frac{\Gamma \vdash \mathbf{valid}}{\Gamma \vdash \mathbf{T}(prop) = Prop : \mathbf{Type}}\ (\text{T-prop})$$

$$\frac{\Gamma \vdash P : Prop}{\Gamma \vdash \mathbf{T}(prf(P)) = \mathbf{Prf}(P) : \mathbf{Type}}\ (\text{T-prf})$$

Sommaire

Les types inductifs

Remarques

- La notion de types inductifs introduit celle d'objets canoniques.
- Un objet canonique d'un type inductif A est terme de type A construit uniquement par l'application des constructeurs de A (sans autre constante et sans variable).
- L'application d'un récurseur à un objet non-canonique (par exemple une variable) ne se réduit pas en general (par la réduction mentionnée précédemment).

Égalité Intentionnelle VS Égalité Extensionnelle

Exemple dans *UTT*

Considérons :

- $g =_{df} [x : N]Plus_x(x)$, avec
  $Plus_x =_{df} Rec([y : N]N, x, [y : N]succ)$ où
  $Plus_x(y) = x + y$.
- $f \equiv Mult_2 =_{df} Rec([y : N]N, 0, [x : N][y : N]succ(succ(y)))$ où
  $Mult_2(x) = 2.x$

Fonctions extensionnellement égales mais pas intentionnellement égales.

- D'un point de vue ensembliste ces deux fonctions ont le même graphe, les mêmes couples antécédents/images.
- Il n'existe pas un terme commun auquel les deux fonctions peuvent se réduire.
- Les récurseurs appliqués à des objets non-canoniques comme des variables ne se réduisent pas.

Les types finis

Représentation d'ensembles finis comme types inductifs

Définition

Notons $[n]$ l'ensemble $\{i \in \mathbb{N}/1 \leq i \leq n\}$.

- On définit le type $F_n \equiv M[\bar{\Theta}_n]$ où $\bar{\Theta}_n \equiv < \underbrace{X, \ldots, X}_{nfois} >$.
- Les termes $\iota_i[\bar{\Theta}_n] : F_n$ représentent les objets canoniques de F_n et symbolisent les éléments de l'ensemble $[n]$.
- A toute application $\sigma : [n] \rightarrow [m]$ correspond le terme $E[\bar{\Theta}_n]([z : F_n]F_m, \iota_{\sigma(1)}[\bar{\Theta}_m], ..., \iota_{\sigma(n)}[\bar{\Theta}_m]) : (F_n)F_m$.

Remarque :
Par la suite, pour plus de clarté lorsqu'il s'agira de types finis, nous noterons :
E^m à la place de $E[\bar{\Theta}_m]$, ι^m à la place de $\iota[\bar{\Theta}_m]$.

La ϑ-réduction

Prenons trois types finis F_n, F_m et F_p.
Aux applications $g : [m] \to [p]$ et $f : [n] \to [m]$ correspondent les termes

- $E^m([y : F_m]F_p, \iota^p_{g(1)}, ..., \iota^p_{g(m)}) : (F_m)F_p$ et
- $E^n([z : F_n]F_m, \iota^m_{f(1)}, ..., \iota^m_{f(n)}) : (F_n)F_m$ respectivement.

La ϑ-réduction

Prenons trois types finis F_n, F_m et F_p.
Aux applications $g : [m] \to [p]$ et $f : [n] \to [m]$ correspondent les termes

- $E^m([y : F_m]F_p, \iota^p_{g(1)}, \ldots, \iota^p_{g(m)}) : (F_m)F_p$ et
- $E^n([z : F_n]F_m, \iota^m_{f(1)}, \ldots, \iota^m_{f(n)}) : (F_n)F_m$ respectivement.

Dans une étude précédente, Soloviev et Chemouil ont étudié la réduction suivante dans le cadre d'un λ-calcul simplement typé avec types inductifs.

$$E^m([y : F_m]F_p, \iota^p_{g(1)}, \ldots, \iota^p_{g(m)})(E^n([z : F_n]F_m, \iota^m_{f(1)}, \ldots, \iota^m_{f(n)})(k))$$
$$\vartheta'$$
$$E^n([x : F_n]F_p, \iota^p_{g\circ f(1)}, \ldots, \iota^p_{g\circ f(n)})(k).$$

La ϑ-réduction

Généralisation de la ϑ'-réduction

ϑ-réduction

Soit la relation de récriture ϑ

$$E^m(C, a_1, \ldots, a_m)(E^n(C_1, \iota^m_{\sigma(1)}, \ldots, \iota^m_{\sigma(n)})(k))$$

$$\vartheta$$

$$E^n(C_2, a_{\sigma(1)}, \ldots, a_{\sigma(n)})(k)$$

où

- A est un type quelconque.
- $C \equiv [y : F_m]A : (y : F_m)$**Type**,
- $C_1 \equiv [z : F_n]F_m : (z : F_n)$**Type** et
- $C_2 \equiv [x : F_n]A : (x : F_n)$**Type**

La ϑ-réduction

Généralisation de la ϑ'-réduction

ϑ-réduction

Soit la relation de récriture ϑ

$$E^m(C, a_1, ..., a_m)(E^n(C_1, \iota^m_{\sigma(1)}, ..., \iota^m_{\sigma(n)})(k))$$

$$\vartheta$$

$$E^n(C_2, a_{\sigma(1)}, ..., a_{\sigma(n)})(k)$$

où

- A est un type quelconque.
- $C \equiv [y : F_m]A : (y : F_m)\mathbf{Type}$,
- $C_1 \equiv [z : F_n]F_m : (z : F_n)\mathbf{Type}$ et
- $C_2 \equiv [x : F_n]A : (x : F_n)\mathbf{Type}$

La ϑ-réduction transforme une égalité extensionnelle en une égalité intentionnelle

La ϑ-réduction

La règle ϑ-eq

Nous rajoutons la ϑ-réduction au calcul sous forme de jugement d'égalité :

$$\frac{\Gamma \vdash E^m(C, a_1, ..., a_m) : (F_m)A \qquad \Gamma \vdash E^n(C_1, \iota^m_{\sigma(1)}, ..., \iota^m_{\sigma(n)}) : (F_n)F_m \qquad \Gamma \vdash k : F_n}{\Gamma \vdash E^m(C, a_1, ..., a_m)(E^n(C_1, \iota^m_{\sigma(1)}, ..., \iota^m_{\sigma(n)})(k)) = E^n(C_2, a_{\sigma(1)}, ..., a_{\sigma(n)})(k) : A}$$

Sommaire

La Sémantique Opérationnelle Typée

Nous rappelons que le UTT^{ϑ} n'est autre que UTT auquel nous rajoutons la règle ϑ-eq.

Healfdene Goguen dans sa thèse sous la direction Rod Burstall et Zhaohui Luo, prouve que UTT vérifie entre autre les propriétés :

- de normalisation forte,
- de confluence et
- de préservation du type,

au moyen d'une sémantique opérationnelle typée "Typed Operational Semantics" développée pour UTT qui se nomme UTT^{S}.

La Sémantique Opérationnelle Typée

Afin de prouver que UTT^{ϑ} vérifie les propriétés :

- de normalisation forte,
- de confluence et
- de préservation du type,

nous emploierons et adapterons la méthode développée par Goguen.
Nous proposons donc une sémantique opérationnelle typée pour UTT^{ϑ}, à savoir $UTT^{\vartheta S}$

La Sémantique Opérationnelle Typée

Étape de la preuve : Nous suivons le même schéma de preuve défini par Goguen :

- définition de $UTT^{\vartheta S}$, puis preuve que $UTT^{\vartheta S}$ vérifie les propriétés :
  - de normalisation forte,
  - de confluence et
  - de préservation du type.
- transfert des ces mêmes propriétés à UTT^{ϑ}.
  - Définition d'un modèle ensembliste pour UTT^{ϑ} afin de justifier un principe d'induction.
  - Preuve que UTT^{ϑ} est correcte par rapport à $UTT^{\vartheta S}$, à l'aide d'un modèle de Kripke défini pour UTT^{ϑ}.

Le système $UTT^{\vartheta S}$

Présentation

Les formes de jugement pour $UTT^{\vartheta S}$ et leurs significations intuitives sont les suivantes :

- $\Gamma \vdash^S$ **valid** signifie que Γ est un contexte valide,
- $\Gamma \vdash^S A \xrightarrow{nf} B$ signifie que la sorte A admet une forme normale qui est la sorte B et que A et B sont des sortes valides dans le contexte Γ,
- $\Gamma \vdash^S M \xrightarrow{nf} P : A$ signifie que le terme M admet une forme normale P de sorte A dans le contexte Γ,
- $\Gamma \vdash^S M \xrightarrow{wh} N : A$ signifie M se réduit en N en un pas par une stratégie externe de réduction, de plus M et N sont des termes de sorte A dans le contexte Γ.

Le système $UTT^{\vartheta S}$

Présentation

Ajout de la règle suivante, en s'assurant que les propriétés vérifiées par UTT^{S} le sont toujours par $UTT^{\vartheta S}$.

$$\frac{\begin{array}{c}\Gamma \vdash^{S} E^{m}(C, a_1, \ldots, a_m) \xrightarrow{nf} E^{m}(C, a_1, \ldots, a_m) : (F_m)A \\ \Gamma \vdash^{S} E^{n}(C_1, \iota^{m}_{\sigma(1)}, \ldots, \iota^{m}_{\sigma(n)}) \xrightarrow{nf} E^{n}(C_1, \iota^{m}_{\sigma(1)}, \ldots, \iota^{m}_{\sigma(n)}) : (F_n)F_m \\ \Gamma \vdash^{S} k \xrightarrow{nf} v : F_n \quad v \in FV(\Gamma)\end{array}}{\Gamma \vdash^{S} E^{m}(C, a_1, \ldots, a_m)(E^{n}(C_1, \iota^{m}_{\sigma(1)}, \ldots, \iota^{m}_{\sigma(n)})(k)) \xrightarrow{wh} E^{n}(C_2, a_{\sigma(1)}, \ldots, a_{\sigma(n)})(k) : A}$$

- A est un type quelconque.
- $C \equiv [y : F_m]A : (y : F_m)$**Type**,
- $C_1 \equiv [z : F_n]F_m : (z : F_n)$**Type** et
- $C_2 \equiv [x : F_n]A : (x : F_n)$**Type**

Sommaire

Sous-typage

Il existe deux grandes approches du sous-typage.

1. L'approche relationnelle (plus traditionnelle).
2. L'approche coercitive.

Sous-typage

Approche relationnelle

Dans l'approche relationnelle, la notion de sous-typage est juste vue comme une relation entre deux types $A_0 < A$.
Un terme t de type A_0 est également considéré comme étant de type A.
Ce fait est illustré par l'utilisation de la règle appelée "subsomption"

$$\frac{\Gamma \vdash t : A_0 \qquad \Gamma \vdash A_0 < A}{\Gamma \vdash t : A}.$$

Sous-typage

Approche relationnelle

Inconvénients :
notamment pour les systèmes possédant des types inductifs.

Exemple

Dans un contexte Γ on a :

- F_3 avec comme éléments 1_{F_3}, 2_{F_3} et 3_{F_3}.
- $N =_{df} M^X[\langle X, (X)X \rangle]$
- $Plus_0 : (N)N$, une fonction définie à l'aide d'un récurseur pour le type N

F_3 sous-type de N ($F_3 < N$).
Les réductions qui s'appliquent à ($Plus_0(1)$ avec $1 =_{df} succ(0)$, ne s'appliquent pas directement à $Plus_0(1_{F_3})$ (1_{F_3} n'est pas un objet canonique de N.)

$$\Gamma \not\vdash Plus_0(1_{F_3}) = 1_{F_3}$$

Sous-typage

Approche coercitive

L'approche coercitive permet de palier au problème vu précédemment :

- La relation de sous-typage ($A_0 < A$) est interprétée par l'existence d'un certain terme $c : A_0 \rightarrow A$, appelé coercion.
- La coercion est une représentation explicite de la transformation du (des éléments du) sous-type vers le (les éléments du) super-type.

Sous-typage

Approche coercitive

Approche coercitive : deux directions

1. Coercions : termes particuliers, imposés par le système.
   Par exemple, dans l'approche présentée par Longo, Milsted et Soloviev, les coercions sont des termes qui, après effacement des informations de types, se η-réduisent à l'identité.
2. Coercions : n'importe quelle fonction entre deux types vérifiant certaines conditions(Z.Luo).

Sommaire

Sommaire

$UTT^{\vartheta}[C]$

Nous nous situons dans le cadre du sous-typage coercitif selon l'approche de Luo.

- $UTT^{\vartheta}[C]$ représente le système UTT^{ϑ} enrichi du sous-typage coercitif, où
- C est un ensemble de triplets (sous-type, type, terme de coercion)

$UTT^{\vartheta}[C]$

Nous nous situons dans le cadre du sous-typage coercitif selon l'approche de Luo.

- $UTT^{\vartheta}[C]$ représente le système UTT^{ϑ} enrichi du sous-typage coercitif, où
- C est un ensemble de triplets (sous-type, type, terme de coercion)

De nouvelles formes de jugement apparaissent :

- $\Gamma \vdash A <_c B$: **Type** affirme que A est un sous-type de B par le biais de la coercion c, avec $c : (El(A))El(B)$.
- $\Gamma \vdash K <_c K'$ affirme que K est une sous-sorte de K' par le biais de la coercion c, avec $c : (K)K'$

$UTT^{\vartheta}[C]$

Distinction entre coercions de base (ensemble C) et coercions dérivées.

$UTT^{\vartheta}[C]$

Distinction entre coercions de base (ensemble C) et coercions dérivées.

Deux systèmes intermédiaires :

- $UTT^{\vartheta}[C]_o$
  - Jugements de sous-typage ($\Gamma \vdash A <_c B$: **Type**).
  - Coercions dérivées par composition (transitivité) au niveau des types.
- $UTT^{\vartheta}[C]_{ok}$
  - Règles de $UTT^{\vartheta}[C]_o$, plus
  - coercions entre les sortes ($\Gamma \vdash K <_c K'$),
  - règles de sous-typage pour le produit dépendant.

$UTT^\vartheta[C]$

Distinction entre coercions de base (ensemble C) et coercions dérivées.

Deux systèmes intermédiaires :

- $UTT^\vartheta[C]_o$
  - Jugements de sous-typage ($\Gamma \vdash A <_c B$: **Type**).
  - Coercions dérivées par composition (transitivité) au niveau des types.
- $UTT^\vartheta[C]_{ok}$
  - Règles de $UTT^\vartheta[C]_o$, plus
  - coercions entre les sortes ($\Gamma \vdash K <_c K'$),
  - règles de sous-typage pour le produit dépendant.

Remarque

Ces deux systèmes ne permettent pas de dériver de nouveaux jugements, sauf de la forme $\Gamma \vdash K <_c K'$.(Conservativité évidente)

$UTT^{\vartheta}[C]$

L'ensemble des coercions de base doit verifier les conditions de cohérence.

Définition (Condition de cohérence)

On dit que C est coherent si $UTT^{\vartheta}[C]_o$ a les propriétés suivantes :

1. Le jugement $\Gamma \vdash A <_c A : \mathbf{Type}$ n'est pas dérivable, quelque soit le contexte Γ,le type A et le terme c.
2. Si les jugements $\Gamma \vdash A <_c B : \mathbf{Type}$ et $\Gamma \vdash A <_{c'} B : \mathbf{Type}$ sont dérivable alors on a $\Gamma \vdash c = c' : (El(A))El(B)$

Nous présumons que C est cohérent.

$UTT^\vartheta[C]$

$UTT^\vartheta[C]$ inclut, en plus des règles de $UTT^\vartheta[C]_{ok}$, les règles suivantes qui illustrent le rôle d'un terme de coercion c :

Règles Coercitives

- $$\frac{\Gamma \vdash f : (x : K)K' \qquad \Gamma \vdash k : K_0 \qquad \Gamma \vdash K_0 <_c K}{\Gamma \vdash f(k) : [c(k)/x]K'} \; (CA.1)$$
- $$\frac{\Gamma \vdash f = f' : (x : K)K' \qquad \Gamma \vdash k_1 = k_2 : K_0 \qquad \Gamma \vdash K_0 <_c K}{\Gamma \vdash f(k_1) = f'(k_2) : [c(k_1)/x]K'} \; (CA.2)$$
- $$\frac{\Gamma \vdash f : (x : K)K' \qquad \Gamma \vdash k_0 : K_0 \qquad \Gamma \vdash K_0 <_c K}{\Gamma \vdash f(k_0) = f(c(k_0)) : [c(k_0)/x]K'} \; (CD) .$$

- Ces règles établissent la transition entre un jugement qui est un jugement de sous-typage ($\Gamma \vdash K_0 <_c K$), et un jugement qui ne l'est pas.
- Les coercions sont vues comme un mécanisme d'abréviation.
- On peut utiliser $f(k_0)$ comme une abréviation de $f(c(k_0))$.

Sommaire

$UTT^{\vartheta}[C]$

- Démontrer que $UTT^{\vartheta}[C]$ est une extension conservative de UTT^{ϑ} :
  tout jugement dérivable dans $UTT^{\vartheta}[C]$, qui n'est pas un jugement de sous-typage ou de sous-sorte et qui ne contient pas d'application coercitive, est dérivable dans UTT^{ϑ}.
- Problème de conservativité : un des principaux problèmes pour le sous-typage comme mécanisme d'abréviation.
- La cohérence est une condition nécessaire pour la preuve de conservativité

$UTT^{\vartheta}[C]$

La preuve de conservativité consiste en trois parties majeures :

- Lemmes sur les propriétés métathéoriques générales de $UTT^{\vartheta}[C]$.
- Elimination de la transitivité pour les sous-sortes, dans le calcul $UTT^{\vartheta}[C]_{ok}$.
- La fonction de complétion des coercions est totale.

$UTT^{\vartheta}[C]$

Propriétés métathéoriques

Théorème (Elimination de certaines règles admissibles)

Il existe un algorithme **E** qui transforme toute $UTT^{\vartheta}[C]$-dérivation d en une dérivation du même jugement sans les règles de substitution, d'affaiblissement, et de retypage de contexte.

- $\dfrac{\Gamma_1, \Gamma_2 \vdash J \qquad \Gamma_1, \Gamma_3 \vdash \mathbf{valid}}{\Gamma_1, \Gamma_3 \Gamma_2 \vdash J}$ (wkn)
- $\dfrac{\Gamma, x : K, \Gamma' \vdash J \qquad \Gamma \vdash K = K'}{\Gamma, x : K', \Gamma' \vdash J}$ (retypage)
- $\dfrac{\Gamma, x : K, \Gamma' \vdash K' \mathbf{kind} \qquad \Gamma \vdash k : K}{\Gamma, [k/x]\Gamma' \vdash [k/x]K' \mathbf{kind}}$ (subst)

- L'induction sur la taille des dérivations ne convient pas.
- Emploi d'une induction utilisant la définition de l'algorithme d'élimination.

$UTT^{\vartheta}[C]$

Propriétés métathéoriques

Théorème (Jugements Présupposés)

Il existe des algorithmes qui transforment toute $UTT^{\vartheta}[C]$-dérivation d de

1) $\Gamma_1, \Gamma_2 \vdash J$ en une dérivation V_{Γ_1}(d) de $\Gamma_1 \vdash$ **valid**.
2) $\Gamma_1, x : K, \Gamma_2 \vdash J$ en une dérivation K_{Γ_1}(d) de $\Gamma_1 \vdash K$**Kind**.
3) $\Gamma \vdash (x : K_1)K_2$ en une dérivation $p\_^{\circ}$(d) de $\Gamma_1, x : K_1 \vdash K_2$**Kind**.
4) $\Gamma \vdash K_1 = K_2$ en une dérivation $\mathbf{l}_=$(d) de $\Gamma \vdash K_1$**kind** et $\mathbf{r}_=$(d) de $\Gamma \vdash K_2$**kind** ;
5) $\Gamma \vdash k_1 = k_2 : K$ en une dérivation $l_=$(d) de $\Gamma \vdash k_1 : K$ et $r_=$(d) de $\Gamma \vdash k_2 : K$;
6) $\Gamma \vdash \Sigma : K$ en une dérivation **kd**(d) de $\Gamma \vdash K$**kind** (Σ denote un terme ou une égalité entre termes) ;
7) $\Gamma \vdash A <_c B :$ **Type** ou $\Gamma \vdash K <_c K'$ en des dérivations $\mathbf{l}_<$(d) de $\Gamma \vdash A :$ **Type** ou $\Gamma \vdash K$**kind**, $\mathbf{r}_<$(d) de $\Gamma \vdash B :$ **Type** ou $\Gamma \vdash K'$**kind**, **co**(d) de $\Gamma \vdash c : (El(A))El(B)$ ou $\Gamma \vdash c : (K)K'$ respectivement.

$UTT^{\vartheta}[C]$

Elimination de la transitivité pour les sous-sortes

- $$\frac{\Gamma \vdash K_1 <_c K_2 \qquad \Gamma \vdash K_2 = K_2'}{\Gamma \vdash K_1 <_c K_2'} \text{ (SK.5)} \qquad \frac{\Gamma \vdash K_1 <_c K_2 \qquad \Gamma \vdash K_1 = K_1'}{\Gamma \vdash K_1' <_c K_2} \text{ (SK.6)}$$
- $$\frac{\Gamma \vdash K <_{c_1} K' \qquad \Gamma \vdash K' <_{c_2} K''}{\Gamma \vdash K <_{[x:K]c_2(c_1(x))} K''} \text{ (SK.7)}$$

$UTT^{\vartheta}[C]$

Elimination de la transitivité pour les sous-sortes

- $$\frac{\Gamma \vdash K_1 <_c K_2 \qquad \Gamma \vdash K_2 = K_2'}{\Gamma \vdash K_1 <_c K_2'}\ (SK.5) \qquad \frac{\Gamma \vdash K_1 <_c K_2 \qquad \Gamma \vdash K_1 = K_1'}{\Gamma \vdash K_1' <_c K_2}\ (SK.6)$$
- $$\frac{\Gamma \vdash K <_{c_1} K' \qquad \Gamma \vdash K' <_{c_2} K''}{\Gamma \vdash K <_{[x:K]c_2(c_1(x))} K''}\ (SK.7)$$

Théorème(Elimination de la transitivité)

Il existe un algorithme qui transforme toute dérivation du jugement $\Gamma \vdash K <_c K'$ dans $UTT^{\vartheta}[C]^-_{ok}$ en une dérivation du jugement $\Gamma \vdash K <_{c'} K'$ dans le même calcul mais ne contenant pas les règles SK.5, SK.6 et SK.7 telles que $\Gamma \vdash c' = c : (K)K'$ dans UTT^{ϑ}.

$UTT^\vartheta[C]$

Elimination de la transitivité pour les sous-sortes

- $$\frac{\Gamma \vdash K_1 <_c K_2 \qquad \Gamma \vdash K_2 = K_2'}{\Gamma \vdash K_1 <_c K_2'}\ (SK.5) \qquad \frac{\Gamma \vdash K_1 <_c K_2 \qquad \Gamma \vdash K_1 = K_1'}{\Gamma \vdash K_1' <_c K_2}\ (SK.6)$$
- $$\frac{\Gamma \vdash K <_{c_1} K' \qquad \Gamma \vdash K' <_{c_2} K''}{\Gamma \vdash K <_{[x:K]c_2(c_1(x))} K''}\ (SK.7)$$

Théorème(Elimination de la transitivité)

Il existe un algorithme qui transforme toute dérivation du jugement $\Gamma \vdash K <_c K'$ dans $UTT^\vartheta[C]^-_{ok}$ en une dérivation du jugement $\Gamma \vdash K <_{c'} K'$ dans le même calcul mais ne contenant pas les règles SK.5, SK.6 et SK.7 telles que $\Gamma \vdash c' = c : (K)K'$ dans UTT^ϑ.

Corollaire

Si les conditions de cohérence sont satisfaites pour le sous-typage alors dans $UTT^\vartheta[C]_{ok}$, on a :

a) si $\Gamma \vdash K = K'$ alors $\Gamma \vdash K <_c K'$ n'est pas dérivable ;

b) si $\Gamma \vdash K <_c K'$ et $\Gamma \vdash K <_{c'} K'$ sont dérivables, alors $\Gamma \vdash c' = c : (K)K'$.

$UTT^{\vartheta}[C]$

Complétion des coercions

La transformation Ψ : l'idée.
Ψ transforme une dérivation dans $UTT^{\vartheta}[C]$ en une dérivation dans UTT^{ϑ}.

$UTT^{\vartheta}[C]$

Complétion des coercions

La transformation Ψ : l'idée.
Ψ transforme une dérivation dans $UTT^{\vartheta}[C]$ en une dérivation dans UTT^{ϑ}.

- Considérons une inférence faisant appel à la règle d'application coercive (CA.1)

$$\frac{\Gamma \vdash f : (x : K)K' \qquad \Gamma \vdash k : K_0 \qquad \Gamma \vdash K_0 <_c K}{\Gamma \vdash f(k) : [c(k)/x]K'} \ (\mathrm{CA.1})$$

$UTT^{\vartheta}[C]$

Complétion des coercions

La transformation Ψ : l'idée.
Ψ transforme une dérivation dans $UTT^{\vartheta}[C]$ en une dérivation dans UTT^{ϑ}.

- Considérons une inférence faisant appel à la règle d'application coercive $(CA.1)$

$$\frac{\Gamma \vdash f : (x : K)K' \qquad \Gamma \vdash k : K_0 \qquad \Gamma \vdash K_0 <_c K}{\Gamma \vdash f(k) : [c(k)/x]K'} \ (\mathrm{CA.1})$$

- on obtient

$$\frac{\Gamma \vdash f : (x : K)K' \qquad \dfrac{\Gamma \vdash c : (K_0)K \qquad \Gamma \vdash k : K_0}{\Gamma \vdash c(k) : K} \ (5.5)}{\Gamma \vdash f(c(k)) : [c(k)/x]K'} \ (5.5)$$

$UTT^{\vartheta}[C]$

Quelques définitions et notations

Définition(Constituants d'un jugement)

Nous appelons constituants d'un jugement $\Gamma \vdash J$, les éléments suivants :

- les sortes des variables de Γ ;
- les sortes des variables liées ;
- Si $J \equiv K\textbf{kind}$ ou $J \equiv K_1 = K_2$ alors K, respectivement K_1 et K_2 sont des constituants ;
- Si $J \equiv k : K$ ou $J \equiv k_1 = k_2 : K$ alors k et K, respectivement k_1, k_2 et K sont des constituants ;
- Si $J \equiv A <_c B : \textbf{Type}$ alors c, A, B et **Type** sont des constituants ;
- Si $J \equiv K_1 <_c K_2$ alors c,K_1 et K_2 sont des constituants.

Ces constituants sont considérés différents dans chacune de leurs occurrences.

$UTT^{\vartheta}[C]$

Complétion des coercions

Exemple de transformation

- prenons

$$d \equiv \frac{\Gamma \overset{d_1}{\vdash} k = k' : K \qquad \Gamma \overset{d_2}{\vdash} k' = k'' : K}{\Gamma \vdash k = k'' : K} \quad (2.6)$$

$UTT^{\vartheta}[C]$

Complétion des coercions

Exemple de transformation

- prenons

$$d \equiv \frac{\Gamma \stackrel{d_1}{\vdash} k = k' : K \qquad \Gamma \stackrel{d_2}{\vdash} k' = k'' : K}{\Gamma \vdash k = k'' : K}\ (2.6)$$

- on obtient $\Psi(d) \equiv$

$$\frac{\Gamma_1 \stackrel{\Psi(d_1)}{\vdash} k_1 = k_1' : K_1 \qquad \dfrac{\Gamma_1 \stackrel{?_1}{\vdash} k_1' = k_2' : K_1 \qquad \dfrac{\dfrac{\Gamma_2 \stackrel{\Psi(d_2)}{\vdash} k_2' = k_2'' : K_2 \qquad \stackrel{?_2}{\vdash} \Gamma_2 = \Gamma_1}{\Gamma_1 \vdash k_2' = k_2'' : K_2}\ (3.3) \qquad \Gamma_1 \stackrel{?_3}{\vdash} K_2 = K_1}{\Gamma_1 \vdash k_2' = k_2'' : K_1}\ (2.6)}{\Gamma_1 \vdash k_1' = k_2'' : K_1}\ (2.6)}{\Gamma_1 \vdash k_1 = k_2'' : K_1}$$

$UTT^{\vartheta}[C]$

Complétion des coercions

Exemple de transformation

- prenons

$$d \equiv \dfrac{\Gamma \overset{d_1}{\vdash} k = k' : K \qquad \Gamma \overset{d_2}{\vdash} k' = k'' : K}{\Gamma \vdash k = k'' : K}\ (2.6)$$

- on obtient $\Psi(d) \equiv$

$$\dfrac{\Gamma_1 \overset{\Psi(d_1)}{\vdash} k_1 = k'_1 : K_1 \qquad \dfrac{\Gamma_1 \overset{?_1}{\vdash} k'_1 = k'_2 : K_1 \qquad \dfrac{\dfrac{\Gamma_2 \overset{\Psi(d_2)}{\vdash} k'_2 = k''_2 : K_2 \qquad \overset{?_2}{\vdash} \Gamma_2 = \Gamma_1}{\Gamma_1 \vdash k'_2 = k''_2 : K_2}\ (3.3) \qquad \Gamma_1 \overset{?_3}{\vdash} K_2 = K_1}{\Gamma_1 \vdash k'_2 = k''_2 : K_1}}{\Gamma_1 \vdash k'_1 = k''_2 : K_1}\ (2.6)}{\Gamma_1 \vdash k_1 = k''_2 : K_1}\ (2.6)$$

Les égalités $?_1$, $?_2$ et $?_3$ sont prouvables si l'on peut prouver $\Psi_{d_1}(\Gamma \vdash k' : K) = \Psi_{d_2}(\Gamma \vdash k' : K)$

$UTT^{\vartheta}[C]$

Complétion des coercions

Nous avons démontré les lemmes suivants.

Lemme

Soit d une $UTT^{\vartheta}[C]$-dérivation d'un jugement J de la forme :
$\Gamma_1, \Gamma_2 \vdash G$, ou $\Gamma_1, x : K, \Gamma_2 \vdash G$, ou $\Gamma \vdash (x : K_1)K_2$, ou $\Gamma \vdash K_1 = K_2$, ou $\Gamma \vdash k_1 = k_2 : K$, ou $\Gamma \vdash \Sigma : K$ (Σ dénote un terme ou une égalité de termes), ou encore $\Gamma \vdash A <_c B :$ **Type** (ou $\Gamma \vdash K <_c K'$).
f est un des algorithmes de jugements présupposés tels que J' soit le jugement final de $f(d)$.
Si $\Psi(d)$ est définie alors $\Psi(f(d))$ est définie de plus

$$\Psi_{f(d)}(J') = \Psi_d(J')$$

$UTT^{\vartheta}[C]$

Complétion des coercions

Nous avons démontré les lemmes suivants.

Lemme

Soit d une $UTT^{\vartheta}[C]$-dérivation d'un jugement J de la forme :
$\Gamma_1, \Gamma_2 \vdash G$, ou $\Gamma_1, x : K, \Gamma_2 \vdash G$, ou $\Gamma \vdash (x : K_1)K_2$, ou $\Gamma \vdash K_1 = K_2$, ou $\Gamma \vdash k_1 = k_2 : K$, ou $\Gamma \vdash \Sigma : K$ (Σ dénote un terme ou une égalité de termes), ou encore $\Gamma \vdash A <_c B : \mathbf{Type}$ (ou $\Gamma \vdash K <_c K'$).
f est un des algorithmes de jugements présupposés tels que J' soit le jugement final de $f(d)$.
Si $\Psi(d)$ est définie alors $\Psi(f(d))$ est définie de plus

$$\Psi_{f(d)}(J') = \Psi_d(J')$$

Exemple

Soient les dérivations $\Gamma \overset{d}{\vdash} k = k' : K$ et $\Gamma \overset{r_=(d)}{\vdash} k' : K$.
Si $\Psi(d)$ est définie alors $\Psi(r_=(d))$ est définie, de plus
$\Psi_{r_=(d)}(\Gamma \vdash k' : K) = \Psi_d(\Gamma \vdash k' : K)$

$UTT^{\vartheta}[C]$

Complétion des coercions

ainsi que

Lemme

Présumons que les conditions de cohérence sont satisfaites pour le sous-typage. Soient d et d' deux dérivations dans $UTT^{\vartheta}[C]^{-}$ du même jugement : a)$\Gamma \vdash \textbf{valid}$ ou b)$\Gamma \vdash K\textbf{kind}$; ou de deux jugements : c)$\Gamma \vdash k : K$ et $\Gamma \vdash k : K'$.
Si $\Psi(d)$ et $\Psi(d')$ sont définies alors

a) $\Psi_d(\Gamma \vdash \textbf{valid}) = \Psi_{d'}(\Gamma \vdash \textbf{valid})$,

b) $\Psi_d(\Gamma \vdash K\textbf{kind}) = \Psi_{d'}(\Gamma \vdash K\textbf{kind})$,

c) $\Psi_d(\Gamma \vdash k : K) = \Psi_{d'}(\Gamma \vdash k : K')$.

$UTT^{\vartheta}[C]$

Complétion des coercions

Les deux lemmes précédents nous permettent de prouver que :

Théorème (Ψ est totale)

Si C est cohérent, alors la transformation Ψ est définie pour toute dérivation dans $UTT^{\vartheta}[C]$.

car nous serons en mesure de prouver les égalités nécessaires à la construction de $\Psi(d)$.

$UTT^{\vartheta}[C]$

Complétion des coercions

Reprenons l'exemple vu auparavant :

- prenons

$$d \equiv \dfrac{\Gamma \overset{d_1}{\vdash} k = k' : K \qquad \Gamma \overset{d_2}{\vdash} k' = k'' : K}{\Gamma \vdash k = k'' : K}\;(2.6)$$

- on obtient $\Psi(d) \equiv$

$$\dfrac{\Gamma_1 \overset{\Psi(d_1)}{\vdash} k_1 = k_1' : K_1 \qquad \dfrac{\Gamma_1 \overset{?_1}{\vdash} k_1' = k_2' : K_1 \qquad \dfrac{\dfrac{\Gamma_2 \overset{\Psi(d_2)}{\vdash} k_2' = k_2'' : K_2 \qquad \overset{?_2}{\vdash} \Gamma_2 = \Gamma_1}{\Gamma_1 \vdash k_2' = k_2'' : K_2}\;(3.3) \qquad \Gamma_1 \overset{?_3}{\vdash} K_2 = K_1}{\Gamma_1 \vdash k_2' = k_2'' : K_1}}{\Gamma_1 \vdash k_1' = k_2'' : K_1}\;(2.6)}{\Gamma_1 \vdash k_1 = k_2'' : K_1}\;(2.6)$$

$UTT^{\vartheta}[C]$

Complétion des coercions

Reprenons l'exemple vu auparavant :

- prenons

$$d \equiv \dfrac{\Gamma \overset{d_1}{\vdash} k = k' : K \qquad \Gamma \overset{d_2}{\vdash} k' = k'' : K}{\Gamma \vdash k = k'' : K}\ (2.6)$$

- on obtient $\Psi(d) \equiv$

$$\dfrac{\Gamma_1 \overset{\Psi(d_1)}{\vdash} k_1 = k_1' : K_1 \qquad \dfrac{\Gamma_1 \overset{?_1}{\vdash} k_1' = k_2' : K_1 \qquad \dfrac{\dfrac{\Gamma_2 \overset{\Psi(d_2)}{\vdash} k_2' = k_2'' : K_2 \qquad \overset{?_2}{\vdash} \Gamma_2 = \Gamma_1}{\Gamma_1 \vdash k_2' = k_2'' : K_2}\ (3.3) \qquad \Gamma_1 \overset{?_3}{\vdash} K_2 = K_1}{\Gamma_1 \vdash k_2' = k_2'' : K_1}}{\Gamma_1 \vdash k_1' = k_2'' : K_1}\ (2.6)}{\Gamma_1 \vdash k_1 = k_2'' : K_1}\ (2.6)$$

Grâce aux lemmes précédents on sait que :

- $\Psi_{r_=(d_1)}(\Gamma \vdash k' : K) = \Psi_{l_=(d_2)}(\Gamma \vdash k' : K)$, car $r_=(d_1)$ et $l_=(d_2)$ sont deux dérivations du même jugement.
- $\Psi_{d_1}(\Gamma \vdash k' : K) = \Psi_{r_=(d_1)}(\Gamma \vdash k' : K)$, et $\Psi_{d_2}(\Gamma \vdash k' : K) = \Psi_{l_=(d_2)}(\Gamma \vdash k' : K)$

Ce qui nous permet d'obtenir $\Psi_{d_1}(\Gamma \vdash k' : K) = \Psi_{d_2}(\Gamma \vdash k' : K)$

$UTT^{\vartheta}[C]$

Conservativité

Théorème (Conservativité)

Soit J un jugement dérivable dans $UTT^{\vartheta}[C]$. Si C est cohérent et J n'est pas un jugement de sous-typage ou de sous-sortes, alors J est dérivable dans UTT^{ϑ} si et seulement si il existe une $UTT^{\vartheta}[C]$-dérivation d telle que $\Psi_d(J) \equiv J$.

Sommaire

Récapitulatif

Nous avons défini le calcul $UTT^{\vartheta}[C]$ qui est une théorie des types comprenant :

- des types dépendants,
- des types inductifs,
- un univers imprédicatif des propositions,
- un univers prédicatif de types,
- une réduction non standard pour les fonctions entre types finis (ϑ-réduction), en plus des réductions standards,
- la notion de sous-typage coercitif.

Récapitulatif

Nous avons défini le calcul $UTT^{\vartheta}[C]$ qui est une théorie des types comprenant :

- des types dépendants,
- des types inductifs,
- un univers imprédicatif des propositions,
- un univers prédicatif de types,
- une réduction non standard pour les fonctions entre types finis (ϑ-réduction), en plus des réductions standards,
- la notion de sous-typage coercitif.

Nous nous sommes également assurés que les propriétés suivantes sont vérifiées :

- de normalisation forte,
- de confluence et
- de préservation du type,
- $UTT^{\vartheta}[C]$ est extension conservative de UTT^{ϑ}.

Intérêts

Enrichissement la classe des systèmes de coercions cohérents :

- Prenons F_2, F_3, F_4, et F_5 des types finis,
- Les coercions
  - $c_1 \equiv E[\bar{\Theta}_3]([y : F_3]F_4, 1_{F_5}, 2_{F_5}, 3_{F_5}) : (F_3)F_5$, avec $(\forall i_{F_3} : F_3, c_1(i_{F_3}) = i_{F_5})$ et
  - $c_2 \equiv E[\bar{\Theta}_2]([z : F_2]F_3, 1_{F_3}, 2_{F_3}) : (F_2)F_3$, avec $(\forall i_{F_2} : F_2, c_2(i_{F_2}) = i_{F_3})$
  - $c_3 \equiv E[\bar{\Theta}_4]([y : F_4]F_5, 1_{F_5}, 2_{F_5}, 3_{F_5}, 4_{F_5}) : (F_4)F_5$, avec $(\forall i_{F_4} : F_4, c_3(i_{F_4}) = i_{F_5})$
  - $c_4 \equiv E[\bar{\Theta}_2]([z : F_2]F_4, 1_{F_4}, 2_{F_4}) : (F_2)F_4$, avec $(\forall i_{F_2} : F_2, c_4(i_{F_2}) = i_{F_4})$

Intérêts

Enrichissement la classe des systèmes de coercions cohérents :

- Prenons F_2, F_3, F_4, et F_5 des types finis,
- Les coercions
  - $c_1 \equiv E[\bar{\Theta}_3]([y : F_3]F_4, 1_{F_5}, 2_{F_5}, 3_{F_5}) : (F_3)F_5$, avec $(\forall i_{F_3} : F_3, c_1(i_{F_3}) = i_{F_5})$ et
  - $c_2 \equiv E[\bar{\Theta}_2]([z : F_2]F_3, 1_{F_3}, 2_{F_3}) : (F_2)F_3$, avec $(\forall i_{F_2} : F_2, c_2(i_{F_2}) = i_{F_3})$
  - $c_3 \equiv E[\bar{\Theta}_4]([y : F_4]F_5, 1_{F_5}, 2_{F_5}, 3_{F_5}, 4_{F_5}) : (F_4)F_5$, avec $(\forall i_{F_4} : F_4, c_3(i_{F_4}) = i_{F_5})$
  - $c_4 \equiv E[\bar{\Theta}_2]([z : F_2]F_4, 1_{F_4}, 2_{F_4}) : (F_2)F_4$, avec $(\forall i_{F_2} : F_2, c_4(i_{F_2}) = i_{F_4})$

$$\begin{tikzcd}
 & F_3 \arrow[dr, "c_1"] & \\
F_2 \arrow[ur, "c_2"] \arrow[dr, "c_4"'] & & F_5 \\
 & F_4 \arrow[ur, "c_3"'] &
\end{tikzcd}$$

Intérêts

Enrichissement la classe des systèmes de coercions cohérents :

- Prenons F_2, F_3, F_4, et F_5 des types finis,
- Les coercions
  - $c_1 \equiv E[\bar{\Theta}_3]([y : F_3]F_4, 1_{F_5}, 2_{F_5}, 3_{F_5}) : (F_3)F_5$, avec $(\forall i_{F_3} : F_3, c_1(i_{F_3}) = i_{F_5})$ et
  - $c_2 \equiv E[\bar{\Theta}_2]([z : F_2]F_3, 1_{F_3}, 2_{F_3}) : (F_2)F_3$, avec $(\forall i_{F_2} : F_2, c_2(i_{F_2}) = i_{F_3})$
  - $c_3 \equiv E[\bar{\Theta}_4]([y : F_4]F_5, 1_{F_5}, 2_{F_5}, 3_{F_5}, 4_{F_5}) : (F_4)F_5$, avec $(\forall i_{F_4} : F_4, c_3(i_{F_4}) = i_{F_5})$
  - $c_4 \equiv E[\bar{\Theta}_2]([z : F_2]F_4, 1_{F_4}, 2_{F_4}) : (F_2)F_4$, avec $(\forall i_{F_2} : F_2, c_4(i_{F_2}) = i_{F_4})$

$$
\begin{tikzcd}
 & F_3 \arrow[dr, "c_1"] & \\
F_2 \arrow[ur, "c_2"] \arrow[dr, "c_4"'] & & F_5 \\
 & F_4 \arrow[ur, "c_3"'] &
\end{tikzcd}
$$

- $c_1 \circ c_2(x) \rhd_{\vartheta} h(x)$ et $c_3 \circ c_4(x) \rhd_{\vartheta} h(x)$,
  avec $h \equiv E[\bar{\Theta}_2]([x : F_2]F_5, 1_{F_5}, 2_{F_5}) : (F_2)F_5$

Intérêts

Enrichissement la classe des systèmes de coercions cohérents :

- Prenons F_2, F_3, F_4, et F_5 des types finis,
- Les coercions
  - $c_1 \equiv E[\bar{\Theta}_3]([y : F_3]F_4, 1_{F_5}, 2_{F_5}, 3_{F_5}) : (F_3)F_5$, avec $(\forall i_{F_3} : F_3, c_1(i_{F_3}) = i_{F_5})$ et
  - $c_2 \equiv E[\bar{\Theta}_2]([z : F_2]F_3, 1_{F_3}, 2_{F_3}) : (F_2)F_3$, avec $(\forall i_{F_2} : F_2, c_2(i_{F_2}) = i_{F_3})$
  - $c_3 \equiv E[\bar{\Theta}_4]([y : F_4]F_5, 1_{F_5}, 2_{F_5}, 3_{F_5}, 4_{F_5}) : (F_4)F_5$, avec $(\forall i_{F_4} : F_4, c_3(i_{F_4}) = i_{F_5})$
  - $c_4 \equiv E[\bar{\Theta}_2]([z : F_2]F_4, 1_{F_4}, 2_{F_4}) : (F_2)F_4$, avec $(\forall i_{F_2} : F_2, c_4(i_{F_2}) = i_{F_4})$

$$
\begin{tikzcd}
 & F_3 \arrow[dr, "c_1"] & \\
F_2 \arrow[ur, "c_2"] \arrow[dr, "c_4"'] & & F_5 \\
 & F_4 \arrow[ur, "c_3"'] &
\end{tikzcd}
$$

- $c_1 \circ c_2(x) \rhd_\vartheta h(x)$ et $c_3 \circ c_4(x) \rhd_\vartheta h(x)$,
  avec $h \equiv E[\bar{\Theta}_2]([x : F_2]F_5, 1_{F_5}, 2_{F_5}) : (F_2)F_5$
- Système cohérent dans $UTT^\vartheta[C]$ car $c_1 \circ c_2(x) = c_3 \circ c_4(x)$.

Intérêts

L'intérêt d'un système comme $UTT^{\vartheta}[C]$ est manifeste pour tous les problèmes faisant intervenir des ensembles finis (combinatoire, manipulation de graphes, etc).

- Amélioration des performances des assistants à la preuve,
- Enrichissement de la classe des systèmes de coercions cohérents.(Il suffit de vérifier la cohérence au niveau ensembliste)