Théorie des ensembles dans la théorie homotopique des types

Jérémy Ledent,
encadré par Bas Spitters & Freek Wiedijk

Radboud University of Nijmegen, Pays Bas

3 septembre 2014

Théorie homotopique des types (HoTT)

- Théorie des types dépendants (Per Martin-Löf, 1975).

Théorie homotopique des types (HoTT)

- Théorie des types dépendants (Per Martin-Löf, 1975).
- Calcul des constructions inductives (T. Coquand, 1984).

Théorie homotopique des types (HoTT)

- Théorie des types dépendants (Per Martin-Löf, 1975).
- Calcul des constructions inductives (T. Coquand, 1984).
- Coq (1989).

Théorie homotopique des types (HoTT)

- Théorie des types dépendants (Per Martin-Löf, 1975).
- Calcul des constructions inductives (T. Coquand, 1984).
- Coq (1989).

$$\boxed{\text{HoTT} = \text{MLTT} + \text{Univalence} + \text{Types inductifs supérieurs}}$$

Théorie homotopique des types (HoTT)

- Théorie des types dépendants (Per Martin-Löf, 1975).
- Calcul des constructions inductives (T. Coquand, 1984).
- Coq (1989).

$$\boxed{\text{HoTT} = \text{MLTT} + \text{Univalence} + \text{Types inductifs supérieurs}}$$

- Nouveaux fondements des mathématiques.

Théorie homotopique des types (HoTT)

- Théorie des types dépendants (Per Martin-Löf, 1975).
- Calcul des constructions inductives (T. Coquand, 1984).
- Coq (1989).

$$\boxed{\text{HoTT} = \text{MLTT} + \text{Univalence} + \text{Types inductifs supérieurs}}$$

- Nouveaux fondements des mathématiques.
- De forts lien avec la théorie de l'homotopie.

Théorie homotopique des types (HoTT)

- Théorie des types dépendants (Per Martin-Löf, 1975).
- Calcul des constructions inductives (T. Coquand, 1984).
- Coq (1989).

$$\boxed{\text{HoTT} = \text{MLTT} + \text{Univalence} + \text{Types inductifs supérieurs}}$$

- Nouveaux fondements des mathématiques.
- De forts lien avec la théorie de l'homotopie.
- Mathématiques constructives.

Correspondance de Curry-Howard

- $\mathcal{U}$ est un **univers**.
- $A : \mathcal{U}$ est un **type**.
- $x : A$ est un **terme**.

Types	Logique	Ensembles
A	proposition	ensemble
$x : A$	preuve	élément

Correspondance de Curry-Howard

- $\mathcal{U}$ est un **univers**.
- $A : \mathcal{U}$ est un **type**.
- $x : A$ est un **terme**.

Types	Logique	Ensembles
A	proposition	ensemble
$x : A$	preuve	élément
$A \times B$	$A \wedge B$	produit cartésien

Correspondance de Curry-Howard

- $\mathcal{U}$ est un **univers**.
- $A : \mathcal{U}$ est un **type**.
- $x : A$ est un **terme**.

Types	Logique	Ensembles
A	proposition	ensemble
$x : A$	preuve	élément
$A \times B$	$A \wedge B$	produit cartésien
$A + B$	$A \vee B$	union disjointe

Correspondance de Curry-Howard

- $\mathcal{U}$ est un **univers**.
- $A : \mathcal{U}$ est un **type**.
- $x : A$ est un **terme**.

Types	Logique	Ensembles
A	proposition	ensemble
$x : A$	preuve	élément
$A \times B$	$A \wedge B$	produit cartésien
$A + B$	$A \vee B$	union disjointe
$A \to B$	$A \Rightarrow B$	fonctions de A dans B

Correspondance de Curry-Howard

- $\mathcal{U}$ est un **univers**.
- $A : \mathcal{U}$ est un **type**.
- $x : A$ est un **terme**.

Types	Logique	Ensembles
A	proposition	ensemble
$x : A$	preuve	élément
$A \times B$	$A \wedge B$	produit cartésien
$A + B$	$A \vee B$	union disjointe
$A \to B$	$A \Rightarrow B$	fonctions de A dans B
$P : A \to \mathcal{U}$	prédicat	famille d'ensembles

Correspondance de Curry-Howard

- $\mathcal{U}$ est un **univers**.
- $A : \mathcal{U}$ est un **type**.
- $x : A$ est un **terme**.

Types	Logique	Ensembles
A	proposition	ensemble
$x : A$	preuve	élément
$A \times B$	$A \wedge B$	produit cartésien
$A + B$	$A \vee B$	union disjointe
$A \to B$	$A \Rightarrow B$	fonctions de A dans B
$P : A \to \mathcal{U}$	prédicat	famille d'ensembles
$\prod_{(x:A)} P(x)$	$\forall_{x:A} P(x)$	produit cartésien

Correspondance de Curry-Howard

- $\mathcal{U}$ est un **univers**.
- $A : \mathcal{U}$ est un **type**.
- $x : A$ est un **terme**.

Types	Logique	Ensembles
A	proposition	ensemble
$x : A$	preuve	élément
$A \times B$	$A \wedge B$	produit cartésien
$A + B$	$A \vee B$	union disjointe
$A \to B$	$A \Rightarrow B$	fonctions de A dans B
$P : A \to \mathcal{U}$	prédicat	famille d'ensembles
$\prod_{(x:A)} P(x)$	$\forall_{x:A} P(x)$	produit cartésien
$\sum_{(x:A)} P(x)$	$\exists_{x:A} P(x)$	union disjointe, sous-ensemble

Égalité propositionnelle

- Pour $x, y : A$, on aimerait définir un type $(x = y)$.
  $\longrightarrow$ Type inductif

Égalité propositionnelle

- Pour $x, y : A$, on aimerait définir un type $(x = y)$.
  $\longrightarrow$ Type inductif
- Un seul constructeur

$$\mathsf{refl} : \prod_{x:A} (x = x)$$

Égalité propositionnelle

- Pour $x, y : A$, on aimerait définir un type $(x = y)$.
  $\longrightarrow$ Type inductif
- Un seul constructeur

$$\mathrm{refl} : \prod_{x:A} (x = x)$$

- Principe d'induction :
  « Pour prouver $\prod_{(x,y)} \prod_{(p:x=y)} C(x, y, p)$
  il suffit de prouver $C(x, x, \mathrm{refl}_x)$ *pour tout* x. *»*

Égalité propositionnelle

- Pour $x, y : A$, on aimerait définir un type $(x = y)$.
  $\longrightarrow$ Type inductif
- Un seul constructeur

$$\mathrm{refl} : \prod_{x:A} (x = x)$$

- Principe d'induction :
  « Pour prouver $\prod_{(x,y)} \prod_{(p:x=y)} C(x, y, p)$
  il suffit de prouver $C(x, x, \mathrm{refl}_x)$ *pour tout* x*. »*
- N'implique pas : ~~« La seule preuve d'égalité est refl. »~~

Interprétation homotopique

- Axiome K / UIP : $\prod_{(x,y)} \prod_{(p,q:x=y)} (p = q)$

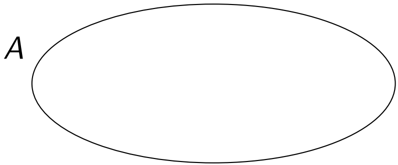
Interprétation homotopique

- ~~Axiome *K* / *UIP* : $\prod_{(x,y)} \prod_{(p,q:x=y)}(p = q)$~~
- V. Voevodsky (2006) : Modèle de MLTT avec des "ensembles simpliciaux".

Interprétation homotopique

- ~~Axiome K / UIP : $\prod_{(x,y)}\prod_{(p,q:x=y)}(p=q)$~~
- V. Voevodsky (2006) : Modèle de MLTT avec des "ensembles simpliciaux".

type A $\leftrightarrow$ espace topologique

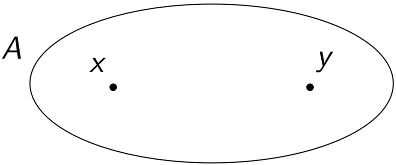


Interprétation homotopique

- ~~Axiome K / UIP : $\prod_{(x,y)} \prod_{(p,q:x=y)}(p = q)$~~

- V. Voevodsky (2006) : Modèle de MLTT avec des "ensembles simpliciaux".

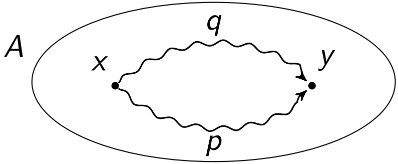
type A	$\leftrightarrow$	espace topologique
termes $x, y : A$	$\leftrightarrow$	points



Interprétation homotopique

- ~~Axiome K / UIP : $\prod_{(x,y)} \prod_{(p,q:x=y)} (p = q)$~~
- V. Voevodsky (2006) : Modèle de MLTT avec des “ensembles simpliciaux”.

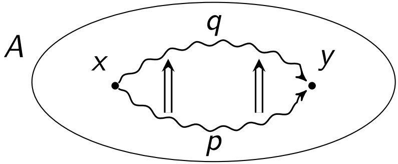
type A	$\leftrightarrow$	espace topologique
termes $x, y : A$	$\leftrightarrow$	points
$p, q : x = y$	$\leftrightarrow$	chemins de x vers y



Interprétation homotopique

- ~~Axiome K / UIP : $\prod_{(x,y)} \prod_{(p,q:x=y)}(p = q)$~~
- V. Voevodsky (2006) : Modèle de MLTT avec des “ensembles simpliciaux”.

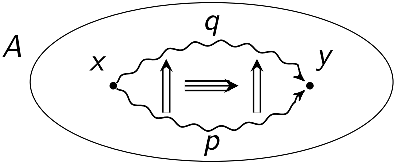
type A	$\leftrightarrow$	espace topologique
termes $x, y : A$	$\leftrightarrow$	points
$p, q : x = y$	$\leftrightarrow$	chemins de x vers y



Interprétation homotopique

- ~~Axiome K / UIP : $\prod_{(x,y)} \prod_{(p,q:x=y)} (p = q)$~~
- V. Voevodsky (2006) : Modèle de MLTT avec des "ensembles simpliciaux".

type A	$\leftrightarrow$	espace topologique
termes $x, y : A$	$\leftrightarrow$	points
$p, q : x = y$	$\leftrightarrow$	chemins de x vers y



h-Niveaux

Définition

- A est une h-proposition si $\prod_{(x,y:A)}(x = y)$.
- hProp $:= \sum_{(A:\mathcal{U})} \mathsf{ishProp}(A)$.

h-Niveaux

Définition

- A est une h-proposition si $\prod_{(x,y:A)}(x = y)$.
- hProp $:= \sum_{(A:\mathcal{U})} \mathrm{ishProp}(A)$.
- A est un h-ensemble si $\prod_{(x,y:A)} \prod_{(p,q:x=y)}(p = q)$.
- hSet $:= \sum_{(A:\mathcal{U})} \mathrm{ishSet}(A)$.

h-Niveaux

Définition

- A est une h-proposition si $\prod_{(x,y:A)}(x = y)$.
- hProp $:= \sum_{(A:\mathcal{U})} \mathrm{ishProp}(A)$.
- A est un h-ensemble si $\prod_{(x,y:A)} \prod_{(p,q:x=y)}(p = q)$.
- hSet $:= \sum_{(A:\mathcal{U})} \mathrm{ishSet}(A)$.

Connecteurs logiques usuels :

$$\begin{array}{rcl} A \vee B & := & \|A + B\| \\ \exists_{x:A} P(x) & := & \left\| \sum_{(x:A)} P(x) \right\| \end{array}$$

h-Niveaux

Définition

- A est une h-proposition si $\prod_{(x,y:A)}(x = y)$.
- hProp $:= \sum_{(A:\mathcal{U})} \mathrm{ishProp}(A)$.
- A est un h-ensemble si $\prod_{(x,y:A)} \prod_{(p,q:x=y)}(p = q)$.
- hSet $:= \sum_{(A:\mathcal{U})} \mathrm{ishSet}(A)$.

Connecteurs logiques usuels :

$$\begin{array}{rcl} A \vee B & := & \|A + B\| \\ \exists_{x:A} P(x) & := & \left\|\sum_{(x:A)} P(x)\right\| \end{array}$$

$\longrightarrow$ hSet est un modèle de la théorie des ensembles structurelle (E. Rijke & B. Spitters, 2013).

Axiome d'Univalence

Proposé par V. Voevodsky (2009).

Axiome d'Univalence

Proposé par V. Voevodsky (2009).

Définition

Les types A et B sont équivalents, noté $A \simeq B$, ssi :

- $f : A \to B$
- $g : B \to A$
- $\prod_{(a:A)}(g \circ f(a) = a)$
- $\prod_{(b:B)}(f \circ g(b) = b)$

Axiome d'Univalence

Proposé par V. Voevodsky (2009).

Définition

Les types A et B sont équivalents, noté $A \simeq B$, ssi :

- $f : A \to B$
- $g : B \to A$
- $\prod_{(a:A)}(g \circ f(a) = a)$
- $\prod_{(b:B)}(f \circ g(b) = b)$

Axiome d'Univalence :

$$\boxed{(A = B) \simeq (A \simeq B)}$$

Types inductifs supérieurs

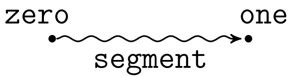
Proposé par M. Shulman, P. Lumsdaine (2011).

Types inductifs supérieurs

Proposé par M. Shulman, P. Lumsdaine (2011).

Exemple

```
Inductive interval : Type :=
  | zero : interval
  | one : interval
  | segment : zero = one.
```



Définition

Soit V le type inductif supérieur généré par :

- $\mathrm{set}(A, f) : V$, où $A : \mathcal{U}$ et $f : (A \to V)$.
  $\longrightarrow$ *Idée :* $\mathrm{set}(A, f) := \{f(a) \mid a : A\}$

Définition

Soit V le type inductif supérieur généré par :

- $\mathrm{set}(A, f) : V$, où $A : \mathcal{U}$ et $f : (A \to V)$.
  $\longrightarrow$ *Idée :* $\mathrm{set}(A, f) := \{f(a) \mid a : A\}$
- setext : Si on a A, B, f et g tels que

$$(\forall a, \exists b, f(a) = g(b)) \wedge (\forall b, \exists a, f(a) = g(b))$$

alors, $\quad \mathrm{set}(A, f) = \mathrm{set}(B, g)$

Définition

Soit V le type inductif supérieur généré par :

- $\mathrm{set}(A, f) : V$, où $A : \mathcal{U}$ et $f : (A \to V)$.
  $\longrightarrow$ *Idée :* $\mathrm{set}(A, f) := \{f(a) \mid a : A\}$
- setext : Si on a A, B, f et g tels que
  $$(\forall a, \exists b, f(a) = g(b)) \wedge (\forall b, \exists a, f(a) = g(b))$$
  alors, $\quad \mathrm{set}(A, f) = \mathrm{set}(B, g)$
- 0-trunc : $\prod_{(x,y:V)} \prod_{(p,q:x=y)} (p = q)$.

Définition

Soit V le type inductif supérieur généré par :

- $\mathrm{set}(A, f) : V$, où $A : \mathcal{U}$ et $f : (A \to V)$.
  $\longrightarrow$ *Idée :* $\mathrm{set}(A, f) := \{f(a) \mid a : A\}$
- setext : Si on a A, B, f et g tels que

$$(\forall a, \exists b, f(a) = g(b)) \wedge (\forall b, \exists a, f(a) = g(b))$$

  alors, $\quad \mathrm{set}(A, f) = \mathrm{set}(B, g)$
- 0-trunc : $\prod_{(x,y:V)} \prod_{(p,q:x=y)} (p = q)$.

Exemples

- $\varnothing := \mathrm{set}(\bot, \mathrm{rec}_\bot(V))$

Définition

Soit V le type inductif supérieur généré par :

- $\mathrm{set}(A, f) : V$, où $A : \mathcal{U}$ et $f : (A \to V)$.
  $\longrightarrow$ *Idée :* $\mathrm{set}(A, f) := \{f(a) \mid a : A\}$
- setext : Si on a A, B, f et g tels que

$$(\forall a, \exists b, f(a) = g(b)) \wedge (\forall b, \exists a, f(a) = g(b))$$

  alors, $\quad \mathrm{set}(A, f) = \mathrm{set}(B, g)$
- 0-trunc : $\prod_{(x,y:V)} \prod_{(p,q:x=y)} (p = q)$.

Exemples

- $\varnothing := \mathrm{set}(\bot, \mathrm{rec}_\bot(V))$
- Si $u : V$, on construit $\{u\} := \mathrm{set}(Unit, \lambda x.\, u)$.

Propriétés

On définit la relation $\in : V \to V \to \text{hProp}$:

Définition

$u \in set(A, f) := \left(\exists (a : A).\, f(a) = u\right)$

Propriétés

On définit la relation $\in : V \to V \to \mathrm{hProp}$:

Définition

$$u \in set(A, f) := \left(\exists (a : A).\, f(a) = u\right)$$

Théorème

$(V, \in)$ vérifie les axiomes suivants :

extensionalité	paire	fonction
ensemble vide	infini	remplacement
$\in$-induction	union	séparation

Propriétés

On définit la relation $\in : V \to V \to \mathrm{hProp}$:

Définition

$$u \in set(A, f) := \left(\exists (a : A).\, f(a) = u\right)$$

Théorème

$(V, \in)$ vérifie les axiomes suivants :

extensionalité	paire	fonction
ensemble vide	infini	remplacement
$\in$-induction	union	séparation

Théorème

Dans HoTT + axiome du choix, $(V, \in)$ vérifie tous les axiomes de la théorie des ensembles de Zermelo-Fraenkel, ZFC.

Théorie Homotopique des Types

La hiérarchie cumulative

Formalisation en Coq

Conclusion

Implémentations de HoTT

$\longrightarrow$ Pas de "vraie" implémentation à l'heure actuelle.

Implémentations de HoTT

$\longrightarrow$ Pas de "vraie" implémentation à l'heure actuelle.

- Librairies Coq et Agda.
  $\rightarrow$ Types privés.

Implémentations de HoTT

$\longrightarrow$ Pas de “vraie” implémentation à l’heure actuelle.

- Librairies Coq et Agda.
  $\rightarrow$ Types privés.
- Types inductifs supérieurs en Coq (B. Barras, 2013).

Implémentations de HoTT

$\longrightarrow$ Pas de "vraie" implémentation à l'heure actuelle.

- Librairies Coq et Agda.
  $\rightarrow$ Types privés.
- Types inductifs supérieurs en Coq (B. Barras, 2013).
- Cubical (T. Coquand, 2014).

Implémentations de HoTT

$\longrightarrow$ Pas de "vraie" implémentation à l'heure actuelle.

- Librairies COQ et AGDA.
  $\rightarrow$ Types privés.
- Types inductifs supérieurs en COQ (B. Barras, 2013).
- CUBICAL (T. Coquand, 2014).

Types "Privés"

- Utilisés pour implémenter les types inductifs supérieurs.

Types “Privés”

- Utilisés pour implémenter les types inductifs supérieurs.
- Présents en Agda à l'origine.
- Implémentés en Coq par Y. Berthot (2013).

Types "Privés"

- Utilisés pour implémenter les types inductifs supérieurs.
- Présents en Agda à l'origine.
- Implémentés en Coq par Y. Berthot (2013).

Idée :

```
Inductive I : Type :=
  | zero : I
  | one : I.
Axiom segment : zero = one.
```

Types “Privés”

- Utilisés pour implémenter les types inductifs supérieurs.
- Présents en AGDA à l'origine.
- Implémentés en COQ par Y. Berthot (2013).

Idée :

```
Inductive I : Type :=
  | zero : I
  | one : I.
Axiom segment : zero = one.
```

Problème

Principe d'induction trop fort.
$\longrightarrow$ Notation `Private Inductive` pour le cacher.

Univers explicites

- Polymorphisme d'univers (M. Sozeau, 2014).
- Coq version "Trunk".

Univers explicites

- Polymorphisme d'univers (M. Sozeau, 2014).
- Coq version "Trunk".
- Ajouts récents, non-documentés, quelques bugs...

Univers explicites

- Polymorphisme d'univers (M. Sozeau, 2014).
- Coq version "Trunk".
- Ajouts récents, non-documentés, quelques bugs...

Sans Polymorphisme :

`Universe U U'.` $\rightarrow$ fixe deux niveaux d'univers.

Univers explicites

- Polymorphisme d'univers (M. Sozeau, 2014).
- Coq version "Trunk".
- Ajouts récents, non-documentés, quelques bugs...

Sans Polymorphisme :

`Universe U U'.` $\rightarrow$ fixe deux niveaux d'univers.

`Type@{U}` $\rightarrow$ univers au niveau `U`.

Univers explicites

- Polymorphisme d'univers (M. Sozeau, 2014).
- Coq version "Trunk".
- Ajouts récents, non-documentés, quelques bugs...

Sans Polymorphisme :

`Universe U U'.` $\rightarrow$ fixe deux niveaux d'univers.

`Type@{U}` $\rightarrow$ univers au niveau U.

```
Inductive V : Type@{U'} :=
  | set : forall (A : Type@{U}), ...
  | ...
```

Univers explicites

- Polymorphisme d'univers (M. Sozeau, 2014).
- Coq version "Trunk".
- Ajouts récents, non-documentés, quelques bugs...

Sans Polymorphisme :

`Universe U U'.` $\rightarrow$ fixe deux niveaux d'univers.

`Type@{U}` $\rightarrow$ univers au niveau `U`.

```
Inductive V : Type@{U'} :=
  | set : forall (A : Type@{U}), ...
  | ...
```

Avec Polymorphisme :

`V@{U' U}` $\rightarrow$ V où les deux univers sont `U` et `U'`.

Résultats

- Une section du “HoTT Book” formalisée.

Résultats

- Une section du “HoTT Book” formalisée.
- $\simeq$ 800 lignes de code.

Résultats

- Une section du "HoTT Book" formalisée.
- $\simeq$ 800 lignes de code.
- Ajouté à la librairie HoTT de Coq.

Résultats

- Une section du “HoTT Book” formalisée.
- $\simeq$ 800 lignes de code.
- Ajouté à la librairie HoTT de Coq.

Ce qu’a apporté la formalisation :

- Principe d’induction de V corrigé.

Résultats

- Une section du "HoTT Book" formalisée.
- $\simeq$ 800 lignes de code.
- Ajouté à la librairie HoTT de Coq.

Ce qu'a apporté la formalisation :

- Principe d'induction de V corrigé.
- Renforcement d'un lemme.
  $\longrightarrow$ Lien entre V et hSet.

Conclusion

- Contribution à la librairie HoTT de COQ.

Conclusion

- Contribution à la librairie HoTT de COQ.
- Expérimentation sur les types inductifs supérieurs :
  – V est “non-standard”.
  – Principe d'induction corrigé.

Conclusion

- Contribution à la librairie HoTT de Coq.
- Expérimentation sur les types inductifs supérieurs :
  - V est “non-standard”.
  - Principe d’induction corrigé.

Merci de votre attention !