Théorie des types sous-extensionnelle

Jérémy Ledent,
encadré par Hugo Herbelin

Laboratoire PPS, équipe πr^2, Université Paris 7

3 septembre 2015

Pure Type Systems (PTS)

Spécification d'un PTS :

- Un ensemble de sortes S.
- Un ensemble d'axiomes $\mathsf{Ax} \in \mathsf{S} \times \mathsf{S}$.
- Un ensemble de relations $\mathsf{Rel} \in \mathsf{S} \times \mathsf{S} \times \mathsf{S}$.

$$M, N ::= x \in \mathcal{V} \mid s \in \mathsf{S} \mid MN \mid \lambda x : M.N \mid \Pi x : M.N$$

Pure Type Systems (PTS)

$$\textsc{Nil}\ \frac{}{\vdash \mathsf{wf}}$$

$$\textsc{Cons}\ \frac{\Gamma \vdash A : s \qquad x \notin \mathsf{dom}(\Gamma)}{\Gamma, x : A \vdash \mathsf{wf}}$$

$$\textsc{Sort}\ \frac{\Gamma \vdash \mathsf{wf} \qquad (s_1, s_2) \in \mathsf{Ax}}{\Gamma \vdash s_1 : s_2}$$

$$\textsc{Var}\ \frac{\Gamma \vdash \mathsf{wf} \qquad \Gamma(x) = A}{\Gamma \vdash x : A}$$

$$\textsc{Prod}\ \frac{\Gamma \vdash A : s_1 \qquad \Gamma, x : A \vdash B : s_2 \qquad (s_1, s_2, s_3) \in \mathsf{Rel}}{\Gamma \vdash \Pi x : A.B : s_3}$$

$$\textsc{Lam}\ \frac{\Gamma \vdash A : s_1 \qquad \Gamma, x : A \vdash M : B \qquad \Gamma, x : A \vdash B : s_2 \qquad (s_1, s_2, s_3) \in \mathsf{Rel}}{\Gamma \vdash \lambda x : A.M : \Pi x : A.B}$$

$$\textsc{App}\ \frac{\Gamma \vdash M : \Pi x : A.B \qquad \Gamma \vdash N : A}{\Gamma \vdash M\ N : B[x := N]}$$

$$\textsc{Conv}\ \frac{\Gamma \vdash M : A \qquad \Gamma \vdash A' : s \qquad A \equiv A'}{\Gamma \vdash M : A'}$$

Pure Type Systems (PTS)

$$\textsc{Nil}\ \frac{}{\vdash \mathsf{wf}}$$

$$\textsc{Cons}\ \frac{\Gamma \vdash A : s \qquad x \notin \mathsf{dom}(\Gamma)}{\Gamma, x : A \vdash \mathsf{wf}}$$

$$\textsc{Sort}\ \frac{\Gamma \vdash \mathsf{wf} \qquad (s_1, s_2) \in \mathsf{Ax}}{\Gamma \vdash s_1 : s_2}$$

$$\textsc{Var}\ \frac{\Gamma \vdash \mathsf{wf} \qquad \Gamma(x) = A}{\Gamma \vdash x : A}$$

$$\textsc{Prod}\ \frac{\Gamma \vdash A : s_1 \qquad \Gamma, x : A \vdash B : s_2 \qquad (s_1, s_2, s_3) \in \mathsf{Rel}}{\Gamma \vdash \Pi x : A.B : s_3}$$

$$\textsc{Lam}\ \frac{\Gamma \vdash A : s_1 \qquad \Gamma, x : A \vdash M : B \qquad \Gamma, x : A \vdash B : s_2 \qquad (s_1, s_2, s_3) \in \mathsf{Rel}}{\Gamma \vdash \lambda x : A.M : \Pi x : A.B}$$

$$\textsc{App}\ \frac{\Gamma \vdash M : \Pi x : A.B \qquad \Gamma \vdash N : A}{\Gamma \vdash M\ N : B[x := N]}$$

$$\textsc{Conv}\ \frac{\Gamma \vdash M : A \qquad \Gamma \vdash A' : s \qquad A \equiv A'}{\Gamma \vdash M : A'}$$

Théorie intensionnelle (ITT)

Deux notions d'égalité cohabitent :

- La conversion '$\equiv$'
  - Dans les PTS, β-conversion $\equiv_\beta$.
  - En Coq, un peu plus riche : `let`, `match`, etc.

Théorie intensionnelle (ITT)

Deux notions d'égalité cohabitent :

- La conversion '$\equiv$'
  - Dans les PTS, β-conversion $\equiv_\beta$.
  - En Coq, un peu plus riche : `let`, `match`, etc.

Intuitions

$\rightarrow$ « Égal par définition »

Exemples :

- Si on a défini $f := x \mapsto x^2$, alors $f\ 3 \equiv 3^2$

Théorie intensionnelle (ITT)

Deux notions d'égalité cohabitent :

- La conversion '$\equiv$'
  - Dans les PTS, β-conversion $\equiv_\beta$.
  - En Coq, un peu plus riche : `let`, `match`, etc.

Intuitions

$\rightarrow$ « Égal par définition »
$\rightarrow$ Côté calculatoire du raisonnement

Exemples :

- Si on a défini $f := x \mapsto x^2$, alors $f\ 3 \equiv 3^2$
- $3^2 \equiv 9$

Théorie intensionnelle (ITT)

Deux notions d'égalité cohabitent :

- La conversion '$\equiv$'
  - Dans les PTS, β-conversion $\equiv_\beta$.
  - En Coq, un peu plus riche : `let`, `match`, etc.

Intuitions

$\rightarrow$ « Égal par définition »
$\rightarrow$ Côté calculatoire du raisonnement

Exemples :

- Si on a défini $f := x \mapsto x^2$, alors $f\ 3 \equiv 3^2$
- $3^2 \equiv 9$
- $x + 0 \not\equiv 0 + x$

Théorie intensionnelle (ITT)

Deux notions d'égalité cohabitent :

- L'égalité propositionnelle '$=$'
  - En Coq, par un type inductif :

```
Inductive eq (A : Type) (x : A) : A -> Type :=
 | refl : eq A x x.
```

  - Dans les PTS, trois constantes : `eq`, `refl`, `eq_rect`.

Théorie intensionnelle (ITT)

Deux notions d'égalité cohabitent :

- L'égalité propositionnelle '$=$'
  - En Coq, par un type inductif :

```
Inductive eq (A : Type) (x : A) : A -> Type :=
 | refl : eq A x x.
```

  - Dans les PTS, trois constantes : `eq`, `refl`, `eq_rect`.

Intuition

$\rightarrow$ La « bonne » notion d'égalité.

Théorie intensionnelle (ITT)

Deux notions d'égalité cohabitent :

- L'égalité propositionnelle '$=$'
  - En Coq, par un type inductif :

```
Inductive eq (A : Type) (x : A) : A -> Type :=
 | refl : eq A x x.
```

  - Dans les PTS, trois constantes : `eq`, `refl`, `eq_rect`.

Intuition

$\rightarrow$ La « bonne » notion d'égalité.

Remarque : $\equiv \subset =$

Théorie extensionnelle (ETT)

Règle de réflexion :

$$\frac{\textsc{Ref} \quad \Gamma \vdash h : M = N}{M \equiv N}$$

Théorie extensionnelle (ETT)

Règle de réflexion :

$$\frac{\textsc{Ref}}{\Gamma \vdash h : M = N}{M \equiv N}$$

Conséquences :

- Extensionalité des fonctions (FunExt) :

$$(\Pi(x : A).\ f\ x = g\ x) \to f = g$$

Théorie extensionnelle (ETT)

Règle de réflexion :

$$\frac{\textsc{Ref}}{\Gamma \vdash h : M = N}{M \equiv N}$$

Conséquences :

- Extensionalité des fonctions (FunExt) :

$$(\Pi(x : A).\ f\ x = g\ x) \to f = g$$

- Unicité des preuves d'identité (UIP) :

$$\Pi(x\ y : A).\ \Pi(h\ h' : x = y).\ h = h'$$

Comparaison

	ITT	ETT
Implémentations	Coq, Agda	NuPRL

Comparaison

	ITT	ETT
Implémentations	Coq, Agda	NuPRL
Type-checking	décidable	indécidable

Comparaison

	ITT	ETT
Implémentations	Coq, Agda	NuPRL
Type-checking	décidable	indécidable
FunExt	non	oui
UIP	non	oui

Comparaison

	ITT	ETT
Implémentations	Coq, Agda	NuPRL
Type-checking	décidable	indécidable
FunExt	non	oui
UIP	non	oui
Programmation avec types dépendants	lourd à utiliser	simple

- Exemple

```
Definition rev A n : List A n -> List A n :=
  | nil => nil
  | cons a n l' => (rev A n l') @ [a]
```

Comparaison

	ITT	ETT
Implémentations	Coq, Agda	NuPRL
Type-checking	décidable	indécidable
FunExt	non	oui
UIP	non	oui
Programmation avec types dépendants	lourd à utiliser	simple
Normalisation forte	oui	non

- Exemple

  $h : \mathsf{Unit} = (\mathsf{Unit} \to \mathsf{Unit}) \vdash (\lambda x.\ x\ x)\ (\lambda x.\ x\ x) : \mathsf{Unit}$

État de l'art

- En Coq (intensionnel)
  - Program/Russell (Matthieu Sozeau, 2006)
  - Bibliothèque *Heq* (Chung-Kil Hur, 2010)

État de l'art

- En Coq (intensionnel)
  - Program/Russell (Matthieu Sozeau, 2006)
  - Bibliothèque *Heq* (Chung-Kil Hur, 2010)
- Extensionnel : Andromeda (Andrej Bauer, 2014)

État de l'art

- En Coq (intensionnel)
  - Program/Russell (Matthieu Sozeau, 2006)
  - Bibliothèque *Heq* (Chung-Kil Hur, 2010)
- Extensionnel : Andromeda (Andrej Bauer, 2014)
- Théorie ni intensionnelle, ni extensionnelle
  - Calculus of Algebraic Constructions (Frédéric Blanqui & Jean-Pierre Jouannaud, 1999)
  - Observational Type Theory (Thorsten Altenkirch & Conor McBride, 2006)
  - Coq Modulo Theory (Pierre-Yves Strub, 2010)

État de l'art

- En Coq (intensionnel)
  - Program/Russell (Matthieu Sozeau, 2006)
  - Bibliothèque *Heq* (Chung-Kil Hur, 2010)
- Extensionnel : Andromeda (Andrej Bauer, 2014)
- Théorie ni intensionnelle, ni extensionnelle
  - Calculus of Algebraic Constructions (Frédéric Blanqui & Jean-Pierre Jouannaud, 1999)
  - Observational Type Theory (Thorsten Altenkirch & Conor McBride, 2006)
  - Coq Modulo Theory (Pierre-Yves Strub, 2010)

Sous-extensionnalité

Idée : enrichir la conversion.

$$\equiv_\beta \subset \equiv_{\beta\mathcal{R}} \subset =$$

Sous-extensionnalité

Idée : enrichir la conversion.

$$\equiv_\beta \subset \equiv_{\beta\mathcal{R}} \subset =$$

$\mathcal{R}$ est un ensemble de règles de réécriture :

- $0 + x \to x$
- $\mathsf{S}(x) + y \to \mathsf{S}(x + y)$
- $(x \otimes y) \otimes z \to x \otimes (y \otimes z)$
- $\ldots$
- $x + 0 \to x$
- $x + \mathsf{S}(y) \to \mathsf{S}(x + y)$
- $x \cup y = y \cup x$

Sous-extensionnalité

Idée : enrichir la conversion.

$$\equiv_\beta \subset \equiv_{\beta\mathcal{R}} \subset =$$

$\mathcal{R}$ est un ensemble de règles de réécriture :

- $0 + x \to x$
- $\mathsf{S}(x) + y \to \mathsf{S}(x + y)$
- $(x \otimes y) \otimes z \to x \otimes (y \otimes z)$
- $\ldots$
- $x + 0 \to x$
- $x + \mathsf{S}(y) \to \mathsf{S}(x + y)$
- $x \cup y = y \cup x$

- $\mathcal{R} = \varnothing$ $\to$ intensionnel
- $\mathcal{R} = \{\text{égalités prouvables}\}$ $\to \approx$ extensionnel

Sous-extensionnalité

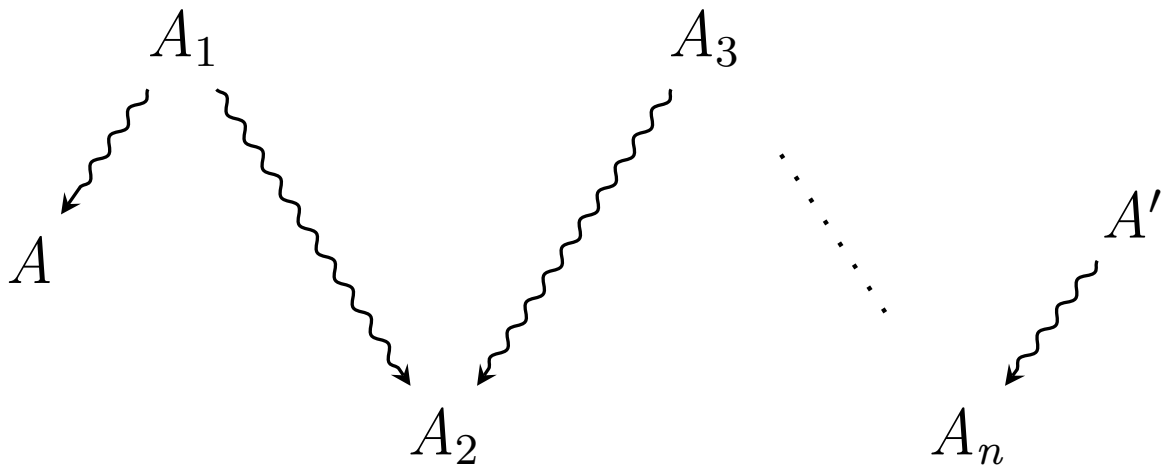
Système $\mathrm{PTS}^{\mathcal{R}}$:

$$\frac{\Gamma \vdash M : A \qquad \Gamma \vdash A' : s \qquad (A \rightsquigarrow^*_{\beta\mathcal{R}} A' \text{ ou } A' \rightsquigarrow^*_{\beta\mathcal{R}} A)}{\Gamma \vdash M : A'} \textsc{Conv}_{\mathcal{R}}$$

Sous-extensionnalité

Système $\mathrm{PTS}^{\mathcal{R}}$:

$$\textsc{Conv}_{\mathcal{R}}\quad \frac{\Gamma \vdash M : A \qquad \Gamma \vdash A' : s \qquad (A \rightsquigarrow^*_{\beta\mathcal{R}} A' \text{ ou } A' \rightsquigarrow^*_{\beta\mathcal{R}} A)}{\Gamma \vdash M : A'}$$



Sous-extensionnalité

Système $\mathrm{PTS}^{\mathcal{R}}$:

$$\textsc{Conv}_{\mathcal{R}}\ \frac{\Gamma \vdash M : A \qquad \Gamma \vdash A' : s \qquad (A \rightsquigarrow^*_{\beta\mathcal{R}} A' \text{ ou } A' \rightsquigarrow^*_{\beta\mathcal{R}} A)}{\Gamma \vdash M : A'}$$

- λ-cube-algébrique (Barbanera, Hernández, Geuvers - 1997)
- CAC (Blanqui, Jouannaud, Okada - 1999)

Sous-extensionnalité

Système $\mathrm{PTS}^{\mathcal{R}}$:

$$\textsc{Conv}_{\mathcal{R}}\quad \frac{\Gamma \vdash M : A \qquad \Gamma \vdash A' : s \qquad (A \rightsquigarrow^*_{\beta\mathcal{R}} A' \text{ ou } A' \rightsquigarrow^*_{\beta\mathcal{R}} A)}{\Gamma \vdash M : A'}$$

- λ-cube-algébrique (Barbanera, Hernández, Geuvers - 1997)
- CAC (Blanqui, Jouannaud, Okada - 1999)

Problème : on perd vite la décidabilité !

Conversion explicite

(Van Doorn, Geuvers, Wiedijk - 2013)

Conversion explicite

(Van Doorn, Geuvers, Wiedijk - 2013)

$$\textsc{Conv}\quad \frac{\Gamma \vdash M : A \qquad \Gamma \vdash A' : s \qquad A \equiv A'}{\Gamma \vdash M : A'}$$

Conversion explicite

(Van Doorn, Geuvers, Wiedijk - 2013)

$$\textsc{Conv}\ \frac{\Gamma \vdash M : A \qquad \Gamma \vdash A' : s \qquad A \equiv A'}{\Gamma \vdash M : A'}$$

Exemple, en Coq :

```
Theorem ex : 3*(17+22)-75 = 42.
Proof.
 reflexivity.
Qed.
```

Conversion explicite

(Van Doorn, Geuvers, Wiedijk - 2013)

$$\textsc{Conv-expl}\quad \frac{\Gamma \vdash M : A \qquad \Gamma \vdash A' : s \qquad \Gamma \vdash H : A \equiv A'}{\Gamma \vdash M^{H} : A'}$$

Conversion explicite

(Van Doorn, Geuvers, Wiedijk - 2013)

$$\frac{\Gamma \vdash M : A \qquad \Gamma \vdash A' : s \qquad \Gamma \vdash H : A \equiv A'}{\Gamma \vdash M^H : A'} \textsc{Conv-expl}$$

$$H ::= \underline{H} \mid \overleftarrow{H} \mid H \cdot H' \mid \langle H, [x : A]H' \rangle \mid \beta(M) \mid \iota(M^H) \mid \ldots$$

Conversion explicite

(Van Doorn, Geuvers, Wiedijk - 2013)

$$\textsc{Conv-expl}\quad \frac{\Gamma \vdash M : A \qquad \Gamma \vdash A' : s \qquad \Gamma \vdash H : A \equiv A'}{\Gamma \vdash M^H : A'}$$

$$H ::= \underline{H} \mid \overleftarrow{H} \mid H \cdot H' \mid \langle H, [x : A]H' \rangle \mid \beta(M) \mid \iota(M^H) \mid \ldots$$

$\textsc{Refl}$ $\textsc{Sym}$ $\textsc{Trans}$ $\textsc{Prod-eq}$ $\textsc{Lam-eq}$ $\textsc{App-eq}$

$\textsc{Beta}$

$$\textsc{Iota}\quad \frac{\Gamma \vdash M : A \qquad \Gamma \vdash A' : s \qquad \Gamma \vdash H : A \equiv A'}{\Gamma \vdash \iota(M^H) : M^H \equiv M}$$

Théorie des types sous-extensionnelle

Système $\mathrm{PTS}^{R}_{\text{Sub-Ext}}$:

- Spécification $(\mathsf{S}, \mathsf{Ax}, \mathsf{Rel})$ d'un PTS
- Ensemble de règles de réécriture $\mathcal{R} = \mathcal{R}_{\mathsf{dec}} \uplus \mathcal{R}_{\mathsf{ndec}}$ tel que $\equiv_{\beta\mathcal{R}_{\mathsf{dec}}}$ soit décidable

Théorie des types sous-extensionnelle

Système $\mathrm{PTS}^{R}_{\textsc{Sub-Ext}}$:

- Spécification $(\mathsf{S}, \mathsf{Ax}, \mathsf{Rel})$ d'un PTS
- Ensemble de règles de réécriture $\mathcal{R} = \mathcal{R}_{\mathsf{dec}} \uplus \mathcal{R}_{\mathsf{ndec}}$ tel que $\equiv_{\beta\mathcal{R}_{\mathsf{dec}}}$ soit décidable

$$\textsc{Conv-dec}\ \frac{\Gamma \vdash M : A \qquad \Gamma \vdash A' : s \qquad (A \rightsquigarrow^{*}_{\beta\mathcal{R}_{\mathsf{dec}}} A' \text{ ou } A' \rightsquigarrow^{*}_{\beta\mathcal{R}_{\mathsf{dec}}} A)}{\Gamma \vdash M : A'}$$

$$\textsc{Conv-ndec}\ \frac{\Gamma \vdash M : A \qquad \Gamma \vdash H : A \equiv A' \bullet s}{\Gamma \vdash M^{H} : A'}$$

Sources d'extensionnalité

Comparons M^H et $\texttt{eq\_rect}\ P\ M\ h$.

$\texttt{eq\_rect}\ P\ M\ h$	M^H
$h : a = b$	$H : A \equiv A'$
$M : P(a)$	$M : A$
$\texttt{eq\_rect}\ P\ M\ h : P(b)$	$M^H : A'$

Sources d'extensionnalité

Comparons M^H et $\texttt{eq\_rect}\ P\ M\ h$.

$\texttt{eq\_rect}\ P\ M\ h$	M^H
$h : a = b$	$H : A \equiv A'$
$M : P(a)$	$M : A$
$\texttt{eq\_rect}\ P\ M\ h : P(b)$	$M^H : A'$

Deux différences :

- Dans $\equiv$, les réécritures passent sous les λ et les Π.
  $\rightarrow$ FunExt$_{\mathcal{R}}$

Sources d'extensionnalité

Comparons M^H et $\texttt{eq\_rect}\ P\ M\ h$.

$\texttt{eq\_rect}\ P\ M\ h$	M^H
$h : a = b$	$H : A \equiv A'$
$M : P(a)$	$M : A$
$\texttt{eq\_rect}\ P\ M\ h : P(b)$	$M^H : A'$

Deux différences :

- Dans $\equiv$, les réécritures passent sous les λ et les Π.
  $\rightarrow$ FunExt$_{\mathcal{R}}$
- Règles de calcul :
  - $M^H \rightsquigarrow M$
  - $\texttt{eq\_rect}\ P\ M\ \texttt{refl} \rightsquigarrow M$

Nouvelle idée

Dans les PTS avec égalité (**sans** $\equiv_{\beta\mathcal{R}}$ et **sans** M^H) :

Autoriser `eq_rect` $P\ M\ h \rightsquigarrow M$ *pour une classe d'égalités plus grande que* $h = \{\texttt{refl}\}$.

Nouvelle idée

Dans les PTS avec égalité (**sans** $\equiv_{\beta\mathcal{R}}$ et **sans** M^H) :

Autoriser `eq_rect` $P\ M\ h \rightsquigarrow M$ *pour une classe d'égalités plus grande que* $h = \{\texttt{refl}\}$.

Par exemple :

- $\mathcal{R}$ ensemble d'égalités
  - prouvables dans le contexte vide
  - vivant dans des types petits
- h obtenue par congruence à partir des égalités de $\mathcal{R}$

Nouvelle idée

Dans les PTS avec égalité (**sans** $\equiv_{\beta\mathcal{R}}$ et **sans** M^H) :

Autoriser `eq_rect` $P\ M\ h \rightsquigarrow M$ *pour une classe d'égalités plus grande que* $h = \{\texttt{refl}\}$.

Par exemple :

- $\mathcal{R}$ ensemble d'égalités
  - prouvables dans le contexte vide
  - vivant dans des types petits
- h obtenue par congruence à partir des égalités de $\mathcal{R}$

$\rightarrow$ Décidabilité ?

$\rightarrow$ Normalisation ?

Résumé du stage

- Définition du système $\mathrm{PTS}_{\textsc{Sub-Ext}}$
  - Combinaison de CAC et de la conversion explicite
  - Modification du travail de Van Doorn et. al.
    $\Gamma \vdash H : M \equiv N \quad \rightarrow \quad \Gamma \vdash H : M \equiv N \bullet A$

Résumé du stage

- Définition du système $\mathrm{PTS}_{\textsc{Sub-Ext}}$
  - Combinaison de CAC et de la conversion explicite
  - Modification du travail de Van Doorn et. al.
    $\Gamma \vdash H : M \equiv N \quad \rightarrow \quad \Gamma \vdash H : M \equiv N \bullet A$
- Équivalences entre $\mathrm{PTS}_{\textsc{Sub-Ext}}$ et systèmes voisins

Résumé du stage

- Définition du système $\mathrm{PTS}_{\textsc{Sub-Ext}}$
  - Combinaison de CAC et de la conversion explicite
  - Modification du travail de Van Doorn et. al.
    $\Gamma \vdash H : M \equiv N \quad \rightarrow \quad \Gamma \vdash H : M \equiv N \bullet A$
- Équivalences entre $\mathrm{PTS}_{\textsc{Sub-Ext}}$ et systèmes voisins
- Méta-théorie
  - Préservation de la cohérence et de la normalisation
  - Notion de modèles abstraits (A. Miquel, thèse)

Conclusion

- Une nouvelle classe de systèmes $\mathrm{PTS}_{\textsc{Sub-Ext}}$.

Conclusion

- Une nouvelle classe de systèmes $\mathrm{PTS}_{\textsc{Sub-Ext}}$.
- Plusieurs questions ouvertes :
  - Comment bien choisir $\mathcal{R}_{\mathsf{dec}}$ et $\mathcal{R}_{\mathsf{ndec}}$?

Conclusion

- Une nouvelle classe de systèmes $\mathrm{PTS}_{\textsc{Sub-Ext}}$.
- Plusieurs questions ouvertes :
  - Comment bien choisir $\mathcal{R}_{\mathsf{dec}}$ et $\mathcal{R}_{\mathsf{ndec}}$?
  - Quelles instances de FunExt et de UIP sont prouvables ?

Conclusion

- Une nouvelle classe de systèmes $\mathrm{PTS}_{\textsc{Sub-Ext}}$.
- Plusieurs questions ouvertes :
  - Comment bien choisir $\mathcal{R}_{\mathsf{dec}}$ et $\mathcal{R}_{\mathsf{ndec}}$?
  - Quelles instances de FunExt et de UIP sont prouvables ?
  - Questions de normalisation.

Conclusion

- Une nouvelle classe de systèmes $\mathrm{PTS}_{\textsc{Sub-Ext}}$.
- Plusieurs questions ouvertes :
  - Comment bien choisir $\mathcal{R}_{\mathsf{dec}}$ et $\mathcal{R}_{\mathsf{ndec}}$?
  - Quelles instances de FunExt et de UIP sont prouvables ?
  - Questions de normalisation.
  - Fournir une implémentation expérimentale.

Conclusion

- Une nouvelle classe de systèmes $\mathrm{PTS}_{\textsc{Sub-Ext}}$.
- Plusieurs questions ouvertes :
  - Comment bien choisir $\mathcal{R}_{\mathsf{dec}}$ et $\mathcal{R}_{\mathsf{ndec}}$?
  - Quelles instances de FunExt et de UIP sont prouvables ?
  - Questions de normalisation.
  - Fournir une implémentation expérimentale.
- Nouvelle idée : règle de calcul de `eq_rect`.