

# UN CALCUL DES CONSTRUCTIONS GRADUEL

OÙ L'ON MET DU TYPAGE DYNAMIQUE DANS UN LANGAGE BIEN STATIQUE

---

*Meven Lennon-Bertrand*, Kenji Maillard, Nicolas Tabareau et Éric Tanter  
Équipe Gallinette, Nantes

5 février 2021 – Journée LHC

## UN PEU DE CONTEXTE

### Typage graduel

- Combine typage statique et dynamique (C#, TypeScript, Python >3.5...)
- Discipliné!

### Typage graduel

- Combine typage statique et dynamique (C#, TypeScript, Python >3.5...)
- Discipliné!

### Ce bon vieux CIC

Cf. Kenji

## Typage graduel

- Combine typage statique et dynamique (C#, TypeScript, Python >3.5...)
- Discipliné!

## Ce bon vieux CIC

Cf. Kenji

## Mais... Pourquoi?

- Les types dépendants c'est DUR :  
    `head : vect N (n + 1) -> N`  
    `head ((filter even v) : vect N ...)`
- Un axiome *avec un comportement calculatoire*
- Peut-on mélanger types graduels et dépendants?

## LE TYPAGE GRADUEL EN ACCÉLÉRÉ

## Mélanger discipline statique et dynamique

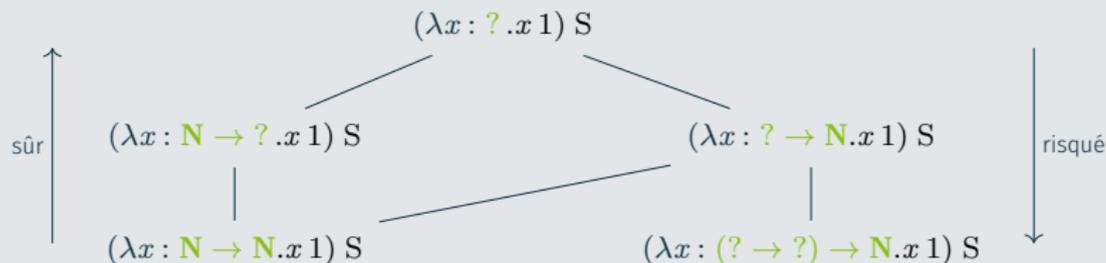
$t : \mathbf{N} \rightarrow \mathbf{N}, u : \mathbf{N} \vdash t u$	type statiquement
$t : \mathbf{N} \rightarrow \mathbf{N}, u : \mathbf{B} \not\vdash t u$	échec statique
$t : \mathbf{N} \rightarrow \mathbf{N}, u : ? \vdash t u$	vérification dynamique

## Mélanger discipline statique et dynamique

$t : \mathbf{N} \rightarrow \mathbf{N}, u : \mathbf{N} \vdash t u$  type statiquement  
 $t : \mathbf{N} \rightarrow \mathbf{N}, u : \mathbf{B} \not\vdash t u$  échec statique  
 $t : \mathbf{N} \rightarrow \mathbf{N}, u : ? \vdash t u$  vérification dynamique

## Propriété fondamentale = Gradualité

Précision  $\sqsubseteq$  + les termes moins précis *typent* plus et *échouent* moins.



## Lambda-calcul simplement typé

$$T := \mathbf{B} \mid \mathbf{N} \mid T \rightarrow T$$

$$t := x \mid \underline{n} \mid \underline{b} \mid \lambda x : T. t \mid t t \mid t + t \mid \text{if } t \text{ then } t \text{ else } t$$

$$\frac{\Gamma, x : T_1 \vdash t : T_2}{\Gamma \vdash \lambda x : T_1. t : T_1 \rightarrow T_2}$$

$$\frac{\Gamma \vdash t_1 : T_1 \quad \Gamma \vdash t_2 : T_2 \quad \Gamma \vdash t_3 : T_3 \quad T_1 = \mathbf{B} \quad T_2 = T \quad T_3 = T}{\Gamma \vdash \text{if } t_1 \text{ then } t_2 \text{ else } t_3 : T}$$

...

Lambda-calcul **graduellement** typé

$$T := \mathbf{B} \mid \mathbf{N} \mid T \rightarrow T \mid ?$$

$$t := x \mid \underline{n} \mid \underline{b} \mid \lambda x : T. t \mid t t \mid t + t \mid \text{if } t \text{ then } t \text{ else } t$$

$$\frac{\Gamma, x : T_1 \vdash t : T_2}{\Gamma \vdash \lambda x : T_1. t : T_1 \rightarrow T_2}$$

$$\frac{\Gamma \vdash t_1 : T_1 \quad \Gamma \vdash t_2 : T_2 \quad \Gamma \vdash t_3 : T_3 \quad T_1 \sim \mathbf{B} \quad T_2 \sim T \quad T_3 \sim T}{\Gamma \vdash \text{if } t_1 \text{ then } t_2 \text{ else } t_3 : T}$$

...

Lambda-calcul **graduellement** typé

$$T := \mathbf{B} \mid \mathbf{N} \mid T \rightarrow T \mid ?$$

$$t := x \mid \underline{n} \mid \underline{b} \mid \lambda x : T.t \mid tt \mid t + t \mid \text{if } t \text{ then } t \text{ else } t$$

$$\frac{\Gamma, x : T_1 \vdash t : T_2}{\Gamma \vdash \lambda x : T_1.t : T_1 \rightarrow T_2}$$

$$\frac{\Gamma \vdash t_1 : T_1 \quad \Gamma \vdash t_2 : T_2 \quad \Gamma \vdash t_3 : T_3 \quad T_1 \sim \mathbf{B} \quad T_2 \sim T \quad T_3 \sim T}{\Gamma \vdash \text{if } t_1 \text{ then } t_2 \text{ else } t_3 : T}$$

...

## Consistence

$$? \sim (\mathbf{N} \rightarrow ?) \quad \mathbf{N} \not\sim \mathbf{B} \quad (\mathbf{N} \rightarrow ?) \sim (? \rightarrow \mathbf{B}) \quad ((? \rightarrow ?) \rightarrow ?) \not\sim (\mathbf{N} \rightarrow ?)$$

Absolument pas transitive!

Subject reduction est cassée...

$$\vdash (\lambda x : ?.x + 1) \text{ true} : \mathbf{N}$$
$$(\lambda x : ?.x + 1) \text{ true} \mapsto \text{true} + 1$$
$$\not\vdash \text{true} + 1$$

Subject reduction est cassée...

$$\begin{aligned} &\vdash (\lambda x : ?.x + 1) \text{ true} : \mathbf{N} \\ &(\lambda x : ?.x + 1) \text{ true} \mapsto \text{true} + 1 \\ &\not\vdash \text{true} + 1 \end{aligned}$$

Et les vérifications dynamiques?

Subject reduction est cassée...

$$\begin{aligned} &\vdash (\lambda x : ?.x + 1) \text{ true} : \mathbf{N} \\ &(\lambda x : ?.x + 1) \text{ true} \mapsto \text{true} + 1 \\ &\not\vdash \text{true} + 1 \end{aligned}$$

Et les vérifications dynamiques?

Calcul de cast

$$\vdash (\lambda x : ?.x + 1) \text{ true} \rightsquigarrow (\lambda x : ?.( \langle \mathbf{N} \Leftarrow ? \rangle x ) + 1) ( \langle ? \Leftarrow \mathbf{B} \rangle \text{true} )$$

## Subject reduction est cassée...

$$\begin{aligned} &\vdash (\lambda x : ?.x + 1) \text{ true} : \mathbf{N} \\ &(\lambda x : ?.x + 1) \text{ true} \mapsto \text{true} + 1 \\ &\not\vdash \text{true} + 1 \end{aligned}$$

Et les vérifications dynamiques?

## Calcul de cast

$$\begin{aligned} &\vdash (\lambda x : ?.x + 1) \text{ true} \rightsquigarrow (\lambda x : ?.( \langle \mathbf{N} \Leftarrow ? \rangle x ) + 1) ( \langle ? \Leftarrow \mathbf{B} \rangle \text{true} ) \\ &(\lambda x : ?.( \langle \mathbf{N} \Leftarrow ? \rangle x ) + 1) ( \langle ? \Leftarrow \mathbf{B} \rangle \text{true} ) \mapsto ( \langle \mathbf{N} \Leftarrow ? \rangle \langle ? \Leftarrow \mathbf{B} \rangle \text{true} ) + 1 \mapsto \\ &\quad ( \langle \mathbf{N} \Leftarrow \mathbf{B} \rangle \text{true} ) + 1 \mapsto \text{raise} \end{aligned}$$

Subject reduction est cassée...

$$\begin{aligned} &\vdash (\lambda x : ?.x + 1) \text{ true} : \mathbf{N} \\ &(\lambda x : ?.x + 1) \text{ true} \mapsto \text{true} + 1 \\ &\not\vdash \text{true} + 1 \end{aligned}$$

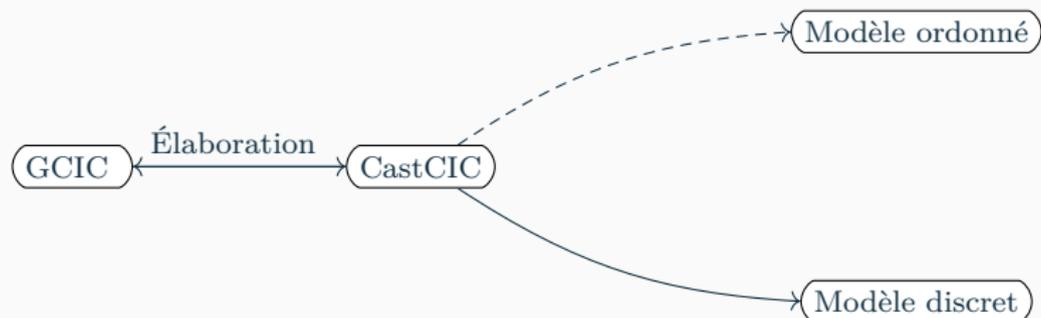
Et les vérifications dynamiques ?

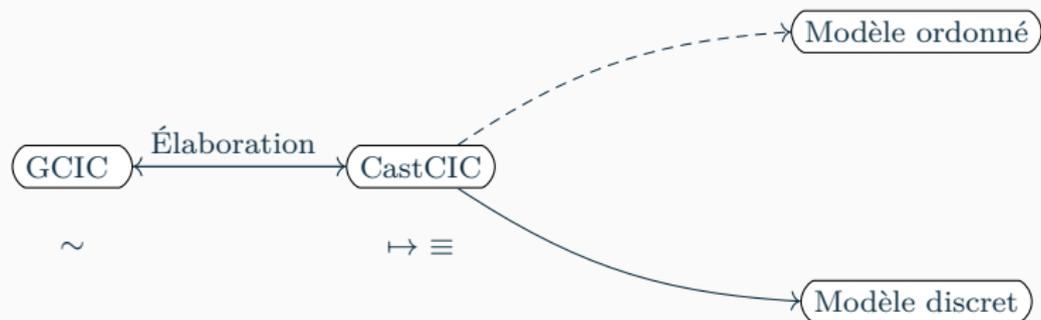
Calcul de cast

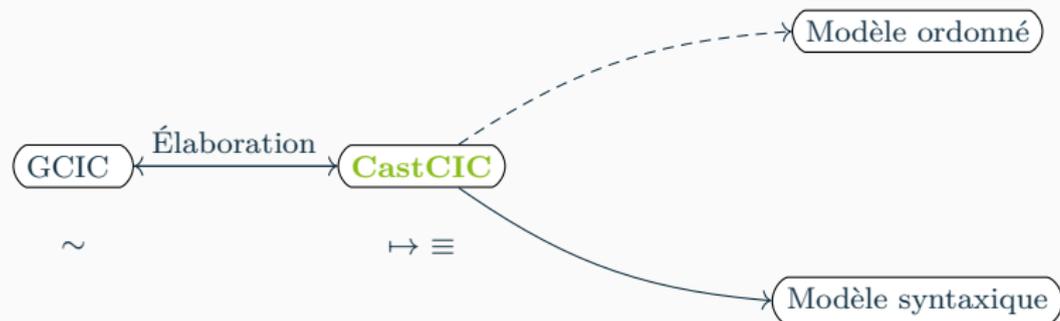
$$\begin{aligned} &\vdash (\lambda x : ?.x + 1) \text{ true} \rightsquigarrow (\lambda x : ?.( \langle \mathbf{N} \Leftarrow ? \rangle x ) + 1) ( \langle ? \Leftarrow \mathbf{B} \rangle \text{true} ) \\ &(\lambda x : ?.( \langle \mathbf{N} \Leftarrow ? \rangle x ) + 1) ( \langle ? \Leftarrow \mathbf{B} \rangle \text{true} ) \mapsto ( \langle \mathbf{N} \Leftarrow ? \rangle \langle ? \Leftarrow \mathbf{B} \rangle \text{true} ) + 1 \mapsto \\ &\quad ( \langle \mathbf{N} \Leftarrow \mathbf{B} \rangle \text{true} ) + 1 \mapsto \text{raise} \end{aligned}$$

Gradualité : si  $A \sqsubseteq B$  alors  $( \langle B \Leftarrow A \rangle, \langle A \Leftarrow B \rangle ) : A \triangleleft B$

ET MAINTENANT, CIC!







## Que demande le peuple?

- Gradualité
- Conservativité par rapport à CIC
- Normalisation

## Que demande le peuple?

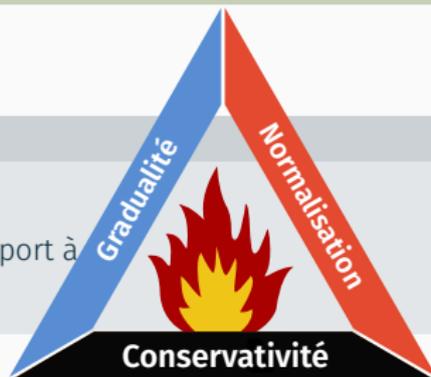
- Gradualité
- Conservativité par rapport à CIC
- Normalisation

## Hélas...

Conservativité + Gradualité  $\Rightarrow ? \rightarrow ? \triangleleft ? \Rightarrow$  Plongement du  $\lambda$ -calcul pur

Que demande le peuple ?

- Gradualité
- Conservativité par rapport à
- Normalisation

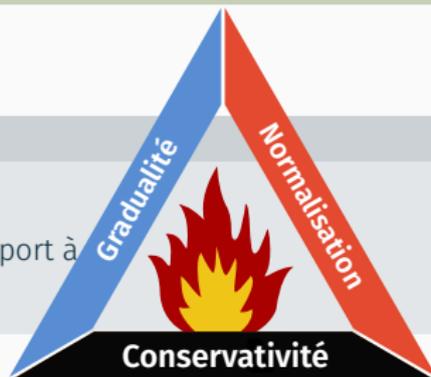


Hélas...

Conservativité + Gradualité  $\Rightarrow ? \rightarrow ? \triangleleft ? \Rightarrow$  Plongement du  $\lambda$ -calcul pur

Que demande le peuple ?

- Gradualité
- Conservativité par rapport à
- Normalisation



Hélas...

Conservativité + Gradualité  $\Rightarrow ? \rightarrow ? \triangleleft ? \Rightarrow$  Plongement du  $\lambda$ -calcul pur

Faites votre choix!

## Typage

$$\dots \quad \frac{\Gamma \vdash T : \square}{\Gamma \vdash ?_T : T} \quad \frac{\Gamma \vdash T : \square}{\Gamma \vdash \text{raise}_T : T} \quad \frac{\Gamma \vdash A : \square \quad \Gamma \vdash B : \square \quad \Gamma \vdash t : A}{\Gamma \vdash \langle B \Leftarrow A \rangle t : B}$$

## Typage

$$\dots \quad \frac{\Gamma \vdash T : \square}{\Gamma \vdash ?_T : T} \quad \frac{\Gamma \vdash T : \square}{\Gamma \vdash \text{raise}_T : T} \quad \frac{\Gamma \vdash A : \square \quad \Gamma \vdash B : \square \quad \Gamma \vdash t : A}{\Gamma \vdash \langle B \Leftarrow A \rangle t : B}$$

## Réduction

- ? et raise comme erreurs :

$$?_{\Pi x:A.B} u \mapsto ?_{B[x:=u]}$$

- erreur de cast :

$$\langle \mathbf{N} \Leftarrow \mathbf{B} \rangle \text{true} \mapsto \text{raise}_{\mathbf{N}}$$

- propagation de cast :

$$\langle A' \rightarrow B' \Leftarrow A \rightarrow B \rangle (\lambda x : A. t) \mapsto \lambda x' : A'. \langle B' \Leftarrow B \rangle (t [x := \langle A \Leftarrow A' \rangle x'])$$

- $?_{\square}$  avec cast comme constructeur/destructeur :

$$\langle X \Leftarrow ?_{\square} \rangle \langle ?_{\square} \Leftarrow \mathbf{N} \rangle n \mapsto \langle X \Leftarrow \mathbf{N} \rangle n$$

## Modèle syntaxique

- guide les règles de réduction
- prouve la normalisation (relative)

## Modèle syntaxique

- guide les règles de réduction
- prouve la normalisation (relative)

## Modèle ordonné

- modèle la précision  $\sqsubseteq$
- prouve  $A \sqsubseteq B \Rightarrow (\langle B \leftarrow A \rangle, \langle A \leftarrow B \rangle) : A \triangleleft B$

## Modèle syntaxique

- guide les règles de réduction
- prouve la normalisation (relative)

## Modèle ordonné

- modèle la précision  $\sqsubseteq$
- prouve  $A \sqsubseteq B \Rightarrow (\langle B \leftarrow A \rangle, \langle A \leftarrow B \rangle) : A \triangleleft B$

## Alors on a gagné?

- pour  $\Gamma \vdash t : T \wedge t \sqsubseteq u \Rightarrow \Gamma \vdash u : U$  il faut des propriétés sur  $\mapsto$
- $\sqsubseteq$  propositionnelle invariante par réduction
- $\sqsubseteq$  syntaxique qui est une simulation pour  $\mapsto$

WHAT NOW?

---

## Ce dont je n'ai pas parlé

- l'élaboration bidirectionnelle, c'est cool
- le type identité, c'est compliqué

## Ce dont je n'ai pas parlé

- l'élaboration bidirectionnelle, c'est cool
- le type identité, c'est compliqué

## À emporter

- le typage dynamique et dépendant, c'est possible
- les choix, c'est inévitable
- les modèles syntaxiques, c'est pratique
- le travail sur la source, c'est irremplaçable

## Ce dont je n'ai pas parlé

- l'élaboration bidirectionnelle, c'est cool
- le type identité, c'est compliqué

## À emporter

- le typage dynamique et dépendant, c'est possible
- les choix, c'est inévitable
- les modèles syntaxiques, c'est pratique
- le travail sur la source, c'est irremplaçable

Et demain, une implémentation expérimentale ?

MERCI!