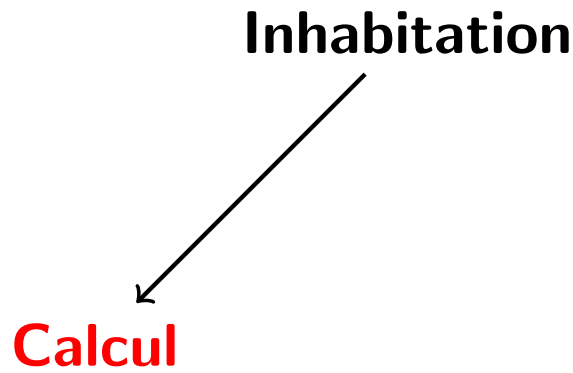
Inhabitation du Call-by-Push-Value

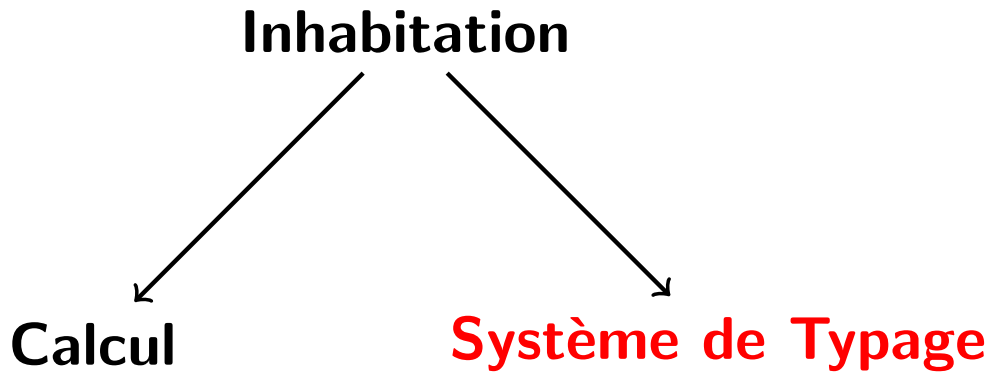
Victor ARRIAL
sous la direction de D.Kesner et G.Guerrieri

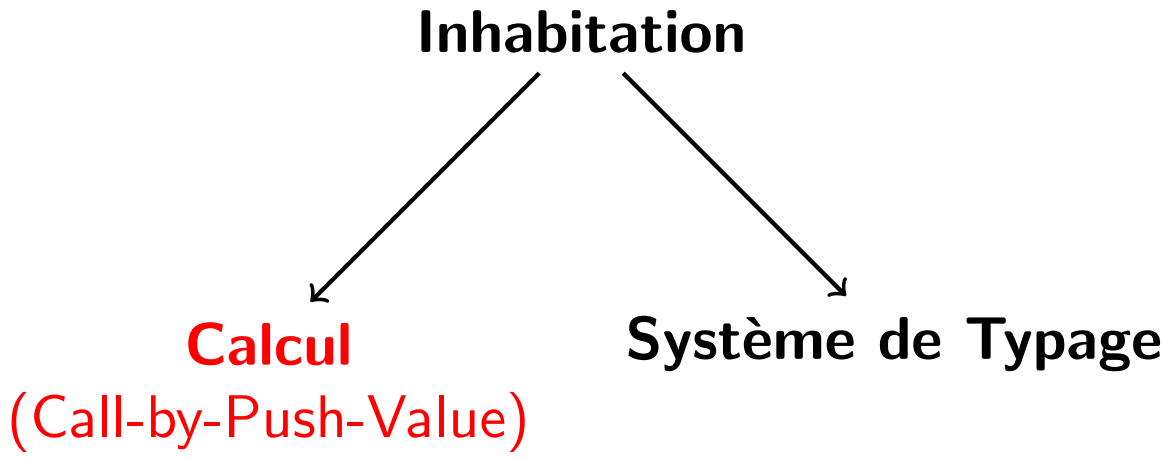
IRIF – Université de Paris

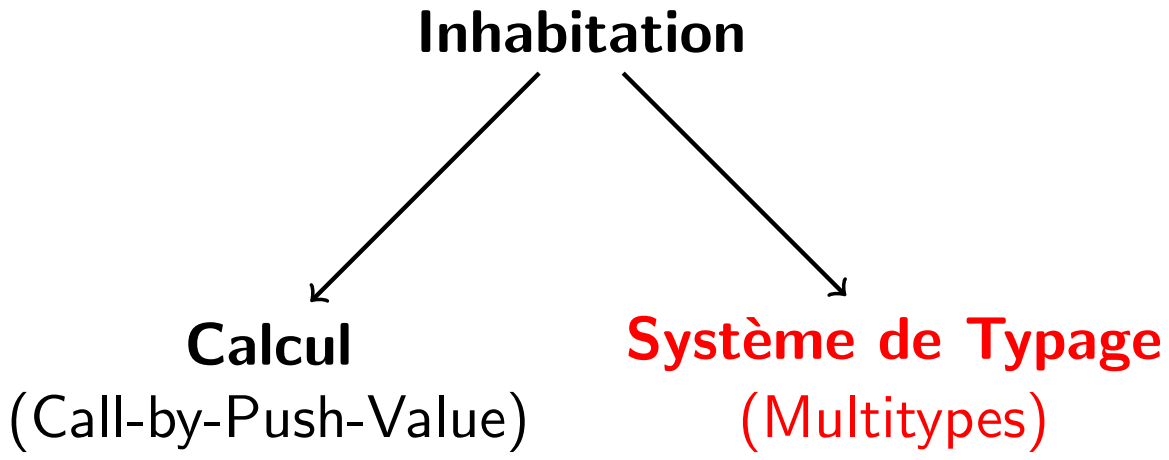
SCALP - 5 Novembre 2021

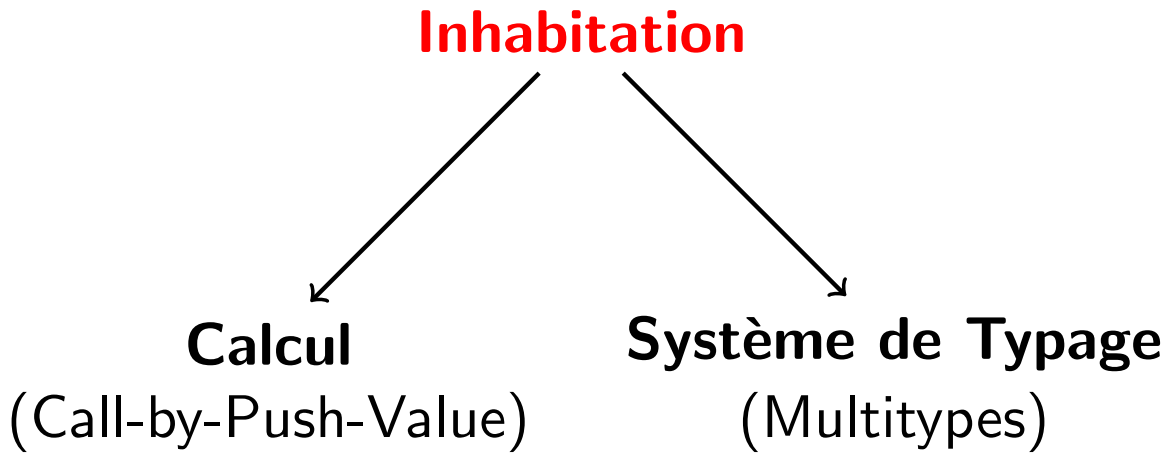
Inhabitation


Inhabitation
Calcul


Inhabitation
Calcul
Système de Typage


Inhabitation
Calcul
(Call-by-Push-Value)
Système de Typage


Inhabitation
Calcul
(Call-by-Push-Value)
Système de Typage
(Multitypes)


Inhabitation
Calcul
(Call-by-Push-Value)
Système de Typage
(Multitypes)

Différentes Stratégies d'Évaluation

Différentes Stratégies d'Évaluation

Appel par Valeur

Différentes Stratégies d'Évaluation

Appel par Valeur

$fst\ (2 * 3,\ 4 + 5)$

Différentes Stratégies d'Évaluation

Appel par Valeur

$$\mathit{fst}\ (2 * 3,\ 4 + 5) \quad \rightarrow \quad \mathit{fst}\ (6,\ 4 + 5)$$

Différentes Stratégies d'Évaluation

Appel par Valeur

$$\mathit{fst}\ (2 * 3,\ 4 + 5) \quad \rightarrow \quad \mathit{fst}\ (6,\ 4 + 5) \quad \rightarrow \quad \mathit{fst}\ (6,\ 9)$$

Différentes Stratégies d'Évaluation

Appel par Valeur

$$\mathit{fst}\ (2 * 3,\ 4 + 5) \quad \rightarrow \quad \mathit{fst}\ (6,\ 4 + 5) \quad \rightarrow \quad \mathit{fst}\ (6,\ 9) \quad \rightarrow \quad 6$$

Différentes Stratégies d'Évaluation

Appel par Valeur

$$\mathit{fst}\ (2 * 3,\ 4 + 5) \quad \rightarrow \quad \mathit{fst}\ (6,\ 4 + 5) \quad \rightarrow \quad \mathit{fst}\ (6,\ 9) \quad \rightarrow \quad 6$$

Différentes Stratégies d'Évaluation

Appel par Valeur

$$\mathit{fst}\ (2 * 3,\ 4 + 5) \quad \rightarrow \quad \mathit{fst}\ (6,\ 4 + 5) \quad \rightarrow \quad \mathit{fst}\ (6,\ 9) \quad \rightarrow \quad 6$$

Appel par Nom

Différentes Stratégies d'Évaluation

Appel par Valeur

$$\mathit{fst}\ (2 * 3,\ 4 + 5) \quad \rightarrow \quad \mathit{fst}\ (6,\ 4 + 5) \quad \rightarrow \quad \mathit{fst}\ (6,\ 9) \quad \rightarrow \quad 6$$

Appel par Nom

$$\mathit{fst}\ (2 * 3,\ 4 + 5)$$

Différentes Stratégies d'Évaluation

Appel par Valeur

$$\mathit{fst}\ (2 * 3,\ 4 + 5) \quad \rightarrow \quad \mathit{fst}\ (6,\ 4 + 5) \quad \rightarrow \quad \mathit{fst}\ (6,\ 9) \quad \rightarrow \quad 6$$

Appel par Nom

$$\mathit{fst}\ (2 * 3,\ 4 + 5) \quad \rightarrow \quad 2 * 3$$

Différentes Stratégies d'Évaluation

Appel par Valeur

$$\mathit{fst}\ (2 * 3,\ 4 + 5) \quad \rightarrow \quad \mathit{fst}\ (6,\ 4 + 5) \quad \rightarrow \quad \mathit{fst}\ (6,\ 9) \quad \rightarrow \quad 6$$

Appel par Nom

$$\mathit{fst}\ (2 * 3,\ 4 + 5) \quad \rightarrow \quad 2 * 3 \quad \rightarrow \quad 6$$

Différentes Stratégies d'Évaluation

Appel par Valeur

$$\mathit{fst}\ (2 * 3,\ 4 + 5) \quad \rightarrow \quad \mathit{fst}\ (6,\ 4 + 5) \quad \rightarrow \quad \mathit{fst}\ (6,\ 9) \quad \rightarrow \quad 6$$

Appel par Nom

$$\mathit{fst}\ (2 * 3,\ 4 + 5) \quad \rightarrow \quad 2 * 3 \quad \rightarrow \quad 6$$

Conséquences :

Différentes Stratégies d'Évaluation

Appel par Valeur

$$\mathit{fst}\ (2 * 3,\ 4 + 5) \quad \rightarrow \quad \mathit{fst}\ (6,\ 4 + 5) \quad \rightarrow \quad \mathit{fst}\ (6,\ 9) \quad \rightarrow \quad 6$$

Appel par Nom

$$\mathit{fst}\ (2 * 3,\ 4 + 5) \quad \rightarrow \quad 2 * 3 \quad \rightarrow \quad 6$$

Conséquences :

Nombre de réductions

Différentes Stratégies d'Évaluation

Appel par Valeur

$$\mathit{fst}\ (2 * 3,\ 4 + 5) \quad \rightarrow \quad \mathit{fst}\ (6,\ 4 + 5) \quad \rightarrow \quad \mathit{fst}\ (6,\ 9) \quad \rightarrow \quad 6$$

Appel par Nom

$$\mathit{fst}\ (2 * 3,\ 4 + 5) \quad \rightarrow \quad 2 * 3 \quad \rightarrow \quad 6$$

Conséquences :

- Nombre de réductions
- Sémantique des programmes

Différentes Stratégies d'Évaluation

Appel par Valeur

$fst\ (2 * 3, \Omega)$

Appel par Nom

$fst\ (2 * 3, \Omega)$

Conséquences :

- Nombre de réductions
- Sémantique des programmes

Différentes Stratégies d'Évaluation

Appel par Valeur

$fst\ (2 * 3, \Omega) \quad \rightarrow \quad fst\ (6, \Omega)$

Appel par Nom

$fst\ (2 * 3, \Omega)$

Conséquences :

- Nombre de réductions
- Sémantique des programmes

Différentes Stratégies d'Évaluation

Appel par Valeur

$fst\ (2 * 3, \Omega) \quad \rightarrow \quad fst\ (6, \Omega)$ 

Appel par Nom

$fst\ (2 * 3, \Omega)$

Conséquences :

- Nombre de réductions
- Sémantique des programmes

Différentes Stratégies d'Évaluation

Appel par Valeur

$fst\ (2 * 3, \Omega) \quad \rightarrow \quad fst\ (6, \Omega)$ ↻

Appel par Nom

$fst\ (2 * 3, \Omega) \quad \rightarrow \quad 2 * 3$

Conséquences :

- Nombre de réductions
- Sémantique des programmes

Différentes Stratégies d'Évaluation

Appel par Valeur

$fst\ (2 * 3, \Omega) \quad \rightarrow \quad fst\ (6, \Omega)$ 

Appel par Nom

$fst\ (2 * 3, \Omega) \quad \rightarrow \quad 2 * 3 \quad \rightarrow \quad 6$

Conséquences :

- Nombre de réductions
- Sémantique des programmes

Différentes Stratégies d'Évaluation

Appel par Valeur

$fst\ (2 * 3, \Omega) \quad \rightarrow \quad fst\ (6, \Omega)$ 

Appel par Nom

$fst\ (2 * 3, \Omega) \quad \rightarrow \quad 2 * 3 \quad \rightarrow \quad 6$

Conséquences :

Nombre de réductions

Sémantique des programmes

Call-by-Push-Value

Call-by-Push-Value

Généralisation des deux stratégies :

Call-by-Push-Value

Généralisation des deux stratégies :

Un **unique** formalisme[1]

[1] P.B. Levy, Call-by-Push-Value : A Subsuming Paradigm, 2003

Call-by-Push-Value

Généralisation des deux stratégies :

- Un **unique** formalisme
- La **syntaxe** contrôle l'exécution

Call-by-Push-Value

Généralisation des deux stratégies :

- Un **unique** formalisme
- La **syntaxe** contrôle l'exécution
- Opérateurs **autorisent**/**bloquent** la réduction

Call-by-Push-Value

Généralisation des deux stratégies :

- Un **unique** formalisme
- La **syntaxe** contrôle l'exécution
- Opérateurs **autorisent**/**bloquent** la réduction
- Permet d'**encoder CBN**/**CBV** et d'autres stratégies[2,3]

[2] J-Y. Girard, Linear Logic, 1987

[3] G. Guerrieri, G. Manzonetto, The Bang Calculus and the Two Girard's Translations, 2019

Le $\lambda!$-calcul : Syntaxe, Réductions et Clashes

$$t, u \ ::= \ x \in \mathcal{V} \mid tu \mid \lambda x.u$$

Le λ!-calcul : Syntaxe, Réductions et Clashes

$$t, u \ ::= \ x \in \mathcal{V} \mid tu \mid \lambda x.u \mid !u \quad \text{Valeur}$$

Le $\lambda!$-calcul : Syntaxe, Réductions et Clashes

$$t, u \ ::= \ x \in \mathcal{V} \mid tu \mid \lambda x.u \mid !u \mid \mathit{der}(u)$$

Le $\lambda!$-calcul : Syntaxe, Réductions et Clashes

$$t, u \ ::= \ x \in \mathcal{V} \mid tu \mid \lambda x.u \mid !u \mid der(u) \mid u[x := v]$$

Le $\lambda!$-calcul : Syntaxe, Réductions et Clashes

$$t, u \ ::= \ x \in \mathcal{V} \mid tu \mid \lambda x.u \mid !u \mid der(u) \mid u[x := v]$$

Réductions :

Le $\lambda!$-calcul : Syntaxe, Réductions et Clashes

$$t, u \ ::= \ x \in \mathcal{V} \mid tu \mid \lambda x.u \mid !u \mid der(u) \mid u[x := v]$$

Réductions :

$$(\lambda x.t)\, u$$

Le $\lambda!$-calcul : Syntaxe, Réductions et Clashes

$$t, u \ ::= \ x \in \mathcal{V} \mid tu \mid \lambda x.u \mid !u \mid der(u) \mid u[x := v]$$

Réductions :

$$(\lambda x.t)\, u \quad \mapsto_{dB} \qquad t[x := u]$$

Le $\lambda!$-calcul : Syntaxe, Réductions et Clashes

$$t, u \ ::= \ x \in \mathcal{V} \mid tu \mid \lambda x.u \mid !u \mid der(u) \mid u[x := v]$$

Réductions :

$$\begin{array}{rcl} (\lambda x.t)\, u & \mapsto_{dB} & t[x := u] \\ t[x := \quad (!u)] & & \end{array}$$

Le $\lambda!$-calcul : Syntaxe, Réductions et Clashes

$$t, u \ ::= \ x \in \mathcal{V} \mid tu \mid \lambda x.u \mid !u \mid der(u) \mid u[x := v]$$

Réductions :

$$\begin{array}{rcl} (\lambda x.t)\, u & \mapsto_{dB} & t[x := u] \\ t[x := \ (!u)] & \mapsto_{s!} & t\{x := u\} \end{array}$$

Le $\lambda!$-calcul : Syntaxe, Réductions et Clashes

$$t, u \ ::= \ x \in \mathcal{V} \mid tu \mid \lambda x.u \mid !u \mid der(u) \mid u[x := v]$$

Réductions :

$$\begin{array}{rcl} (\lambda x.t)\, u & \mapsto_{dB} & t[x := u] \\ t[x := (!u)] & \mapsto_{s!} & t\{x := u\} \\ der(\ !t\ ) & & \end{array}$$

Le $\lambda!$-calcul : Syntaxe, Réductions et Clashes

$$t, u \ ::= \ x \in \mathcal{V} \mid tu \mid \lambda x.u \mid !u \mid der(u) \mid u[x := v]$$

Réductions :

$$\begin{array}{rcl} (\lambda x.t)\, u & \mapsto_{dB} & t[x := u] \\ t[x := (!u)] & \mapsto_{s!} & t\{x := u\} \\ der(\ !t\ ) & \mapsto_{d!} & t \end{array}$$

Le $\lambda!$-calcul : Syntaxe, Réductions et Clashes

$$t, u \ ::= \ x \in \mathcal{V} \mid tu \mid \lambda x.u \mid !u \mid der(u) \mid u[x := v]$$

Réductions :

$$\begin{array}{rcl} (\lambda x.t)\, u & \mapsto_{dB} & t[x := u] \\ t[x := (!u)] & \mapsto_{s!} & t\{x := u\} \\ der(!t) & \mapsto_{d!} & t \end{array}$$

Clashes :

Le $\lambda!$-calcul : Syntaxe, Réductions et Clashes

$$t, u \ ::= \ x \in \mathcal{V} \mid tu \mid \lambda x.u \mid !u \mid der(u) \mid u[x := v]$$

Réductions :

$$\begin{array}{rcl} (\lambda x.t)\, u & \mapsto_{dB} & t[x := u] \\ t[x := (!u)] & \mapsto_{s!} & t\{x := u\} \\ der(!t) & \mapsto_{d!} & t \end{array}$$

Clashes :

$$(!t)\, u \qquad \ldots$$

Le $\lambda!$-calcul : Syntaxe, Réductions et Clashes

$$t, u \ ::= \ x \in \mathcal{V} \mid tu \mid \lambda x.u \mid !u \mid der(u) \mid u[x := v]$$

Réductions :

$$\begin{array}{rcl} (\lambda x.t)\, u & \mapsto_{dB} & t[x := u] \\ t[x := (!u)] & \mapsto_{s!} & t\{x := u\} \\ der(!t) & \mapsto_{d!} & t \end{array}$$

Clashes :

$$(!t)\, u \qquad \ldots$$

Le $\lambda!$-calcul : Syntaxe, Réductions et Clashes

$$t, u \ ::= \ x \in \mathcal{V} \mid tu \mid \lambda x.u \mid !u \mid der(u) \mid u[x := v]$$

Réductions :

$$\begin{array}{rcl} L\,\langle \lambda x.t\rangle\, u & \mapsto_{dB} & L\,\langle t[x := u]\rangle \\ t[x := (!u)] & \mapsto_{s!} & t\{x := u\} \\ der(\ !t\ ) & \mapsto_{d!} & t \end{array}$$

Clashes :

$$(!t)\, u \qquad \ldots$$

Le $\lambda!$-calcul : Syntaxe, Réductions et Clashes

$$t, u \ ::= \ x \in \mathcal{V} \mid tu \mid \lambda x.u \mid !u \mid der(u) \mid u[x := v]$$

Réductions :

$$\begin{array}{rcl} L\,\langle \lambda x.t\rangle\, u & \mapsto_{dB} & L\,\langle t[x := u]\rangle \\ t[x := L\,\langle !u\rangle] & \mapsto_{s!} & L\,\langle t\{x := u\}\rangle \\ der(\quad !t\quad) & \mapsto_{d!} & \quad t \end{array}$$

Clashes :

$$(!t)\, u \qquad \ldots$$

Le $\lambda!$-calcul : Syntaxe, Réductions et Clashes

$$t, u \ ::= \ x \in \mathcal{V} \mid tu \mid \lambda x.u \mid !u \mid der(u) \mid u[x := v]$$

Réductions :

$$\begin{array}{rcl} L\langle \lambda x.t\rangle\, u & \mapsto_{dB} & L\langle t[x := u]\rangle \\ t[x := L\langle !u\rangle] & \mapsto_{s!} & L\langle t\{x := u\}\rangle \\ der(L\langle !t\rangle) & \mapsto_{d!} & L\langle t\rangle \end{array}$$

Clashes :

$$(!t)\, u \qquad \ldots$$

Le $\lambda!$-calcul : Syntaxe, Réductions et Clashes

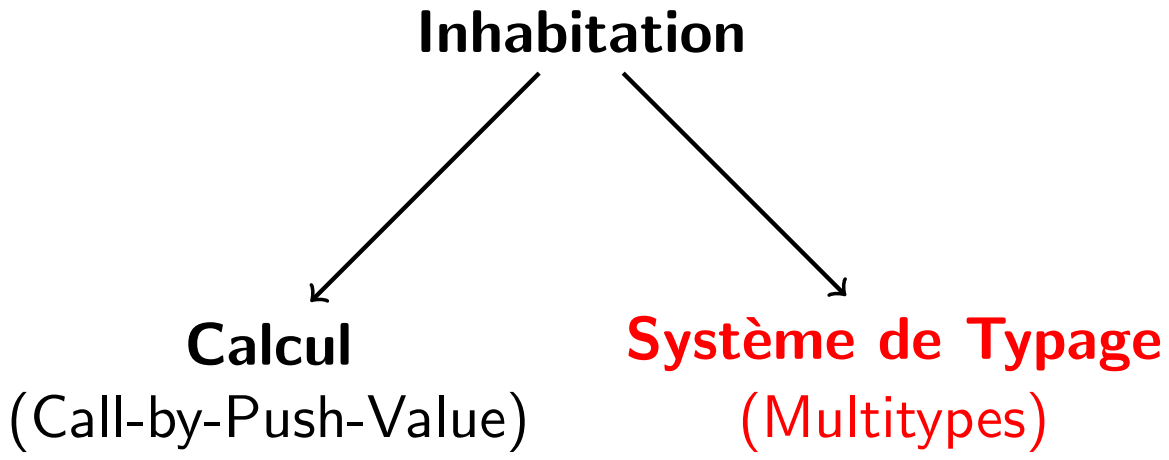
$$t, u \ ::= \ x \in \mathcal{V} \mid tu \mid \lambda x.u \mid !u \mid der(u) \mid u[x := v]$$

Réductions :

$$\begin{array}{rcl} L\langle \lambda x.t\rangle\, u & \mapsto_{dB} & L\langle t[x := u]\rangle \\ t[x := L\langle !u\rangle] & \mapsto_{s!} & L\langle t\{x := u\}\rangle \\ der(L\langle !t\rangle) & \mapsto_{d!} & L\langle t\rangle \end{array}$$

Clashes :

$$L\langle !t\rangle\, u \qquad \ldots$$


Inhabitation
Calcul
(Call-by-Push-Value)
Système de Typage
(Multitypes)

Types : Simples vs. Intersections

Types : Simples vs. Intersections

Motivations générales :

Types : Simples vs. Intersections

Motivations générales :

Limiter les erreurs $fst(4)$

Types : Simples vs. Intersections

Motivations générales :

- Limiter les erreurs $fst(4)$
- Garantir la terminaison

Types : Simples vs. Intersections

Motivations générales :

- Limiter les erreurs — $fst(4)$
- Garantir la terminaison
- Spécification partielle

Types : Simples vs. Intersections

Motivations générales :

- Limiter les erreurs
- Garantir la terminaison
- Spécification partielle

$fst(4)$

Types Simples

$4 : int$

Types : Simples vs. Intersections

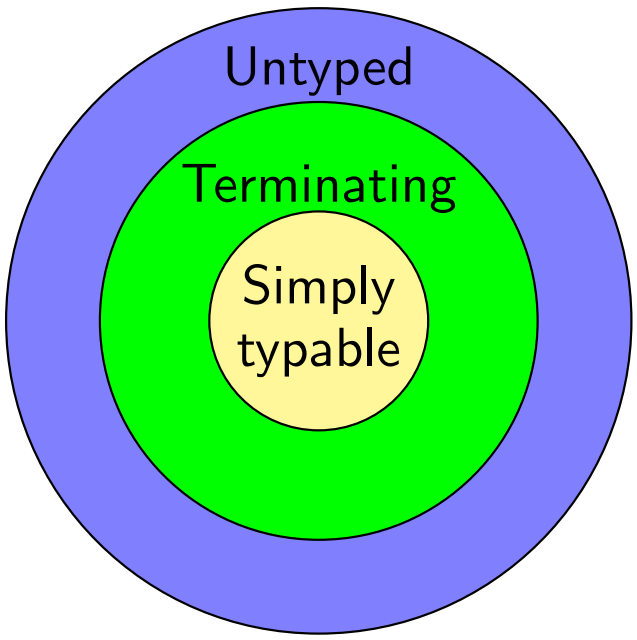
Motivations générales :

- Limiter les erreurs
- Garantir la terminaison
- Spécification partielle

$fst(4)$

Types Simples

$4 : int$



Types : Simples vs. Intersections

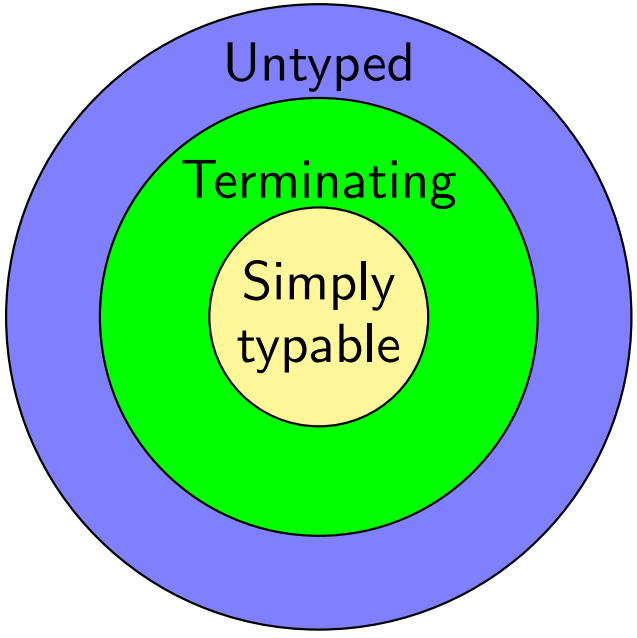
Motivations générales :

- Limiter les erreurs
- Garantir la terminaison
- Spécification partielle

$fst(4)$

Types Simples

$4 : int$



Types Intersections

$4 : int \cap float$

Types : Simples vs. Intersections

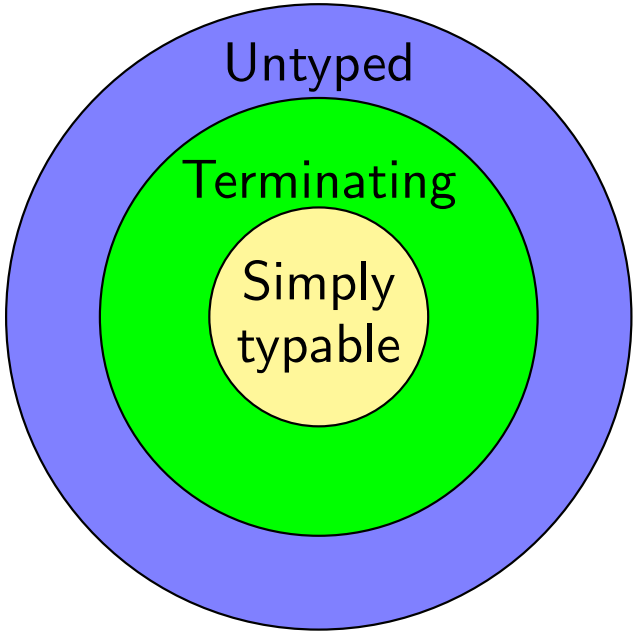
Motivations générales :

- Limiter les erreurs
- Garantir la terminaison
- Spécification partielle

$fst(4)$

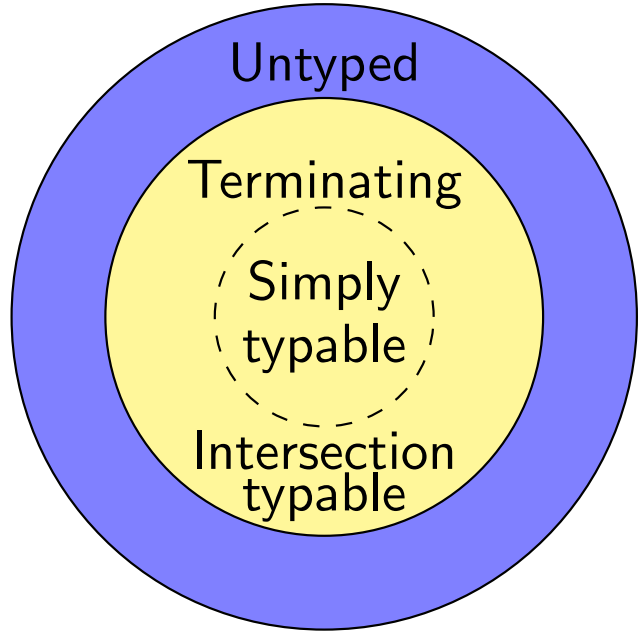
Types Simples

$4 : int$



Types Intersections

$4 : int \cap float$



Types Intersections : Idempotents vs. Non-Idempotents

Types Intersections : Idempotents vs. Non-Idempotents

Associativité : $(A \cap B) \cap C \;=\; A \cap (B \cap C)$

Commutativité : $A \cap B \;=\; B \cap A$

Types Intersections : Idempotents vs. Non-Idempotents

Associativité : $(A \cap B) \cap C = A \cap (B \cap C)$

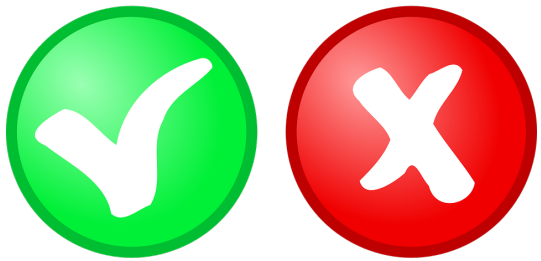
Commutativité : $A \cap B = B \cap A$

Idempotence	
$A \cap A = A$	

Types Intersections : Idempotents vs. Non-Idempotents

Associativité : $(A \cap B) \cap C = A \cap (B \cap C)$

Commutativité : $A \cap B = B \cap A$

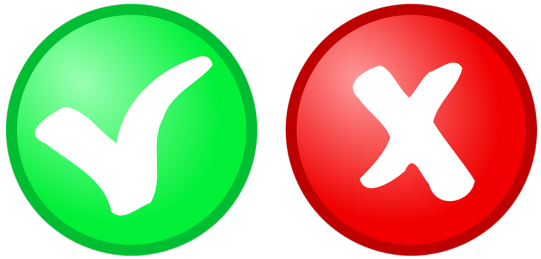
Idempotence	
$A \cap A = A$	
Propriété qualitatives[4] 	

[4] M.Coppo, M. Dezani-Ciancaglini, 1980

Types Intersections : Idempotents vs. Non-Idempotents

Associativité : $(A \cap B) \cap C = A \cap (B \cap C)$

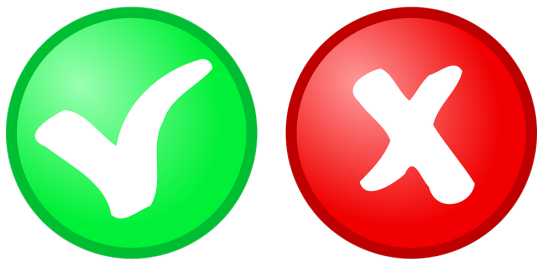
Commutativité : $A \cap B = B \cap A$

Idempotence	**Non-Idempotence** (Parfum Logique Linéaire)
$A \cap A = A$	$A \cap A \neq A$
Propriété qualitatives 	

Types Intersections : Idempotents vs. Non-Idempotents

Associativité : $(A \cap B) \cap C = A \cap (B \cap C)$

Commutativité : $A \cap B = B \cap A$

Idempotence	**Non-Idempotence** (Parfum Logique Linéaire)
$A \cap A = A$	$A \cap A \neq A$
Propriété qualitatives 	Propriétés quantitatives[5] 

[5] D. de Carvalho, Sémantiques de la logique linéaire et temps de calcul, Thèse de doctorat, 2007

Types Intersections Non-Idempotents = Multitypes

Types Intersections Non-Idempotents = Multitypes

Notation : $[\sigma_1 , \cdots , \sigma_n]$ symbolise $\sigma_1 \cap \cdots \cap \sigma_n$

Types Intersections Non-Idempotents = Multitypes

Notation : $[\sigma_1 , \cdots , \sigma_n]$ symbolise $\sigma_1 \cap \cdots \cap \sigma_n$

Séquent : $\Gamma \vdash t : \sigma$

Types Intersections Non-Idempotents = Multitypes

Notation : $[\sigma_1 , \cdots , \sigma_n]$ symbolise $\sigma_1 \cap \cdots \cap \sigma_n$

Séquent : $\Gamma \vdash t : \sigma$

Types Intersections Non-Idempotents = Multitypes

Notation : $[\sigma_1 , \cdots , \sigma_n]$ symbolise $\sigma_1 \cap \cdots \cap \sigma_n$

Séquent : $\Gamma \vdash t : \sigma$

Types Intersections Non-Idempotents = Multitypes

Notation : $[\sigma_1 , \cdots , \sigma_n]$ symbolise $\sigma_1 \cap \cdots \cap \sigma_n$

Séquent : $\Gamma \vdash t : \sigma$

Règles :

$$\frac{\Gamma_1 \vdash t : \mathcal{M} \to \sigma \qquad \Gamma_2 \vdash u : \mathcal{M}}{\Gamma_1 + \Gamma_2 \vdash tu : \sigma}\ (app)$$

Types Intersections Non-Idempotents = Multitypes

Notation : $[\sigma_1 , \cdots , \sigma_n]$ symbolise $\sigma_1 \cap \cdots \cap \sigma_n$

Séquent : $\Gamma \vdash t : \sigma$

Règles :

$$\frac{\Gamma_1 \vdash t : \mathcal{M} \to \sigma \qquad \Gamma_2 \vdash u : \mathcal{M}}{\Gamma_1 + \Gamma_2 \vdash tu : \sigma}\ (app)$$

Types Intersections Non-Idempotents = Multitypes

Notation : $[\sigma_1 , \cdots , \sigma_n]$ symbolise $\sigma_1 \cap \cdots \cap \sigma_n$

Séquent : $\Gamma \vdash t : \sigma$

Règles :

$$\frac{\Gamma_1 \vdash t : \mathcal{M} \to \sigma \qquad \Gamma_2 \vdash u : \mathcal{M}}{\Gamma_1 + \Gamma_2 \vdash tu : \sigma}\ (app)$$

Types Intersections Non-Idempotents = Multitypes

Notation : $[\sigma_1 , \cdots , \sigma_n]$ symbolise $\sigma_1 \cap \cdots \cap \sigma_n$

Séquent : $\Gamma \vdash t : \sigma$

Règles :

$$\frac{\Gamma_1 \vdash t : \mathcal{M} \to \sigma \qquad \Gamma_2 \vdash u : \mathcal{M}}{\Gamma_1 + \Gamma_2 \vdash tu : \sigma}\ (app)$$

Types Intersections Non-Idempotents = Multitypes

Notation : $[\sigma_1 , \cdots , \sigma_n]$ symbolise $\sigma_1 \cap \cdots \cap \sigma_n$

Séquent : $\Gamma \vdash t : \sigma$

Règles :

$$\frac{\Gamma_1 \vdash t : \mathcal{M} \to \sigma \qquad \Gamma_2 \vdash u : \mathcal{M}}{\Gamma_1 + \Gamma_2 \vdash tu : \sigma}\ (app)$$

$$\frac{(\Gamma_i \vdash t : \sigma_i)_{i \in I}}{+_i \Gamma_i \vdash t : [\sigma_i]_{i \in I}}\ (many)$$

Types Intersections Non-Idempotents = Multitypes

Notation : $[\sigma_1 , \cdots , \sigma_n]$ symbolise $\sigma_1 \cap \cdots \cap \sigma_n$

Séquent : $\Gamma \vdash t : \sigma$

Règles :

$$\frac{\Gamma_1 \vdash t : \mathcal{M} \to \sigma \qquad \Gamma_2 \vdash u : \mathcal{M}}{\Gamma_1 + \Gamma_2 \vdash tu : \sigma}\ (app)$$

$$\frac{(\Gamma_i \vdash t : \sigma_i)_{i \in I}}{+_i \Gamma_i \vdash t : [\sigma_i]_{i \in I}}\ (many)$$

Types Intersections Non-Idempotents = Multitypes

Notation : $[\sigma_1 , \cdots , \sigma_n]$ symbolise $\sigma_1 \cap \cdots \cap \sigma_n$

Séquent : $\Gamma \vdash t : \sigma$

Règles :

$$\frac{\Gamma_1 \vdash t : \mathcal{M} \to \sigma \qquad \Gamma_2 \vdash u : \mathcal{M}}{\Gamma_1 + \Gamma_2 \vdash tu : \sigma}\ (app)$$

$$\frac{}{\emptyset \vdash t : [\,]}\ (many)$$

Types Intersections Non-Idempotents = Multitypes

Notation : $[\sigma_1 , \cdots , \sigma_n]$ symbolise $\sigma_1 \cap \cdots \cap \sigma_n$

Séquent : $\Gamma \vdash t : \sigma$

Règles :

$$\frac{\Gamma_1 \vdash t : \mathcal{M} \to \sigma \qquad \Gamma_2 \vdash u : \mathcal{M}}{\Gamma_1 + \Gamma_2 \vdash tu : \sigma}\ (app)$$

$$\frac{(\Gamma_i \vdash t : \sigma_i)_{i \in I}}{+_i \Gamma_i \vdash t : [\sigma_i]_{i \in I}}\ (many)$$

Types Intersections Non-Idempotents = Multitypes

Notation : $[\sigma_1 , \cdots , \sigma_n]$ symbolise $\sigma_1 \cap \cdots \cap \sigma_n$

Séquent : $\Gamma \vdash t : \sigma$

Règles :

$$\frac{\Gamma_1 \vdash t : \mathcal{M} \to \sigma \qquad \Gamma_2 \vdash u : \mathcal{M}}{\Gamma_1 + \Gamma_2 \vdash tu : \sigma}\ (app)$$

$$\frac{(\Gamma_i \vdash t : \sigma_i)_{i \in I}}{+_i \Gamma_i \vdash t : [\sigma_i]_{i \in I}}\ (many)$$

Exemple :

Types Intersections Non-Idempotents = Multitypes

Notation : $[\sigma_1 , \cdots , \sigma_n]$ symbolise $\sigma_1 \cap \cdots \cap \sigma_n$

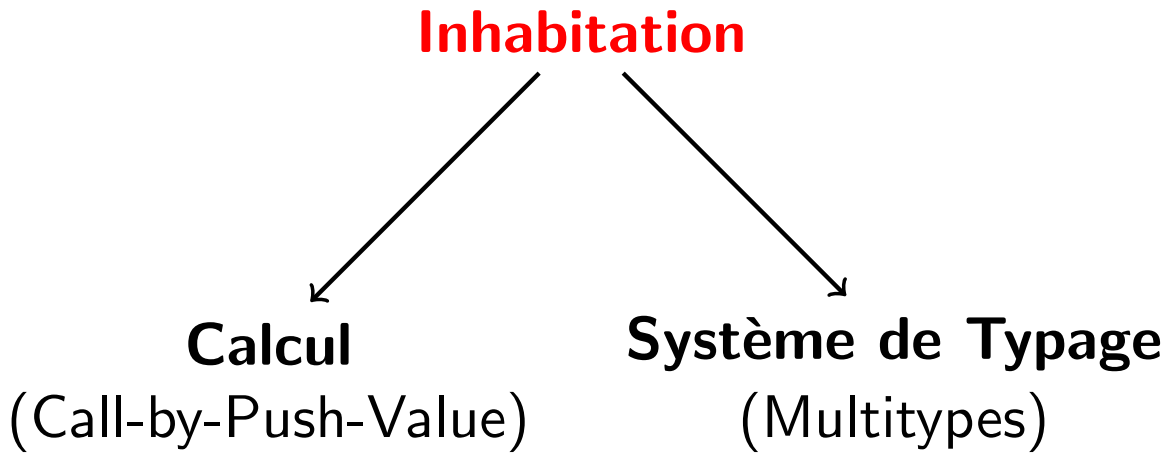
Séquent : $\Gamma \vdash t : \sigma$

Règles :

$$\frac{\Gamma_1 \vdash t : \mathcal{M} \to \sigma \qquad \Gamma_2 \vdash u : \mathcal{M}}{\Gamma_1 + \Gamma_2 \vdash tu : \sigma}\ (app)$$

$$\frac{(\Gamma_i \vdash t : \sigma_i)_{i \in I}}{+_i \Gamma_i \vdash t : [\sigma_i]_{i \in I}}\ (many)$$

Exemple : $\emptyset \vdash \lambda x.xx : [\tau,\ [\tau] \to \sigma] \to \sigma$


Inhabitation
Calcul
(Call-by-Push-Value)
Système de Typage
(Multitypes)

Typing et Inhabitation : Problèmes Duaux

Typing et Inhabitation : Problèmes Duaux

Types : Pose **deux questions** :

Typing et Inhabitation : Problèmes Duaux

Types : Pose **deux questions** :

	Typing $? \vdash t : ?$	
Types Simples		
Types Idempotents		
Types non-Idempotents		

Typing et Inhabitation : Problèmes Duaux

Types : Pose **deux questions** :

	Typing $? \vdash t : ?$	
Types Simples	Décidable	
Types Idempotents		
Types non-Idempotents		

Typing et Inhabitation : Problèmes Duaux

Types : Pose **deux questions** :

	Typing $? \vdash t : ?$	
Types Simples	Décidable	
Types Idempotents	Indécidable	
Types non-Idempotents	Indécidable	

Typing et Inhabitation : Problèmes Duaux

Types : Pose **deux questions** :

	Typing $? \vdash t : ?$	**Inhabitation** $\Gamma \vdash ? : \sigma$
Types Simples	Décidable	
Types Idempotents	Indécidable	
Types non-Idempotents	Indécidable	

Typing et Inhabitation : Problèmes Duaux

Types : Pose **deux questions** :

	Typing $? \vdash t : ?$	**Inhabitation** $\Gamma \vdash ? : \sigma$
Types Simples	Décidable	Décidable
Types Idempotents	Indécidable	
Types non-Idempotents	Indécidable	

Typing et Inhabitation : Problèmes Duaux

Types : Pose **deux questions** :

	Typing $? \vdash t : ?$	**Inhabitation** $\Gamma \vdash ? : \sigma$
Types Simples	Décidable	Décidable
Types Idempotents	Indécidable	Indécidable[6]
Types non-Idempotents	Indécidable	

[6] P. Urzyczyn, The Emptiness Problem for Intersection Types, 1999

Typing et Inhabitation : Problèmes Duaux

Types : Pose **deux questions** :

	Typing $? \vdash t : ?$	**Inhabitation** $\Gamma \vdash ? : \sigma$
Types Simples	Décidable	Décidable
Types Idempotents	Indécidable	Indécidable
Types non-Idempotents	Indécidable	(CBN) Décidable[7]

[7] A. Bucciarelli, D. Kesner, S. Ronchi Della Rocca, Inhabitation for Intersection Types, 2018

Typing et Inhabitation : Problèmes Duaux

Types : Pose **deux questions** :

	Typing $? \vdash t : ?$	**Inhabitation** $\Gamma \vdash ? : \sigma$
Types Simples	Décidable	Décidable
Types Idempotents	Indécidable	Indécidable
Types non-Idempotents	Indécidable	(CBN) Décidable (CBV) ?

L'Objet du Stage

L'inhabitation dans le $\lambda!$-calcul avec le système de typage $\mathcal{U}$

L'Objet du Stage

L'inhabitation dans le $\lambda!$-calcul avec le système de typage $\mathcal{U}$

L'Objet du Stage

L'inhabitation dans le $\lambda!$-calcul avec le système de typage $\mathcal{U}$

L'Objet du Stage

L'inhabitation dans le $\lambda!$-calcul avec le système de typage $\mathcal{U}$

L'Objet du Stage

L'inhabitation dans le $\lambda!$-calcul avec le système de typage $\mathcal{U}$

$$\frac{}{x : [\sigma] \vdash x : \sigma}\,(ax) \qquad \frac{\Gamma_u \vdash u : \mathcal{M} \to \sigma \qquad \Gamma_v \vdash v : \mathcal{M}}{\Gamma_u + \Gamma_v \vdash uv : \sigma}\,(app)$$

$$\frac{\Gamma_u \vdash u : \sigma \qquad \Gamma_v \vdash v : \Gamma_u(x)}{(\Gamma_u \backslash\backslash x) + \Gamma_v \vdash u[x := v] : \sigma}\,(es) \qquad \frac{\Gamma_{t'} \vdash t' : [\sigma]}{\Gamma_{t'} \vdash der(t') : \sigma}\,(dr)$$

$$\frac{\Gamma_{t'} \vdash t' : \tau}{\Gamma_{t'} \backslash\backslash x \vdash \lambda x.t' : \Gamma_{t'}(x) \to \tau}\,(abs) \qquad \frac{(\Gamma_k \vdash t' : \tau_k)_{k \in K}}{+_{k \in K}\Gamma_k \vdash !t' : [\tau_k]_{k \in K}}\,(bg)$$

L'Objet du Stage

L'inhabitation dans le $\lambda!$-calcul avec le système de typage $\mathcal{U}$

$$\frac{}{x : [\sigma] \vdash x : \sigma}\ (ax) \qquad \frac{\Gamma_u \vdash u : \mathcal{M} \to \sigma \qquad \Gamma_v \vdash v : \mathcal{M}}{\Gamma_u + \Gamma_v \vdash uv : \sigma}\ (app)$$

$$\frac{\Gamma_u \vdash u : \sigma \qquad \Gamma_v \vdash v : \Gamma_u(x)}{(\Gamma_u \backslash\backslash x) + \Gamma_v \vdash u[x := v] : \sigma}\ (es) \qquad \frac{\Gamma_{t'} \vdash t' : [\sigma]}{\Gamma_{t'} \vdash der(t') : \sigma}\ (dr)$$

$$\frac{\Gamma_{t'} \vdash t' : \tau}{\Gamma_{t'} \backslash\backslash x \vdash \lambda x.t' : \Gamma_{t'}(x) \to \tau}\ (abs) \qquad \frac{(\Gamma_k \vdash t' : \tau_k)_{k \in K}}{+_{k \in K} \Gamma_k \vdash !t' : [\tau_k]_{k \in K}}\ (bg)$$

L'Objet du Stage

L'inhabitation dans le $\lambda!$-calcul avec le système de typage $\mathcal{U}$

$$\frac{}{x : [\sigma] \vdash x : \sigma}\,(ax) \qquad \frac{\Gamma_u \vdash u : \mathcal{M} \to \sigma \qquad \Gamma_v \vdash v : \mathcal{M}}{\Gamma_u + \Gamma_v \vdash uv : \sigma}\,(app)$$

$$\frac{\Gamma_u \vdash u : \sigma \qquad \Gamma_v \vdash v : \Gamma_u(x)}{(\Gamma_u \backslash\backslash x) + \Gamma_v \vdash u[x := v] : \sigma}\,(es) \qquad \frac{\Gamma_{t'} \vdash t' : [\sigma]}{\Gamma_{t'} \vdash der(t') : \sigma}\,(dr)$$

$$\frac{\Gamma_{t'} \vdash t' : \tau}{\Gamma_{t'} \backslash\backslash x \vdash \lambda x.t' : \Gamma_{t'}(x) \to \tau}\,(abs) \qquad \frac{(\Gamma_k \vdash t' : \tau_k)_{k \in K}}{+_{k \in K} \Gamma_k \vdash\, !t' : [\tau_k]_{k \in K}}\,(bg)$$

L'Objet du Stage

L'inhabitation dans le $\lambda!$-calcul avec le système de typage $\mathcal{U}$

$$\frac{}{x : [\sigma] \vdash x : \sigma}\ (ax) \qquad \frac{\Gamma_u \vdash u : \mathcal{M} \to \sigma \qquad \Gamma_v \vdash v : \mathcal{M}}{\Gamma_u + \Gamma_v \vdash uv : \sigma}\ (app)$$

$$\frac{\Gamma_u \vdash u : \sigma \qquad \Gamma_v \vdash v : \Gamma_u(x)}{(\Gamma_u \backslash\backslash x) + \Gamma_v \vdash u[x := v] : \sigma}\ (es) \qquad \frac{\Gamma_{t'} \vdash t' : [\sigma]}{\Gamma_{t'} \vdash der(t') : \sigma}\ (dr)$$

$$\frac{\Gamma_{t'} \vdash t' : \tau}{\Gamma_{t'} \backslash\backslash x \vdash \lambda x.t' : \Gamma_{t'}(x) \to \tau}\ (abs) \qquad \frac{(\Gamma_k \vdash t' : \tau_k)_{k \in K}}{+_{k \in K} \Gamma_k \vdash !t' : [\tau_k]_{k \in K}}\ (bg)$$

Theorem ([8])

Un terme est $\mathcal{U}$-typable si et seulement s'il se réduit à une forme normale sans clash.

[8] Bucciarelli, Kesner, Rios, Viso, The Bang Calculus Revisited, 2020

Décidabilité de l'Inhabitation : Algorithme

Décidabilité de l'Inhabitation : Algorithme

Problème de typage : $\Gamma \vdash ? : \sigma$

Décidabilité de l'Inhabitation : Algorithme

Problème de typage : $\Gamma \vdash ? : \sigma$

$\rightarrow$ Reconstruire des **dérivations** de typage

Décidabilité de l'Inhabitation : Algorithme

Problème de typage : $\Gamma \vdash ? : \sigma$

$\rightarrow$ Reconstruire des **dérivations** de typage

Propriétés :

Décidabilité de l'Inhabitation : Algorithme

Problème de typage : $\Gamma \vdash ? : \sigma$

$\rightarrow$ Reconstruire des **dérivations** de typage

Propriétés :

- **Correction :**

Les réponses de l'algorithme sont des solutions

Décidabilité de l'Inhabitation : Algorithme

Problème de typage : $\Gamma \vdash ? : \sigma$

$\rightarrow$ Reconstruire des **dérivations** de typage

Propriétés :

- **Correction :**

Les réponses de l'algorithme sont des solutions

- **Complétude :**

Lorsque une solution existe, l'algorithme donne une solution.

Décidabilité de l'Inhabitation : Algorithme

Problème de typage : $\Gamma \vdash ? : \sigma$

$\rightarrow$ Reconstruire des **dérivations** de typage

Propriétés :

- **Correction :**

Les réponses de l'algorithme sont des solutions

- **Complétude :**

Lorsque une solution existe, l'algorithme donne une solution.

- **Terminaison :**

La recherche de solution termine.

Décidabilité de l'Inhabitation : Algorithme

Problème de typage : $\Gamma \vdash ? : \sigma$

$\rightarrow$ Reconstruire **toutes les dérivations** de typage

Propriétés :

- **Correction :**

Les réponses de l'algorithme sont des solutions

- **Complétude :**

Lorsque une solution existe, l'algorithme donne une solution.

- **Terminaison :**

La recherche de solution termine.

Décidabilité de l'Inhabitation : Algorithme

Problème de typage : $\Gamma \vdash ? : \sigma$

$\rightarrow$ Reconstruire **toutes les dérivations** de typage

Propriétés :

- **Correction :**

Les réponses de l'algorithme sont des solutions

- **Complétude :**

Lorsque une solution existe, l'algorithme donne toutes les solutions.

- **Terminaison :**

La recherche de solution termine.

Décidabilité de l'Inhabitation : Algorithme

Problème de typage : $\Gamma \vdash ? : \sigma$

$\rightarrow$ Reconstruire **toutes les dérivations** de typage

Propriétés :

- **Correction :**

Les réponses de l'algorithme sont des solutions

- **Complétude :**

Lorsque une solution existe, l'algorithme donne toutes les solutions.

- **Terminaison :**

La recherche de toutes les solutions termine.

L'algorithme est non-déterministe

Décidabilité de l'Inhabitation : Algorithme

Problème de typage : $\Gamma \vdash ? : \sigma$

$\rightarrow$ Reconstruire **toutes les dérivations** de typage

Règle de l'algorithme :

Décidabilité de l'Inhabitation : Algorithme

Problème de typage : $\Gamma \vdash ? : \sigma$

$\rightarrow$ Reconstruire **toutes les dérivations** de typage

Règle de l'algorithme :

Décidabilité de l'Inhabitation : Algorithme

Problème de typage : $\Gamma \vdash ? : \sigma$

$\rightarrow$ Reconstruire **toutes les dérivations** de typage

Règle de l'algorithme :

$$\frac{}{\mathsf{inh}(\ ,\ )}$$

Décidabilité de l'Inhabitation : Algorithme

Problème de typage : $\Gamma \vdash ? : \sigma$

$\rightarrow$ Reconstruire **toutes les dérivations** de typage

Règle de l'algorithme :

$$\frac{}{\mathsf{inh}(\Gamma, \sigma)}$$

Décidabilité de l'Inhabitation : Algorithme

Problème de typage : $\Gamma \vdash ? : \sigma$

$\rightarrow$ Reconstruire **toutes les dérivations** de typage

Règle de l'algorithme :

$$\frac{}{\mathsf{inh}(\Gamma, \sigma)}\ (\mathsf{App})$$

Décidabilité de l'Inhabitation : Algorithme

Problème de typage : $\Gamma \vdash ? : \sigma$

$\rightarrow$ Reconstruire **toutes les dérivations** de typage

Règle de l'algorithme :

$$\frac{}{\mathsf{inh}(\Gamma, \sigma)}\ (\mathsf{App})$$

Règle de typage :

$$\frac{\Gamma_u \vdash u : \mathcal{M} \to \sigma \qquad \Gamma_v \vdash v : \mathcal{M}}{\Gamma_u + \Gamma_v \vdash uv : \sigma}\ (app)$$

Décidabilité de l'Inhabitation : Algorithme

Problème de typage : $\Gamma \vdash ? : \sigma$

$\rightarrow$ Reconstruire **toutes les dérivations** de typage

Règle de l'algorithme :

$$\frac{}{\mathsf{inh}(\Gamma, \sigma)}\ (\mathrm{App})$$

Règle de typage :

$$\frac{\Gamma_u \vdash u : \mathcal{M} \to \sigma \qquad \Gamma_v \vdash v : \mathcal{M}}{\Gamma_u + \Gamma_v \vdash uv : \sigma}\ (app)$$

Décidabilité de l'Inhabitation : Algorithme

Problème de typage : $\Gamma \vdash ? : \sigma$

$\rightarrow$ Reconstruire **toutes les dérivations** de typage

Règle de l'algorithme :

$$\frac{}{\mathsf{inh}(\Gamma, \sigma)}\ (\mathrm{App})$$

Règle de typage :

$$\frac{\Gamma_u \vdash u : \mathcal{M} \to \sigma \qquad \Gamma_v \vdash v : \mathcal{M}}{\Gamma_u + \Gamma_v \vdash uv : \sigma}\ (app)$$

Décidabilité de l'Inhabitation : Algorithme

Problème de typage : $\Gamma \vdash ? : \sigma$

$\rightarrow$ Reconstruire **toutes les dérivations** de typage

Règle de l'algorithme :

$$\frac{\Vdash \mathsf{inh}(\Gamma_u, \mathcal{M} \to \sigma) \qquad \Vdash \mathsf{inh}(\Gamma_v, \mathcal{M})}{\mathsf{inh}(\Gamma, \sigma)}\ (\mathsf{App})$$

Règle de typage :

$$\frac{\Gamma_u \vdash u : \mathcal{M} \to \sigma \qquad \Gamma_v \vdash v : \mathcal{M}}{\Gamma_u + \Gamma_v \vdash uv : \sigma}\ (app)$$

Décidabilité de l'Inhabitation : Algorithme

Problème de typage : $\Gamma \vdash ? : \sigma$

$\rightarrow$ Reconstruire **toutes les dérivations** de typage

Règle de l'algorithme :

$$\frac{u \Vdash \mathsf{inh}(\Gamma_u, \mathcal{M} \to \sigma) \qquad v \Vdash \mathsf{inh}(\Gamma_v, \mathcal{M})}{\mathsf{inh}(\Gamma, \sigma)}\ (\mathsf{App})$$

Règle de typage :

$$\frac{\Gamma_u \vdash u : \mathcal{M} \to \sigma \qquad \Gamma_v \vdash v : \mathcal{M}}{\Gamma_u + \Gamma_v \vdash uv : \sigma}\ (app)$$

Décidabilité de l'Inhabitation : Algorithme

Problème de typage : $\Gamma \vdash ? : \sigma$

$\rightarrow$ Reconstruire **toutes les dérivations** de typage

Règle de l'algorithme :

$$\frac{u \Vdash \mathsf{inh}(\Gamma_u, \mathcal{M} \to \sigma) \qquad v \Vdash \mathsf{inh}(\Gamma_v, \mathcal{M})}{uv \Vdash \mathsf{inh}(\Gamma, \sigma)}\ (\mathsf{App})$$

Règle de typage :

$$\frac{\Gamma_u \vdash u : \mathcal{M} \to \sigma \qquad \Gamma_v \vdash v : \mathcal{M}}{\Gamma_u + \Gamma_v \vdash uv : \sigma}\ (app)$$

Décidabilité de l'Inhabitation : Algorithme

Problème de typage : $\Gamma \vdash ? : \sigma$

$\rightarrow$ Reconstruire **toutes les dérivations** de typage

Règle de l'algorithme :

$$\frac{\Gamma = \Gamma_u + \Gamma_v \quad \Big| \quad u \Vdash \mathsf{inh}(\Gamma_u, \mathcal{M} \to \sigma) \qquad v \Vdash \mathsf{inh}(\Gamma_v, \mathcal{M})}{uv \Vdash \mathsf{inh}(\Gamma, \sigma)}\ (\mathsf{App})$$

Règle de typage :

$$\frac{\Gamma_u \vdash u : \mathcal{M} \to \sigma \qquad \Gamma_v \vdash v : \mathcal{M}}{\Gamma_u + \Gamma_v \vdash uv : \sigma}\ (app)$$

Décidabilité de l'Inhabitation : Algorithme

Problème de typage : $\Gamma \vdash ? : \sigma$

$\rightarrow$ Reconstruire **toutes les dérivations** de typage

Règle de l'algorithme :

$$\frac{\Gamma = \Gamma_u + \Gamma_v \quad \Big| \quad u \Vdash \mathsf{inh}(\Gamma_u, \mathcal{M} \to \sigma) \qquad v \Vdash \mathsf{inh}(\Gamma_v, \mathcal{M})}{uv \Vdash \mathsf{inh}(\Gamma, \sigma)}\ (\mathsf{App})$$

Règle de typage :

$$\frac{\Gamma_u \vdash u : \mathcal{M} \to \sigma \qquad \Gamma_v \vdash v : \mathcal{M}}{\Gamma_u + \Gamma_v \vdash uv : \sigma}\ (app)$$

Décidabilité de l'Inhabitation : Algorithme

Problème de typage : $\Gamma \vdash ? : \sigma$

$\rightarrow$ Reconstruire **toutes les dérivations** de typage

Règle de l'algorithme :

$$\frac{\Gamma = \Gamma_u + \Gamma_v \text{ Trouver } \mathcal{M} \quad \Big| \quad u \Vdash \mathsf{inh}(\Gamma_u, \mathcal{M} \to \sigma) \qquad v \Vdash \mathsf{inh}(\Gamma_v, \mathcal{M})}{uv \Vdash \mathsf{inh}(\Gamma, \sigma)} \text{(App)}$$

Règle de typage :

$$\frac{\Gamma_u \vdash u : \mathcal{M} \to \sigma \qquad \Gamma_v \vdash v : \mathcal{M}}{\Gamma_u + \Gamma_v \vdash uv : \sigma} (app)$$

Problème de Finitude : Formes Normales

Theorem (Réduction du Sujet[9])

Si $\Gamma \vdash t : \sigma$ *et* t *se réduit en* u *alors* $\Gamma \vdash u : \sigma$.

[9] Bucciarelli, Kesner, Rios, Viso, The Bang Calculus Revisited, 2020

Problème de Finitude : Formes Normales

Theorem (Réduction du Sujet[9])

Si $\Gamma \vdash t : \sigma$ *et* t *se réduit en* u *alors* $\Gamma \vdash u : \sigma$.

[9] Bucciarelli, Kesner, Rios, Viso, The Bang Calculus Revisited, 2020

Problème de Finitude : Formes Normales

Theorem (Réduction du Sujet)

Si $\Gamma \vdash t : \sigma$ *et* t *se réduit en* u *alors* $\Gamma \vdash u : \sigma$.

Famille **infinie** de solutions :

$$u,$$

Problème de Finitude : Formes Normales

Theorem (Réduction du Sujet)

Si $\Gamma \vdash t : \sigma$ *et* t *se réduit en* u *alors* $\Gamma \vdash u : \sigma$.

Famille **infinie** de solutions :

$$u, \qquad der(!u)$$

Problème de Finitude : Formes Normales

Theorem (Réduction du Sujet)

Si $\Gamma \vdash t : \sigma$ *et* t *se réduit en* u *alors* $\Gamma \vdash u : \sigma$.

Famille **infinie** de solutions :

$$u, \qquad der(!u), \qquad der(!der(!u)), \qquad \ldots$$

Problème de Finitude : Formes Normales

Theorem (Réduction du Sujet)

Si $\Gamma \vdash t : \sigma$ *et* t *se réduit en* u *alors* $\Gamma \vdash u : \sigma$.

Famille **infinie** de solutions :

$$u, \qquad der(!u), \qquad der(!der(!u)), \qquad \ldots$$

Terminaison

Problème de Finitude : Formes Normales

Theorem (Réduction du Sujet)

Si $\Gamma \vdash t : \sigma$ *et* t *se réduit en* u *alors* $\Gamma \vdash u : \sigma$.

Famille **infinie** de solutions :

$$u, \qquad der(!u), \qquad der(!der(!u)), \qquad ...$$

Terminaison

Solution : Limiter à une "base" des solutions

Problème de Finitude : Formes Normales

Theorem (Réduction du Sujet)

Si $\Gamma \vdash t : \sigma$ *et* t *se réduit en* u *alors* $\Gamma \vdash u : \sigma$.

Famille **infinie** de solutions :

$$u, \qquad der(!u), \qquad der(!der(!u)), \qquad ...$$

Terminaison

Solution : Limiter à une "base" des solutions

$\rightarrow$ Restreindre aux **formes normales**

Problème de Finitude : Formes Normales

Theorem (Réduction du Sujet)

Si $\Gamma \vdash t : \sigma$ *et* t *se réduit en* u *alors* $\Gamma \vdash u : \sigma$.

Famille **infinie** de solutions :

$$u, \qquad der(!u), \qquad der(!der(!u)), \qquad \ldots$$

Terminaison

Solution : Limiter à une "base" des solutions

$\rightarrow$ Restreindre aux **formes normales**

Theorem (Characterisation)

Un terme est $\mathcal{U}$*-typable si et seulement s'il se réduit à une forme normale sans clash.*

Problème de Finitude : Approximants

Problème de Finitude : Approximants

$$\frac{(\Pi_k \triangleright \Gamma_k \vdash t' : \tau_k)_{k \in K}}{+_{k \in K} \Gamma_k \vdash !t' : [\tau_k]_{k \in K}} \ (bg)$$

Problème de Finitude : Approximants

$$\frac{(\Pi_k \triangleright \Gamma_k \vdash t' : \tau_k)_{k\in K}}{+_{k\in K}\Gamma_k \vdash !t' : [\tau_k]_{k\in K}}\ (bg) \qquad \frac{}{\emptyset \vdash !t : [\,]}\ (bg)$$

Problème de Finitude : Approximants

$$\frac{(\Pi_k \rhd \Gamma_k \vdash t' : \tau_k)_{k\in K}}{+_{k\in K}\Gamma_k \vdash !t' : [\tau_k]_{k\in K}}\ (bg) \qquad \frac{}{\emptyset \vdash !t : []}\ (bg)$$

Famille **infinie** de solutions :

$$x(!t)$$

Problème de Finitude : Approximants

$$\frac{(\Pi_k \rhd \Gamma_k \vdash t' : \tau_k)_{k\in K}}{+_{k\in K}\Gamma_k \vdash !t' : [\tau_k]_{k\in K}}\ (bg) \qquad \frac{}{\emptyset \vdash !x : [\,]}\ (bg)$$

Famille **infinie** de solutions :

$$x(!t), \qquad x(!x)$$

Problème de Finitude : Approximants

$$\frac{(\Pi_k \rhd \Gamma_k \vdash t' : \tau_k)_{k\in K}}{+_{k\in K}\Gamma_k \vdash !t' : [\tau_k]_{k\in K}}\ (bg) \qquad\qquad \frac{}{\emptyset \vdash !x(!x) : []}\ (bg)$$

Famille **infinie** de solutions :

$$x(!t), \qquad x(!x), \qquad x(!(x(!x))), \qquad \ldots$$

Problème de Finitude : Approximants

$$\frac{(\Pi_k \rhd \Gamma_k \vdash t' : \tau_k)_{k\in K}}{+_{k\in K}\Gamma_k \vdash !t' : [\tau_k]_{k\in K}}\ (bg) \qquad\qquad \frac{}{\emptyset \vdash !t : [\,]}\ (bg)$$

Famille **infinie** de solutions :

$$x(!t), \qquad x(!x), \qquad x(!(x(!x))), \qquad \ldots$$

Terminaison

Problème de Finitude : Approximants

$$\frac{(\Pi_k \triangleright \Gamma_k \vdash t' : \tau_k)_{k \in K}}{+_{k \in K} \Gamma_k \vdash !t' : [\tau_k]_{k \in K}}\ (bg) \qquad \frac{}{\emptyset \vdash !\bot : []}\ (bg)$$

Famille **infinie** de solutions :

$$x(!t), \qquad x(!x), \qquad x(!(x(!x))), \qquad \ldots$$

Terminaison

Solution : Introduire une représentation canonique $\bot$

Problème de Finitude : Approximants

$$\frac{(\Pi_k \triangleright \Gamma_k \vdash t' : \tau_k)_{k\in K}}{+_{k\in K}\Gamma_k \vdash !t' : [\tau_k]_{k\in K}}\ (bg) \qquad\qquad \frac{}{\emptyset \vdash !\bot : []}\ (bg)$$

Famille **infinie** de solutions :

$$x(!t), \qquad x(!x), \qquad x(!(x(!x))), \qquad \ldots$$

Terminaison

Solution : Introduire une représentation canonique $\bot$

$\rightarrow$ Approximants

Exploration Infinie des Dérivations Potentielles

Exploration Infinie des Dérivations Potentielles

$$\frac{\Gamma_u \vdash u : \mathcal{M} \to \sigma \qquad \Gamma_v \vdash v : \mathcal{M}}{\Gamma_u + \Gamma_v \vdash uv : \sigma}\ (app)$$

Exploration Infinie des Dérivations Potentielles

$$\frac{\Gamma_u \vdash u : \mathcal{M} \to \sigma \qquad \Gamma_v \vdash v : \mathcal{M}}{\Gamma_u + \Gamma_v \vdash uv : \sigma}\ (app)$$

Exploration Infinie des Dérivations Potentielles

$$\frac{\Gamma_u \vdash u : \mathcal{M} \to \sigma \qquad \Gamma_v \vdash v : \mathcal{M}}{\Gamma_u + \Gamma_v \vdash uv : \sigma}\ (app)$$

Theorem

Dans la dérivation d'une forme normale, le type de la fonction apparait dans un des types du context.

Exploration Infinie des Dérivations Potentielles

$$\frac{\Gamma_u \vdash u : \mathcal{M} \to \sigma \qquad \Gamma_v \vdash v : \mathcal{M}}{\Gamma_u + \Gamma_v \vdash uv : \sigma}\ (app)$$

Theorem

Dans la dérivation d'une forme normale, le type de la fonction apparait dans un des types du context.

Solution : Rechercher $\mathcal{M}$ dans les types du context Γ_u.

Exploration Infinie des Dérivations Potentielles

$$\frac{\Gamma_u \vdash u : \mathcal{M} \to \sigma \qquad \Gamma_v \vdash v : \mathcal{M}}{\Gamma_u + \Gamma_v \vdash uv : \sigma}\ (app)$$

Theorem

Dans la dérivation d'une forme normale, le type de la fonction apparait dans un des types du context.

Solution : Rechercher $\mathcal{M}$ dans les types du context Γ_u.

$\rightarrow$ Un algorithme de recherche de type à n'importe quel niveau

Décidabilité de l'Inhabitation : Algorithme

Problème de typage : $\Gamma \vdash ? : \sigma$

$\longrightarrow$ Reconstruire **toutes les dérivations** de typage

Propriétés :

- **Correction :**

Les réponses de l'algorithme sont des solutions

- **Complétude :**

Lorsque une solution existe, l'algorithme donne toutes les solutions.

- **Terminaison :**

La recherche de toutes les solutions termine.

Décidabilité de l'Inhabitation : Algorithme

Problème de typage : $\Gamma \vdash ? : \sigma$

$\longrightarrow$ Reconstruire **toutes les dérivations** de typage

Propriétés :

- **Correction :**

Les réponses de l'algorithme sont des solutions

- **Complétude :** (Formes Normales et Approximants)

Lorsque une solution existe, l'algorithme donne toutes les solutions.

- **Terminaison :**

La recherche de toutes les solutions termine.

Décidabilité de l'Inhabitation : Algorithme

Problème de typage : $\Gamma \vdash ? : \sigma$

$\longrightarrow$ Reconstruire **toutes les dérivations** de typage

Propriétés :

- **Correction :**

Les réponses de l'algorithme sont des solutions

- **Complétude :** (Formes Normales et Approximants)

Lorsque une solution existe, l'algorithme donne toutes les solutions.

- **Terminaison :** (Recherche de types)

La recherche de toutes les solutions termine.

Conclusion

Ce que nous avons fait :

Conclusion

Ce que nous avons fait :

- **Algorithme** donne une base des solutions

Conclusion

Ce que nous avons fait :

- **Algorithme** donne une base des solutions

Conclusion

Ce que nous avons fait :

- **Algorithme** donne une base des solutions

Correction Complétude Terminaison

Conclusion

Ce que nous avons fait :

- **Algorithme** donne une base des solutions

Correction Complétude Terminaison

Décidabilité de l'**Inhabitation** pour le $\lambda!$-calculus avec system $\mathcal{U}$

Conclusion

Ce que nous avons fait :

- **Algorithme** donne une base des solutions

Correction Complétude Terminaison

Décidabilité de l'**Inhabitation** pour le $\lambda!$-calculus avec system $\mathcal{U}$

Ce qu'il reste à faire :

- Formaliser des **restrictions** (CBN/CBV) (Almost finished)

Conclusion

Ce que nous avons fait :

- **Algorithme** donne une base des solutions

Correction Complétude Terminaison

Décidabilité de l'**Inhabitation** pour le $\lambda!$-calculus avec system $\mathcal{U}$

Ce qu'il reste à faire :

- Formaliser des **restrictions** (CBN/CBV) (Almost finished)
- Réaliser une **implémentation**

Conclusion

Ce que nous avons fait :

- **Algorithme** donne une base des solutions

Correction Complétude Terminaison

Décidabilité de l'**Inhabitation** pour le $\lambda!$-calculus avec system $\mathcal{U}$

Ce qu'il reste à faire :

- Formaliser des **restrictions** (CBN/CBV) (Almost finished)
- Réaliser une **implémentation**
- Charactériser la **solvabilité** dans le $\lambda!$-calculus[9]

[9] A. Bucciarelli, D. Kesner, and S. Ronchi Della Rocca. Solvability = Typability + Inhabitation, 2021.

Merci de votre attention !