Quantitative Inhabitation for Different Lambda Calculi in a Unifying Framework

Victor Arrial [1] Giulio Guerrieri [2,3] Delia Kesner [1,4]

[1] Université Paris Cité, Paris [2] Aix Marseille Univ, Marseille

[3] Edinburgh Research Centre, Huawei, Edinburgh

[4] Institut Universitaire de France

Boston, January 20, 2023

Quantitative Inhabitation for Different Lambda Calculi in a Unifying Framework

Victor Arrial [1] Giulio Guerrieri [2,3] Delia Kesner [1,4]

[1] Université Paris Cité, Paris [2] Aix Marseille Univ, Marseille

[3] Edinburgh Research Centre, Huawei, Edinburgh

[4] Institut Universitaire de France

Boston, January 20, 2023

What is Inhabitation ?

What is Inhabitation ?

Typing Problem:

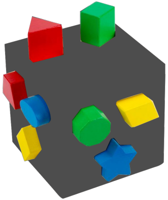
t

What is Inhabitation ?

Typing Problem:

$$\Gamma \vdash t : \sigma$$

What is Inhabitation ?



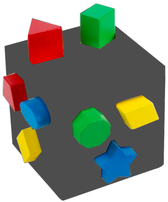
Typing Problem:

$$\Gamma \vdash t : \sigma$$

Computational: **[Mil'78]**

Typers

What is Inhabitation ?



Typing Problem:

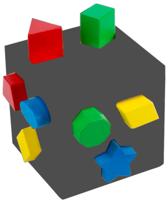
$\Gamma \vdash t : \sigma$

Inhabitation Problem (IP):

Computational: **[Mil'78]**

Typers

What is Inhabitation ?



Typing Problem:

$\Gamma \vdash t : \sigma$

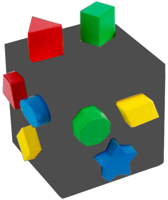
Inhabitation Problem (IP):

$\Gamma \qquad \sigma$

Computational: **[Mil'78]**

Typers

What is Inhabitation ?

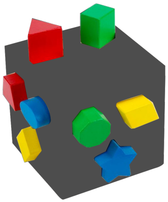


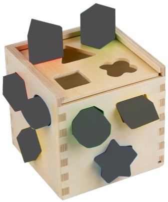
Typing Problem:
$\Gamma \vdash t : \sigma$

Inhabitation Problem (IP):
$\Gamma \vdash t : \sigma$

Computational: **[Mil'78]**
Typers

What is Inhabitation ?





Typing Problem:
$\Gamma \vdash t : \sigma$

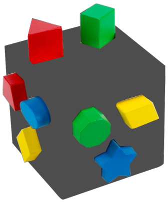
Inhabitation Problem (IP):
$\Gamma \vdash t : \sigma$

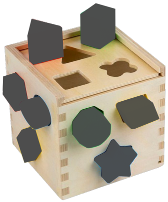
Computational: **[Mil'78]**
Typers

Computational: **[HuOr'20]**
Program Synthesis

Logical: **[HoMi'94]**
Proof Search and Logic Programming

What is Inhabitation ?





Typing Problem:

$\Gamma \vdash t : \sigma$

Inhabitation Problem (IP):

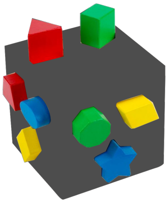
$\Gamma \vdash t : \sigma$

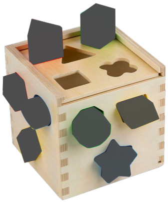
Computational: **[HuOr'20]**
Program Synthesis

Computational: **[Mil'78]**
Typers

Logical: **[HoMi'94]**
Proof Search and Logic Programming

What is Inhabitation ?





Typing Problem:

$\Gamma \vdash t : \sigma$

Inhabitation Problem (IP):

$\Gamma \vdash t : \sigma$

Computational: **[Mil'78]**
Typers

Computational: **[HuOr'20]**
Program Synthesis

Logical: **[HoMi'94]**
Proof Search and Logic Programming

Quantitative Inhabitation for Different Lambda Calculi in a Unifying Framework

Quantitative Inhabitation for Different Lambda Calculi in a Unifying Framework

Unifying Frameworks

Unifying Frameworks

Different Models of Computation:

Call-by-Name	Call-by-Value
NAME	VALUE

Quantitative Inhabitation for Different Lambda Calculi in a Unifying Framework

Quantitative Inhabitation for Different Lambda Calculi in a Unifying Framework

Unifying Frameworks

Different Models of Computation:

Call-by-Name

Call-by-Value

Unifying Frameworks:

- Call-by-Push-Value **[Levy'99]**

Unifying Frameworks

Different Models of Computation:

Call-by-Name

Call-by-Value

Unifying Frameworks:

- Call-by-Push-Value **[Levy'99]**
- **Bang Calculus [EG'16]**

Unifying Frameworks

Different Models of Computation:

Call-by-Name

NAME

Call-by-Value

VALUE

Unifying Frameworks:

- Call-by-Push-Value **[Levy'99]**
- **Bang Calculus [EG'16]**:

$$t, u \quad ::= \quad x \mid \lambda x.t \mid tu$$

BANG

Different Models of Computation:

Call-by-Name NAME

Call-by-Value VALUE

Unifying Frameworks:

- Call-by-Push-Value **[Levy'99]**
- **Bang Calculus [EG'16]**:

$$\begin{array}{rcl} t, u & ::= & x \mid \lambda x.t \mid tu \\ & \mid & !\,t \end{array}$$

Values

BANG

Unifying Frameworks

Different Models of Computation:

Call-by-Name

NAME

Call-by-Value

VALUE

Unifying Frameworks:

- Call-by-Push-Value **[Levy'99]**
- **Bang Calculus [EG'16]**:

$$
\begin{array}{rcll}
t, u & ::= & x \mid \lambda x.t \mid tu & \\
 & \mid & !\,t & \texttt{Values} \\
 & \mid & \texttt{der}(t) & \texttt{Computations}
\end{array}
$$

BANG

Unifying Frameworks

Different Models of Computation:

Call-by-Name

NAME

Call-by-Value

VALUE

Unifying Frameworks:

- Call-by-Push-Value **[Levy'99]**
- **Distant Bang Calculus [EG'16] [BKRV'20]**:

$$
\begin{array}{lcll}
t, u & ::= & x \mid \lambda x.t \mid tu & \\
 & \mid & !\,t & \texttt{Values} \\
 & \mid & \texttt{der}(\,t\,) & \texttt{Computations} \\
 & \mid & t\,[\,x := u\,] & \texttt{Let}
\end{array}
$$

BANG

Distant Bang: A Subsuming Paradigm

Distant Bang: A Subsuming Paradigm

t^{N} : NAME $\rightarrow$ BANG

Distant Bang: A Subsuming Paradigm

t^{N} : NAME $\rightarrow$ BANG

Static Properties: **[BKRV'20]**

NAME $\qquad$ t normal form

Distant Bang: A Subsuming Paradigm

$$t^{N} : \quad \text{NAME} \quad \rightarrow \quad \text{BANG}$$

Static Properties: **[BKRV'20]**

NAME $\quad t$ normal form $\quad \Leftrightarrow \quad t^{N}$ normal form $\quad$ BANG

Distant Bang: A Subsuming Paradigm

t^{N} : NAME $\rightarrow$ BANG

Static Properties: **[BKRV'20]**

NAME t normal form $\Leftrightarrow$ t^{N} normal form BANG

Dynamic Properties: **[BKRV'20]**

NAME $t \twoheadrightarrow u$

Distant Bang: A Subsuming Paradigm

t^{N} : NAME $\to$ BANG

Static Properties: **[BKRV'20]**

NAME t normal form $\Leftrightarrow$ t^{N} normal form BANG

Dynamic Properties: **[BKRV'20]**

NAME $t \twoheadrightarrow u$ $\Leftrightarrow$ $t^{N} \twoheadrightarrow u^{N}$ BANG

Distant Bang: A Subsuming Paradigm

t^N : NAME $\to$ BANG

t^V : VALUE $\to$ BANG

Static Properties: **[BKRV'20]**

NAME t normal form $\Leftrightarrow$ t^N normal form BANG

Dynamic Properties: **[BKRV'20]**

NAME $t \twoheadrightarrow u$ $\Leftrightarrow$ $t^N \twoheadrightarrow u^N$ BANG

Distant Bang: A Subsuming Paradigm

t^{N} : NAME $\to$ BANG

t^{V} : VALUE $\to$ BANG

Static Properties: **[BKRV'20]**

NAME	t normal form	$\Leftrightarrow$	t^{N} normal form	BANG
VALUE	t normal form	$\Leftrightarrow$	t^{V} normal form	

Dynamic Properties: **[BKRV'20]**

NAME	$t \twoheadrightarrow u$	$\Leftrightarrow$	$t^{N} \twoheadrightarrow u^{N}$	BANG
VALUE	$t \twoheadrightarrow u$	$\Leftrightarrow$	$t^{V} \twoheadrightarrow u^{V}$	

Distant Bang: A Subsuming Paradigm

t^{N} : NAME $\to$ BANG

t^{V} : VALUE $\to$ BANG

Static Properties: [BKRV'20]

NAME	t normal form	$\Leftrightarrow$	t^{N} normal form	BANG
VALUE	t normal form	$\Leftrightarrow$	t^{V} normal form	

Dynamic Properties: [BKRV'20]

NAME	$t \twoheadrightarrow u$	$\Leftrightarrow$	$t^{N} \twoheadrightarrow u^{N}$	BANG
VALUE	$t \twoheadrightarrow u$	$\Leftrightarrow$	$t^{V} \twoheadrightarrow u^{V}$	

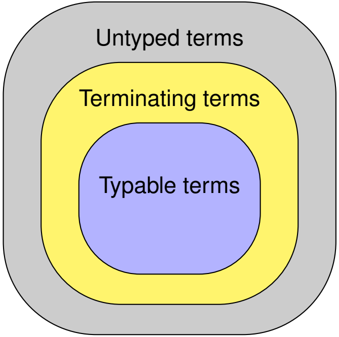
Can we do the same thing with inhabitation ?

Quantitative Inhabitation for Different Lambda Calculi in a Unifying Framework

Quantitative Inhabitation for Different Lambda Calculi in a Unifying Framework

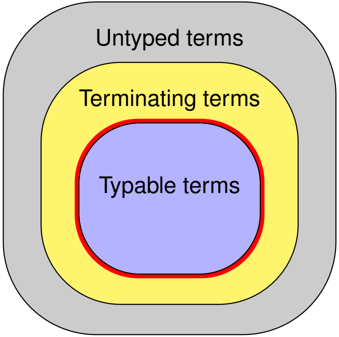
Simple Types Versus Intersection Types

$$A, B ::= \sigma \mid A \Rightarrow B$$



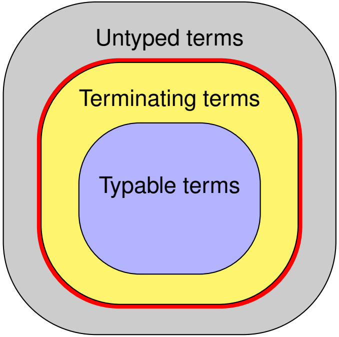
Simple Types Versus Intersection Types

$$A, B ::= \sigma \mid A \Rightarrow B$$



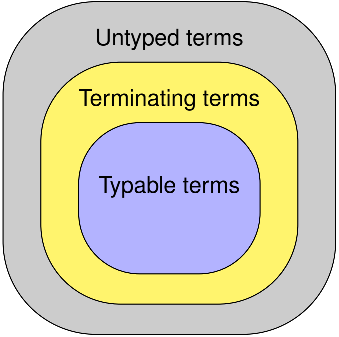
Simple Types Versus Intersection Types

$$A, B ::= \sigma \mid A \Rightarrow B$$



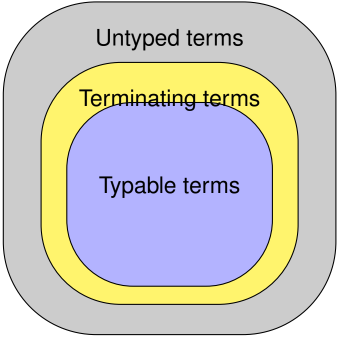
Simple Types Versus Intersection Types

$$A, B ::= \sigma \mid A \Rightarrow B \mid A \cap B$$



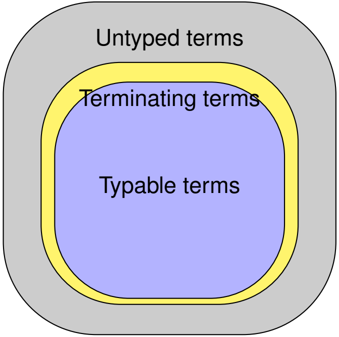
Simple Types Versus Intersection Types

$$A, B ::= \sigma \mid A \Rightarrow B \mid A \cap B$$



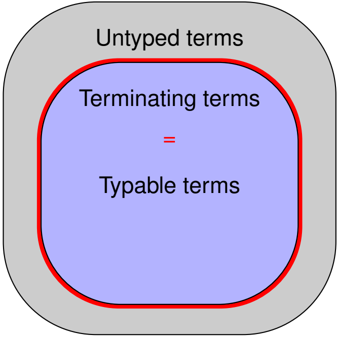
Simple Types Versus Intersection Types

$$A, B ::= \sigma \mid A \Rightarrow B \mid A \cap B$$



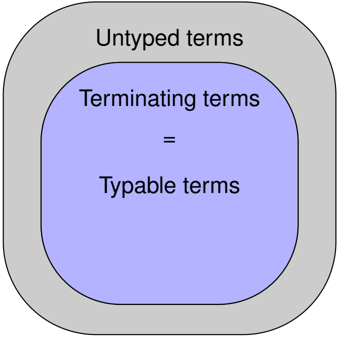
Simple Types Versus Intersection Types

$$A, B ::= \sigma \mid A \Rightarrow B \mid A \cap B$$



Simple Types Versus Intersection Types

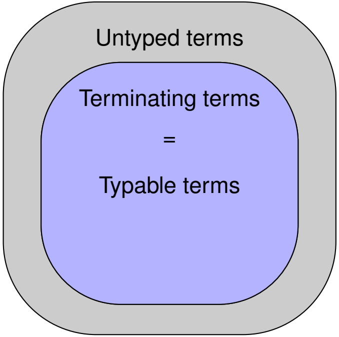
$$A, B ::= \sigma \mid A \Rightarrow B \mid {\color{red} A \cap B}$$



- **Associativity**:
  $A \cap (B \cap C) = (A \cap B) \cap C$

Simple Types Versus Intersection Types

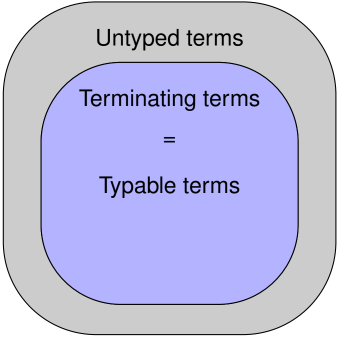
$$A, B ::= \sigma \mid A \Rightarrow B \mid \color{red}{A \cap B}$$



- **Associativity**:
  $A \cap (B \cap C) = (A \cap B) \cap C$
- **Commutativity**:
  $A \cap B = B \cap A$

Simple Types Versus Intersection Types

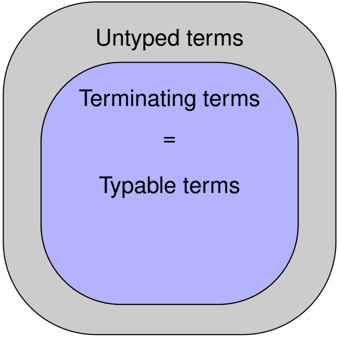
$$A, B ::= \sigma \mid A \Rightarrow B \mid \color{red}{A \cap B}$$



- **Associativity**:
  $A \cap (B \cap C) = (A \cap B) \cap C$
- **Commutativity**:
  $A \cap B = B \cap A$
- **Idempotency**?

Simple Types Versus Intersection Types

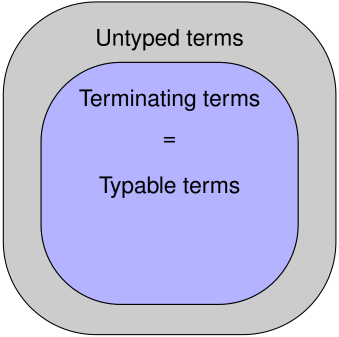
$$A, B ::= \sigma \mid A \Rightarrow B \mid A \cap B$$



- **Associativity**:
  $A \cap (B \cap C) = (A \cap B) \cap C$
- **Commutativity**:
  $A \cap B = B \cap A$
- **Idempotency**?

Idempotent [CoDe'78],[CoDe'80] $A \cap A = A$	

$$A, B ::= \sigma \mid A \Rightarrow B \mid A \cap B$$

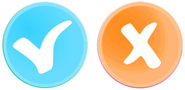


- **Associativity**:
  $A \cap (B \cap C) = (A \cap B) \cap C$
- **Commutativity**:
  $A \cap B = B \cap A$
- **Idempotency**?

Idempotent
[CoDe'78],[CoDe'80]

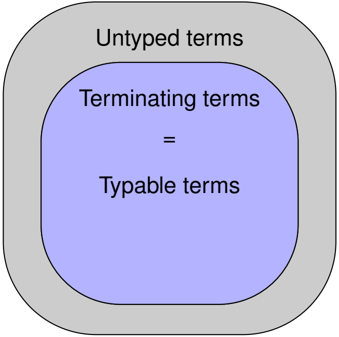
$A \cap A = A$

Qualitative properties

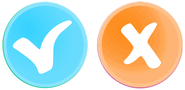


Simple Types Versus Intersection Types

$$A, B ::= \sigma \mid A \Rightarrow B \mid A \cap B$$

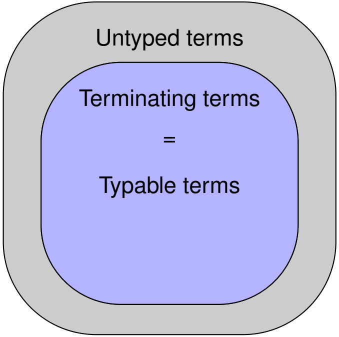


- **Associativity**:
  $A \cap (B \cap C) = (A \cap B) \cap C$
- **Commutativity**:
  $A \cap B = B \cap A$
- **Idempotency**?

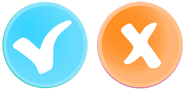
Idempotent **[CoDe'78],[CoDe'80]**	**Non-Idempotent** **[Gard'94], [Kfou'00]**
$A \cap A = A$	$A \cap A \neq A$
Qualitative properties 	

Simple Types Versus Intersection Types

$$A, B ::= \sigma \mid A \Rightarrow B \mid A \cap B$$



- **Associativity**:
  $A \cap (B \cap C) = (A \cap B) \cap C$
- **Commutativity**:
  $A \cap B = B \cap A$
- **Idempotency**?

Idempotent **[CoDe'78],[CoDe'80]** $A \cap A = A$	**Non-Idempotent** **[Gard'94], [Kfou'00]** $A \cap A \neq A$
Qualitative properties 	Quantitative properties **[dCarv'07]** 

Typability and Inhabitation in Intersection Types

	Typing $? \vdash t : \ ?$	**Inhabitation** $\Gamma \vdash \ ? : \sigma$

	Typing $? \vdash t : \; ?$	**Inhabitation** $\Gamma \vdash \; ? : \sigma$
Simple Types		
Idempotent Types		
Non-Idempotent Types		

	Typing $? \vdash t : ?$	**Inhabitation** $\Gamma \vdash ? : \sigma$
Simple Types	Decidable	
Idempotent Types		
Non-Idempotent Types		

Typability and Inhabitation in Intersection Types

	Typing $? \vdash t : ?$	**Inhabitation** $\Gamma \vdash ? : \sigma$
Simple Types	Decidable	
Idempotent Types	Indecidable	
Non-Idempotent Types	Indecidable	

	Typing $? \vdash t : \, ?$	**Inhabitation** $\Gamma \vdash \, ? : \sigma$
Simple Types	Decidable	Decidable
Idempotent Types	Indecidable	
Non-Idempotent Types	Indecidable	

	Typing $? \vdash t : \ ?$	**Inhabitation** $\Gamma \vdash \ ? : \sigma$
Simple Types	Decidable	Decidable
Idempotent Types	Indecidable	Indecidable **[Urz'99]**
Non-Idempotent Types	Indecidable	

	Typing $? \vdash t : ?$	**Inhabitation** $\Gamma \vdash ? : \sigma$
Simple Types	Decidable	Decidable
Idempotent Types	Indecidable	Indecidable **[Urz'99]**
Non-Idempotent Types	Indecidable	(CBN) Decidable **[BKR'18]**

	Typing $? \vdash t : ?$	**Inhabitation** $\Gamma \vdash ? : \sigma$
Simple Types	Decidable	Decidable
Idempotent Types	Indecidable	Indecidable **[Urz'99]**
Non-Idempotent Types	Indecidable	(CBN) Decidable **[BKR'18]** (CBV) ?

	Typing $? \vdash t : ?$	**Inhabitation** $\Gamma \vdash ? : \sigma$
Simple Types	Decidable	Decidable
Idempotent Types	Indecidable	Indecidable **[Urz'99]**
Non-Idempotent Types	Indecidable	(CBN) Decidable **[BKR'18]** (CBV) ?

Typability and Inhabitation in Intersection Types

	Typing $? \vdash t : ?$	**Inhabitation** $\Gamma \vdash ? : \sigma$
Simple Types	Decidable	Decidable
Idempotent Types	Indecidable	Indecidable **[Urz'99]**
Non-Idempotent Types	Indecidable	(CBN) Decidable **[BKR'18]** (CBV) Decidable

Intersection Types and Distant Bang Calculus

Three Typing Systems: **[BKRV'20]**

$\boxed{\text{NAME}} : \mathcal{N}$ $\boxed{\text{VALUE}} : \mathcal{V}$ $\boxed{\text{BANG}} : \mathcal{B}$

Three Typing Systems: **[BKRV'20]**

NAME : $\mathcal{N}$ VALUE : $\mathcal{V}$ BANG : $\mathcal{B}$

Static Properties: **[BKRV'20]**

NAME $\Gamma \vdash_{\mathcal{N}} t : \sigma$

Three Typing Systems: **[BKRV'20]**

NAME : $\mathcal{N}$ VALUE : $\mathcal{V}$ BANG : $\mathcal{B}$

Static Properties: **[BKRV'20]**

NAME $\Gamma \vdash_{\mathcal{N}} t : \sigma \quad \Leftrightarrow \quad \Gamma \vdash_{\mathcal{B}} t^{N} : \sigma$ BANG

Three Typing Systems: **[BKRV'20]**

NAME : $\mathcal{N}$ VALUE : $\mathcal{V}$ BANG : $\mathcal{B}$

Static Properties: **[BKRV'20]**

NAME	$\Gamma \vdash_{\mathcal{N}} t : \sigma$	$\Leftrightarrow$	$\Gamma \vdash_{\mathcal{B}} t^{N} : \sigma$	BANG
VALUE	$\Gamma \vdash_{\mathcal{V}} t : \sigma$	$\Leftrightarrow$	$\Gamma \vdash_{\mathcal{B}} t^{V} : \sigma$	

Quantitative Inhabitation for Different Lambda Calculi in a Unifying Framework

Coming Back to Inhabitation

Coming Back to Inhabitation

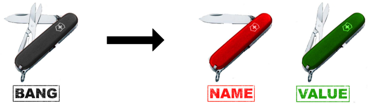
First Goal

- **Decidability** of the (more general) $\boxed{\textbf{BANG}}$ Inhabitation Problem (IP).

Coming Back to Inhabitation

First Goal + More Ambitious Second Goal

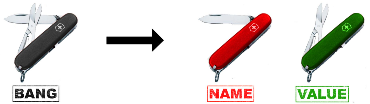
- **Decidability** of the (more general) BANG Inhabitation Problem (IP).
- **Decidability** of the NAME and VALUE IP from **decidability** of the BANG IP.



Coming Back to Inhabitation

First Goal + More Ambitious Second Goal

- **Decidability** of the (more general) $\boxed{\text{BANG}}$ Inhabitation Problem (IP).
- **Decidability** of the $\boxed{\text{NAME}}$ and $\boxed{\text{VALUE}}$ IP from **decidability** of the $\boxed{\text{BANG}}$ IP.

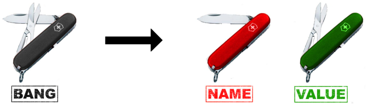


More Ambitious Third Goal

Coming Back to Inhabitation

First Goal + More Ambitious Second Goal

- **Decidability** of the (more general) BANG Inhabitation Problem (IP).
- **Decidability** of the NAME and VALUE IP from **decidability** of the BANG IP.



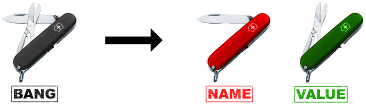
More Ambitious Third Goal

- Decidability by **finding all inhabitants** in the BANG IP.

Coming Back to Inhabitation

First Goal + More Ambitious Second Goal

- **Decidability** of the (more general) BANG Inhabitation Problem (IP).
- **Decidability** of the NAME and VALUE IP from **decidability** of the BANG IP.



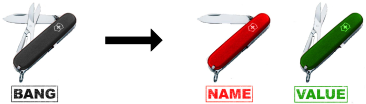
More Ambitious Third Goal

- Decidability by **finding all inhabitants** in the BANG IP.
- Decidability of the NAME and VALUE IP by **finding all inhabitants** from those of the BANG IP.

Coming Back to Inhabitation

First Goal + More Ambitious Second Goal

- **Decidability** of the (more general) BANG Inhabitation Problem (IP).
- **Decidability** of the NAME and VALUE IP from **decidability** of the BANG IP.



More Ambitious Third Goal

- Decidability by **finding all inhabitants** in the BANG IP.
- Decidability of the NAME and VALUE IP by **finding all inhabitants** from those of the BANG IP.
- Using generic properties so that other encodable models of computation can use these results.

Solving the Inhabitation Problem - Methodology

Solving the Inhabitation Problem - Methodology

Instead of **just one** solution:

$$\Gamma \vdash \mathbf{t} : \sigma$$

We want to compute **all** solutions:

$$\mathtt{Sol}(\Gamma, \sigma) \ := \ \{\, \mathbf{t} \mid \Gamma \vdash \mathbf{t} : \sigma \,\}$$

Solving the Inhabitation Problem - Methodology

Instead of **just one** solution:

$$\Gamma \vdash \mathbf{t} : \sigma$$

We want to compute **all** solutions:

$$\texttt{Sol}(\Gamma, \sigma) \; := \; \{ \mathbf{t} \mid \Gamma \vdash \mathbf{t} : \sigma \}$$

Solving the Inhabitation Problem - Methodology

Instead of **just one** solution:

$$\Gamma \vdash \mathbf{t} : \sigma$$

We want to compute **all** solutions:

$$\texttt{Sol}(\Gamma, \sigma) \;:=\; \{\, \mathbf{t} \mid \Gamma \vdash \mathbf{t} : \sigma \,\}$$

Problem

- The set $\texttt{Sol}(\Gamma, \sigma)$ is either empty of infinite **BANG**

Solving the Inhabitation Problem - Methodology

Instead of **just one** solution:

$$\Gamma \vdash \mathbf{t} : \sigma$$

We want to compute **all** solutions:

$$\mathtt{Sol}(\Gamma, \sigma) \;:=\; \{\, \mathbf{t} \mid \Gamma \vdash \mathbf{t} : \sigma \,\}$$

Problem

- The set $\mathtt{Sol}(\Gamma, \sigma)$ is either empty of infinite **BANG**

We compute a **finite** generator:

$$\mathtt{Basis}(\Gamma, \sigma)$$

Which is **correct** and **complete**:

$$\mathtt{span}(\,\mathtt{Basis}(\Gamma, \sigma)\,) \;=\; \mathtt{Sol}(\Gamma, \sigma)$$

Solving the Inhabitation Problem - Methodology

Instead of **just one** solution:

$$\Gamma \vdash \mathbf{t} : \sigma$$

We want to compute **all** solutions:

$$\mathtt{Sol}(\Gamma, \sigma) \;:=\; \{\, \mathbf{t} \mid \Gamma \vdash \mathbf{t} : \sigma \,\}$$

Problem

The set $\mathtt{Sol}(\Gamma, \sigma)$ is either empty of infinite **BANG**

We compute a **finite** generator:

$$\mathtt{Basis}(\Gamma, \sigma)$$

Which is **correct** and **complete**:

$$\mathtt{span}(\,\mathtt{Basis}(\Gamma, \sigma)\,) \;=\; \mathtt{Sol}(\Gamma, \sigma)$$

Solving the Inhabitation Problem - Methodology

Instead of **just one** solution:

$$\Gamma \vdash \mathbf{t} : \sigma$$

We want to compute **all** solutions:

$$\texttt{Sol}(\Gamma, \sigma) \;:=\; \{\, \mathbf{t} \mid \Gamma \vdash \mathbf{t} : \sigma \,\}$$

Problem

- The set $\texttt{Sol}(\Gamma, \sigma)$ is either empty of infinite **BANG**

We compute a **finite** generator:

$$\texttt{Basis}(\Gamma, \sigma)$$

Which is **correct** and **complete**:

$$\texttt{span}(\texttt{Basis}(\Gamma, \sigma)) \;=\; \texttt{Sol}(\Gamma, \sigma)$$

Solving the Inhabitation Problem - Methodology

Instead of **just one** solution:

$$\Gamma \vdash \mathbf{t} : \sigma$$

We want to compute **all** solutions:

$$\mathtt{Sol}(\Gamma, \sigma) \ := \ \{\, \mathbf{t} \mid \Gamma \vdash \mathbf{t} : \sigma \,\}$$

Problem

The set $\mathtt{Sol}(\Gamma, \sigma)$ is either empty of infinite BANG

We compute a **finite** generator:

$$\mathtt{Basis}(\Gamma, \sigma)$$

Which is **correct** and **complete**:

$$\mathtt{span}(\,\mathtt{Basis}(\Gamma, \sigma)\,) \ = \ \mathtt{Sol}(\Gamma, \sigma)$$

Theorem

For any typing (Γ, σ), $\mathtt{Basis}_{\mathcal{B}}(\Gamma, \sigma)$ ***exists**, is **finite**, **correct** and **complete**.* BANG

Solving the Inhabitation Problem - Methodology

Instead of **just one** solution:

$$\Gamma \vdash \mathbf{t} : \sigma$$

We want to compute **all** solutions:

$$\mathtt{Sol}(\Gamma, \sigma) \ := \ \{\, \mathbf{t} \mid \Gamma \vdash \mathbf{t} : \sigma \,\}$$

Problem

The set $\mathtt{Sol}(\Gamma, \sigma)$ is either empty of infinite BANG

We compute a **finite** generator:

$$\mathtt{Basis}(\Gamma, \sigma)$$

Which is **correct** and **complete**:

$$\mathtt{span}(\,\mathtt{Basis}(\Gamma, \sigma)\,) \ = \ \mathtt{Sol}(\Gamma, \sigma)$$

Theorem

*For any typing (Γ, σ), $\mathtt{Basis}_{\mathcal{B}}(\Gamma, \sigma)$ **exists**, is **finite**, **correct** and **complete**.* BANG

Following the Typing and a Grammar

Computing the basis:
Recreate typing trees, but only on elements of the `Basis`.

Computing the basis:
Recreate typing trees, but only on elements of the `Basis`.

Follows two sets of rules:

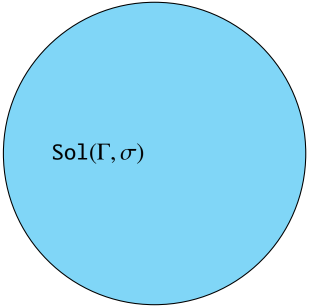
Following the Typing and a Grammar

Computing the basis:

Recreate typing trees, but only on elements of the `Basis`.

Follows two sets of rules:

- Typing rules



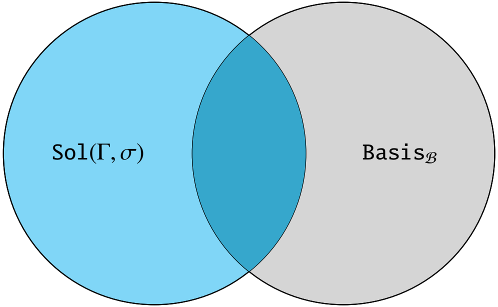
Following the Typing and a Grammar

Computing the basis:

Recreate typing trees, but only on elements of the `Basis`.

Follows two sets of rules:

- Typing rules
- Grammar rules

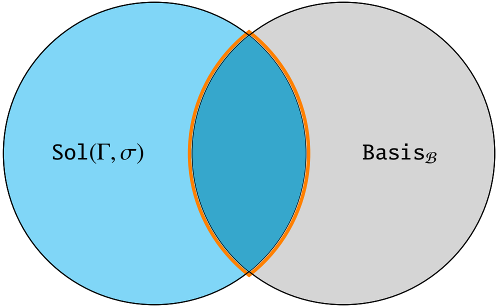


Computing the basis:

Recreate typing trees, but only on elements of the `Basis`.

Follows two sets of rules:

- Typing rules
- Grammar rules



The Full Algorithm

$$\frac{g \rightsquigarrow \mathsf{Var} \quad \mid}{x \Vdash_g H^{x:[\sigma]}(\emptyset;\sigma)}\ \textsf{VAR}
\qquad
\frac{\begin{array}{c} g \rightsquigarrow \mathsf{Der}(g') \\ [\sigma] \Vvdash S(\tau,\diamond) \end{array} \ \Bigg| \ a \Vdash_{g'} H^{x:[\tau]}(\Gamma;[\sigma])}{\mathsf{der}(a) \Vdash_g H^{x:[\tau]}(\Gamma;\sigma)}\ \textsf{DR}$$

$$\frac{\begin{array}{c} g \rightsquigarrow \mathsf{App}(g_a,g_b) \\ \Gamma = \Gamma_a + \Gamma_b \\ \mathcal{M} \Rightarrow \sigma \Vvdash S(\tau, \diamond \Rightarrow \sigma) \end{array} \ \Bigg| \ a \Vdash_{g_a} H^{x:[\tau]}(\Gamma_a; \mathcal{M} \Rightarrow \sigma) \quad b \Vdash_{g_b} N(\Gamma_b;\mathcal{M})}{ab \Vdash_g H^{x:[\tau]}(\Gamma;\sigma)}\ \textsf{APP}$$

$$\frac{g \rightsquigarrow g' \ \mid \ a \Vdash_{g'} H^{x:[\tau]}(\Gamma;\sigma)}{a \Vdash_g H^{x:[\tau]}(\Gamma;\sigma)}\ \textsf{H-H}
\qquad
\frac{\begin{array}{c} g \rightsquigarrow g' \\ \Gamma = \Gamma' + x:[\tau] \\ \sigma \Vvdash S(\tau,\diamond) \end{array} \ \Bigg| \ a \Vdash_{g'} H^{x:[\tau]}(\Gamma';\sigma)}{a \Vdash_g N(\Gamma;\sigma)}\ \textsf{N-H}
\qquad
\frac{g \rightsquigarrow g' \ \mid \ a \Vdash_{g'} N(\Gamma;\sigma)}{a \Vdash_g N(\Gamma;\sigma)}\ \textsf{N-N}$$

$$\frac{\begin{array}{c} g \rightsquigarrow \mathsf{Lam}(g') \\ \mathtt{fix}\ \ x \notin \mathsf{dom}(\Gamma) \end{array} \ \Bigg| \ a \Vdash_{g'} N(\Gamma, x:\mathcal{M};\sigma)}{\lambda x.a \Vdash_g N(\Gamma;\mathcal{M} \Rightarrow \sigma)}\ \textsf{ABS}
\qquad
\frac{\begin{array}{c} g \rightsquigarrow \mathsf{Bng}(g') \\ I \neq \emptyset \\ \Gamma = +_{i \in I}\Gamma_i \end{array} \ \Bigg| \ (a_i \Vdash_{g'} N(\Gamma_i;\tau_i))_{i \in I} \quad \uparrow_{i \in I} a_i}{! \bigvee_{i \in I} a_i \Vdash_g N(\Gamma;[\tau_i]_{i \in I})}\ \textsf{BG}
\qquad
\frac{g \rightsquigarrow \mathsf{Bng}(\bot) \quad \mid}{!\bot \Vdash_g N(\emptyset;[\,])}\ \textsf{BG}_\bot$$

$$\frac{\begin{array}{c} g \rightsquigarrow \mathsf{Sub}(g_a,g_b) \\ \Gamma = \Gamma_a + \Gamma_b + z:[\rho], \quad \mathtt{fix}\ \ y \notin \mathsf{dom}(\Gamma) \cup \{x\} \\ n \in \llbracket 0, \mathsf{sz}(\rho) \rrbracket, \ \mathcal{M} \Vvdash S(\rho,[\diamond_1,\ldots,\diamond_n]) \end{array} \ \Bigg| \ a \Vdash_{g_a} H^{x:[\tau]}(\Gamma_a, y:\mathcal{M};\sigma) \quad b \Vdash_{g_b} H^{z:[\rho]}(\Gamma_b;\mathcal{M})}{a[y \backslash b] \Vdash_g H^{x:[\tau]}(\Gamma;\sigma)}\ \textsf{ES-H}$$

$$\frac{\begin{array}{c} g \rightsquigarrow \mathsf{Sub}(g_a,g_b) \\ \Gamma = \Gamma_a + \Gamma_b, \quad \mathtt{fix}\ \ y \notin \mathsf{dom}(\Gamma) \cup \{x\} \\ n \in \llbracket 1, \mathsf{sz}(\tau) \rrbracket, \ [\rho_i]_{i \in \llbracket 1,n \rrbracket} \Vvdash S(\tau,[\diamond_1,\ldots,\diamond_n]) \\ j \in \llbracket 1,n \rrbracket, \ \sigma \Vvdash S(\rho_j,\diamond) \end{array} \ \Bigg| \ a \Vdash_{g_a} H^{y:[\rho_j]}(\Gamma_a, y:[\rho_i]_{i \in \llbracket 1,n \rrbracket \backslash j};\sigma) \quad b \Vdash_{g_b} H^{x:[\tau]}(\Gamma_b;[\rho_i]_{i \in \llbracket 1,n \rrbracket})}{a[y \backslash b] \Vdash_g H^{x:[\tau]}(\Gamma;\sigma)}\ \textsf{ES-CH}$$

$$\frac{\begin{array}{c} g \rightsquigarrow \mathsf{Sub}(g_a,g_b) \\ \Gamma = \Gamma_a + \Gamma_b + z:[\tau], \quad \mathtt{fix}\ \ y \notin \mathsf{dom}(\Gamma) \\ n \in \llbracket 0, \mathsf{sz}(\tau) \rrbracket, \ \mathcal{M} \Vvdash S(\tau,[\diamond_1,\ldots,\diamond_n]) \end{array} \ \Bigg| \ a \Vdash_{g_a} N(\Gamma_a, y:\mathcal{M};\sigma) \quad b \Vdash_{g_b} H^{z:[\tau]}(\Gamma_b;\mathcal{M})}{a[y \backslash b] \Vdash_g N(\Gamma;\sigma)}\ \textsf{ES-N}$$

The Full Algorithm

$$\dfrac{g \rightsquigarrow \mathsf{Var} \quad \mid}{x \Vdash_g H^{x:[\sigma]}(\emptyset;\sigma)}\ \text{VAR} \qquad \dfrac{\begin{array}{c} g \rightsquigarrow \mathsf{Der}(g') \\ [\sigma] \Vvdash S(\tau,\diamond) \end{array} \;\Big|\; a \Vdash_{g'} H^{x:[\tau]}(\Gamma;[\sigma])}{\mathsf{der}(a) \Vdash_g H^{x:[\tau]}(\Gamma;\sigma)}\ \text{DR}$$

$$\dfrac{\begin{array}{c} g \rightsquigarrow \mathsf{App}(g_a, g_b) \\ \Gamma = \Gamma_a + \Gamma_b \\ \mathcal{M} \Rightarrow \sigma \Vvdash S(\tau, \diamond \Rightarrow \sigma) \end{array} \;\Big|\; a \Vdash_{g_a} H^{x:[\tau]}(\Gamma_a; \mathcal{M} \Rightarrow \sigma) \quad b \Vdash_{g_b} N(\Gamma_b; \mathcal{M})}{ab \Vdash_g H^{x:[\tau]}(\Gamma;\sigma)}\ \text{APP}$$

$$\dfrac{g \rightsquigarrow g' \;\mid\; a \Vdash_{g'} H^{x:[\tau]}(\Gamma;\sigma)}{a \Vdash_g H^{x:[\tau]}(\Gamma;\sigma)}\ \text{H-H} \qquad \dfrac{\begin{array}{c} g \rightsquigarrow g' \\ \Gamma = \Gamma' + x:[\tau] \\ \sigma \Vvdash S(\tau,\diamond) \end{array} \;\Big|\; a \Vdash_{g'} H^{x:[\tau]}(\Gamma';\sigma)}{a \Vdash_g N(\Gamma;\sigma)}\ \text{N-H} \qquad \dfrac{g \rightsquigarrow g' \;\mid\; a \Vdash_{g'} N(\Gamma;\sigma)}{a \Vdash_g N(\Gamma;\sigma)}\ \text{N-N}$$

$$\dfrac{\begin{array}{c} g \rightsquigarrow \mathsf{Lam}(g') \\ \mathtt{fix}\ x \notin \mathsf{dom}(\Gamma) \end{array} \;\Big|\; a \Vdash_{g'} N(\Gamma, x:\mathcal{M};\sigma)}{\lambda x.a \Vdash_g N(\Gamma;\mathcal{M} \Rightarrow \sigma)}\ \text{ABS} \qquad \dfrac{\begin{array}{c} g \rightsquigarrow \mathsf{Bng}(g') \\ I \neq \emptyset \\ \Gamma = +_{i \in I}\Gamma_i \end{array} \;\Big|\; (a_i \Vdash_{g'} N(\Gamma_i;\tau_i))_{i\in I} \quad \uparrow_{i \in I} a_i}{!\bigvee_{i\in I} a_i \Vdash_g N(\Gamma;[\tau_i]_{i\in I})}\ \text{BG} \qquad \dfrac{g \rightsquigarrow \mathsf{Bng}(\bot) \quad \mid}{!\bot \Vdash_g N(\emptyset;[\,])}\ \text{BG}_\bot$$

$$\dfrac{\begin{array}{c} g \rightsquigarrow \mathsf{Sub}(g_a,g_b) \\ \Gamma = \Gamma_a + \Gamma_b + z:[\rho], \quad \mathtt{fix}\ \ y \notin \mathsf{dom}(\Gamma) \cup \{x\} \\ n \in \llbracket 0, \mathsf{sz}(\rho) \rrbracket,\ \mathcal{M} \Vvdash S(\rho,[\diamond_1,\dots,\diamond_n]) \end{array} \;\Big|\; a \Vdash_{g_a} H^{x:[\tau]}(\Gamma_a, y:\mathcal{M};\sigma) \quad b \Vdash_{g_b} H^{z:[\rho]}(\Gamma_b;\mathcal{M})}{a[y \backslash b] \Vdash_g H^{x:[\tau]}(\Gamma;\sigma)}\ \text{ES-H}$$

$$\dfrac{\begin{array}{c} g \rightsquigarrow \mathsf{Sub}(g_a,g_b) \\ \Gamma = \Gamma_a + \Gamma_b, \quad \mathtt{fix}\ \ y \notin \mathsf{dom}(\Gamma) \cup \{x\} \\ n \in \llbracket 1, \mathsf{sz}(\tau) \rrbracket,\ [\rho_i]_{i \in \llbracket 1,n \rrbracket} \Vvdash S(\tau,[\diamond_1,\dots,\diamond_n]) \\ j \in \llbracket 1, n \rrbracket,\ \sigma \Vvdash S(\rho_j,\diamond) \end{array} \;\Big|\; a \Vdash_{g_a} H^{y:[\rho_j]}(\Gamma_a, y:[\rho_i]_{i\in\llbracket 1,n \rrbracket \setminus j};\sigma) \quad b \Vdash_{g_b} H^{x:[\tau]}(\Gamma_b;[\rho_i]_{i\in\llbracket 1,n \rrbracket})}{a[y \backslash b] \Vdash_g H^{x:[\tau]}(\Gamma;\sigma)}\ \text{ES-CH}$$

$$\dfrac{\begin{array}{c} g \rightsquigarrow \mathsf{Sub}(g_a,g_b) \\ \Gamma = \Gamma_a + \Gamma_b + z:[\tau], \quad \mathtt{fix}\ \ y \notin \mathsf{dom}(\Gamma) \\ n \in \llbracket 0, \mathsf{sz}(\tau) \rrbracket,\ \mathcal{M} \Vvdash S(\tau,[\diamond_1,\dots,\diamond_n]) \end{array} \;\Big|\; a \Vdash_{g_a} N(\Gamma_a, y:\mathcal{M};\sigma) \quad b \Vdash_{g_b} H^{z:[\tau]}(\Gamma_b;\mathcal{M})}{a[y \backslash b] \Vdash_g N(\Gamma;\sigma)}\ \text{ES-N}$$

The Full Algorithm

$$\dfrac{g \rightsquigarrow \mathsf{Var} \mid}{x \Vdash_g H^{x:[\sigma]}(\emptyset;\sigma)}\,\textsf{VAR} \qquad \dfrac{\begin{array}{c} g \rightsquigarrow \mathtt{Der}(g') \\ [\sigma] \Vvdash S(\tau,\Diamond) \end{array} \;\middle|\; a \Vdash_{g'} H^{x:[\tau]}(\Gamma;[\sigma])}{\mathsf{der}(a) \Vdash_g H^{x:[\tau]}(\Gamma;\sigma)}\,\textsf{DR}$$

$$\dfrac{g \rightsquigarrow g' \mid a \Vdash_{g'} H^{x:[\tau]}(\Gamma;\sigma)}{a \Vdash_g H^{x:[\tau]}(\Gamma;\sigma)}\,\textsf{H-H} \qquad \dfrac{\begin{array}{c} g \rightsquigarrow g' \\ \Gamma = \Gamma' + x : [\tau] \\ \sigma \Vvdash S(\tau,\Diamond) \end{array} \;\middle|\; a \Vdash_{g'} H^{x:[\tau]}(\Gamma';\sigma)}{a \Vdash_g N(\Gamma;\sigma)}\,\textsf{N-H} \qquad \dfrac{g \rightsquigarrow g' \mid a \Vdash_{g'} N(\Gamma;\sigma)}{a \Vdash_g N(\Gamma;\sigma)}\,\textsf{N-N}$$

$$\dfrac{\begin{array}{c} g \rightsquigarrow \mathsf{App}(g_a, g_b) \\ \Gamma = \Gamma_a + \Gamma_b \\ \mathcal{M} \Rightarrow \sigma \Vvdash S(\tau, \Diamond \Rightarrow \sigma) \end{array} \;\middle|\; a \Vdash_{g_a} H^{x:[\tau]}(\Gamma_a; \mathcal{M} \Rightarrow \sigma) \quad b \Vdash_{g_b} N(\Gamma_b; \mathcal{M})}{ab \Vdash_g H^{x:[\tau]}(\Gamma;\sigma)}\,\textsf{APP}$$

$$\dfrac{n \in [\![0, \mathrm{sz}(\rho)]\!],\ \mathcal{M} \Vvdash S(\rho, [\Diamond_1, \ldots, \Diamond_n]) \;\middle|\; a \Vdash_{g_a} H^{x:[\tau]}(\Gamma_a, y : \mathcal{M}; \sigma) \quad b \Vdash_{g_b} H^{z:[\rho]}(\Gamma_b; \mathcal{M})}{a[y\backslash b] \Vdash_g H^{x:[\tau]}(\Gamma;\sigma)}\,\textsf{ES-H}$$

$$\dfrac{\begin{array}{c} g \rightsquigarrow \mathsf{Sub}(g_a, g_b) \\ \Gamma = \Gamma_a + \Gamma_b, \quad \mathtt{fix}\ \ y \notin \mathrm{dom}(\Gamma) \cup \{x\} \\ n \in [\![1, \mathrm{sz}(\tau)]\!],\ [\rho_i]_{i \in [\![1,n]\!]} \Vvdash S(\tau, [\Diamond_1, \ldots, \Diamond_n]) \\ j \in [\![1, n]\!],\ \sigma \Vvdash S(\rho_j, \Diamond) \end{array} \;\middle|\; a \Vdash_{g_a} H^{y:[\rho_j]}(\Gamma_a, y : [\rho_i]_{i \in [\![1,n]\!] \backslash j}; \sigma) \quad b \Vdash_{g_b} H^{x:[\tau]}(\Gamma_b; [\rho_i]_{i \in [\![1,n]\!]})}{a[y\backslash b] \Vdash_g H^{x:[\tau]}(\Gamma;\sigma)}\,\textsf{ES-CH}$$

$$\dfrac{\begin{array}{c} g \rightsquigarrow \mathsf{Sub}(g_a, g_b) \\ \Gamma = \Gamma_a + \Gamma_b + z : [\tau], \quad \mathtt{fix}\ \ y \notin \mathrm{dom}(\Gamma) \\ n \in [\![0, \mathrm{sz}(\tau)]\!],\ \mathcal{M} \Vvdash S(\tau, [\Diamond_1, \ldots, \Diamond_n]) \end{array} \;\middle|\; a \Vdash_{g_a} N(\Gamma_a, y : \mathcal{M}; \sigma) \quad b \Vdash_{g_b} H^{z:[\tau]}(\Gamma_b; \mathcal{M})}{a[y\backslash b] \Vdash_g N(\Gamma;\sigma)}\,\textsf{ES-N}$$

The Full Algorithm

$$\frac{g \rightsquigarrow \mathsf{Var} \quad |}{x \Vdash_g H^{x:[\sigma]}(\emptyset; \sigma)}\,\text{VAR}
\qquad
\frac{\begin{array}{c} g \rightsquigarrow \mathsf{Der}(g') \\ [\sigma] \Vdash S(\tau, \Diamond) \end{array} \;\Big|\; a \Vdash_{g'} H^{x:[\tau]}(\Gamma; [\sigma])}{\mathsf{der}(a) \Vdash_g H^{x:[\tau]}(\Gamma; \sigma)}\,\text{DR}$$

$$\frac{g \rightsquigarrow g' \;\;|\;\; a \Vdash_{g'} H^{x:[\tau]}(\Gamma; \sigma)}{a \Vdash_g H^{x:[\tau]}(\Gamma; \sigma)}\,\text{H-H}
\qquad
\frac{\begin{array}{c} g \rightsquigarrow g' \\ \Gamma = \Gamma' + x : [\tau] \\ \sigma \Vdash S(\tau, \Diamond) \end{array} \;\Big|\; a \Vdash_{g'} H^{x:[\tau]}(\Gamma'; \sigma)}{a \Vdash_g N(\Gamma; \sigma)}\,\text{N-H}
\qquad
\frac{g \rightsquigarrow g' \;\;|\;\; a \Vdash_{g'} N(\Gamma; \sigma)}{a \Vdash_g N(\Gamma; \sigma)}\,\text{N-N}$$

$$\frac{\begin{array}{c} g \rightsquigarrow \mathsf{App}(g_a, g_b) \\ \Gamma = \Gamma_a + \Gamma_b \\ \mathcal{M} \Rightarrow \sigma \Vdash S(\tau, \Diamond \Rightarrow \sigma) \end{array} \;\Bigg|\; a \Vdash_{g_a} H^{x:[\tau]}(\Gamma_a; \mathcal{M} \Rightarrow \sigma) \quad b \Vdash_{g_b} N(\Gamma_b; \mathcal{M})}{ab \Vdash_g H^{x:[\tau]}(\Gamma; \sigma)}\,\text{APP}$$

$$\frac{n \in [\![0, \mathsf{sz}(\rho)]\!],\ \mathcal{M} \Vdash S(\rho, [\Diamond_1, \ldots, \Diamond_n]) \;\;|\;\; a \Vdash_{g_a} H^{x:[\tau]}(\Gamma_a, y : \mathcal{M}; \sigma) \quad b \Vdash_{g_b} H^{z:[\rho]}(\Gamma_b; \mathcal{M})}{a[y \backslash b] \Vdash_g H^{x:[\tau]}(\Gamma; \sigma)}\,\text{ES-H}$$

$$\frac{\begin{array}{c} g \rightsquigarrow \mathsf{Sub}(g_a, g_b) \\ \Gamma = \Gamma_a + \Gamma_b, \qquad \mathtt{fix}\ \ y \notin \mathrm{dom}(\Gamma) \cup \{x\} \\ n \in [\![1, \mathsf{sz}(\tau)]\!],\ [\rho_i]_{i \in [\![1,n]\!]} \Vdash S(\tau, [\Diamond_1, \ldots, \Diamond_n]) \\ j \in [\![1, n]\!],\ \sigma \Vdash S(\rho_j, \Diamond) \end{array} \;\Bigg|\; a \Vdash_{g_a} H^{y:[\rho_j]}(\Gamma_a, y : [\rho_i]_{i \in [\![1,n]\!] \backslash j}; \sigma) \quad b \Vdash_{g_b} H^{x:[\tau]}(\Gamma_b; [\rho_i]_{i \in [\![1,n]\!]})}{a[y \backslash b] \Vdash_g H^{x:[\tau]}(\Gamma; \sigma)}\,\text{ES-CH}$$

$$\frac{\begin{array}{c} g \rightsquigarrow \mathsf{Sub}(g_a, g_b) \\ \Gamma = \Gamma_a + \Gamma_b + z : [\tau], \qquad \mathtt{fix}\ \ y \notin \mathrm{dom}(\Gamma) \\ n \in [\![0, \mathsf{sz}(\tau)]\!],\ \mathcal{M} \Vdash S(\tau, [\Diamond_1, \ldots, \Diamond_n]) \end{array} \;\Bigg|\; a \Vdash_{g_a} N(\Gamma_a, y : \mathcal{M}; \sigma) \quad b \Vdash_{g_b} H^{z:[\tau]}(\Gamma_b; \mathcal{M})}{a[y \backslash b] \Vdash_g N(\Gamma; \sigma)}\,\text{ES-N}$$

The Full Algorithm

$$\frac{g \rightsquigarrow \mathsf{Var}}{x \Vdash_g H^{x:[\sigma]}(\emptyset;\sigma)}\ \textsc{var} \qquad \frac{\begin{array}{c} g \rightsquigarrow \mathsf{Der}(g') \\ [\sigma] \Vdash S(\tau,\diamond) \end{array} \;\Big|\; a \Vdash_{g'} H^{x:[\tau]}(\Gamma;[\sigma])}{\mathsf{der}(a) \Vdash_g H^{x:[\tau]}(\Gamma;\sigma)}\ \textsc{dr}$$

$$\frac{g \rightsquigarrow g' \;\mid\; a \Vdash_{g'} H^{x:[\tau]}(\Gamma;\sigma)}{a \Vdash_g H^{x:[\tau]}(\Gamma;\sigma)}\ \textsc{h-h} \qquad \frac{\begin{array}{c} g \rightsquigarrow g' \\ \Gamma = \Gamma' + x:[\tau] \\ \sigma \Vdash S(\tau,\diamond) \end{array} \;\Big|\; a \Vdash_{g'} H^{x:[\tau]}(\Gamma';\sigma)}{a \Vdash_g N(\Gamma;\sigma)}\ \textsc{n-h} \qquad \frac{g \rightsquigarrow g' \;\mid\; a \Vdash_{g'} N(\Gamma;\sigma)}{a \Vdash_g N(\Gamma;\sigma)}\ \textsc{n-n}$$

$$\frac{\begin{array}{c} g \rightsquigarrow \mathsf{App}(g_a, g_b) \\ \Gamma = \Gamma_a + \Gamma_b \\ \mathcal{M} \Rightarrow \sigma \Vdash S(\tau, \diamond \Rightarrow \sigma) \end{array} \;\Bigg|\; a \Vdash_{g_a} H^{x:[\tau]}(\Gamma_a; \mathcal{M} \Rightarrow \sigma) \quad b \Vdash_{g_b} N(\Gamma_b; \mathcal{M})}{ab \Vdash_g H^{x:[\tau]}(\Gamma;\sigma)}\ \text{APP}$$

$$\frac{n \in [\![0, \mathrm{sz}(\rho)]\!],\ \mathcal{M} \Vdash S(\rho, [\diamond_1, \ldots, \diamond_n]) \;\mid\; a \Vdash_{g_a} H^{x:[\tau]}(\Gamma_a, y:\mathcal{M};\sigma) \quad b \Vdash_{g_b} H^{z:[\rho]}(\Gamma_b;\mathcal{M})}{a[y\backslash b] \Vdash_g H^{x:[\tau]}(\Gamma;\sigma)}\ \textsc{es-h}$$

$$\frac{\begin{array}{c} g \rightsquigarrow \mathsf{Sub}(g_a, g_b) \\ \Gamma = \Gamma_a + \Gamma_b, \qquad \mathtt{fix}\ \ y \notin \mathrm{dom}(\Gamma) \cup \{x\} \\ n \in [\![1, \mathrm{sz}(\tau)]\!],\ [\rho_i]_{i \in [\![1,n]\!]} \Vdash S(\tau, [\diamond_1, \ldots, \diamond_n]) \\ j \in [\![1, n]\!],\ \sigma \Vdash S(\rho_j, \diamond) \end{array} \;\Bigg|\; a \Vdash_{g_a} H^{y:[\rho_j]}(\Gamma_a, y:[\rho_i]_{i \in [\![1,n]\!] \setminus j};\sigma) \quad b \Vdash_{g_b} H^{x:[\tau]}(\Gamma_b; [\rho_i]_{i \in [\![1,n]\!]})}{a[y\backslash b] \Vdash_g H^{x:[\tau]}(\Gamma;\sigma)}\ \textsc{es-ch}$$

$$\frac{\begin{array}{c} g \rightsquigarrow \mathsf{Sub}(g_a, g_b) \\ \Gamma = \Gamma_a + \Gamma_b + z:[\tau], \qquad \mathtt{fix}\ \ y \notin \mathrm{dom}(\Gamma) \\ n \in [\![0, \mathrm{sz}(\tau)]\!],\ \mathcal{M} \Vdash S(\tau, [\diamond_1, \ldots, \diamond_n]) \end{array} \;\Bigg|\; a \Vdash_{g_a} N(\Gamma_a, y:\mathcal{M};\sigma) \quad b \Vdash_{g_b} H^{z:[\tau]}(\Gamma_b;\mathcal{M})}{a[y\backslash b] \Vdash_g N(\Gamma;\sigma)}\ \textsc{es-n}$$

The Full Algorithm

$$\dfrac{g \leadsto \mathsf{Var}}{x \Vdash_g H^{x:[\sigma]}(\emptyset;\sigma)}\ \text{VAR} \qquad \dfrac{\begin{matrix} g \leadsto \mathsf{Der}(g') \\ [\sigma] \Vvdash S(\tau,\diamond) \end{matrix} \;\Big|\; a \Vdash_{g'} H^{x:[\tau]}(\Gamma;[\sigma])}{\mathsf{der}(a) \Vdash_g H^{x:[\tau]}(\Gamma;\sigma)}\ \text{DR}$$

$$\dfrac{g \leadsto g' \;\mid\; a \Vdash_{g'} H^{x:[\tau]}(\Gamma;\sigma)}{a \Vdash_g H^{x:[\tau]}(\Gamma;\sigma)}\ \text{H-H} \qquad \dfrac{\begin{matrix} g \leadsto g' \\ \Gamma = \Gamma' + x:[\tau] \\ \sigma \Vvdash S(\tau,\diamond) \end{matrix} \;\Big|\; a \Vdash_{g'} H^{x:[\tau]}(\Gamma';\sigma)}{a \Vdash_g N(\Gamma;\sigma)}\ \text{N-H} \qquad \dfrac{g \leadsto g' \;\mid\; a \Vdash_{g'} N(\Gamma;\sigma)}{a \Vdash_g N(\Gamma;\sigma)}\ \text{N-N}$$

$$\dfrac{\begin{matrix} g \leadsto \mathsf{App}(g_a, g_b) \\ \Gamma = \Gamma_a + \Gamma_b \\ \mathcal{M} \Rightarrow \sigma \Vvdash S(\tau, \diamond \Rightarrow \sigma) \end{matrix} \;\Bigg|\; a \Vdash_{g_a} H^{x:[\tau]}(\Gamma_a; \mathcal{M} \Rightarrow \sigma) \quad b \Vdash_{g_b} N(\Gamma_b; \mathcal{M})}{ab \Vdash_g H^{x:[\tau]}(\Gamma;\sigma)}\ \text{APP}$$

$$\dfrac{n \in \llbracket 0, \mathrm{sz}(\rho)\rrbracket,\ \mathcal{M} \Vvdash S(\rho, [\diamond_1,\dots,\diamond_n]) \;\mid\; a \Vdash_{g_a} H^{x:[\tau]}(\Gamma_a, y:\mathcal{M};\sigma) \quad b \Vdash_{g_b} H^{z:[\rho]}(\Gamma_b;\mathcal{M})}{a[y\backslash b] \Vdash_g H^{x:[\tau]}(\Gamma;\sigma)}\ \text{ES-H}$$

$$\dfrac{\begin{matrix} g \leadsto \mathsf{Sub}(g_a, g_b) \\ \Gamma = \Gamma_a + \Gamma_b, \quad \mathtt{fix}\ \ y \notin \mathrm{dom}(\Gamma) \cup \{x\} \\ n \in \llbracket 1, \mathrm{sz}(\tau)\rrbracket,\ [\rho_i]_{i\in\llbracket 1,n\rrbracket} \Vvdash S(\tau, [\diamond_1,\dots,\diamond_n]) \\ j \in \llbracket 1, n\rrbracket,\ \sigma \Vvdash S(\rho_j, \diamond) \end{matrix} \;\Bigg|\; a \Vdash_{g_a} H^{y:[\rho_j]}(\Gamma_a, y:[\rho_i]_{i\in\llbracket 1,n\rrbracket\setminus j};\sigma) \quad b \Vdash_{g_b} H^{x:[\tau]}(\Gamma_b;[\rho_i]_{i\in\llbracket 1,n\rrbracket})}{a[y\backslash b] \Vdash_g H^{x:[\tau]}(\Gamma;\sigma)}\ \text{ES-CH}$$

$$\dfrac{\begin{matrix} g \leadsto \mathsf{Sub}(g_a, g_b) \\ \Gamma = \Gamma_a + \Gamma_b + z:[\tau], \quad \mathtt{fix}\ \ y \notin \mathrm{dom}(\Gamma) \\ n \in \llbracket 0, \mathrm{sz}(\tau)\rrbracket,\ \mathcal{M} \Vvdash S(\tau, [\diamond_1,\dots,\diamond_n]) \end{matrix} \;\Bigg|\; a \Vdash_{g_a} N(\Gamma_a, y:\mathcal{M};\sigma) \quad b \Vdash_{g_b} H^{z:[\tau]}(\Gamma_b;\mathcal{M})}{a[y\backslash b] \Vdash_g N(\Gamma;\sigma)}\ \text{ES-N}$$

The Full Algorithm

$$\frac{g \rightsquigarrow \mathsf{Var} \;\mid}{x \Vdash_g H^{x:[\sigma]}(\emptyset;\sigma)}\,\text{VAR}
\qquad
\frac{\begin{array}{c} g \rightsquigarrow \mathsf{Der}(g') \\ [\sigma] \Vvdash S(\tau, \diamond) \end{array} \;\Bigg|\; a \Vdash_{g'} H^{x:[\tau]}(\Gamma;[\sigma])}{\mathsf{der}(a) \Vdash_g H^{x:[\tau]}(\Gamma;\sigma)}\,\text{DR}$$

$$\frac{\begin{array}{c} g \rightsquigarrow \mathsf{App}(g_a, g_b) \\ \Gamma = \Gamma_a + \Gamma_b \\ \mathcal{M} \Rightarrow \sigma \Vvdash S(\tau, \diamond \Rightarrow \sigma) \end{array} \;\Bigg|\; a \Vdash_{g_a} H^{x:[\tau]}(\Gamma_a; \mathcal{M} \Rightarrow \sigma) \quad b \Vdash_{g_b} N(\Gamma_b; \mathcal{M})}{ab \Vdash_g H^{x:[\tau]}(\Gamma;\sigma)}\,\text{APP}$$

$$\frac{g \rightsquigarrow g' \;\mid\; a \Vdash_{g'} H^{x:[\tau]}(\Gamma;\sigma)}{a \Vdash_g H^{x:[\tau]}(\Gamma;\sigma)}\,\text{H-H}
\qquad
\frac{\begin{array}{c} g \rightsquigarrow g' \\ \Gamma = \Gamma' + x : [\tau] \\ \sigma \Vvdash S(\tau, \diamond) \end{array} \;\Bigg|\; a \Vdash_{g'} H^{x:[\tau]}(\Gamma';\sigma)}{a \Vdash_g N(\Gamma;\sigma)}\,\text{N-H}
\qquad
\frac{g \rightsquigarrow g' \;\mid\; a \Vdash_{g'} N(\Gamma;\sigma)}{a \Vdash_g N(\Gamma;\sigma)}\,\text{N-N}$$

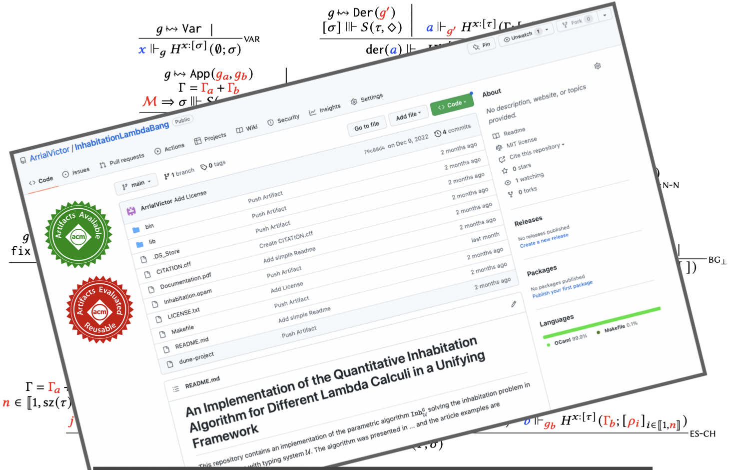
$$\frac{\begin{array}{c} g \rightsquigarrow \mathsf{Lam}(g') \\ \mathtt{fix}\;\; x \notin \mathsf{dom}(\Gamma) \end{array} \;\Bigg|\; a \Vdash_{g'} N(\Gamma, x : \mathcal{M}; \sigma)}{\lambda x.a \Vdash_g N(\Gamma; \mathcal{M} \Rightarrow \sigma)}\,\text{ABS}
\qquad
\frac{\begin{array}{c} g \rightsquigarrow \mathsf{Bng}(g') \\ I \neq \emptyset \\ \Gamma = +_{i \in I} \Gamma_i \end{array} \;\Bigg|\; \left(a_i \Vdash_{g'} N(\Gamma_i; \tau_i)\right)_{i \in I} \quad \uparrow_{i \in I} a_i}{! \bigvee_{i \in I} a_i \Vdash_g N(\Gamma; [\tau_i]_{i \in I})}\,\text{BG}
\qquad
\frac{g \rightsquigarrow \mathsf{Bng}(\bot) \;\mid}{!\bot \Vdash_g N(\emptyset; [\,])}\,\text{BG}_{\bot}$$

$$\frac{\begin{array}{c} g \rightsquigarrow \mathsf{Sub}(g_a, g_b) \\ \Gamma = \Gamma_a + \Gamma_b + z : [\rho], \qquad \mathtt{fix}\;\; y \notin \mathsf{dom}(\Gamma) \cup \{x\} \\ n \in \llbracket 0, \mathsf{sz}(\rho) \rrbracket, \;\; \mathcal{M} \Vvdash S(\rho, [\diamond_1, \ldots, \diamond_n]) \end{array} \;\Bigg|\; a \Vdash_{g_a} H^{x:[\tau]}(\Gamma_a, y : \mathcal{M}; \sigma) \quad b \Vdash_{g_b} H^{z:[\rho]}(\Gamma_b; \mathcal{M})}{a[y \backslash b] \Vdash_g H^{x:[\tau]}(\Gamma;\sigma)}\,\text{ES-H}$$

$$\frac{\begin{array}{c} g \rightsquigarrow \mathsf{Sub}(g_a, g_b) \\ \Gamma = \Gamma_a + \Gamma_b, \qquad \mathtt{fix}\;\; y \notin \mathsf{dom}(\Gamma) \cup \{x\} \\ n \in \llbracket 1, \mathsf{sz}(\tau) \rrbracket, \;\; [\rho_i]_{i \in \llbracket 1, n \rrbracket} \Vvdash S(\tau, [\diamond_1, \ldots, \diamond_n]) \\ j \in \llbracket 1, n \rrbracket, \;\; \sigma \Vvdash S(\rho_j, \diamond) \end{array} \;\Bigg|\; a \Vdash_{g_a} H^{y:[\rho_j]}(\Gamma_a, y : [\rho_i]_{i \in \llbracket 1, n \rrbracket \setminus j}; \sigma) \quad b \Vdash_{g_b} H^{x:[\tau]}(\Gamma_b; [\rho_i]_{i \in \llbracket 1, n \rrbracket})}{a[y \backslash b] \Vdash_g H^{x:[\tau]}(\Gamma;\sigma)}\,\text{ES-CH}$$

$$\frac{\begin{array}{c} g \rightsquigarrow \mathsf{Sub}(g_a, g_b) \\ \Gamma = \Gamma_a + \Gamma_b + z : [\tau], \qquad \mathtt{fix}\;\; y \notin \mathsf{dom}(\Gamma) \\ n \in \llbracket 0, \mathsf{sz}(\tau) \rrbracket, \;\; \mathcal{M} \Vvdash S(\tau, [\diamond_1, \ldots, \diamond_n]) \end{array} \;\Bigg|\; a \Vdash_{g_a} N(\Gamma_a, y : \mathcal{M}; \sigma) \quad b \Vdash_{g_b} H^{z:[\tau]}(\Gamma_b; \mathcal{M})}{a[y \backslash b] \Vdash_g N(\Gamma;\sigma)}\,\text{ES-N}$$

The Full Algorithm and its Implementation



github/ArrialVictor/InhabitationLambdaBang

Properties of the Inhabitation Algorithm

Non-deterministic algorithm

Non-deterministic algorithm

Theorem

- ✓ *The inhabitation algorithm terminates.*

Non-deterministic algorithm

Theorem

- *The inhabitation algorithm terminates.*
- *The algorithm is sound and complete (i.e. it exactly computes* $\mathtt{Basis}_{\mathcal{B}}(\Gamma, \sigma)$*).*

Properties of the Inhabitation Algorithm

Non-deterministic algorithm

Theorem

- *The inhabitation algorithm terminates.*
- *The algorithm is sound and complete (i.e. it exactly computes* $\texttt{Basis}_{\mathcal{B}}(\Gamma, \sigma)$*).*

More Ambitious Third Goal

- Decidability by **finding all inhabitants** in the $\boxed{\textbf{BANG}}$ IP.

Non-deterministic algorithm

Theorem

- *The inhabitation algorithm terminates.*
- *The algorithm is sound and complete (i.e. it exactly computes* $\mathtt{Basis}_{\mathcal{B}}(\Gamma, \sigma)$*).*

More Ambitious Third Goal

- Decidability by **finding all inhabitants** in the **BANG** IP.
- Decidability of the **NAME** and **VALUE** IP by **finding all inhabitants** from those of the **BANG** IP.
- Using generic properties so that other encodable models of computation can use these results.

Solving NAME Inhabitation - Standard Methology

Solving NAME Inhabitation - Standard Methology

Theorem ([BKR'14])

For any typing (Γ, σ), $\texttt{Basis}_{\mathcal{N}}(\Gamma, \sigma)$ ***exists***, *is* ***finite***, ***correct*** *and* ***complete***. NAME

Theorem ([BKR'14])

For any typing (Γ, σ), $\mathtt{Basis}_{\mathcal{N}}(\Gamma, \sigma)$ ***exists***, *is* ***finite***, ***correct*** *and* ***complete***. NAME

Built an algorithm computing $\mathtt{Basis}_{\mathcal{N}}(\Gamma, \sigma)$: **[BKR'14]**

$$\frac{\mathtt{a} \Vdash \mathtt{T}(\Gamma + \mathtt{x} : \mathtt{A}, \tau) \qquad \mathtt{x} \notin \mathtt{dom}(\Gamma)}{\lambda \mathtt{x}.\mathtt{a} \Vdash \mathtt{T}(\Gamma, \mathtt{A} \to \tau)}\ (\mathtt{Abs})$$

$$\frac{(\mathtt{a}_i \Vdash \mathtt{T}(\Gamma_i, \sigma_i))_{i \in I} \qquad \uparrow_{i \in I} \mathtt{a}_i}{\bigvee_{i \in I} \mathtt{a}_i \Vdash \mathtt{TI}(+_{i \in I} \Gamma_i, [\sigma_i]_{i \in I})}\ (\mathtt{Union})$$

$$\frac{\Gamma = \Gamma_1 + \Gamma_2 \qquad \mathtt{a} \Vdash \mathtt{H}^{\mathtt{x}:[\mathtt{A}_1 \to \ldots \mathtt{A}_n \to \mathtt{B} \to \tau]}(\Gamma_1, \mathtt{B} \to \tau) \qquad \mathtt{b} \Vdash \mathtt{TI}(\Gamma_2, \mathtt{B}) \qquad n \geq 0}{\mathtt{ab} \Vdash \mathtt{H}^{\mathtt{x}:[\mathtt{A}_1 \to \ldots \mathtt{A}_n \to \mathtt{B} \to \tau]}(\Gamma, \tau)}\ (\mathtt{Head}_{>0})$$

$$\frac{}{\mathtt{x} \Vdash \mathtt{H}^{\mathtt{x}:[\tau]}(\emptyset, \tau)}\ (\mathtt{Head}_0)$$

$$\frac{\mathtt{a} \Vdash \mathtt{H}^{\mathtt{x}:[\mathtt{A}_1 \to \ldots \mathtt{A}_n \to \tau]}(\Gamma, \tau)}{\mathtt{a} \Vdash \mathtt{T}(\Gamma + \mathtt{x} : [\mathtt{A}_1 \to \ldots \mathtt{A}_n \to \tau], \tau)}\ (\mathtt{Head})$$

Solving NAME Inhabitation : through BANG Inhabitation

The `Basis` is preserved by the embedding:

Theorem

NAME $\quad t \in \mathtt{Basis}_{\mathcal{N}}(\Gamma, \sigma)$

The `Basis` is preserved by the embedding:

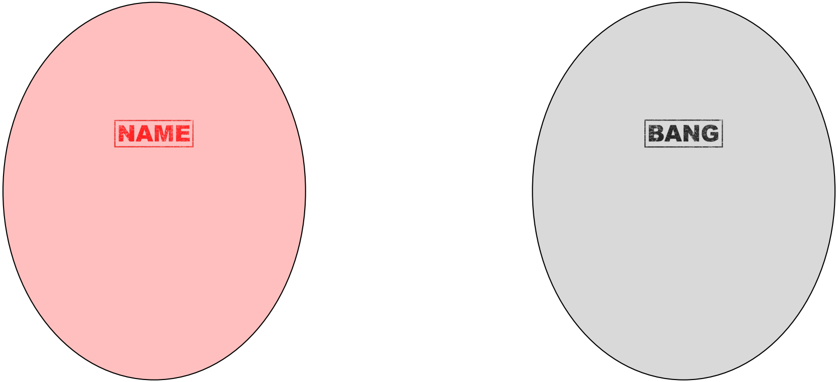
Theorem

NAME $\quad t \in \mathtt{Basis}_{\mathcal{N}}(\Gamma, \sigma) \iff t^{N} \in \mathtt{Basis}_{\mathcal{B}}(\Gamma, \sigma) \quad$ BANG

The `Basis` is preserved by the embedding:

Theorem

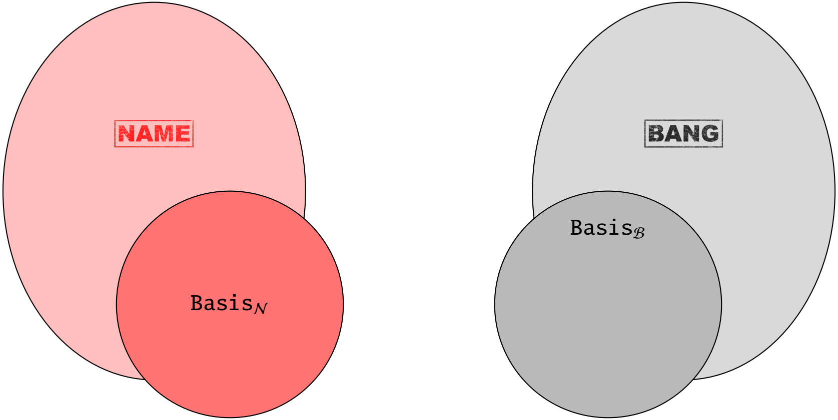
NAME $\quad t \in \mathtt{Basis}_{\mathcal{N}}(\Gamma, \sigma) \Leftrightarrow t^{N} \in \mathtt{Basis}_{\mathcal{B}}(\Gamma, \sigma) \quad$ BANG



The `Basis` is preserved by the embedding:

Theorem

NAME $\quad t \in \mathtt{Basis}_{\mathcal{N}}(\Gamma, \sigma) \Leftrightarrow t^{N} \in \mathtt{Basis}_{\mathcal{B}}(\Gamma, \sigma) \quad$ BANG

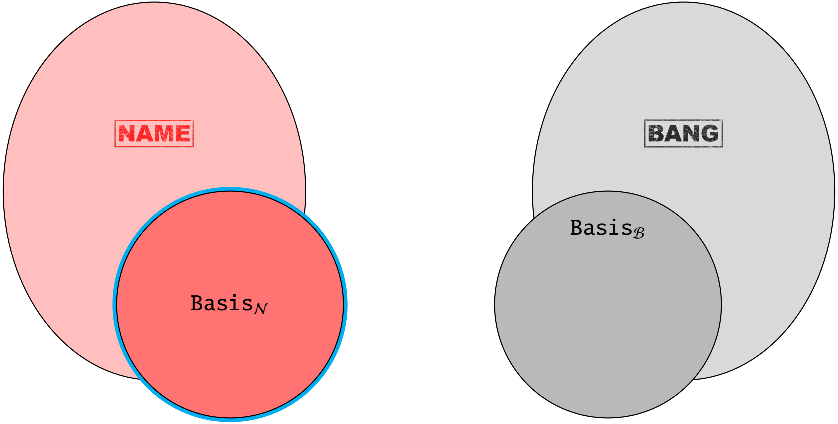


Solving NAME Inhabitation : through BANG Inhabitation

The `Basis` is preserved by the embedding:

Theorem

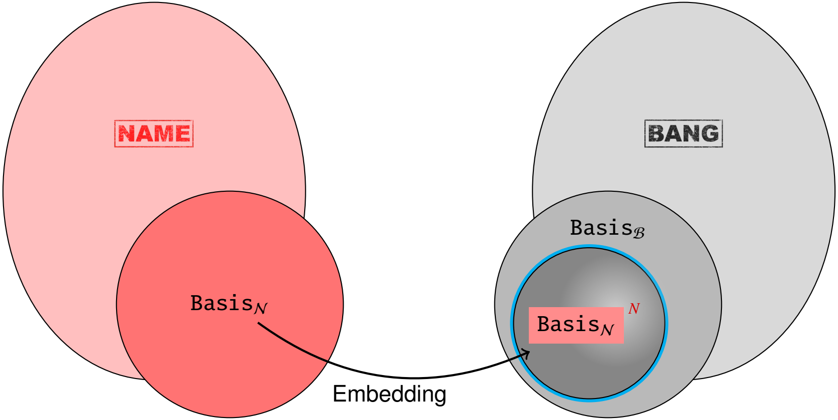
NAME $\quad t \in \texttt{Basis}_{\mathcal{N}}(\Gamma, \sigma) \Leftrightarrow t^{N} \in \texttt{Basis}_{\mathcal{B}}(\Gamma, \sigma) \quad$ BANG



The `Basis` is preserved by the embedding:

Theorem

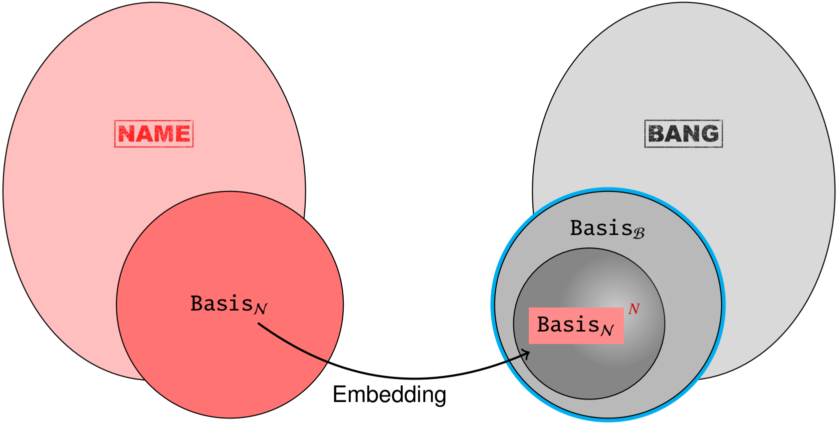
NAME $\quad t \in \texttt{Basis}_{\mathcal{N}}(\Gamma, \sigma) \quad \Leftrightarrow \quad t^{N} \in \texttt{Basis}_{\mathcal{B}}(\Gamma, \sigma) \quad$ BANG



The `Basis` is preserved by the embedding:

Theorem

NAME $\quad t \in \texttt{Basis}_{\mathcal{N}}(\Gamma, \sigma) \Leftrightarrow t^{N} \in \texttt{Basis}_{\mathcal{B}}(\Gamma, \sigma) \quad$ BANG

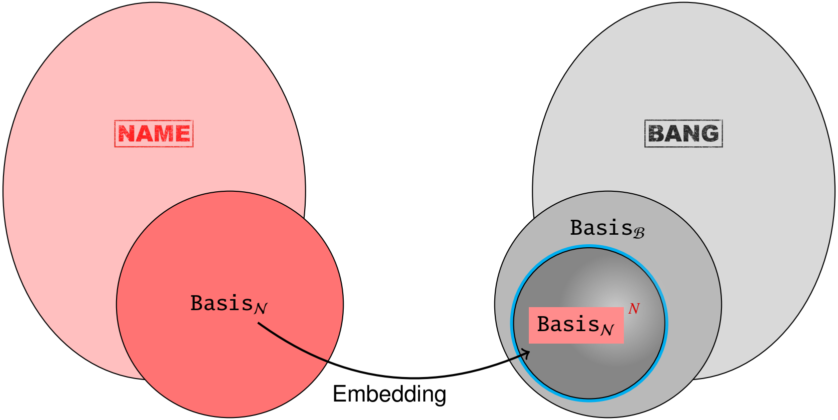


Solving NAME Inhabitation : through BANG Inhabitation

The `Basis` is preserved by the embedding:

Theorem

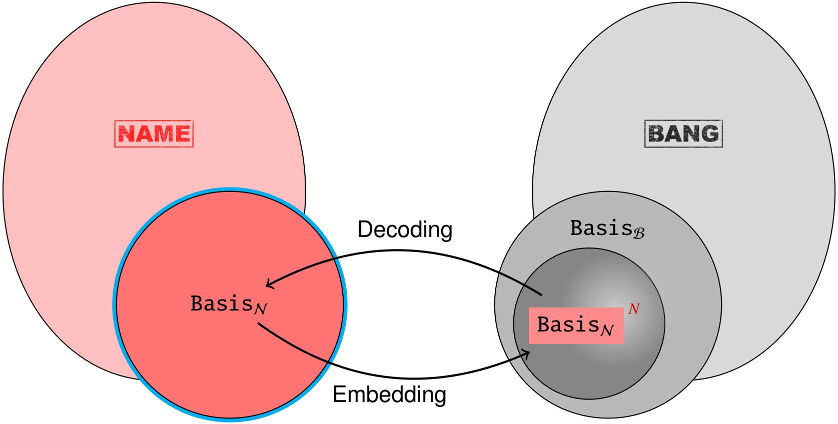
NAME $\quad t \in \mathtt{Basis}_{\mathcal{N}}(\Gamma, \sigma) \quad \Leftrightarrow \quad t^{N} \in \mathtt{Basis}_{\mathcal{B}}(\Gamma, \sigma) \quad$ BANG



The `Basis` is preserved by the embedding:

Theorem

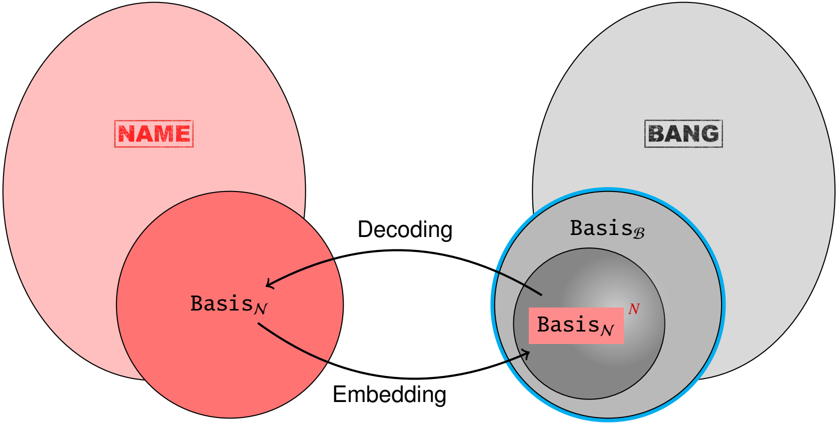
NAME $\quad t \in \mathtt{Basis}_{\mathcal{N}}(\Gamma, \sigma) \iff t^{N} \in \mathtt{Basis}_{\mathcal{B}}(\Gamma, \sigma) \quad$ BANG



The `Basis` is preserved by the embedding:

Theorem

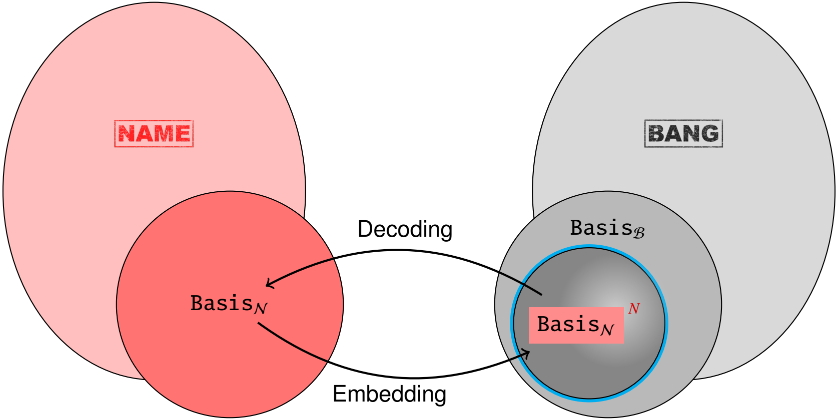
NAME $\quad t \in \mathtt{Basis}_{\mathcal{N}}(\Gamma, \sigma) \iff t^{N} \in \mathtt{Basis}_{\mathcal{B}}(\Gamma, \sigma) \quad$ BANG



The `Basis` is preserved by the embedding:

Theorem

NAME $\quad t \in \mathtt{Basis}_{\mathcal{N}}(\Gamma, \sigma) \iff t^{N} \in \mathtt{Basis}_{\mathcal{B}}(\Gamma, \sigma) \quad$ BANG

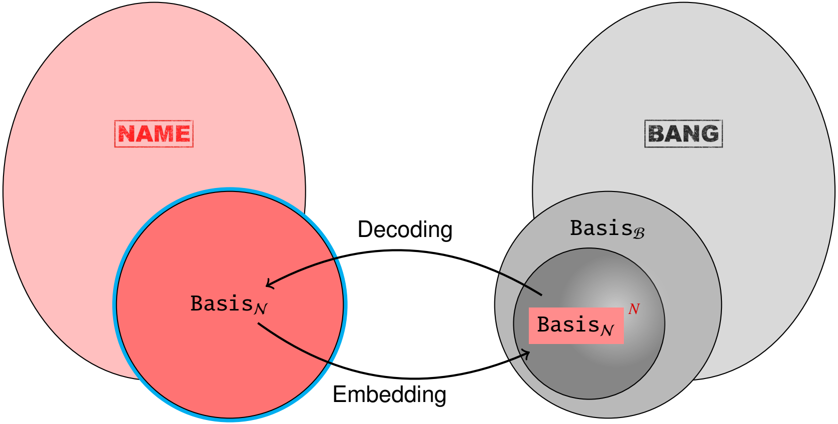


Solving NAME Inhabitation : through BANG Inhabitation

The `Basis` is preserved by the embedding:

Theorem

NAME $\quad t \in \mathtt{Basis}_{\mathcal{N}}(\Gamma, \sigma) \Leftrightarrow t^{N} \in \mathtt{Basis}_{\mathcal{B}}(\Gamma, \sigma) \quad$ BANG



Solving VALUE Inhabitation - Usual Methology

Solving **VALUE** Inhabitation - Usual Methology

Theorem

For any typing (Γ, σ), $\texttt{Basis}_{\mathcal{V}}(\Gamma, \sigma)$ ***exists***, *is* ***finite***, ***correct*** *and* ***complete***. **VALUE**

Solving VALUE Inhabitation - Usual Methology

Theorem

For any typing (Γ, σ)*,* $\mathtt{Basis}_{\mathcal{V}}(\Gamma, \sigma)$ ***exists****, is* ***finite****,* ***correct*** *and* ***complete****.* VALUE

Built an algorithm computing $\mathtt{Basis}_{\mathcal{V}}(\Gamma, \sigma)$:

Theorem

For any typing (Γ, σ), $\mathtt{Basis}_{\mathcal{V}}(\Gamma,\sigma)$ ***exists**, is **finite**, **correct** and **complete**.* **VALUE**

Built an algorithm computing $\mathtt{Basis}_{\mathcal{V}}(\Gamma,\sigma)$:

$$\frac{\mid}{x \Vdash H_{F}^{x:[\sigma]}(\emptyset;\sigma)}\,\text{VAR-FUN} \qquad \frac{I \neq \emptyset \quad \Gamma = x : [\sigma_i]_{i \in I} \mid}{x \Vdash N(\Gamma;[\sigma]_{i\in I})}\,\text{VAR-VAL} \qquad \frac{\mid}{\bot_{\mathsf{v}} \Vdash N(\emptyset;[\,])}\,\text{VAR}_{\bot}$$

$$\frac{\Gamma = \Gamma_1 + \Gamma_2 \quad [\mathcal{M} \Rightarrow \sigma] \Vvdash S(\tau,[\Diamond \Rightarrow \sigma]) \;\Big|\; a_1 \Vdash H_Q^{x:[\tau]}(\Gamma_1;[\mathcal{M} \Rightarrow \sigma]) \quad a_2 \Vdash N(\Gamma_2;\mathcal{M})}{a_1 a_2 \Vdash H_A^{x:[\tau]}(\Gamma;\sigma)}\,\text{APP}_Q \qquad \frac{\mid}{\lambda x.\bot \Vdash N(\emptyset;[\,])}\,\text{ABS}_{\bot}$$

$$\frac{\Gamma = \Gamma' + x : [\tau] \quad \sigma \Vvdash S(\tau,\Diamond) \;\Big|\; a \Vdash H_A^{x:[\tau]}(\Gamma';\sigma)}{a \Vdash N(\Gamma;\sigma)}\,\text{N-H}_A \qquad \frac{I \neq \emptyset \quad \Gamma = +_{i\in I}\Gamma_i \quad \mathtt{fix}\ x \in \mathrm{dom}(\Gamma) \;\Big|\; (\,a_i \Vdash N(\Gamma_i, x : \mathcal{M}_i;\sigma_i))_{i\in I} \quad \uparrow_{i \in I} a_i}{\lambda x. \bigvee_{i\in I} a_i \Vdash N(\Gamma;[\mathcal{M}_i \Rightarrow \sigma_i]_{i\in i})}\,\text{ABS}$$

$$\frac{\Gamma = \Gamma_a + \Gamma_b + z : [\rho], \quad \mathtt{fix}\ y \notin \mathrm{dom}(\Gamma) \cup \{x\} \quad n \in \llbracket 0, \mathrm{sz}(\rho) \rrbracket, \ \mathcal{M} \Vvdash S(\rho,[\Diamond_1,\ldots,\Diamond_n]) \;\Big|\; a \Vdash H_Q^{x:[\tau]}(\Gamma_a, y : \mathcal{M};\sigma) \quad b \Vdash H_A^{z:[\rho]}(\Gamma_b;\mathcal{M})}{a[y\backslash b] \Vdash H_Q^{x:[\tau]}(\Gamma;\sigma)}\,\text{ES-H}_Q$$

$$\frac{\Gamma = \Gamma_a + \Gamma_b, \quad \mathtt{fix}\ y \notin \mathrm{dom}(\Gamma) \cup \{x\} \quad n \in \llbracket 1, \mathrm{sz}(\tau) \rrbracket, \ [\rho_i]_{i \in \llbracket 1,n \rrbracket} \Vvdash S(\tau,[\Diamond_1,\ldots,\Diamond_n]) \quad j \in \llbracket 1, n \rrbracket, \ \sigma \Vvdash S(\rho_j,\Diamond) \;\Big|\; a \Vdash H_Q^{y:[\rho_j]}(\Gamma_a, y : [\rho_i]_{i \in \llbracket 1,n \rrbracket \backslash j};\sigma) \quad b \Vdash H_A^{x:[\tau]}(\Gamma_b;[\rho_i]_{i\in\llbracket 1,n \rrbracket})}{a[y\backslash b] \Vdash H_Q^{x:[\tau]}(\Gamma;\sigma)}\,\text{ES-CH}_Q$$

$$\frac{\Gamma = \Gamma_a + \Gamma_b + z : [\tau], \quad \mathtt{fix}\ y \notin \mathrm{dom}(\Gamma) \quad n \in \llbracket 0, \mathrm{sz}(\tau) \rrbracket, \ \mathcal{M} \Vdash S(\tau,[\Diamond_1,\ldots,\Diamond_n]) \;\Big|\; a \Vdash N(\Gamma_a, y : \mathcal{M};\sigma) \quad b \Vdash H_A^{z:[\tau]}(\Gamma_b;\mathcal{M})}{a[y\backslash b] \Vdash N(\Gamma;\sigma)}\,\text{ES-N}$$

The `Basis` is preserved by the embedding:

Theorem

VALUE $t \in \texttt{Basis}_{\mathcal{V}}(\Gamma, \sigma)$

Solving VALUE Inhabitation : through BANG Inhabitation

The `Basis` is preserved by the embedding:

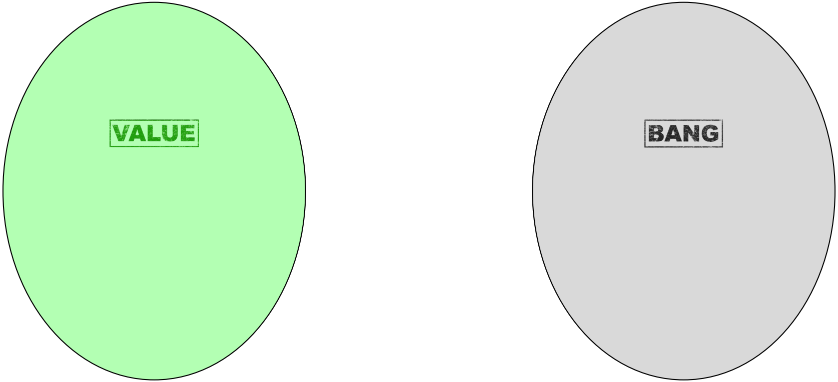
Theorem

VALUE $\quad t \in \mathtt{Basis}_{\mathcal{V}}(\Gamma, \sigma) \quad \Leftrightarrow \quad t^{V} \in \mathtt{Basis}_{\mathcal{B}}(\Gamma, \sigma)$ BANG

The `Basis` is preserved by the embedding:

Theorem

VALUE $\quad t \in \mathtt{Basis}_{\mathcal{V}}(\Gamma, \sigma) \Leftrightarrow t^{V} \in \mathtt{Basis}_{\mathcal{B}}(\Gamma, \sigma) \quad$ BANG

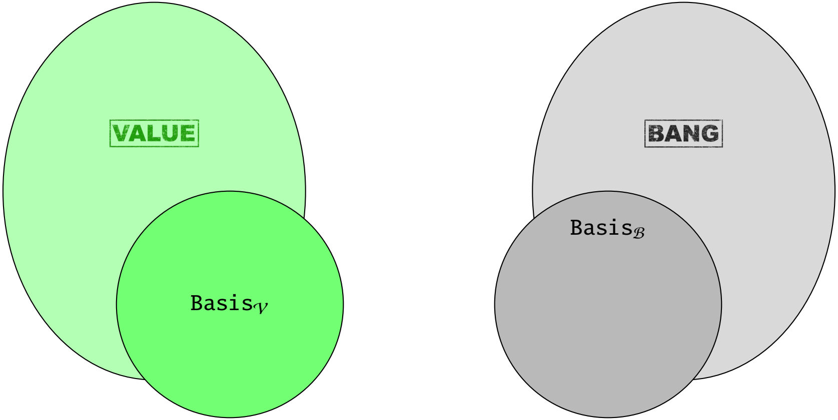


Solving VALUE Inhabitation : through BANG Inhabitation

The Basis is preserved by the embedding:

Theorem

VALUE $\quad t \in \mathtt{Basis}_{\mathcal{V}}(\Gamma, \sigma) \quad \Leftrightarrow \quad t^{V} \in \mathtt{Basis}_{\mathcal{B}}(\Gamma, \sigma) \quad$ BANG

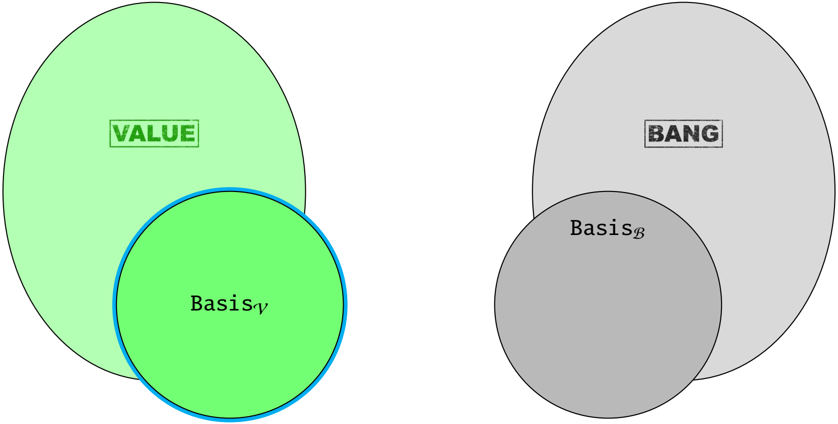


Solving VALUE Inhabitation : through BANG Inhabitation

The `Basis` is preserved by the embedding:

Theorem

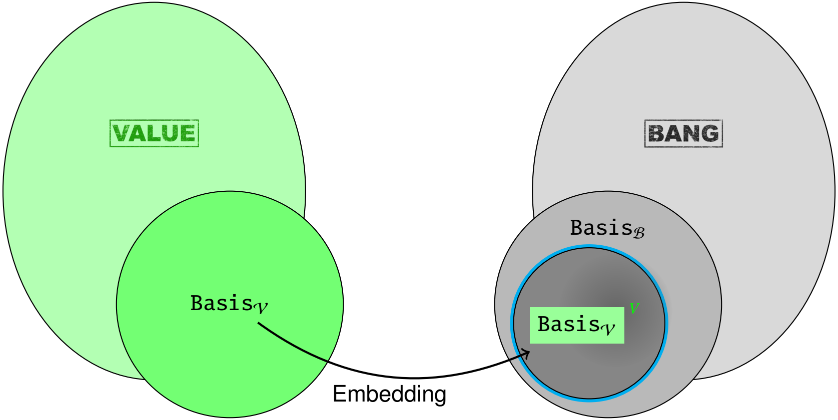
VALUE $\quad t \in \mathtt{Basis}_{\mathcal{V}}(\Gamma, \sigma) \Leftrightarrow t^{V} \in \mathtt{Basis}_{\mathcal{B}}(\Gamma, \sigma) \quad$ BANG



The Basis is preserved by the embedding:

Theorem

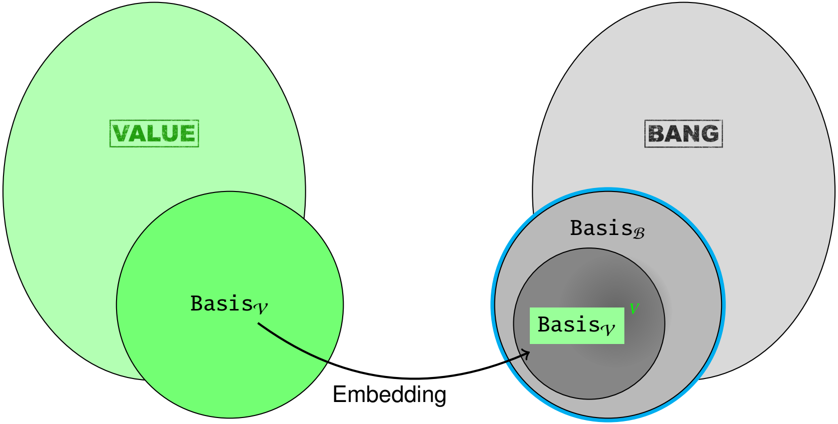
VALUE $\quad t \in \mathtt{Basis}_{\mathcal{V}}(\Gamma, \sigma) \Leftrightarrow t^{V} \in \mathtt{Basis}_{\mathcal{B}}(\Gamma, \sigma) \quad$ BANG



The `Basis` is preserved by the embedding:

Theorem

VALUE $\quad t \in \mathtt{Basis}_{\mathcal{V}}(\Gamma, \sigma) \Leftrightarrow t^{V} \in \mathtt{Basis}_{\mathcal{B}}(\Gamma, \sigma) \quad$ BANG

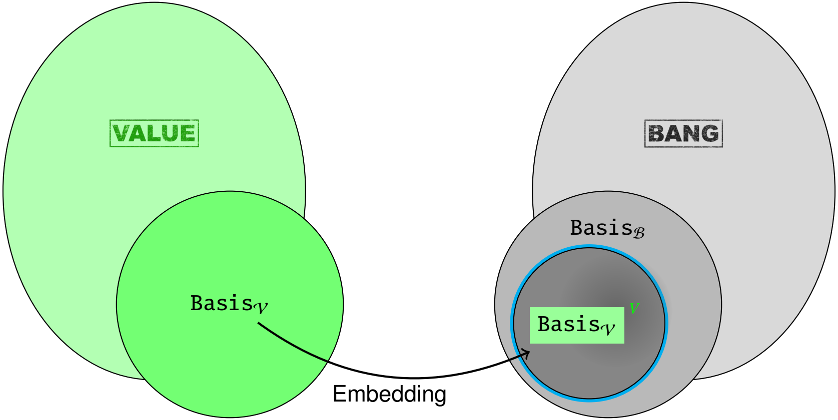


Solving VALUE Inhabitation : through BANG Inhabitation

The `Basis` is preserved by the embedding:

Theorem

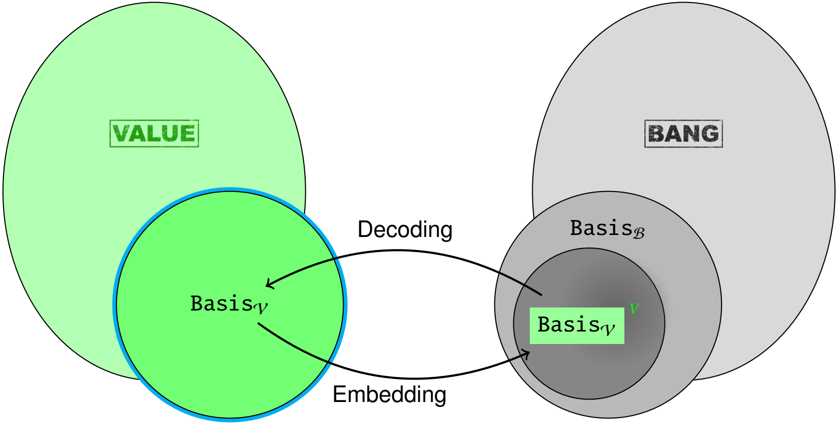
$$\text{VALUE} \qquad t \in \mathtt{Basis}_{\mathcal{V}}(\Gamma, \sigma) \quad \Leftrightarrow \quad t^{V} \in \mathtt{Basis}_{\mathcal{B}}(\Gamma, \sigma) \qquad \text{BANG}$$



The `Basis` is preserved by the embedding:

Theorem

VALUE $\quad t \in \mathtt{Basis}_{\mathcal{V}}(\Gamma, \sigma) \quad \Leftrightarrow \quad t^{V} \in \mathtt{Basis}_{\mathcal{B}}(\Gamma, \sigma) \quad$ BANG

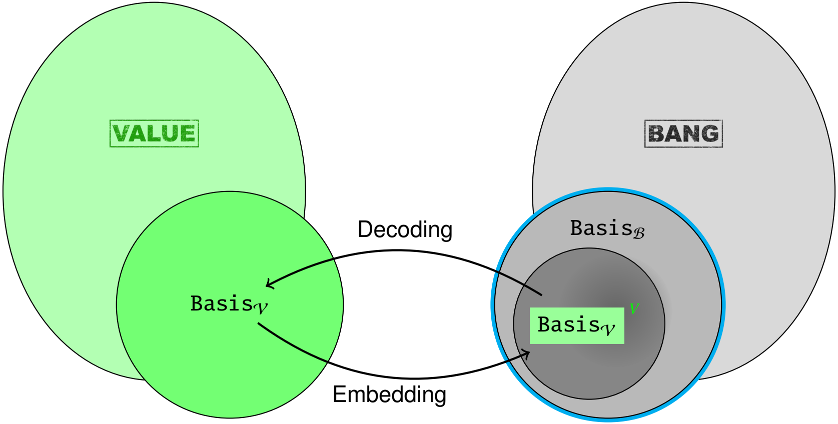


Solving VALUE Inhabitation : through BANG Inhabitation

The `Basis` is preserved by the embedding:

Theorem

VALUE $\quad t \in \texttt{Basis}_{\mathcal{V}}(\Gamma, \sigma) \quad \Leftrightarrow \quad t^{V} \in \texttt{Basis}_{\mathcal{B}}(\Gamma, \sigma) \quad$ BANG

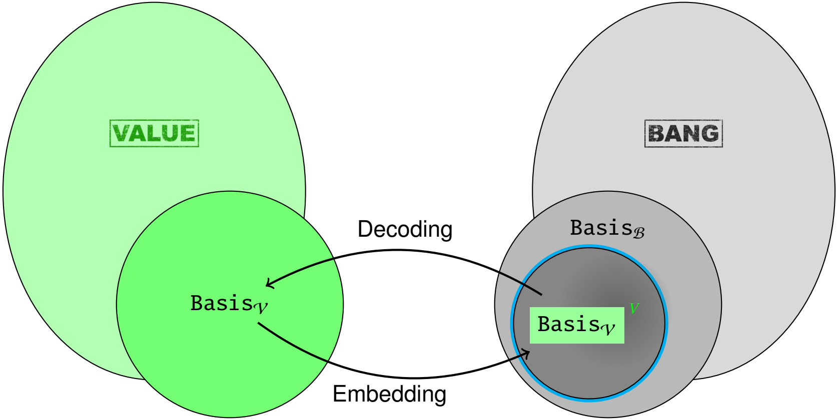


Solving VALUE Inhabitation : through BANG Inhabitation

The Basis is preserved by the embedding:

Theorem

VALUE $\quad t \in \texttt{Basis}_{\mathcal{V}}(\Gamma, \sigma) \quad \Leftrightarrow \quad t^{V} \in \texttt{Basis}_{\mathcal{B}}(\Gamma, \sigma) \quad$ BANG

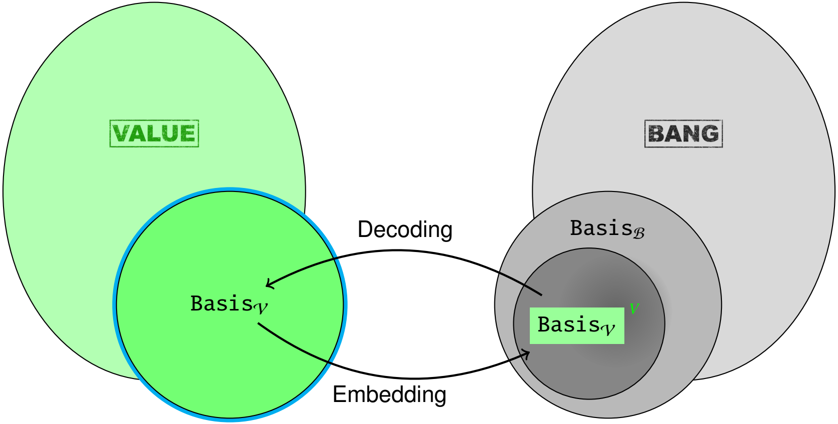


Solving VALUE Inhabitation : through BANG Inhabitation

The `Basis` is preserved by the embedding:

Theorem

VALUE $\quad t \in \mathtt{Basis}_{\mathcal{V}}(\Gamma, \sigma) \Leftrightarrow t^{V} \in \mathtt{Basis}_{\mathcal{B}}(\Gamma, \sigma) \quad$ BANG



Properties of the Indirect NAME and VALUE Algorithm

Properties of the Indirect NAME and VALUE Algorithm

Theorem

- *The inhabitation algorithm terminates.*
- *The algorithm is sound and complete*
  (i.e. it exactly computes $\mathtt{Basis}_{\mathcal{B}}(\Gamma, \sigma)$*).*

BANG

Properties of the Indirect NAME and VALUE Algorithm

Theorem

- *The inhabitation algorithm terminates.*
- *The algorithm is sound and complete*
  (i.e. it exactly computes $\mathtt{Basis}_{\mathcal{B}}(\Gamma, \sigma)$*).*

BANG

More Ambitious Third Goal

- Decidability by **finding all inhabitants** in the BANG IP.
- Decidability of the NAME and VALUE IP by **finding all inhabitants** from those of the BANG IP.
- Using generic properties so that other encodable models of computation can use these results.

Properties of the Indirect NAME and VALUE Algorithm

Theorem

- *The inhabitation algorithm terminates.*
- *The algorithm is sound and complete (i.e. it exactly computes* $\texttt{Basis}\ (\Gamma, \sigma)$*).*

BANG $\longrightarrow$ NAME VALUE

More Ambitious Third Goal

- Decidability by **finding all inhabitants** in the BANG IP.
- Decidability of the NAME and VALUE IP by **finding all inhabitants** from those of the BANG IP.
- Using generic properties so that other encodable models of computation can use these results.

Properties of the Indirect NAME and VALUE Algorithm

Theorem

- *The inhabitation algorithm terminates.*
- *The algorithm is sound and complete (i.e. it exactly computes* $\texttt{Basis}\ (\Gamma, \sigma)$*).*

More Ambitious Third Goal

- Decidability by **finding all inhabitants** in the BANG IP.
- Decidability of the NAME and VALUE IP by **finding all inhabitants** from those of the BANG IP.
- Using generic properties so that other encodable models of computation can use these results.

Properties of the Indirect NAME and VALUE Algorithm

Theorem

- *The inhabitation algorithm terminates.*
- *The algorithm is sound and complete (i.e. it exactly computes* $\texttt{Basis}\ (\Gamma, \sigma)$*).*

More Ambitious Third Goal

- Decidability by **finding all inhabitants** in the BANG IP.
- Decidability of the NAME and VALUE IP by **finding all inhabitants** from those of the BANG IP.
- Using generic properties so that other encodable models of computation can use these results.

Properties of the Indirect NAME and VALUE Algorithm

Theorem

- *The inhabitation algorithm terminates.*
- *The algorithm is sound and complete (i.e. it exactly computes* $\texttt{Basis}\ (\Gamma, \sigma)$*).*

More Ambitious Third Goal

- Decidability by **finding all inhabitants** in the BANG IP.
- Decidability of the NAME and VALUE IP by **finding all inhabitants** from those of the BANG IP.
- Using generic properties so that other encodable models of computation can use these results.

Properties of the Indirect NAME and VALUE Algorithm

Theorem

- *The inhabitation algorithm terminates.*
- *The algorithm is sound and complete*
  (i.e. it exactly computes $\texttt{Basis}\ (\Gamma, \sigma)$*).*

More Ambitious Third Goal

- Decidability by **finding all inhabitants** in the BANG IP.
- Decidability of the NAME and VALUE IP by **finding all inhabitants** from those of the BANG IP.
- Using generic properties so that other encodable models of computation can use these results.

Conclusion

Conclusion

Summary:

- Solving the generalized inhabitation problem
- A several-for-one deal: BANG NAME VALUE OTHERS
- An implementation: (`github/ArrialVictor/InhabitationLambdaBang`)

Conclusion

Summary:

- Solving the generalized inhabitation problem
- A several-for-one deal: BANG NAME VALUE OTHERS
- An implementation: (`github/ArrialVictor/InhabitationLambdaBang`)

Further questions and ongoing work:

- Solvability (for Different Calculi in a Unified Framework)
- Strengthening inhabitation for lambda-calculus with pattern matching **[BKRdR'21]**

Conclusion

Summary:

- Solving the generalized inhabitation problem
- A several-for-one deal: BANG NAME VALUE OTHERS
- An implementation: (`github/ArrialVictor/InhabitationLambdaBang`)

Further questions and ongoing work:

- Solvability (for Different Calculi in a Unified Framework)
- Strengthening inhabitation for lambda-calculus with pattern matching **[BKRdR'21]**

Thanks for your attention!