# Dissecting Denotational Semantics

## From the well-established $\mathcal{H}^*$ to the more recent quantitative coeffects

Flavien BREUVART

PPS, Paris Denis Diderot

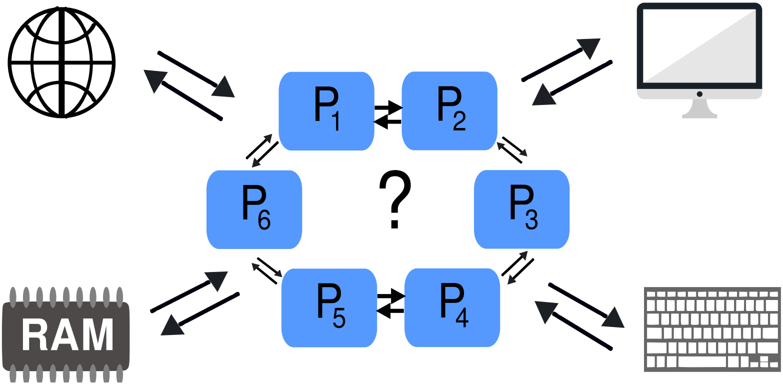
23 October 2015

# Quelles interactions entre un programme et son environnement?

Un programme est une suite d'instructions formelles...

# Quelles interactions entre un programme et son environnement?

Un programme est une suite d'instructions formelles...




... exécutées dans un environnement complexe...

# Quelles interactions entre un programme et son environnement?

Un programme est une suite d'instructions formelles...


... exécutées dans un environnement complexe...**Que l'on va abstraire**

# Quelles interactions entre un programme et son environnement?

Un programme est une suite d'instructions formelles...

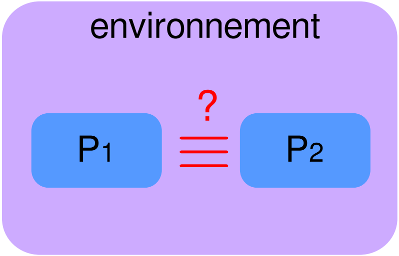
**Une thèse, Deux problèmes**

| Équivalence | Coeffets |
| --- | --- |
| Comparer deux programmes du point de vue utilisateur | Typer les besoins du programme sur l'environnement |

... exécutées dans un environnement complexe...**Que l'on va abstraire**

# Équivalence de programme

Deux programmes sont équivalents si l'on peut les interchanger librement dans **tout** environnement

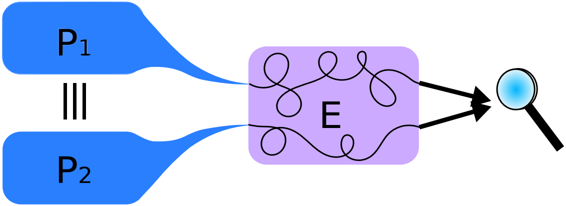



Intérêt pratique: Prouver/infirmer la correction d'une optimisation, d'une transformation de programme.

Intérêt théorique: Mieux comprendre le langage, quels sont les primitives?

# Équivalence observationnelle de programme

Deux programmes sont équivalents si l'on peut les interchanger librement dans **tout** environnement



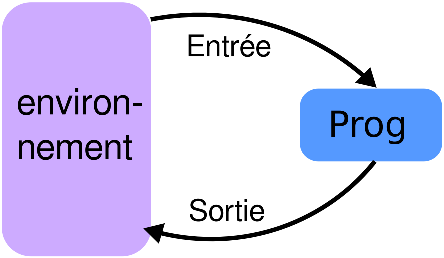

**$P_1$ observationnellement équivalent à $P_2$**

Pour tout environnement $E$

$$\texttt{Obs}[E(\!| P_1 |\!)] \;=\; \texttt{Obs}[E(\!| P_2 |\!)]$$

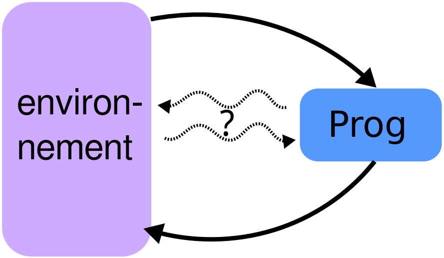
# Coeffets

Nécessité d'un paradigme: un programme est **une fonction**.



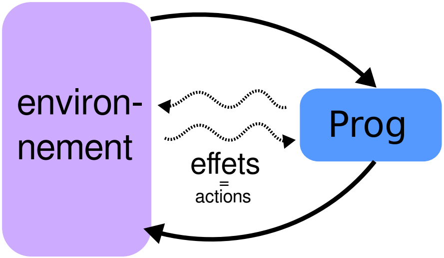
# Coeffets

Nécessité d'un paradigme: un programme est **une fonction**.
Dans le monde réel, un programme est rarement une routine entrée-sortie



# Coeffets

Nécessité d'un paradigme: un programme est **une fonction**.
Dans le monde réel, un programme est rarement une routine entrée-sortie

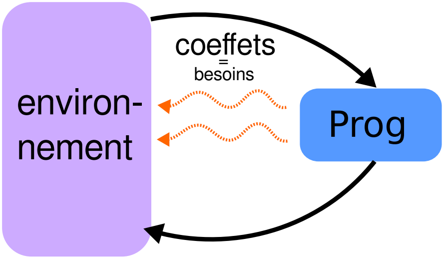



**Exemples d'effets/action/monad**

Affichage à l'écran, lecture du clavier, lecture mémoire...

# Coeffets

Nécessité d'un paradigme: un programme est **une fonction**.
Dans le monde réel, un programme est rarement une routine entrée-sortie

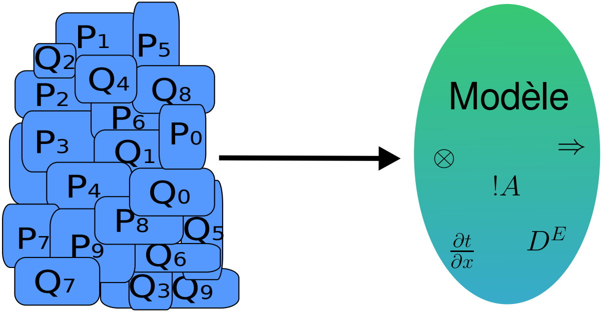



## Exemples de coeffets/besoins/comonad

- existence d'un fichier,
- borne sur la taille d'un tableau,
- borne sur le nombre de copies d'un argument,
- ordonancement (*scheduling*)...

# La sémantique: une vue d'ensemble

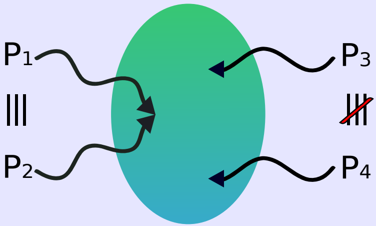
On ne s'intéresse pas à un programme en particulier,
mais à la structure du langage de programmation dans son ensemble.




Chaque programme est une **fonction de ce modèle**,
et leur interaction se modélise par des **opérations mathématiques**.

# Ma thèse

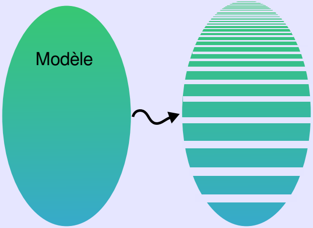
## Quels modèles pour l'équivalence observationnelle?




Chercher tous les modèles pour:

- mieux comprendre $\equiv$,
- développer des outils.

## Comprendre les coeffets via les modèles




Disséquer les modèles pour:

- comprendre les coeffets,
- comprendre les modèles usuels

# Preliminaries

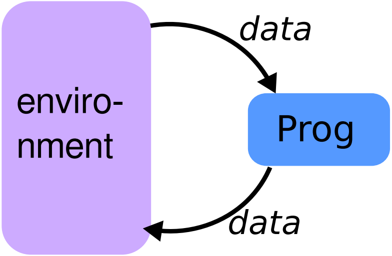
Je vais maintenant parler en anglais,
merci pour votre attention.

I will now switch to English,
thank you for your patience.

# $\lambda$-calculus: the core of functional programs

```
let rec map f = function
| []   -> []
| t::l -> (f t)::(map f l)
```

Everything is function...
Higher order functional program

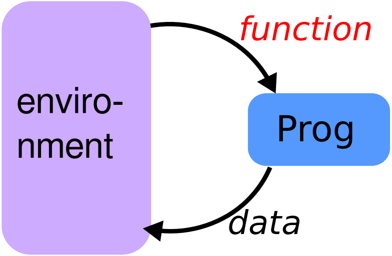


# λ-calculus: the core of functional programs

```
let rec map f = function
| []    -> []
| t::l -> (f t)::(map f l)
```

Everything is function...
Higher order functional program

functional input

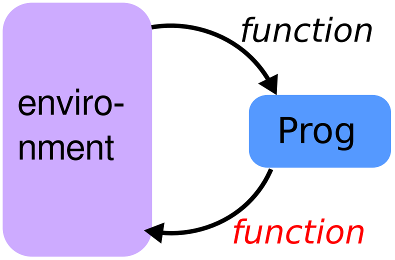


# $\lambda$-calculus: the core of functional programs

```
let rec map f = function
| []    -> []
| t::l -> (f t)::(map f l)
```

Everything is function...
Higher order functional program

functional input and functional output



# $\lambda$-calculus: the core of functional programs

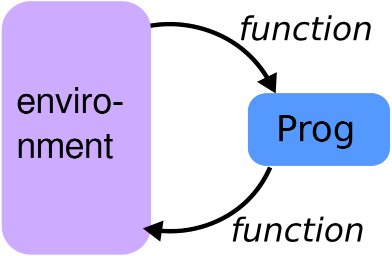
```
let rec map f = function
| []    -> []
| t::l -> (f t)::(map f l)
```

**Everything is function...**
**Higher order functional program**

functional input and functional output

abstraction: creating functions,

application: executing functions



# $\lambda$-calculus: the core of functional programs

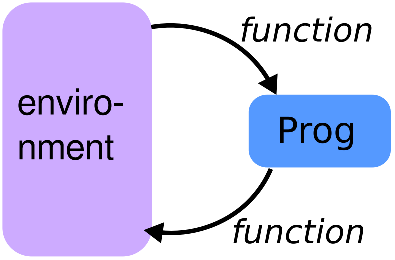
```
let rec map f = function
| []   -> []
| t::l -> (f t)::(map f l)
```

**Everything is function...**
**Higher order functional program**

functional input and functional output

abstraction: creating functions,

application: executing functions



# $\lambda$-calculus: the core of functional programs

```
let rec map f = function
| []    -> []
| t::l -> (f t)::(map f l)
```

**Everything is function...**
**Higher order functional program**

functional input and functional output

abstraction: creating functions,

application: executing functions

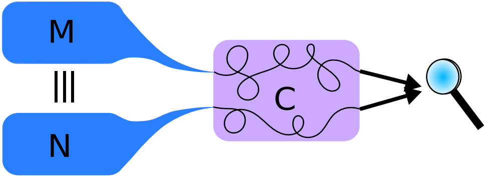
**...and function is everything**
**untyped $\lambda$-calculus: the core fragment**

Only functional inputs and only functional outputs

abstraction: $\lambda x.M$, application: $M\ N$, variable: $x$

No other constructors.

# Observational equivalence



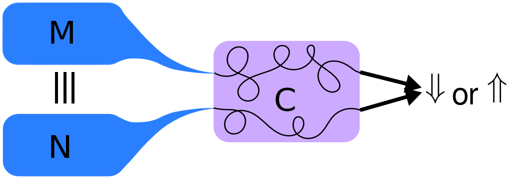

**$M$ observationally equivalent to $N$**

For all ~~environment~~ context $C$,

$$\texttt{Obs}[C(\!|M|\!)] \quad = \texttt{Obs}[C(\!|N|\!)]$$

Programmes: $\lambda$-terms
Environments: Contexts.

# Observational equivalence




**$M$ observationally equivalent to $N$**

For all ~~environment~~ context $C$,

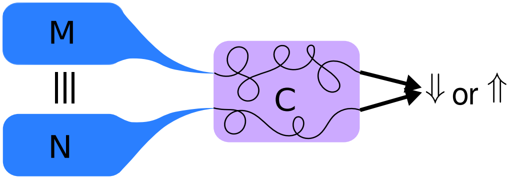
$$C(\!|M|\!)\Downarrow \;\Leftrightarrow\; C(\!|N|\!)\Downarrow$$

Programmes: $\lambda$-terms
Environments: Contexts.
Observation: Convergence for head reduction:

$M\Downarrow$ if $M \rightarrow \cdots \rightarrow \lambda x_1 ... x_m . y\ M_1 \cdots M_n$

$M\Uparrow$ if $M \rightarrow \cdots\cdots$

# Observational equivalence: $\mathcal{H}^*$




**$M$ observationally equivalent to $N$: $M \equiv_{\mathcal{H}*} N$**

For all ~~environment~~ context $C$,

$$C(\!|M|\!) \Downarrow \quad \Leftrightarrow \qquad C(\!|N|\!) \Downarrow$$

Programmes: $\lambda$-terms
Environments: Contexts.
Observation: Convergence for head reduction:
$M\Downarrow$ if $M \to \cdots \to \lambda x_1 ... x_m . y\ M_1 \cdots M_n$
$M\Uparrow$ if $M \to \cdots\cdots$

} $\mathcal{H}^*$

# Denotational semantics

## A model: How to see a language as a whole

| Programming language | $\xrightarrow{[\![-]\!]}$ | Abstract representation |
|---|---|---|
| data-structures | $\rightsquigarrow$ | sets/objects |
| programs | $\rightsquigarrow$ | functions/morphisms |
| primitives | $\rightsquigarrow$ | algebraic operations |
| evaluation | $\rightsquigarrow$ | equality |

## Full abstraction for $\mathcal{H}^*$

$$[\![M]\!] = [\![N]\!] \quad \text{iff} \quad M \equiv_{\mathcal{H}^*} N$$

# Looking for models for the untyped $\lambda$-calculus

Sets and functions $S^S \simeq S$ $\rightsquigarrow$ Impossible (Cantor theorem)

# Looking for models for the untyped $\lambda$-calculus

| | | |
|---|---|---|
| Sets and functions | $S^S \simeq S$ | $\rightsquigarrow$ Impossible (Cantor theorem) |
| Sets and cont. fun. | $\texttt{Cont}[S, S] \simeq S$ | $\rightsquigarrow$ What is continuous? |

# Looking for models for the untyped $\lambda$-calculus

| | | | | | |
|---|---|---|---|---|---|
| Sets and functions | $S^S$ | $\simeq$ | $S$ | $\leadsto$ | Impossible (Cantor theorem) |
| Sets and cont. fun. | $\texttt{Cont}[S, S]$ | $\simeq$ | $S$ | $\leadsto$ | What is continuous? |
| Domains | $\texttt{Scott}[D, D]$ | $\simeq$ | $D$ | $\leadsto$ | Too general |

# extensional K-models [Krivine1993]

| | | | | | |
|---|---|---|---|---|---|
| Sets and functions | $S^S$ | $\simeq$ | $S$ | $\rightsquigarrow$ | Impossible (Cantor theorem) |
| Sets and cont. fun. | $\texttt{Cont}[S,S]$ | $\simeq$ | $S$ | $\rightsquigarrow$ | What is continuous? |
| Domains | $\texttt{Scott}[D,D]$ | $\simeq$ | $D$ | $\rightsquigarrow$ | Too general |
| K-models | $\texttt{Scott}[\mathcal{I}(P),\mathcal{I}(P)]$ | $\simeq$ | $\mathcal{I}(P)$ | $\rightsquigarrow$ | What use? |
| $\Leftrightarrow$ | $(P \Rightarrow P)$ | $\simeq$ | $P$ | $\rightsquigarrow$ | compact/trace |

## K-Models

A preorder $(P, \leqslant)$ and a bijection "$\rightarrow$": $(P \Rightarrow P) \simeq P$, i.e.:

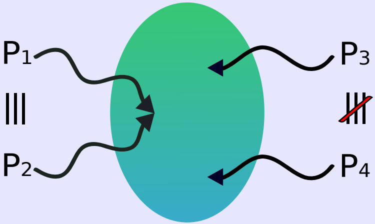
$$\forall \alpha \in P, \exists a \in \mathcal{P}_f(P), \exists \alpha' \in P,$$
$$\alpha = a \rightarrow \alpha'$$

## Operator $\Rightarrow$

$$P \Rightarrow P \ \simeq P$$
$$\mathcal{P}_f(P)^{op} \times P \ \simeq P$$

# My thesis

## Characterising K-models that are fully abstract for $\mathcal{H}^*$




Looking for models in order to:

- understand $\mathcal{H}^*$ more deeply,
- develop new tools for FA.

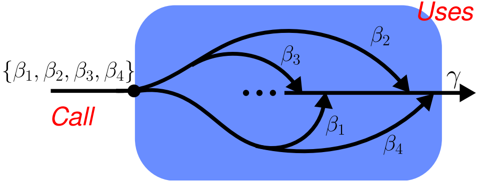
## Understanding $B_{\mathcal{S}}$LL via models of LL


Dissecting models of LL in order to:

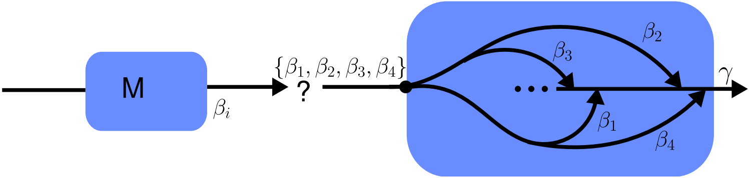
- model and understand $B_{\mathcal{S}}$LL,
- understand models of LL.

# Linear decomposition $B \Rightarrow C := !B \multimap C$

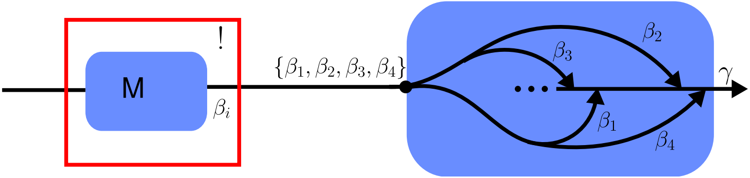
# Linear decomposition $B \Rightarrow C := !B \multimap C$



# Linear decomposition $B \Rightarrow C := !B \multimap C$



# Linear decomposition $B \Rightarrow C := !B \multimap C$

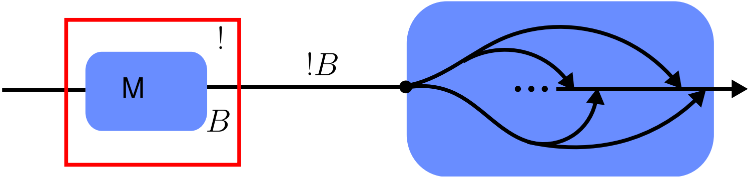



## Linear logic (LL)

$$\frac{\Gamma \vdash B}{\Gamma, !A \vdash B}\ \texttt{Weak} \qquad \frac{\Gamma, A \vdash B}{\Gamma, !A \vdash B}\ \texttt{Der}$$

$$\frac{\Gamma, !A, !A \vdash B}{\Gamma, !A \vdash B}\ \texttt{Contr} \qquad \frac{!A_1, \cdots, !A_n \vdash B}{!A_1, \cdots !A_n \vdash !B}\ \texttt{Prom}$$

# Linear decomposition $B \Rightarrow C := !B \multimap C$

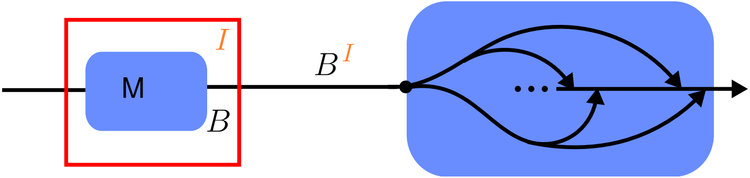



## Linear logic (LL)

$$\frac{\Gamma \vdash B}{\Gamma, !A \vdash B}\ \texttt{Weak} \qquad \frac{\Gamma, A \vdash B}{\Gamma, !A \vdash B}\ \texttt{Der}$$

$$\frac{\Gamma, !A, !A \vdash B}{\Gamma, !A \vdash B}\ \texttt{Contr} \qquad \frac{!A_1, \cdots, !A_n \vdash B}{!A_1, \cdots !A_n \vdash !B}\ \texttt{Prom}$$

# Coeffect decomposition $A \Rightarrow B := A^I \multimap B$

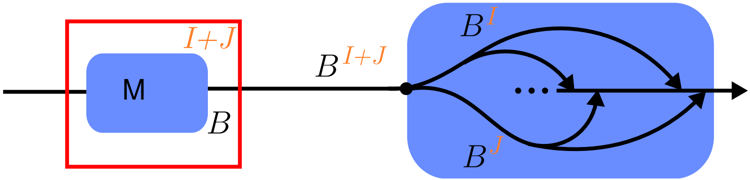



## Bounded logics ($\mathrm{B}_{\mathcal{S}}\mathrm{LL}$)

$$\frac{\Gamma \vdash B}{\Gamma, A^0 \vdash B}\ \texttt{Weak} \qquad \frac{\Gamma, A \vdash B}{\Gamma, A^1 \vdash B}\ \texttt{Der}$$

$$\frac{\Gamma, A^I, A^J \vdash B}{\Gamma, A^{I+J} \vdash B}\ \texttt{Contr} \qquad \frac{A_1^{I_1}, \cdots, !A_n^{I_n} \vdash B}{A_1^{I_1 * J}, \cdots A_n^{I_n * J} \vdash B^J}\ \texttt{Prom}$$

# Coeffect decomposition $A \Rightarrow B := A^I \multimap B$

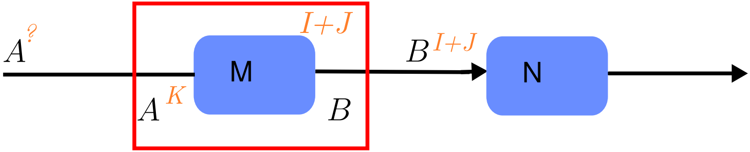



## Bounded logics ($B_{\mathcal{S}}LL$)

$$\frac{\Gamma \vdash B}{\Gamma, A^0 \vdash B}\ \texttt{Weak} \qquad \frac{\Gamma, A \vdash B}{\Gamma, A^1 \vdash B}\ \texttt{Der}$$

$$\frac{\Gamma, A^I, A^J \vdash B}{\Gamma, A^{I+J} \vdash B}\ \texttt{Contr} \qquad \frac{A_1^{I_1}, \cdots, !A_n^{I_n} \vdash B}{A_1^{I_1 * J}, \cdots A_n^{I_n * J} \vdash B^J}\ \texttt{Prom}$$

# Coeffect decomposition $A \Rightarrow B := A^I \multimap B$

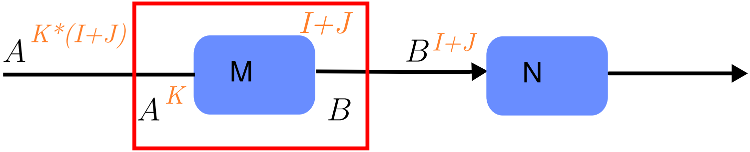



## Bounded logics ($B_{\mathcal{S}}LL$)

$$\frac{\Gamma \vdash B}{\Gamma, A^0 \vdash B}\ \texttt{Weak} \qquad \frac{\Gamma, A \vdash B}{\Gamma, A^1 \vdash B}\ \texttt{Der}$$

$$\frac{\Gamma, A^I, A^J \vdash B}{\Gamma, A^{I+J} \vdash B}\ \texttt{Contr} \qquad \frac{A_1^{I_1}, \cdots, !A_n^{I_n} \vdash B}{A_1^{I_1 * J}, \cdots A_n^{I_n * J} \vdash B^J}\ \texttt{Prom}$$

# Coeffect decomposition $A \Rightarrow B := A^I \multimap B$

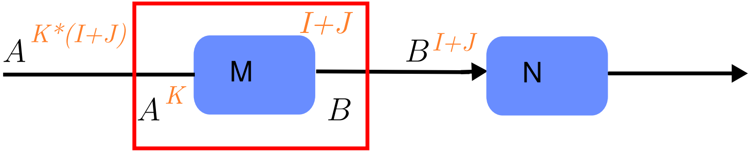



## Bounded logics ($B_{\mathcal{S}}LL$)

$$\frac{\Gamma \vdash B}{\Gamma, A^0 \vdash B}\ \texttt{Weak} \qquad \frac{\Gamma, A \vdash B}{\Gamma, A^1 \vdash B}\ \texttt{Der}$$

$$\frac{\Gamma, A^I, A^J \vdash B}{\Gamma, A^{I+J} \vdash B}\ \texttt{Contr} \qquad \frac{A_1^{I_1}, \cdots, !A_n^{I_n} \vdash B}{A_1^{I_1 * J}, \cdots A_n^{I_n * J} \vdash B^J}\ \texttt{Prom}$$

# Coeffect decomposition $A \Rightarrow B := A^I \multimap B$




## Bounded logics ($B_{\mathcal{S}}LL$)

Coeffects are represented by semirings $(\mathcal{S}, +, 0, *, 1)$

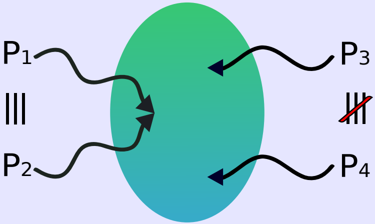
$$\frac{\Gamma \vdash B}{\Gamma, A^0 \vdash B}\ \texttt{Weak} \qquad \frac{\Gamma, A \vdash B}{\Gamma, A^1 \vdash B}\ \texttt{Der}$$

$$\frac{\Gamma, A^I, A^J \vdash B}{\Gamma, A^{I+J} \vdash B}\ \texttt{Contr} \qquad \frac{A_1^{I_1}, \cdots, !A_n^{I_n} \vdash B}{A_1^{I_1 * J}, \cdots A_n^{I_n * J} \vdash B^J}\ \texttt{Prom}$$

One $B_{\mathcal{S}}LL$ per semiring

# My thesis

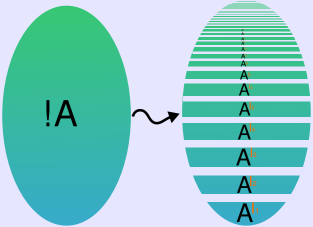
## Characterising K-models that are fully abstract for $\mathcal{H}^*$




Looking for models in order to:

- understand $\mathcal{H}^*$ more deeply,
- develop new tools for FA.

## Understanding $B_{\mathcal{S}}LL$ via models of LL




Dissecting models of LL in order to:

- model and understand $B_{\mathcal{S}}LL$,
- understand models of LL.

# The result

**The result (informally):**

A K-model $P$ is fully abstract for $\mathcal{H}^*$ iff $P$ is hyperimmune,

Non-well-founded chains can exist in $P$ but *cannot be accessible* to terms.

$$\alpha_1 = a_{1,1} \to \cdots a_{1,i_1} \cdots \to a_{1,g(1)} \to \alpha'_1$$
$$\Cup$$
$$\alpha_2 = a_{2,1} \to \cdots a_{2,i_2} \cdots \to a_{2,g(2)} \to \alpha'_2$$
$$\Cup$$
$$\alpha_3 = a_{3,1} \to \cdots a_{3,i_3} \cdots \to a_{3,g(3)} \to \alpha'_3$$
$$\ddots$$

**Hyperimmune models**

A K-model $P$ is *hyperimmune* when, $\forall (\alpha_n)_{n\geqslant 0}, \forall g : \mathbb{N} \to \mathbb{N}$,

if for all $n$, $\quad \alpha_n = a_{n,1}\to\cdots\to a_{n,g(n)}\to\alpha'_n \quad$ and $\quad \alpha_{n+1} \in \bigcup_{k\leqslant g(n)} a_{n,k}$

then $g$ is not recursive

# $\mathcal{D}_\infty$ is FA for $\mathcal{H}^*$ [Hyland76,Wadsworth76]

## Scott's $\mathcal{D}_\infty$ :

$$\mathcal{D}_0 = \{*\}$$

$$\mathcal{D}_{n+1} = \mathcal{D}_n \cup (\mathcal{A}_f(\mathcal{D}_n) \times \mathcal{D}_n) - \{(\emptyset, *)\}$$

$$\mathcal{D}_\infty = \bigcup_n \mathcal{D}_n$$

$$* = \emptyset \to *$$

$$(a, \alpha) = a \to \alpha$$

## Hyperimmunity of $\mathcal{D}_\infty$

$$\mathcal{D}_n \supseteq \mathcal{D}_{n-1} \supseteq \cdots \supseteq \mathcal{D}_0$$

$$(a, \alpha) = a \to \alpha$$

$$\uplus$$

$$(b, \beta) = b \to \beta$$

$$\ddots$$

$$* = \emptyset \to *$$

# $\mathcal{D}^*_\infty$ is not FA for $\mathcal{H}^*$ [CDZ1987]

## Coppo,Dezani&Zacchi's $\mathcal{D}^*_\infty$ (or Norm):

$\mathcal{D}^*_0 = \{p, q\}$ $\quad q = \{p\} \to q$ $\quad p = \{q\} \to p$

## Non-hyperimmunity of $\mathcal{D}^*_\infty$

$$
\begin{array}{l}
p = \{q\} \to p \\
\quad\ \Cup \\
\quad\ q = \{p\} \to q \\
\qquad\quad \Cup \\
\qquad\quad p = \{q\} \to p \\
\qquad\qquad\qquad \ddots
\end{array}
$$

$$\alpha_{2n+1} = p$$
$$\alpha_{2n} = q$$
$$g(n) = 1$$

Actually, $\mathcal{D}^*_\infty$ is fully abstract for $\mathcal{H}$

# Previous works

[Hyl76,Wad76]

**Full abstraction of $\mathcal{D}_\infty$**

[Milner1977]

**Milner's theorem for PCF**

There is a unique domain fully abstract for PCF (continuous, extensional and up to iso.)

[Gouy1995]

**And for the pure $\lambda$-calculus? (head reduction)**

There exist many non-isomorphic models fully abstract for $\mathcal{H}^*$

[Manzonetto2009]

**Sufficient condition**

Any well-stratified model, *i.e.*, st

$$|f|_{k+1}(|a|_k) = |f(a)|_k \qquad |f|_0(\perp) = |f(\perp)|_0$$

is fully abstract for $\mathcal{H}^*$.

# The result

**Theorem** (Informal)

A K-model $P$ is fully abstract for $\mathcal{H}^*$ iff $P$ is hyperimmune,

The actual theorem splits in two equivalent forms:

**Theorem** (semantic)

For any K-model $P$ that

- and is approximable:

then $P$ FA iff $P$ hyperimmune.

$\rightsquigarrow$ semantic proof

**Theorem** (syntactic)

For any K-model $P$ that

- and is sensible for $\boldsymbol{\Lambda}_{\tau(P)}$:

then $P$ FA iff $P$ hyperimmune.

$\rightsquigarrow$ syntactic proof

# Semantic proof

## Approximability of $[\![\_]\!]$

$$[\![M]\!] \quad = \quad [\![\mathbf{BT}(M)]\!]_{ind} := \bigcup_{\substack{T \subseteq \mathbf{BT}(M) \\ T \text{ finite}}} [\![T]\!]$$

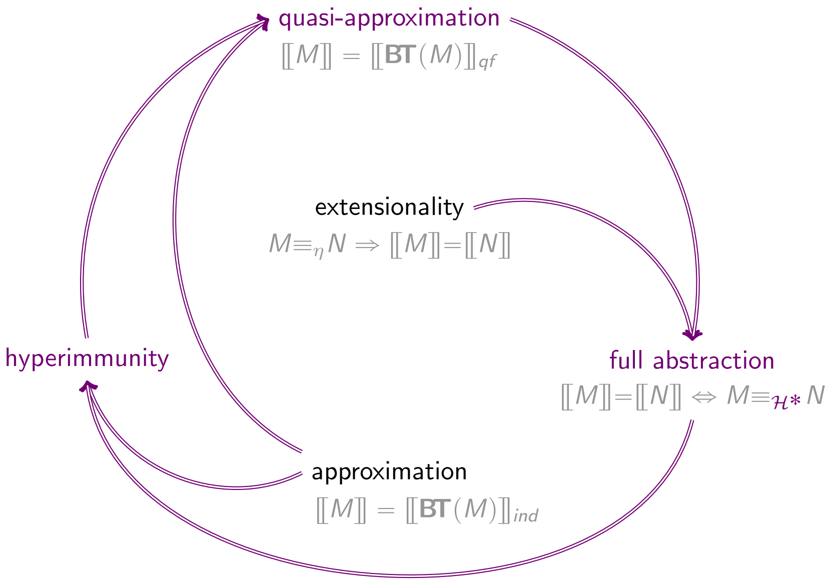
## Quasi-approximability of $[\![\_]\!]$

$$[\![M]\!] \quad = \quad [\![\mathbf{BT}(M)]\!]_{qf} := \bigcup_{\substack{T \subseteq \mathbf{BT}(M) \\ T \text{ quasi-finite}}} [\![T]\!]$$

Key properties of quasi-finite böhm trees:

- closed by $\supseteq$ and $\equiv_{\eta\infty}$,
- $T \subseteq \mathbf{BT}(M)$ and $T$ quasi-finite imply $T$ recursive.

# Semantic proof



# The result

**Theorem** (Informal)

A K-model $P$ is fully abstract for $\mathcal{H}^*$ iff $P$ is hyperimmune,

The actual theorem splits in two equivalent forms:

**Theorem** (semantic)

For any K-model $P$ that

- and is approximable:

then $P$ FA iff $P$ hyperimmune.

$\rightsquigarrow$ semantic proof

**Theorem** (syntactic)

For any K-model $P$ that

- and is sensible for $\mathbf{\Lambda}_{\tau(P)}$:

then $P$ FA iff $P$ hyperimmune.

$\rightsquigarrow$ syntactic proof

# Syntactic proof

**Sensibility for $\mathbf{\Lambda}_{\tau(P)}$**

$$\forall \alpha \in P \qquad \tau_\alpha(M)\Downarrow \quad \Leftrightarrow \quad \alpha \in [\![M]\!]$$

$$
\begin{array}{llll}
[\![M]\!] \neq [\![N]\!] & \Longleftrightarrow \ \exists \alpha \in P, & \alpha \in [\![M]\!] - [\![N]\!] & \text{or conv.} \\
& \Longleftrightarrow \ \exists \alpha \in P, & \tau_\alpha(M)\Downarrow \text{ and } \tau_\alpha(N)\Uparrow & \text{or conv.} \\
& \overset{(*)}{\Longleftrightarrow} \ \exists C \in \Lambda^{(\!|\cdot|\!)}, & C(\!|M|\!)\Downarrow \text{ and } C(\!|N|\!)\Uparrow & \text{or conv.} \\
& \Longleftrightarrow & M \not\equiv_{\mathcal{H}^*} N &
\end{array}
$$

$(*)$ By induction on $\tau_\alpha(M)\Downarrow$ using:

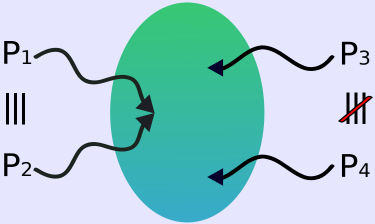
**Key lemma** $(M = \lambda x.x)$

$$(\exists N, (*) \text{ is false for } M = \lambda x.x) \quad \Rightarrow \quad P \text{ not hyperimmune}$$

Proof: co-inductively construct $(\alpha_n)_n$ by unfolding $\tau_\alpha(N)\Uparrow$.

# My thesis

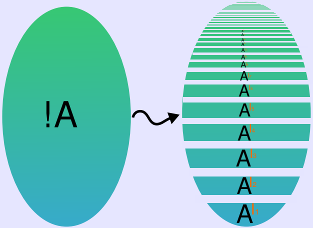
## Characterising K-models that are fully abstract for $\mathcal{H}^*$




Looking for models in order to:

- understand $\mathcal{H}^*$ more deeply,
- develop new tools for FA.

## Understanding $B_{\mathcal{S}}LL$ via models of LL




Dissecting models of LL in order to:

- model and understand $B_{\mathcal{S}}LL$,
- understand models of LL.

# From semantics to syntax

$\mathcal{L}$: linear category

## $\mathcal{L}[!1, 1]$ forms a left-semiring

$$0 := \mathtt{w}_{\mathbb{1}} \qquad 1 := \mathtt{d}_{\mathbb{1}}$$

$$I + J := \ !\mathbb{1} \xrightarrow{\mathtt{c}_{\mathbb{1}}} !\mathbb{1} \otimes !\mathbb{1} \xrightarrow{I \otimes J} \mathbb{1} \otimes \mathbb{1} \xrightarrow{\lambda} \mathbb{1}$$

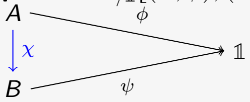
$$I \cdot J := \ !\mathbb{1} \xrightarrow{\mathtt{p}_{\mathbb{1}}} !!\mathbb{1} \xrightarrow{!I} !\mathbb{1} \xrightarrow{J} \mathbb{1}$$

## The slice $\mathcal{L}_{/\mathbb{1}}$

**objects:** $(A, \phi)$ with

$$A \in \mathcal{L} \qquad \phi \in \mathcal{L}[A, \mathbb{1}],$$

**morphisms:** $\mathcal{L}_{/\mathbb{1}}[(A, \phi), (B, \psi)]$




## Theorem

The sliced category $\mathcal{L}_{/\mathbb{1}}$ is a model of B$_{\tilde{\mathcal{L}}[!1,1]}$LL:

$$\phi^I := \quad !A \xrightarrow{!\phi} !\mathbb{1} \xrightarrow{I} \mathbb{1}$$

# From syntax to semantics

$\mathcal{L}$: linear category, $\mathcal{S}$ : semiring

## $\mathcal{S}$-Stratification of a model $\mathcal{L}$ of LL

$$(\_)^{\_} : \mathcal{L} \times \mathcal{S} \to \mathcal{L} \qquad \varrho_{A,J} : !A \Longrightarrow A^J$$

Such that:

$\varrho_{A,J}$ is an epimorphism,

plus 6 commuting diagrams

## Def epimorphism

$$A \overset{epi}{\to} B \overset{\phi}{\to} C$$
$$= A \overset{epi}{\to} B \overset{\psi}{\to} C$$

$$\Rightarrow$$

$$\phi = \psi$$

## Theorem

The $\mathcal{S}$-stratification of a model of LL yields a model of $\mathrm{B}_{\mathcal{S}}\mathrm{LL}$

Applications in Rel, ScottL and Coh.

# From syntax to semantics

$\mathcal{L}$: linear category, $\mathcal{S}$ : semiring

## $\mathcal{S}$-Stratification of a model $\mathcal{L}$ of LL

$(\_)^{-} : \mathcal{L} \times \mathcal{S} \to \mathcal{L}$ $\varrho_{A,J} : !A \Longrightarrow A^J$

Such that:

$\varrho_{A,J}$ is an epimorphism,

plus 6 commuting diagrams

## Def epimorphism

$$A \stackrel{epi}{\to} B \stackrel{\phi}{\to} C = A \stackrel{epi}{\to} B \stackrel{\psi}{\to} C$$

$$\Rightarrow$$

$$\phi = \psi$$

## Theorem

The $\mathcal{S}$-stratification of a model of LL yields a model of $\mathrm{B}_{\mathcal{S}}\mathrm{LL}$

## $\mathcal{S}$-stratification of Rel iff $[\![-]\!] : \mathcal{S} \to \mathcal{P}(\mathbb{N})$

$I \leqslant J$ implies $[\![I]\!] \subseteq [\![J]\!]$,

$[\![I]\!] \oplus [\![J]\!] \subseteq [\![I+J]\!]$, $\{0\} \subseteq [\![0_{\mathcal{S}}]\!]$,

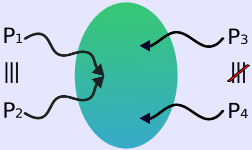
$[\![I]\!] \odot [\![J]\!] \subseteq [\![I*J]\!]$, $\{1\} \subseteq [\![1_{\mathcal{S}}]\!]$.

# From syntax to semantics and back

| Category $\mathcal{L}$ | maximal interpretation $\mathcal{S} \to ?$ | internal semiring $\tilde{\mathcal{L}}[!1, 1]$ |
|---|---|---|
| $\mathtt{Rel}$ | $\mathcal{P}(\mathbb{N})$ | $\mathcal{P}(\mathbb{N})$ |
| $\mathtt{Rel}^{\mathcal{R}}$ [CES10] | $\mathcal{P}(\mathcal{R})$ | $\mathcal{P}(\mathcal{R})$ |
| $\mathtt{ScottL}$ [Ehr12] | $\mathbb{B}_\perp$ | $\mathbb{B}_\perp$ |
| $\mathtt{Coh}^{\mathbb{N}}$ [Gir88] | $\mathcal{P}(\mathbb{N})$ | $\mathbb{N}_f$ |
| $\mathtt{Coh}^{\mathbb{B}}$ [Gir88] | $\mathcal{P}(\mathbb{B})$ | $\mathbb{B}_f$ |

# Closing old problems and opening new ones

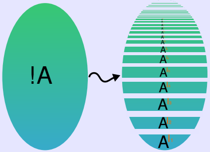
## Characterising K-models that are fully abstract for $\mathcal{H}^*$




Many side contributions:

- Theory of calculi with tests,
- Inequational = equational full abst. in K-models,
- A new, more general, proof of the approximation theorem

## Understanding $B_{\mathcal{S}}LL$ via models of LL




Many works in progress:

- A maximal stratification
- Toward dependent $B_{\mathcal{S}}LL$
- Effects and coeffects
- Classifying models of LL
- Splitting Semirings
- Distributing monads

# Works in Progress

## A maximal stratification

Another semiring emerging from the stratification.

## Toward dependence

A semantical study of resource dependence “à la BLL”.

## Effects and coeffects

Studying the interaction between effects and coeffects.

## Classifying models of LL

Can we classify linear categories with their internal semiring?

## Splitting Semirings

The notion of semiring seems to be splittable into interacting linear and non-linear versions.

## Distributing monads

Many linear cat. comes from a monadic distribution in Set. What links with coeffects?

# Dependence?

**$B_{\mathcal{S}}LL$ is weak as a logic**

Not the case of BLL or $D_\ell$PCF which resources are dependent.

$$A \vdash \quad B^{y \leqslant x} \multimap C^{z \leqslant x}$$

# Dependence?

**$B_{\mathcal{S}}LL$ is weak as a logic**

Not the case of BLL or $D_\ell$PCF which resources are dependent.

$$A \vdash_{x:\mathbb{N}} B^{y\leqslant x} \multimap C^{z\leqslant x}$$

- the elements of the "semiring" are dependent on some resource context

# Dependence?

**$B_{\mathcal{S}}LL$ is weak as a logic**

Not the case of BLL or $D_\ell$PCF which resources are dependent.

$$\frac{A \vdash_{x:\mathbb{N}} B^{y\leqslant x} \multimap C^{z\leqslant x}}{A^{x\leqslant 1} \vdash (B^{y\leqslant x} \multimap C^{z\leqslant x})^{x\leqslant 1}}\ \texttt{Prom}$$

- the elements of the "semiring" are dependent on some resource context and exponentials can modify this context: they are binders,

# Dependence?

**$B_{\mathcal{S}}LL$ is weak as a logic**

Not the case of BLL or $D_\ell$PCF which resources are dependent.

$$\frac{A \vdash_{x:\mathbb{N}} B^{y\leqslant x} \multimap C^{z\leqslant x}}{A^{x\leqslant 1} \vdash (B^{y\leqslant x} \multimap C^{z\leqslant x})^{x\leqslant 1}}\ \texttt{Prom} \qquad \frac{B^{y\leqslant 1} \multimap C^{z\leqslant 1} \vdash D}{(B^{y\leqslant x} \multimap C^{z\leqslant x})^{x\leqslant 1} \vdash D}\ \texttt{Der}$$

- the elements of the "semiring" are dependent on some resource context and exponentials can modify this context: they are binders,
- derivation can perform global rewriting over formulas,

# Dependence?

**$B_{\mathcal{S}}$LL is weak as a logic**

Not the case of BLL or $D_\ell$PCF which resources are dependent.

$$\dfrac{\dfrac{\dfrac{\Pi_1}{A \vdash_{x:\mathbb{N}} B^{y\leqslant x} \multimap C^{z\leqslant x}}}{A^{x\leqslant 1} \vdash (B^{y\leqslant x} \multimap C^{z\leqslant x})^{x\leqslant 1}}\ \texttt{Prom} \qquad \dfrac{\dfrac{\Pi_2}{B^{y\leqslant 1} \multimap C^{z\leqslant 1} \vdash D}}{(B^{y\leqslant x} \multimap C^{z\leqslant x})^{x\leqslant 1} \vdash D}\ \texttt{Der}}{A^{x\leqslant 1} \vdash D}\ \texttt{Cut}$$

- the elements of the "semiring" are dependent on some resource context and exponentials can modify this context: they are binders,
- derivation can perform global rewriting over formulas,
- cut-elimination can perform global rewriting over proofs.

# Toward dependence

## Co-classical $\mathcal{U}$

$u \in \mathcal{L}$ with $\epsilon, \rho$:

$$\mathbb{1} \xrightarrow{\epsilon_u} u$$

$$u \otimes u \xrightarrow{\rho_u} u$$

## $\mathcal{L}[!1, 1] \rightsquigarrow \biguplus_{u,v \in \mathcal{U}} \mathcal{L}[!u, v] \simeq \mathcal{U}_!$

$$0_{u,v} := !u \xrightarrow{w_u} \mathbb{1} \xrightarrow{\epsilon_v} v \qquad 1_{u,v} := !u \xrightarrow{d_u} u$$

$$I +_{u,v} J := !u \xrightarrow{c_u} !u \otimes !u \xrightarrow{I \otimes J} u \otimes v \xrightarrow{\rho_v} v$$

$$I \cdot_{u,v,w} J := !u \xrightarrow{p_u} !!u \xrightarrow{!I} !v \xrightarrow{J} w$$

## $\mathcal{U}$ acts on $\mathcal{U}_!$

If $I \in \mathcal{U}_![u, v]$ and $\iota \in \mathcal{U}[v, w]$, then $I; \iota \in \mathcal{U}_![u, w]$

## A dependent $B_{\mathcal{S}}LL$?

$$\frac{A_1{}^{I_1}, \cdots, A_n{}^{I_n} \vdash_u B}{A_1{}^{I_1 \cdot J}, \cdots A_n{}^{I_n \cdot J} \vdash_v B^J}\ \texttt{Prom} \qquad \frac{\Gamma, A[C^{I \cdot \iota}/C^I] \vdash_v B}{\Gamma, A^{1 \cdot \iota} \vdash_v B}\ \texttt{Der}$$

where $J \in \mathcal{L}[!u, v]$ and $\alpha \in \mathcal{L}[u, v]$

# Definition of K-models

## Extensional K-model [Krivine 1993]:

A preorder $(D, \leqslant)$ with a bijection "$\to$" from $\mathcal{A}_f(D)^{op} \times D$ to $D$.

### Antichains

$\mathcal{A}_f(D)$ are finite antichains over $D$, i.e. $a \in \mathcal{A}_f(D)$ if:

$$\forall \alpha, \beta \in a, \quad \alpha \not\leqslant \beta$$

### Order on antichains

The order $\leqslant$ on $D$ extends to $\mathcal{A}_f(D)$ by $a \leqslant b$ iff:

$$\forall \alpha \in a, \exists \beta \in b, \quad \alpha \leqslant \beta$$

We can replace $\mathcal{A}_f(D)$ by $\mathcal{P}_f(D), \mathcal{M}_f(D)$ or intersections

Sub-class of filter models.

Contain historical models: $\mathcal{D}_\infty$, $P_\infty$, $D^*_\infty$...

Contain all well-stratified filter models.

# K-models are the reflexive elements of $\text{ScottL}_!$

**A model of linear logic: ScottL [Ehrhard2012]**

**Objects:** Posets **Morphisms:** linear fct. $\mathcal{I}(D) \to \mathcal{I}(P)$

**Exponential:** Finite antichains $!D = \mathcal{A}_f(D)$

$\mathcal{I}(D)$ represents the complete lattice of initial segments over $D$
a function is said linear if it preserves every sups

**The Kleisli category: $\text{ScottL}_!$**

**Objects:** Posets **Morphisms:** continuous fct. $\mathcal{I}(D) \to \mathcal{I}(P)$

**Identities:** $1_D = id_{\mathcal{I}(D)}$

**Composition:** the function composition

**Cartesian product:** $\&_{i\in I} D_i := \{(i,\alpha) | i \in I, \alpha \in A_D\}$

**Exponential object:** $A \Rightarrow B = \mathcal{A}_f(A)^{op} \times B$

# Examples (well-stratified)

**Reminder on well-stratification:**

It is the approximation by projections $(|.|_k)_{k\in\mathbb{N}}$ s.t.

$$|f|_{k+1}(x_k) = |f(x)|_k \qquad |f|_0(\bot) = |f(\bot)|_0$$

**(Equivalent presentation of) Well-stratified K-models:**

They are extensional completions of (where $\sigma$ is a permutations)

$$U_0^{A,\sigma} = A \qquad \forall\alpha\in A,\quad \alpha = \emptyset \to \sigma(\alpha)$$

**The completion preserves hyperimmunity**

A completion $\bar{E}$ is hyperimmune iff $E$ is.

**Well-stratified models are hyperimmune**

for all $\alpha_1 \in A$ and $g:\mathbb{N}\to\mathbb{N}$,

$$\alpha_1 = \emptyset\to\cdots\to\emptyset\to\sigma^n(\alpha_1)$$

# Our main result

**Theorem**

For any extensional K-models $D$, the following are equivalent when $D$ respects approximation theorem:

- $D$ is hyperimmune,
- $D$ is inequationally fully abstract for $\mathcal{H}^*$,
- $D$ is fully abstract for $\mathcal{H}^*$.

# Semantic proof: Böhm trees

## Definition of the Böhm tree $\mathbf{BT}(M)$ of a term $M$

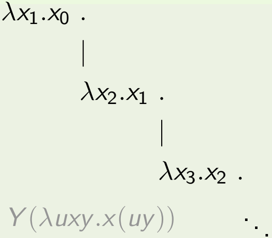
**BT** is a coinductive structure defined by:

If $M$ head diverges, $\mathbf{BT}(M) = \Omega$,

if $M \rightarrow_h^* \lambda x_1 ... x_n . y\ N_1 \cdots N_k$ then

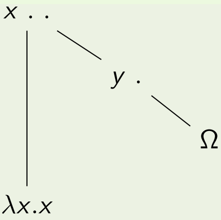
$$\mathbf{BT}(M) = \lambda x_1 ... x_n . y\ \mathbf{BT}(N_1) \cdots \mathbf{BT}(N_k).$$

### Example: $\mathbf{BT}(J\ x_0)$




$J = Y(\lambda uxy.x(uy))$

### Example: $\mathbf{BT}(x\ (II)\ (y\ YI))$



# Approximant

## Definition of the approximation $M \subseteq_{\mathbf{BT}} N$

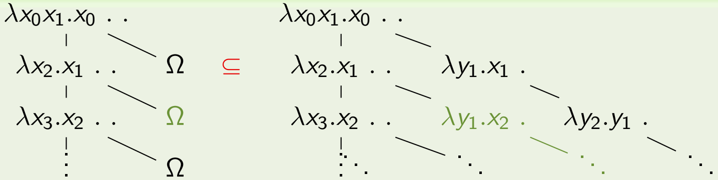
$M$ is an approximant of $N$ if $\mathbf{BT}(M)$ is a subtree of $\mathbf{BT}(N)$ with truncated edges replaced by $\Omega$'s. It if the largest relation st.:

$\Omega \subseteq V$ for all $V$

If for all $i \leqslant k$, $U_i \subseteq V_i$, then

$$(\lambda x_1 \ldots x_n.y\ U_1 \cdots U_k) \subseteq (\lambda x_1 \ldots x_n.y\ V_1 \cdots V_k).$$

## Example $Y(\lambda uxy.x(uy)\Omega) \subseteq_{\mathbf{BT}} Y(\lambda uxy.x(uy)(Jx))$



# $\eta\infty$-order

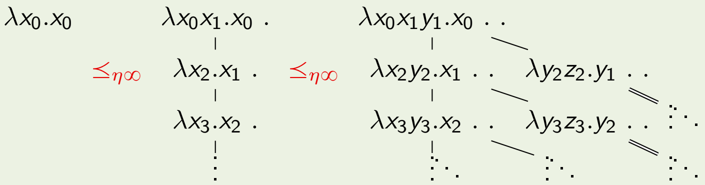
## Definition of the order $\preceq_{\eta\infty}$

We said $M \preceq_{\eta\infty} N$ if **BT**$(N)$ is the result of infinitely many $\eta$-expansions on **BT**$(M)$. Or co-inductively $M \preceq_{\eta\infty} N$ if:

either $M$ and $N$ diverges,

or $N \to_h^* \lambda x_1 ... x_n.y\ N_1 \cdots N_k$ and there is $\lambda x_1 ... x_n.y\ M_1 \cdots M_k \succeq_\eta M$ such that $N_i \preceq_{\eta\infty} M_i$ for all $i \leqslant k$.

## Example: $I \preceq_{\eta\infty} J \preceq_{\eta\infty} Y\ (\lambda uxyz.x\ (uy)\ (uz))$

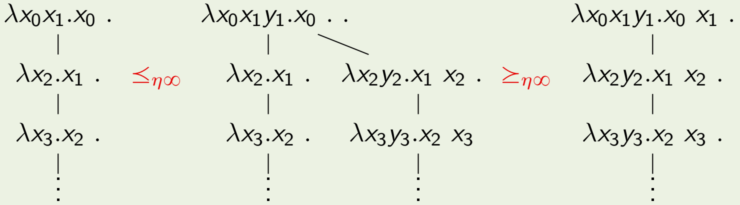


# Full abstraction and $\eta\infty$-order

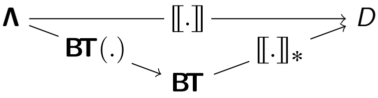
**Theorem [Barendregt]**

$M \equiv_o N$ iff they are $\eta\infty$-bounded.

**Example:** $J \equiv_o Y\ (\lambda uxyz.x\ y\ (uz))$



# Semantic proof: quasi-approximation




## Several interpretations of **BT**s

finite/inductive one:

$$[\![U]\!]_{ind} := \bigcup_{V \subseteq_f U} [\![V]\!].$$

coninductive one,

quasi-finite one:

$$[\![U]\!]_{ind} := \bigcup_{V \subseteq_{qf} U} [\![V]\!]_{coind}.$$

## Quasi-finite BTs

$V \subseteq_{qf} [\![M]\!] \quad \Rightarrow \quad V$ recursive,

stable by $\supseteq$ and $\equiv_{\eta\infty}$.

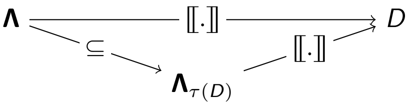
## Diagram

If $* = ind \rightsquigarrow$ approximation,

If

$* = qf \rightsquigarrow$ quasi-approximation,

# $\mathbf{\Lambda}_{\tau(D)}$: a model-specific language




**$\mathbf{\Lambda}_{\tau(D)}$ extends $\mathbf{\Lambda}$ with elements of $D$**

We add the operators $\bar{\epsilon}_\alpha$, $\bar{\tau}_\alpha(Q)$ and $\tau_\alpha(M)$ for all $\alpha \in D$

**We have some control over the assertion $\alpha \in [\![M]\!]$**

Upto the approximation theorem,
$D$ co-defines all prime algebraic elements:

$$\tau_\alpha(M)\Downarrow \quad \Leftrightarrow \quad \alpha \in [\![M]\!]$$

**Full abstraction via deffinissability**

Upto the approximation theorem,
$D$ is fully abstract for $\Lambda_{\tau(D)}$
by defining all prime algebraic:

$$\forall \alpha \in D, \qquad [\![\bar{\epsilon}_\alpha]\!] = \downarrow \alpha$$

# Syntax of tests

## 2 syntactic kinds

(term) $M, N := x \mid \lambda x.M \mid M\ N \mid \sum_{i\leqslant n} \bar{\tau}_{\alpha_i}(Q_i) \quad , \forall(\alpha_i)_i \in D^n$

(test) $P, Q := \sum_{i\leqslant n} P_i \mid \prod_{i\leqslant n} P_i \mid \tau_\alpha(M) \quad , \forall \alpha \in D$

## Polarised view: tests are processes

$\tau_\alpha(M) \quad \simeq \quad M * \alpha \qquad \bar{\tau}_\alpha(Q) * \pi \quad \simeq \quad Q\cdot(\bar{\alpha} * \pi)$

## Reduction strategy (extending head reduction)

$\tau(M)$: infinite application,

$\bar{\tau}(Q)$: infinite abstraction,

$\Sigma_i E_i$: may non-determinism,

$\Pi_i Q_i$: must non-determinism.

# Tests and typing

## Type inference procedural for $\vdash \omega : 2$

$\vdash \omega : 2$

## Reduction of $\tau_2(\omega)$

$\tau_2(\omega)$

## Notations

$\omega = \lambda x.x\ x$

$D$ is the completion on $\mathbb{N}$ with $n = [0, n[ \to 0$

# Tests and typing

## Type inference procedural for $\vdash \omega : 2$

$$\frac{x : \{0,1\} \vdash x\ x : 0}{\vdash \omega : 2}\ 2 = \{0,1\} \rightarrow 0$$

## Reduction of $\tau_2(\omega)$

$\tau_2(\omega) \rightarrow \tau_0(\bar{\epsilon}_{\{0,1\}}\ \bar{\epsilon}_{\{0,1\}})$

## Notations

$\omega = \lambda x.x\ x$

$D$ is the completion on $\mathbb{N}$ with $n = [0, n[ \rightarrow 0$

# Tests and typing

## Type inference procedural for $\vdash \omega : 2$

$$\dfrac{\dfrac{x' : \{0\}, x : \{0,1\} \vdash x'\ x : 0 \qquad x' : \{1\}, x : \{0,1\} \vdash x'\ x : 0}{x : \{0,1\} \vdash x\ x : 0}\ \text{Choice}}{\vdash \omega : 2}\ 2 = \{0,1\} \to 0$$

## Reduction of $\tau_2(\omega)$

$$\begin{aligned} \tau_2(\omega) &\to \tau_0(\bar{\epsilon}_{\{0,1\}}\ \bar{\epsilon}_{\{0,1\}}) \\ &\to \tau_0(\bar{\epsilon}_0\ \bar{\epsilon}_{\{0,1\}}) + \tau_0(\bar{\epsilon}_1\ \bar{\epsilon}_{\{0,1\}}) \end{aligned}$$

## Notations

$\omega = \lambda x.x\ x$

$D$ is the completion on $\mathbb{N}$ with

$n = [0, n[ \to 0$

# Tests and typing

## Type inference procedural for $\vdash \omega : 2$

$$\dfrac{\dfrac{\dfrac{\dots}{x' : \{0\}, x : \{0,1\} \vdash x'\ x : 0} \qquad \dfrac{x : \{0,1\} \vdash x : 0 \qquad 0 \leqslant 0}{x' : \{1\}, x : \{0,1\} \vdash x'\ x : 0}\ 1 = 0 \to 0}{x : \{0,1\} \vdash x\ x : 0}\ \text{Choice}}{\vdash \omega : 2}\ 2 = \{0,1\} \to 0$$

## Reduction of $\tau_2(\omega)$

$$\begin{aligned}\tau_2(\omega) &\to \tau_0(\bar{\epsilon}_{\{0,1\}}\ \bar{\epsilon}_{\{0,1\}}) \\ &\to \tau_0(\bar{\epsilon}_0\ \bar{\epsilon}_{\{0,1\}}) + \tau_0(\bar{\epsilon}_1\ \bar{\epsilon}_{\{0,1\}}) \\ &\to \tau_0(\bar{\epsilon}_0\ \bar{\epsilon}_{\{0,1\}}) + \tau_0(\bar{\tau}_0(\tau_0(\bar{\epsilon}_{\{0,1\}})))\end{aligned}$$

## Notations

$\omega = \lambda x.x\ x$

$D$ is the completion on $\mathbb{N}$ with

$n = [0, n[ \to 0$

# Tests and typing

## Type inference procedural for $\vdash \omega : 2$

$$\dfrac{\dfrac{\dfrac{\dots}{x' : \{0\}, x : \{0,1\} \vdash x'\ x : 0} \quad \dfrac{x : \{1\} \vdash x : 0 \quad \overline{0 \leqslant 0}}{x' : \{1\}, x : \{0,1\} \vdash x'\ x : 0}\ 1 = 0 \to 0}{x : \{0,1\} \vdash x\ x : 0}\ \text{Choice}}{\vdash \omega : 2}\ 2 = \{0,1\} \to 0$$

## Reduction of $\tau_2(\omega)$

$$\begin{aligned}
\tau_2(\omega) &\to \tau_0(\bar\epsilon_{\{0,1\}}\ \bar\epsilon_{\{0,1\}}) \\
&\to \tau_0(\bar\epsilon_0\ \bar\epsilon_{\{0,1\}}) + \tau_0(\bar\epsilon_1\ \bar\epsilon_{\{0,1\}}) \\
&\to \tau_0(\bar\epsilon_0\ \bar\epsilon_{\{0,1\}}) + \tau_0(\bar\tau_0(\tau_0(\bar\epsilon_{\{0,1\}}))) \\
&\to \tau_0(\bar\epsilon_0\ \bar\epsilon_{\{0,1\}}) + \tau_0(\bar\epsilon_{\{0,1\}})
\end{aligned}$$

## Notations

$\omega = \lambda x.x\ x$

$D$ is the completion on $\mathbb{N}$ with

$n = [0, n[ \to 0$

# Tests and typing

## Type inference procedural for $\vdash \omega : 2$

$$\dfrac{\dfrac{\dots}{x' : \{0\}, x : \{0,1\} \vdash x'\ x : 0} \qquad \dfrac{\dfrac{}{x : \{0,1\} \vdash x : 0} \quad \dfrac{}{0 \leqslant 0}}{x' : \{1\}, x : \{0,1\} \vdash x'\ x : 0}\ 1 = 0 \to 0}{\dfrac{x : \{0,1\} \vdash x\ x : 0}{\vdash \omega : 2}\ 2 = \{0,1\} \to 0}\ \text{Choice}$$

## Reduction of $\tau_2(\omega)$

$$\begin{aligned}
\tau_2(\omega) &\to \tau_0(\bar{\epsilon}_{\{0,1\}}\ \bar{\epsilon}_{\{0,1\}}) \\
&\to \tau_0(\bar{\epsilon}_0\ \bar{\epsilon}_{\{0,1\}}) + \tau_0(\bar{\epsilon}_1\ \bar{\epsilon}_{\{0,1\}}) \\
&\to \tau_0(\bar{\epsilon}_0\ \bar{\epsilon}_{\{0,1\}}) + \tau_0(\bar{\tau}_0(\tau_0(\bar{\epsilon}_{\{0,1\}}))) \\
&\to \tau_0(\bar{\epsilon}_0\ \bar{\epsilon}_{\{0,1\}}) + \tau_0(\bar{\epsilon}_{\{0,1\}}) \\
&\to \tau_0(\bar{\epsilon}_0\ \bar{\epsilon}_{\{0,1\}}) + \epsilon
\end{aligned}$$

## Notations

$\omega = \lambda x.x\ x$

$D$ is the completion on $\mathbb{N}$ with $n = [0, n[ \to 0$

# Infinite derivation

$\tau_p(J\ \bar{\epsilon}_p)$

$x : p \vdash J\ x : p$

## Notations

$J = Y\ (\lambda uxy.x\ (u\ y))$,

$Y$ is a fixpoint.

e.g., $Y = (\lambda gf.f\ (ggf))(\lambda gf.f\ (ggf))$

## Model: $D_\infty^*$

Completion of $\{p, q\}$ with:

$$p = \{q\} \rightarrow p$$
$$q = \{p\} \rightarrow q$$

# Infinite derivation

$$\tau_p(J\ \bar{\epsilon}_p) \ \to^* \tau_p(\lambda y.\bar{\epsilon}_p\ (J\ y))$$

$$\frac{x : p \vdash \lambda y.x\ (J\ y) : p}{x : p \vdash J\ x : p}\ J\ x \to_h^* \lambda y.x\ (J\ y)$$

## Notations

$J = Y\ (\lambda uxy.x\ (u\ y))$,

$Y$ is a fixpoint.

e.g., $Y = (\lambda gf.f\ (ggf))(\lambda gf.f\ (ggf))$

## Model: $D_\infty^*$

Completion of $\{p, q\}$ with:

$$p = \{q\} \to p$$
$$q = \{p\} \to q$$

# Infinite derivation

$$\tau_p(J\ \bar{\epsilon}_p) \to^* \tau_p(\lambda y.\bar{\epsilon}_p\ (J\ y)) \to \tau_p(\bar{\epsilon}_p\ (J\ \bar{\epsilon}_q))$$

$$\dfrac{\dfrac{x : p, y : q \vdash x\ (J\ y) : p}{x : p \vdash \lambda y.x\ (J\ y) : p}\ p = q \to p}{x : p \vdash J\ x : p}\ J\ x \to_h^* \lambda y.x\ (J\ y)$$

## Notations

$J = Y\ (\lambda uxy.x\ (u\ y))$,

$Y$ is a fixpoint.

e.g., $Y = (\lambda gf.f\ (ggf))(\lambda gf.f\ (ggf))$

## Model: $D_\infty^*$

Completion of $\{p, q\}$ with:

$$p = \{q\} \to p$$
$$q = \{p\} \to q$$

# Infinite derivation

$$\cfrac{\cfrac{\cfrac{y : q \vdash J\ y : q \qquad p \leqslant p}{x : p, y : q \vdash x\ (J\ y) : p}\ p = q \to p}{x : p \vdash \lambda y.x\ (J\ y) : p}\ p = q \to p}{x : p \vdash J\ x : p}\ J\ x \to_h^* \lambda y.x\ (J\ y)$$

$$\begin{aligned} \tau_p(J\ \bar{\epsilon}_p) &\to^* \tau_p(\lambda y.\bar{\epsilon}_p\ (J\ y)) \\ &\to \tau_p(\bar{\epsilon}_p\ (J\ \bar{\epsilon}_q)) \\ &\to \tau_p(\bar{\tau}_p(\tau_q(J\ \bar{\epsilon}_q))) \end{aligned}$$

## Notations

$J = Y\ (\lambda uxy.x\ (u\ y))$,

$Y$ is a fixpoint.

e.g., $Y = (\lambda gf.f\ (ggf))(\lambda gf.f\ (ggf))$

## Model: $D_\infty^*$

Completion of $\{p, q\}$ with:

$$p = \{q\} \to p$$

$$q = \{p\} \to q$$

# Infinite derivation

$$\dfrac{\dfrac{\dfrac{y : q \vdash J\ y : q \qquad \overline{p \leqslant p}}{x : p, y : q \vdash x\ (J\ y) : p}\ p = q{\to}p}{x : p \vdash \lambda y.x\ (J\ y) : p}\ p = q{\to}p}{x : p \vdash J\ x : p}\ J\ x \to_h^* \lambda y.x\ (J\ y)$$

$$\begin{aligned}\tau_p(J\ \bar\epsilon_p) &\to^* \tau_p(\lambda y.\bar\epsilon_p\ (J\ y))\\ &\to \tau_p(\bar\epsilon_p\ (J\ \bar\epsilon_q))\\ &\to \tau_p(\bar\tau_p(\tau_q(J\ \bar\epsilon_q)))\\ &\to \tau_q(J\ \bar\epsilon_q)\end{aligned}$$

**Notations**

$J = Y\ (\lambda uxy.x\ (u\ y))$,

$Y$ is a fixpoint.

e.g., $Y = (\lambda gf.f\ (ggf))(\lambda gf.f\ (ggf))$

**Model: $D_\infty^*$**

Completion of $\{p, q\}$ with:

$$p = \{q\}{\to}p$$
$$q = \{p\}{\to}q$$

# Infinite derivation

$$\dfrac{\dfrac{\dfrac{\dfrac{\dots}{y : q \vdash J\ y : q} \qquad \dfrac{}{p \leqslant p}}{x : p, y : q \vdash x\ (J\ y) : p}\ p = q{\to}p}{x : p \vdash \lambda y.x\ (J\ y) : p}\ p = q{\to}p}{x : p \vdash J\ x : p}\ J\ x \to_h^* \lambda y.x\ (J\ y)$$

$$\begin{aligned} \tau_p(J\ \bar{\epsilon}_p) &\to^* \tau_p(\lambda y.\bar{\epsilon}_p\ (J\ y)) \\ &\to \tau_p(\bar{\epsilon}_p\ (J\ \bar{\epsilon}_q)) \\ &\to \tau_p(\bar{\tau}_p(\tau_q(J\ \bar{\epsilon}_q))) \\ &\to \tau_q(J\ \bar{\epsilon}_q) \\ &\to \dots \end{aligned}$$

## Notations

$J = Y\ (\lambda uxy.x\ (u\ y))$,

$Y$ is a fixpoint.

e.g., $Y = (\lambda gf.f\ (ggf))(\lambda gf.f\ (ggf))$

## Model: $D_\infty^*$

Completion of $\{p, q\}$ with:

$$p = \{q\}{\to}p$$
$$q = \{p\}{\to}q$$

# Procedural of $\tau_\alpha(M)$

$\Gamma \vdash M : \alpha$

# Procedural of $\tau_\alpha(M)$

$$\frac{\Gamma \vdash \lambda x_1 ... x_n . x_k \ N_1 \cdots N_m : \alpha}{\Gamma \vdash M : \alpha} \ M \to_h^* \lambda x_1 ... x_n . x_k \ N_1 \cdots N_m$$

# Procedural of $\tau_\alpha(M)$

$$\dfrac{\dfrac{\Gamma \vdash \lambda x_1 ... x_n . x_k \; N_1 \cdots N_m : a_1 \to \cdots \to a_n \to \alpha'}{\Gamma \vdash \lambda x_1 ... x_n . x_k \; N_1 \cdots N_m : \alpha} \; \alpha = a_1 \to \cdots \to a_n \to \alpha'}{\Gamma \vdash M : \alpha} \; M \to_h^* \lambda x_1 ... x_n . x_k \; N_1 \cdots N_m$$

# Procedural of $\tau_\alpha(M)$

$$\dfrac{\dfrac{\dfrac{\Gamma' \vdash x_k\ N_1 \cdots N_m : \alpha'}{\Gamma \vdash \lambda x_1 ... x_n . x_k\ N_1 \cdots N_m : a_1 \to \cdots \to a_n \to \alpha'}\ \Gamma' = (\Gamma, (x_i : a_i)_i)}{\Gamma \vdash \lambda x_1 ... x_n . x_k\ N_1 \cdots N_m : \alpha}\ \alpha = a_1 \to \cdots \to a_n \to \alpha'}{\Gamma \vdash M : \alpha}\ M \to_h^* \lambda x_1 ... x_n . x_k\ N_1 \cdots N_m$$

# Procedural of $\tau_\alpha(M)$

$$\dfrac{\dfrac{\dfrac{\dfrac{\Gamma', x'_k : \beta \vdash x'_k\ N_1 \cdots N_m : \alpha'}{\Gamma' \vdash x_k\ N_1 \cdots N_m : \alpha'}\ \exists \beta \in a_k}{\Gamma \vdash \lambda x_1 ... x_n . x_k\ N_1 \cdots N_m : a_1 \to \cdots \to a_n \to \alpha'}\ \Gamma' = (\Gamma, (x_i : a_i)_i)}{\Gamma \vdash \lambda x_1 ... x_n . x_k\ N_1 \cdots N_m : \alpha}\ \alpha = a_1 \to \cdots \to a_n \to \alpha'}{\Gamma \vdash M : \alpha}\ M \to_h^* \lambda x_1 ... x_n . x_k\ N_1 \cdots N_m$$

# Procedural of $\tau_\alpha(M)$

$$
\dfrac{\dfrac{\dfrac{\Gamma', x'_k : b_1 \to \cdots \to b_m \to \beta' \vdash x'_k\ N_1 \cdots N_m : \alpha'}{\dfrac{\Gamma', x'_k : \beta \vdash x'_k\ N_1 \cdots N_m : \alpha'}{\Gamma' \vdash x_k\ N_1 \cdots N_m : \alpha'}\ \exists \beta \in a_k}\ \beta = b_1 \to \cdots \to b_m \to \beta'}{\Gamma \vdash \lambda x_1 \ldots x_n . x_k\ N_1 \cdots N_m : a_1 \to \cdots \to a_n \to \alpha'}\ \Gamma' = (\Gamma, (x_i : a_i)_i)}{\dfrac{\Gamma \vdash \lambda x_1 \ldots x_n . x_k\ N_1 \cdots N_m : \alpha}{\Gamma \vdash M : \alpha}\ M \to_h^* \lambda x_1 \ldots x_n . x_k\ N_1 \cdots N_m}\ \alpha = a_1 \to \cdots \to a_n \to \alpha'
$$

# Procedural of $\tau_\alpha(M)$

$$
\cfrac{\cfrac{\cfrac{\cfrac{\Gamma' \vdash N_i : \gamma_i \qquad \alpha' \leqslant \beta'}{\Gamma', x'_k : b_1 \to \cdots \to b_m \to \beta' \vdash x'_k\ N_1 \cdots N_m : \alpha'}\ \forall i, \forall \gamma_i \in b_i}{\Gamma', x'_k : \beta \vdash x'_k\ N_1 \cdots N_m : \alpha'}\ \beta = b_1 \to \cdots \to b_m \to \beta'}{\cfrac{\Gamma' \vdash x_k\ N_1 \cdots N_m : \alpha'}{\Gamma \vdash \lambda x_1 ... x_n . x_k\ N_1 \cdots N_m : a_1 \to \cdots \to a_n \to \alpha'}\ \Gamma' = (\Gamma, (x_i : a_i)_i)}\ \exists \beta \in a_k}{\cfrac{\Gamma \vdash \lambda x_1 ... x_n . x_k\ N_1 \cdots N_m : \alpha}{\Gamma \vdash M : \alpha}\ M \to_h^* \lambda x_1 ... x_n . x_k\ N_1 \cdots N_m}\ \alpha = a_1 \to \cdots \to a_n \to \alpha'
$$

# Procedural of $\tau_\alpha(M)$

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dots}{\Gamma' \vdash N_i : \gamma_i} \qquad \alpha' \leqslant \beta'}{\Gamma', x'_k : b_1 \to \cdots \to b_m \to \beta' \vdash x'_k\ N_1 \cdots N_m : \alpha'}\ \forall i, \forall \gamma_i \in b_i}{\Gamma', x'_k : \beta \vdash x'_k\ N_1 \cdots N_m : \alpha'}\ \beta = b_1 \to \cdots \to b_m \to \beta'}{\Gamma' \vdash x_k\ N_1 \cdots N_m : \alpha'}\ \exists \beta \in a_k}{\Gamma \vdash \lambda x_1 \dots x_n . x_k\ N_1 \cdots N_m : a_1 \to \cdots \to a_n \to \alpha'}\ \Gamma' = (\Gamma, (x_i : a_i)_i)}{\dfrac{\Gamma \vdash \lambda x_1 \dots x_n . x_k\ N_1 \cdots N_m : \alpha}{\Gamma \vdash M : \alpha}\ M \to_h^* \lambda x_1 \dots x_n . x_k\ N_1 \cdots N_m}\ \alpha = a_1 \to \cdots \to a_n \to \alpha'
$$

# Procedural of $\tau_\alpha(M)$

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\cdots}{\Gamma' \vdash N_i : \gamma_i} \quad \alpha' \leqslant \beta'}{\Gamma', x'_k : b_1 \to \cdots \to b_m \to \beta' \vdash x'_k\ N_1 \cdots N_m : \alpha'}\ \forall i, \forall \gamma_i \in b_i}{\Gamma', x'_k : \beta \vdash x'_k\ N_1 \cdots N_m : \alpha'}\ \beta = b_1 \to \cdots \to b_m \to \beta'}{\Gamma' \vdash x_k\ N_1 \cdots N_m : \alpha'}\ \exists \beta \in a_k}{\Gamma \vdash \lambda x_1 \ldots x_n . x_k\ N_1 \cdots N_m : a_1 \to \cdots \to a_n \to \alpha'}\ \Gamma' = (\Gamma, (x_i : a_i)_i)}{\Gamma \vdash \lambda x_1 \ldots x_n . x_k\ N_1 \cdots N_m : \alpha}\ \alpha = a_1 \to \cdots \to a_n \to \alpha'}{\Gamma \vdash M : \alpha}\ M \to_h^* \lambda x_1 \ldots x_n . x_k\ N_1 \cdots N_m
$$

**4 possible failures**

- $M$ diverges
- $a_k = \varnothing$
- $\alpha' \nleqslant \beta'$
- infinite derivation

**Consistence**

This procedure succeeds iff $\alpha \in [\![M]\!]^\Gamma$

by the approximation theorem's hypothesis

**$\Lambda_{\tau(D)}$**

The $\lambda$-calculus with D-tests internalizes this reduction:

$$\tau_\alpha(M)\Downarrow \Leftrightarrow \alpha \in [\![M]\!]$$

# Böhm trees and tests

## Typing of $\vdash J : \mu{\to}\mu$

$$\vdash J : \mu{\to}\mu$$

## Böhm tree

$$J^{\mu\to\mu} \, .$$

## Completion of $\{*, \mu\}$

$$* = \emptyset{\to}* \qquad \mu = \{*\}{\to}*$$

## Derivation of $\tau_{\mu\to\mu}(J)$

$$\tau_{\mu\to\mu}(J)$$

# Böhm trees and tests

## Typing of $\vdash J : \mu\to\mu$

$$\frac{\vdash \lambda xy.x\ (J\ y) : \mu\to\mu}{\vdash J : \mu\to\mu}$$

## Böhm tree

$$(\lambda xy.x\ (J\ y))^{\mu\to\mu}\ .$$

## Completion of $\{*, \mu\}$

$$* = \emptyset\to* \qquad \mu = \{*\}\to*$$

## Derivation of $\tau_{\mu\to\mu}(J)$

$$\tau_{\mu\to\mu}(J)\ \to^*\ \tau_{\mu\to\mu}(\lambda xy.x\ (J\ y))$$

# Böhm trees and tests

## Typing of $\vdash J : \mu \to \mu$

$$\dfrac{\dfrac{x : \mu, y : *, \vdash x\ (J\ y) : *}{\vdash \lambda xy.x\ (J\ y) : \mu \to \mu}}{\vdash J : \mu \to \mu}$$

## Böhm tree

$$\lambda x^{\mu} y^{*}.(x\ (J\ y))^{*}\ .$$

## Completion of $\{*, \mu\}$

$$* = \emptyset \to * \qquad \mu = \{*\} \to *$$

## Derivation of $\tau_{\mu \to \mu}(J)$

$$\begin{aligned} \tau_{\mu\to\mu}(J) &\to^* \tau_{\mu\to\mu}(\lambda xy.x\ (J\ y)) \\ &\to^2 \tau_{*}(\bar{\epsilon}_{\mu}\ (J\ \bar{\epsilon}_{*})) \end{aligned}$$

# Böhm trees and tests

## Typing of $\vdash J : \mu{\to}\mu$

$$\dfrac{\dfrac{\dfrac{y : * \vdash J\ y : * \qquad * \leqslant *}{x : \mu, y : *, \vdash x\ (J\ y) : *}}{\vdash \lambda xy.x\ (J\ y) : \mu{\to}\mu}}{\vdash J : \mu{\to}\mu}$$

## Completion of $\{*, \mu\}$

$$* = \emptyset{\to}* \qquad \mu = \{*\}{\to}*$$

## Derivation of $\tau_{\mu\to\mu}(J)$

$$\begin{aligned}\tau_{\mu\to\mu}(J) &\to^* \tau_{\mu\to\mu}(\lambda xy.x\ (J\ y))\\ &\to^2 \tau_*(\bar{\epsilon}_\mu\ (J\ \bar{\epsilon}_*))\\ &\to \tau_*(\bar{\tau}_*(\tau_*(J\ \bar{\epsilon}_*)))\end{aligned}$$

## Böhm tree

$$\lambda x^\mu y^*.x^\mu\ (J\ y)^*\ .$$

# Böhm trees and tests

## Typing of $\vdash J : \mu \to \mu$

$$\dfrac{\dfrac{\dfrac{y : * \vdash J\ y : * \qquad \overline{* \leqslant *}}{x : \mu, y : *, \vdash x\ (J\ y) : *}}{\vdash \lambda xy.x\ (J\ y) : \mu \to \mu}}{\vdash J : \mu \to \mu}$$

## Completion of $\{*, \mu\}$

$$* = \emptyset \to * \qquad \mu = \{*\} \to *$$

## Derivation of $\tau_{\mu\to\mu}(J)$

$$\begin{aligned}\tau_{\mu\to\mu}(J) &\to^* \tau_{\mu\to\mu}(\lambda xy.x\ (J\ y)) \\ &\to^2 \tau_*(\bar{\epsilon}_\mu\ (J\ \bar{\epsilon}_*)) \\ &\to \tau_*(\bar{\tau}_*(\tau_*(J\ \bar{\epsilon}_*))) \\ &\to \tau_*(J\ \bar{\epsilon}_*)\end{aligned}$$

## Böhm tree

$$\lambda x^\mu y^*.x^\mu\ (J\ y)^*\ .$$

# Böhm trees and tests

## Typing of $\vdash J : \mu \to \mu$

$$\dfrac{\dfrac{\dfrac{\dfrac{y : * \vdash \lambda z.y\ (J\ z) : *}{y : * \vdash J\ y : *} \qquad \dfrac{}{* \leqslant *}}{x : \mu, y : *, \vdash x\ (J\ y) : *}}{\vdash \lambda xy.x\ (J\ y) : \mu \to \mu}}{\vdash J : \mu \to \mu}$$

## Böhm tree

$\lambda x^{\mu} y^{*}.x^{\mu}$ .

|

$(\lambda z.y\ (J\ z))^{*}$

## Completion of $\{*, \mu\}$

$$* = \emptyset \to * \qquad \mu = \{*\} \to *$$

## Derivation of $\tau_{\mu\to\mu}(J)$

$$\begin{aligned}
\tau_{\mu\to\mu}(J) &\to^{*} \tau_{\mu\to\mu}(\lambda xy.x\ (J\ y)) \\
&\to^{2} \tau_{*}(\bar{\epsilon}_{\mu}\ (J\ \bar{\epsilon}_{*})) \\
&\to \tau_{*}(\bar{\tau}_{*}(\tau_{*}(J\ \bar{\epsilon}_{*}))) \\
&\to \tau_{*}(J\ \bar{\epsilon}_{*}) \\
&\to^{*} \tau_{*}(\lambda y.\bar{\epsilon}_{*\ (Jy)})
\end{aligned}$$

# Böhm trees and tests

## Typing of $\vdash J : \mu \to \mu$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{y : *, z : \emptyset \vdash y\ (J\ z) : *}{y : * \vdash \lambda z.y\ (J\ z) : *}}{y : * \vdash J\ y : *} \qquad \dfrac{}{* \leqslant *}}{x : \mu, y : *, \vdash x\ (J\ y) : *}}{\vdash \lambda xy.x\ (J\ y) : \mu \to \mu}}{\vdash J : \mu \to \mu}$$

## Completion of $\{*, \mu\}$

$$* = \emptyset \to * \qquad \mu = \{*\} \to *$$

## Derivation of $\tau_{\mu\to\mu}(J)$

$$\begin{aligned}
\tau_{\mu\to\mu}(J) &\to^* \tau_{\mu\to\mu}(\lambda xy.x\ (J\ y)) \\
&\to^2 \tau_*(\bar{\epsilon}_\mu\ (J\ \bar{\epsilon}_*)) \\
&\to \tau_*(\bar{\tau}_*(\tau_*(J\ \bar{\epsilon}_*))) \\
&\to \tau_*(J\ \bar{\epsilon}_*) \\
&\to^* \tau_*(\lambda y.\bar{\epsilon}_{*\ (Jy)}) \\
&\to \tau_*(\bar{\epsilon}_*\ (J\Omega))
\end{aligned}$$

## Böhm tree

$$\lambda x^\mu y^*.x^\mu\ .$$

$$|$$

$$\lambda z^\emptyset.(y\ (J\ z))^*$$

# Böhm trees and tests

## Typing of $\vdash J : \mu \to \mu$

$$\dfrac{\dfrac{\dfrac{\dfrac{* \leqslant *}{y : *, z : \emptyset \vdash y\ (J\ z) : *}}{y : * \vdash \lambda z.y\ (J\ z) : *}}{y : * \vdash J\ y : *} \quad \dfrac{}{* \leqslant *}}{\dfrac{x : \mu, y : *, \vdash x\ (J\ y) : *}{\dfrac{\vdash \lambda xy.x\ (J\ y) : \mu \to \mu}{\vdash J : \mu \to \mu}}}$$

## Completion of $\{*, \mu\}$

$$* = \emptyset \to * \qquad \mu = \{*\} \to *$$

## Derivation of $\tau_{\mu\to\mu}(J)$

$$\begin{aligned}
\tau_{\mu\to\mu}(J) &\to^* \tau_{\mu\to\mu}(\lambda xy.x\ (J\ y)) \\
&\to^2 \tau_*(\bar{\epsilon}_\mu\ (J\ \bar{\epsilon}_*)) \\
&\to \tau_*(\bar{\tau}_*(\tau_*(J\ \bar{\epsilon}_*))) \\
&\to \tau_*(J\ \bar{\epsilon}_*) \\
&\to^* \tau_*(\lambda y.\bar{\epsilon}_{*\ (Jy)}) \\
&\to \tau_*(\bar{\epsilon}_*\ (J\Omega)) \\
&\to \tau_*(\bar{\epsilon}_*)
\end{aligned}$$

## Böhm tree

$$\lambda x^\mu y^*.x^\mu\ .$$

$$|$$

$$\lambda z^\emptyset.y^*\ (J\ z)^\emptyset$$

# Böhm trees and tests

## Typing of $\vdash J : \mu \to \mu$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{* \leqslant *}}{y : *, z : \emptyset \vdash y\ (J\ z) : *}}{y : * \vdash \lambda z.y\ (J\ z) : *}}{y : * \vdash J\ y : *} \quad \dfrac{}{* \leqslant *}}{\dfrac{x : \mu, y : *, \vdash x\ (J\ y) : *}{\dfrac{\vdash \lambda xy.x\ (J\ y) : \mu \to \mu}{\vdash J : \mu \to \mu}}}$$

## Böhm tree

$$\lambda x^{\mu} y^{*}.x^{\mu}\ .$$
$$|$$
$$\lambda z^{\emptyset}.y^{*}\ (J\ z)^{\emptyset}$$

## Completion of $\{*, \mu\}$

$$* = \emptyset \to * \qquad \mu = \{*\} \to *$$

## Derivation of $\tau_{\mu\to\mu}(J)$

$$\begin{aligned}
\tau_{\mu\to\mu}(J) &\to^{*} \tau_{\mu\to\mu}(\lambda xy.x\ (J\ y)) \\
&\to^{2} \tau_{*}(\bar{\epsilon}_{\mu}\ (J\ \bar{\epsilon}_{*})) \\
&\to \tau_{*}(\bar{\tau}_{*}(\tau_{*}(J\ \bar{\epsilon}_{*}))) \\
&\to \tau_{*}(J\ \bar{\epsilon}_{*}) \\
&\to^{*} \tau_{*}(\lambda y.\bar{\epsilon}_{*\ (Jy)}) \\
&\to \tau_{*}(\bar{\epsilon}_{*}\ (J\Omega)) \\
&\to \tau_{*}(\bar{\epsilon}_{*}) \\
&\to \epsilon
\end{aligned}$$

# Böhm trees and tests

## Typing of $\vdash J : \mu \to \mu$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\overline{* \leqslant *}}{y : *, z : \varnothing \vdash y\ (J\ z) : *}}{y : * \vdash \lambda z.y\ (J\ z) : *}}{y : * \vdash J\ y : *} \quad \overline{* \leqslant *}}{x : \mu, y : *, \vdash x\ (J\ y) : *}}{\dfrac{\vdash \lambda xy.x\ (J\ y) : \mu \to \mu}{\vdash J : \mu \to \mu}}$$

## Böhm tree

$$\lambda x^{\mu} y^{*}.x^{\mu}\ .$$
$$|$$
$$\lambda z^{\varnothing}.y^{*}\ \Omega^{\varnothing}$$

## Completion of $\{*, \mu\}$

$$* = \varnothing \to * \qquad \mu = \{*\} \to *$$

## Derivation of $\tau_{\mu \to \mu}(J)$

$$\begin{aligned}
\tau_{\mu\to\mu}(J) &\to^{*} \tau_{\mu\to\mu}(\lambda xy.x\ (J\ y)) \\
&\to^{2} \tau_{*}(\bar{\epsilon}_{\mu}\ (J\ \bar{\epsilon}_{*})) \\
&\to \tau_{*}(\bar{\tau}_{*}(\tau_{*}(J\ \bar{\epsilon}_{*}))) \\
&\to \tau_{*}(J\ \bar{\epsilon}_{*}) \\
&\to^{*} \tau_{*}(\lambda y.\bar{\epsilon}_{*\ (Jy)}) \\
&\to \tau_{*}(\bar{\epsilon}_{*}\ (J\Omega)) \\
&\to \tau_{*}(\bar{\epsilon}_{*}) \\
&\to \epsilon
\end{aligned}$$

# Formalisation of $B_{\mathcal{S}}LL$

## Ordered semiring $(|\mathcal{S}|, +, 0, *, 1, \leqslant)$

$+$ and $*$ are associative

$+$ is commutative and distribute over $*$

0 is neutral for $+$ and 1 is neutral for $*$

$+$ and $*$ are monotone for $\leqslant$.

## Examples

$(\mathbb{B}ool, \vee, ff, \wedge, tt, \{ff \leqslant tt\})$

$(\mathbb{N}, +_{\mathbb{N}}, 0_{\mathbb{N}}, *_{\mathbb{N}}, 1_{\mathbb{N}}, \leqslant_{\mathbb{N}})$

## lax-semiring

We can relax:

$(I + J)\cdot K \leqslant (I\cdot K) + (J\cdot K)$

## Formulas and Sequents $\Gamma \vdash A$

(types) $\quad A, B, C := \alpha \mid A \otimes B \mid A \multimap B \mid A^I \qquad , \forall I \in \mathcal{S}$

$$\frac{\Gamma \vdash B}{\Gamma, A^0 \vdash B}\ \texttt{Weak} \qquad \frac{\Gamma, A \vdash B}{\Gamma, A^1 \vdash B}\ \texttt{Der} \qquad \frac{\Gamma, A^I, A^J \vdash B}{\Gamma, A^{I+J} \vdash B}\ \texttt{Contr}$$

$$\frac{A_1{}^{I_1}, \cdots, A_n{}^{I_n} \vdash B}{A_1{}^{I_1 * J}, \cdots A_n{}^{I_n * J} \vdash B^J}\ \texttt{J-Prom} \qquad \frac{\Gamma, A^I \vdash B \qquad J \geqslant I}{\Gamma, A^J \vdash B}\ \mathcal{S}\texttt{wL}$$

# Brunel&al's categorical semantic

## Ordered semiring $\mathcal{S}$ as a bimonoidal category

The preordered dual category:

Objects: $I, J, K \in \mathcal{S}$ Morphisms: $\mathcal{S}[I, J]$ singleton if $I \geqslant J$

The sum and product are two monoidal products

## Bounded exponential situation

a symmetric monoidal category (model of IMLL) $(\mathcal{A}, \otimes, \multimap, \mathbf{1})$,

a bifunctor: $(\_)^{-} : \mathcal{A} \times \mathcal{S} \to \mathcal{A}$

6 natural transformations:

$$p' : A^{I*J} \Longrightarrow (A^I)^J \qquad d' : A^1 \Longrightarrow A$$

$$c' : A^{I+J} \Longrightarrow A^I \otimes A^J \qquad w' : A^0 \Longrightarrow \mathbf{1}$$

$$m' : A^I \otimes B^I \Longrightarrow (A \otimes B)^I \qquad n' : \mathbf{1} \Longrightarrow \mathbf{1}^I$$

plus 20 commutative diagrams.

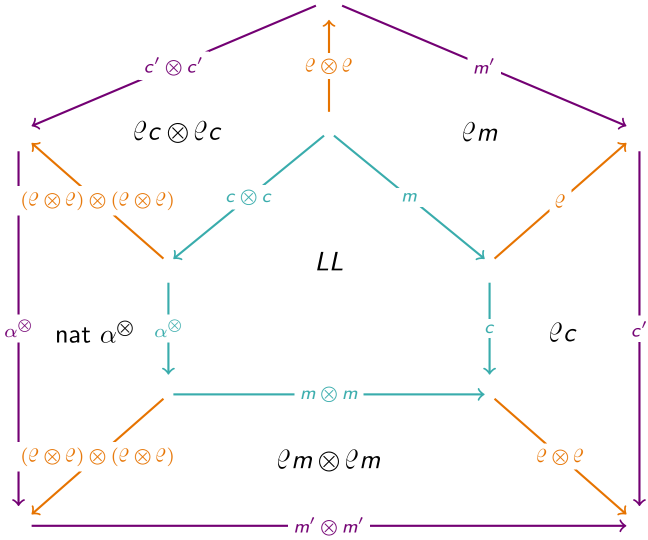
# Stratification diagrams

$$\begin{array}{ccc} !A & \xrightarrow{\mathtt{d}_A} & A \\ \varrho_{A,1}\big\downarrow & \varrho_{\mathtt{d}} & \big\| \\ A^1 & \xrightarrow[\mathtt{d}'_A]{} & A \end{array} \qquad \begin{array}{ccc} !A & \xrightarrow{\mathtt{w}_A} & \mathbf{1} \\ \varrho_{A,0}\big\downarrow & \varrho_{\mathtt{w}} & \big\| \\ A^0 & \xrightarrow[\mathtt{w}'_A]{} & \mathbf{1} \end{array} \qquad \begin{array}{ccc} \mathbf{1} & \xrightarrow{\mathtt{m}_{\mathbf{1}}} & !\mathbf{1} \\ \big\| & \varrho_{\mathtt{m}_{\mathbf{1}}} & \big\downarrow \varrho_I \\ \mathbf{1} & \xrightarrow[\mathtt{m}'_{\mathbf{1},I}]{} & \mathbf{1}^I \end{array}$$

$$\begin{array}{ccc} !A & \xrightarrow{\mathtt{p}_A} & !!A \\ & & \big\downarrow !(\varrho_{A,I}) \\ \varrho_{A,I\cdot J,}\big\downarrow & \varrho_{\mathtt{p}} & !(A^I) \\ & & \big\downarrow \varrho_{A^I,J} \\ A^{I\cdot J} & \xrightarrow{\mathtt{p}'_{A,I,J}} & {A^I}^J \end{array} \qquad \begin{array}{ccc} !A & \xrightarrow{\mathtt{c}_A} & !A \otimes !A \\ \varrho_{A,I+J}\big\downarrow & \varrho_{\mathtt{c}} & \big\downarrow \varrho\otimes\varrho \\ A^{I+J} & \xrightarrow{\mathtt{c}'_{A,I,J}} & A^I \otimes B^J \end{array} \qquad \begin{array}{ccc} !A \otimes !B & \xrightarrow{\mathtt{m}_{A,B}} & !(A \otimes B) \\ \varrho\otimes\varrho\big\downarrow & \varrho_{\mathtt{m}} & \big\downarrow \varrho_{A\otimes B,I} \\ A^I \otimes B^I & \xrightarrow{\mathtt{m}'_{A,B,I,J}} & (A\otimes B)^I \end{array}$$

($\mathtt{d}', \mathtt{p}', \mathtt{w}', \mathtt{c}'$ and $\mathtt{m}'$ are uniquely determined)

# An example of required diagram



# Stratifications

Stratifications of the reational model with the free exponential:

**Semiring interpretation**
$[\![-]\!] : \mathcal{S} \to \mathcal{P}(\mathbb{N})$

$I \leqslant J$ implies $[\![I]\!] \subseteq [\![J]\!]$,

$[\![I]\!] \oplus [\![J]\!] \subseteq [\![I+J]\!]$, $\{0\} \subseteq [\![0_{\mathcal{S}}]\!]$,
$[\![I]\!] \odot [\![J]\!] \subseteq [\![I*J]\!]$, $\{1\} \subseteq [\![1_{\mathcal{S}}]\!]$.

***Rel*: a model of $\mathrm{B}_{\mathcal{S}}\mathrm{LL}$**
(given $[\![-]\!] : \mathcal{S} \to \mathcal{P}(\mathbb{N})$)

**Bounded exponential:**
$A^J = \{[a_1, \ldots, a_n] \in \mathcal{M}_f(A) \mid n \in [\![J]\!]\}$
**epi transformation:**
$\varrho_{J,A} = \{(u, u) \mid u \in A^J\}$

Stratifications of the reational model with the non-free exponentials:

**Semiring interpretation**
$[\![-]\!] : \mathcal{S} \to \mathcal{P}(\mathcal{R})$

**$Rel^{\mathcal{R}}$ : a model of $\mathrm{B}_{\mathcal{S}}\mathrm{LL}$**
(given $[\![-]\!] : \mathcal{S} \to \mathcal{P}(\mathcal{R})$)

**Surprising emergence of (lax-)semirings**

$\mathcal{P}(\mathbb{N})$, $\mathcal{P}(\overline{\mathbb{N}})$, $\mathcal{P}(Poly)$, $\mathcal{P}(\mathbb{N}_f\langle Aff_1^c\rangle)$, $\mathcal{P}(\mathbb{N}_f\langle\mathbb{R}^+\rangle)$, $\mathcal{P}(\mathbb{B})$...

# The model Rel of LL

## The category *Rel* : a model of IMLL

**Objects:** Sets **Morphisms:** Relations

**Exponential:** $!A = \mathcal{M}_f(A)$

## Semiring interpretation $[\![-]\!] : \mathcal{S} \to \mathcal{P}(\mathbb{N})$

$I \leqslant J$ implies $[\![I]\!] \subseteq [\![J]\!]$,

$[\![I]\!] \oplus [\![J]\!] \subseteq [\![I+J]\!], \quad \{0\} \subseteq [\![0_\mathcal{S}]\!]$,

$[\![I]\!] \odot [\![J]\!] \subseteq [\![I*J]\!], \quad \{1\} \subseteq [\![1_\mathcal{S}]\!]$.

## *Rel* : a model of $B_\mathcal{S}LL$ (given $[\![-]\!] : \mathcal{S} \to \mathcal{P}(\mathbb{N})$)

**Bounded exponential:**

$A^J = \{[a_1, ...., a_n] \in \mathcal{M}_f(A) \mid n \in [\![J]\!]\}$

**epi transformation:**

$\varrho_{J,A} = \{(u, u) \mid u \in A^J\}$

# The model Rel of LL

## The category *Rel* : a model of IMLL

**Objects:** Sets    **Morphisms:** Relations

**Exponential:** $!A = \mathcal{M}_f(A)$

## Semiring interpretation $[\![-]\!] : \mathcal{S} \to \mathcal{P}(\mathbb{N})$

$I \leqslant J$ implies $[\![I]\!] \subseteq [\![J]\!]$,

$[\![I]\!] \oplus [\![J]\!] \subseteq [\![I+J]\!]$, $\{0\} \subseteq [\![0_{\mathcal{S}}]\!]$,

$[\![I]\!] \odot [\![J]\!] \subseteq [\![I*J]\!]$, $\{1\} \subseteq [\![1_{\mathcal{S}}]\!]$.

## *Rel* : a model of B$_{\mathcal{S}}$LL (given $[\![-]\!] : \mathcal{S} \to \mathcal{P}(\mathbb{N})$)

**Bounded exponential:**

$A^J = \{[a_1, ...., a_n] \in \mathcal{M}_f(A) \mid n \in [\![J]\!]\}$

**epi transformation:**

$\varrho_{J,A} = \{(u, u) \mid u \in A^J\}$

## Exemples of semiring interpretations

| | | | |
|---|---|---|---|
| $(\mathbb{B}ool, ff \leqslant tt)$ | $[\![ff]\!] := \{0\}$, $[\![tt]\!] := \mathbb{N}$ | $A^{ff} = \{[]\}$ | $A^{tt} = !A$ |
| $(\mathbb{B}ool, id)$ | $[\![ff]\!] := \{0\}$, $[\![tt]\!] := [1, \infty[$ | $A^{ff} = \{[]\}$ | $A^{tt} = !A - \{[]\}$ |
| $(\mathbb{N}, id)$ | $[\![n]\!] := \{n\}$ | $A^n = \{[a_1, ..., a_n]\}$ | |
| $(\mathbb{N}, \leqslant_{\mathbb{N}})$ | $[\![n]\!] := [0, n]$ | $A^n = \{[a_1, ..., a_m] \mid m \leqslant n\}$ | |

# The model Rel$^{\mathcal{R}}$ of LL [Carraro&Al]

**Multiplicity semiring $\mathcal{R}$**

A semiring with coalgebraic constraints (associativity of additive splitting...)

**$Rel^{\mathcal{R}}$ : a model of ILL**

**Exponential:** $!A = \mathcal{R}_f\langle A\rangle$

the set of finitely supported functions from $A$ to $\mathcal{R}$.

(It is a generalization of Rel$^{\mathbb{N}}$ since $\mathcal{M}_f(A) = \mathbb{N}_f\langle A\rangle$.)

**Interpretation $[\![-]\!] : \mathcal{S} \to \mathcal{P}(\mathcal{R})$**

Similar to the interpretation in $\mathcal{P}(\mathbb{N})$.

**Universal interpretation**

$[\![.]\!] : \mathcal{S} \to \mathcal{P}(\mathcal{M}_f(\mathcal{S}_*))$

**$Rel^{\mathcal{R}}$ : a model of B$_{\mathcal{S}}$LL**

**Bounded exponential:** $A^J = \{u \in \mathcal{R}_f\langle A\rangle \mid \Sigma_{a\in A} u(a) \in [\![J]\!]\}$

**epimorphism:** $\varrho_{J,A} = \{(u,u) \mid u \in A^J\}$

# Examples and interest

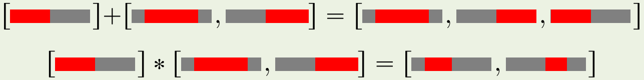
## Counting resources

$\mathbb{N}$: if $t : A^{n} \multimap U$ then a CbNAM on $(t\ s)$ will evaluate at most $n$ times the program $s$.

$\mathbb{P}oly$: if $t$ is typable then $(t\ s)$ has a polynomial head reduction on the size on $s$.

$\mathbb{R}^{+}$: if $t : A^{r} \multimap U$ then the expected value over the number of evaluations of $s$ in the execution of $(t\ s)$ is at most $r$ (in presence of probabilistic operations).

## A more complicated example: Ghica&Smith's



This computes the sequentially of an execution (with scheduling op).
A term $t : A^{[\ ,\ ]} \multimap B$ will use its argument two times: once during the first third of the execution time and once during the last third.

# The operations $\oplus$ and $\odot$

## The cs-semiring $\mathcal{P}(\mathbb{N})$

$p \oplus q = \{m + n \mid \forall m \in p,\ \forall n \in q\}$
$p \odot q = \{n_1 + \ldots + n_m \mid \forall m \in p,\ \forall n_1, \ldots, n_m \in q\}$

## Not a full semiring

$$(\{1\}\oplus\{1\}) \odot \{1,2\} = \{2,4\}, \quad (\{1\}\odot\{1,2\}) \oplus (\{1\}\odot\{1,2\}) = \{2,3,4\}$$

## The cs-semiring $\mathcal{P}(\mathcal{R})$

$$\alpha \oplus \beta := \{p + q \mid p \in \alpha, q \in \beta\},$$

$$\alpha \odot \beta := \left\{\sum_{i=1}^{h} p_i \cdot q_i \mid h \geqslant 0, \sum_{i=1}^{h} q_i \in \beta, \forall i \leqslant h, p_i \in \alpha\right\}.$$

# Multiplicity [Carraro&Al]

## A semiring $\mathcal{R}$ has multiplicities if

(MS1) it is positive: $p+q = 0 \;\Rightarrow\; p = q = 0$

(MS2) it is discreet: $p+q = 1 \;\Rightarrow\; p = 0$ or $q = 0$

(MS3) it has additive splitting properties.

$$p_1+q_1 = p_2+q_2 \;\Rightarrow\; \exists (r_{ij})_{1\leqslant i,j\leqslant 2}, p_i = r_{i1}+r_{i2},\; q_j = r_{1j}+r_{2j}.$$

(MS4) it has multiplicative k-ary splitting property:

$$q_1+q_2 = rp \;\Rightarrow \exists k, r_1, \ldots, r_k, p_{1,1}, \ldots, p_{1,k}, p_{2,1}, \ldots, p_{2,k},$$

$$r = \sum_{j\leqslant k} r_j,\;\; q_i = \sum_{j\leqslant k} r_j p_{i,j},\;\; \forall j \leqslant k, p = p_{1,j}+p_{2,j}$$

### Examples

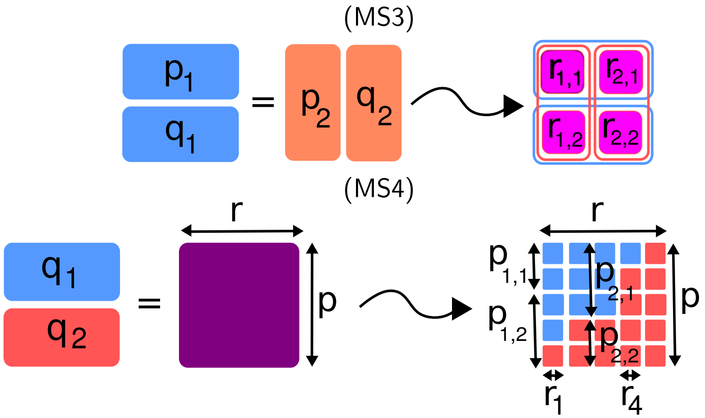
| | |
|---|---|
| $\mathbb{N}$ | $\mathcal{M}_f(\mathbb{M})$ |
| $\bar{\mathbb{N}}$ | $Poly/_{(X^2=0)}$ |

### $\mathcal{R}$ morally behave like $\mathbb{N}$

$\mathcal{R}$ contain $\mathbb{N}$ as sub semiring

The sum in $\mathcal{R}$ is the one of $\mathbb{N}$

# Multiplicity [Carraro&Al]



# The model $\text{Rel}^{\mathcal{R}}$ of LL [Carraro&Al]

## Definition of $\mathcal{R}_f\langle A\rangle$

Given a semiring $\mathcal{R}$ and a set $A$, the space of functions $f : A \to \mathcal{R}$ of **finite support** is denoted $\mathcal{R}_f\langle A\rangle$. It is a semimodule over $\mathcal{R}$ with:

A commutative sum: $(f+g)(x) = f(x)+g(x)$

An external product: $(I\cdot f)(x) = I\cdot f(x)$

## $Rel^{\mathcal{R}}$ : a model of MELL

**Exponential:** $!A = \mathcal{R}_f\langle A\rangle$

$$\delta_A = \{(u, V) \mid V \in !!A, u = \Sigma_{v\in !A} V(v)\cdot v\} \qquad \epsilon_A = \{([1\cdot a], a) \mid a \in A\}$$

$$c_A = \{(u, (v, w) \mid u = v+w \in !A\} \quad w_A = \{([], *)\} \quad n = \{(*, [J\cdot *])|J\in\mathcal{R}\}$$

$$m_{A,B} = \{((u, v), w) \mid \forall a, u(a) = \Sigma_{b\in B} w(a, b), \forall b, v(b) = \Sigma_{a\in A} w(a, b)\}$$