Towards a resource based approximation theory of programs

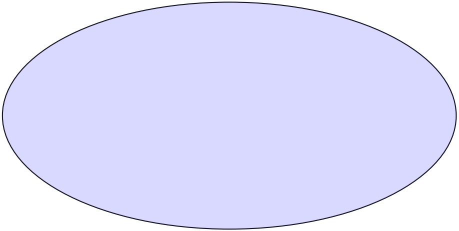
Soutenance de thèse de **Davide Barbarossa**

barbarossa@lipn.univ-paris13.fr

https://lipn.univ-paris13.fr/~barbarossa/

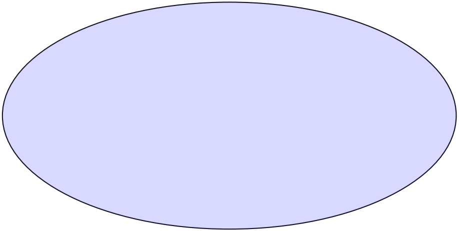
Laboratoire d'Informatique Paris-Nord, Université Sorbonne Paris Nord
Dipartimento di matematica e fisica, Università Roma Tre

Encadrants: Giulio Manzonetto Lorenzo Tortora de Falco



Mathematics

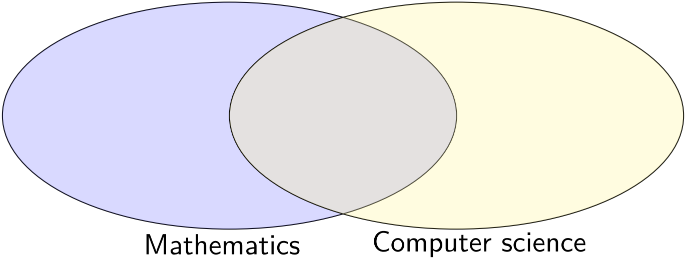
Me:
"What is a proof?"



Mathematics

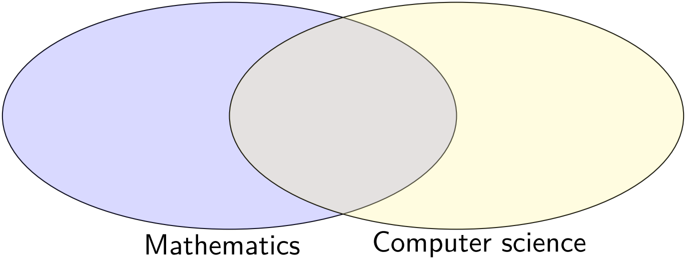
Me:
"What is a proof?"

BHK/Curry-Howard/Realizability:
"A program!"



Me:
"What is a proof?"

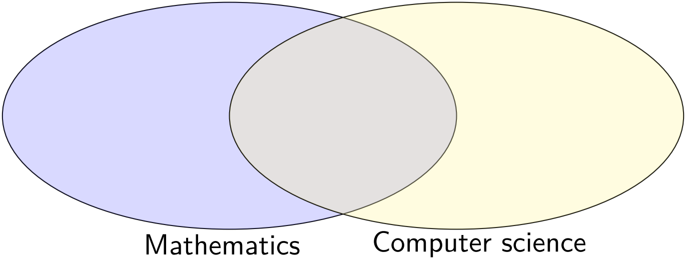
BHK/Curry-Howard/Realizability:
"A program!"



We can go even deeper (see Girard)... but this is another story

Me:
"What is a proof?"

BHK/Curry-Howard/Realizability:
"A program!"



We can go even deeper (see Girard)... but this is another story

Curry-Howard

$$\dfrac{\dfrac{\pi : A \vdash B}{\vdash A \to B} \qquad \rho : \ \vdash A}{\vdash B}\, cut \quad \longrightarrow \quad \pi\{\rho / \overline{A \vdash A}\} : \ \vdash B$$

Curry-Howard

$$\dfrac{\dfrac{\pi : A^x \vdash B}{\lambda x.\pi :\ \vdash A \to B} \qquad \rho :\ \vdash A}{(\lambda x.\pi)\rho :\ \vdash B}\,cut \quad \longrightarrow \quad \pi\{\rho/x\} :\ \vdash B$$

Curry-Howard

$$\dfrac{\dfrac{\pi : A^x \vdash B}{\lambda x.\pi : \ \vdash A \to B} \qquad \rho : \ \vdash A}{(\lambda x.\pi)\rho : \ \vdash B}\,\mathit{cut} \quad \longrightarrow \quad \pi\{\rho/x\} : \ \vdash B$$

This is not Turing-complete!

Curry-Howard

$$\dfrac{\dfrac{\pi : A^x \vdash B}{\lambda x.\pi : \ \vdash A \to B} \qquad \rho : \ \vdash A}{(\lambda x.\pi)\rho : \ \vdash B}\,cut \quad \longrightarrow \quad \pi\{\rho/x\} : \ \vdash B$$

This is not Turing-complete!

What's the underlying untyped programming language?

Curry-Howard

$$\dfrac{\dfrac{\pi : A^x \vdash B}{\lambda x.\pi : \ \vdash A \to B} \qquad \rho : \ \vdash A}{(\lambda x.\pi)\rho : \ \vdash B}\, cut \quad \longrightarrow \quad \pi\{\rho/x\} : \ \vdash B$$

This is not Turing-complete!

What's the underlying untyped programming language?

λ-calculus

$$\begin{aligned} M \ ::= \ & x && \textit{(datas or place holders)} \\ & \mid \lambda x.M && \textit{(function of the variable } x \textit{ given by the ``law'' } M\textit{)} \\ & \mid MM && \textit{(function application)} \end{aligned}$$

$$(\lambda x.M)N \longrightarrow_\lambda M\{N/x\}$$

Curry-Howard

$$\dfrac{\dfrac{\pi : A^x \vdash B}{\lambda x.\pi : \vdash A \to B} \qquad \rho : \vdash A}{(\lambda x.\pi)\rho : \vdash B}\,cut \quad \longrightarrow \quad \pi\{\rho/x\} : \vdash B$$

This is not Turing-complete!

What's the underlying untyped programming language?

λ-calculus   This is Turing-complete!

$M ::= x$ *(datas or place holders)*
$\mid \lambda x.M$ *(function of the variable x given by the "law" M)*
$\mid MM$ *(function application)*

$$(\lambda x.M)N \longrightarrow_{\lambda} M\{N/x\}$$

$(\lambda x.\, \texttt{add}\, 17\, (\texttt{multiply}\, x\, x))\, 5$

$(\lambda x.\, \mathtt{add}\, 17\, (\mathtt{multiply}\, x\, x))\ 5$

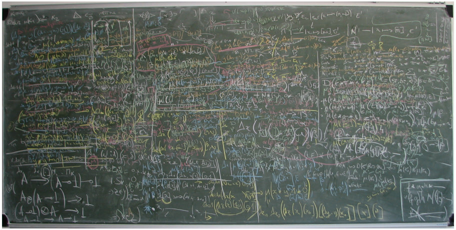
$(\lambda x.\ \texttt{add}\ 17\ (\texttt{multiply}\ x\ x))\ 5$

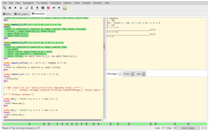
$(\lambda x.\ \texttt{add}\ 17\ (\texttt{multiply}\ x\ x))\ 5$

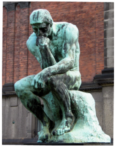
$$(\lambda x.\ \mathtt{add}\ 17\ (\mathtt{multiply}\ x\ x))\ 5 \longrightarrow_{\lambda} \mathtt{add}\ 17\ (\mathtt{multiply}\ 5\ 5)$$

$$(\lambda x.\ \texttt{add}\,17\,(\texttt{multiply}\ x\ x))\ 5 \longrightarrow_\lambda \texttt{add}\,17\,(\texttt{multiply}\ 5\ 5) \qquad (= 42)$$

$$(\lambda x.\ \texttt{add}\ 17\ (\texttt{multiply}\ x\ x))\ 5 \longrightarrow_{\lambda} \texttt{add}\ 17\ (\texttt{multiply}\ 5\ 5) \qquad (= 42)$$







Taylor expansion of a program (Ehrhard-Regnier, '00s)

$(\lambda x.\,\texttt{add}\,17\,(\texttt{multiply}\,x\,x))\,5 \longrightarrow_\lambda \texttt{add}\,17\,(\texttt{multiply}\,5\,5)$

Taylor expansion of a program (Ehrhard-Regnier, '00s)

$(\lambda x.\, \texttt{add}\, 17\, (\texttt{multiply}\, x\, x))\, 5 \longrightarrow_\lambda \texttt{add}\, 17\, (\texttt{multiply}\, 5\, 5)$

$(\lambda x.\, \texttt{add}\, [17]\, [\texttt{multiply}\, [x]\, [x]])\, [5, 5] \longrightarrow_\lambda \texttt{add}\, [17]\, [\texttt{multiply}\, [5]\, [5]]$

Taylor expansion of a program (Ehrhard-Regnier, '00s)

$(\lambda x.\, \texttt{add}\, 17\, (\texttt{multiply}\, x\, x))\, 5 \longrightarrow_\lambda \texttt{add}\, 17\, (\texttt{multiply}\, 5\, 5)$

$(\lambda x.\, \texttt{add}\, [17]\, [\texttt{multiply}\, [x]\, [x]])\, [5, 5] \longrightarrow_\lambda \texttt{add}\, [17]\, [\texttt{multiply}\, [5]\, [5]]$

$(\lambda x.\, \texttt{add}\, [17]\, [\texttt{multiply}\, [x]\, [x]])\, [5, 5, 5] \longrightarrow_\lambda \mathrm{error}$

Taylor expansion of a program (Ehrhard-Regnier, '00s)

$(\lambda x.\, \texttt{add}\, 17\, (\texttt{multiply}\, x\, x))\, 5 \longrightarrow_\lambda \texttt{add}\, 17\, (\texttt{multiply}\, 5\, 5)$

$(\lambda x.\, \texttt{add}\, [17]\, [\texttt{multiply}\, [x]\, [x]])\, [5, 5] \longrightarrow_\lambda \texttt{add}\, [17]\, [\texttt{multiply}\, [5]\, [5]]$

$(\lambda x.\, \texttt{add}\, [17]\, [\texttt{multiply}\, [x]\, [x]])\, [5, 5, 5] \longrightarrow_\lambda \mathrm{error}$

$(\lambda x.\, \texttt{add}\, [17]\, [\texttt{multiply}\, [x]\, [x]])\, [\,] \longrightarrow_\lambda \mathrm{error}$

Taylor expansion of a program (Ehrhard-Regnier, '00s)

$(\lambda x.\, \texttt{add}\, 17\, (\texttt{multiply}\, x\, x))\, 5 \longrightarrow_\lambda \texttt{add}\, 17\, (\texttt{multiply}\, 5\, 5)$

$(\lambda x.\, \texttt{add}\, [17]\, [\texttt{multiply}\, [x]\, [x]])\, [5, 5] \longrightarrow_\lambda \texttt{add}\, [17]\, [\texttt{multiply}\, [5]\, [5]]$

$(\lambda x.\, \texttt{add}\, [17]\, [\texttt{multiply}\, [x]\, [x]])\, [5, 5, 5] \longrightarrow_\lambda \mathrm{error}$

$(\lambda x.\, \texttt{add}\, [17]\, [\texttt{multiply}\, [x]\, [x]])\, [\,] \longrightarrow_\lambda \mathrm{error}$

Resource λ-calculus

$$t ::= x \mid \lambda x.t \mid t\,[t, \ldots, t]$$

Qualitative Taylor expansion

$$\mathcal{T}(MN) = \{t[u_1, \ldots, u_k] \mid k \in \mathbb{N},\ t \in \mathcal{T}(M),\ u_1, \ldots, u_k \in \mathcal{T}(N)\}$$

Taylor expansion of a program (Ehrhard-Regnier, '00s)

Resource λ-calculus

$$t ::= x \mid \lambda x.t \mid t\,[t, \ldots, t]$$

Qualitative Taylor expansion

$$\mathcal{T}(MN) = \{t[u_1, \ldots, u_k] \mid k \in \mathbb{N},\ t \in \mathcal{T}(M),\ u_1, \ldots, u_k \in \mathcal{T}(N)\}$$

Someone said Taylor?! $\Theta(F)(x) = \sum_n \frac{1}{n!}(\mathtt{D}^{(n)} F \cdot x^n)(0)$ with $(\mathtt{D}^{(n)} F \cdot a)(y) := \frac{d^n F}{dx^n}(y) \cdot a$

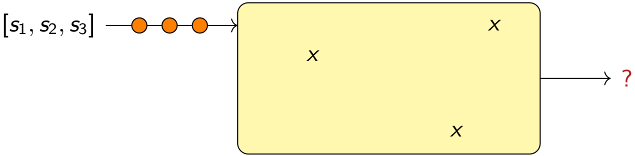
Quantitative Taylor expansion

$$\Theta(M) := \sum_{t \in \mathcal{T}(M)} \frac{1}{\mathrm{m}(t)} t$$

$$\Theta(Fx) = \Theta(F) \sum \frac{1}{n!} x^n = \sum \frac{1}{n!} (\mathtt{D}^{(n)} \Theta(F) \bullet x^n) 0$$

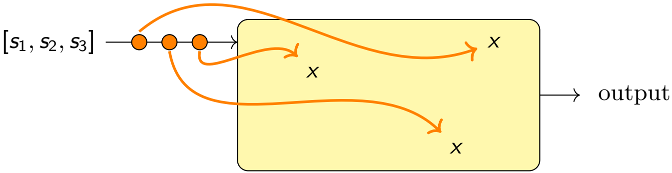
Reduction:

$$(\lambda x.t)[s_1, s_2, s_3] \to ?$$



Reduction:

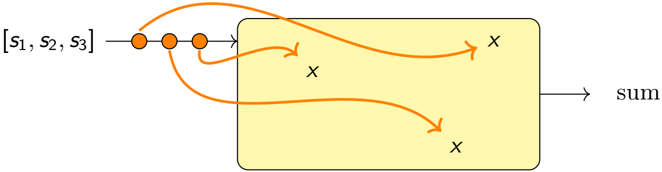
$$(\lambda x.t)[s_1, s_2, s_3] \to t\{s_1/x^{(1)}, s_2/x^{(2)}, s_3/x^{(3)}\}$$



We need formal (*idempotent*) sum $\mathbb{T} = t_1 + \cdots + t_n$ of resource terms.

Reduction:

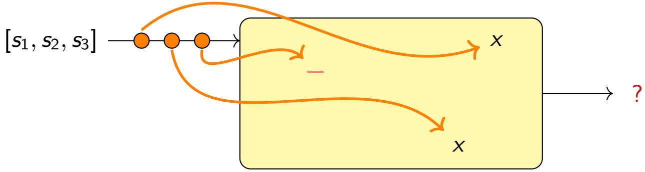
$$(\lambda x.t)[s_1, s_2, s_3] \to \sum_{\sigma \in \mathfrak{S}_3} t\{s_{\sigma(1)}/x^{(1)}, s_{\sigma(2)}/x^{(2)}, s_{\sigma(3)}/x^{(3)}\}$$



We need formal (*idempotent*) sum $\mathbb{T} = \ t_1 + \cdots + t_n$ of resource terms.

Reduction:

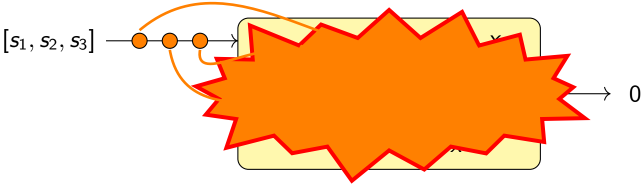
$$(\lambda x.t)[s_1, s_2, s_3] \rightarrow \ ?$$



We need formal (*idempotent*) sum $\mathbb{T} = t_1 + \cdots + t_n$ of resource terms.

Reduction:

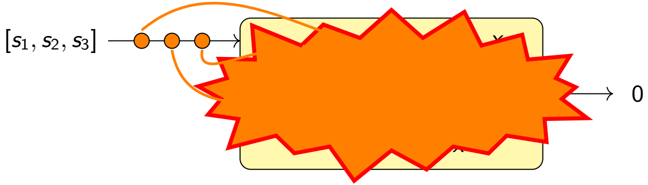
$$(\lambda x.t)[s_1, s_2, s_3] \to 0$$



We need formal (*idempotent*) sum $\mathbb{T} = t_1 + \cdots + t_n$ of resource terms.

Reduction:

$$(\lambda x.t)[s_1, s_2, s_3] \to 0$$



Main Properties

- Linearity: no erase/duplicate non-empty bags (unless $\twoheadrightarrow 0$).
- Strong Normalisation: trivial, erases exactly one λ.
- Confluence: locally confluent + strongly normalising.

Böhm trees (Barendregt, '70s)

If M is unsolvable then $\mathrm{BT}(M) := \bot$. If $M \twoheadrightarrow_{\mathrm{h}} \lambda\vec{x}.y\, Q_1 \dots Q_k$ then:

$\mathrm{BT}(M) :=$ $\lambda\vec{x}.y$

$\mathrm{BT}(Q_1) \quad \cdots \quad \mathrm{BT}(Q_k)$

Coinduction!

Set $\mathcal{A}$ of Böhm approximants: $P \;::=\; \bot \mid \lambda\vec{x}.y\, P \dots P$

$\mathcal{A}$ is endowed with a preorder $\sqsubseteq$ generated by taking $\bot \sqsubseteq P$ for all P

Set $\mathcal{A}(M)$ of the Böhm approximants of $M \in \Lambda$:

$$\mathcal{A}(M) := \{P \in \mathcal{A} \mid \exists N \in \Lambda \text{ s.t. } M \twoheadrightarrow_{\lambda} N \sqsupseteq P\}$$

Approximation Theorem

$$\mathrm{BT}(M) = \sup_{P \in \mathcal{A}(M)} P$$

Böhm trees (Barendregt, '70s)

If M is unsolvable then $\mathrm{BT}(M) := \bot$. If $M \twoheadrightarrow_{\mathrm{h}} \lambda\vec{x}.y\, Q_1 \ldots Q_k$ then:

$\mathrm{BT}(M) :=$ $\lambda\vec{x}.y$

$\mathrm{BT}(Q_1) \quad \cdots \quad \mathrm{BT}(Q_k)$

Coinduction!

Understanding the relation between the term and its full Taylor expansion might be the starting point of a renewing of the theory of approximations.

T. Ehrhard, L. Regnier ('03)

The differential lambda-calculus

Set $\mathcal{A}$ of

$\mathcal{A}$ is endo | P

Set $\mathcal{A}(M)$

$$\mathcal{A}(M) := \{P \in \mathcal{A} \mid \exists N \in \Lambda \text{ s.t. } M \twoheadrightarrow_\lambda N \sqsupseteq P\}$$

Approximation Theorem

$$\mathrm{BT}(M) = \sup_{P \in \mathcal{A}(M)} P$$

Classic results via labelled reduction

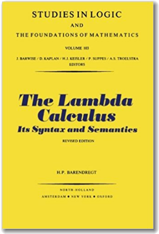
Scott's Continuity → Berry's Stability

Scott's Continuity → Contextuality of BTs

Scott's Continuity → Genericity Property

topological argument

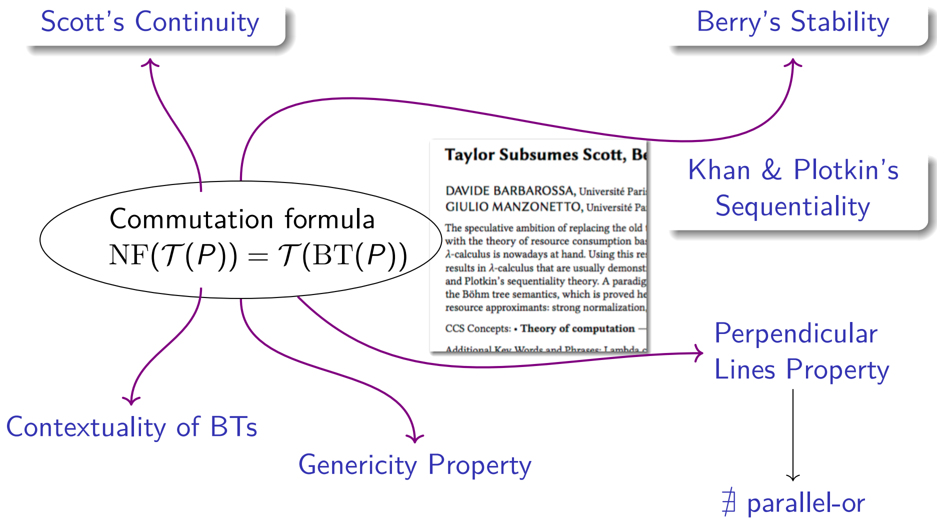
Chapter 14



Khan & Plotkin's Sequentiality → Berry's Stability

Khan & Plotkin's Sequentiality → Perpendicular Lines Property → $\nexists$ parallel-or

Classic results via Resource Approximation



Unsolvables are computationally meaningless

Genericity Property

Let U unsolvable. If $\exists\, \mathrm{nf}(C(\!| U |\!))$, then C is constant on $\Lambda/_{=_\lambda}$.

Unsolvables are computationally meaningless

Genericity Property

Let U unsolvable. If $\exists\, \mathrm{nf}(C(\!|U|\!))$, then C is constant on $\Lambda/_{=_\lambda}$.

Proof. $C(\!|U|\!)$ normalisable $\quad\Rightarrow\quad \exists t \in \mathrm{NF}(\mathcal{T}(C(\!|U|\!)))$ such that:

$$\text{“}\mathrm{nf}_\beta(C(\!|U|\!)) = t\text{” and all its bags are singletons.}$$

So $\exists t' \in \mathcal{T}(C(\!|U|\!))$ such that:

$$t' = c(\!|s_1, \ldots, s_k|\!) \longrightarrow\!\!\!\!\rightarrow t + \mathbb{T}$$

for some $c \in \mathcal{T}(C(\!|\cdot|\!))$ and $s_1, \ldots, s_k \in \mathcal{T}(U)$.

Unsolvables are computationally meaningless

Genericity Property

Let U unsolvable. If $\exists\, \mathrm{nf}(C\llparenthesis U \rrparenthesis)$, then C is constant on $\Lambda/_{=_\lambda}$.

Proof. $C\llparenthesis U \rrparenthesis$ normalisable $\quad \Rightarrow \quad \exists t \in \mathrm{NF}(\mathcal{T}(C\llparenthesis U \rrparenthesis))$ such that:

$$\text{“}\mathrm{nf}_\beta(C\llparenthesis U \rrparenthesis) = t\text{” and all its bags are singletons.}$$

So $\exists t' \in \mathcal{T}(C\llparenthesis U \rrparenthesis)$ such that:

$$\begin{array}{ccc} t' = c\llparenthesis s_1, \ldots, s_k \rrparenthesis & \twoheadrightarrow & t + \mathbb{T} \\ \downarrow\!\!\downarrow & \nearrow\!\!\!\nearrow & \\ c\llparenthesis \mathrm{nf}(s_1), \ldots, \mathrm{nf}(s_k) \rrparenthesis & & \end{array}$$

for some $c \in \mathcal{T}(C\llparenthesis \cdot \rrparenthesis)$ and $s_1, \ldots, s_k \in \mathcal{T}(U)$.

Unsolvables are computationally meaningless

Genericity Property

Let U unsolvable. If $\exists\,\mathrm{nf}(C\llparenthesis U\rrparenthesis)$, then C is constant on $\Lambda/_{=_\lambda}$.

Proof. $C\llparenthesis U\rrparenthesis$ normalisable $\quad\Rightarrow\quad \exists t \in \mathrm{NF}(\mathcal{T}(C\llparenthesis U\rrparenthesis))$ such that:

$$\text{“}\mathrm{nf}_\beta(C\llparenthesis U\rrparenthesis) = t\text{” and all its bags are singletons.}$$

So $\exists t' \in \mathcal{T}(C\llparenthesis U\rrparenthesis)$ such that:

$$\begin{array}{ccc} t' = c\llparenthesis s_1, \dots, s_k\rrparenthesis & \twoheadrightarrow & t + \mathbb{T} \\ \downarrow\!\!\downarrow & \nearrow\!\!\nearrow & \\ c\llparenthesis 0, \dots, 0\rrparenthesis & & \end{array}$$

for some $c \in \mathcal{T}(C\llparenthesis\cdot\rrparenthesis)$ and $s_1, \dots, s_k \in \mathcal{T}(U)$. $\quad(U \text{ unsolvable } \Rightarrow \mathrm{nf}(s_i) = 0)$

Unsolvables are computationally meaningless

Genericity Property

Let U unsolvable. If $\exists\,\mathrm{nf}(C(\!| U |\!))$, then C is constant on $\Lambda/_{=_\lambda}$.

Proof. $C(\!| U |\!)$ normalisable $\quad\Rightarrow\quad$ $\exists t \in \mathrm{NF}(\mathcal{T}(C(\!| U |\!)))$ such that:

$$\text{“}\mathrm{nf}_\beta(C(\!| U |\!)) = t\text{” and all its bags are singletons.}$$

So $\exists t' \in \mathcal{T}(C(\!| U |\!))$ such that:

$$\begin{array}{ccc} t' = c(\!| s_1, \ldots, s_k |\!) & \twoheadrightarrow & t + \mathbb{T} \\ \downarrow\!\!\downarrow & \nearrow\!\!\!\nearrow & \\ 0 = c(\!| 0, \ldots, 0 |\!) & & \end{array}$$

for some $c \in \mathcal{T}(C(\!| \cdot |\!))$ and $s_1, \ldots, s_k \in \mathcal{T}(U)$.

Unsolvables are computationally meaningless

Genericity Property

Let U unsolvable. If $\exists\, \mathrm{nf}(C\llparenthesis U \rrparenthesis)$, then C is constant on $\Lambda/_{=_\lambda}$.

Proof. $C\llparenthesis U \rrparenthesis$ normalisable $\Rightarrow$ $\exists t \in \mathrm{NF}(\mathcal{T}(C\llparenthesis U \rrparenthesis))$ such that:

$$\text{“}\mathrm{nf}_\beta(C\llparenthesis U \rrparenthesis) = t\text{” and all its bags are singletons.}$$

So $\exists t' \in \mathcal{T}(C\llparenthesis U \rrparenthesis)$ such that:

$$\begin{array}{ccc} t' = c\llparenthesis s_1, \dots, s_k \rrparenthesis & \twoheadrightarrow & t + \mathbb{T} \\ \downarrow & \nearrow & \\ 0 \neq c\llparenthesis 0, \dots, 0 \rrparenthesis & & \end{array}$$

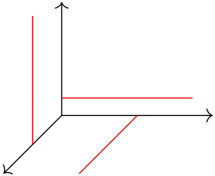
for some $c \in \mathcal{T}(C\llparenthesis \cdot \rrparenthesis)$ and $s_1, \dots, s_k \in \mathcal{T}(U)$. No hole can occur in c!

$$\text{Therefore : } t' = c\llparenthesis s_1, \dots, s_k \rrparenthesis = c \in \mathcal{T}(C\llparenthesis M \rrparenthesis) \text{ and hence } t \in \mathrm{NF}(\mathcal{T}(C\llparenthesis M \rrparenthesis)).$$

Since all bags of t are singletons, “$t = \mathrm{nf}_\beta(C\llparenthesis M \rrparenthesis)$”. □

Perpendicular Lines Property

PLP: *If* $\lambda\vec{z}.F : \Lambda \times \cdots \times \Lambda \to \Lambda$ *is constant (mod ...) on n perpendicular lines, then it must be constant (mod ...) everywhere.*



Perpendicular Lines Property

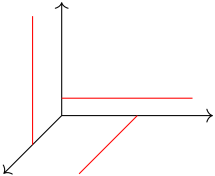
PLP: *If* $\lambda\vec{z}.F : \Lambda \times \cdots \times \Lambda \to \Lambda$ *is constant (mod ...) on n perpendicular lines, then it must be constant (mod ...) everywhere.*

True in $\Lambda/_{=_\mathcal{B}}$, Barendregt's Book 1984
Proof: via Sequentiality.

 in $\Lambda^o/_{=_\mathcal{B}}$

False in $\Lambda^o/_{=_\lambda}$, Barendregt & Statman 1999
Counterexample: via Plotkin's terms.

True in $\Lambda/_{=_\lambda}$, De Vrijer & Endrullis 2008
Proof: via Reduction under Substitution.



PLP	$=_\lambda$	$=_\mathcal{B}$
open	✓	✓
closed	X	?

Idea of the proof

Perpendicular Lines Property

$$\forall Z \begin{cases} (\lambda\vec{z}.F)\ Z\ M_{12}\ \ldots\ldots\ M_{1n} & =_{\mathcal{B}} \quad N_1 \\ (\lambda\vec{z}.F)\ M_{21}\ Z\ \ldots\ldots\ M_{2n} & =_{\mathcal{B}} \quad N_2 \\ \qquad\qquad \ddots & \quad\vdots \qquad \vdots \\ (\lambda\vec{z}.F)\ M_{n1}\ \ldots\ M_{n(n-1)}\ Z & =_{\mathcal{B}} \quad N_n \end{cases} \quad \Rightarrow \quad \forall\vec{Z},\ (\lambda\vec{z}.F)\vec{Z} =_{\mathcal{B}} N_1.$$

Idea of the proof

Perpendicular Lines Property

$$\forall Z \begin{cases} (\lambda\vec{z}.F)\ Z\ M_{12}\ \ldots\ldots\ M_{1n} & =_{\mathcal{B}} & N_1 \\ (\lambda\vec{z}.F)\ M_{21}\ Z\ \ldots\ldots\ M_{2n} & =_{\mathcal{B}} & N_2 \\ \qquad\qquad \ddots & \vdots & \vdots \\ (\lambda\vec{z}.F)\ M_{n1}\ \ldots\ M_{n(n-1)}\ Z & =_{\mathcal{B}} & N_n \end{cases} \quad\Rightarrow\quad \forall\vec{Z},\ (\lambda\vec{z}.F)\vec{Z} =_{\mathcal{B}} N_1.$$

How can $\mathrm{BT}((\lambda z.F)N)$ be independent from N?

1. N is erased during the reduction;
2. N is “hidden” behind an unsolvable;
3. N is “pushed to infinity”.

Idea of the proof

Perpendicular Lines Property

$$\forall Z \begin{cases} (\lambda\vec{z}.F)\ Z\ M_{12}\ \ldots\ldots\ M_{1n} & =_\tau & N_1 \\ (\lambda\vec{z}.F)\ M_{21}\ Z\ \ldots\ldots\ M_{2n} & =_\tau & N_2 \\ \qquad\qquad\ddots & \vdots & \vdots \\ (\lambda\vec{z}.F)\ M_{n1}\ \ldots\ M_{n(n-1)}\ Z & =_\tau & N_n \end{cases} \quad\Rightarrow\quad \forall\vec{Z},\ (\lambda\vec{z}.F)\vec{Z} =_\tau N_1.$$

How can $\mathrm{nf}((\lambda z.t)b)$ be independent from b?

1. $(\lambda z.t)b \twoheadrightarrow 0$ for all b, i.e. $t \twoheadrightarrow 0$;
2. b is erased during the reduction;
3. b is “hidden” behind an unsolvable;
4. b is “pushed to infinity”.

Idea of the proof

Perpendicular Lines Property

$$\forall Z \begin{cases} (\lambda \vec{z}.F)\ Z\ M_{12} \ldots\ldots M_{1n} & =_\tau & N_1 \\ (\lambda \vec{z}.F)\ M_{21}\ Z \ldots\ldots M_{2n} & =_\tau & N_2 \\ \ddots & \vdots & \vdots \\ (\lambda \vec{z}.F)\ M_{n1} \ldots M_{n(n-1)}\ Z & =_\tau & N_n \end{cases} \Rightarrow \quad \forall \vec{Z},\ (\lambda \vec{z}.F)\vec{Z} =_\tau N_1.$$

How can $\mathrm{nf}((\lambda z.t)b)$ be independent from b?

1. $(\lambda z.t)b \twoheadrightarrow 0$ for all b, i.e. $t \twoheadrightarrow 0$;
2. b is erased during the reduction;
3. ~~b is "hidden" behind an unsolvable~~ (no unsolvables);
4. b is "pushed to infinity".

Idea of the proof

Perpendicular Lines Property

$$\forall Z \begin{cases} (\lambda \vec{z}.F)\ Z\ M_{12} \ldots\ldots M_{1n} & =_\tau & N_1 \\ (\lambda \vec{z}.F)\ M_{21}\ Z \ldots\ldots M_{2n} & =_\tau & N_2 \\ \ddots & \vdots & \vdots \\ (\lambda \vec{z}.F)\ M_{n1} \ldots M_{n(n-1)}\ Z & =_\tau & N_n \end{cases} \quad \Rightarrow \quad \forall \vec{Z},\ (\lambda \vec{z}.F)\vec{Z} =_\tau N_1.$$

How can $\mathrm{nf}((\lambda z.t)b)$ be independent from b?

1. $(\lambda z.t)b \twoheadrightarrow 0$ for all b, i.e. $t \twoheadrightarrow 0$;
2. b is erased during the reduction;
3. ~~b is "hidden" behind an unsolvable~~ (no unsolvables);
4. ~~b is "pushed to infinity"~~ (SN).

Idea of the proof

Perpendicular Lines Property

$$\forall Z \begin{cases} (\lambda\vec{z}.F)\ Z\ M_{12}\ \ldots\ldots\ M_{1n} & =_\tau & N_1 \\ (\lambda\vec{z}.F)\ M_{21}\ Z\ \ldots\ldots\ M_{2n} & =_\tau & N_2 \\ \ddots & \vdots & \vdots \\ (\lambda\vec{z}.F)\ M_{n1}\ \ldots\ M_{n(n-1)}\ Z & =_\tau & N_n \end{cases} \quad \Rightarrow \quad \forall\vec{Z},\ (\lambda\vec{z}.F)\vec{Z} =_\tau N_1.$$

How can $\mathrm{nf}((\lambda z.t)b)$ be independent from b?

1. $(\lambda z.t)b \twoheadrightarrow 0$ for all b, i.e. $t \twoheadrightarrow 0$;
2. $b = []$ is erased during the reduction (linearity);
3. ~~b is "hidden" behind an unsolvable~~ (no unsolvables);
4. ~~b is "pushed to infinity"~~ (SN).

Idea of the proof

Perpendicular Lines Property

$$\forall Z \begin{cases} (\lambda\vec{z}.F)\ Z\ M_{12} \ldots\ldots M_{1n} & =_\tau & N_1 \\ (\lambda\vec{z}.F)\ M_{21}\ Z \ldots\ldots M_{2n} & =_\tau & N_2 \\ \ddots & \vdots & \vdots \\ (\lambda\vec{z}.F)\ M_{n1} \ldots M_{n(n-1)}\ Z & =_\tau & N_n \end{cases} \quad \Rightarrow \quad \forall\vec{Z},\ (\lambda\vec{z}.F)\vec{Z} =_\tau N_1.$$

Lemma

Under the assuption above, if $t \in \mathcal{T}(F)$ then:

$$\mathrm{nf}(t) \neq 0 \quad \Rightarrow \quad \vec{z} \notin t.$$

By induction on the size of c.

PLP	$=_\lambda$	$=_\mathcal{B}$
open	✓	✓
closed	X	?

Idea of the proof

Perpendicular Lines Property

$$\forall Z \begin{cases} (\lambda\vec{z}.F)\ Z\ M_{12} \ldots\ldots M_{1n} & =_\tau & N_1 \\ (\lambda\vec{z}.F)\ M_{21}\ Z \ldots\ldots M_{2n} & =_\tau & N_2 \\ \ddots & \vdots & \vdots \\ (\lambda\vec{z}.F)\ M_{n1} \ldots M_{n(n-1)}\ Z & =_\tau & N_n \end{cases} \quad \Rightarrow \quad \forall\vec{Z},\ (\lambda\vec{z}.F)\vec{Z} =_\tau N_1.$$

Our proof does not need open terms!

PLP holds in $\Lambda^o/_{=_\mathcal{B}}$ ✓

PLP	$=_\lambda$	$=_\mathcal{B}$
open	✓	✓
closed	X	✓

The $\lambda\mu$-calculus (Parigot '92)

Terms

$$M ::= x \mid \lambda x.M \mid MM \mid \mu\alpha.{}_{\beta}|M|$$

Reduction

$$(\lambda x.M)N \to_{\lambda} M\{N/x\}$$

$$\mu\alpha.{}_{\beta}|\mu\gamma.{}_{\eta}|M|| \to_{\rho} \mu\alpha.{}_{\eta}|M|\{\beta/\gamma\}$$

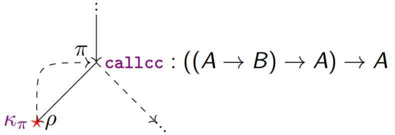
$$(\mu\alpha.{}_{\beta}|M|)N \to_{\mu} \mu\alpha.({}_{\beta}|M|)_{\alpha}N$$

Impure functional programming lang:

Continuations

$\texttt{callcc} := \lambda y.\mu\alpha.{}_{\alpha}|y(\lambda x.\mu\delta.{}_{\alpha}|x|)|$

Classical logic:



The resource $\lambda\mu$-calculus

Define the set $\lambda\mu^{\mathrm{r}}$ of resource $\lambda\mu$-terms:

$$t ::= x \mid \lambda x.t \mid t\,[t, \ldots, t] \mid \mu\alpha.{}_{\beta}|t|$$

The resource $\lambda\mu$-calculus

Define the set $\lambda\mu^{\mathrm{r}}$ of resource $\lambda\mu$-terms:

$$t ::= \; x \; \mid \lambda x.t \; \mid t\,[t,\ldots,t] \; \mid \mu\alpha._{\beta}|t|$$

Reduction: $(\lambda x.t)[\vec{s}] \to_{\lambda} t\langle[\vec{s}]/x\rangle$

The resource $\lambda\mu$-calculus

Define the set $\lambda\mu^{\mathrm{r}}$ of resource $\lambda\mu$-terms:

$$t ::= x \mid \lambda x.t \mid t\,[t,\ldots,t] \mid \mu\alpha.{}_{\beta}|t|$$

Reduction: $(\lambda x.t)[\vec{s}] \to_{\lambda} t\langle[\vec{s}]/x\rangle \qquad \mu\alpha.{}_{\beta}|\mu\gamma.{}_{\eta}|t|| \to_{\rho} \mu\alpha.{}_{\eta}|t|\{\beta/\gamma\}$

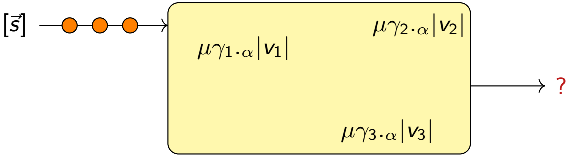
The resource $\lambda\mu$-calculus

Define the set $\lambda\mu^{\mathrm{r}}$ of resource $\lambda\mu$-terms:

$$t ::= x \mid \lambda x.t \mid t\,[t,\ldots,t] \mid \mu\alpha._\beta|t|$$

Reduction: $(\lambda x.t)[\vec{s}] \to_\lambda t\langle[\vec{s}]/x\rangle \qquad \mu\alpha._\beta|\mu\gamma._\eta|t|| \to_\rho \mu\alpha._\eta|t|\{\beta/\gamma\}$

$$(\mu\alpha._\beta|t|)[\vec{s}] \to_\mu ?$$



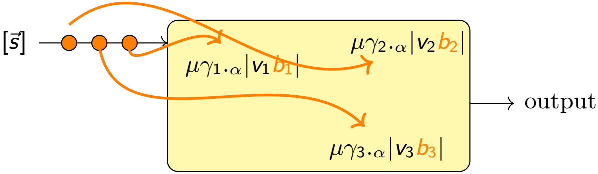
The resource $\lambda\mu$-calculus

Define the set $\lambda\mu^{\mathrm{r}}$ of resource $\lambda\mu$-terms:

$$t ::= x \mid \lambda x.t \mid t\,[t,\dots,t] \mid \mu\alpha.{}_{\beta}|t|$$

Reduction: $(\lambda x.t)[\vec{s}] \to_{\lambda} t\langle[\vec{s}]/x\rangle \qquad \mu\alpha.{}_{\beta}|\mu\gamma.{}_{\eta}|t|| \to_{\rho} \mu\alpha.{}_{\eta}|t|\{\beta/\gamma\}$

$$(\mu\alpha.{}_{\beta}|t|)[\vec{s}] \to_{\mu} \mu\alpha.{}_{\beta}|t|\{\dots,\ {}_{\alpha}|(\cdot)b_i|/_{\alpha|\cdot|^{(i)}},\dots\}$$



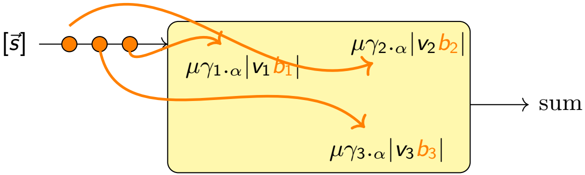
The resource $\lambda\mu$-calculus

Define the set $\lambda\mu^{\mathrm{r}}$ of resource $\lambda\mu$-terms:

$$t ::= x \mid \lambda x.t \mid t\,[t,\dots,t] \mid \mu\alpha.{}_{\beta}|t|$$

Reduction: $(\lambda x.t)[\vec{s}] \to_{\lambda} t\langle[\vec{s}]/x\rangle \qquad \mu\alpha.{}_{\beta}|\mu\gamma.{}_{\eta}|t|| \to_{\rho} \mu\alpha.{}_{\eta}|t|\{\beta/\gamma\}$

$$(\mu\alpha.{}_{\beta}|t|)[\vec{s}] \to_{\mu} \sum_{b_1 * \cdots * b_k = [\vec{s}]} \mu\alpha.{}_{\beta}|t|\{\dots, {}_{\alpha}|(\cdot)\,b_i|/_{{}_{\alpha}|\cdot|^{(i)}}, \dots\}$$



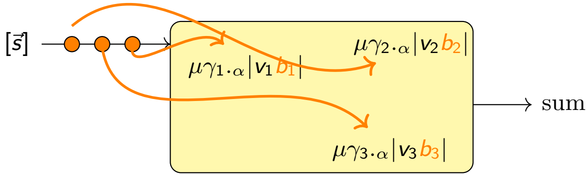
The resource $\lambda\mu$-calculus

Define the set $\lambda\mu^{\mathrm{r}}$ of resource $\lambda\mu$-terms:

$$t ::= x \mid \lambda x.t \mid t\,[t,\ldots,t] \mid \mu\alpha.{}_{\beta}|t|$$

Reduction: $(\lambda x.t)[\vec{s}] \to_{\lambda} t\langle[\vec{s}]/x\rangle \qquad \mu\alpha.{}_{\beta}|\mu\gamma.{}_{\eta}|t|| \to_{\rho} \mu\alpha.{}_{\eta}|t|\{\beta/\gamma\}$

$$(\mu\alpha.{}_{\beta}|t|)[\vec{s}] \to_{\mu} \sum_{b_1 * \cdots * b_k = [\vec{s}]} \mu\alpha.{}_{\beta}|t|\{\ldots, {}_{\alpha}|(\cdot)b_i|/_{\alpha|\cdot|^{(i)}}, \ldots\}$$



Strong normalisation: Not immediate, multiset order

Confluence

1. Add coefficients: gain contextuality of reduction on sums
2. Prove confluence in the setting with coefficients
3. Show that this entails the confluence of the calculus without coefficients

Qualitative Taylor expansion

Same definition, plus: $\mathcal{T}(\mu\alpha._\beta|M|) := \{\mu\alpha._\beta|t| \mid t \in \mathcal{T}(M)\}$.

Simulation property

If $M \to N$ then:

- for all $s \in \mathcal{T}(M)$ there is $\mathbb{T} \subseteq \mathcal{T}(N)$ s.t. $s \twoheadrightarrow \mathbb{T}$
- for all $s' \in \mathcal{T}(N)$ there is $s \in \mathcal{T}(M)$ s.t. $s \twoheadrightarrow s' + \textit{something}$

Non-interference property

Let $t, s \in \mathcal{T}(M)$. Then: $\mathrm{nf}(t) \cap \mathrm{nf}(s) \neq \emptyset \Rightarrow t = s$.

Main results

- The equivalence equating $\mathrm{NF}(\mathcal{T}(\cdot))$'s is a sensible $\lambda\mu$-theory.
- PLP and Stability hold in $\lambda\mu$-calculus (thus also $\nexists\,\texttt{por}$).

Conclusions

- Resource approximation (Taylor expansion) is a powerful tool for studying the properties of the language
- Replaces coinductive arguments by inductive ones
- It adapts to other programming languages

- Böhm trees for $\lambda\mu$-calculus?
- What about Saurin's $\Lambda\mu$-calculus?
- What is an approximation of a program?

For more, see the thesis!

- Call-by-value
- Homology and proofs
- Philosophy


ANY
QUESTIONS
?