Approximating functional programs: Taylor subsumes Scott, Berry, Kahn and Plotkin

Davide Barbarossa and Giulio Manzonetto
barbarossa@lipn.univ-paris13.fr
LIPN, Université Sorbonne Paris Nord

May 5, 2020
**Possible behaviours of a program** $F = F\{x\}$

<table>
<thead>
<tr>
<th>Normalizing</th>
<th>Meaningless</th>
</tr>
</thead>
<tbody>
<tr>
<td>$z \leftarrow 1$</td>
<td>while(True){</td>
</tr>
<tr>
<td>$z \leftarrow \frac{z + x/z}{2}$</td>
<td>DoNothing</td>
</tr>
<tr>
<td>write $i$-th digit of $z$</td>
<td>}</td>
</tr>
<tr>
<td>$F(2) = 1, 5$</td>
<td>$F(2)$ produces no information!</td>
</tr>
</tbody>
</table>
### Possible behaviours of a program $F = F\{x\}$

<table>
<thead>
<tr>
<th>Normalizing</th>
<th>Solvable (Babylonians)</th>
<th>Meaningless</th>
</tr>
</thead>
</table>
| $z \leftarrow 1$
$z \leftarrow \frac{z + x/z}{2}$
write $i$-th digit of $z$ | $z \leftarrow 1; i \leftarrow 0$
while(True){
  $i++$
  $z \leftarrow \frac{z + x/z}{2}$
  write $i$-th digit of $z$
}

$F(2) = 1, 5$ | $F(2) \rightarrow 1, 41$
$\quad \rightarrow 1, 414$
$\quad \rightarrow 1, 4142$
$\quad \cdots \rightarrow \infty \sqrt{2}$ | while(True){
  DoNothing
}

$F(2)$ produces no information!
### Böhm Trees

#### λ-calculus (Church)

The set $\Lambda$ of programs is given by $M ::= x \mid \lambda x. M \mid MM$.

**Computation step:** $(\lambda x. M)N \to M\{N/x\}$.

#### Böhm trees (Barendregt)

The map $BT : \Lambda \to B$ associates each $\lambda$-term $F$ with its *Böhm tree*:

$$BT(F) := BT(hnf(F)), \quad BT(F) := \bot \text{ if } F \text{ is unsolvable},$$

$$BT(\lambda \vec{x}. y \ Q_1 \ldots \ Q_k) := \lambda \vec{x}. y \ BT(Q_1) \ldots BT(Q_k)$$

The equivalence $\equiv_{BT}$ is a $\lambda$-theory. So $B_\Lambda \simeq \Lambda / \equiv_{BT}$ is a semantics for $\Lambda$.

The set of all normal forms is dense in $B_\Lambda$ (in analogy with $\mathbb{Q}$ dense in $\mathbb{R}$).
Finite approximants

- The set $\mathcal{A}$ of finite approximants is defined as:

$$P ::= \bot | \lambda \vec{x}. y \ P \ldots P$$

with the intuition that $\bot$ means no information.

- Fix $\leq$ the preorder on $\mathcal{A}$ generated by taking $\bot \leq P$ for all $P$.

- The set $\mathcal{A}(F)$ of the finite approximants of $F \in \Lambda$ is:

$$\mathcal{A}(F) := \{P \in \mathcal{A} \ s.t \ \exists N \in \Lambda \ s.t \ F \rightarrow_{\beta} N \geq P\}$$

Approximation Theorem

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

in analogy to the fact that $\sqrt{2}$ is the limit of Babylonian Program$(2)$. 
Derivatives!

<table>
<thead>
<tr>
<th>Analysis</th>
<th>$F(x)$</th>
<th>$\lambda$-calculus</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application</td>
<td>$F(x)$</td>
<td>$F \times$</td>
</tr>
<tr>
<td>Taylor expansion $\Theta(\cdot)$</td>
<td>$\sum^n \frac{1}{n!} F^{(n)}(0)x^n$</td>
<td>$\sum^n \frac{1}{n!} (D^n \Theta(F) \bullet x^n)0$</td>
</tr>
</tbody>
</table>
**Differential \(\lambda\)-calculus**

Programs live in the module \(\mathbb{Q}^+\langle\Lambda^r\rangle_\infty\) and are subject to the equation:

\[
D(\lambda x. M) \cdot N = \lambda x. \left( \frac{d}{dx} M \cdot N \right)
\]

where \(\frac{d}{dx} (PQ) \cdot N := (\frac{d}{dx} P \cdot N) \cdot Q + (DP \cdot (\frac{d}{dx} Q \cdot N)) \cdot Q\)

is the linear substitution of \(N\) in \(M\) for \(x\).

**Ehrhard and Régnier:**

\(\Theta\) defines a function \(\Lambda \to \mathbb{Q}^+\langle\Lambda^r\rangle_\infty\) (called the *full Taylor expansion*):

\[
\Theta(\cdot) = \sum_{t \in \mathcal{T}(\cdot)} \frac{1}{m(t)} t
\]

where \(m(t) \in \mathbb{N}\) is difficult and \(\mathcal{T}(\cdot) : \Lambda \to \mathcal{P}(\Lambda^r)\) is easy (i.e. inductive). Furthermore,

\[
NF(\Theta(\cdot)) = \Theta(BT(\cdot)).
\]
Define the set $\Lambda^r$ of Resource terms:

$$ t ::= x \mid \lambda x.t \mid t[t, \ldots, t] $$
Define the set $\Lambda^r$ of Resource terms:

$$t ::= x \mid \lambda x.t \mid t[t, \ldots, t]$$
Define the set \( \Lambda^r \) of Resource terms:

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

Reduction:

\[
(\lambda x. t)[s_1, s_2, s_3] \rightarrow ?
\]
Define the set $\Lambda^r$ of Resource terms:

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

Reduction:

$$(\lambda x.t)[s_1, s_2, s_3] \rightarrow t\{s_1/x^{(1)}, s_2/x^{(2)}, s_3/x^{(3)}\}$$
Define the set $\Lambda^r$ of Resource terms:

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

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

Reduction:

$$(\lambda x.t)[s_1, s_2, s_3] \rightarrow \sum_{\sigma \in \mathcal{S}_3} t\{s_{\sigma(1)}/x^{(1)}, s_{\sigma(2)}/x^{(2)}, s_{\sigma(3)}/x^{(3)}\}$$
Define the set \( \Lambda^r \) of Resource terms:

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

We need formal (idempotent) sum \( T = t_1 + \cdots + t_n \) of resource terms.

Reduction:

\[
(\lambda x.t)[s_1, s_2, s_3] \rightarrow ?
\]
Define the set $\Lambda^r$ of Resource terms:

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

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

Reduction:

\[ (\lambda x.t)[s_1, s_2, s_3] \rightarrow 0 \]
Resource terms live a tough life

They may experience non-determinism:

$$\Delta[x, y] := (\lambda x.x[x])[y, y'] \rightarrow y[y'] + y'[y]$$

But also starvation:

$$\Delta[\Delta, \Delta] \rightarrow (\lambda x.x[x])[\Delta] \rightarrow 0$$

As well as surfeit:

$$(\lambda x\lambda y.x)[l][l] \rightarrow (\lambda y.l)[l] \rightarrow 0$$

Summing up: $$\lambda x.t)[s_1, \ldots, s_n] \not\rightarrow 0 \quad \Rightarrow \quad t \text{ uses each } s_i \text{ exactly once!}$$

Main Properties:

- **Linearity:** Cannot erase non-empty bags (unless annihilating).
- **Strong Normalization:** Trivial, as there is no duplication.
- **Confluence:** Locally confluent + strongly normalizing.
Qualitative Taylor Expansion

The (support of the full) Taylor expansion is the map $T(\cdot) : \Lambda \rightarrow \mathcal{P}(\Lambda^r)$:

\[
\begin{align*}
T(x) &= \{x\} \\
T(\lambda x. M) &= \{\lambda x. t \in \Lambda^r \text{ s.t. } t \in T(M)\} \\
T(MN) &= \{t[s_1, \ldots, s_k] \in \Lambda^r \text{ s.t. } k \in \mathbb{N}, t \in T(M), s_i \in T(N)\}.
\end{align*}
\]
Qualitative Taylor Expansion

The (support of the full) **Taylor expansion** is the map $\mathcal{T}(:) : \Lambda \to \mathcal{P}(\Lambda^r)$:

- $\mathcal{T}(x) = \{x\}$
- $\mathcal{T}(\lambda x. M) = \{\lambda x. t \in \Lambda^r \text{ s.t. } t \in \mathcal{T}(M)\}$
- $\mathcal{T}(MN) = \{t[s_1, \ldots, s_k] \in \Lambda^r \text{ s.t. } k \in \mathbb{N}, t \in \mathcal{T}(M), s_i \in \mathcal{T}(N)\}$

**Examples:**

- $\mathcal{T}(\lambda x.x) = \{\lambda x.x\}$
- $\mathcal{T}(\lambda x.xx) = \{\lambda x.x[x^n] \mid n \in \mathbb{N}\}$
- $\mathcal{T}(\Omega) = \{(\lambda x.x[x^{n_0}])[\lambda x.x[x^{n_1}], \ldots, \lambda x.x[x^{n_k}] \mid k, n_0, \ldots, n_k \in \mathbb{N}\}$
- $\mathcal{T}(\Delta_f) = \{\lambda x.f[x^n][x^k] \mid n, k \in \mathbb{N}\}$
- $\mathcal{T}(Y) = \{\lambda f. t[s_1, \ldots, s_k] \mid k \in \mathbb{N}, t, s_1, \ldots, s_k \in \mathcal{T}(\Delta_f)\}$

where $Y = \lambda f. \Delta_f \Delta_f$ and $\Delta_f = \lambda x.f(xx)$. 
Computing the normal form:

\[ \text{NF}(\mathcal{T}(M)) = \bigcup_{t \in \mathcal{T}(M)} \text{nf}(t) \]

Examples

\[ \text{NF}(\mathcal{T}(Y)) = \{ \lambda f. f \, 1, \lambda f. f[f \, 1], \lambda f. f[f \, 1, f \, 1], \lambda f. f[f \, 1, f[f \, 1], f[f[f \, 1]]], \ldots \} \].

\[ \text{NF}(\mathcal{T}(\Omega)) = \emptyset. \text{ This is the case for all unsolvables.} \]
Approximating through resources

Computing the normal form:

\[
\text{NF}(\mathcal{T}(M)) = \bigcup_{t \in \mathcal{T}(M)} \text{nf}(t)
\]

Examples

\[
\text{NF}(\mathcal{T}(Y)) = \{\lambda f.f1, \lambda f.f[f1], \lambda f.f[f1, f1], \lambda f.f[f1, f[f1]], f[f[f1]], \ldots \}.
\]

\[
\text{NF}(\mathcal{T}(\Omega)) = \emptyset. \text{ This is the case for all unsolvables.}
\]

Taylor Expansion of Böhm Trees

\[
\mathcal{T}(\bot) := \emptyset
\]

\[
\mathcal{T}(\mathcal{BT}(M)) := \bigcup_{P \in \mathcal{A}(M)} \mathcal{T}(P)
\]
Approximating through resources

Computing the normal form:

\[
\text{NF}(\mathcal{T}(M)) = \bigcup_{t \in \mathcal{T}(M)} \text{nf}(t)
\]

Examples

\[
\text{NF}(\mathcal{T}(Y)) = \{\lambda f. f1, \lambda f. [f1], \lambda f. [f1, f1], \lambda f. [f1, f[f1]], \ldots \}.
\]
\[
\text{NF}(\mathcal{T}(\Omega)) = \emptyset. \text{ This is the case for all unsolvables.}
\]

Taylor Expansion of Böhm Trees

\[
\mathcal{T}(\bot) := \emptyset
\]
\[
\mathcal{T}(\text{BT}(M)) := \bigcup_{P \in \mathcal{A}(M)} \mathcal{T}(P)
\]

The diagram commutes!
A common structure

1. Source language $\mapsto$ Resource version
   (gain confluence and strong normalization)

2. via a Taylor Expansion, providing:
   - static analysis (coherence/cliques),
   - dynamic analysis (normalization)

3. and (when possible) a:

   **Commutation Theorem**
   \[
   \text{NF}(\mathcal{T}(P)) = \mathcal{T}(\text{BT}(P))
   \]

   **Corollary**
   \[
   \text{BT}(M) = \text{BT}(N) \iff \text{NF}(\mathcal{T}(M)) = \text{NF}(\mathcal{T}(N))
   \]
A common structure

1. Source language \(\mapsto\) Resource version
   (gain confluence and strong normalization)

2. via a Taylor Expansion, providing:
   static analysis (coherence/cliques),
   dynamic analysis (normalization)

3. and (when possible) a:
   Commutation Theorem

\[ \text{CNF}(T(P)) = T(BT(P)) \]

Corollary

\[ BT(M) = BT(N) \iff \text{CNF}(T(M)) = \text{CNF}(T(N)) \]

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

Ehrhard and Régnier
Chapter 14

Scott's Continuity

Berry's Stability

Khan & Plotkin's Sequentiality

Contextuality of BTs

topological argument

Genericity Property

Perpendicular Lines Property

\[ \neg \text{ parallel or} \]
Classic results via Resource Approximation

Scott’s Continuity

Berry’s Stability

Khan & Plotkin’s Sequentiality

Commutation Theorem

\[ \text{NF}(T(P)) = T(BT(P)) \]

Contextuality of BTs

Genericity Property

Perpendicular Lines Property

\[ \not\parallel \] parallel or
Equality mod $\text{BT}$ is a $\lambda$-theory

Contextuality of Böhm trees

Let $C(\cdot \cdot)$ be a context.

$$\text{BT}(M) = \text{BT}(N) \quad \Rightarrow \quad \text{BT}(C(M)) = \text{BT}(C(N))$$
Equality mod \textsf{NFT} is a \(\lambda\)-theory

Monotonicity of contexts w.r.t. \(\leq_{\text{NFT}}\)

Let \(C(\cdot)\) be a context.

\[
\text{NF}(T(M)) \subseteq \text{NF}(T(N)) \Rightarrow \text{NF}(T(C\langle M\rangle)) \subseteq \text{NF}(T(C\langle N\rangle))
\]

**Proof.** Induction on \(C\). The interesting case is \(C = C_1 \cdot C_2\).
Equality mod NFT is a $\lambda$-theory

Monotonicity of contexts w.r.t. $\leq_{\text{NFT}}$

Let $C(\cdot)$ be a context.

$$\text{NF}(\mathcal{T}(M)) \subseteq \text{NF}(\mathcal{T}(N)) \Rightarrow \text{NF}(\mathcal{T}(C(M))) \subseteq \text{NF}(\mathcal{T}(C(N)))$$

Proof. Induction on $C$. The interesting case is $C = C_1 C_2$.

$t \in \text{NF}(\mathcal{T}(C(M))) \Rightarrow \exists t' \in \mathcal{T}((C_1(M))(C_2(M)))$ such that:

$$t' = s_1[u_1, \ldots, u_k] \longrightarrow t + T$$

with $s_1 \subseteq \text{NF}(\mathcal{T}(C_1(M)))$

and $u_1, \ldots, u_k \subseteq \text{NF}(\mathcal{T}(C_2(M)))$
Equality mod NFT is a $\lambda$-theory

Monotonicity of contexts w.r.t. $\leq_{\text{NFT}}$

Let $C(\cdot)$ be a context.

\[
\text{NF}(\mathcal{T}(M)) \subseteq \text{NF}(\mathcal{T}(N)) \Rightarrow \text{NF}(\mathcal{T}(C(M))) \subseteq \text{NF}(\mathcal{T}(C(N)))
\]

**Proof.** Induction on $C$. The interesting case is $C = C_1 \; C_2$.

\[t \in \text{NF}(\mathcal{T}(C(M))) \Rightarrow \exists t' \in \mathcal{T}((C_1(M))(C_2(M)))\text{ such that :}
\]

\[
t' = s_1[u_1, \ldots, u_k] \quad \Rightarrow t + \top
\]

\[
\downarrow
\]

\[
\text{nf}(s_1)[\text{nf}(u_1), \ldots, \text{nf}(u_k)]
\]

with $\text{nf}(s_1) \subseteq \text{NF}(\mathcal{T}(C_1(M))) \subseteq \text{NF}(\mathcal{T}(C_1(N)))$

and $\text{nf}(u_1), \ldots, \text{nf}(u_k) \subseteq \text{NF}(\mathcal{T}(C_2(M))) \subseteq \text{NF}(\mathcal{T}(C_2(N)))$.

Easily conclude that $t \in \text{NF}(\mathcal{T}(C(N)))$. □
Unsolvables are computationally meaningless

**Genericity Property**

Let $U$ unsolvable. If $C(U)$ has a $\beta$-nf, then $C(U) =_\beta C(M) \forall M \in \Lambda$. 

Proof.

$C(L)U \text{ normalizable} \Rightarrow \exists t \in NF(T(CLUM))$ such that:

"$\text{nf}_{\beta}(C(L)U) = t$" and all its bags are singletons.

So $\exists t' \in T(CLUM)$ such that:

$t' = cLs_1,...,s_kM \rightarrow \rightarrow \rightarrow t + T$ for some $c \in T(CLM)$ and $s_1,...,s_k \in T(U)$.

No hole can occur in $c$!

Therefore:

$t' = cLs_1,...,s_kM = c \in T(CLM)$ and hence $t \in NF(T(CLM))$.

□
Unsolvables are computationally meaningless

Genericity Property

Let $U$ unsolvable. If $C[U]$ has a $\beta$-nf, then $C[U] =\beta C[M] \forall M \in \Lambda$.

Proof. $C[U]$ normalizable $\Rightarrow \exists t \in NF(T(C[U]))$ such that:

"$\text{nf}_\beta(C[U]) = t$" and all its bags are singletons.

So $\exists t' \in T(C[U])$ such that:

$$t' = c(s_1, \ldots, s_k) \Rightarrow t + T$$

for some $c \in T(C[\cdot])$ and $s_1, \ldots, s_k \in T(U)$. 
**Unsolvables are computationally meaningless**

**Genericity Property**

Let $U$ unsolvable. If $C(U)$ has a $\beta$-nf, then $C(U) = C(M) \forall M \in \Lambda$.

**Proof.** $C(U)$ normalizable $\Rightarrow \exists t \in \text{NF}(T(C(U)))$ such that:

"$\text{nf}_\beta(C(U)) = t$" and all its bags are singletons.

So $\exists t' \in T(C(U))$ such that:

$$t' = c\langle s_1, \ldots, s_k \rangle \Rightarrow t + \top$$

for some $c \in T(C(\cdot))$ and $s_1, \ldots, s_k \in T(U)$. 
Unsolvables are computationally meaningless

Genericity Property

Let $U$ unsolvable. If $C\{U\}$ has a $\beta$-nf, then $C\{U\} = \beta C\{M\}$ $\forall M \in \Lambda$.

Proof. $C\{U\}$ normalizable $\Rightarrow \exists t \in \text{NF}(\mathcal{T}(C\{U\}))$ such that:

"$\text{nf}_\beta(C\{U\}) = t$" and all its bags are singletons.

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

$$
t' = c\{s_1, \ldots, s_k\} \quad \Rightarrow \quad t + \top
\downarrow
\Rightarrow
\quad c\{0, \ldots, 0\}
$$

for some $c \in \mathcal{T}(C\{\cdot\})$ and $s_1, \ldots, s_k \in \mathcal{T}(U)$. ($U$ unsolvable $\Rightarrow \text{nf}(s_i) = 0$)
Unsolvables are computationally meaningless

Genericity Property
Let \( U \) unsolvable. If \( C\{U\} \) has a \( \beta \)-nf, then \( C\{U\} \equiv_\beta C\{M\} \ \forall M \in \Lambda. \)

Proof. \( C\{U\} \) normalizable \( \Rightarrow \ \exists t \in \text{NF}(T(C\{U\})) \) such that:

"\( \text{nf}_\beta(C\{U\}) = t \)" and all its bags are singletons.

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

\[
t' = c\{s_1, \ldots, s_k\} \quad \Rightarrow \quad t + \top
\]

\[
0 = c\{0, \ldots, 0\}
\]

for some \( c \in T(C\{\cdot\}) \) and \( s_1, \ldots, s_k \in T(U) \).
Unsolvables are computationally meaningless

Genericity Property

Let $U$ unsolvable. If $C(U)$ has a $\beta$-nf, then $C(U) =_\beta C(M)$ $\forall M \in \Lambda$.

**Proof.** $C(U)$ normalizable $\Rightarrow \exists t \in \text{NF}(T(C(U)))$ such that:

"$\text{nf}_\beta(C(U)) = t$" and all its bags are singletons.

So $\exists t' \in T(C(U))$ such that:

$$t' = c(s_1, \ldots, s_k) \rightarrow t + T$$

for some $c \in T(C(\cdot))$ and $s_1, \ldots, s_k \in T(U)$.

No hole can occur in $c$!

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

Since and bags of $t$ are singletons, "$t = \text{nf}_\beta(C(M))$".

□
**Perpendicular Lines Property**

**PLP:** If a context $C(\cdot, \ldots, \cdot) : \Lambda^n \to \Lambda$ is constant on $n$ perpendicular lines, then it must be constant everywhere.
**Perpendicular Lines Property**

**PLP:** If a context $C(\cdot, \ldots, \cdot) : \Lambda^n \to \Lambda$ is constant on $n$ perpendicular lines, then it must be constant everywhere.

**True** in $\Lambda^\circ / \approx_{\text{BT}}$, Barendregt’s Book 1982

Proof: via Sequentiality.

**??** in $\Lambda^\circ / \approx_{\text{BT}}$

**False** in $\Lambda^\circ / \approx_{\beta}$, Barendregt & Statman 1999

Counterexample: via Plotkin’s terms.

**True** in $\Lambda / \approx_{\beta}$, De Vrijer & Endrullis 2008

Proof: via Reduction under Substitution.

<table>
<thead>
<tr>
<th>PLP</th>
<th>$\beta$</th>
<th>BT</th>
</tr>
</thead>
<tbody>
<tr>
<td>open</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>closed</td>
<td>X</td>
<td>?</td>
</tr>
</tbody>
</table>
Idea of the proof

Perpendicular Lines Property

\[ \forall Z \left\{ \begin{array}{l}
C(\langle Z, M_{12}, \ldots, M_{1n} \rangle) =_{\text{BT}} N_1 \\
C(\langle M_{21}, Z, \ldots, M_{2n} \rangle) =_{\text{BT}} N_2 \\
\vdots \quad \vdots \\
C(\langle M_{n1}, \ldots, M_{n(n-1)}, Z \rangle) =_{\text{BT}} N_n
\end{array} \right\} \Rightarrow \forall \bar{Z}, C(\langle \bar{Z} \rangle) =_{\text{BT}} N_1. \]
Idea of the proof

Perpendicular Lines Property

$$\forall Z \left\{ \begin{array}{l}
C(Z, M_{12}, \ldots, M_{1n}) =_{BT} N_1 \\
C(M_{21}, Z, \ldots, M_{2n}) =_{BT} N_2 \\
\quad \vdots \\
C(M_{n1}, \ldots, M_{n(n-1)}, Z) =_{BT} N_n
\end{array} \right. \Rightarrow \forall \tilde{Z}, C(\tilde{Z}) =_{BT} N_1.$$  

How can a context $C(\cdot)$ be constant in $\Lambda/ =_{BT}$?

1. $C(\cdot)$ does not contain the hole at all (the trivial case);
2. the hole is erased during its reduction;
3. the hole is “hidden” behind an unsolvable;
4. the hole is never erased but “pushed into infinity”.

D. Barbarossa and G. Manzonetto

May 5, 2020 16/17
Idea of the proof

Perpendicular Lines Property

\[ \forall Z \left\{ \begin{array}{l}
C(Z, M_{12}, \ldots, M_{1n}) = \text{NFT} \quad N_1 \\
C(M_{21}, Z, \ldots, M_{2n}) = \text{NFT} \quad N_2 \\
\quad \vdots \\
C(M_{n1}, \ldots, M_{n(n-1)}, Z) = \text{NFT} \quad N_n
\end{array} \right\} \Rightarrow \forall \tilde{Z}, C(\tilde{Z}) = \text{NFT} \quad N_1. \]

How can a \( c \in \mathcal{T}(C(\cdot)) \) s.t. \( \text{nf}(c) \neq 0 \) be constant in \( \Lambda' \)?

1. \( c \) does not contain the hole at all (the trivial case);
2. the hole is erased during its reduction;
3. the hole is “hidden” behind an unsolvable;
4. the hole is never erased but “pushed into infinity”.

D. Barbarossa and G. Manzonetto

Approximating functional programs: Taylor sub.
Idea of the proof

Perpendicular Lines Property

\[
\forall \vec{Z} \begin{cases}
C(\langle Z, M_{11}, \ldots, M_{1n} \rangle) =_{\text{NFT}} N_1 \\
C(\langle M_{21}, Z, \ldots, M_{2n} \rangle) =_{\text{NFT}} N_2 \\
\vdots \\
C(\langle M_{n1}, \ldots, M_{n(n-1)}, Z \rangle) =_{\text{NFT}} N_n
\end{cases} \Rightarrow \forall \vec{Z}, C(\vec{Z}) =_{\text{NFT}} N_1.
\]

How can a \( c \in T(\langle C(\cdot) \rangle) \) s.t. \( \text{nf}(c) \neq 0 \) be constant in \( \Lambda' \)?

1. \( c \) does not contain the hole at all (the trivial case);
2. the hole is erased during its reduction (linearity);
3. the hole is “hidden” behind an unsolvable;
4. the hole is never erased but “pushed into infinity”.

D. Barbarossa and G. Manzonetto
Approximating functional programs: Taylor subsumes Scott, Berry, Kahn and Plotkin
May 5, 2020 16/17
Idea of the proof

Perpendicular Lines Property

\[ \forall Z \begin{cases} C(\langle Z, M_{12}, \ldots, M_{1n} \rangle) =_{\text{NFT}} N_1 \\ C(\langle M_{21}, Z, \ldots, M_{2n} \rangle) =_{\text{NFT}} N_2 \\ \vdots \\ C(\langle M_{n1}, \ldots, M_{n(n-1)}, Z \rangle) =_{\text{NFT}} N_n \end{cases} \Rightarrow \forall \vec{Z}, C(\vec{Z}) =_{\text{NFT}} N_1. \]

How can a \( c \in \mathcal{T}(C(\cdot)) \) s.t. \( \text{nf}(c) \neq 0 \) be constant in \( \Lambda' \)?

1. \( c \) does not contain the hole at all (the trivial case);
2. the hole is erased during its reduction (linearity);
3. the hole is “hidden” behind an unsolvable (strong normalization);
4. the hole is never erased but “pushed into infinity”.

D. Barbarossa and G. Manzonetto
Approximating functional programs: Taylor subsumes Scott, Berry, Kahn and Plotkin
May 5, 2020 16 / 17
Idea of the proof

Perpendicular Lines Property

\[ \forall Z \begin{cases} C(\langle Z, M_{12}, \ldots, M_{1n} \rangle) =_{\text{NFT}} N_1 \\ C(\langle M_{21}, Z, \ldots, M_{2n} \rangle) =_{\text{NFT}} N_2 \\ \vdots \\ C(\langle M_{n1}, \ldots, M_{n(n-1)}, Z \rangle) =_{\text{NFT}} N_n \end{cases} \Rightarrow \forall \tilde{Z}, C(\langle \tilde{Z} \rangle) =_{\text{NFT}} N_1. \]

How can a \( c \in T(\langle C(\cdot) \rangle) \) s.t. \( \text{nf}(c) \neq 0 \) be constant in \( \Lambda' \) ?

1. \( c \) does not contain the hole at all (the trivial case!);
2. the hole is erased during its reduction (linearity);
3. the hole is “hidden” behind an unsolvable (strong normalization);
4. the hole is never erased but “pushed into infinity” (finiteness).
Idea of the proof

**Perpendicular Lines Property**

\[ \forall \vec{Z} \left\{ \begin{array}{ll}
C(\langle \vec{Z}, M_{12}, \ldots, M_{1n} \rangle) & =_{\text{NFT}} N_1 \\
C(\langle M_{21}, \vec{Z}, \ldots, M_{2n} \rangle) & =_{\text{NFT}} N_2 \\
\vdots & \vdots \\
C(\langle M_{n1}, \ldots, M_{n(n-1)}, \vec{Z} \rangle) & =_{\text{NFT}} N_n
\end{array} \right. \Rightarrow \forall \vec{Z}, C(\langle \vec{Z} \rangle) =_{\text{NFT}} N_1. \]

Claim.

If \( c \in T(C(\langle \cdot \rangle)) \) then:

\[ \text{nf}(c) \neq 0 \Rightarrow c \text{ contains no hole.} \]

By induction on the size of \( c \).
Idea of the proof

Perpendicular Lines Property

\[ \forall Z \left\{ \begin{array}{c}
C(\langle Z, M_{12}, \ldots, M_{1n} \rangle) =_{\text{NFT}} N_1 \\
C(\langle M_{21}, Z, \ldots, M_{2n} \rangle) =_{\text{NFT}} N_2 \\
\vdots \\
C(\langle M_{n1}, \ldots, M_{n(n-1)}, Z \rangle) =_{\text{NFT}} N_n
\end{array} \right\} \Rightarrow \forall \vec{Z}, C(\langle \vec{Z} \rangle) =_{\text{NFT}} N_1. \]

Our proof does not need open terms!

<table>
<thead>
<tr>
<th>PLP</th>
<th>$\beta$</th>
<th>BT</th>
</tr>
</thead>
<tbody>
<tr>
<td>open</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>closed</td>
<td>X</td>
<td>✓</td>
</tr>
</tbody>
</table>
The End!