Böhm Trees as Higher-Order Recursion Schemes

Pierre Clairambault
University of Cambridge

Andrzej Murawski
University of Warwick

June 4, 2013

Outline of the talk

I. Background and motivations

Higher-order recursion schemes

Higher-order recursion schemes are an abstract form of functional programs, often used as generators for infinite trees.

Example

With:

- Non-terminals $S : o, F : (o \to o) \to o \to o, G : (o \to o) \to o$,
- Terminals $\mathrm{a} : o \to o$, $\mathrm{b} : o \to o \to o$, $\mathrm{c} : o$.

$$\begin{array}{rcl} S & = & G\ \mathrm{a} \\ F\ f\ x & = & f\ (f\ x) \\ G\ f & = & \mathrm{b}\ (f\ \mathrm{c})\ (G\ (F\ f)) \end{array}$$

S

Higher-order recursion schemes

Higher-order recursion schemes are an abstract form of functional programs, often used as generators for infinite trees.

Example

With:

- Non-terminals $S : o, F : (o \to o) \to o \to o, G : (o \to o) \to o$,
- Terminals $\mathrm{a} : o \to o$, $\mathrm{b} : o \to o \to o$, $\mathrm{c} : o$.

$$\begin{array}{rcl} S & = & G\ \mathrm{a} \\ F\ f\ x & = & f\ (f\ x) \\ G\ f & = & \mathrm{b}\ (f\ \mathrm{c})\ (G\ (F\ f)) \end{array}$$

Ga

Higher-order recursion schemes

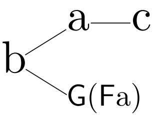
Higher-order recursion schemes are an abstract form of functional programs, often used as generators for infinite trees.

Example

With:

- Non-terminals $S : o, F : (o \to o) \to o \to o, G : (o \to o) \to o$,
- Terminals $\mathrm{a} : o \to o$, $\mathrm{b} : o \to o \to o$, $\mathrm{c} : o$.

$$
\begin{array}{rcl}
S & = & G\ \mathrm{a} \\
F\ f\ x & = & f\ (f\ x) \\
G\ f & = & \mathrm{b}\ (f\ \mathrm{c})\ (G\ (F\ f))
\end{array}
$$



Higher-order recursion schemes

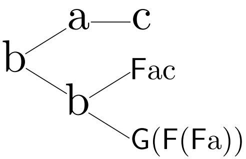
Higher-order recursion schemes are an abstract form of functional programs, often used as generators for infinite trees.

Example

With:

- Non-terminals $S : o, F : (o \to o) \to o \to o, G : (o \to o) \to o$,
- Terminals $\mathrm{a} : o \to o$, $\mathrm{b} : o \to o \to o$, $\mathrm{c} : o$.

$$\begin{aligned} S &= G\ \mathrm{a} \\ F\ f\ x &= f\ (f\ x) \\ G\ f &= \mathrm{b}\ (f\ \mathrm{c})\ (G\ (F\ f)) \end{aligned}$$



Higher-order recursion schemes

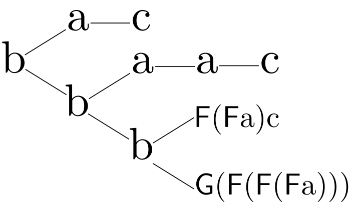
Higher-order recursion schemes are an abstract form of functional programs, often used as generators for infinite trees.

Example

With:

- Non-terminals $S : o, F : (o \to o) \to o \to o, G : (o \to o) \to o$,
- Terminals $\mathrm{a} : o \to o$, $\mathrm{b} : o \to o \to o$, $\mathrm{c} : o$.

$$\begin{array}{rcl} S & = & G\ \mathrm{a} \\ F\ f\ x & = & f\ (f\ x) \\ G\ f & = & \mathrm{b}\ (f\ \mathrm{c})\ (G\ (F\ f)) \end{array}$$



Higher-order recursion schemes

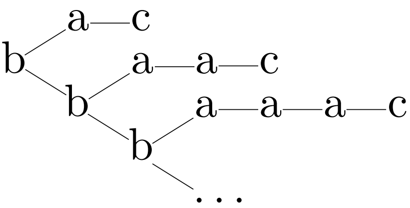
Higher-order recursion schemes are an abstract form of functional programs, often used as generators for infinite trees.

Example

With:

- Non-terminals $S : o, F : (o \to o) \to o \to o, G : (o \to o) \to o$,
- Terminals $\mathrm{a} : o \to o$, $\mathrm{b} : o \to o \to o$, $\mathrm{c} : o$.

$$\begin{array}{rcl} S & = & G\ \mathrm{a} \\ F\ f\ x & = & f\ (f\ x) \\ G\ f & = & \mathrm{b}\ (f\ \mathrm{c})\ (G\ (F\ f)) \end{array}$$



Alternative presentation : the λY-calculus

Types. Simple types on one atom o.

$$\theta, \theta' ::= o \mid \theta \to \theta'$$

Terms.

$$M, N ::= x \mid \lambda x^{\theta}.M \mid M\ N \mid Y_{\theta}$$

Typing rules.

$$\frac{}{\Gamma, x : \theta \vdash x : \theta} \qquad \frac{}{\Gamma \vdash Y_{\theta} : (\theta \to \theta) \to \theta} \qquad \frac{\Gamma, x : \theta \vdash M : \theta'}{\Gamma \vdash \lambda x^{\theta}.M : \theta \to \theta'}$$

$$\frac{\Gamma \vdash M : \theta \to \theta' \qquad \Gamma \vdash N : \theta}{\Gamma \vdash M\ N : \theta'}$$

Reduction.

$$\begin{array}{rcl} (\lambda x^{\theta}.M)\ N & \to_{\beta} & M[N/x] \\ Y_{\theta}\ M & \to_{\delta} & M\ (Y_{\theta}\ M) \\ M & \to_{\eta} & \lambda x^{\theta}.M\ x \end{array}$$

(in the last, $x \notin fv(M)$ and M has type $\theta \to \theta'$)

Relationship of HORS and λY

Example

$$\begin{array}{rcl} S & = & G\ \mathrm{a} \\ F\ f\ x & = & f\ (f\ x) \\ G\ f & = & \mathrm{b}\ (f\ \mathrm{c})\ (G\ (F\ f)) \end{array}$$

Relationship of HORS and λY

Example

$$
\begin{array}{rcl}
S & = & G\ \mathrm{a} \\
F & = & \lambda f.\lambda x.f\ (f\ x) \\
G & = & \lambda f.\mathrm{b}\ (f\ \mathrm{c})\ (G\ (F\ f))
\end{array}
$$

Relationship of HORS and λY

Example

$$
\begin{aligned}
S &= G\ \mathrm{a} \\
G &= \lambda f.\mathrm{b}\ (f\ \mathrm{c})\ (G\ (\lambda x.f\ (f\ x)))
\end{aligned}
$$

Relationship of HORS and λY

Example

$$
\begin{aligned}
S &= G\ \mathrm{a} \\
G &= Y\ (\lambda G.\lambda f.\mathrm{b}\ (f\ \mathrm{c})\ (G\ (\lambda x.f\ (f\ x))))
\end{aligned}
$$

Relationship of HORS and λY

Example

$$S \quad = \quad Y\ (\lambda G.\lambda f.\mathrm{b}\ (f\ \mathrm{c})\ (G\ (\lambda x.f\ (f\ x))))\ \mathrm{a}$$

Relationship of HORS and λY

Example

$$S \quad = \quad Y\ (\lambda G.\lambda f.\mathrm{b}\ (f\ \mathrm{c})\ (G\ (\lambda x.f\ (f\ x))))\ \mathrm{a}$$

Which is a λY-term of type o in context

$$\mathrm{a} : o \to o, \mathrm{b} : o \to o \to o, \mathrm{c} : o$$

We write $\Gamma_{\leqslant 1}$ for such contexts of order less than 1.

Relationship of HORS and λY

Example

$$S \quad = \quad Y\ (\lambda G.\lambda f.\mathrm{b}\ (f\ \mathrm{c})\ (G\ (\lambda x.f\ (f\ x))))\ \mathrm{a}$$

Which is a λY-term of type o in context

$$\mathrm{a} : o \to o, \mathrm{b} : o \to o \to o, \mathrm{c} : o$$

We write $\Gamma_{\leqslant 1}$ for such contexts of order less than 1.

Proposition (Salvati, Walukiewicz)

There is a correspondence between HORS and λY-terms:

$$\Gamma_{\leqslant 1} \vdash M : o$$

Applications to program verification (Kobayashi)

Theorem (Ong, 06)

Monadic Second-Order logic (MSO) is decidable on infinite trees generated by HORS.

Example (Kobayashi)

Application to verification of correct resource usage.

```
let rec g() = if _ then close() else (read(); g()) in g()
```

$$Y\ (\lambda G.\lambda k.\mathrm{br}\ (\mathrm{close}\ k)\ (\mathrm{read}\ (G\ k)))\ \bullet$$

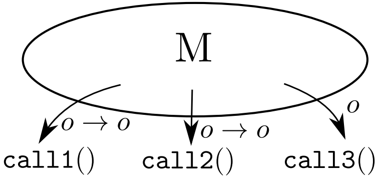
with terminals:

$$\begin{aligned} \mathrm{br} &: o \to o \to o \\ \mathrm{read} &: o \to o \\ \mathrm{close} &: o \to o \\ \bullet &: o \end{aligned}$$

One can automatically check that all finite paths have the form $\mathrm{read}^*\mathrm{close}$.

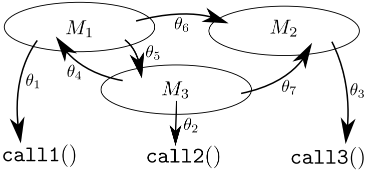
Higher-order model checking for open programs?

Closed programs *vs.* open programs. HORS naturally represent the behaviour of **closed** programs.



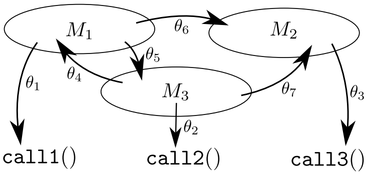
Higher-order model checking for open programs?

Closed programs *vs.* open programs. HORS naturally represent the behaviour of **closed** programs.



Higher-order model checking for open programs?

Closed programs *vs.* open programs. HORS naturally represent the behaviour of **closed** programs.



Question

Can we represent **open** *simply-typed functional programs as HORS?*

What representation for open programs?

Representation of open programs

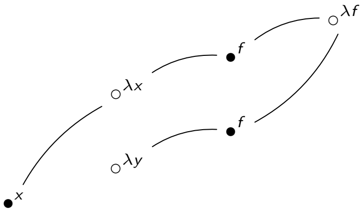
Game semantics.

- **Computation** is a game between the *program* and the *environment*.
- A program M is represented as a **strategy** describing all its interactions against arbitrary environments.
- A purely functional program M is represented as an **innocent strategy**.

Innocent strategies and Böhm trees.

- *Innocent strategies* are abstract representations of **Böhm trees**.

$$\lambda f.f(\lambda x.f(\lambda y.x)) : \ ((o \longrightarrow o) \longrightarrow o) \longrightarrow o$$



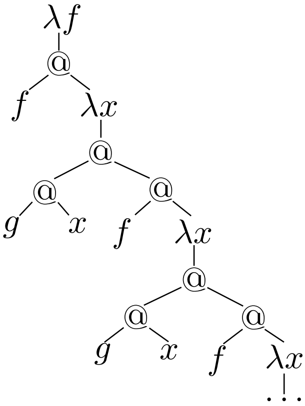
- Böhm trees are notions of infinite normal forms for recursive higher-order open programs.
- We want to generate *Böhm trees* as HORS.

Böhm trees

Consider the term:

$$g : o \to o \to o \vdash \lambda f^{(o \to o) \to o}.Y_o\ (\lambda y^o.f\ (\lambda x^o.g\ x\ y)) : ((o \to o) \to o) \to o$$

Its Böhm tree starts with:

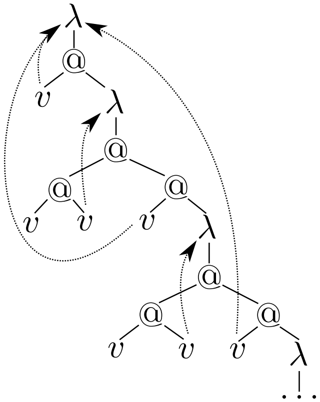


Böhm trees

Consider the term:

$$g : o \to o \to o \vdash \lambda f^{(o\to o)\to o}.Y_o\ (\lambda y^o.f\ (\lambda x^o.g\ x\ y)) : ((o \to o) \to o) \to o$$

Its Böhm tree starts with:



How can we generate representations of pointers within HORS?

De Bruijn levels

Definition

De Bruijn levels are a variable naming convention where:

- Variable names are natural numbers,
- Each variable is given the smallest index not yet present in the context.

Example

The term

$$g : o \to o \to o \vdash \lambda f.f\ (\lambda x.g\ x\ (f\ (\lambda x.g\ x\ (f\ x))$$

is represented by:

$$0 : o \to o\ \ \to o \vdash \lambda 1.1\ (\lambda 2.0\ 2\ (1\ (\lambda 3.0\ 3\ (1\ 3))))$$

Proposition

Two terms M and M' have the same De Bruijn levels representation iff they are α-equivalent.

(not to be confused with **De Bruijn indices**)

Our result

Question

Can we relate HORS and arbitrary Böhm trees?

$$\begin{tikzcd} \Gamma_{\leqslant 1} \vdash M : o \arrow[rr, bend left, "\subseteq"] & & \Gamma \vdash M : \theta \arrow[ll, bend left, dotted, "?"] \end{tikzcd}$$

Theorem

For any λY-term $\Gamma \vdash M : \theta$ there is a term:

$$\Gamma_{rep} \vdash M_{rep} : o$$

with

$$\Gamma_{rep} = \{\, z : o,\ succ : o \to o,\ var : o \to o,\ app : o \to o \to o,\ lam : o \to o \to o \,\}$$

such that M_{rep} evaluates a representation of the Böhm tree of M where binders are represented by **De Bruijn levels**.

We also prove the same result for terms of finitary PCF (PCF_f).

II. Our construction

II.1 Formal statement of the result

Partial terms and Böhm trees

Definition

- The $\lambda\perp$-calculus is the simply-typed λ-calculus over a base type o and a constant $\perp : o$.
- If $\Gamma \vdash M : \theta$ is a $\lambda\perp$-term, its **Böhm tree** $BT(M)$ is its η-long β-normal form.
- The $\lambda\perp^{\infty}$-calculus consists of (potentially) infinite $\lambda\perp$-terms. Terms of the $\lambda\perp^{\infty}$-calculus forms a *complete partial order*.

Definition

If $\Gamma \vdash M : \theta$ is a λY-term, its n**-th approximation** is the $\lambda\perp$-term defined as:

$$\begin{array}{rcl@{\qquad\qquad}rcl} x \upharpoonright n & = & x & \lambda x.M \upharpoonright n & = & \lambda x.(M \upharpoonright n) \\ M\ N \upharpoonright n & = & (M \upharpoonright n)\ (N \upharpoonright n) & Y_\theta \upharpoonright n & = & \lambda f^{\theta\to\theta}.f^n(\perp_\theta) \end{array}$$

Definition

If $\Gamma \vdash M : \theta$ is a λY-term, its **Böhm tree** is defined as:

$$BT(M) = \bigsqcup_n BT(M \upharpoonright n)$$

Representation of De Bruijn terms in λY

We represent terms with binders as Böhm trees of type o in the context:

$$\Gamma_{rep} = \{\, z : o,\ succ : o \to o,\ var : o \to o,\ app : o \to o \to o,\ lam : o \to o \to o \,\}$$

Definition

Let $\nu : \Gamma \to \mathbb{N}$ associate a index to all free variables of M. Writing $\overline{n} = succ^n\ z$, we define

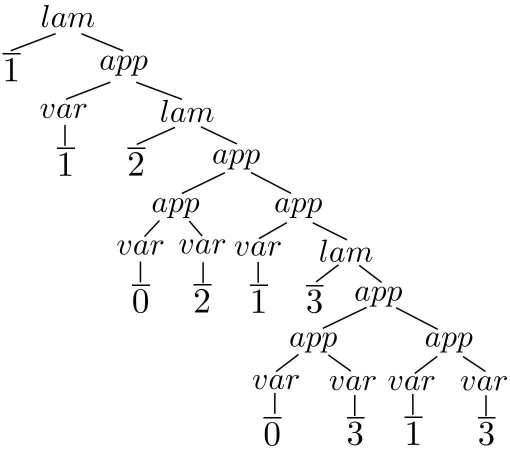
$$\begin{array}{rcl}
\mathrm{rep}_\nu(\perp, n) & = & \perp \\
\mathrm{rep}_\nu(\lambda x.M, n) & = & lam\ \overline{n+1}\ \mathrm{rep}_{\nu \oplus \{x \mapsto n+1\}}(M, n+1) \\
\mathrm{rep}_\nu(x, n) & = & var\ \overline{\nu(x)} \\
\mathrm{rep}_\nu(MN, n) & = & app\ \mathrm{rep}_\nu(M, n)\ \mathrm{rep}_\nu(N, n)
\end{array}$$

For $\Gamma \vdash M : \theta$ with $\Gamma = \{x_1 : \theta_1, \ldots, x_n : \theta_n\}$, setting $\nu(x_i) = i - 1$, we define

$$\mathrm{rep}(M) = \mathrm{rep}_\nu(M, n)$$

Example

Representation of $g : o \to o \to o \vdash \lambda f.f\ (\lambda x.g\ x\ (f\ (\lambda x.g\ x\ (f\ x)).$



(Keeping in mind that $\overline{n} = succ\ (succ\ \ldots\ (succ\ z)\ldots))$

Formal statement

Theorem

Let $\Gamma \vdash M : \theta$ be a λY-term. There exists a λY-term $\Gamma_{rep} \vdash M_{rep} : o$ (a HORS) such that:

$$BT(M_{rep}) = \mathrm{rep}(BT(M))$$

Write θ^* for $\theta[o \to o/o]$ and $M^* = M[o \to o/o]$. There is a finite λ-term:

$$\Gamma_{rep} \vdash \downarrow_\theta : \theta^* \to o \to o$$

Such that for $\vdash M : \theta$, setting:

$$M_{rep} = \downarrow_\theta \; M^* \; \overline{0}$$

validates the above theorem.

II.2 Derivation of the construction

Normalization by evaluation

Semantic technique for computing normal forms of λ-terms.

Theorem (Berger, Schwichtenberg)

There is a set-theoretic interpretation of the simply-typed λ-calculus:

$$[\![-]\!] : \Lambda \to \mathrm{Set}$$

and for each type θ, a function

$$\mathrm{nbe} : [\![\theta]\!] \to \Lambda$$

such that for each term $\vdash M : \theta$,

$$\mathrm{nbe}([\![M]\!]) \cong_{\beta\eta} M$$

is the β-normal η-long form of M.

Normalization by evaluation for the simply-typed λ-calculus

Step 1: Interpretation. Let E be a set containing representations of terms.

$$\begin{array}{rcl} [\![o]\!] & = & E \\ [\![x]\!]_\rho & = & \rho(x) \\ [\![M\ N]\!]_\rho & = & [\![M]\!]_\rho([\![N]\!]_\rho) \end{array} \qquad \begin{array}{rcl} [\![\theta \to \theta']\!] & = & [\![\theta]\!] \to [\![\theta']\!] \\ [\![\lambda x^\theta . M]\!]_\rho & = & \boldsymbol{\lambda} a^{[\![\theta]\!]} . [\![M]\!]_{\rho \oplus \{x \mapsto a\}} \end{array}$$

Where all the right hand side operations are operations on sets and functions.

Step 2: Reification. The normal form of $\vdash M : \theta$ can be **extracted** from $[\![M]\!]$ by the following:

$$\begin{array}{rcl} \Downarrow_\theta & : & [\![\theta]\!] \to E \\ \Downarrow_o x & = & x \\ \Downarrow_{\theta_1 \to \theta_2} x & = & \mathit{lam}\ n\ \Downarrow_{\theta_2} (x\ (\Uparrow_{\theta_1} (\mathit{var}\ n))) \quad (n \text{ fresh}) \end{array}$$

$$\begin{array}{rcl} \Uparrow_\theta & : & E \to [\![\theta]\!] \\ \Uparrow_o e & = & e \\ \Uparrow_{\theta_1 \to \theta_2} e & = & \boldsymbol{\lambda} x^{[\![\theta_1]\!]} . \Uparrow_{\theta_2} \mathit{app}\ e\ (\Downarrow_{\theta_2} x) \end{array}$$

by setting $\mathrm{nbe(M)} = \Downarrow_\theta [\![M]\!]$.

Example

$$
\begin{array}{rcl}
\Downarrow_{o\to o\to o} [\![\lambda x^o.\lambda y^o.x]\!] & = & \mathit{lam}\ 0\ (\Downarrow_{o\to o} [\![\lambda x^o.\lambda y^o.x]\!]\ (\Uparrow_o (\mathit{var}\ 0))) \\
& = & \mathit{lam}\ 0\ (\Downarrow_{o\to o} [\![\lambda x^o.\lambda y^o.x]\!]\ (\mathit{var}\ 0)) \\
& = & \mathit{lam}\ 0\ (\mathit{lam}\ 1\ (\Downarrow_o [\![\lambda x^o.\lambda y^o.x]\!]\ (\mathit{var}\ 0)\ (\Uparrow_o (\mathit{var}\ 1)))) \\
& = & \mathit{lam}\ 0\ (\mathit{lam}\ 1\ ([\![\lambda x^o.\lambda y^o.x]\!]\ (\mathit{var}\ 0)\ (\mathit{var}\ 1))) \\
& = & \mathit{lam}\ 0\ (\mathit{lam}\ 1\ ((\boldsymbol{\lambda} a^E.\boldsymbol{\lambda} b^E.a)\ (\mathit{var}\ 0)\ (\mathit{var}\ 1))) \\
& = & \mathit{lam}\ 0\ (\mathit{lam}\ 1\ (\mathit{var}\ 0))
\end{array}
$$

Remarks.

- Normal form obtained by **evaluation** in the model,
- Need for generation of fresh variable indices.

Generating De Bruijn levels (Berger, Schwichtenberg)

Expressions. $e \in E$ are replaced with **indexed expressions**

$$f \in \mathbb{N} \to E = \widehat{E}$$

Constructors. *var*, *lam*, *app* are replaced with compositional variants.

$$\begin{array}{rcl}
\widehat{var} & = & \boldsymbol{\lambda} v^N.\boldsymbol{\lambda} n^N.var\ v : N \to \widehat{E} \\
\widehat{app} & = & \boldsymbol{\lambda} e_1^{\widehat{E}}.\boldsymbol{\lambda} e_2^{\widehat{E}}.\boldsymbol{\lambda} n^N.app\ (e_1\ n)\ (e_2\ n) : \widehat{E} \to \widehat{E} \to \widehat{E} \\
\widehat{lam} & = & \boldsymbol{\lambda} f^{N \to \widehat{E}}.\boldsymbol{\lambda} n^N.lam\ n\ (f\ n\ (succ\ n)) : (N \to \widehat{E}) \to \widehat{E}
\end{array}$$

Reify and reflect. They are generalized:

$$\begin{array}{rclcrcl}
\Downarrow_o x & = & x & \quad & \Downarrow_{\theta_1 \to \theta_2} x & = & \widehat{lam}\ (\boldsymbol{\lambda} n^N. \Downarrow_{\theta_2} (x\ (\Uparrow_{\theta_1}\ \widehat{var}\ n))) \\
\Uparrow_o e & = & e & & \Uparrow_{\theta_1 \to \theta_2} e & = & \boldsymbol{\lambda} x^{[\![\theta_1]\!]}. \Uparrow_{\theta_2} \widehat{app}\ e\ (\Downarrow_{\theta_2} x)
\end{array}$$

Normalization by evaluation. The interpretation $[\![-]\!]$ is now based on $\widehat{E}$ instead of E. NBE is obtained for $\vdash M : \theta$ by:

$$\mathrm{nbe}(M) = \Downarrow_\theta [\![M]\!]\ 0$$

Continuity and NBE for λY

Model. Standard pointed ω-cpo model of the λY-calculus:

$$\begin{array}{rcl} [\![o]\!] & = & \widehat{E} \\ [\![Y_\theta]\!] & = & \boldsymbol{\lambda} f^\theta . \bigsqcup_n f^n(\bot) \end{array}$$

where E is the pointed ω-cpo of possibly infinite expressions.

Normalization by evaluation. From a λY-term $\vdash M : \theta$,

$$\mathrm{nbe}(M) = \Downarrow_\theta [\![M]\!]\ 0 \in E$$

$\hookrightarrow$ Obtain an infinite normal form.

Proof. By continuity, and validity of NBE on the λ-calculus.

Internalization

The semantic ingredients used in NBE for λY can be expressed within the λY-calculus.

Expressions are terms of λY:

$$\Gamma_{rep} \vdash M : o$$

Term families is the type $\widehat{E} = o \to o$.

Interpretation is the substitution $\theta^* = \theta[o \to o/o]$ and $M^* = M[o \to o/o]$.

Term formers are the following:

$$\begin{array}{rcl} \widehat{var} & = & \lambda v^o.\lambda n^o.var\ v \\ \widehat{lam} & = & \lambda f^{o \to o}.\lambda n^o.lam\ n\ (f\ \ n\ (succ\ n)) \\ \widehat{app} & = & \lambda e_1^o.\lambda e_2^o.\lambda n^o.app\ (e_1\ \ n)\ (e_2\ \ n) \end{array}$$

Reify/reflect are now terms of the λY-calculus:

$$\begin{array}{rcl} \downarrow_o & = & \lambda x^o.x \\ \uparrow_o & = & \lambda e^o.e \end{array} \qquad \begin{array}{rcl} \downarrow_{\theta_1 \to \theta_2} & = & \lambda x^{\theta_1^* \to \theta_2^*}.\widehat{lam}\ (\boldsymbol{\lambda} n^N.\downarrow_{\theta_2} (x\ (\uparrow_{\theta_1}\ \widehat{var}\ \ n))) \\ \uparrow_{\theta_1 \to \theta_2} & = & \lambda e^o.\lambda x^{\theta_1^*}.\uparrow_{\theta_2} \widehat{app}\ e\ (\downarrow_{\theta_2}\ x) \end{array}$$

Internalization

Theorem

If $\vdash M : \theta$ is a λY-term, then the term M_{rep} defined as:

$$\Gamma_{rep} \vdash \downarrow_\theta \; M^* \; \overline{0} : o$$

satisfies:

$$BT(M_{rep}) = \mathrm{rep}(BT(M))$$

Proof.

$$\begin{tikzcd}
\lambda Y \arrow[r, "(-)^*"] \arrow[dr, "[\![-]\!]"'] & \lambda Y \arrow[r, "\downarrow_\theta - \overline{0}"] \arrow[d, "[\![-]\!]'"] & \mathrm{HORS} \arrow[d, "\{\!|-|\!\}"] \\
& \omega\text{-cpo} \arrow[r, "\Downarrow_\theta - 0"'] & E
\end{tikzcd}$$

$\square$

Internalization

Theorem

If $\vdash M : \theta$ *is a* λY*-term, then the term* M_{rep} *defined as:*

$$\Gamma_{rep} \vdash \downarrow_\theta \; M^* \; \overline{0} : o$$

satisfies:

$$BT(M_{rep}) = \mathrm{rep}(BT(M))$$

Proof.

$$\begin{tikzcd}
\lambda Y \arrow[r, "(-)^*"] \arrow[dr, "\llbracket - \rrbracket"'] & \lambda Y \arrow[r, "\downarrow_\theta - \overline{0}"] \arrow[d, "\llbracket - \rrbracket'"] & \mathrm{HORS} \arrow[d, "BT(-)"] \\
 & \omega\text{-cpo} \arrow[r, "\Downarrow_\theta - 0"'] & E
\end{tikzcd}$$

□

II.3 Extension to PCF_f

Extension to PCF_f

Definition

The types and terms of PCF_f are defined as follows.

$$
\begin{array}{rcl}
\theta, \theta' & ::= & \mathrm{B} \mid \theta \to \theta' \\
M, N & ::= & x \mid \lambda x^{\theta}.M \mid M\ N \mid Y_{\theta} \\
 & & tt \mid f\!f \mid \mathrm{if}\ M\ \mathrm{then}\ N\ \mathrm{else}\ N'
\end{array}
$$

equipped with the standard operational semantics.

Definition (PCF Böhm trees)

The notion of (infinite) normal forms we consider is:

$$
\frac{}{\Gamma \vdash \bot : \mathrm{B}} \qquad \frac{}{\Gamma \vdash tt : \mathrm{B}} \qquad \frac{}{\Gamma \vdash f\!f : \mathrm{B}} \qquad \frac{\Gamma, x_1 : A_1, \ldots, x_n : A_n \vdash M : \mathrm{B}}{\Gamma \vdash \lambda \overrightarrow{x}.M : \overrightarrow{A} \to \mathrm{B}}
$$

$$
\frac{\Gamma \vdash M_i : \theta_i \quad (1 \leqslant i \leqslant n) \qquad \Gamma \vdash N_1 : \mathrm{B} \qquad \Gamma \vdash N_2 : \mathrm{B} \qquad (x : \overrightarrow{\theta} \to \mathrm{B}) \in \Gamma}{\Gamma \vdash \mathrm{if}\ x\ \overrightarrow{M}\ \mathrm{then}\ N_1\ \mathrm{else}\ N_2 : \mathrm{B}}
$$

They correspond to innocent strategies.

The NBE translation for PCF_f

Representation. In the ω-cpo E of infinitary terms $\Gamma_{pcf} \vdash M : o$, with:

$$\Gamma_{pcf} = \Gamma_{rep} \cup \{tt : o, ff : o, if : o \to o \to o \to o\}$$

Semantics. Standard domain semantics of PCF, based on:

$$[\![\mathrm{B}]\!] = \widehat{E} \to \widehat{E} \to \widehat{E}$$

Reflect and reify. Direct adaptations of those for λY.

Internalization. Follows the same lines as for λY.

Normal forms. The normal forms generated are **infinitary** PCF **Böhm trees**, or equivalently, **innocent strategies**.

III. Consequences and conclusions

Consequences

Corollary

The following problems are equivalent:

(1) *Equivalence of HORS,*

(2) *Böhm tree equivalence for λY,*

(3) *Game semantics equivalence for PCF_f,*

(4) *Distinguishability by contexts with state and control operators for terms of PCF_f.*

By MSO model-checking on HORS, we also get:

Corollary

The following problems are decidable for PCF_f and λY terms:

(1) *Normalizability, solvability,*

(2) *Finiteness,*

(3) *Finite prefix.*

Conclusions

Contributions.

- Representation of arbitrary Böhm trees as HORS,
- NBE: internalization of NBE for $\lambda Y/\mathrm{PCF}_f$ inside λY.

Future work.

- Transport decidability/undecidability results,
- Model-checking strategies, compositional model-checking,
- Similar representation results for other notions of observation.

Thank you.