Recurrence Extraction for Functional Programs Through Call-by-Push-Value

Alex Kavvos
Department of Computer Science, Aarhus University

j.w.w. Edward Morehouse, Daniel Licata, and Norman Danner

Athens Programming Language Seminar, 27 Dec 2019

arXiv:1911.04588

Recurrence Extraction for Functional Programs Through Call-by-Push-Value

Alex Kavvos

Department of Mathematics and Computer Science, Wesleyan University

j.w.w. Edward Morehouse, Daniel Licata, and Norman Danner

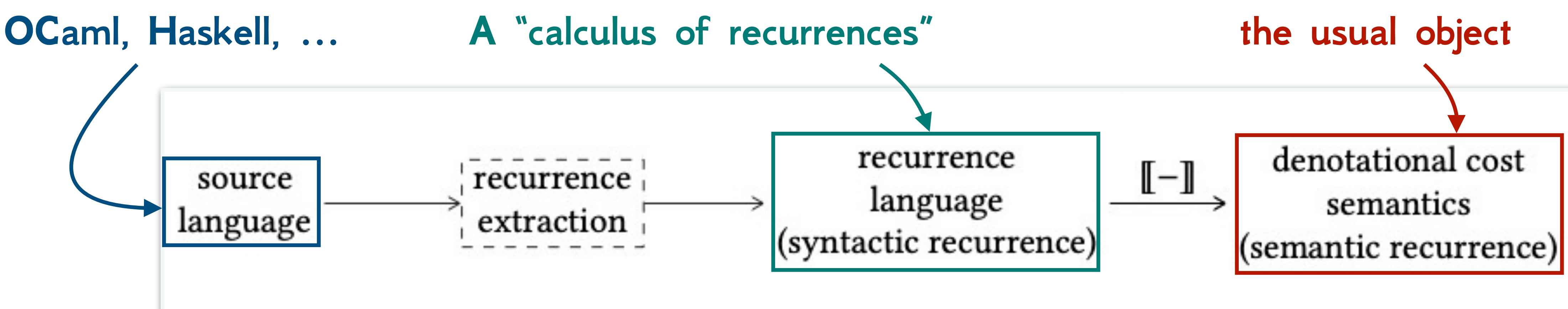
Athens Programming Language Seminar, 27 Dec 2019

arXiv:1911.04588

Recurrence Extraction for FP

- To evaluate the complexity of a program, we **extract** a **recurrence relation**, and then we **solve** it — it can compute running time **exactly**, or approximately.

- **Recurrence extraction** is about making the first step **formal** and **algorithmic**.



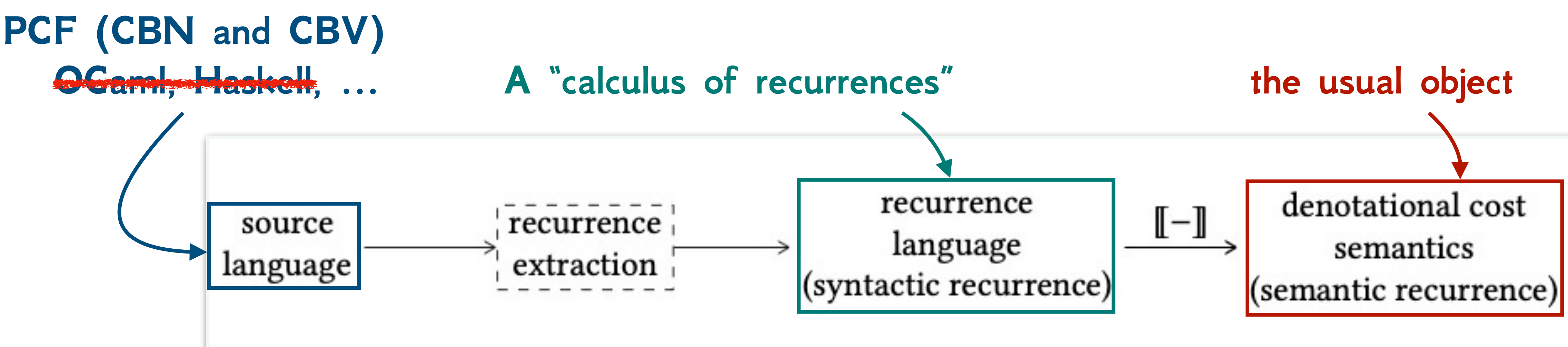
- Solved for **inductive types** in a **total language** [Danner et al. @ ICFP 2015]

- But how can we do it for **recursive** functional programs — both CBN and CBV?

Recurrence Extraction for FP

- To evaluate the complexity of a program, we **extract** a **recurrence relation**, and then we **solve** it — it can compute running time **exactly**, or approximately.

- **Recurrence extraction** is about making the first step **formal** and **algorithmic**.



- Solved for **inductive types** in a **total language** [Danner et al. @ ICFP 2015]

- But how can we do it for **recursive** functional programs — both CBN and CBV?

Recurrence Extraction for FP

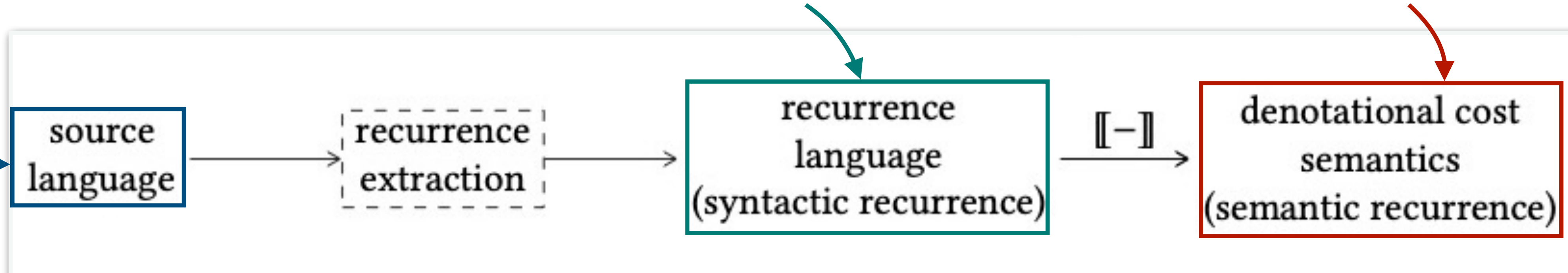
- To evaluate the complexity of a program, we **extract** a **recurrence relation**, and then we **solve** it — it can compute running time **exactly**, or approximately.

- **Recurrence extraction** is about making the first step **formal** and **algorithmic**.

PCF (CBN and CBV)
~~OCaml, Haskell,~~ …

PCF with costs – CBN only (≈ maths)
~~A "calculus of recurrences"~~

the usual object



- Solved for **inductive types** in a **total language** [Danner et al. @ ICFP 2015]

- But how can we do it for **recursive** functional programs — both CBN and CBV?

Recurrence Extraction for FP

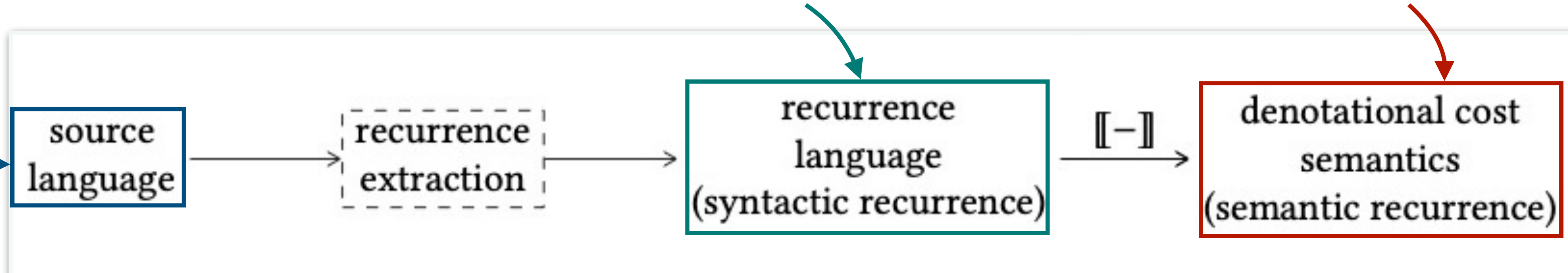
- To evaluate the complexity of a program, we **extract** a **recurrence relation**, and then we **solve** it — it can compute running time **exactly**, or approximately.

- **Recurrence extraction** is about making the first step **formal** and **algorithmic**.

PCF (CBN and CBV)
~~OCaml, Haskell~~, …

PCF with costs – **CBN** only ($\approx$ maths)
~~A "calculus of recurrences"~~

sized domains
~~the usual object~~



- Solved for **inductive types** in a **total language** [Danner et al. @ ICFP 2015]

- But how can we do it for **recursive** functional programs — both CBN and CBV?

PCF

= simply-typed λ-calculus + fixpoints (CBN & CBV)

Constants

$$\frac{n \in \mathbb{N}}{\Gamma \vdash \underline{n} : \mathsf{nat}} \qquad \frac{\Gamma \vdash M, N : \mathsf{nat} \quad \mathbf{op} \in \{+, *, -, \div, \mathrm{mod}\}}{\Gamma \vdash M \;\mathbf{op}\; N : \mathsf{nat}}$$

Function Types

$$\frac{\Gamma, x : A \vdash M : B}{\Gamma \vdash \lambda x.\ M : A \to B} \qquad \frac{\Gamma \vdash M : A \to B \quad \Gamma \vdash N : A}{\Gamma \vdash M\,N : B}$$

CBN recursion

$$\frac{\Gamma, x : A \vdash M : A}{\Gamma \vdash \mathsf{fix}\ x.\ M : A}$$

CBV recursion

$$\frac{\Gamma, f : A \to B, x : A \vdash M : B}{\Gamma \vdash \mathsf{rec}\, f(x) = M : A \to B}$$

CBN big-step rules

$$\frac{M \downarrow^m \lambda x.\ P \quad P[N/x] \downarrow^n V}{M\,N \downarrow^{m+n+1} V} \qquad \frac{M[\mathsf{fix}\ x.\ M/x] \downarrow^n V}{\mathsf{fix}\ x.\ M \downarrow^{n+1} V}$$

CBV big-step rules

$$\frac{M \downarrow^m \lambda x.\ P \quad N \downarrow^n W \quad P[W/x] \downarrow^k V}{M\,N \downarrow^{m+n+k+1} V}$$

$$\frac{M \downarrow^m \mathsf{rec}\, f(x) = P \quad N \downarrow^n W \quad P[\mathsf{rec}\, f(x) = P/f, W/x] \downarrow^k V}{M\,N \downarrow^{m+n+k+1} V}$$

PCFc

= simply-typed λ + fixpoints + costs (CBN only)

Constants

$$\frac{n \in \mathbb{N}}{\Gamma \vdash \underline{n} : \mathsf{nat}} \qquad \frac{\Gamma \vdash M, N : \mathsf{nat} \quad \mathbf{op} \in \{+, *, -, \div, \mathrm{mod}\}}{\Gamma \vdash M \;\mathbf{op}\; N : \mathsf{nat}}$$

Function Types

$$\frac{\Gamma, x : A \vdash M : B}{\Gamma \vdash \lambda x.\, M : A \to B} \qquad \frac{\Gamma \vdash M : A \to B \quad \Gamma \vdash N : A}{\Gamma \vdash M\, N : B}$$

CBN recursion

$$\frac{\Gamma, x : A \vdash M : A}{\Gamma \vdash \mathsf{fix}\ x.\, M : A}$$

CBV recursion

$$\frac{\Gamma, f : A \to B, x : A \vdash M : B}{\Gamma \vdash \mathsf{rec}\, f(x) = M : A \to B}$$

Costs

$$\frac{\widehat{n} \in \{\mathbf{0}, \mathbf{1}\}}{\Gamma \vdash \widehat{n} : \mathbb{C}} \qquad \frac{\Gamma \vdash M : \mathbb{C} \quad \Gamma \vdash N : \mathbb{C}}{\Gamma \vdash M \boxplus N : \mathbb{C}}$$

CBN big-step rules

$$\frac{M \Downarrow^{\cancel{m}} \lambda x.\, P \quad P[N/x] \Downarrow^{\cancel{n}} V}{M\,N \Downarrow^{\cancel{m+n+1}} V} \qquad \frac{M[\mathsf{fix}\ x.\, M/x] \Downarrow^{\cancel{n}} V}{\mathsf{fix}\ x.\, M \Downarrow^{\cancel{n+1}} V}$$

Cost big-step rules

$$\widehat{n} \stackrel{\text{def}}{=} \underbrace{\mathbf{1} \boxplus \mathbf{1} \boxplus \ldots \boxplus \mathbf{0}}_{\text{right-associated}}$$

$$\frac{\widehat{n} \in \{\mathbf{0}, \mathbf{1}\}}{\widehat{n} \Downarrow \widehat{n}} \qquad \frac{M \Downarrow \widehat{m} \quad N \Downarrow \widehat{n}}{M \boxplus N \Downarrow \widehat{m+n}}$$

The Size Preorder

$$\Gamma \vdash M \leqslant N : A$$

"as recurrences, M is less than or equal to N at type A"

The central device of PCFc is the size preorder. Its rules are dictated by the proof of correctness of the extraction.

Extraction for CBV

- Every PCF type is translated to **two** PCFc types:
  - A type of **potentials** (= "size," use-cost)
  - A type of **complexities** (= cost + potential)
- We then extract a **complexity** from each term. Given $E : ||A||$ we write

$$E_c \stackrel{\text{def}}{=} \pi_1(E) : \mathbb{C}$$
$$E_p \stackrel{\text{def}}{=} \pi_2(E) : \langle\!\langle A \rangle\!\rangle$$
$$C +_c E \stackrel{\text{def}}{=} \langle C \boxplus E_c, E_p \rangle : ||A||$$

Complexity of A **Potential of A**

$$||A|| \stackrel{\text{def}}{=} \mathbb{C} \times \langle\!\langle A \rangle\!\rangle$$

$$\langle\!\langle \mathsf{nat} \rangle\!\rangle \stackrel{\text{def}}{=} \mathsf{nat}$$
$$\langle\!\langle A_1 \times A_2 \rangle\!\rangle \stackrel{\text{def}}{=} \langle\!\langle A_1 \rangle\!\rangle \times \langle\!\langle A_2 \rangle\!\rangle$$
$$\langle\!\langle A \to B \rangle\!\rangle \stackrel{\text{def}}{=} \langle\!\langle A \rangle\!\rangle \to ||B||$$

$$||x|| \stackrel{\text{def}}{=} \langle \mathbf{0}, x \rangle$$
$$||M - \underline{n}|| \stackrel{\text{def}}{=} \left\langle ||M||_c \,, ||M||_p - \underline{n} \right\rangle$$
$$||M - N|| \stackrel{\text{def}}{=} \left\langle ||M||_c \boxplus ||N||_c \,, ||M||_p \right\rangle \text{ (if } N \neq \underline{n})$$
$$||\lambda x.\ M|| \stackrel{\text{def}}{=} \langle \mathbf{0}, \lambda x. ||M|| \rangle$$
$$||M\ N|| \stackrel{\text{def}}{=} \mathbf{1} \boxplus ||M||_c \boxplus ||N||_c +_c (||M||_p\ ||N||_p)$$
$$||\mathsf{rec}\, f(x) = M|| \stackrel{\text{def}}{=} \langle \mathbf{0}, \mathsf{fix}\ f.\ \lambda x.\ ||M|| \rangle$$

Given potentials for the inputs

this recurrence

gives the complexity of M

Theorem: $\Gamma \vdash M : A \Longrightarrow \langle\!\langle \Gamma \rangle\!\rangle \vdash ||M|| : ||A||$

How to state correctness for CBV

- Because of function types, we must use a logical relation.
- **Theorem**: there exist relations

$$(M : \mathcal{T}_A^{\mathsf{PCF,closed}}) \precsim_A (E : \mathcal{T}_{\|A\|}^{\mathsf{PCF}c\mathsf{,closed}}) \quad \text{and} \quad (V : \mathcal{V}_A^{\mathsf{PCF,closed}}) \precsim_A^{\mathrm{val}} (E : \mathcal{T}_{\langle\!\langle A \rangle\!\rangle}^{\mathsf{PCF}c\mathsf{,closed}})$$

("M/V is bounded above by recurrence E"), such that

$$\begin{array}{lcl}
M \precsim_A E & \Longrightarrow & \text{if } E_c{\downarrow} \text{ then } \exists n, V.\ \begin{cases} M \downarrow^n V \\ \widehat{n} \leqslant E_c \\ V \precsim_A^{\mathrm{val}} E_p \end{cases} \\
\underline{n} \precsim_{\mathsf{nat}}^{\mathrm{val}} E & \Longrightarrow & \underline{n} \leqslant E \\
\langle V_1, V_2 \rangle \precsim_{A_1 \times A_2}^{\mathrm{val}} E & \Longrightarrow & \begin{cases} V_1 \precsim_{A_1}^{\mathrm{val}} \pi_1(E) \\ V_2 \precsim_{A_2}^{\mathrm{val}} \pi_2(E) \end{cases} \\
\lambda x.M \precsim_{A \to B}^{\mathrm{val}} E & \Longrightarrow & \forall (N \precsim_A^{\mathrm{val}} X).\ M[N/x] \precsim_B E\,X \\
\mathsf{rec}\, f(x) = P \precsim_{A \to B}^{\mathrm{val}} E & \Longrightarrow & \forall (N \precsim_A^{\mathrm{val}} X).\ P[\mathsf{rec}\, f(x) = P/f, N/x] \precsim_B E\,X
\end{array}$$

and

$$V \precsim_A^{\mathrm{val}} \|V\|_p \text{ for any CBV PCF value } \cdot \vdash V : A$$

$$M \precsim_A \|M\| \text{ for any CBV PCF term } \cdot \vdash M : A$$

Extraction for CBN

- Every PCF type is translated to a **cost algebra**, viz. a **type** along with a **cost action** on that type.

$$c : \mathbb{C}, x : ||A|| \vdash \alpha_A(c, x) : ||A||$$

$$||\mathsf{nat}|| \stackrel{\text{def}}{=} \mathbb{C} \times \mathsf{nat}$$
$$||A_1 \times A_2|| \stackrel{\text{def}}{=} ||A_1|| \times ||A_2||$$
$$||A \to B|| \stackrel{\text{def}}{=} ||A|| \to ||B||$$

- The algebras are defined inductively on types:

$$\begin{array}{rll}
c : \mathbb{C}, x : \mathbb{C} \times \mathsf{nat} \vdash \alpha_{\mathsf{nat}}(c, x) & \stackrel{\text{def}}{=} \langle c \boxplus \pi_1(x), \pi_2(x) \rangle & : \mathbb{C} \times \mathsf{nat} \\
c : \mathbb{C}, p : ||A_1|| \times ||A_2|| \vdash \alpha_{A_1 \times A_2}(c, p) & \stackrel{\text{def}}{=} \langle \alpha_{A_1}(c, \pi_1(p)), \alpha_{A_2}(c, \pi_2(p)) \rangle & : ||A_1|| \times ||A_2|| \\
c : \mathbb{C}, f : ||A|| \to ||B|| \vdash \alpha_{A \to B}(c, f) & \stackrel{\text{def}}{=} \lambda a.\ \alpha_B\,(c, f(a)) & : ||A|| \to ||B||
\end{array}$$

- Writing $L +_A E \stackrel{\text{def}}{=} \alpha_A(L, E) : ||A||$ we have

$$||x|| \stackrel{\text{def}}{=} x$$
$$||M\ \mathbf{op}\ N|| \stackrel{\text{def}}{=} \text{(same as in CBV)}$$
$$||\lambda x.\ M|| \stackrel{\text{def}}{=} \lambda x.\ \mathbf{1} +_B ||M||$$
$$||M\ N|| \stackrel{\text{def}}{=} ||M||\ ||N||$$
$$||\mathsf{fix}\ x.\ M|| \stackrel{\text{def}}{=} \mathsf{fix}\ x.\ \mathbf{1} +_A ||M||$$

Given complexities for the inputs

this recurrence

gives the complexity of M

Theorem: $\Gamma \vdash M : A \Longrightarrow ||\Gamma|| \vdash ||M|| : ||A||$

How to state correctness for CBN

- Because of function types, we must once more use a logical relation.

- **Theorem**: there exists a relation

$$(M : \mathcal{T}_A^{\mathsf{PCF,closed}}) \precsim (E : \mathcal{T}_{||A||}^{\mathsf{PCF}c\mathsf{,closed}})$$

("M is bounded above by recurrence E"), such that

$$M \precsim_{\mathsf{nat}} E \implies \text{if } E_c{\downarrow} \text{ then } \exists n, V. \begin{cases} M \downarrow^n V \\ \widehat{n} \leqslant E_c : \mathbb{C} \\ V \leqslant E_p : \mathsf{nat} \end{cases}$$

$$M \precsim_{A_1 \times A_2} E \implies \begin{cases} \pi_1(M) \precsim_{A_1} \pi_1(E) \\ \pi_2(M) \precsim_{A_2} \pi_2(E) \end{cases}$$

$$M \precsim_{A \to B} E \implies \forall (N \precsim_A X).\ M\, N \precsim_B E\, X$$

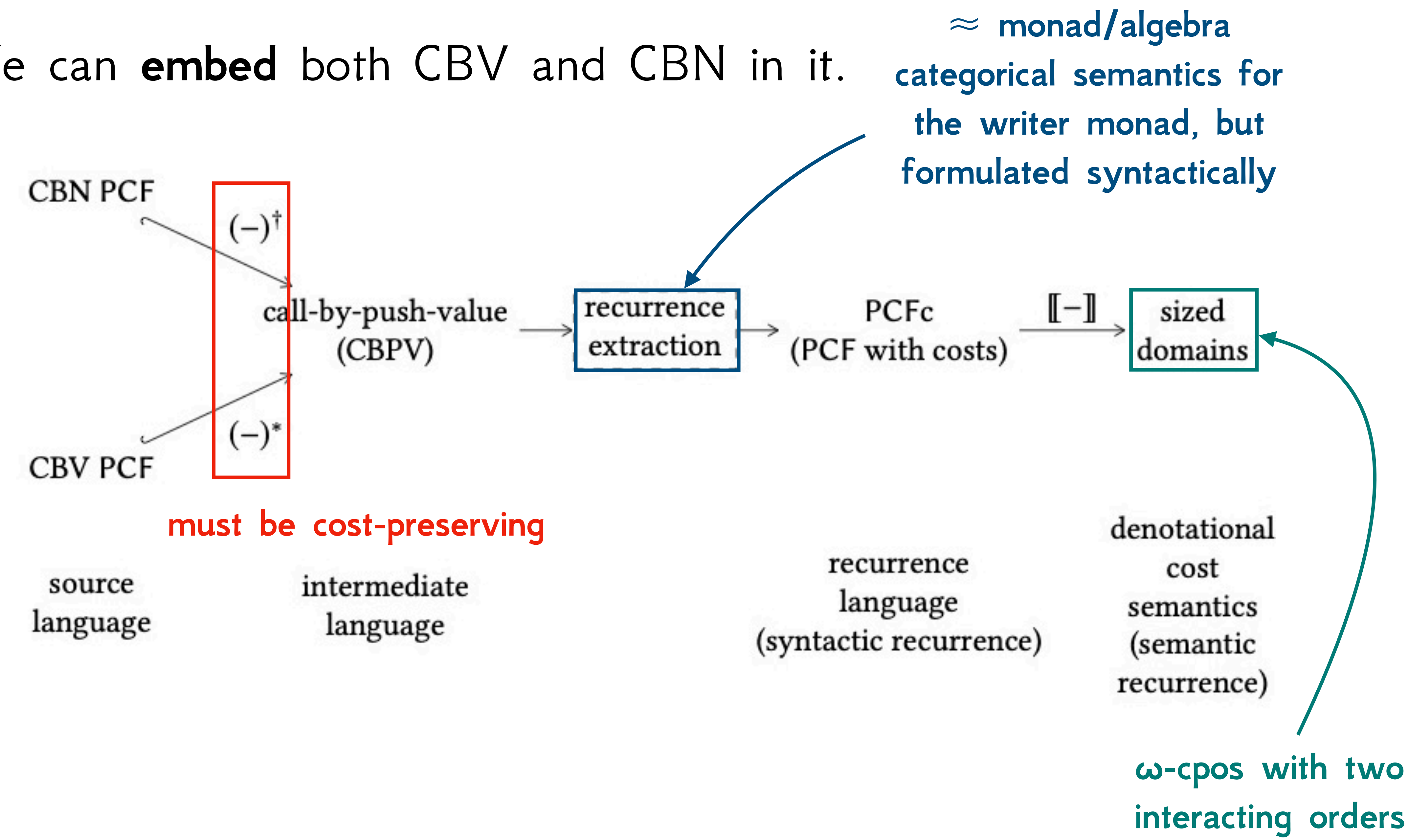
and $M \precsim_A ||M||$ for any CBN PCF term $\cdot \vdash M : A$

Proving any of this is nontrivial.

- A number of difficult issues arise:
  - Translating **CBV to CBN** but **without continuations** — how is it even possible?
  - Handling **nontermination** is nontrivial
  - Obtaining the **logical relation**: defining it as before doesn't work. E.g. in CBV the fixpoint recurs, so IH does not apply.
- How can one even come up with the extraction in the first place?

Two for the price of one: Call-by-Push-Value

- CBPV features **modalities** that control the evaluation of terms.
- We can **embed** both CBV and CBN in it.



Call-by-Push-Value (CBPV)

"A value is, a computation does" — Paul Blain Levy

- CBPV separates **value types** and **computation types.**

$$A ::= \mathsf{nat} \mid \underbrace{A_1 \times A_2}_{\textbf{positive product}} \mid \underbrace{U\underline{B}}_{\textbf{thunks}} \qquad \underline{B} ::= \underbrace{FA}_{\textbf{returners}} \mid \underbrace{\underline{B}_1 \,\&\, \underline{B}_2}_{\textbf{negative products}} \mid A \to \underline{B}$$

- Two sorts of judgments. Only values in contexts!

$$\Gamma \vdash_{\mathsf{v}} V : A \qquad \Gamma \vdash_{\mathsf{c}} M : \underline{B}$$

- Terms of value types are values. For example, we do not have projections at positive product types.

$$\frac{\Gamma \vdash_{\mathsf{v}} V_1 : A_1 \quad \Gamma \vdash_{\mathsf{v}} V_2 : A_2}{\Gamma \vdash_{\mathsf{v}} (V_1, V_2) : A_1 \times A_2} \qquad \frac{\Gamma \vdash_{\mathsf{v}} V : A_1 \times A_2 \quad \Gamma, x_1 : A_1, x_2 : A_2 \vdash_{\mathsf{c}} N : \underline{B}}{\Gamma \vdash_{\mathsf{c}} \mathsf{split}(V; (x_1, x_2).\ N) : \underline{B}}$$

- Terms of computation type can be effectful. We have two effects, **recursion** and **charging a unit cost**.

$$\frac{\Gamma, x : U\underline{B} \vdash_{\mathsf{c}} M : \underline{B}}{\Gamma \vdash_{\mathsf{c}} \mathsf{fix}\ x.\ M : \underline{B}} \qquad \frac{\Gamma \vdash_{\mathsf{c}} M : \underline{B}}{\Gamma \vdash_{\mathsf{c}} \mathsf{charge}.\ M : \underline{B}}$$

- Terms of type FA are **returners**: they perform some effects, and then return a value. Like monads, this value can be **sequentially threaded** into another computation.

$$\frac{\Gamma \vdash_{\mathsf{v}} V : A}{\Gamma \vdash_{\mathsf{c}} \mathsf{return}\ V : FA} \qquad \frac{\Gamma \vdash_{\mathsf{c}} M : FA \quad \Gamma, x : A \vdash_{\mathsf{c}} N : \underline{B}}{\Gamma \vdash_{\mathsf{c}} \mathsf{bind}\ x \Leftarrow M\ \mathsf{in}\ N : \underline{B}}$$

- Computations are **reified** as values: they form **thunks**.

$$\frac{\Gamma \vdash_{\mathsf{c}} M : \underline{B}}{\Gamma \vdash_{\mathsf{v}} \mathsf{thunk}\ M : U\underline{B}} \qquad \frac{\Gamma \vdash_{\mathsf{v}} V : U\underline{B}}{\Gamma \vdash_{\mathsf{c}} \mathsf{force}\ V : \underline{B}}$$

Call-by-Push-Value (CBPV)

"A value is, a computation does" — Paul Blain Levy

$$A ::= \mathsf{nat} \mid \underbrace{A_1 \times A_2}_{\text{positive product}} \mid \underbrace{U\underline{B}}_{\text{thunks}} \qquad \underline{B} ::= \underbrace{FA}_{\text{returners}} \mid \underbrace{\underline{B}_1 \,\&\, \underline{B}_2}_{\text{negative products}} \mid A \to \underline{B}$$

$$M \Downarrow^n T$$

$$\frac{M \Downarrow^m \mathsf{return}\ V \qquad N[V/x] \Downarrow^n T}{\mathsf{bind}\ x \Leftarrow M\ \mathsf{in}\ N \Downarrow^{m+n} T}$$

$$\frac{M \Downarrow^n T}{\mathsf{charge}.\ M \Downarrow^{n+1} T}$$

$$\frac{M[\mathsf{thunk}\ (\mathsf{fix}\ x.\ M)/x] \Downarrow^n T}{\mathsf{fix}\ x.\ M \Downarrow^n T}$$

$$\Gamma \vdash_{\mathsf{v}} V : A \qquad \Gamma \vdash_{\mathsf{c}} M : \underline{B}$$

$$\frac{\Gamma \vdash_{\mathsf{v}} V_1 : A_1 \quad \Gamma \vdash_{\mathsf{v}} V_2 : A_2}{\Gamma \vdash_{\mathsf{v}} (V_1, V_2) : A_1 \times A_2} \qquad \frac{\Gamma \vdash_{\mathsf{v}} V : A_1 \times A_2 \quad \Gamma, x_1 : A_1, x_2 : A_2 \vdash_{\mathsf{c}} N : \underline{B}}{\Gamma \vdash_{\mathsf{c}} \mathsf{split}(V; (x_1, x_2).\ N) : \underline{B}}$$

$$\frac{\Gamma, x : U\underline{B} \vdash_{\mathsf{c}} M : \underline{B}}{\Gamma \vdash_{\mathsf{c}} \mathsf{fix}\ x.\ M : \underline{B}} \qquad \frac{\Gamma \vdash_{\mathsf{c}} M : \underline{B}}{\Gamma \vdash_{\mathsf{c}} \mathsf{charge}.\ M : \underline{B}}$$

$$\frac{\Gamma \vdash_{\mathsf{v}} V : A}{\Gamma \vdash_{\mathsf{c}} \mathsf{return}\ V : FA} \qquad \frac{\Gamma \vdash_{\mathsf{c}} M : FA \quad \Gamma, x : A \vdash_{\mathsf{c}} N : \underline{B}}{\Gamma \vdash_{\mathsf{c}} \mathsf{bind}\ x \Leftarrow M\ \mathsf{in}\ N : \underline{B}}$$

$$\frac{\Gamma \vdash_{\mathsf{c}} M : \underline{B}}{\Gamma \vdash_{\mathsf{v}} \mathsf{thunk}\ M : U\underline{B}} \qquad \frac{\Gamma \vdash_{\mathsf{v}} V : U\underline{B}}{\Gamma \vdash_{\mathsf{c}} \mathsf{force}\ V : \underline{B}}$$

Recurrence extraction for CBPV

Value types => types of **potentials:** they only have use-cost (future, indirect cost).
Computation types => **cost algebras** — they may be evaluated now (current, direct cost).

$$\langle\!\langle \mathsf{nat} \rangle\!\rangle \stackrel{\text{def}}{=} \mathsf{nat} \qquad ||FA|| \stackrel{\text{def}}{=} (\mathbb{C} \times \langle\!\langle A \rangle\!\rangle, \alpha_{FA})$$

$$\langle\!\langle A_1 \times A_2 \rangle\!\rangle \stackrel{\text{def}}{=} \langle\!\langle A_1 \rangle\!\rangle \times \langle\!\langle A_2 \rangle\!\rangle \qquad ||A \to \underline{B}|| \stackrel{\text{def}}{=} (\langle\!\langle A \rangle\!\rangle \to ||\underline{B}||, \alpha_{A \to \underline{B}})$$

$$\langle\!\langle U\underline{B} \rangle\!\rangle \stackrel{\text{def}}{=} ||\underline{B}|| \qquad ||\underline{B}_1 \,\&\, \underline{B}_2|| \stackrel{\text{def}}{=} (||\underline{B}_1|| \times ||\underline{B}_2||, \alpha_{\underline{B}_1 \& \underline{B}_2})$$

$$c : \mathbb{C}, x : \mathbb{C} \times \langle\!\langle A \rangle\!\rangle \vdash \alpha_{FA}(c, x) \stackrel{\text{def}}{=} \langle c \boxplus \pi_1(x), \pi_2(x) \rangle \quad : \mathbb{C} \times \langle\!\langle A \rangle\!\rangle$$

$$c : \mathbb{C}, f : \langle\!\langle A \rangle\!\rangle \to ||\underline{B}|| \vdash \alpha_{A \to \underline{B}}(c, f) \stackrel{\text{def}}{=} \lambda a.\ \alpha_{\underline{B}}(c, f(a)) \quad : \langle\!\langle A \rangle\!\rangle \to ||\underline{B}||$$

$$c : \mathbb{C}, p : ||\underline{B}_1|| \times ||\underline{B}_2|| \vdash \alpha_{\underline{B}_1 \& \underline{B}_2}(c, p) \stackrel{\text{def}}{=} \left\langle \alpha_{\underline{B}_1}(c, \pi_1(p)), \alpha_{\underline{B}_2}(c, \pi_2(p)) \right\rangle \quad : ||\underline{B}_1|| \times ||\underline{B}_2||$$

$$||\mathsf{return}\ V|| \stackrel{\text{def}}{=} \langle \mathbf{0}, \langle\!\langle V \rangle\!\rangle \rangle$$

$$||\mathsf{bind}\ x \Leftarrow M\ \mathsf{in}\ N|| \stackrel{\text{def}}{=} ||M||_c +_{\underline{B}} ||N|| \left[||M||_p / x \right]$$

$$||\mathsf{charge}.\ M|| \stackrel{\text{def}}{=} \mathbf{1} +_{\underline{B}} ||M||$$

$$||\mathsf{fix}\ x.\ M|| \stackrel{\text{def}}{=} \mathsf{fix}\ x.\ ||M||$$

Theorem:
$$\Gamma \vdash_{\mathsf{v}} V : A \Longrightarrow \langle\!\langle \Gamma \rangle\!\rangle \vdash \langle\!\langle V \rangle\!\rangle : \langle\!\langle A \rangle\!\rangle$$
$$\Gamma \vdash_{\mathsf{c}} M : \underline{B} \Longrightarrow \langle\!\langle \Gamma \rangle\!\rangle \vdash ||M|| : ||\underline{B}||$$

How to state correctness for CBPV

The following logical relation does the trick. It's a mix-and-match of the CBV and CBN styles.

$$\tilde{n} \lesssim^{\mathrm{val}}_{\mathsf{nat}} E \overset{\text{def}}{\equiv} \underline{n} \leqslant E$$

$$(V_1, V_2) \lesssim^{\mathrm{val}}_{A_1 \times A_2} E \overset{\text{def}}{\equiv} \begin{cases} V_1 \lesssim^{\mathrm{val}}_{A_1} \pi_1(E) \\ V_2 \lesssim^{\mathrm{val}}_{A_2} \pi_2(E) \end{cases}$$

$$\mathsf{thunk}\ M \lesssim^{\mathrm{val}}_{U\underline{B}} E \overset{\text{def}}{\equiv} M \lesssim^{\mathsf{c}}_{\underline{B}} E$$

$$M \lesssim^{\mathsf{c}}_{FA} E \overset{\text{def}}{\equiv} E_c\!\downarrow \implies \exists n, V.\ \begin{cases} M \Downarrow^n \mathsf{return}\ V \\ \widehat{n} \leqslant E_c \\ V \lesssim^{\mathrm{val}}_{A} E_p \end{cases}$$

$$M \lesssim^{\mathsf{c}}_{A \to \underline{B}} E \overset{\text{def}}{\equiv} \forall (N \lesssim^{\mathrm{val}}_{A} X).\ M\ N \lesssim^{\mathsf{c}}_{\underline{B}} E\ X$$

$$M \lesssim^{\mathsf{c}}_{\underline{B}_1 \& \underline{B}_2} E \overset{\text{def}}{\equiv} \begin{cases} \pi_1(M) \lesssim^{\mathsf{c}}_{\underline{B}_1} \pi_1(E) \\ \pi_2(M) \lesssim^{\mathsf{c}}_{\underline{B}_2} \pi_2(E) \end{cases}$$

Theorem (Bounding Theorem):

$$\cdot \vdash V : A \implies V \lesssim^{\mathrm{val}}_{A} \langle\!\langle V \rangle\!\rangle$$

$$\cdot \vdash M : \underline{B} \implies M \lesssim^{\mathsf{c}}_{\underline{B}} ||M||$$

Proving the original theorems for CBN and CBV

- Use the **Levy embedding** of CBN and CBV into CBPV.
- Sprinkle occurrences of "**charge**" wherever cost should be incurred.
- Prove that the embedding is cost-preserving (a bisimulation-like result).

Denotational semantics of PCFc

- A **sized domain** consists of a set D and
  - An **information order**, i.e. a pointed ω-cpo $(D, \sqsubseteq, \bot)$
  - A **size preorder** $(D, \leqslant, \mathbf{0}, \vee)$

least element

chosen least upper bounds

Why preorder? E.g. the lists [1, 2, 3] and [4, 5, 6] are "size-equal" but not

such that

  - The chosen lubs are continuous wrt the information order.
  - A better-defined bound is a smaller bound: $x \sqsubseteq y \Longrightarrow y \leqslant x$
  - A recursively defined upper bound is an upper bound:
    for a chain $(x_i)_{i\in\omega}$ we have $(\forall i.\ z \leqslant x_i) \Longrightarrow z \leqslant \sqcup_{i\in\omega} x_i$

Conclusions

- We formulated recurrence extraction for both CBN and CBV with recursion **in a uniform way**, and proved it correct.
- The structure of CBPV illuminates the basic concepts of **potential** and **complexity** that occurred in previous work:
  - **values** have **potential** (use-cost)
  - **computations** have **complexity** (direct cost)
- This is the beginning of a theory of **higher-order recurrences**.
- The analysis can easily be extended to CBV inductive types.
- Recursive types much harder — but we have some ideas!