Uniwersytet
Wrocławski

A Simple and Efficient Implementation of Strong Call by Need by an Abstract Machine

Małgorzata Biernacka, Witold Charatonik, Tomasz Drab

University of Wrocław

27th ACM SIGPLAN International Conference on Functional Programming (ICFP) 2022

Why call by need?

Why call by need?

call by name

call by value

Why call by need?

call by name

don’t evaluate unneeded arguments

call by value

Why call by need?

call by name

don’t evaluate unneeded arguments

call by value

evaluate arguments at most once

Why call by need?

call by name

don’t evaluate unneeded arguments

call by value

evaluate arguments at most once

call by need

Why <u>strong</u> reduction?

Why strong reduction?

$2 = \lambda f.\ \lambda x.\ f\ (f\ x)$

Why strong reduction?

$2 = \lambda f.\ \lambda x.\ f\ (f\ x)$

$\times = \lambda n.\ \lambda m.\ \lambda f.\ m\ (n\ f)$

Why <u>strong</u> reduction?

$2 = \lambda f.\ \lambda x.\ f\ (f\ x)$

$\times = \lambda n.\ \lambda m.\ \lambda f.\ m\ (n\ f)$

$\times\ 2\ 2$

Why strong reduction?

$2 = \lambda f.\ \lambda x.\ f\ (f\ x)$

$\times = \lambda n.\ \lambda m.\ \lambda f.\ m\ (n\ f)$

$\times\ 2\ 2 \overset{\text{weak}}{\to\to}$

Why strong reduction?

$2 = \lambda f.\ \lambda x.\ f\ (f\ x)$

$\times = \lambda n.\ \lambda m.\ \lambda f.\ m\ (n\ f)$

$\times\ 2\ 2 \xrightarrow{\text{weak}}\!\!\!\rightarrow \lambda f.\ 2\ (2\ f)$

Why <u>strong</u> reduction?

$2 = \lambda f.\ \lambda x.\ f\ (f\ x)$

$\times = \lambda n.\ \lambda m.\ \lambda f.\ m\ (n\ f)$

$\times\ 2\ 2 \overset{\text{weak}}{\longrightarrow\!\!\!\!\longrightarrow} \lambda f.\ 2\ (2\ f) \overset{\text{weak}}{\not\longrightarrow}$

Why strong reduction?

$2 = \lambda f.\ \lambda x.\ f\ (f\ x)$

$\times = \lambda n.\ \lambda m.\ \lambda f.\ m\ (n\ f)$

$\times\ 2\ 2 \xrightarrow{\text{weak}}\!\!\!\!\rightarrow \lambda f.\ 2\ (2\ f) \overset{\text{weak}}{\not\rightarrow}$

$\times\ 2\ 2 \xrightarrow{\text{strong}}\!\!\!\!\rightarrow \lambda f.\ 2\ (2\ f)$

Why <u>strong</u> reduction?

$2 = \lambda f.\ \lambda x.\ f\ (f\ x)$

$\times = \lambda n.\ \lambda m.\ \lambda f.\ m\ (n\ f)$

$\times\ 2\ 2 \xrightarrow{\text{weak}}\!\!\rightarrow \lambda f.\ 2\ (2\ f) \overset{\text{weak}}{\not\rightarrow}$

$\times\ 2\ 2 \xrightarrow{\text{strong}}\!\!\rightarrow \lambda f.\ 2\ (2\ f) \xrightarrow{\text{strong}} \ldots \rightarrow \lambda f.\ \lambda x.\ f\ (f\ (f\ (f\ x)))$

Why strong reduction?

$2 = \lambda f.\ \lambda x.\ f\ (f\ x)$

$\times = \lambda n.\ \lambda m.\ \lambda f.\ m\ (n\ f)$

$\times\ 2\ 2 \xrightarrow{\text{weak}}\!\!\!\!\rightarrow \lambda f.\ 2\ (2\ f) \overset{\text{weak}}{\not\rightarrow}$

$\times\ 2\ 2 \xrightarrow{\text{strong}}\!\!\!\!\rightarrow \lambda f.\ 2\ (2\ f) \xrightarrow{\text{strong}} \ldots \rightarrow \lambda f.\ \lambda x.\ f\ (f\ (f\ (f\ x))) = 4$

Why strong reduction?

$2 = \lambda f.\ \lambda x.\ f\ (f\ x)$

$\times = \lambda n.\ \lambda m.\ \lambda f.\ m\ (n\ f)$

$\times\ 2\ 2 \xrightarrow{\text{weak}}\!\!\!\rightarrow \lambda f.\ 2\ (2\ f) \overset{\text{weak}}{\not\rightarrow}$

$\times\ 2\ 2 \xrightarrow{\text{strong}}\!\!\!\rightarrow \lambda f.\ 2\ (2\ f) \rightarrow \ldots \rightarrow \lambda f.\ \lambda x.\ f\ (f\ (f\ (f\ x))) = 4$

strong reduction applied in type checking in proof assistants

In what sense <u>efficient</u>?

Time complexity as a function of the two parameters:

- size of the initial term
- number of β-reductions

[Accattoli, Barenbaum, Mazza, ICFP 2014]

In what sense efficient?

Time complexity as a function of the two parameters:

- size of the initial term
- number of β-reductions

[Accattoli, Barenbaum, Mazza, ICFP 2014]

Implementation is *efficient* if it is linear in both parameters.

cf. [Accattoli, Guerrieri, FSEN 2017]

Abstract Machines
for Strong call by Need

Abstract Machines for Strong call by Need

- [Ager, Biernacki, Danvy, Midtgaard, 2003]
  implement
  full normalization in call by need

Abstract Machines for Strong call by Need

- [Ager, Biernacki, Danvy, Midtgaard, 2003]
  implement
  full normalization in call by need

- [Biernacka, Charatonik, FSCD 2019]
  implement
  [Balabonski, Barenbaum, Bonelli, Kesner, ICFP 2017]

Abstract Machines for Strong call by Need

- [Ager, Biernacki, Danvy, Midtgaard, 2003]
  implement
  full normalization in call by need

- [Biernacka, Charatonik, FSCD 2019]
  implement
  [Balabonski, Barenbaum, Bonelli, Kesner, ICFP 2017]

both machines suffer from exponential overhead

Further Plan of the Presentation

- derivation of the efficient machine

Further Plan of the Presentation

- derivation of the efficient machine (quasi-*efficient*)

Further Plan of the Presentation

- derivation of the efficient machine (quasi-*efficient*)
- correctness and complexity

Further Plan of the Presentation

- derivation of the efficient machine (quasi-*efficient*)
- correctness and complexity
- remark on empirical method

Further Plan of the Presentation

- derivation of the efficient machine (quasi-*efficient*)
- correctness and complexity
- remark on empirical method
- remark on implicit sharing

Functional Correspondence

Definitional Interpreter

Functional Correspondence

CPS translation

Definitional Interpreter

Functional Correspondence

CPS translation

defunctionalization

Definitional Interpreter

Functional Correspondence

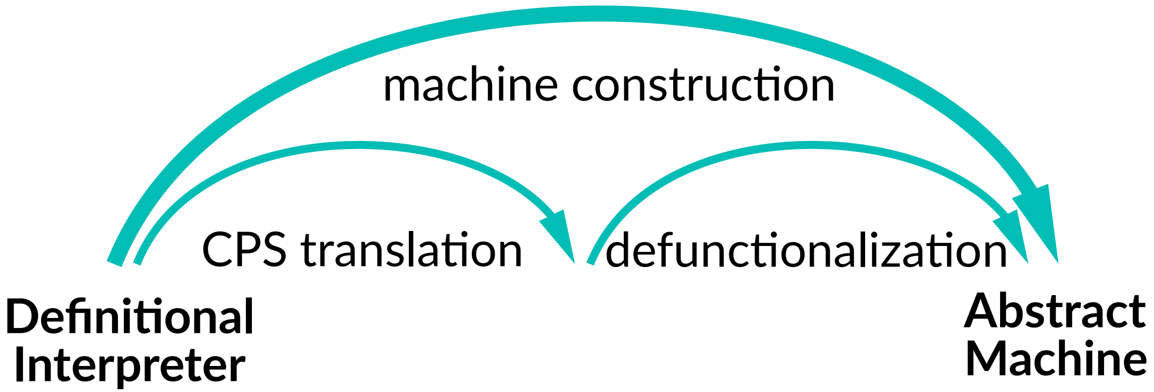
CPS translation

defunctionalization

Definitional Interpreter

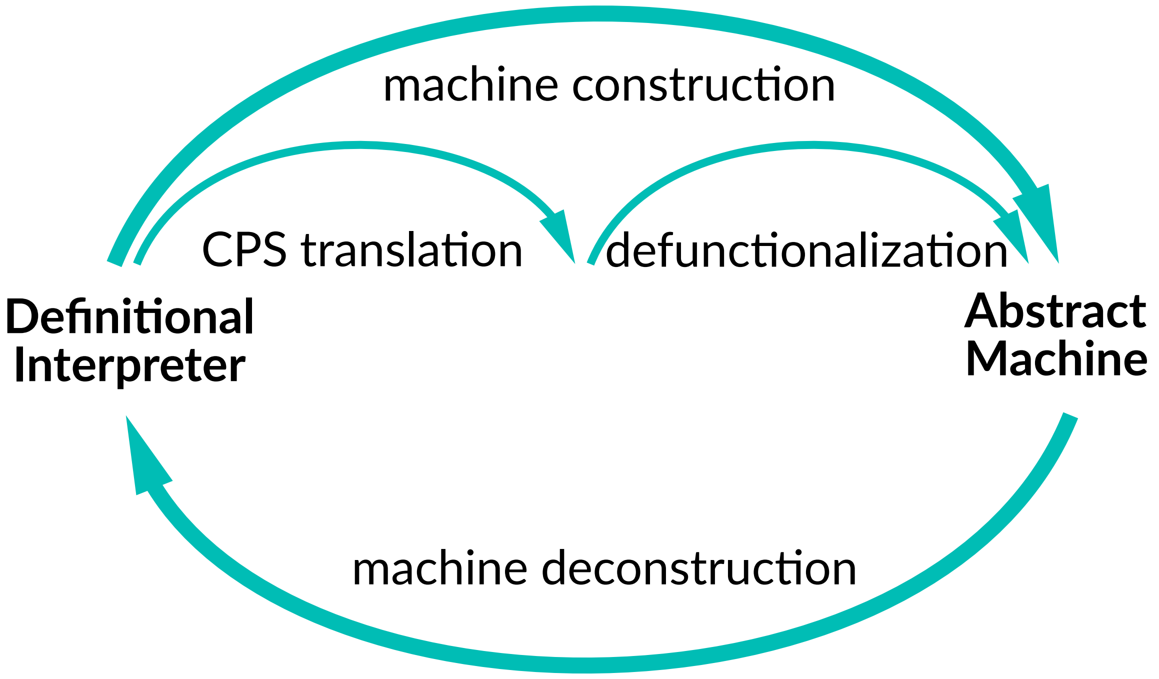
[John C. Reynolds, 1972]

Functional Correspondence



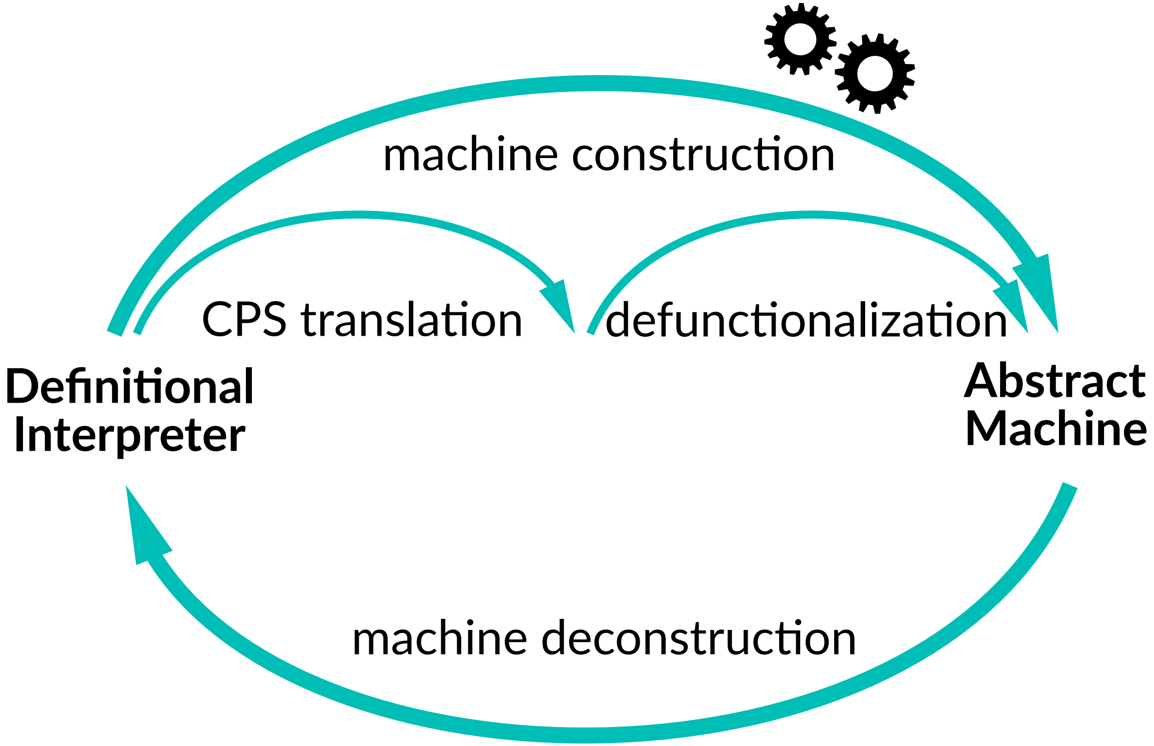
[John C. Reynolds, 1972]
[Ager, Danvy, Biernacki, Midtgaard, PPDP 2003]

Functional Correspondence



[John C. Reynolds, 1972]
[Ager, Danvy, Biernacki, Midtgaard, PPDP 2003]

Functional Correspondence



[John C. Reynolds, 1972]
[Ager, Danvy, Biernacki, Midtgaard, PPDP 2003]
[Buszka, Biernacki, PPDP 2021]

Derivation

[Crégut, 1990]

KN

abstract machine

Derivation

[Crégut, 1990] [García-Pérez, Nogueira, JFP 2019]

KN
abstract machine

implements $\dashrightarrow$

normal order
reduction strategy

Derivation

[Munk, 2008] [Crégut, 1990] [García-Pérez, Nogueira, JFP 2019]

<functional code> $\xleftarrow{\text{deconstruction}}$ KN $\xrightarrow{\text{implements}}$ normal order

abstract machine

reduction strategy

Derivation

[Munk, 2008] [Crégut, 1990] [García-Pérez, Nogueira, JFP 2019]

<functional code>

deconstruction

KN

abstract machine

implements

normal order

reduction strategy

introduction of
memoizations and other changes

<functional code>

Derivation

[Munk, 2008] [Crégut, 1990] [García-Pérez, Nogueira, JFP 2019]

<functional code>

deconstruction

KN

abstract machine

implements

normal order

reduction strategy

introduction of
memoizations and other changes

<functional code>

construction

RKNL

abstract machine

RKNL

$$\langle (t_1\ t_2, e), s, \sigma \rangle_\triangledown \longrightarrow \langle (t_1, e), \square\ (t_2, e) :: s, \sigma \rangle_\triangledown \tag{1}$$

$$\langle (\lambda x.t, e), s, \sigma \rangle_\triangledown \longrightarrow \langle \ell := (\lambda x.t, e), s, \sigma * [\ell \mapsto \bot_{\text{✗}}] \rangle_\vartriangle \tag{2}$$

$$\langle (x, e), s, \sigma \rangle_\triangledown \longrightarrow \langle (t, e_2), \ell := \square :: s, \sigma \rangle_\triangledown \text{ where } \ell = e(x),\ \sigma(\ell) = (t, e_2)_{\text{✗}} \tag{3}$$

$$\langle (x, e), s, \sigma \rangle_\triangledown \longrightarrow \langle v, s, \sigma \rangle_\vartriangle \qquad \text{where } \sigma(e(x)) = v_{\checkmark} \vee (v = x \notin e) \tag{4}$$

$$\langle v, \ell := \square :: s, \sigma \rangle_\vartriangle \longrightarrow \langle v, s, \sigma[\ell := v_{\checkmark}] \rangle_\vartriangle \tag{5}$$

$$\langle \ell := (\lambda x.t, e), \square\ (t_2, e_2) :: s, \sigma \rangle_\vartriangle \longrightarrow \langle (t, e[x := \ell_2]), s, \sigma * [\ell_2 \mapsto (t_2, e_2)_{\text{✗}}] \rangle_\triangledown \tag{6}$$

$$\langle \ell := (\lambda x.t, e), \qquad s, \sigma \rangle_\vartriangle \longrightarrow \langle (t, e[x := \ell_2]), \lambda \check{x}.\square :: \ell := \square :: s, \sigma * [\ell_2 \mapsto \check{x}_{\checkmark}] \rangle_\triangledown$$
$$\text{where } \sigma(\ell) = \bot_{\text{✗}} \tag{7}$$

$$\langle \ell := (\lambda x.t, e), \qquad s, \sigma \rangle_\vartriangle \longrightarrow \langle v, s, \sigma \rangle_\vartriangle \qquad \text{where } \sigma(\ell) = v_{\checkmark} \tag{8}$$

$$\langle t, \square\ (t_2, e_2) :: s, \sigma \rangle_\vartriangle \longrightarrow \langle (t_2, e_2), t\ \square :: s, \sigma \rangle_\triangledown \tag{9}$$

$$\langle t_2, t_1\ \square :: s, \sigma \rangle_\vartriangle \longrightarrow \langle t_1\ t_2, s, \sigma \rangle_\vartriangle \tag{10}$$

$$\langle t, \lambda x.\square :: s, \sigma \rangle_\vartriangle \longrightarrow \langle \lambda x.t, s, \sigma \rangle_\vartriangle \tag{11}$$

Elaborate Example Execution

$$0: \quad \langle((\lambda x.c\,x\,x)\,(A\,\Omega),[])\rangle_{\triangledown} \| [] \quad \xrightarrow{(1)}$$

Elaborate Example Execution

$$0: \quad \langle ((\lambda x.c\,x\,x)\,(A\,\Omega),[])\rangle_{\triangledown} \,|\, [] \qquad \xrightarrow{(1)}$$

$$1: \quad \langle (\lambda x.c\,x\,x,[])\rangle_{\triangledown}\ (A\,\Omega,[]) \,|\, [] \qquad \xrightarrow{(2)}$$

Elaborate Example Execution

$$0: \quad \langle((\lambda x.c\,x\,x)\,(A\,\Omega),[])\rangle_{\triangledown} \,\|[] \quad \xrightarrow{(1)}$$

$$1: \quad \langle(\lambda x.c\,x\,x,[])\rangle_{\triangledown}\,(A\,\Omega,[])\|[] \quad \xrightarrow{(2)}$$

$$2: \quad \langle \mathbb{a} := (\lambda x.c\,x\,x,[])\rangle_{\vartriangle}\,(A\,\Omega,[])\|[\mathbb{a} \mapsto \bot_{\text{✗}}] \quad \xrightarrow{(6)}$$

Elaborate Example Execution

$$0 : \quad \langle((\lambda x.c\,x\,x)\,(A\,\Omega),[])\rangle_{\triangledown} \,|\, [] \qquad \xrightarrow{(1)}$$

$$1 : \quad \langle(\lambda x.c\,x\,x,[])\rangle_{\triangledown}\;(A\,\Omega,[]) \,|\, [] \qquad \xrightarrow{(2)}$$

$$2 : \quad \langle \mathbb{a} := (\lambda x.c\,x\,x,[])\rangle_{\vartriangle}\;(A\,\Omega,[]) \,|\, [\mathbb{a} \mapsto \bot_{\times}] \qquad \xrightarrow{(6)}$$

$$3 : \quad \langle(c\,x\,x,[x \mapsto \mathbb{x}])\rangle_{\triangledown} \,|\, [\mathbb{x} \mapsto (A\,\Omega,[])_{\times}] \qquad \xrightarrow{(1)}$$

Elaborate Example Execution

$0: \quad \langle((\lambda x.c\,x\,x)\,(A\,\Omega),[])\rangle_{\triangledown} | [] \quad \xrightarrow{(1)}$

$1: \quad \langle(\lambda x.c\,x\,x,[])\rangle_{\triangledown}\ (A\,\Omega,[]) | [] \quad \xrightarrow{(2)}$

$2: \quad \langle \mathbb{a} := (\lambda x.c\,x\,x,[])\rangle_{\vartriangle}\ (A\,\Omega,[]) | [\mathbb{a} \mapsto \bot_{\text{✗}}] \quad \xrightarrow{(6)}$

$3: \quad \langle(c\,x\,x,[x \mapsto \mathbb{x}])\rangle_{\triangledown} | [\mathbb{x} \mapsto (A\,\Omega,[])_{\text{✗}}] \quad \xrightarrow{(1)}$

$4: \quad \langle(c\,x,[x \mapsto \mathbb{x}])\rangle_{\triangledown}\ x^{\mathbb{x}} | [\mathbb{x} \mapsto (A\,\Omega,[])_{\text{✗}}] \quad \xrightarrow{(1)}$

Elaborate Example Execution

$$0: \quad \langle ((\lambda x.c\,x\,x)\,(A\,\Omega),[]) \rangle_{\triangledown} \,|\, [] \quad \xrightarrow{(1)}$$

$$1: \quad \langle (\lambda x.c\,x\,x,[]) \rangle_{\triangledown}\, (A\,\Omega,[]) \,|\, [] \quad \xrightarrow{(2)}$$

$$2: \quad \langle \mathbb{a} := (\lambda x.c\,x\,x,[]) \rangle_{\vartriangle}\, (A\,\Omega,[]) \,|\, [\mathbb{a} \mapsto \bot_{\text{✗}}] \quad \xrightarrow{(6)}$$

$$3: \quad \langle (c\,x\,x,[x \mapsto \mathbb{x}]) \rangle_{\triangledown} \,|\, [\mathbb{x} \mapsto (A\,\Omega,[])_{\text{✗}}] \quad \xrightarrow{(1)}$$

$$4: \quad \langle (c\,x,[x \mapsto \mathbb{x}]) \rangle_{\triangledown}\, x^{\mathbb{x}} \,|\, [\mathbb{x} \mapsto (A\,\Omega,[])_{\text{✗}}] \quad \xrightarrow{(1)}$$

$$5: \quad \langle (c,[x \mapsto \mathbb{x}]) \rangle_{\triangledown}\, x^{\mathbb{x}}\, x^{\mathbb{x}} \,|\, [\mathbb{x} \mapsto (A\,\Omega,[])_{\text{✗}}] \quad \xrightarrow{(4)}$$

Elaborate Example Execution

$$0:\quad \langle((\lambda x.c\,x\,x)\,(A\,\Omega),[])\rangle_{\triangledown}\,|\,[] \quad \xrightarrow{(1)}$$

$$1:\quad \langle(\lambda x.c\,x\,x,[])\rangle_{\triangledown}\,(A\,\Omega,[])\,|\,[] \quad \xrightarrow{(2)}$$

$$2:\quad \langle \mathbb{a} := (\lambda x.c\,x\,x,[])\rangle_{\vartriangle}\,(A\,\Omega,[])\,|\,[\mathbb{a} \mapsto \bot_{\text{✗}}] \quad \xrightarrow{(6)}$$

$$3:\quad \langle(c\,x\,x,[x \mapsto \mathbb{x}])\rangle_{\triangledown}\,|\,[\mathbb{x} \mapsto (A\,\Omega,[])_{\text{✗}}] \quad \xrightarrow{(1)}$$

$$4:\quad \langle(c\,x,[x \mapsto \mathbb{x}])\rangle_{\triangledown}\,x^{\mathbb{x}}\,|\,[\mathbb{x} \mapsto (A\,\Omega,[])_{\text{✗}}] \quad \xrightarrow{(1)}$$

$$5:\quad \langle(c,[x \mapsto \mathbb{x}])\rangle_{\triangledown}\,x^{\mathbb{x}}\,x^{\mathbb{x}}\,|\,[\mathbb{x} \mapsto (A\,\Omega,[])_{\text{✗}}] \quad \xrightarrow{(4)}$$

$$6:\quad \langle c\rangle_{\vartriangle}\,x^{\mathbb{x}}\,x^{\mathbb{x}}\,|\,[\mathbb{x} \mapsto (A\,\Omega,[])_{\text{✗}}] \quad \xrightarrow{(9)}$$

Elaborate Example Execution

$$0 : \quad \langle ((\lambda x.c\,x\,x)\,(A\,\Omega),[])\rangle_{\triangledown} \,|\, [] \qquad \xrightarrow{(1)}$$

$$1 : \quad \langle (\lambda x.c\,x\,x,[])\rangle_{\triangledown}\ (A\,\Omega,[]) \,|\, [] \qquad \xrightarrow{(2)}$$

$$2 : \quad \langle \mathbb{a} := (\lambda x.c\,x\,x,[])\rangle_{\vartriangle}\ (A\,\Omega,[]) \,|\, [\mathbb{a} \mapsto \bot_{\text{✗}}] \qquad \xrightarrow{(6)}$$

$$3 : \quad \langle (c\,x\,x,[x \mapsto \mathbb{x}])\rangle_{\triangledown} \,|\, [\mathbb{x} \mapsto (A\,\Omega,[])_{\text{✗}}] \qquad \xrightarrow{(1)}$$

$$4 : \quad \langle (c\,x,[x \mapsto \mathbb{x}])\rangle_{\triangledown}\ x^{\mathbb{x}} \,|\, [\mathbb{x} \mapsto (A\,\Omega,[])_{\text{✗}}] \qquad \xrightarrow{(1)}$$

$$5 : \quad \langle (c,[x \mapsto \mathbb{x}])\rangle_{\triangledown}\ x^{\mathbb{x}}\ x^{\mathbb{x}} \,|\, [\mathbb{x} \mapsto (A\,\Omega,[])_{\text{✗}}] \qquad \xrightarrow{(4)}$$

$$6 : \quad \langle c\rangle_{\vartriangle}\ x^{\mathbb{x}}\ x^{\mathbb{x}} \,|\, [\mathbb{x} \mapsto (A\,\Omega,[])_{\text{✗}}] \qquad \xrightarrow{(9)}$$

$$7 : \quad c\ \langle x^{\mathbb{x}}\rangle_{\triangledown}\ x^{\mathbb{x}} \,|\, [\mathbb{x} \mapsto (A\,\Omega,[])_{\text{✗}}] \qquad \xrightarrow{(3)}$$

Elaborate Example Execution

$$0: \quad \langle((\lambda x.c\,x\,x)\,(A\,\Omega),[])\rangle_{\triangledown} \,|\, [] \quad \xrightarrow{(1)}$$

$$1: \quad \langle(\lambda x.c\,x\,x,[])\rangle_{\triangledown}\ (A\,\Omega,[]) \,|\, [] \quad \xrightarrow{(2)}$$

$$2: \quad \langle\mathbb{a} := (\lambda x.c\,x\,x,[])\rangle_{\vartriangle}\ (A\,\Omega,[]) \,|\, [\mathbb{a} \mapsto \bot\,✗] \quad \xrightarrow{(6)}$$

$$3: \quad \langle(c\,x\,x,[x \mapsto \mathbb{x}])\rangle_{\triangledown} \,|\, [\mathbb{x} \mapsto (A\,\Omega,[])\,✗] \quad \xrightarrow{(1)}$$

$$4: \quad \langle(c\,x,[x \mapsto \mathbb{x}])\rangle_{\triangledown}\ x^{\mathbb{x}} \,|\, [\mathbb{x} \mapsto (A\,\Omega,[])\,✗] \quad \xrightarrow{(1)}$$

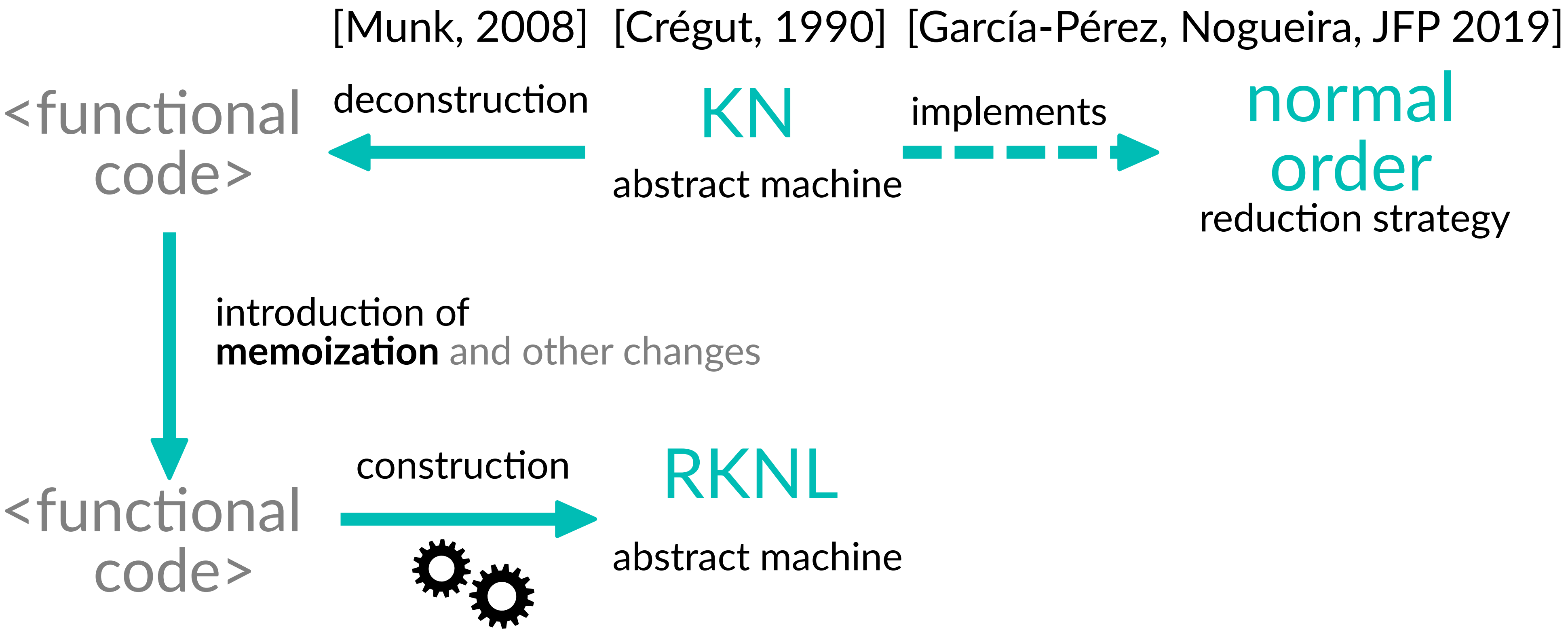
$$5: \quad \langle(c,[x \mapsto \mathbb{x}])\rangle_{\triangledown}\ x^{\mathbb{x}}\ x^{\mathbb{x}} \,|\, [\mathbb{x} \mapsto (A\,\Omega,[])\,✗] \quad \xrightarrow{(4)}$$

$$6: \quad \langle c\rangle_{\vartriangle}\ x^{\mathbb{x}}\ x^{\mathbb{x}} \,|\, [\mathbb{x} \mapsto (A\,\Omega,[])\,✗] \quad \xrightarrow{(9)}$$

$$7: \quad c\ \langle x^{\mathbb{x}}\rangle_{\triangledown}\ x^{\mathbb{x}} \,|\, [\mathbb{x} \mapsto (A\,\Omega,[])\,✗] \quad \xrightarrow{(3)}$$

$$8: \quad c\ \left(\mathbb{x} := \langle(A\,\Omega,[])\rangle_{\triangledown}\right)\ x^{\mathbb{x}} \,|\, [\mathbb{x} \mapsto (A\,\Omega,[])\,✗] \quad \xrightarrow{(1)}$$

Correctness and Complexity



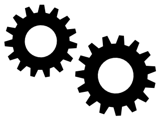
Correctness and Complexity

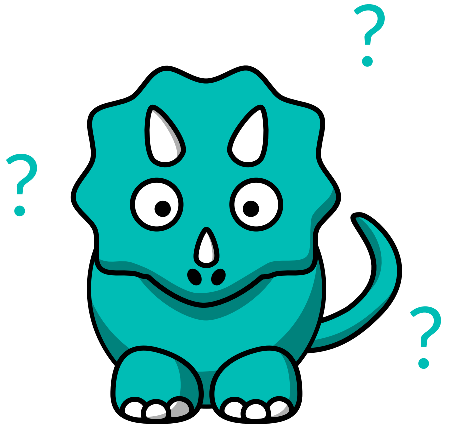
[Munk, 2008] [Crégut, 1990] [García-Pérez, Nogueira, JFP 2019]

<functional code> ← deconstruction — **KN** abstract machine — implements - - → **normal order** reduction strategy

↓ introduction of **memoization** and other changes

<functional code> — construction → **RKNL** abstract machine — implements - - →





Strong call by need

RKNL

abstract machine

Strong call by need

KL

abstract machine for
weak call by **need**

conservatively
extends

RKNL

abstract machine

Strong call by need

normal order

full-reducing strategy
(**strong** call by **name**)

KL

abstract machine for
weak call by **need**

conservatively
extends

RKNL

abstract machine

Strong call by need

normal order

full-reducing strategy
(**strong** call by **name**)

KL

abstract machine for
weak call by **need**

conservatively
extends

RKNL

abstract machine
for **strong** call by **need**

Strong call by need

normal order

full-reducing strategy
(**strong** call by **name**)

KL

abstract machine for
weak call by **need**

implements

conservatively
extends

RKNL

abstract machine
for **strong** call by **need**

Ghost Abstract Machine

$$
\begin{aligned}
\mathit{Applicative\ Stacks} \ni\ & \alpha ::= \ell := \square :: \alpha \mid \square\, c :: \alpha \mid \lceil \square \rceil :: \nu \\
\mathit{Non\text{-}applicative\ Stacks} \ni\ & \nu ::= \ell' := \square :: \nu \mid [\,] \mid \lambda x.\square :: \nu \mid a\,\square :: \alpha \\
\mathit{Confs} \ni\ & k ::= \langle c, \alpha, \sigma, \sigma' \rangle_{\triangledown} \mid \langle v, \alpha, \sigma, \sigma' \rangle_{\vartriangle} \mid \langle n, \nu, \sigma, \sigma' \rangle_{\vartriangle'} \\
\mathit{Transitions} :\ &
\end{aligned}
$$

$$
\begin{aligned}
t &\mapsto \langle (t, [\,]), \lceil \square \rceil :: [\,], [\,] \rangle_{\triangledown} \\
\langle (t_1\, t_2, e), \alpha, \sigma, \sigma' \rangle_{\triangledown} &\to \langle (t_1, e), \square\, (t_2, e) :: \alpha, \sigma, \sigma' \rangle_{\triangledown} && (1) \\
\langle (\lambda x.t, e), \alpha, \sigma, \sigma' \rangle_{\triangledown} &\to \langle \ell' := (\lambda x.t, e), \alpha, \sigma, \sigma' * [\ell' \mapsto \bot_{\text{✗}}] \rangle_{\vartriangle} && (2) \\
\langle (x, e), \alpha, \sigma, \sigma' \rangle_{\triangledown} &\to \langle (t, e_2), \ell := \square :: \alpha, \sigma, \sigma' \rangle_{\triangledown} && (3) \\
&\qquad \text{where } \ell = e(x),\ \ \sigma(\ell) = (t, e_2)_{\text{✗}} \\
\langle (x, e), \alpha, \sigma, \sigma' \rangle_{\triangledown} &\to \langle v, \alpha, \sigma, \sigma' \rangle_{\vartriangle} \qquad \text{where } \sigma(e(x)) = v_{\checkmark} \vee (v = x \notin e) && (4) \\
\langle v, \ell := \square :: \alpha, \sigma, \sigma' \rangle_{\vartriangle} &\to \langle v, \alpha, \sigma[\ell := v_{\checkmark}], \sigma' \rangle_{\vartriangle} && (5) \\
\langle n, \ell' := \square :: \nu, \sigma, \sigma' \rangle_{\vartriangle'} &\to \langle n, \nu, \sigma, \sigma'[\ell' := n_{\checkmark}] \rangle_{\vartriangle'} && (5') \\
\langle \ell' := (\lambda x.t, e), \square\, (t_2, e_2) :: \alpha, \sigma, \sigma' \rangle_{\vartriangle} &\to \langle (t, e[x := \ell_2]), \alpha, \sigma * [\ell_2 \mapsto (t_2, e_2)_{\text{✗}}], \sigma' \rangle_{\triangledown} && (6) \\
\langle \ell' := (\lambda x.t, e), \quad \lceil \square \rceil :: \nu, \sigma, \sigma' \rangle_{\vartriangle} &\to \langle (t, e[x := \ell_2]), \lceil \square \rceil :: \lambda \check{x}.\square :: \ell' := \square :: \nu, \\
&\qquad \sigma * [\ell_2 \mapsto \check{x}_{\checkmark}], \sigma' \rangle_{\triangledown} \qquad \text{where } \sigma'(\ell') = \bot_{\text{✗}} && (7) \\
\langle \ell' := (\lambda x.t, e), \quad \lceil \square \rceil :: \nu, \sigma, \sigma' \rangle_{\vartriangle} &\to \langle n, \nu, \sigma, \sigma' \rangle_{\vartriangle'} \qquad \text{where } \sigma'(\ell') = n_{\checkmark} && (8) \\
\langle a, \square\, (t_2, e_2) :: \alpha, \sigma, \sigma' \rangle_{\vartriangle} &\to \langle (t_2, e_2), \lceil \square \rceil :: a\,\square :: \alpha, \sigma, \sigma' \rangle_{\triangledown} && (9) \\
\langle a, \quad \lceil \square \rceil :: \nu, \sigma, \sigma' \rangle_{\vartriangle} &\to \langle \lceil a \rceil, \nu, \sigma, \sigma' \rangle_{\vartriangle'} && (9a)
\end{aligned}
$$

(10)

Ghost Abstract Machine

$$
\begin{aligned}
\textit{Applicative Stacks} \ni \quad & \alpha ::= \ell := \square :: \alpha \mid \square\, c :: \alpha \mid \lceil \square \rceil :: \nu \\
\textit{Non-applicative Stacks} \ni \quad & \nu ::= \ell' := \square :: \nu \mid [\,] \mid \lambda x.\square :: \nu \mid a\, \square :: \alpha \\
\textit{Confs} \ni \quad & k ::= \langle c, \alpha, \sigma, \sigma' \rangle_{\triangledown} \mid \langle v, \alpha, \sigma, \sigma' \rangle_{\vartriangle} \mid \langle n, \nu, \sigma, \sigma' \rangle_{\vartriangle'} \\
\textit{Transitions}:
\end{aligned}
$$

$$
\begin{aligned}
t &\mapsto \langle (t, [\,]), \lceil \square \rceil :: [\,], [\,] \rangle_{\triangledown} \\
\langle (t_1\, t_2, e), \alpha, \sigma, \sigma' \rangle_{\triangledown} &\to \langle (t_1, e), \square\, (t_2, e) :: \alpha, \sigma, \sigma' \rangle_{\triangledown} && (1) \\
\langle (\lambda x.t, e), \alpha, \sigma, \sigma' \rangle_{\triangledown} &\to \langle \ell' := (\lambda x.t, e), \alpha, \sigma, \sigma' * [\ell' \mapsto \bot_{\text{✗}}] \rangle_{\vartriangle} && (2) \\
\langle (x, e), \alpha, \sigma, \sigma' \rangle_{\triangledown} &\to \langle (t, e_2), \ell := \square :: \alpha, \sigma, \sigma' \rangle_{\triangledown} && (3) \\
& \qquad \text{where } \ell = e(x),\ \sigma(\ell) = (t, e_2)_{\text{✗}} \\
\langle (x, e), \alpha, \sigma, \sigma' \rangle_{\triangledown} &\to \langle v, \alpha, \sigma, \sigma' \rangle_{\vartriangle} \quad \text{where } \sigma(e(x)) = v_{\checkmark} \vee (v = x \notin e) && (4) \\
\langle v, \ell := \square :: \alpha, \sigma, \sigma' \rangle_{\vartriangle} &\to \langle v, \alpha, \sigma[\ell := v_{\checkmark}], \sigma' \rangle_{\vartriangle} && (5) \\
\langle n, \ell' := \square :: \nu, \sigma, \sigma' \rangle_{\vartriangle'} &\to \langle n, \nu, \sigma, \sigma'[\ell' := n_{\checkmark}] \rangle_{\vartriangle'} && (5') \\
\langle \ell' := (\lambda x.t, e), \square\, (t_2, e_2) :: \alpha, \sigma, \sigma' \rangle_{\vartriangle} &\to \langle (t, e[x := \ell_2]), \alpha, \sigma * [\ell_2 \mapsto (t_2, e_2)_{\text{✗}}], \sigma' \rangle_{\triangledown} && (6) \\
\langle \ell' := (\lambda x.t, e), \quad \lceil \square \rceil :: \nu, \sigma, \sigma' \rangle_{\vartriangle} &\to \langle (t, e[x := \ell_2]), \lceil \square \rceil :: \lambda \check{x}.\square :: \ell' := \square :: \nu, \\
& \qquad \sigma * [\ell_2 \mapsto \check{x}_{\checkmark}], \sigma' \rangle_{\triangledown} \quad \text{where } \sigma'(\ell') = \bot_{\text{✗}} && (7) \\
\langle \ell' := (\lambda x.t, e), \quad \lceil \square \rceil :: \nu, \sigma, \sigma' \rangle_{\vartriangle} &\to \langle n, \nu, \sigma, \sigma' \rangle_{\vartriangle'} \quad \text{where } \sigma'(\ell') = n_{\checkmark} && (8) \\
\langle a, \square\, (t_2, e_2) :: \alpha, \sigma, \sigma' \rangle_{\vartriangle} &\to \langle (t_2, e_2), \lceil \square \rceil :: a\, \square :: \alpha, \sigma, \sigma' \rangle_{\triangledown} && (9) \\
\langle a, \quad \lceil \square \rceil :: \nu, \sigma, \sigma' \rangle_{\vartriangle} &\to \langle \lceil a \rceil, \nu, \sigma, \sigma' \rangle_{\vartriangle'} && (9a)
\end{aligned}
$$

Ghost Abstract Machine

$$
\begin{array}{rrcl}
\textit{Applicative Stacks} \ni & \alpha & ::= & \ell := \Box :: \alpha \mid \Box\, c :: \alpha \mid \lceil \Box \rceil :: \nu \\
\textit{Non-applicative Stacks} \ni & \nu & ::= & \ell' := \Box :: \nu \mid [\,] \mid \lambda x.\Box :: \nu \mid a\,\Box :: \alpha \\
\textit{Confs} \ni & k & ::= & \langle c, \alpha, \sigma, \sigma' \rangle_{\triangledown} \mid \langle v, \alpha, \sigma, \sigma' \rangle_{\vartriangle} \mid \langle n, \nu, \sigma, \sigma' \rangle_{\vartriangle'} \\
\textit{Transitions :} & & &
\end{array}
$$

$$
\begin{array}{rcll}
t & \mapsto & \langle (t, [\,]), \lceil \Box \rceil :: [\,], [\,] \rangle_{\triangledown} & \\
\langle (t_1\, t_2, e), \alpha, \sigma, \sigma' \rangle_{\triangledown} & \longrightarrow & \langle (t_1, e), \Box\,(t_2, e) :: \alpha, \sigma, \sigma' \rangle_{\triangledown} & (1) \\
\langle (\lambda x.t, e), \alpha, \sigma, \sigma' \rangle_{\triangledown} & \longrightarrow & \langle \ell' := (\lambda x.t, e), \alpha, \sigma, \sigma' * [\ell' \mapsto \bot_{\times}] \rangle_{\vartriangle} & (2) \\
\langle (x, e), \alpha, \sigma, \sigma' \rangle_{\triangledown} & \longrightarrow & \langle (t, e_2), \ell := \Box :: \alpha, \sigma, \sigma' \rangle_{\triangledown} & (3) \\
 & & \quad \text{where } \ell = e(x),\ \ \sigma(\ell) = (t, e_2)_{\times} & \\
\langle (x, e), \alpha, \sigma, \sigma' \rangle_{\triangledown} & \longrightarrow & \langle v, \alpha, \sigma, \sigma' \rangle_{\vartriangle} \quad \text{where } \sigma(e(x)) = v_{\checkmark} \vee (v = x \notin e) & (4) \\
\langle v, \ell := \Box :: \alpha, \sigma, \sigma' \rangle_{\vartriangle} & \longrightarrow & \langle v, \alpha, \sigma[\ell := v_{\checkmark}], \sigma' \rangle_{\vartriangle} & (5) \\
\langle n, \ell' := \Box :: \nu, \sigma, \sigma' \rangle_{\vartriangle'} & \longrightarrow & \langle n, \nu, \sigma, \sigma'[\ell' := n_{\checkmark}] \rangle_{\vartriangle'} & (5') \\
\langle \ell' := (\lambda x.t, e), \Box\,(t_2, e_2) :: \alpha, \sigma, \sigma' \rangle_{\vartriangle} & \longrightarrow & \langle (t, e[x := \ell_2]), \alpha, \sigma * [\ell_2 \mapsto (t_2, e_2)_{\times}], \sigma' \rangle_{\triangledown} & (6) \\
\langle \ell' := (\lambda x.t, e), \quad \lceil \Box \rceil :: \nu, \sigma, \sigma' \rangle_{\vartriangle} & \longrightarrow & \langle (t, e[x := \ell_2]), \lceil \Box \rceil :: \lambda \check{x}.\Box :: \ell' := \Box :: \nu, & \\
 & & \quad \sigma * [\ell_2 \mapsto \check{x}_{\checkmark}], \sigma' \rangle_{\triangledown} \quad \text{where } \sigma'(\ell') = \bot_{\times} & (7) \\
\langle \ell' := (\lambda x.t, e), \quad \lceil \Box \rceil :: \nu, \sigma, \sigma' \rangle_{\vartriangle} & \longrightarrow & \langle n, \nu, \sigma, \sigma' \rangle_{\vartriangle'} \quad \text{where } \sigma'(\ell') = n_{\checkmark} & (8) \\
\langle a, \Box\,(t_2, e_2) :: \alpha, \sigma, \sigma' \rangle_{\vartriangle} & \longrightarrow & \langle (t_2, e_2), \lceil \Box \rceil :: a\,\Box :: \alpha, \sigma, \sigma' \rangle_{\triangledown} & (9) \\
\langle a, \quad \lceil \Box \rceil :: \nu, \sigma, \sigma' \rangle_{\vartriangle} & \longrightarrow & \langle \lceil a \rceil, \nu, \sigma, \sigma' \rangle_{\vartriangle'} & (9a)
\end{array}
$$

Ghost Abstract Machine

$$
\begin{aligned}
\textit{Applicative Stacks} \ni \quad & \alpha ::= \ell := \square :: \alpha \;\mid\; \square\, c :: \alpha \;\mid\; \lceil \square \rceil :: \nu \\
\textit{Non-applicative Stacks} \ni \quad & \nu ::= \ell' := \square :: \nu \;\mid\; [\,] \;\mid\; \lambda x.\square :: \nu \;\mid\; a\,\square :: \alpha \\
\textit{Confs} \ni \quad & k ::= \langle c, \alpha, \sigma, \sigma' \rangle_{\triangledown} \;\mid\; \langle v, \alpha, \sigma, \sigma' \rangle_{\vartriangle} \;\mid\; \langle n, \nu, \sigma, \sigma' \rangle_{\vartriangle'} \\
\textit{Transitions} :
\end{aligned}
$$

$$t \mapsto \langle (t, [\,]), \lceil \square \rceil :: [\,], [\,] \rangle_{\triangledown}$$

$$\langle (t_1\, t_2, e), \alpha, \sigma, \sigma' \rangle_{\triangledown} \to \langle (t_1, e), \square\, (t_2, e) :: \alpha, \sigma, \sigma' \rangle_{\triangledown} \quad (1)$$

$$\langle (\lambda x.t, e), \alpha, \sigma, \sigma' \rangle_{\triangledown} \to \langle \ell' := (\lambda x.t, e), \alpha, \sigma, \sigma' * [\ell' \mapsto \bot_{\text{✗}}] \rangle_{\vartriangle} \quad (2)$$

$$\langle (x, e), \alpha, \sigma, \sigma' \rangle_{\triangledown} \to \langle (t, e_2), \ell := \square :: \alpha, \sigma, \sigma' \rangle_{\triangledown} \quad (3)$$

$$\text{where } \ell = e(x),\ \ \sigma(\ell) = (t, e_2)_{\text{✗}}$$

$$\langle (x, e), \alpha, \sigma, \sigma' \rangle_{\triangledown} \to \langle v, \alpha, \sigma, \sigma' \rangle_{\vartriangle} \quad \text{where } \sigma(e(x)) = v_{\checkmark} \vee (v = x \notin e) \quad (4)$$

$$\langle v, \ell := \square :: \alpha, \sigma, \sigma' \rangle_{\vartriangle} \to \langle v, \alpha, \sigma[\ell := v_{\checkmark}], \sigma' \rangle_{\vartriangle} \quad (5)$$

$$\langle n, \ell' := \square :: \nu, \sigma, \sigma' \rangle_{\vartriangle'} \to \langle n, \nu, \sigma, \sigma'[\ell' := n_{\checkmark}] \rangle_{\vartriangle'} \quad (5')$$

$$\langle \ell' := (\lambda x.t, e), \square\, (t_2, e_2) :: \alpha, \sigma, \sigma' \rangle_{\vartriangle} \to \langle (t, e[x := \ell_2]), \alpha, \sigma * [\ell_2 \mapsto (t_2, e_2)_{\text{✗}}], \sigma' \rangle_{\triangledown} \quad (6)$$

$$\langle \ell' := (\lambda x.t, e), \lceil \square \rceil :: \nu, \sigma, \sigma' \rangle_{\vartriangle} \to \langle (t, e[x := \ell_2]), \lceil \square \rceil :: \lambda \check{x}.\square :: \ell' := \square :: \nu,$$

$$\sigma * [\ell_2 \mapsto \check{x}_{\checkmark}], \sigma' \rangle_{\triangledown} \quad \text{where } \sigma'(\ell') = \bot_{\text{✗}} \quad (7)$$

$$\langle \ell' := (\lambda x.t, e), \lceil \square \rceil :: \nu, \sigma, \sigma' \rangle_{\vartriangle} \to \langle n, \nu, \sigma, \sigma' \rangle_{\vartriangle'} \quad \text{where } \sigma'(\ell') = n_{\checkmark} \quad (8)$$

$$\langle a, \square\, (t_2, e_2) :: \alpha, \sigma, \sigma' \rangle_{\vartriangle} \to \langle (t_2, e_2), \lceil \square \rceil :: a\,\square :: \alpha, \sigma, \sigma' \rangle_{\triangledown} \quad (9)$$

$$\langle a, \lceil \square \rceil :: \nu, \sigma, \sigma' \rangle_{\vartriangle} \to \langle \lceil a \rceil, \nu, \sigma, \sigma' \rangle_{\vartriangle'} \quad (9a)$$

Potential Function

Table 6. The potential function for RKNL

$$\Phi_{\mathrm{t}}(t_1\ t_2) := 3 + \Phi_{\mathrm{t}}(t_1) + \Phi_{\mathrm{t}}(t_2)$$

$$\Phi_{\mathrm{t}}(\lambda x.t) := 4 + \Phi_{\mathrm{t}}(t)$$

$$\Phi_{\mathrm{t}}(x) := 2$$

$$\Phi_{\mathrm{v}}(t) := 0$$

$$\Phi_{\mathrm{v}}(\ell := (\lambda x.t, e)) := 1$$

$$\Phi_{\mathrm{s}}([\,]) := 0$$

$$\Phi_{\mathrm{s}}(\square\,(t, e) :: s) := 2 + \Phi_{\mathrm{s}}(s) + \Phi_{\mathrm{t}}(t)$$

$$\Phi_{\mathrm{s}}(t\,\square :: s) := 1 + \Phi_{\mathrm{s}}(s)$$

$$\Phi_{\mathrm{s}}(\lambda x.\square :: s) := 1 + \Phi_{\mathrm{s}}(s)$$

$$\Phi_{\mathrm{s}}(\ell := \square :: s) := 1 + \Phi_{\mathrm{s}}(s)$$

$$\Phi_{\sigma}(k) := \sum_{\ell \in k \wedge \sigma(\ell) = (t,e)_{\text{✗}} \wedge \ell := \square \notin s} \Phi_{\mathrm{t}}(t) + \sum_{\ell := (\lambda x.t, e) \in k \wedge \sigma(\ell) = \bot_{\text{✗}} \wedge \ell := \square \notin s} (2 + \Phi_{\mathrm{t}}(t))$$

$$\Phi_{\mathrm{k}}(\langle (t, e), s, \sigma \rangle_{\triangledown}) := \Phi_{\mathrm{t}}(t) + \Phi_{\mathrm{s}}(s) + \Phi_{\sigma}(k)$$

$$\Phi_{\mathrm{k}}(\langle v, s, \sigma \rangle_{\vartriangle}) := \Phi_{\mathrm{v}}(v) + \Phi_{\mathrm{s}}(s) + \Phi_{\sigma}(k)$$

Potential Function

Table 6. The potential function for RKNL

$$
\begin{aligned}
\Phi_{\mathrm{t}}(t_1\ t_2) &:= \boxed{3} + \Phi_{\mathrm{t}}(t_1) + \Phi_{\mathrm{t}}(t_2) \\
\Phi_{\mathrm{t}}(\lambda x.t) &:= 4 + \Phi_{\mathrm{t}}(t) \\
\Phi_{\mathrm{t}}(x) &:= 2 \\
\Phi_{\mathrm{v}}(t) &:= \boxed{0} \\
\Phi_{\mathrm{v}}(\ell := (\lambda x.t, e)) &:= 1
\end{aligned}
\qquad
\begin{aligned}
\Phi_{\mathrm{s}}([\,]) &:= 0 \\
\Phi_{\mathrm{s}}(\square\,(t, e) :: s) &:= \boxed{2} + \Phi_{\mathrm{s}}(s) + \Phi_{\mathrm{t}}(t) \\
\Phi_{\mathrm{s}}(t\,\square :: s) &:= \boxed{1} + \Phi_{\mathrm{s}}(s) \\
\Phi_{\mathrm{s}}(\lambda x.\square :: s) &:= 1 + \Phi_{\mathrm{s}}(s) \\
\Phi_{\mathrm{s}}(\ell := \square :: s) &:= 1 + \Phi_{\mathrm{s}}(s)
\end{aligned}
$$

$$
\Phi_\sigma(k) := \sum_{\ell \in k \wedge \sigma(\ell) = (t,e)_{\text{✗}} \wedge \ell := \square \notin s} \Phi_{\mathrm{t}}(t) + \sum_{\ell := (\lambda x.t, e) \in k \wedge \sigma(\ell) = \bot_{\text{✗}} \wedge \ell := \square \notin s} (2 + \Phi_{\mathrm{t}}(t))
$$

$$
\begin{aligned}
\Phi_{\mathrm{k}}(\langle (t, e), s, \sigma \rangle_{\triangledown}) &:= \Phi_{\mathrm{t}}(t) + \Phi_{\mathrm{s}}(s) + \Phi_\sigma(k) \\
\Phi_{\mathrm{k}}(\langle v, s, \sigma \rangle_{\vartriangle}) &:= \Phi_{\mathrm{v}}(v) + \Phi_{\mathrm{s}}(s) + \Phi_\sigma(k)
\end{aligned}
$$

RKNL

$$
\begin{aligned}
\langle (t_1\ t_2, e), s, \sigma\rangle_{\triangledown} &\longrightarrow \langle (t_1, e), \square\, (t_2, e) :: s, \sigma\rangle_{\triangledown} && (1)\\
\langle (\lambda x.t, e), s, \sigma\rangle_{\triangledown} &\longrightarrow \langle \ell := (\lambda x.t, e), s, \sigma * [\ell \mapsto \bot_{\text{✗}}]\rangle_{\triangle} && (2)\\
\langle (x, e), s, \sigma\rangle_{\triangledown} &\longrightarrow \langle (t, e_2), \ell := \square :: s, \sigma\rangle_{\triangledown} \text{ where } \ell = e(x),\ \sigma(\ell) = (t, e_2)_{\text{✗}} && (3)\\
\langle (x, e), s, \sigma\rangle_{\triangledown} &\longrightarrow \langle v, s, \sigma\rangle_{\triangle} \qquad \text{where } \sigma(e(x)) = v_{\checkmark} \lor (v = x \notin e) && (4)\\
\langle v, \ell := \square :: s, \sigma\rangle_{\triangle} &\longrightarrow \langle v, s, \sigma[\ell := v_{\checkmark}]\rangle_{\triangle} && (5)\\
\langle \ell := (\lambda x.t, e), \square\, (t_2, e_2) :: s, \sigma\rangle_{\triangle} &\longrightarrow \langle (t, e[x := \ell_2]), s, \sigma * [\ell_2 \mapsto (t_2, e_2)_{\text{✗}}]\rangle_{\triangledown} && (6)\\
\langle \ell := (\lambda x.t, e), \qquad s, \sigma\rangle_{\triangle} &\longrightarrow \langle (t, e[x := \ell_2]), \lambda\check{x}.\square :: \ell := \square :: s, \sigma * [\ell_2 \mapsto \check{x}_{\checkmark}]\rangle_{\triangledown} \\
&\qquad \text{where } \sigma(\ell) = \bot_{\text{✗}} && (7)\\
\langle \ell := (\lambda x.t, e), \qquad s, \sigma\rangle_{\triangle} &\longrightarrow \langle v, s, \sigma\rangle_{\triangle} \qquad \text{where } \sigma(\ell) = v_{\checkmark} && (8)\\
\langle t, \square\, (t_2, e_2) :: s, \sigma\rangle_{\triangle} &\longrightarrow \langle (t_2, e_2), t\, \square :: s, \sigma\rangle_{\triangledown} && (9)\\
\langle t_2, t_1\, \square :: s, \sigma\rangle_{\triangle} &\longrightarrow \langle t_1\ t_2, s, \sigma\rangle_{\triangle} && (10)\\
\langle t, \lambda x.\square :: s, \sigma\rangle_{\triangle} &\longrightarrow \langle \lambda x.t, s, \sigma\rangle_{\triangle} && (11)
\end{aligned}
$$

Potential Decrease

$$3 + \Phi_{\mathrm{t}}(t_1) + \Phi_{\mathrm{t}}(t_2) + \Phi_{\mathrm{s}}(s) + \Phi_{\sigma}(k) > \Phi_{\mathrm{t}}(t_1) + 2 + \Phi_{\mathrm{s}}(s) + \Phi_{\mathrm{t}}(t_2) + \Phi_{\sigma}(k) \tag{1}$$

$$4 + \Phi_{\mathrm{t}}(t) + \Phi_{\mathrm{s}}(s) + \Phi_{\sigma}(k) > 1 + \Phi_{\mathrm{s}}(s) + \Phi_{\sigma}(k) + (2 + \Phi_{\mathrm{t}}(t)) \tag{2}$$

$$2 + \Phi_{\mathrm{s}}(s) + (\Phi_{\sigma}(k') + \Phi_{\mathrm{t}}(t)) > \Phi_{\mathrm{t}}(t) + (1 + \Phi_{\mathrm{s}}(s)) + \Phi_{\sigma}(k') \tag{3}$$

$$2 + \Phi_{\mathrm{s}}(s) + \Phi_{\sigma}(k) > \Phi_{\mathrm{v}}(v) + \Phi_{\mathrm{s}}(s) + \Phi_{\sigma}(k) \tag{4}$$

$$\Phi_{\mathrm{v}}(v) + 1 + \Phi_{\mathrm{s}}(s) + \Phi_{\sigma}(k) > \Phi_{\mathrm{v}}(v) + \Phi_{\mathrm{s}}(s) + \Phi_{\sigma}(k) \tag{5}$$

$$1 + \Phi_{\mathrm{s}}(s) + (\Phi_{\sigma}(k') + 2 + \Phi_{\mathrm{t}}(t)) > \Phi_{\mathrm{t}}(t) + 1 + 1 + \Phi_{\mathrm{s}}(s) + \Phi_{\sigma}(k') \tag{7}$$

$$1 + \Phi_{\mathrm{s}}(s) + \Phi_{\sigma}(k) > 0 + \Phi_{\mathrm{s}}(s) + \Phi_{\sigma}(k) \tag{8}$$

$$0 + 2 + \Phi_{\mathrm{s}}(s) + \Phi_{\mathrm{t}}(t_2) + \Phi_{\sigma}(k) > \Phi_{\mathrm{t}}(t_2) + 1 + \Phi_{\mathrm{s}}(s) + \Phi_{\sigma}(k) \tag{9}$$

$$0 + 1 + \Phi_{\mathrm{s}}(s) + \Phi_{\sigma}(k) > 0 + \Phi_{\mathrm{s}}(s) + \Phi_{\sigma}(k) \tag{10}$$

$$0 + 1 + \Phi_{\mathrm{s}}(s) + \Phi_{\sigma}(k) > 0 + \Phi_{\mathrm{s}}(s) + \Phi_{\sigma}(k) \tag{11}$$

Potential Decrease and Increase

$$3 + \Phi_{\mathrm{t}}(t_1) + \Phi_{\mathrm{t}}(t_2) + \Phi_{\mathrm{s}}(s) + \Phi_{\sigma}(k) > \Phi_{\mathrm{t}}(t_1) + 2 + \Phi_{\mathrm{s}}(s) + \Phi_{\mathrm{t}}(t_2) + \Phi_{\sigma}(k) \tag{1}$$

$$4 + \Phi_{\mathrm{t}}(t) + \Phi_{\mathrm{s}}(s) + \Phi_{\sigma}(k) > 1 + \Phi_{\mathrm{s}}(s) + \Phi_{\sigma}(k) + (2 + \Phi_{\mathrm{t}}(t)) \tag{2}$$

$$2 + \Phi_{\mathrm{s}}(s) + (\Phi_{\sigma}(k') + \Phi_{\mathrm{t}}(t)) > \Phi_{\mathrm{t}}(t) + (1 + \Phi_{\mathrm{s}}(s)) + \Phi_{\sigma}(k') \tag{3}$$

$$2 + \Phi_{\mathrm{s}}(s) + \Phi_{\sigma}(k) > \Phi_{\mathrm{v}}(v) + \Phi_{\mathrm{s}}(s) + \Phi_{\sigma}(k) \tag{4}$$

$$\Phi_{\mathrm{v}}(v) + 1 + \Phi_{\mathrm{s}}(s) + \Phi_{\sigma}(k) > \Phi_{\mathrm{v}}(v) + \Phi_{\mathrm{s}}(s) + \Phi_{\sigma}(k) \tag{5}$$

$$1 + 2 + \Phi_{\mathrm{s}}(s) + \Phi_{\mathrm{t}}(t_2) + \Phi_{\sigma}(k) + \Phi_{\mathrm{t}}(t_0) > \Phi_{\mathrm{t}}(t) + 1 + 1 + \Phi_{\mathrm{s}}(s) + \Phi_{\sigma}(k) + \Phi_{\mathrm{t}}(t_2) \tag{6}$$

$$1 + \Phi_{\mathrm{s}}(s) + (\Phi_{\sigma}(k') + 2 + \Phi_{\mathrm{t}}(t)) > \Phi_{\mathrm{t}}(t) + 1 + 1 + \Phi_{\mathrm{s}}(s) + \Phi_{\sigma}(k') \tag{7}$$

$$1 + \Phi_{\mathrm{s}}(s) + \Phi_{\sigma}(k) > 0 + \Phi_{\mathrm{s}}(s) + \Phi_{\sigma}(k) \tag{8}$$

$$0 + 2 + \Phi_{\mathrm{s}}(s) + \Phi_{\mathrm{t}}(t_2) + \Phi_{\sigma}(k) > \Phi_{\mathrm{t}}(t_2) + 1 + \Phi_{\mathrm{s}}(s) + \Phi_{\sigma}(k) \tag{9}$$

$$0 + 1 + \Phi_{\mathrm{s}}(s) + \Phi_{\sigma}(k) > 0 + \Phi_{\mathrm{s}}(s) + \Phi_{\sigma}(k) \tag{10}$$

$$0 + 1 + \Phi_{\mathrm{s}}(s) + \Phi_{\sigma}(k) > 0 + \Phi_{\mathrm{s}}(s) + \Phi_{\sigma}(k) \tag{11}$$

Efficient Implementation

Theorem 5.13.

Let ρ be a sequence of consecutive machine transitions starting from term t_0 to configuration k',

$|\rho|$ be the number of steps in ρ,

and $|\rho|_\beta$ be the number of normal-order β-reductions from t_0 to $\underline{k'}_{\mathrm{k}}$.

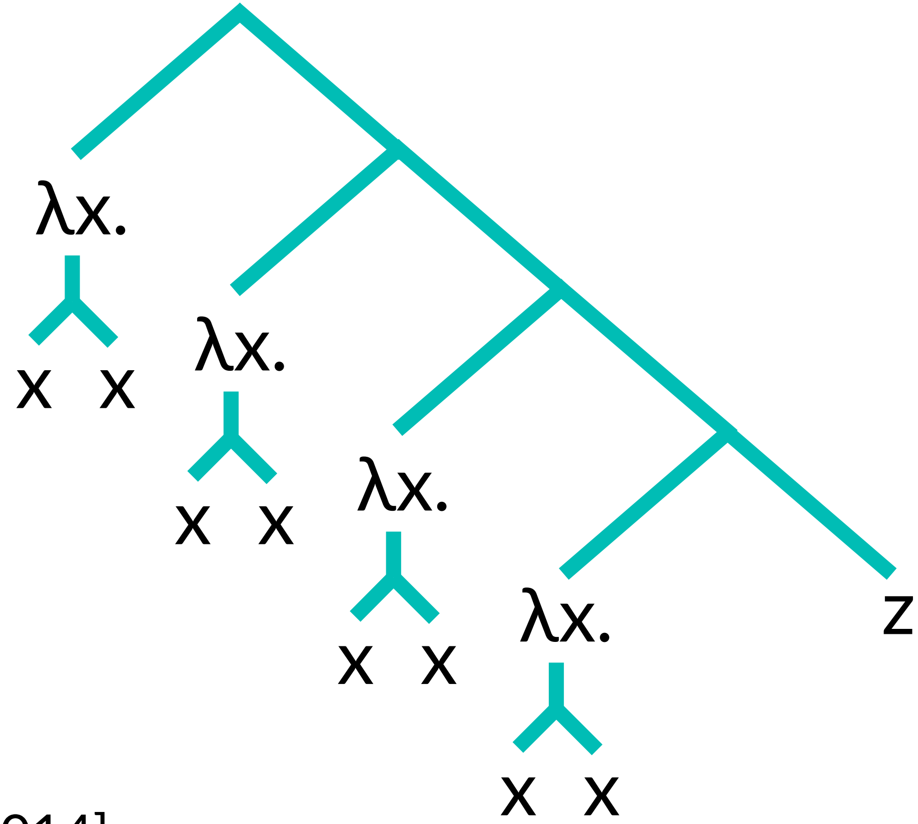
$$\text{Then } |\rho| \leq (|\rho|_\beta + 1) \cdot \Phi_t(t_0).$$

Empirical Method

Table 3. Empirical execution lengths for $1 \leq n \leq 9$

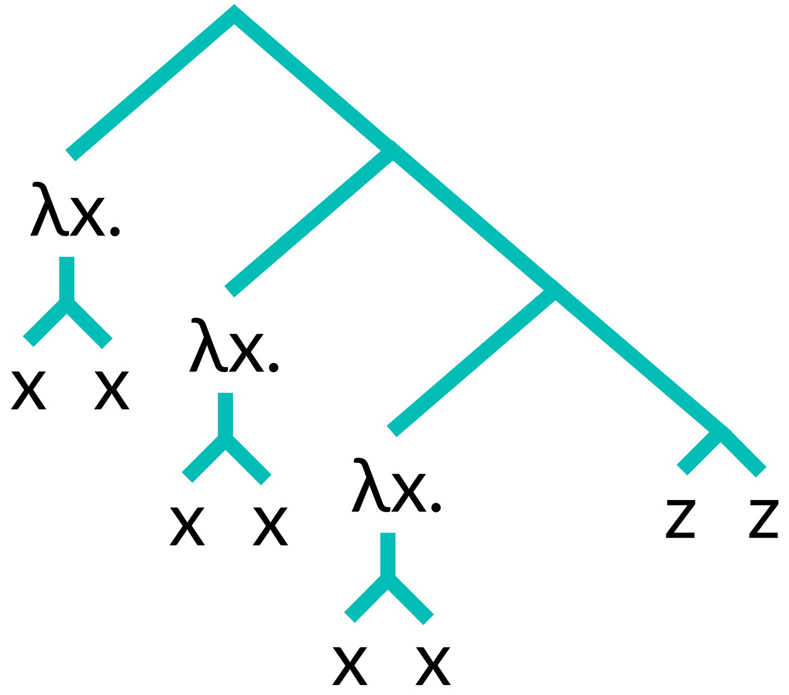
term family	$\xrightarrow{no}$	KN	RKNL
$c_n\, c_2\, I$	$3 \cdot 2^n - 1$	$15 \cdot 2^n - 6$	$10 \cdot 2^n + 5n + 5$
$pred\, c_n$	$6n + 8$	$26n + 25$	$30n + 41$
$\lambda x.c_n\, \omega\, x$	$2^n + 1$	$12 \cdot 2^n - 3$	$9n + 15$
$c_n\, dub\, I$	$2^n + 1$	$23 \cdot 2^n - 14$	$18n + 15$
$c_n\, dub\, (\lambda x.I\, x)$	$2 \cdot 2^n + 1$	$26 \cdot 2^n - 14$	$18n + 20$

Size Explosion Problem



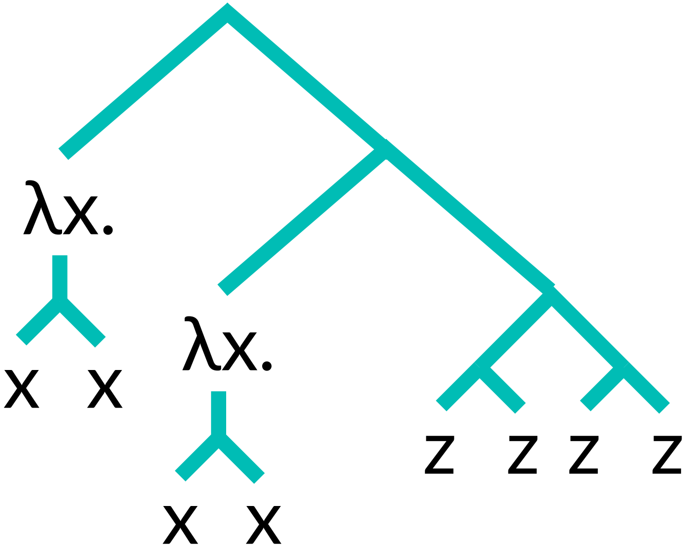
named in
[Accattoli, Dal Lago, CSL-LICS 2014]

Size Explosion Problem



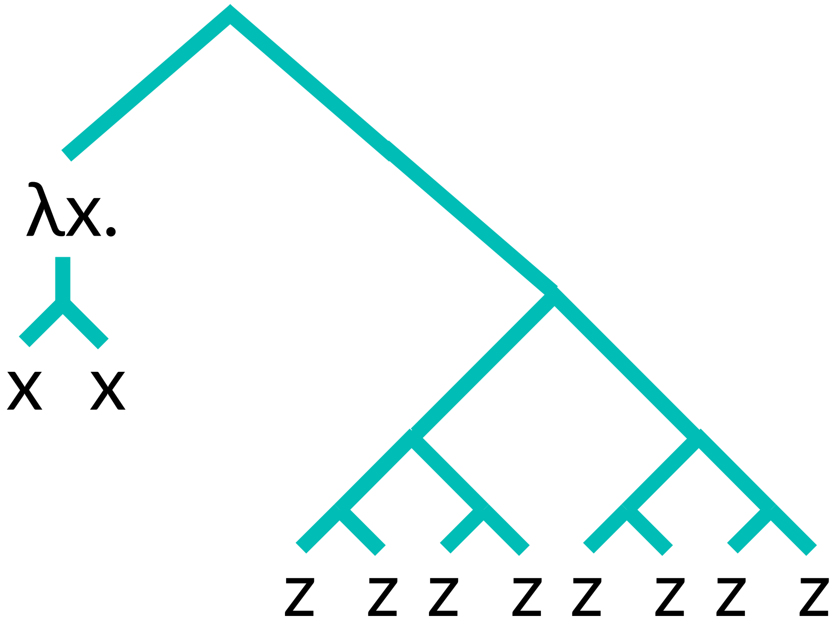
named in
[Accattoli, Dal Lago, CSL-LICS 2014]

Size Explosion Problem



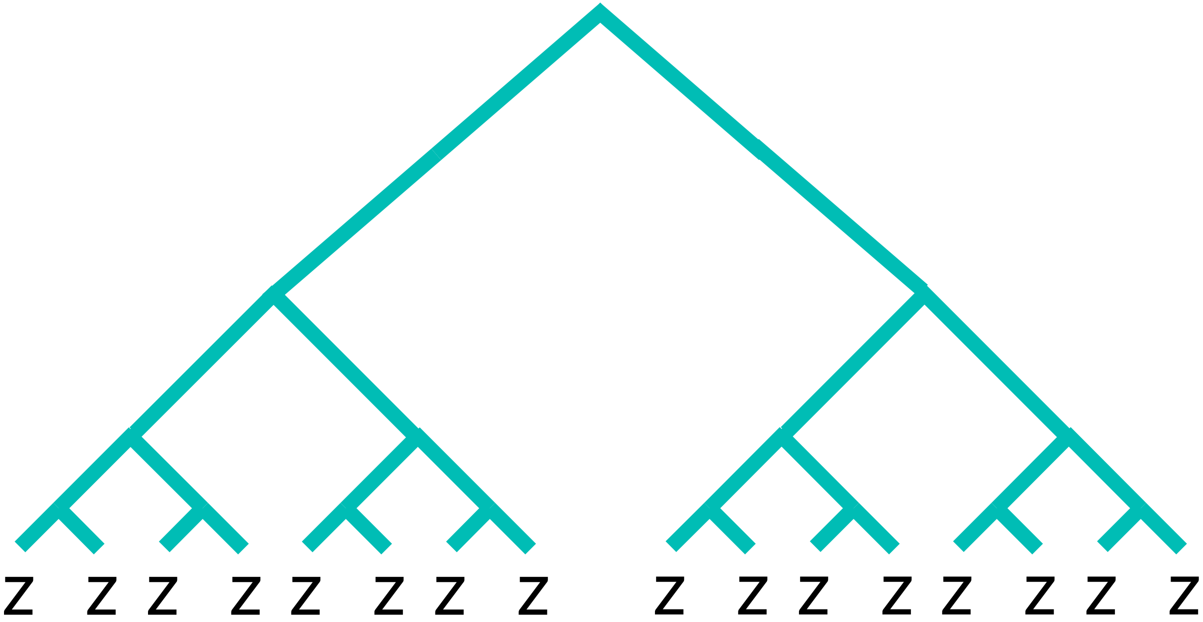
named in
[Accattoli, Dal Lago, CSL-LICS 2014]

Size Explosion Problem



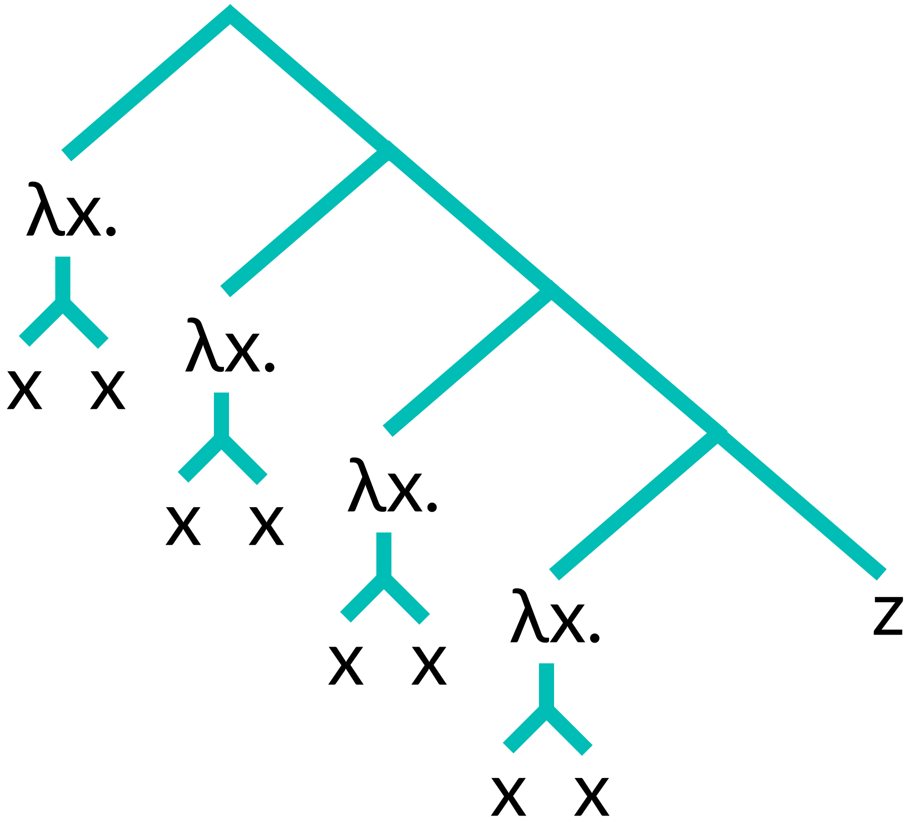
named in
[Accattoli, Dal Lago, CSL-LICS 2014]

Size Explosion Problem

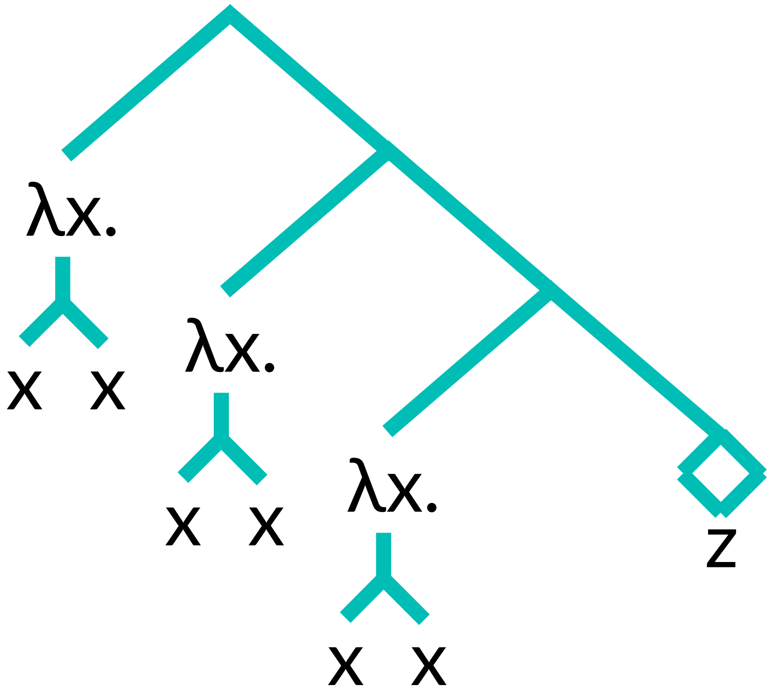


named in
[Accattoli, Dal Lago, CSL-LICS 2014]

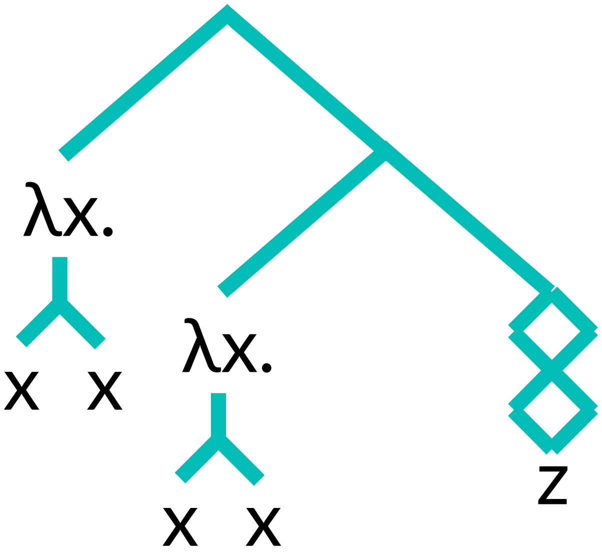
Implicit Sharing



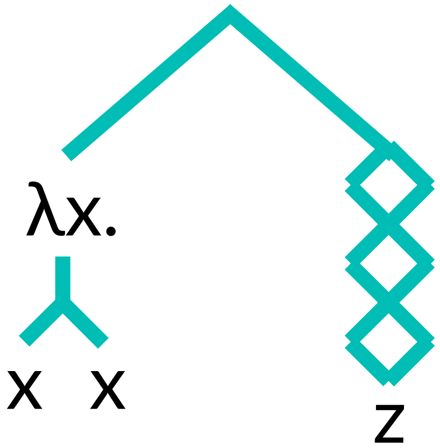
Implicit Sharing



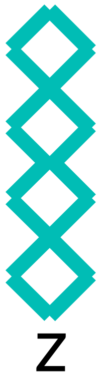
Implicit Sharing



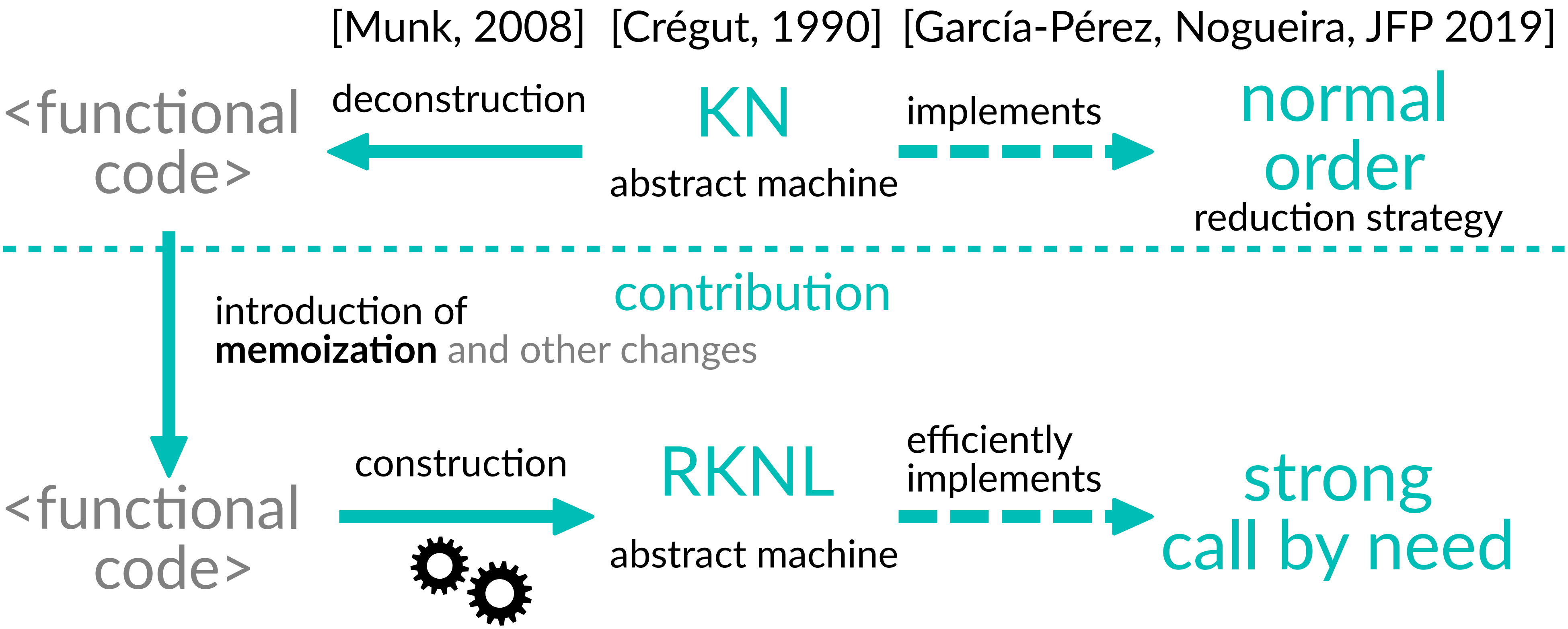
Implicit Sharing



Implicit Sharing



Summary


[Munk, 2008]
[Crégut, 1990]
[García-Pérez, Nogueira, JFP 2019]
<functional code>
deconstruction
KN
abstract machine
implements
normal order
reduction strategy
contribution
introduction of
memoization and other changes
<functional code>
construction
RKNL
abstract machine
efficiently implements
strong call by need

Bonus

Table 2. Elaborate example execution in refocusing notation

0 :	$\langle((\lambda x.c\,x\,x)\,(A\,\Omega),[])\rangle_{\triangledown}\,\|\,[]$	$\xrightarrow{(1)}$
1 :	$\langle(\lambda x.c\,x\,x,[])\rangle_{\triangledown}\,(A\,\Omega,[])\,\|\,[]$	$\xrightarrow{(2)}$
2 :	$\langle \mathbb{a} := (\lambda x.c\,x\,x,[])\rangle_{\vartriangle}\,(A\,\Omega,[])\,\|\,[\mathbb{a}\mapsto\bot\,\text{✗}]$	$\xrightarrow{(6)}$
3 :	$\langle(c\,x\,x,[x\mapsto\mathbb{x}])\rangle_{\triangledown}\,\|\,[\mathbb{x}\mapsto(A\,\Omega,[])\,\text{✗}]$	$\xrightarrow{(1)}$
4 :	$\langle(c\,x,[x\mapsto\mathbb{x}])\rangle_{\triangledown}\,x^{\mathbb{x}}\,\|\,[\mathbb{x}\mapsto(A\,\Omega,[])\,\text{✗}]$	$\xrightarrow{(1)}$
5 :	$\langle(c,[x\mapsto\mathbb{x}])\rangle_{\triangledown}\,x^{\mathbb{x}}\,x^{\mathbb{x}}\,\|\,[\mathbb{x}\mapsto(A\,\Omega,[])\,\text{✗}]$	$\xrightarrow{(4)}$
6 :	$\langle c\rangle_{\vartriangle}\,x^{\mathbb{x}}\,x^{\mathbb{x}}\,\|\,[\mathbb{x}\mapsto(A\,\Omega,[])\,\text{✗}]$	$\xrightarrow{(9)}$
7 :	$c\,\langle x^{\mathbb{x}}\rangle_{\triangledown}\,x^{\mathbb{x}}\,\|\,[\mathbb{x}\mapsto(A\,\Omega,[])\,\text{✗}]$	$\xrightarrow{(3)}$
8 :	$c\,\left(\mathbb{x} := \langle(A\,\Omega,[])\rangle_{\triangledown}\right)\,x^{\mathbb{x}}\,\|\,[\mathbb{x}\mapsto(A\,\Omega,[])\,\text{✗}]$	$\xrightarrow{(1)}$

$$9:\quad c\left(\mathbb{x}:=\langle(A,[])\rangle_{\triangledown}\ (\Omega,[])\right)\ x^{\mathbb{x}}\,|\,[\mathbb{x}\mapsto(A\,\Omega,[])_{\text{✗}}] \quad\xrightarrow{(2)}$$

$$10:\quad c\left(\mathbb{x}:=\langle\mathbb{b}:=(A,[])\rangle_{\vartriangle}\ (\Omega,[])\right)\ x^{\mathbb{x}}\,|\,[\mathbb{x}\mapsto(A\,\Omega,[])_{\text{✗}},\mathbb{b}\mapsto\bot_{\text{✗}}] \quad\xrightarrow{(6)}$$

$$11:\quad c\left(\mathbb{x}:=\langle(\lambda z.I\,z,[y\mapsto\mathbb{y}])\rangle_{\triangledown}\right)\ x^{\mathbb{x}}\,|\,[\mathbb{x}\mapsto(A\,\Omega,[])_{\text{✗}},\mathbb{y}\mapsto(\Omega,[])_{\text{✗}}] \quad\xrightarrow{(2)}$$

$$12:\quad c\left(\mathbb{x}:=\langle\mathbb{d}:=(\lambda z.I\,z,[y\mapsto\mathbb{y}])\rangle_{\vartriangle}\right)\ x^{\mathbb{x}}\,|\,[\mathbb{x}\mapsto(A\,\Omega,[])_{\text{✗}},\mathbb{d}\mapsto\bot_{\text{✗}},\mathbb{y}\mapsto(\Omega,[])_{\text{✗}}] \quad\xrightarrow{(5)}$$

$$13:\quad c\,\langle\mathbb{d}:=(\lambda z.I\,z,[y\mapsto\mathbb{y}])\rangle_{\vartriangle}\ x^{\mathbb{x}}\,|\,\sigma_1*[\mathbb{d}\mapsto\bot_{\text{✗}}] \quad\xrightarrow{(7)}$$

$$14:\quad c\left(\mathbb{d}:=\lambda z_0.\langle(I\,z,e_{yz})\rangle_{\triangledown}\right)\ x^{\mathbb{x}}\,|\,\sigma_1*[\mathbb{d}\mapsto\bot_{\text{✗}},\mathbb{z}\mapsto z_{0\checkmark}] \quad\xrightarrow{(1)}$$

$$15:\quad c\left(\mathbb{d}:=\lambda z_0.\langle(I,e_{yz})\rangle_{\triangledown}\ (z,e_{yz})\right)\ x^{\mathbb{x}}\,|\,\sigma_1*[\mathbb{d}\mapsto\bot_{\text{✗}},\mathbb{z}\mapsto z_{0\checkmark}] \quad\xrightarrow{(2)}$$

$$16:\quad c\left(\mathbb{d}:=\lambda z_0.\langle\mathbb{e}:=(I,e_{yz})\rangle_{\vartriangle}\ (z,e_{yz})\right)\ x^{\mathbb{x}}\,|\,\sigma_1*[\mathbb{d}\mapsto\bot_{\text{✗}},\mathbb{e}\mapsto\bot_{\text{✗}},\mathbb{z}\mapsto z_{0\checkmark}] \quad\xrightarrow{(6)}$$

$$17:\quad c\left(\mathbb{d}:=\lambda z_0.\langle(x,e_{yz}*[x\mapsto\mathbb{w}])\rangle_{\triangledown}\right)\ x^{\mathbb{x}}\,|\,\sigma_1*[\mathbb{d}\mapsto\bot_{\text{✗}},\mathbb{w}\mapsto(z,e_{yz})_{\text{✗}},\mathbb{z}\mapsto z_{0\checkmark}] \quad\xrightarrow{(3)}$$

$$18:\quad c\left(\mathbb{d}:=\lambda z_0.\mathbb{w}:=\langle(z,e_{yz})\rangle_{\triangledown}\right)\ x^{\mathbb{x}}\,|\,\sigma_1*[\mathbb{d}\mapsto\bot_{\text{✗}},\mathbb{w}\mapsto(z,e_{yz})_{\text{✗}},\mathbb{z}\mapsto z_{0\checkmark}] \quad\xrightarrow{(4)}$$

$$19:\quad c\left(\mathbb{d}:=\lambda z_0.\mathbb{w}:=\langle z_0\rangle_{\triangle}\right) x^{\mathbb{x}} \mid \sigma_1 * [\mathbb{d}\mapsto \bot_{\times}, \mathbb{w}\mapsto (z,e_{yz})_{\times}] \quad \xrightarrow{(5)}$$

$$20:\quad c\left(\mathbb{d}:=\lambda z_0.\langle z_0\rangle_{\triangle}\right) x^{\mathbb{x}} \mid \sigma_1 * [\mathbb{d}\mapsto \bot_{\times}] \quad \xrightarrow{(11)}$$

$$21:\quad c\left(\mathbb{d}:=\langle \lambda z_0.z_0\rangle_{\triangle}\right) x^{\mathbb{x}} \mid \sigma_1 * [\mathbb{d}\mapsto \bot_{\times}] \quad \xrightarrow{(5)}$$

$$22:\quad c\,\langle \lambda z_0.z_0\rangle_{\triangle}\, x^{\mathbb{x}} \mid \sigma_1 * [\mathbb{d}\mapsto \lambda z_0.z_{0\,\checkmark}] \quad \xrightarrow{(10)}$$

$$23:\quad \langle c\,\lambda z_0.z_0\rangle_{\triangle}\, x^{\mathbb{x}} \mid \sigma_1 * [\mathbb{d}\mapsto \lambda z_0.z_{0\,\checkmark}] \quad \xrightarrow{(9)}$$

$$24:\quad c\,(\lambda z_0.z_0)\,\langle x^{\mathbb{x}}\rangle_{\triangledown} \mid \sigma_1 * [\mathbb{d}\mapsto \lambda z_0.z_{0\,\checkmark}] \quad \xrightarrow{(4)}$$

$$25:\quad c\,(\lambda z_0.z_0)\,\langle \mathbb{d}:=(\lambda z.I\,z,[y\mapsto \mathbb{y}])\rangle_{\triangle} \mid [\mathbb{d}\mapsto \lambda z_0.z_{0\,\checkmark}, \mathbb{y}\mapsto (\Omega,[])_{\times}] \quad \xrightarrow{(8)}$$

$$26:\quad c\,(\lambda z_0.z_0)\,\langle \lambda z_0.z_0\rangle_{\triangle} \mid [] \quad \xrightarrow{(10)}$$

$$27:\quad \langle c\,(\lambda z_0.z_0)\,\lambda z_0.z_0\rangle_{\triangle} \mid [] \quad \not\rightarrow$$