Reductions and Causality (VI)

jean-jacques.levy@inria.fr
Tsinghua University,
November 18, 2011

http://pauillac.inria.fr/~levy/courses/tsinghua/reductions

Labeled λ-calculus

CENTRE DE RECHERCHE COMMUN
INRIA MICROSOFT RESEARCH

Plan

- complete reductions
- sublattice of complete reductions
- more on canonical representatives
- costs of reductions + sharing
- speculative computations
- other calculi
- connection with event structures
- local compactness proof technique

A labeled λ-calculus (1/2)

- Give names to every redex and try make this naming consistent with permutation equivalence.

- Need give names to every subterm:

$$M, N, \ldots ::= {}^{\alpha}x \mid {}^{\alpha}(MN) \mid {}^{\alpha}(\lambda x.M)$$

- Conversion rule is:

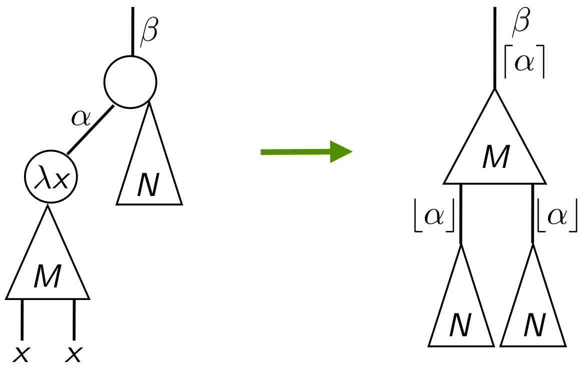
$${}^{\beta}({}^{\alpha}(\lambda x.M)N) \longrightarrow {}^{\beta\lceil\alpha\rceil} M\{x := {}^{\lfloor\alpha\rfloor}N\}$$

α is **name** of redex

where

$${}^{\alpha}({}^{\beta}U) = {}^{\alpha\beta}U \quad \text{and} \quad {}^{\alpha}x\{x := M\} = {}^{\alpha}M$$

A labeled λ-calculus (2/2)

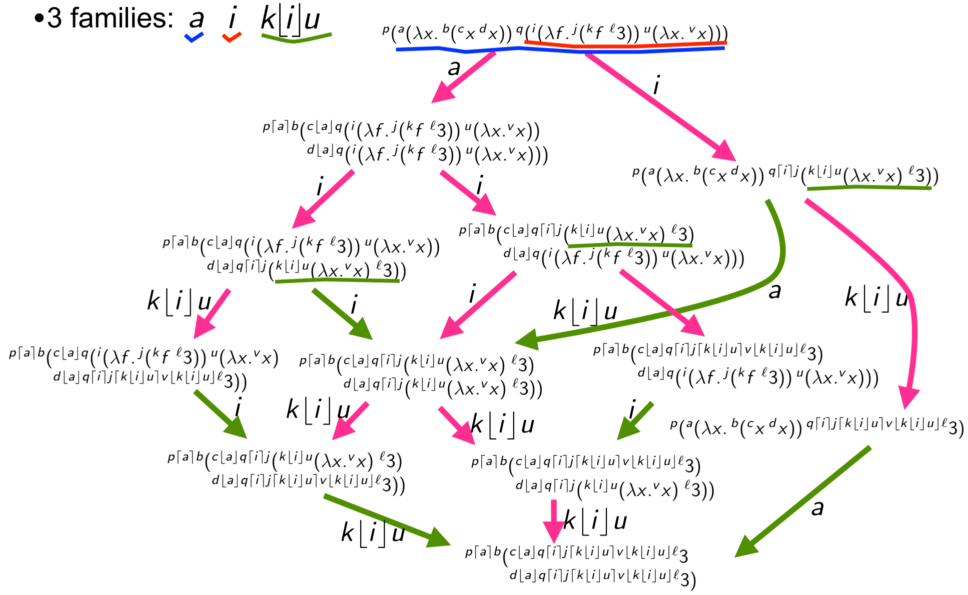


- Labels are nonempty strings of atomic labels:

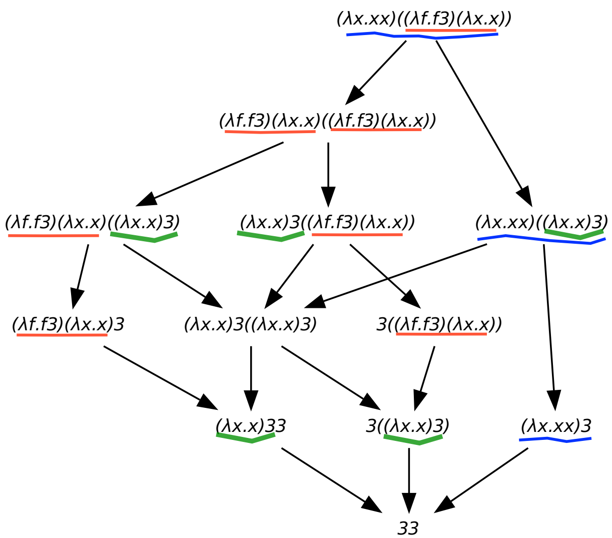
$$\alpha, \beta, \ldots \quad ::= \quad \underbrace{a, b, c, \ldots \mid \lceil\alpha\rceil \mid \lfloor\alpha\rfloor}_{\texttt{atomic labels}} \mid \alpha\beta$$

Our favorite example

- 3 families: a i $k\lfloor i\rfloor u$



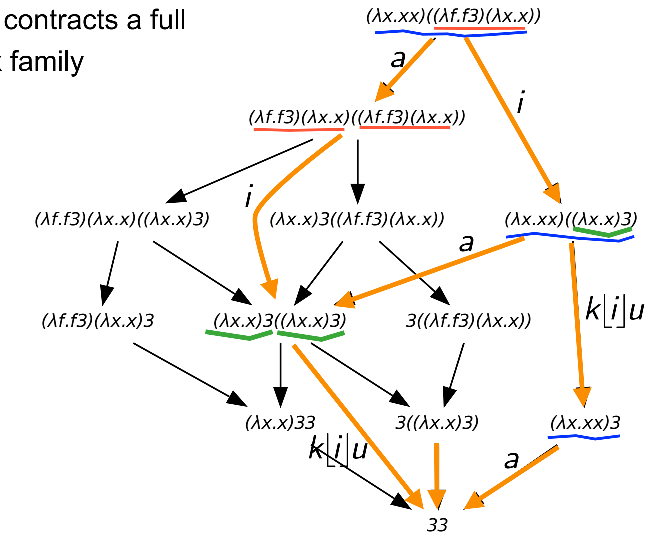
Our favorite example



- 3 redex families: **red**, **blue**, **green**.

Complete reductions (1/5)

- Each step contracts a full single redex family



Complete reductions (2/5)

- **Definition** [complete reductions]

  $\langle \rho, \mathcal{F} \rangle$ is an historical set of redexes when $\mathcal{F}$ is a set of redexes in final term of ρ.

  $\langle \rho, \mathcal{F} \rangle$ is **f-complete** when it is maximum set such that

  $$R, S \in \mathcal{F} \;\text{ implies }\; \langle \rho, R \rangle \sim \langle \rho, S \rangle$$

  An **f-complete reduction** contracts an f-complete set at each step.

- **Proposition** [lattice of f-complete reductions]

  Complete reductions form a sub-lattice of the lattice of reductions.

  Proof simple use of following lemma which implies f-complete parallel moves.

Complete reductions (3/5)

- **Notations**

  $M \stackrel{\alpha}{\Longrightarrow} N$ when $M \stackrel{\mathcal{F}}{\longrightarrow} N$ and $\mathcal{F}$ is the set of redexes with name α in M.

  MaxRedNames(M) when all redexes in M have maximal names.

- **Lemma** [complete reductions preserve max redex names]

  $M \stackrel{\alpha}{\Longrightarrow} N$ and $\text{MaxRedNames}(M)$ implies $\text{MaxRedNames}(N)$

Complete reductions (4/5)

- **Definition** [d-complete reductions]

  $\langle \rho, \mathcal{F} \rangle$ is **d-complete** when it is maximum set such that

  $$\langle \rho_0, R_0 \rangle \leq \langle \rho, \mathcal{F} \rangle \text{ for some } \langle \rho_0, R_0 \rangle$$

  An **d-complete reduction** contracts a d-complete set at each step.

- **Proposition** [below canonical representative]

  Let $\langle \rho_0, R_0 \rangle$ be canonical representative in its family.
  Let $\rho_0 \sqsubseteq \rho$. Then $\langle \rho_0, R_0 \rangle \sim \langle \rho, R \rangle$ iff $\langle \rho_0, R_0 \rangle \leq \langle \rho, R \rangle$.

  Proof difficult.

- **Proposition** [f-complete = d-complete]

  d-complete reductions coincide with f-complete reductions.

Complete reductions (5/5)

- **Proposition** [length of reduction = number of families]

  In complete reductions, number of steps equals the number of contracted redex families.

  Proof application of MaxRedNames lemma.

- **Corollary** [optimal reductions]

  In complete reductions, never redex of same family is contracted twice.

- **Implementation** [optimal reductions]

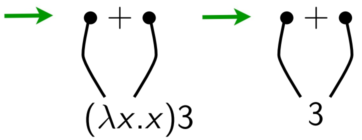
  Can we implement efficiently complete reductions ?

Implementation (1/5)

- **Implementation** `[optimal reductions]` algorithm `[John Lamping, 90 -- Gonthier-Abadi-JJ, 91]`

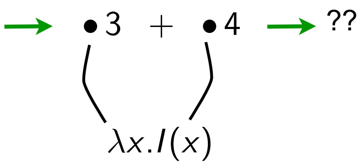
- Sharing of basic values is easy:

$(\lambda x.x + x)((\lambda x.x)3)$ →

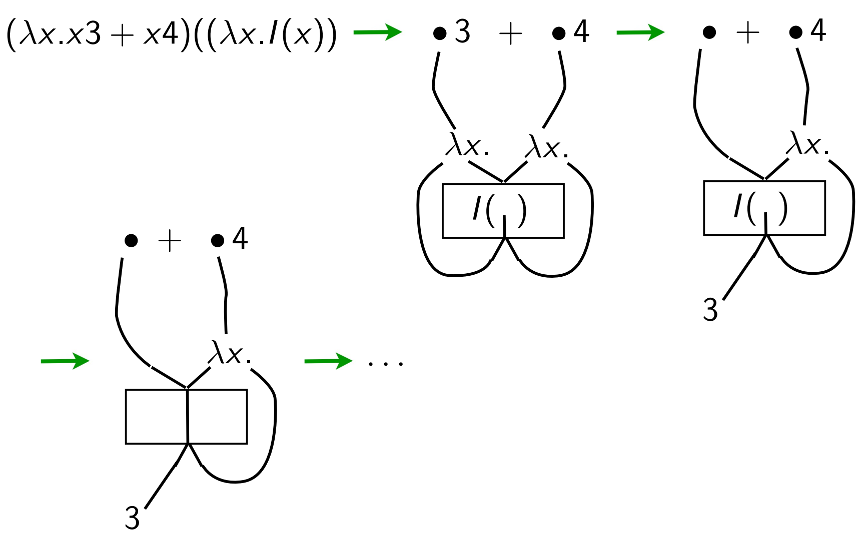


- Problem is sharing of functions:

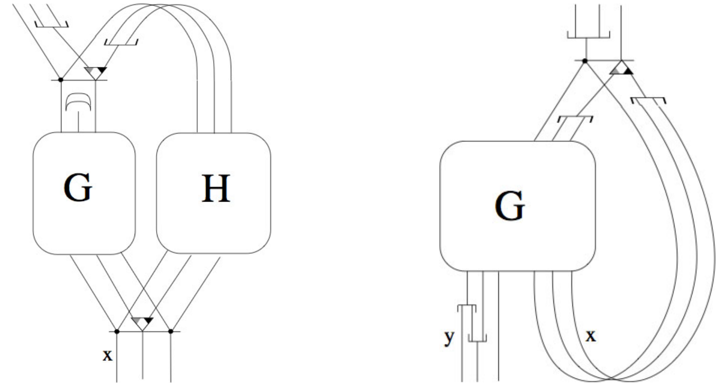
$(\lambda x.x3 + x4)((\lambda x.I(x))$ →



Implementation (2/5)



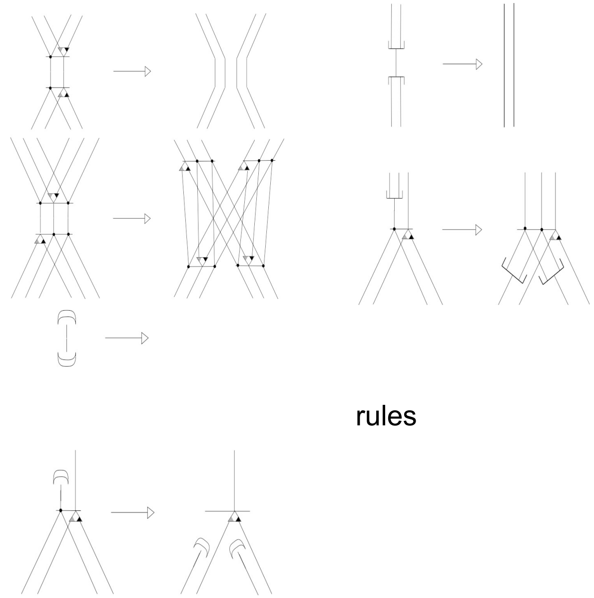
Implementation (3/5)

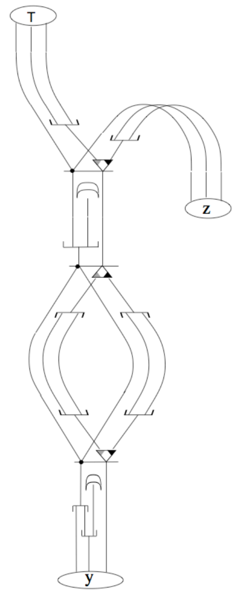


application

λ-abstraction

Implementation (4/5)





$(\lambda x.yx)z$

Implementation (5/5)

- beautiful Lamping's algorithm is unpractical
- highly exponential in the handling of fans node (not elementary recursive) [Asperti, Mairson 2000]
- nice algorithms unsharing paths to bound variables [Wadsworth 92, Shivers-Wand 2010]
- Haskell, Coq, Caml ??

Permutations in Call by value

- **Definition** [call by value]

  A **value** remains a value if computed or substituted by a value

$$V \ ::= \ x \ | \ \lambda x.M$$

The call-by-value reduction strategy is defined by:

$$(\lambda x.M)V \xrightarrow[\text{cbv}]{} M\{x := V\}$$

$$\frac{M \xrightarrow[\text{cbv}]{} M'}{MN \xrightarrow[\text{cbv}]{} M'N} \qquad \frac{N \xrightarrow[\text{cbv}]{} N'}{MN \xrightarrow[\text{cbv}]{} MN'}$$

- **Fact** [permutations in call by value]

  Equivalence by permutations only permute disjoint redexes.

Speculative computations

CENTRE DE RECHERCHE COMMUN

INRIA MICROSOFT RESEARCH

Speculative computations

- **Definition** [speculative call, Boudol-Petri 2010]

$$V \ ::= \ x \ | \ \lambda x.M$$

The speculative reduction strategy is defined by:

$$(\lambda x.M)V \xrightarrow[\text{spec}]{} M\{x := V\}$$

$$(\lambda x.M)N \xrightarrow[\text{spec}]{} (\lambda V?\, M\{x := V\})N$$

$$(\lambda V?\, M)V \xrightarrow[\text{spec}]{} M$$

$$\frac{M \xrightarrow[\text{spec}]{} M'}{MN \xrightarrow[\text{spec}]{} M'N} \qquad \frac{N \xrightarrow[\text{spec}]{} N'}{MN \xrightarrow[\text{spec}]{} MN'} \qquad \frac{M \xrightarrow[\text{spec}]{} M'}{\lambda V?M \xrightarrow[\text{spec}]{} \lambda V?M'}$$

Events structure

CENTRE DE RECHERCHE COMMUN

INRIA
MICROSOFT RESEARCH

Creation of redexes (1/3)

- Causal order between redex families

  In favorite example:

$$\begin{tikzcd} a & i \arrow[d, "\text{creates}"] \\ & k\boxed{i}u \end{tikzcd}$$

- Several redexes can create another one

$$\underline{(\lambda f.f3)(\underline{(\lambda x.\lambda y.4)5})} \longrightarrow (\lambda y.4)3$$

$$\begin{tikzcd} a \arrow[dr] & & i \arrow[dl, "\text{creates}"'] \\ & k\boxed{i}u & \end{tikzcd}$$