Labels, Paths and Linearization of the λ-calculus

Sandra Alves, **Pedro Cunha**, Mário Florido

19 July 2026

Context

Linear term

Every λ binds **exactly** one variable.

- Kfoury
  - New calculus, $\Lambda^{\wedge}$
  - Expansion/Contraction allow for interaction
  - "... enforce the linearity condition on function evaluation..."
- Damas and Florido
  - Closed for the λ-calculus
  - Uses Intersection Types (hard to type with)
  - Syntactic: $\delta = \lambda x.xx \to \lambda x_1 x_2.x_1 x_2$
- Alves and Florido
  - Also closed for λ-calculus
  - Uses Legal Paths (requires all redexes)
  - "Semantic": $\lambda x.xx \to \lambda x.xx$, but $(\lambda x.xx)I \to (\lambda x_1 x_2.x_1 x_2)II$

Proposal of the Paper

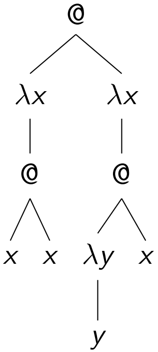
Parametrized Weak Linearization

A (weak) linearization parametrized by redexes, based on paths of the term.

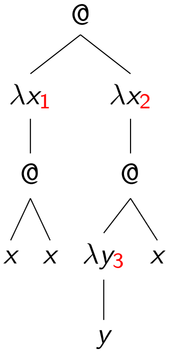
We conjecture that it subsumes the last two.

Labels, Paths and Redexes

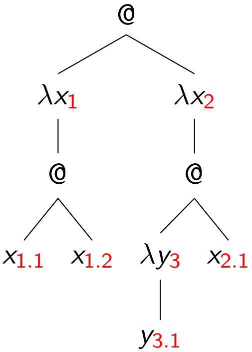
$$(\lambda x.xx)\,(\lambda x.(\lambda y.y)x)$$



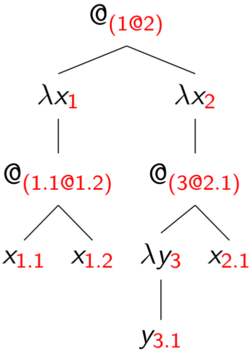
Labels, Paths and Redexes



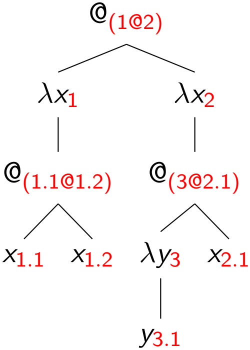
Labels, Paths and Redexes



Labels, Paths and Redexes

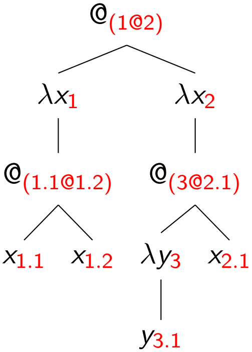


Labels, Paths and Redexes



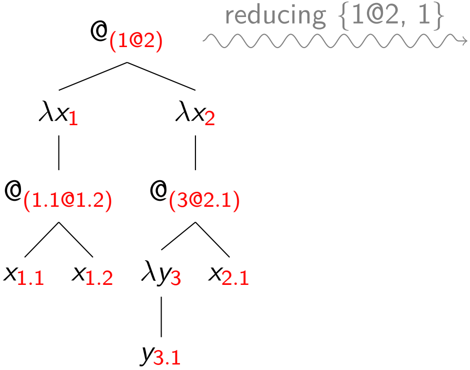
Paths: $1@2, 1$; $1.1@1.2, 1, 1@2, 2,$ ...

Labels, Paths and Redexes



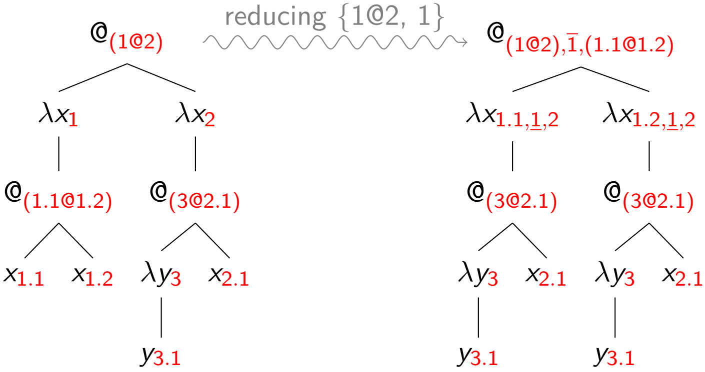
(Direct) Redexes: $1@2, 1$ and $3@2.1, 3$, ...

Labels, Paths and Redexes



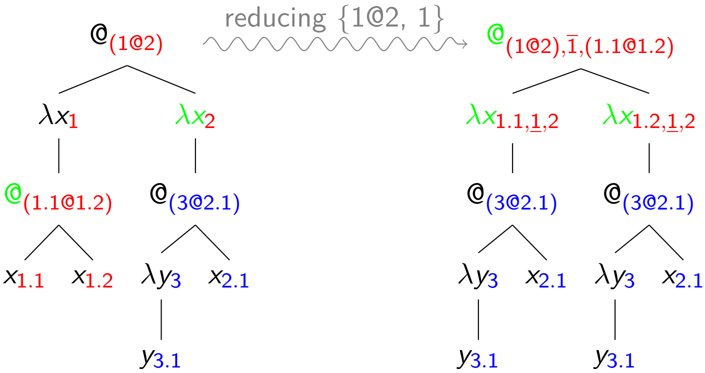
N.B.: these concepts (as well as legal paths) were introduced by Lévy and Asperti.

Labels, Paths and Redexes

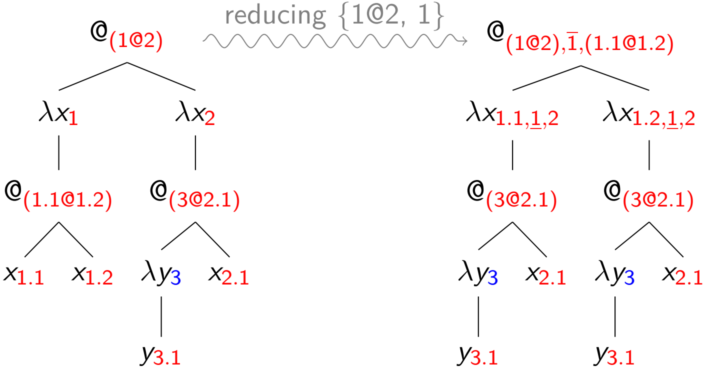


N.B.: these concepts (as well as legal paths) were introduced by Lévy and Asperti.

Labels, Paths and Redexes

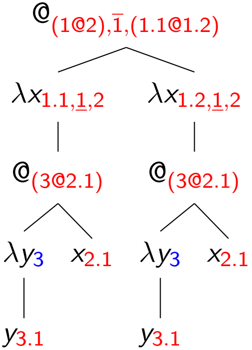


Labels, Paths and Redexes



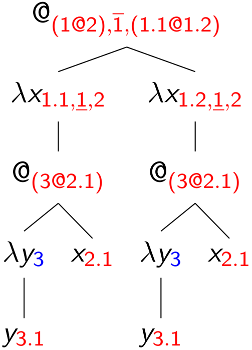
Labels, Paths and Redexes

Problem: how can we distinguish the two redexes with the same degree (i.e. label of the λ-node)?



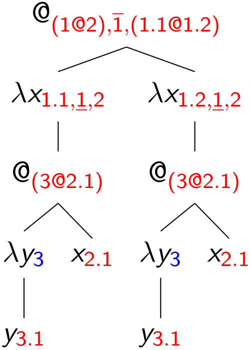
Labels, Paths and Redexes

Use Label Paths: the shortest path from the root to the node.



Labels, Paths and Redexes

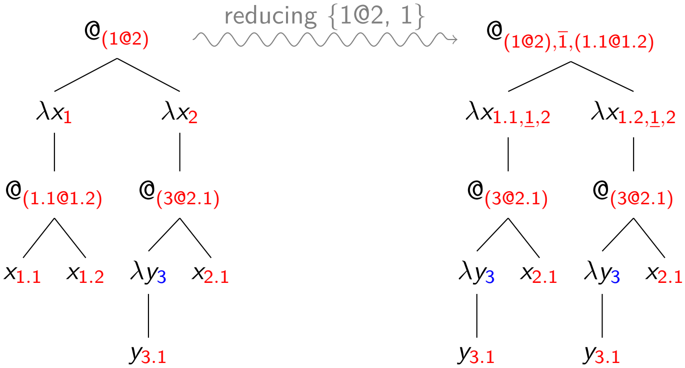
Use Label Paths: the shortest path from the root to the node.



$$r_1 = 1@2, \bar{1}, 1.1@1.2; 1.1, \underline{1}, 2; 3@2.1, 3$$
$$r_2 = 1@2, \bar{1}, 1.1@1.2; 1.2, \underline{1}, 2; 3@2.1, 3$$

Computation: Reduction Sequences

A sequence of redexes $r_1, \ldots, r_n$ of a term T is a reduction sequence if r_i is a direct redex in the tree obtained by reducing $r_1, \ldots, r_{i-1}$.



$r_1 = 1@2 \,|\, 1$; $r_2 = 1@2, \overline{1}, 1.1@1.2 \,|\, 1.1, \underline{1}, 2$ is a reduction sequence.
$r_1 = 1@2 \,|\, 1$; $r_2 = 1@2, 2, 3@2.1 \,|\, 3$ is **not** a reduction sequence.

Redexes: Summary

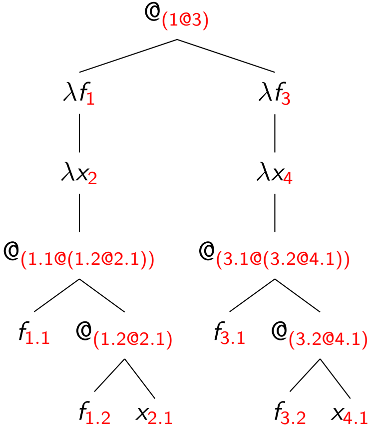
- We label the term (taken as tree)
- Reducing redexes changes these labels
- We consider paths from the root('s label) to a given node('s label)
- A redex is a pair with the (label) path of the @-node and the λ-node
- A computation is a sequence of redexes that "makes sense"

Linearization: overview

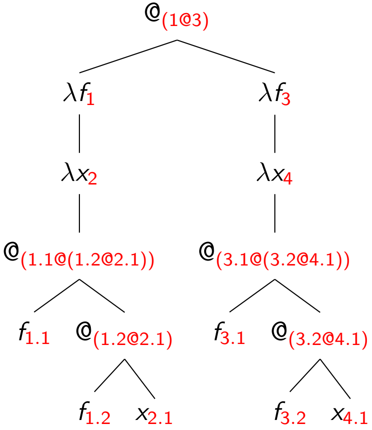
- Two core functions: `expand` and `balance`
- `expand` copies, for a given redex, the λ-node (and dual `@`-node) for each bound variable of the node, binding these to the new nodes (one-to-one)*
  - Thus, the redex becomes linear
- `balance` guarantees** that, after several applications of `expand`, we still have linearity in the redexes previously `expanded`
- We apply `expand` to the reduction sequence and then apply `balance`, which reapplies `expand` until everything is linear

Linearization: `expand`

Consider the term $T = 2\,2 = (\lambda fx.f(fx))(\lambda fx.f(fx))$:



Linearization: `expand`



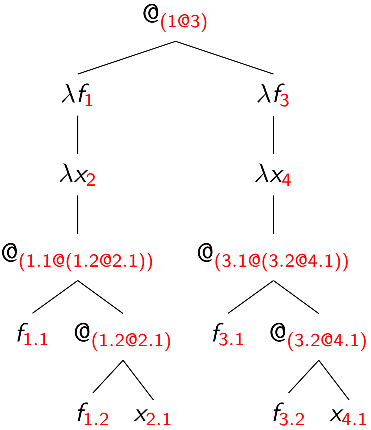
We'll expand it for the reduction sequence

$$r_1 = 1@3 \mid 1$$

$$r_2 = 1@3, \overline{1}, 2; 1.1@(1.2@2.1) \mid 1.1, \underline{1}, 3$$

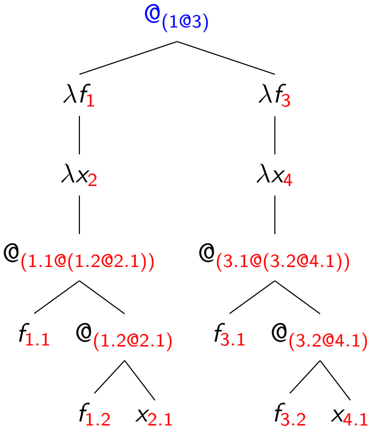
Linearization: `expand`

$$\texttt{expand}(1@3 \mid 1, T)$$



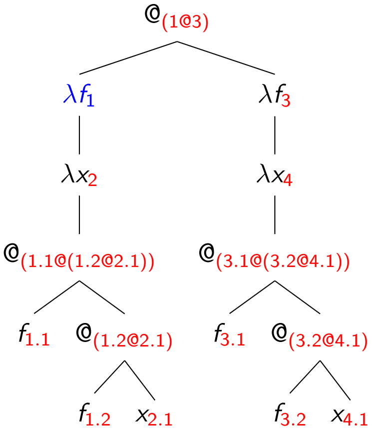
Linearization: `expand`

$$\texttt{expand}(1@3 \mid 1, T)$$



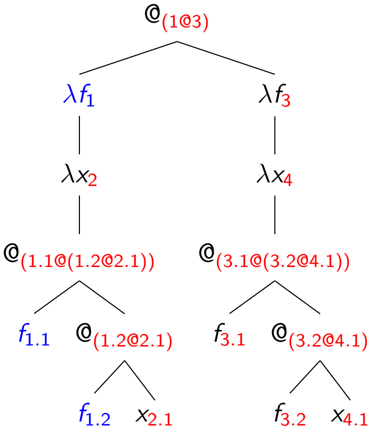
Linearization: `expand`

$$\texttt{expand}(1@3 \mid 1, T)$$



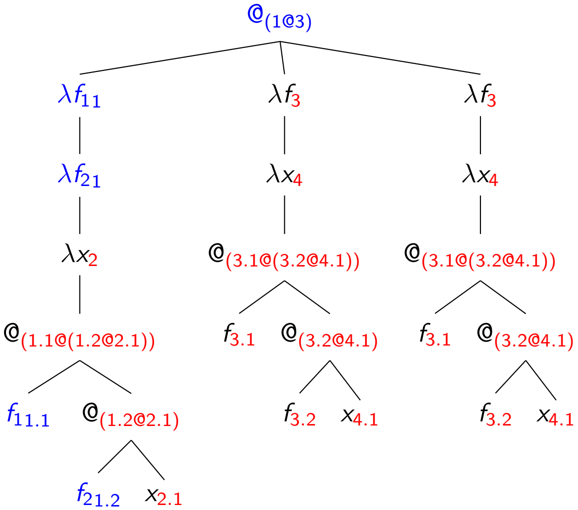
Linearization: `expand`

$$\texttt{expand}(1@3 \mid 1, T)$$



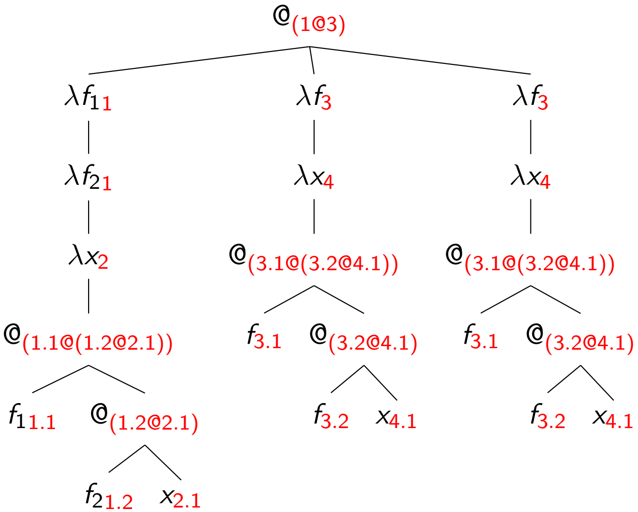
Linearization: `expand`

$\texttt{expand}(1@3 \mid 1, T)$



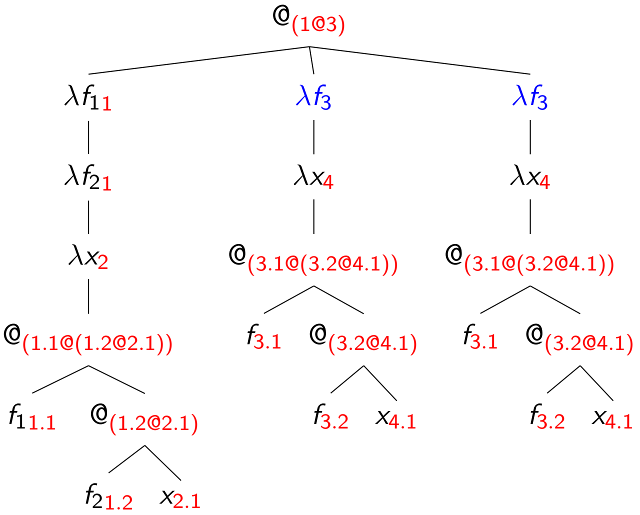
Linearization: `expand`

$$\texttt{expand}(1@3, \overline{1}, 2; 1.1@(1.2@2.1) \mid 1.1, \underline{1}, 3, T)$$



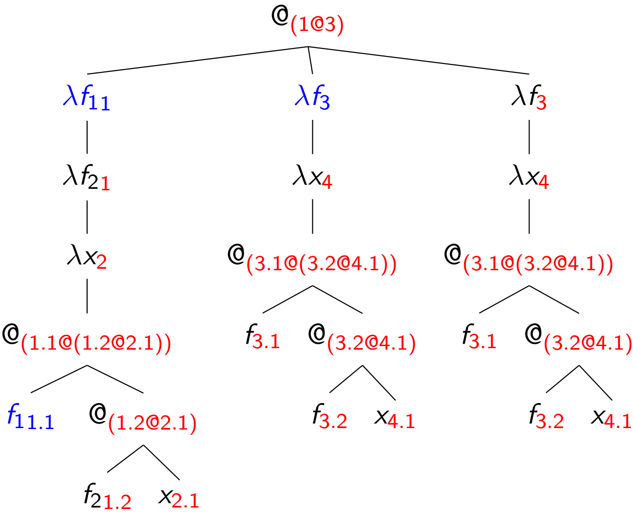
Linearization: `expand`

$$\texttt{expand}(1@3, \overline{1}, 2; 1.1@(1.2@2.1) \mid 1.1, \underline{1}, 3, T)$$



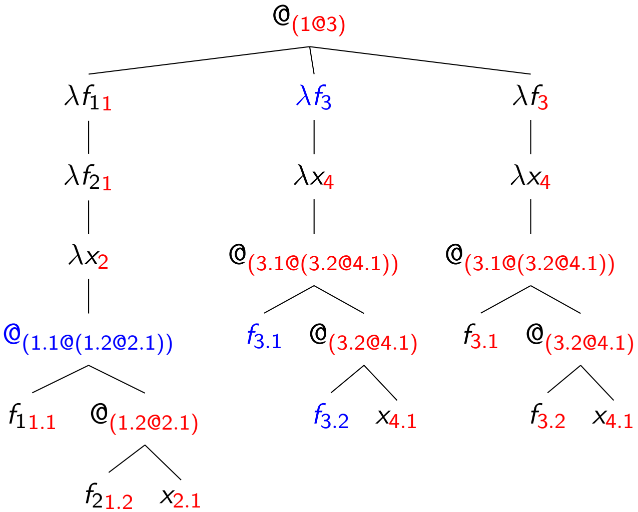
Linearization: `expand`

$$\texttt{expand}(1@3, \overline{1}, 2; 1.1@(1.2@2.1) \mid 1.1, \underline{1}, 3, T)$$



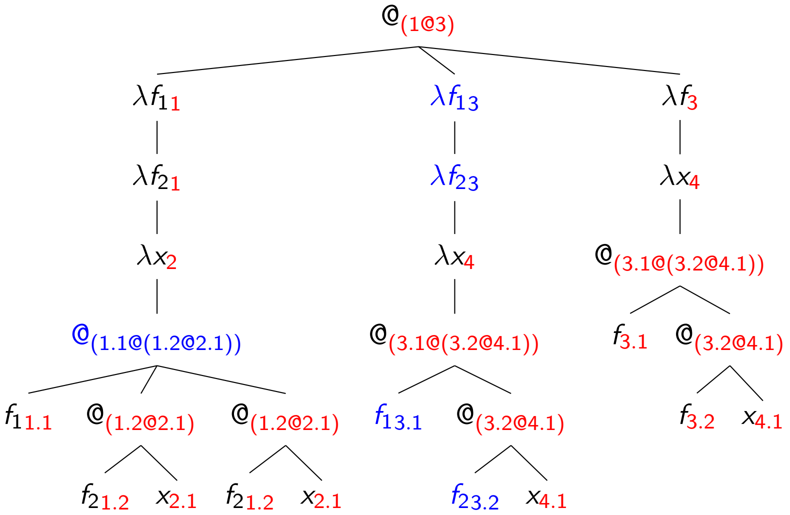
Linearization: `expand`

$$\texttt{expand}(1@3, \overline{1}, 2; 1.1@(1.2@2.1) \mid 1.1, \underline{1}, 3, T)$$



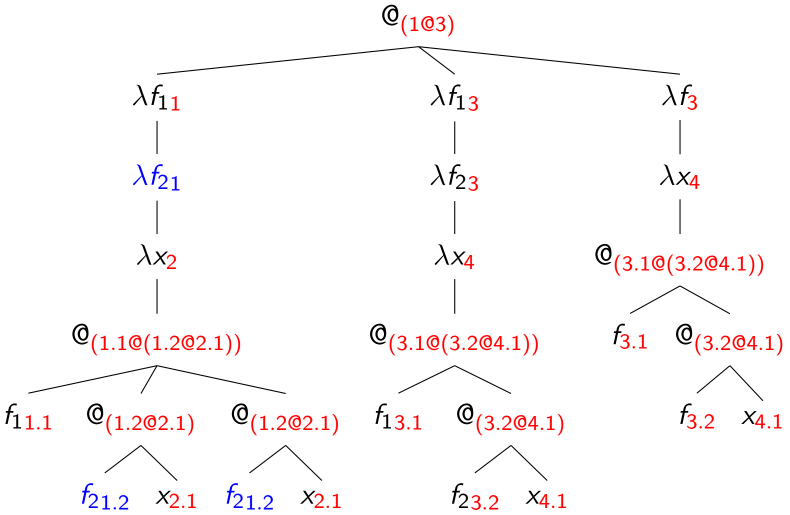
Linearization: `expand`

$$\texttt{expand}(1@3, \overline{1}, 2; 1.1@(1.2@2.1) \mid 1.1, \underline{1}, 3, T)$$



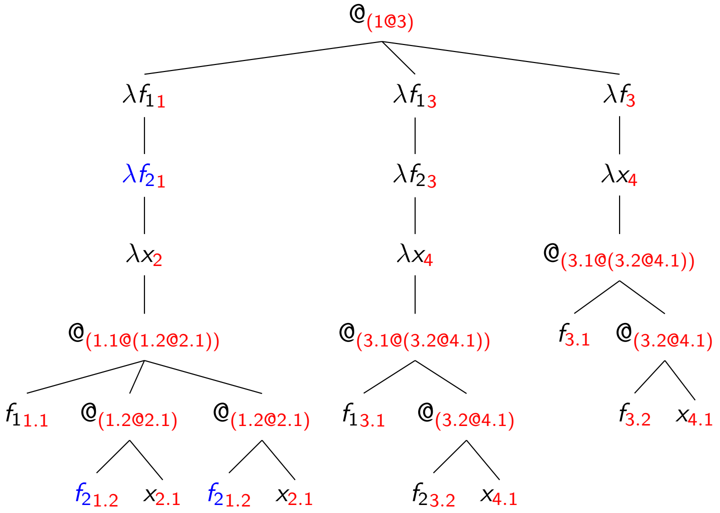
Linearization: `expand`

$$\texttt{expand}(1@3, \overline{1}, 2; 1.1@(1.2@2.1) \mid 1.1, \underline{1}, 3, T)$$

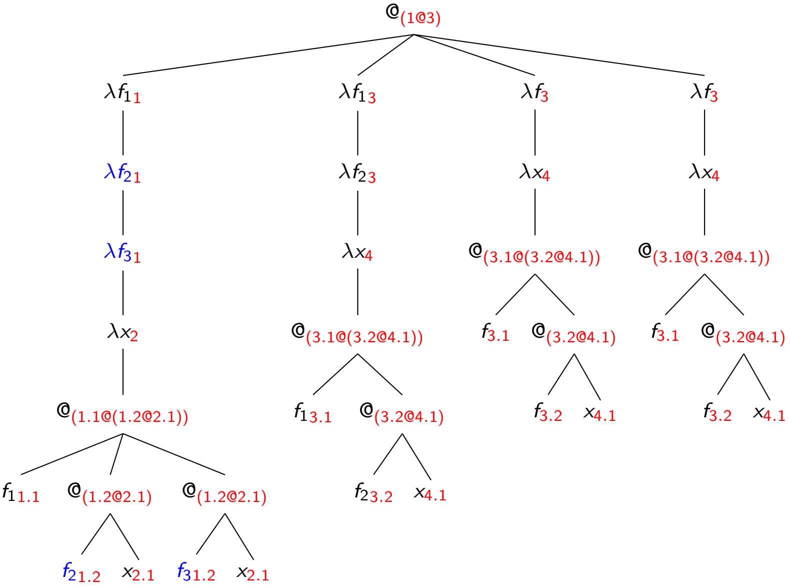


Linearization: `balance`

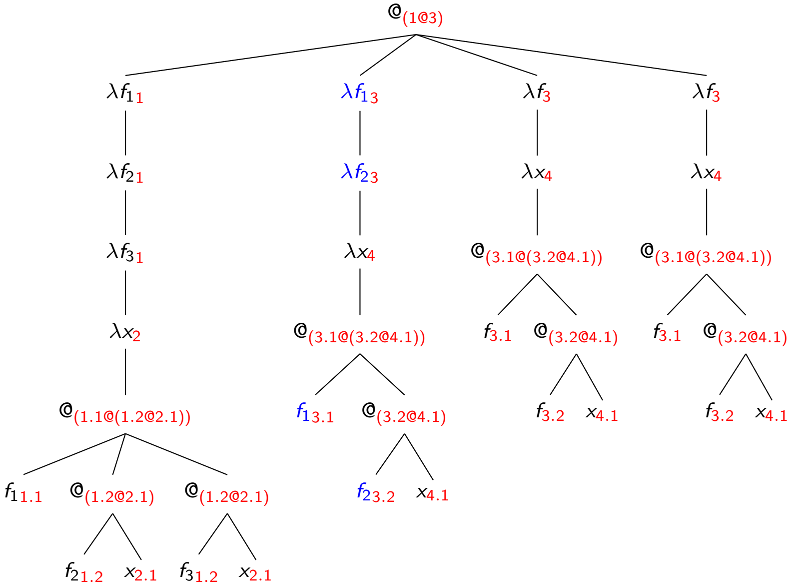
$$\texttt{balance}(T, [r_1, r_2]) = \texttt{balance}(\texttt{expand}(r_2, \texttt{expand}(r_1, T)), [r_1, r_2])$$



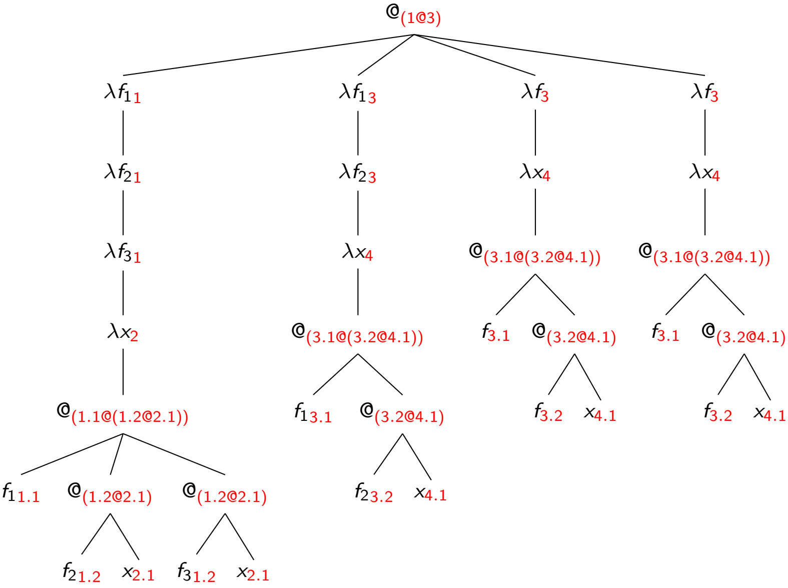
Linearization: `balance`



Linearization: `balance`



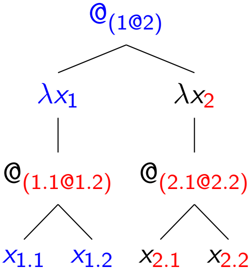
Linearization: `balance`



Linearization of non-terminating terms

Linearizing $\Delta = \delta\,\delta$ for two redexes:

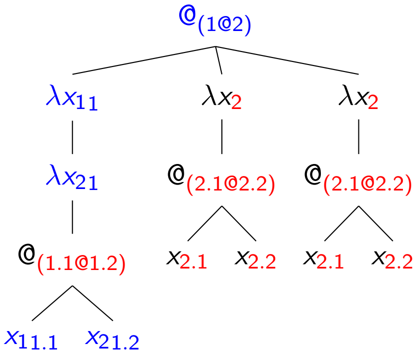
$$\texttt{expand}(1@2 \,|\, 1, T)$$



Linearization of non-terminating terms

Linearizing $\Delta = \delta\,\delta$ for two redexes:

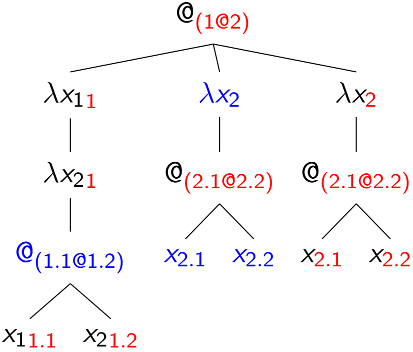
$$\texttt{expand}(1@2 \,|\, 1, T)$$



Linearization of non-terminating terms

Linearizing $\Delta = \delta\,\delta$ for two redexes:

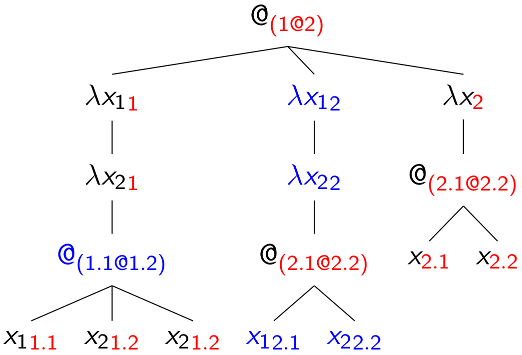
$$\texttt{expand}(1@2, \overline{1}, 1.1@1.2 \,|\, 1.1, \underline{1}, 2; \, T)$$



Linearization of non-terminating terms

Linearizing $\Delta = \delta\,\delta$ for two redexes:

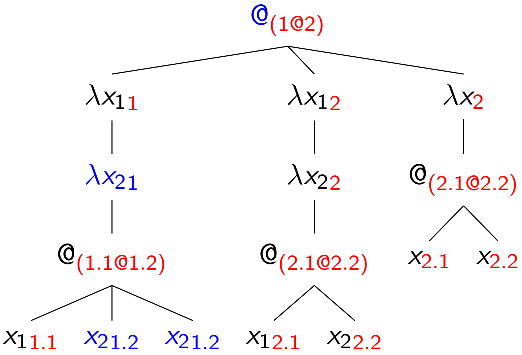
$$\texttt{expand}(1@2, \overline{1}, 1.1@1.2 \mid 1.1, \underline{1}, 2;\, T)$$



Linearization of non-terminating terms

Linearizing $\Delta = \delta\,\delta$ for two redexes:

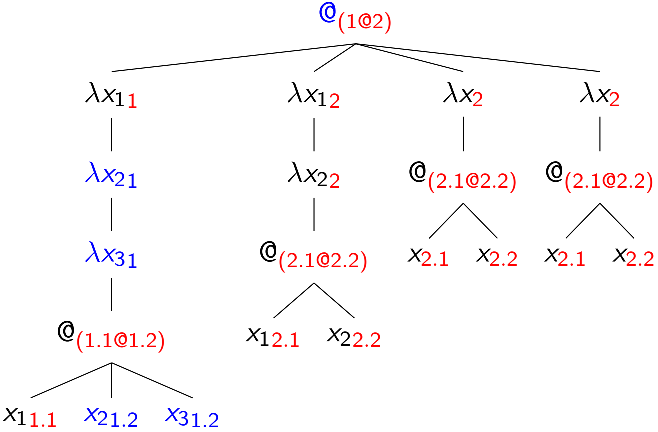
$$\texttt{balance}(T, [r_1, r_2])$$



Linearization of non-terminating terms

Linearizing $\Delta = \delta\,\delta$ for two redexes:

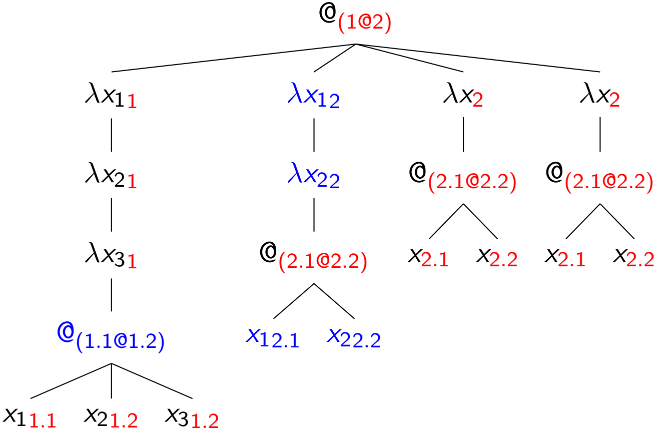
$$\texttt{balance}(T, [r_1, r_2])$$



Linearization of non-terminating terms

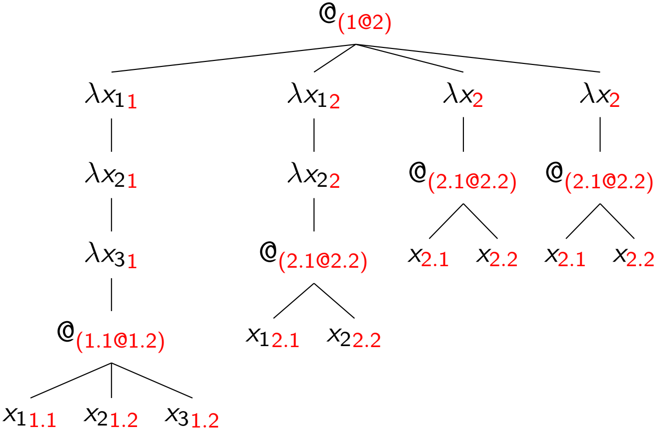
Linearizing $\Delta = \delta\,\delta$ for two redexes:

$$\texttt{balance}(T, [r_1, r_2])$$



Linearization of non-terminating terms

Linearizing $\Delta = \delta\,\delta$ for two redexes:



Concluding Remarks

Pros

- We conjecture that...
  - `balance` terminates + the linearization is correct
  - if parametrized by the maximal strategy (for a SN term), this is the same as Alves and Florido's linearization
  - plus, if we linearize every non-linear term that is leftover, this is the same as Damas and Florido's linearization
- Works for WN terms (and non-terminating terms in general)

Cons

- Work in progress!
- Finding redexes can be complicated
- Linearization is a complex process

Thank you!