



Finite Developments in the Lambda Calculus

Jean-Jacques Levy

► To cite this version:

Jean-Jacques Levy. Finite Developments in the Lambda Calculus. Doctoral. Spain. 2021. hal-03566512

HAL Id: hal-03566512

<https://inria.hal.science/hal-03566512>

Submitted on 11 Feb 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Finite Developments in the λ -calculus

Part I

jean-jacques.levy@inria.fr

ISR 2021

Madrid

July 6, 2021



<http://jeanjacqueslevy.net/talks/21isr>



λ -calculus

function

λ -term

β -reduction

$$I\ x = x$$

$$I = \lambda x.x$$

$$I\ a \longrightarrow a$$

$$K\ x\ y = x$$

$$K = \lambda x.\lambda y.x$$

$$K\ a\ b \longrightarrow (\lambda y.a)\ b \longrightarrow a$$

$$\Delta\ x = x\ x$$

$$\Delta = \lambda x.x\ x$$

$$\Delta\ a \longrightarrow a\ a$$

$$\Omega = \Delta\ \Delta$$

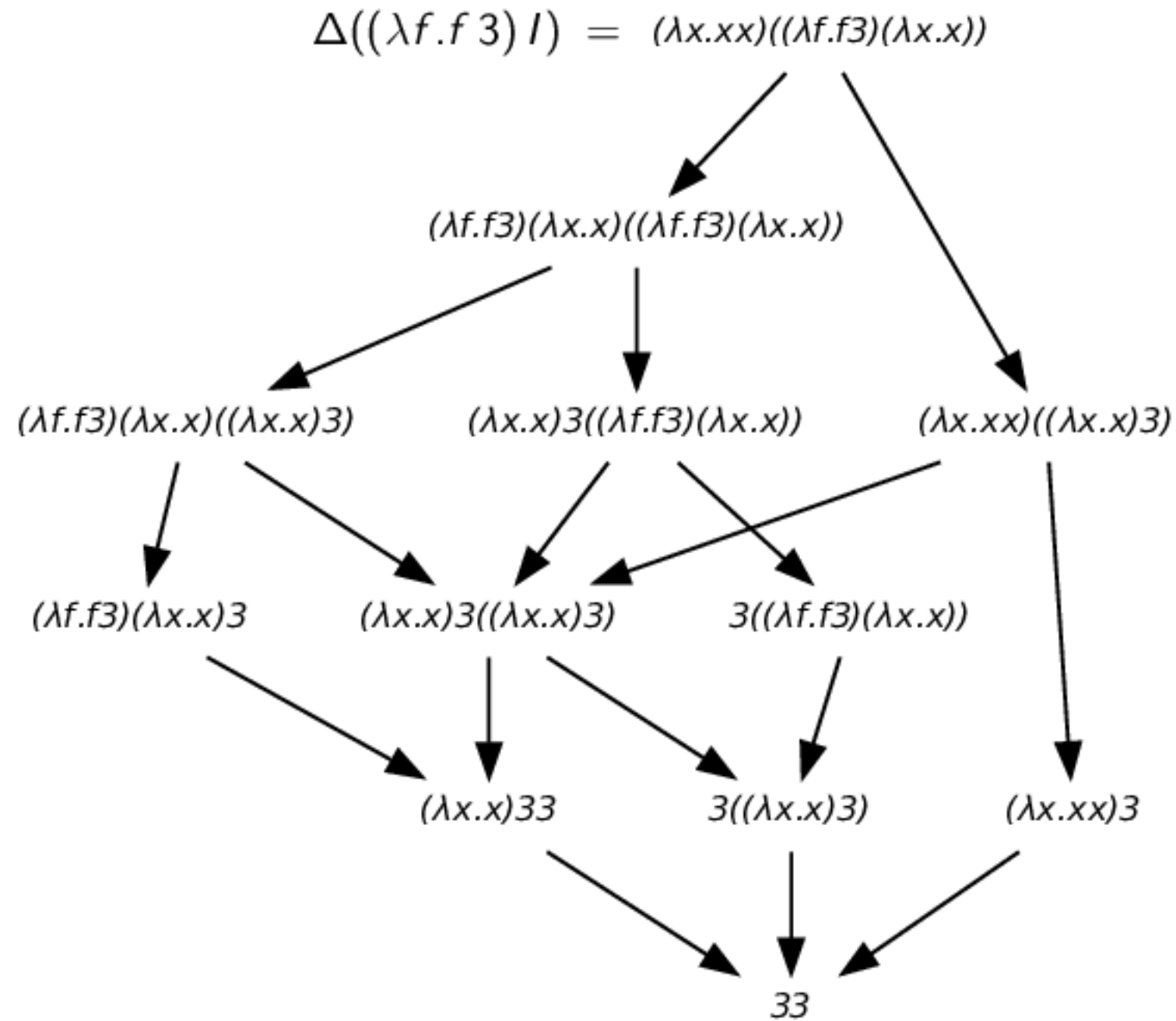
$$\Omega \longrightarrow \Omega$$

Exercise 1

$$\Delta(\lambda x.x\ x\ x) \longrightarrow \dots$$

$$Y_f = (\lambda x.f\ (x\ x))(\lambda x.f\ (x\ x)) \longrightarrow \dots$$

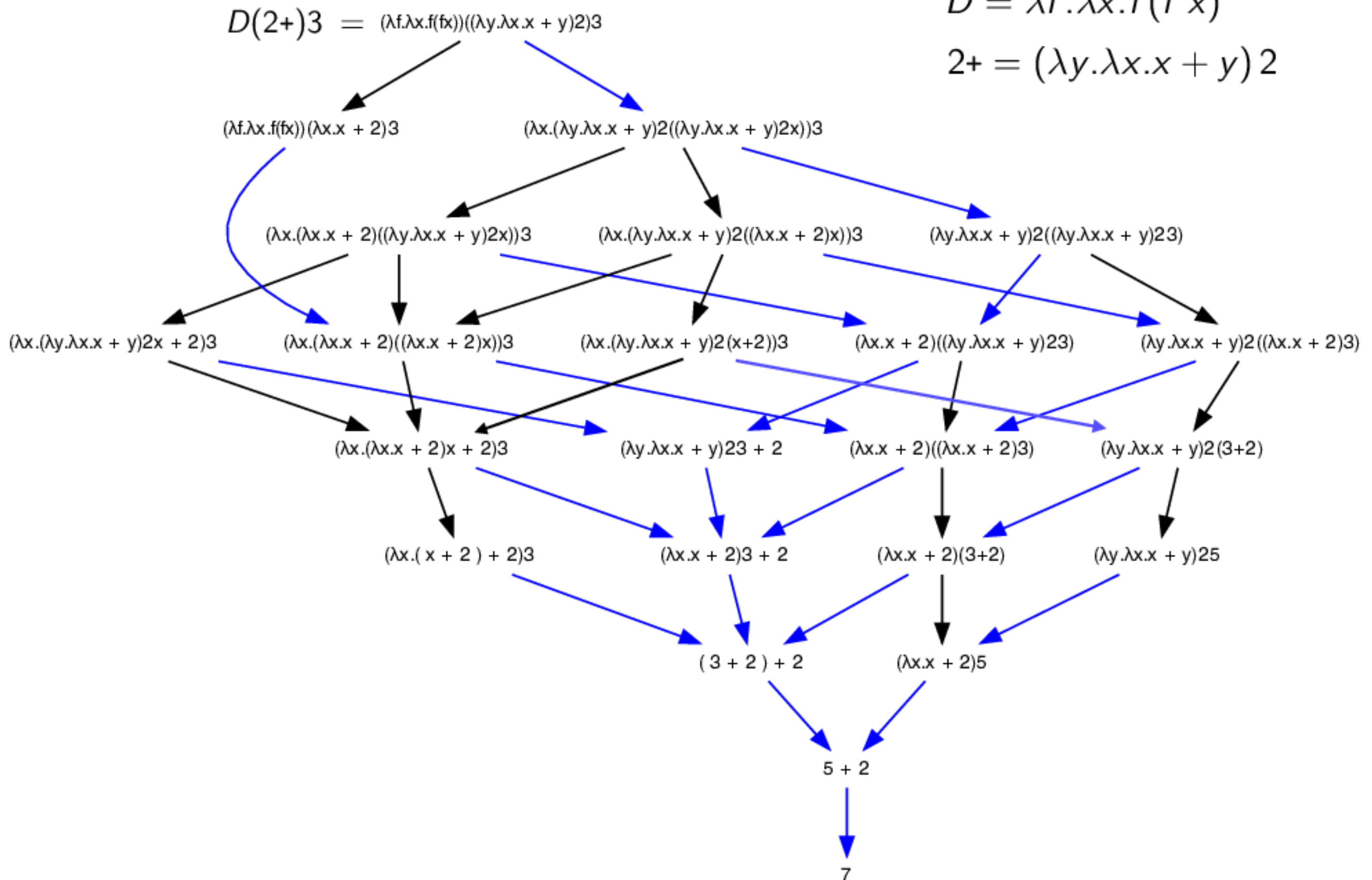
λ -calculus



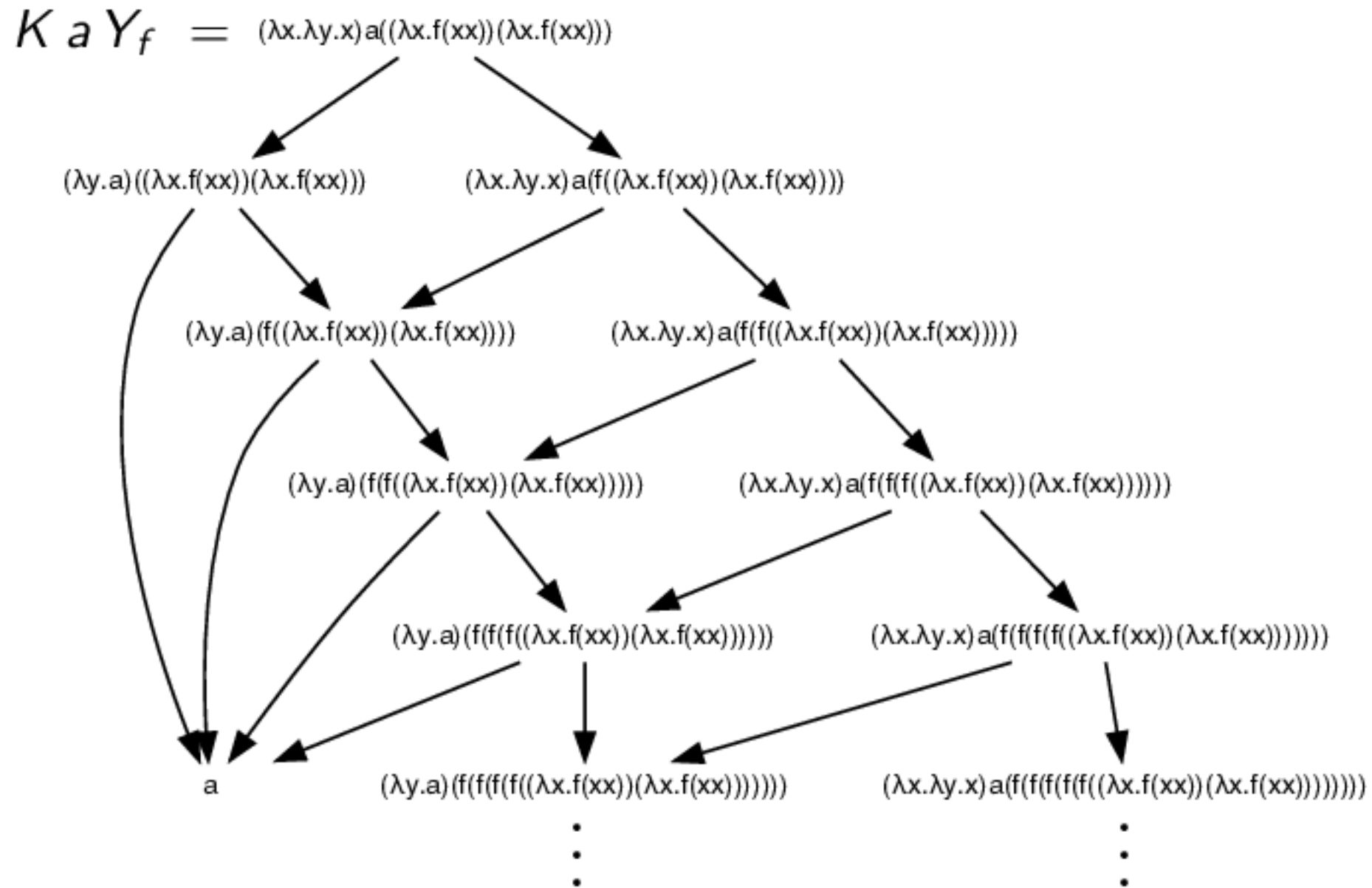
λ -calculus

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

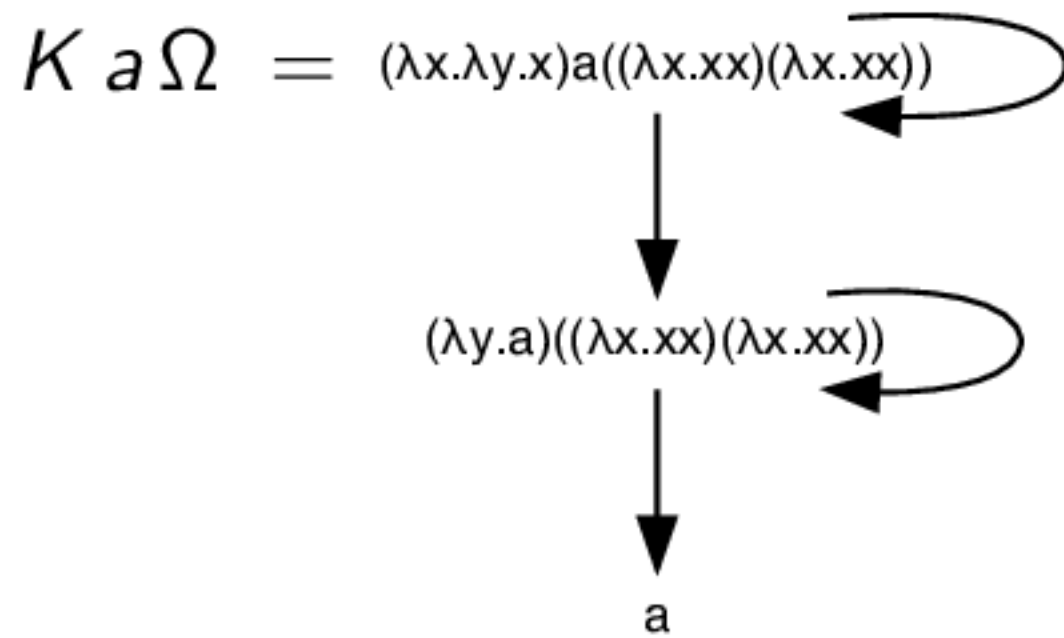
$$2+ = (\lambda y. \lambda x. x + y)\ 2$$



λ -calculus



λ -calculus



Empirical facts

- **deterministic** result when it exists
- multiple reduction strategies
- **terminating** strategy ?
- **efficient** reduction strategy ?
- **worst** reduction strategy ?
- when all reductions are finite ?
- the reduction graph has a **lattice** structure ?

Church-Rosser

CBN - CBV - ..

normalisation

optimal reduction

perpetual reduction

strong normalisation

NO!

Redexes

- a **redex** is any **re**ductible **ex**pression: $(\lambda x.M)N$
- the **β -conversion** rule is:

$$(\lambda x.M)N \longrightarrow M\{x := N\}$$

- a **reduction step** contracts a given redex $R = (\lambda x.A)B$ and is written: $M \xrightarrow{R} N$

- a reduction step contracts a **singleton** set of redexes $M \xrightarrow{\{R\}} N$

- a more precise notation would be with occurrences of subterms. We avoid it here (but it is sometimes mandatory to avoid ambiguity)
- we replaced occurrences by giving names (labels) to redexes.

Bound variables

$$(\lambda x.x (\lambda y.x y))y = (\lambda x.x (\lambda z.x z))y$$



$$y (\lambda z.y z)$$

- names of bound variables are not important
- we consider λ -terms up-to renaming of bound variables (**α -conversion**)
- free variables of M are formally defined by:

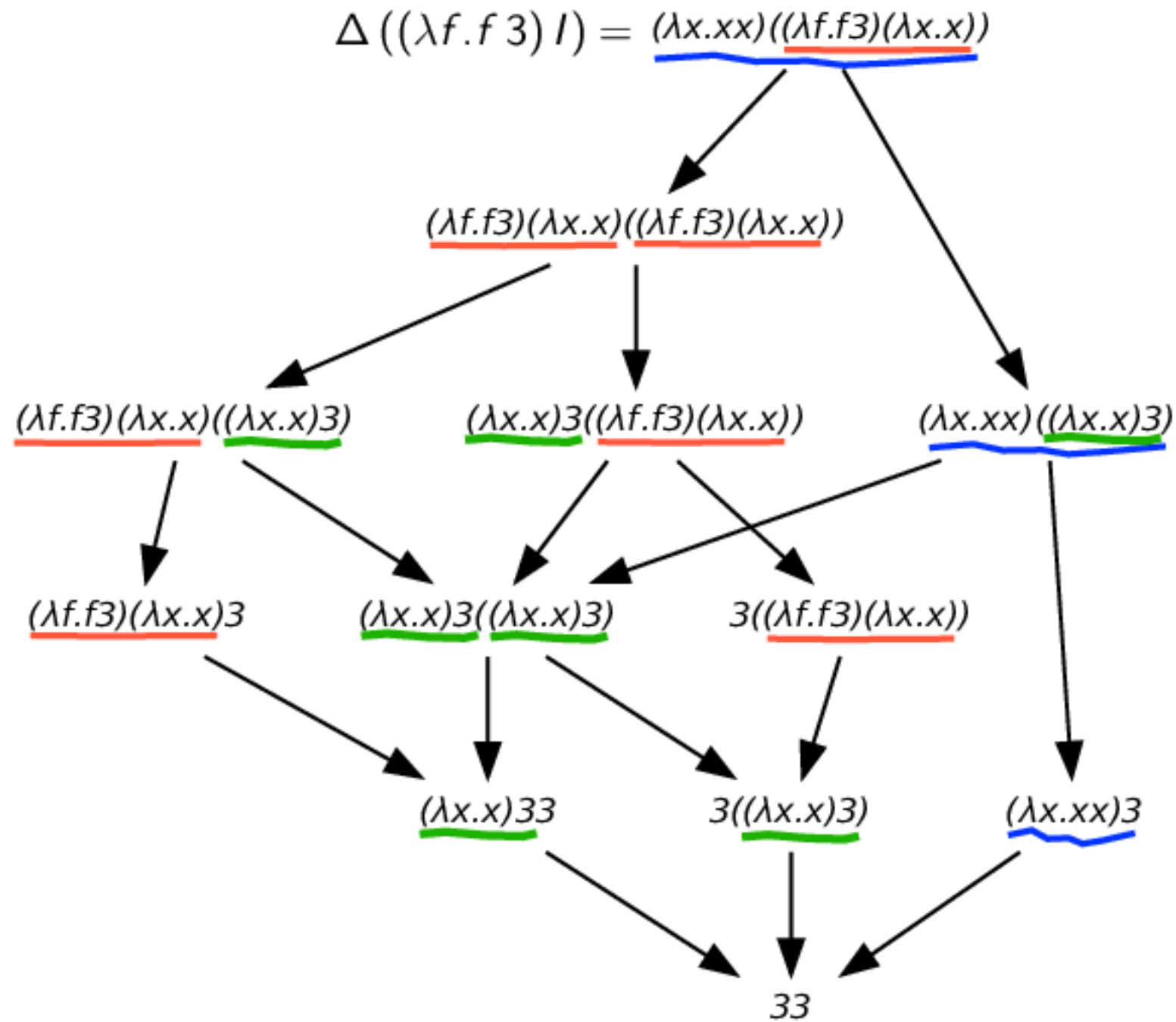
$$FV(x) = \{x\}$$

$$FV(\lambda x.M) = FV(M) - \{x\}$$

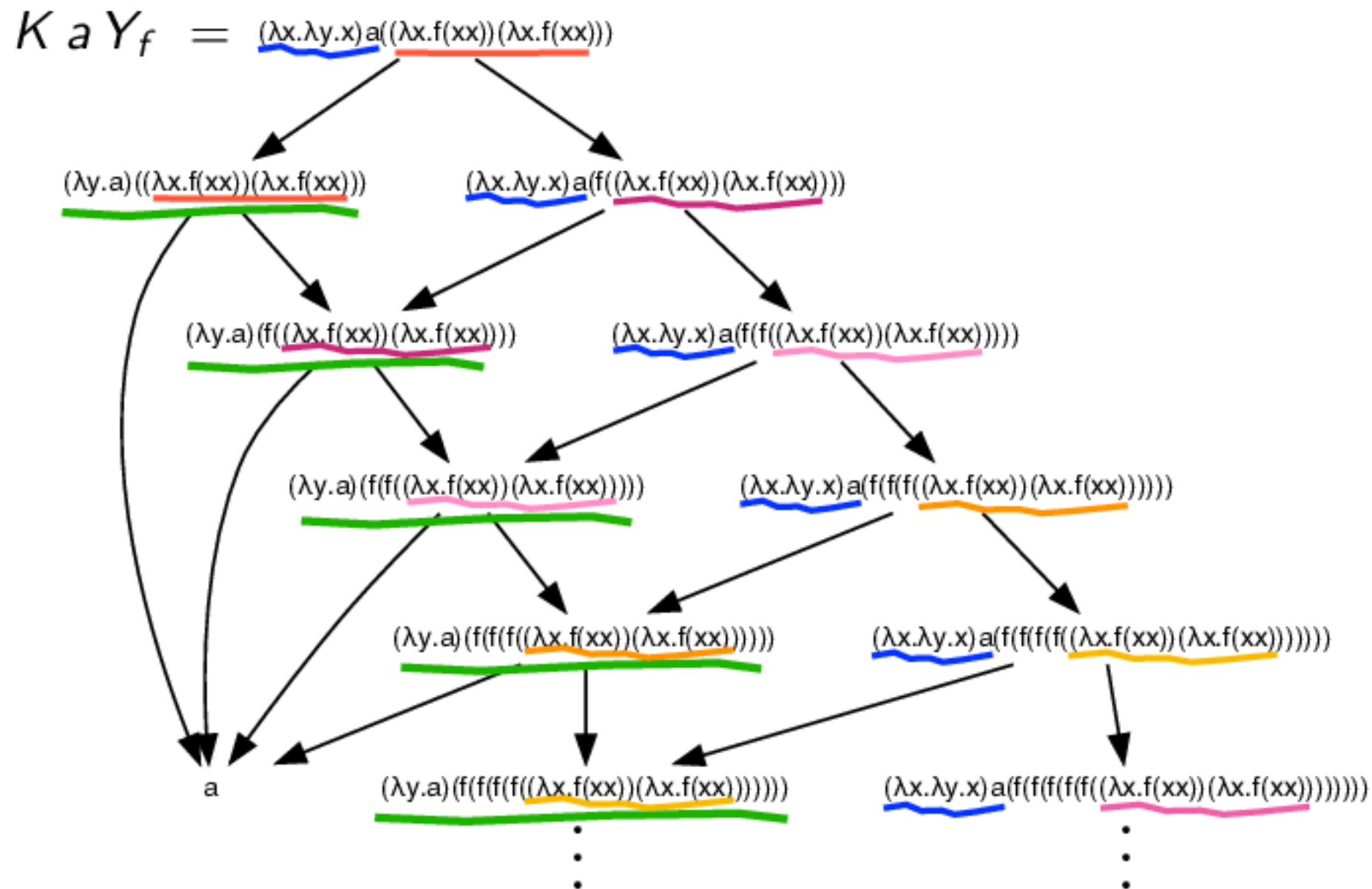
$$FV(MN) = FV(M) \cup FV(N)$$

forget α -conversion

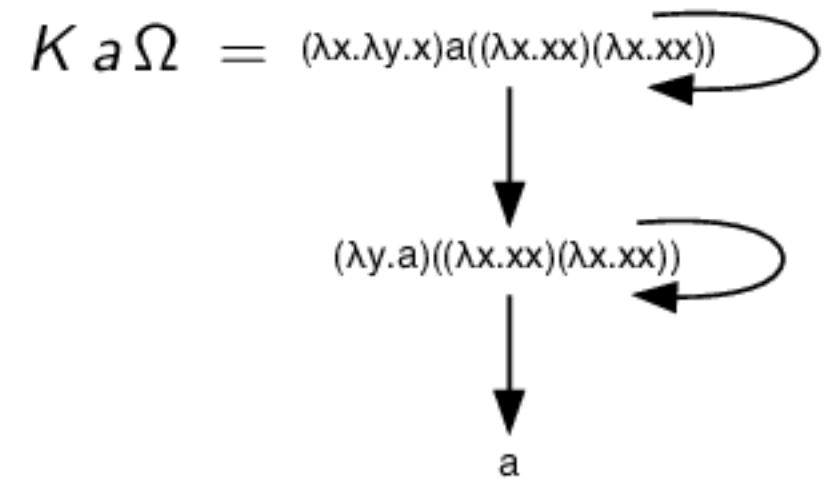
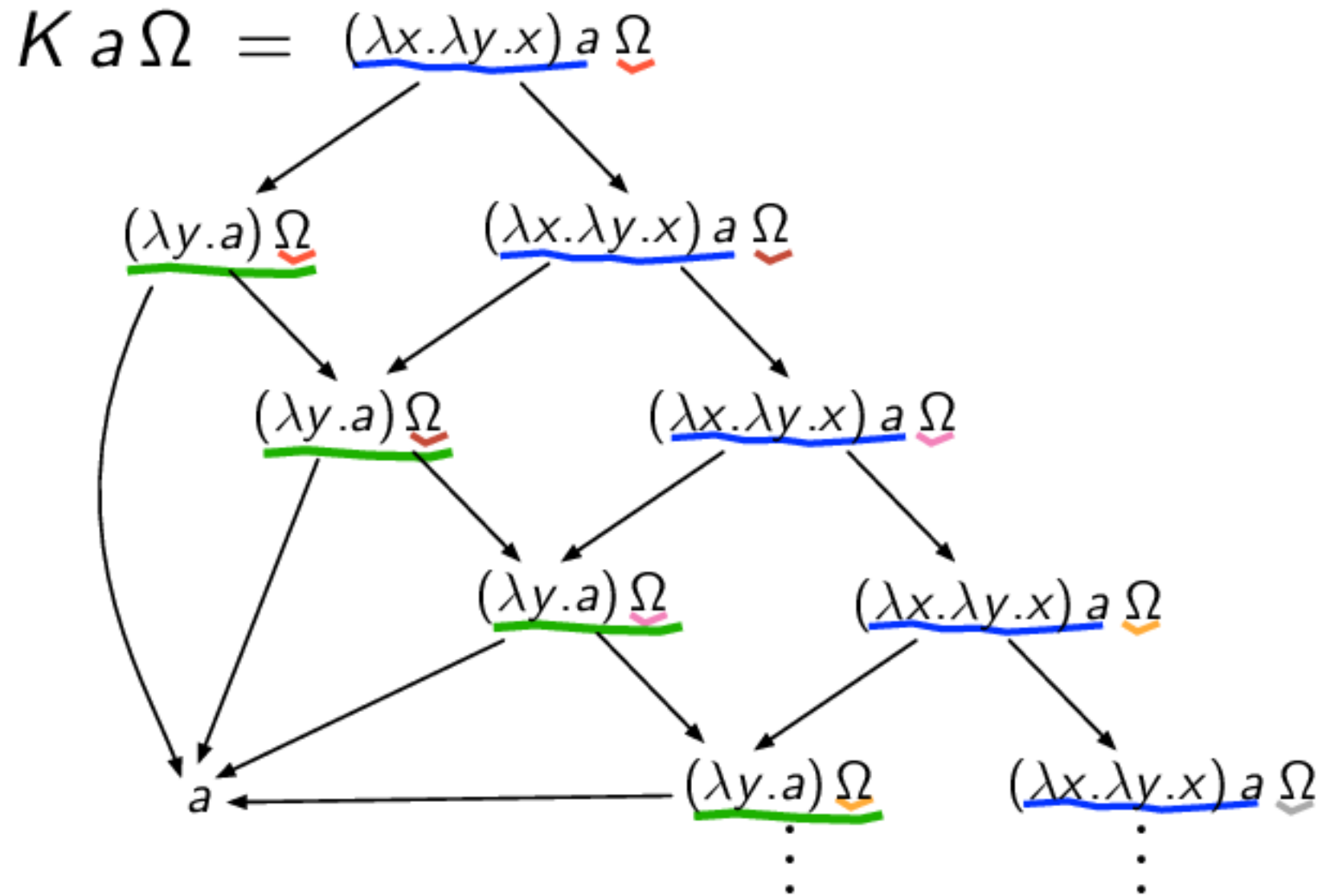
Tracing redexes



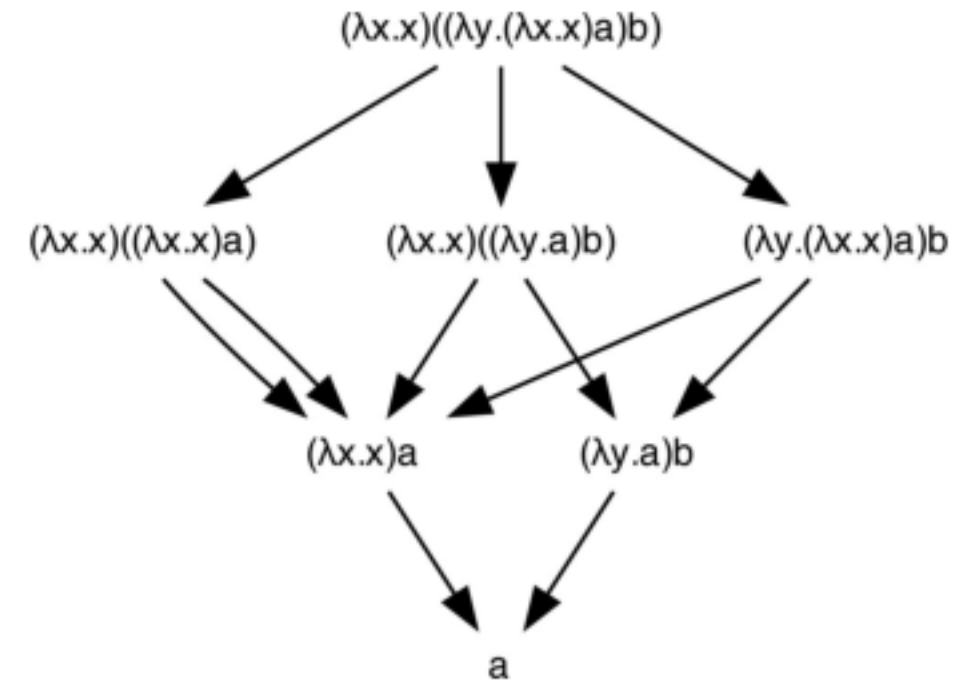
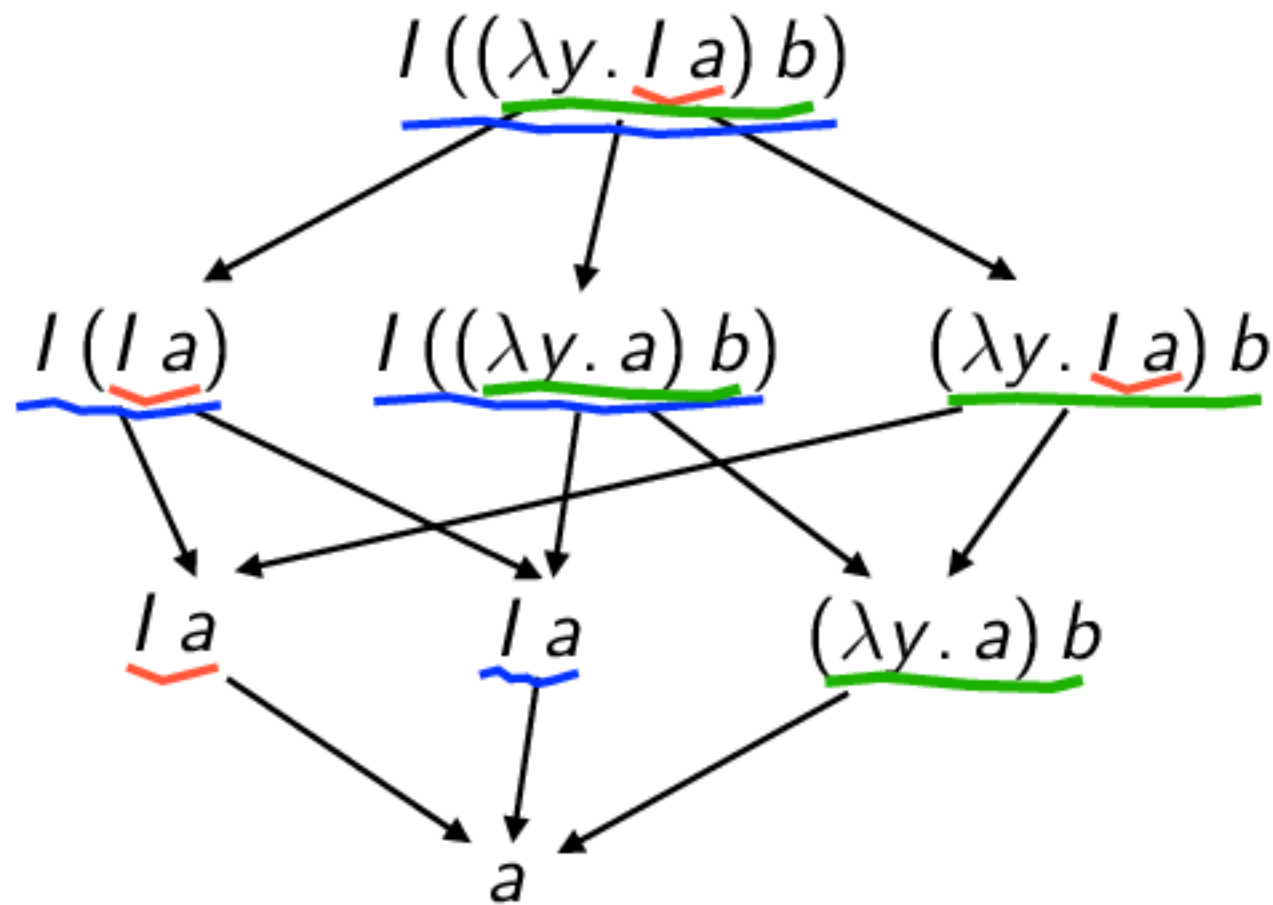
Tracing redexes



Tracing redexes



Tracing redexes



Empirical facts

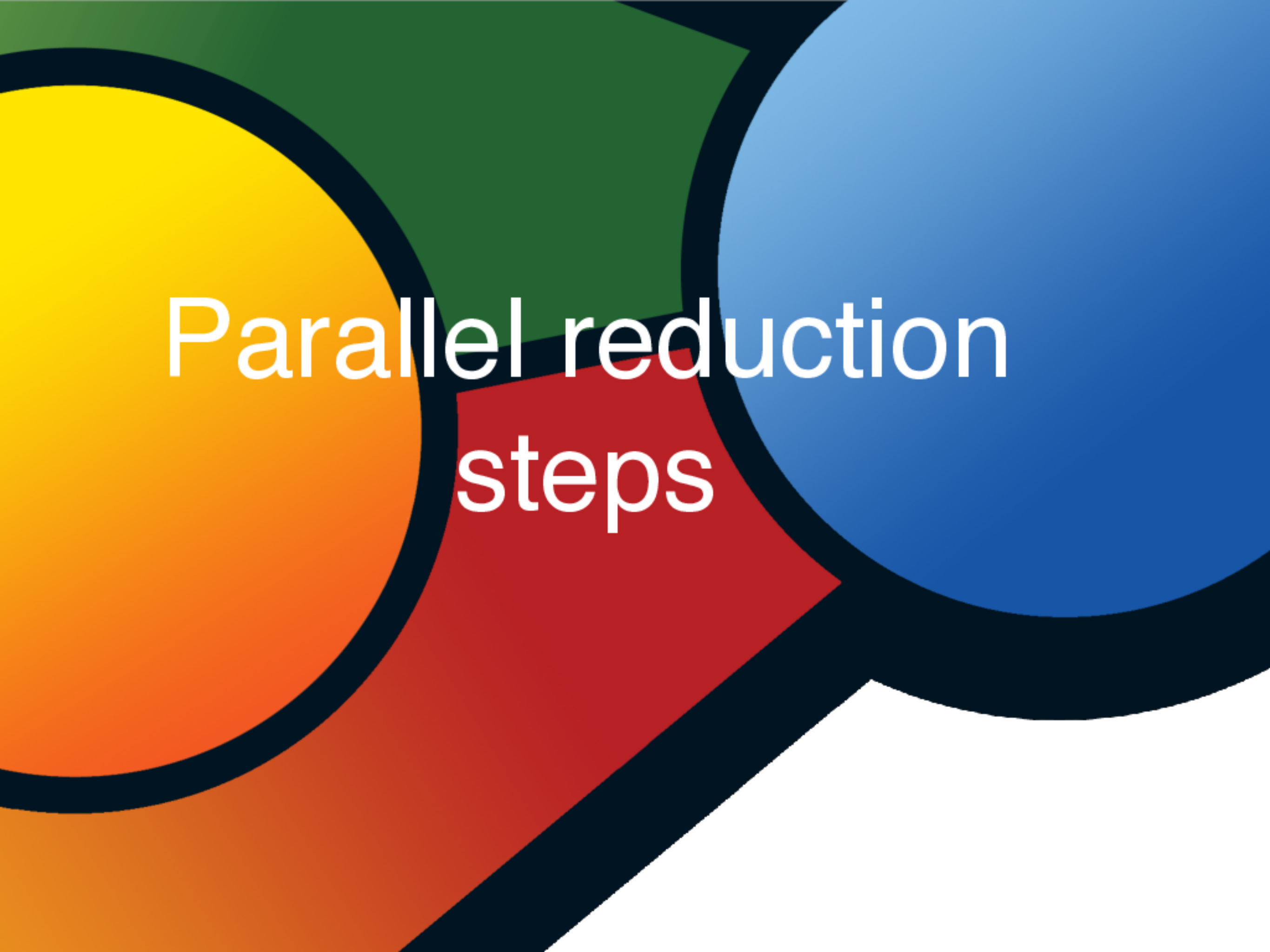
- initial redexes in the initial term
- and **newly** created redexes along reductions
- **infinite** reduction iff length of creation is unbounded ?
- **deterministic** result when finite families of redexes are contracted ?



Finite Developments Theorem

Curry '50

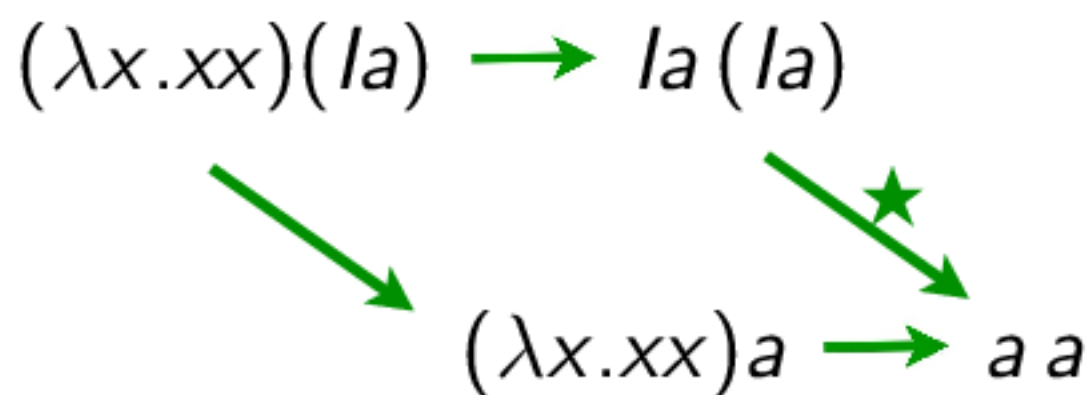
JJL '78



Parallel reduction
steps

Parallel reductions (1/3)

- permutation of reductions has to cope with copies of redexes



- in fact, a parallel reduction $la(la) \not\Rightarrow aa$
- in λ -calculus, need to define parallel reductions for nested sets

Fact In the λ -calculus, disjoint redexes may become nested $(\lambda x.lx)(\Delta y) \rightarrow l(\Delta y)$

Parallel reductions (2/3)

- the axiomatic way (à la Martin-Löf)

$$\text{[Var Axiom]} \quad x \twoheadrightarrow x$$

$$\text{[Const Axiom]} \quad c \twoheadrightarrow c$$

$$\text{[App Rule]} \quad \frac{M \twoheadrightarrow M' \quad N \twoheadrightarrow N'}{MN \twoheadrightarrow M'N'}$$

$$\text{[Abs Rule]} \quad \frac{M \twoheadrightarrow M'}{\lambda x.M \twoheadrightarrow \lambda x.M'}$$

$$\text{[Beta Rule]} \quad \frac{M \twoheadrightarrow M' \quad N \twoheadrightarrow N'}{(\lambda x.M)N \twoheadrightarrow M'\{x := N'\}}$$

inside-out (possibly void) parallel reductions

- examples:

$$(\lambda x.lx)(ly) \twoheadrightarrow (\lambda x.x)y$$

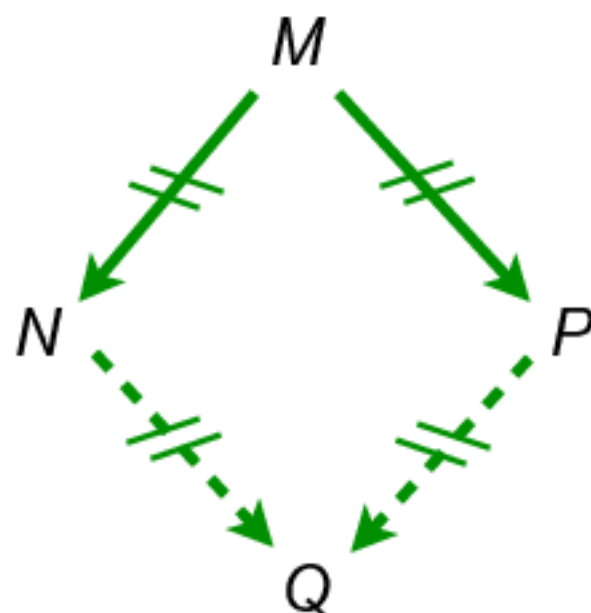
$$(\lambda x.(\lambda y.yy)x)(la) \twoheadrightarrow la(la)$$

$$(\lambda x.(\lambda y.yy)x)(la) \twoheadrightarrow (\lambda y.yy)a$$

Parallel reductions (3/3)

- **Parallel moves lemma** [Curry 50]

If $M \twoheadrightarrow N$ and $M \twoheadrightarrow P$, then $N \twoheadrightarrow Q$ and $P \twoheadrightarrow Q$ for some Q .



lemma 1-1-1-1
(strong confluency)

Enough to prove Church Rosser theorem since $\rightarrow \subset \twoheadrightarrow \subset \rightarrow^*$
[Tait--Martin L f 60?]

Reduction of a set of redexes (1/4)

- Goal: parallel reduction of a **given** set of redexes

$$M, N ::= x \mid \lambda x.M \mid MN \mid (\lambda x.M)^a N$$

$$a, b, c, \dots ::= \text{redex labels}$$

(labeled β -rule)

$$(\lambda x.M)^a N \longrightarrow M\{x := N\}$$

- Substitution as before with **add-on**:

$$((\lambda y.P)^a Q)\{x := N\} = (\lambda y.P\{x := N\})^a Q\{x := N\}$$

Reduction of a set of redexes (2/4)

- let $\mathcal{F}$ be a set of redex labels

$$\text{[Var Axiom]} \quad x \xrightarrow{\mathcal{F}} x$$

$$\text{[Const Axiom]} \quad c \xrightarrow{\mathcal{F}} c$$

$$\text{[App Rule]} \quad \frac{M \xrightarrow{\mathcal{F}} M' \quad N \xrightarrow{\mathcal{F}} N'}{MN \xrightarrow{\mathcal{F}} M'N'}$$

$$\text{[Abs Rule]} \quad \frac{M \xrightarrow{\mathcal{F}} M'}{\lambda x.M \xrightarrow{\mathcal{F}} \lambda x.M'}$$

$$\text{[Beta Rule]} \quad \frac{M \xrightarrow{\mathcal{F}} M' \quad N \xrightarrow{\mathcal{F}} N' \quad a \in \mathcal{F}}{(\lambda x.M)^a N \xrightarrow{\mathcal{F}} M' \{x := N'\}}$$

$$\text{[Redex']} \quad \frac{M \xrightarrow{\mathcal{F}} M' \quad N \xrightarrow{\mathcal{F}} N' \quad a \notin \mathcal{F}}{(\lambda x.M)^a N \xrightarrow{\mathcal{F}} (\lambda x.M')^a N'}$$

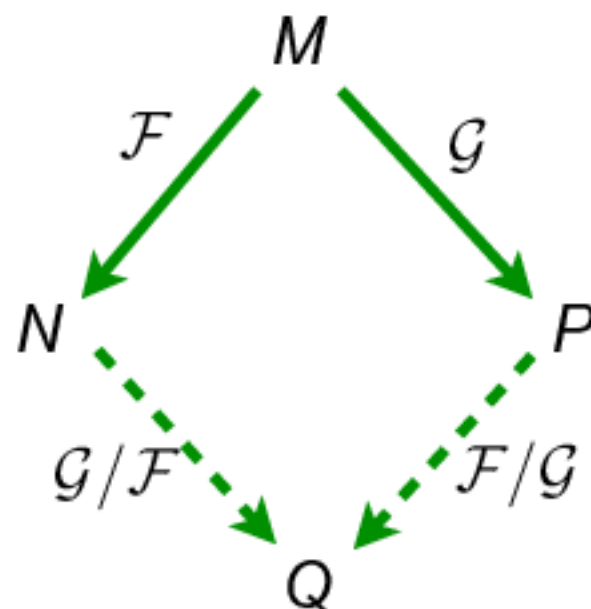
inside-out parallel reductions of redexes labeled in $\mathcal{F}$

- let $\mathcal{F}, \mathcal{G}$ be set of redexes in M and let $M \xrightarrow{\mathcal{F}} N$, then the set $\mathcal{G}/\mathcal{F}$ of **residuals** of $\mathcal{G}$ by $\mathcal{F}$ is the set of $\mathcal{G}$ redexes in N .

Reduction of a set of redexes (3/4)

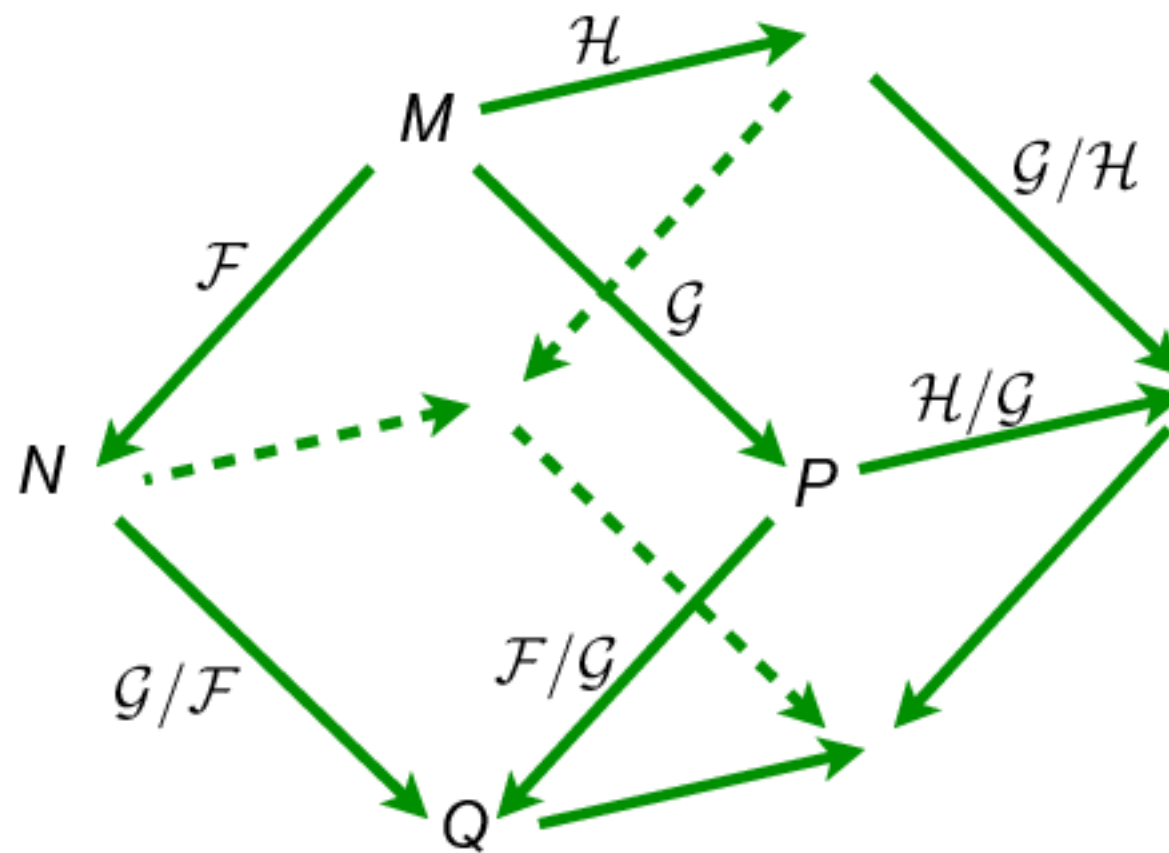
- **Parallel moves lemma** [Curry 50]

If $M \xrightarrow{\mathcal{F}} N$ and $M \xrightarrow{\mathcal{G}} P$, then $N \xrightarrow{\mathcal{G}/\mathcal{F}} Q$ and $P \xrightarrow{\mathcal{F}/\mathcal{G}} Q$ for some Q .

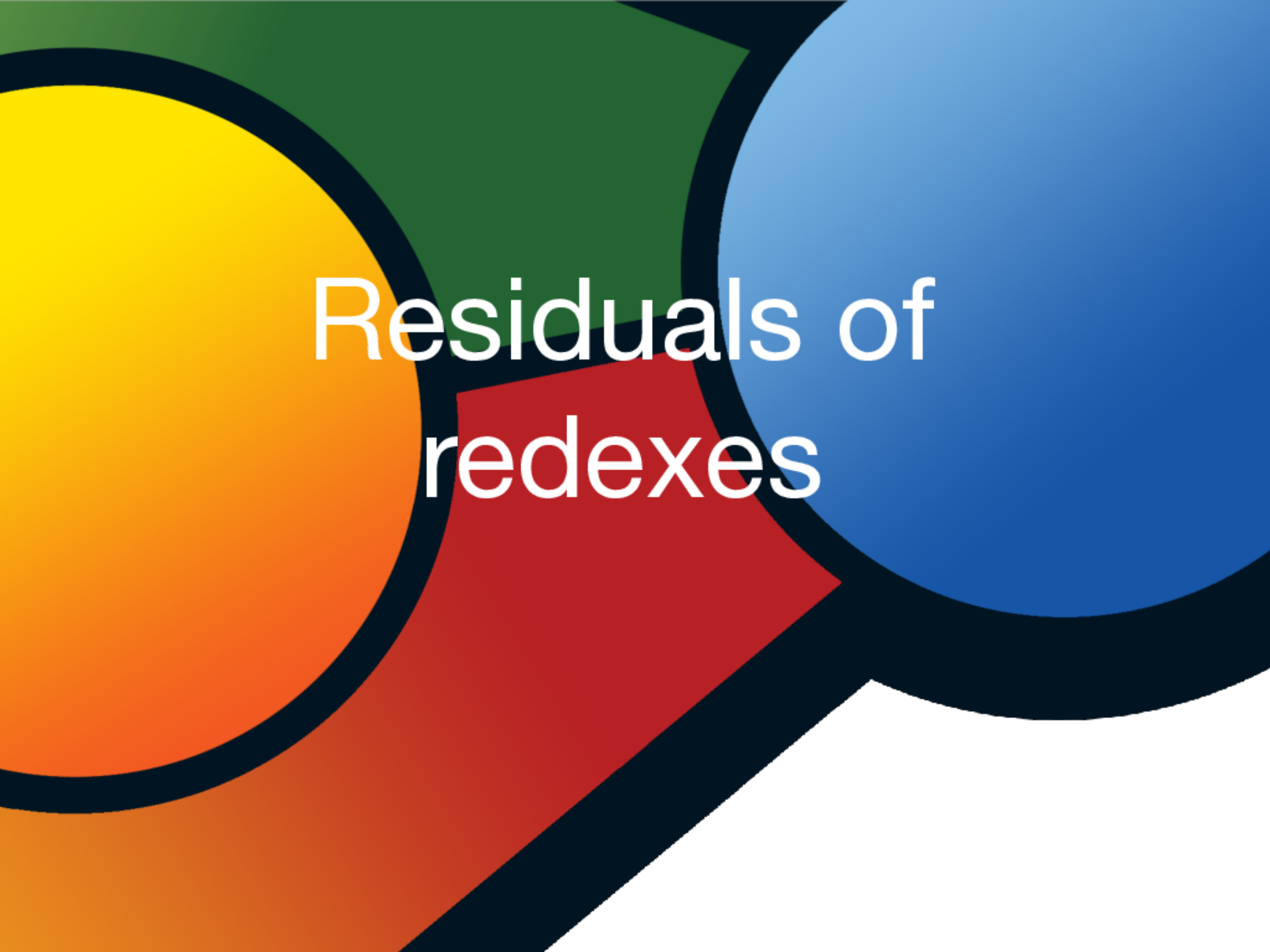


Reduction of a set of redexes (4/4)

- **Parallel moves lemma++** [Curry 50] The Cube Lemma



$$(\mathcal{H}/\mathcal{F})/(\mathcal{G}/\mathcal{F}) = (\mathcal{H}/\mathcal{G})/(\mathcal{F}/\mathcal{G})$$



Residuals of redexes

Redexes

- a **redex** is any **re**ducible **ex**pression: $(\lambda x.M)N$
- a **reduction step** contracts a given redex $R = (\lambda x.A)B$ and is written: $M \xrightarrow{R} N$
- a reduction step contracts a **singleton** set of redexes $M \xrightarrow{\{R\}} N$
- a more precise notation would be with occurrences of subterms. We avoid it here (but it is sometimes mandatory to avoid ambiguity)
- we replaced occurrences by giving names (labels) to redexes.

Residuals of redexes (1/4)

- **residuals** of redexes were defined by considering **labels**
- residuals are redexes with **same labels**
- a closer look w.r.t. their relative positions give following cases:

let $R = (\lambda x.A)B$, let $M \xrightarrow{R} N$ and $S = (\lambda y.C)D$ be an other redex in M . Then:

Residuals of redexes (2/4)

Case 1:

$$M = \dots R \dots \underline{S} \dots \xrightarrow{R} \dots R' \dots \underline{S} \dots = N$$

or

$$M = \dots \underline{S} \dots R \dots \xrightarrow{R} \dots \underline{S} \dots R' \dots = N$$

Case 2:

$$M = \dots \underline{R} \dots \xrightarrow{R} \dots R' \dots = N \quad (R \text{ and } S \text{ coincide})$$

Case 3:

$$M = \dots (\underline{\lambda y. \dots R \dots}) D \dots \xrightarrow{R} \dots (\underline{\lambda y. \dots R' \dots}) D \dots = N$$

Case 4:

$$M = \dots (\underline{\lambda y. C})(\dots R \dots) \dots \xrightarrow{R} \dots (\underline{\lambda y. C})(\dots R' \dots) \dots = N$$

Residuals of redexes (3/4)

Case 3:

$$M = \dots (\lambda x. \dots \underline{S} \dots) B \dots \xrightarrow{R} \dots \dots \underline{S\{x := B\}} \dots \dots = N$$

Case 4:

$$M = \dots (\lambda x. \dots x \dots x \dots) (\dots \underline{S} \dots) \dots$$
$$\searrow^R \dots (\dots \underline{S} \dots) \dots (\dots \underline{S} \dots) \dots \dots = N$$

Residuals of redexes (4/4)

Examples: $\Delta = \lambda x.xx$, $I = \lambda x.x$

$$\Delta(\underline{Ix}) \rightarrow \underline{Ix}(\underline{Ix})$$

$$\underline{Ix}(\Delta(Ix)) \rightarrow \underline{Ix}(\underline{Ix}(\underline{Ix}))$$

$$\underline{I(\Delta(Ix))} \rightarrow \underline{I(\underline{Ix}(\underline{Ix}))}$$

$$\underline{\Delta(Ix)} \rightarrow \underline{Ix}(\underline{Ix})$$

$$Ix(\Delta(\underline{Ix})) \rightarrow Ix(\underline{Ix}(\underline{Ix}))$$

$$\underline{\Delta\Delta} \rightarrow \Delta\Delta$$



Residuals of reductions

Parallel reductions

- Consider reductions where each step is parallel

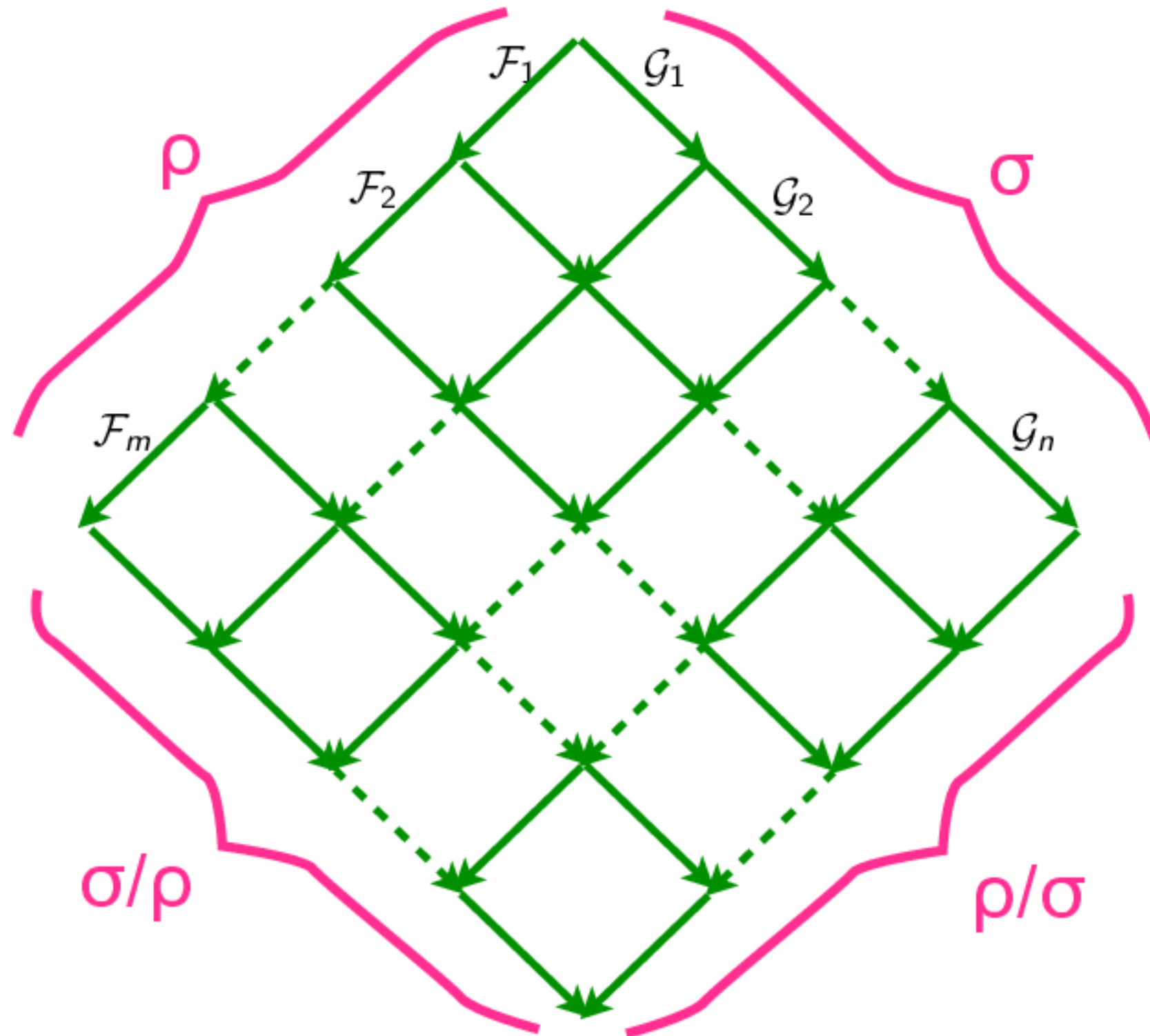
$$\rho : M = M_0 \xrightarrow{\mathcal{F}_1} M_1 \xrightarrow{\mathcal{F}_2} M_2 \cdots \xrightarrow{\mathcal{F}_n} M_n = N$$

- We also write

$$\rho = 0 \text{ when } n = 0$$

$$\rho = \mathcal{F}_1 \mathcal{F}_2 \cdots \mathcal{F}_n \text{ when } M \text{ clear from context}$$

Residuals of reductions (1/4)



Residuals of reductions (2/4)

- **Definition** [JJL 76]

$$\rho/0 = \rho$$

$$\rho/(\sigma \tau) = (\rho/\sigma)/\tau$$

$$(\rho \sigma)/\tau = (\rho/\tau)(\sigma/(\tau/\rho))$$

$\mathcal{F}/\mathcal{G}$ already defined

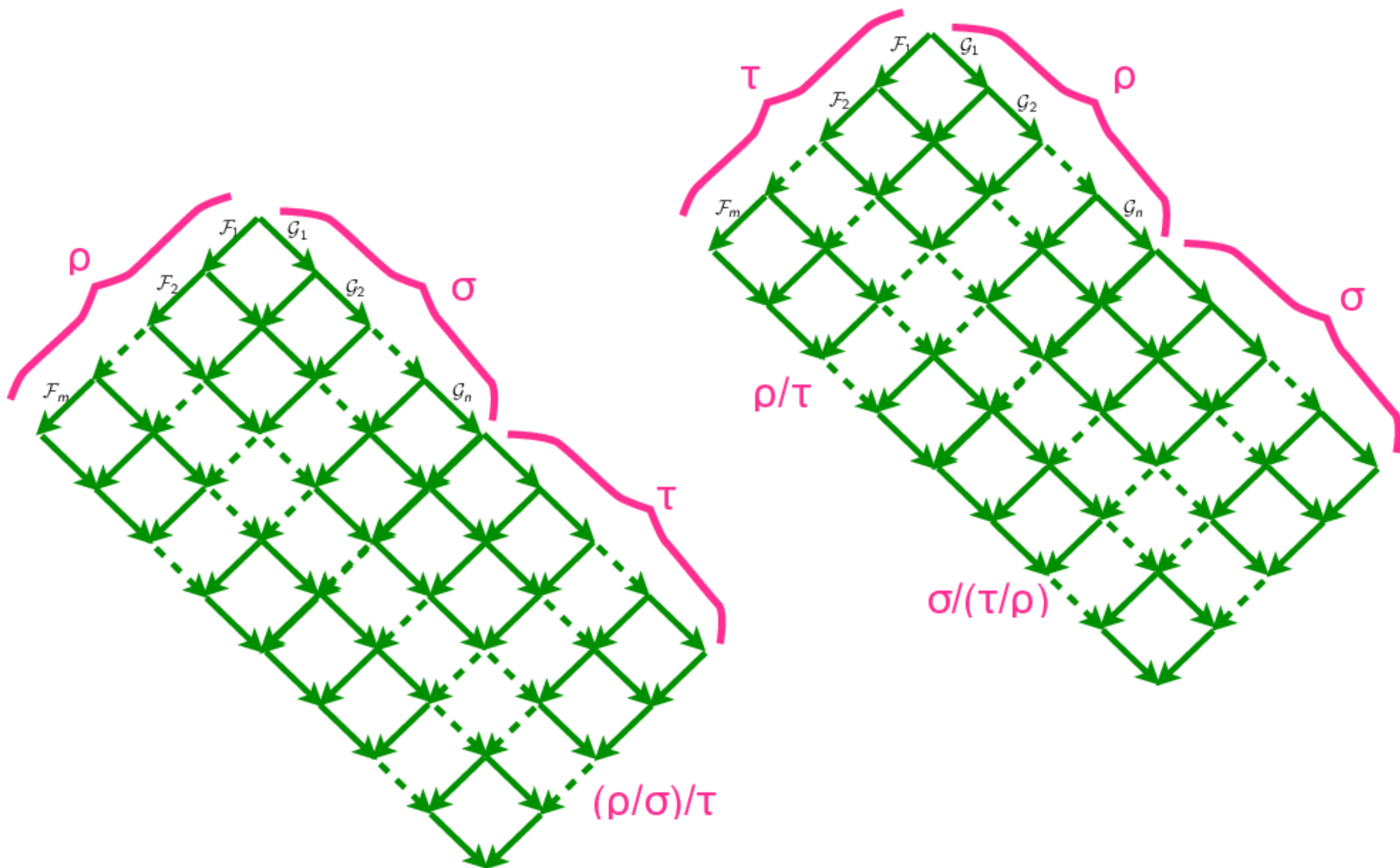
- **Notation**

$$\rho \sqcup \sigma = \rho(\sigma/\rho)$$

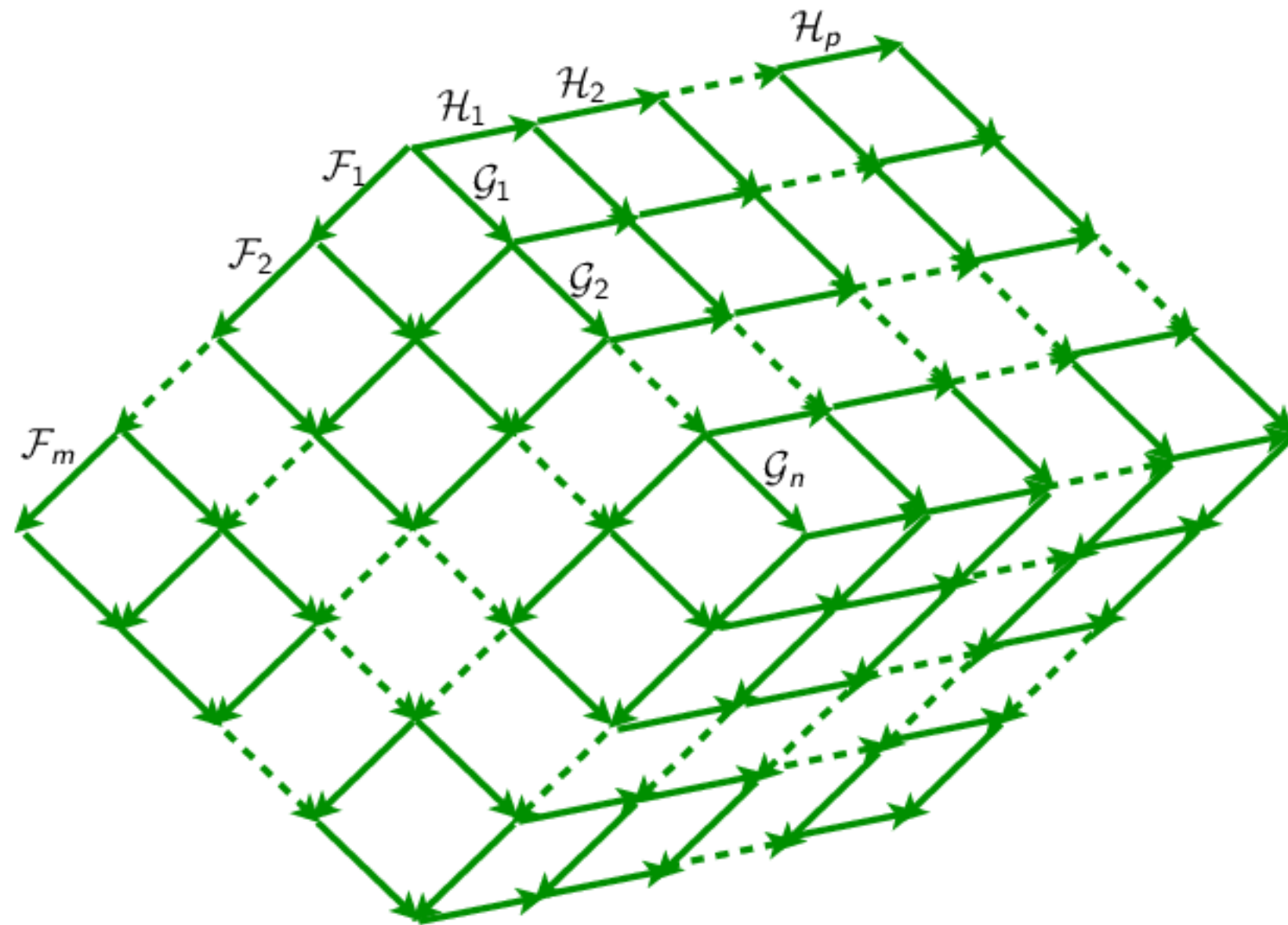
- **Proposition** [Parallel Moves +]:

$\rho \sqcup \sigma$ and $\sigma \sqcup \rho$ are cofinal

Residuals of reductions (3/4)



Residuals of reductions (4/4)



- **Proposition** [Cube Lemma ++]:

$$\tau/(\rho \sqcup \sigma) = \tau/(\sigma \sqcup \rho)$$



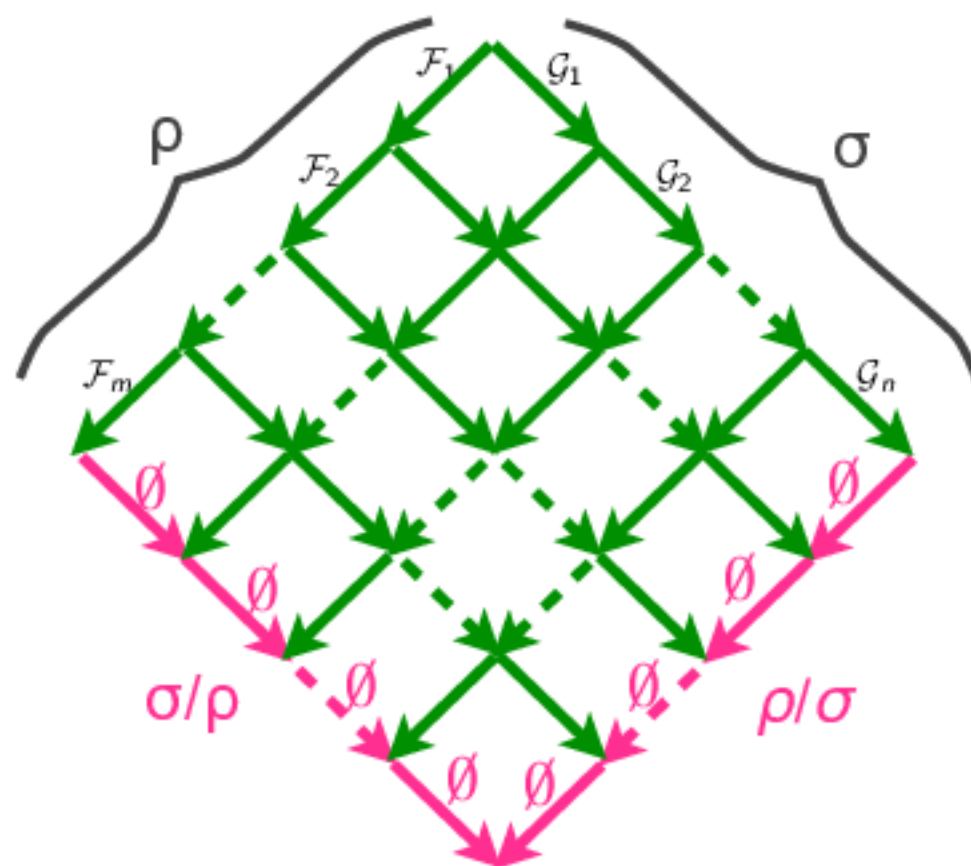
Equivalence by
permutations

Equivalence by permutations (1/4)

- Definition:**

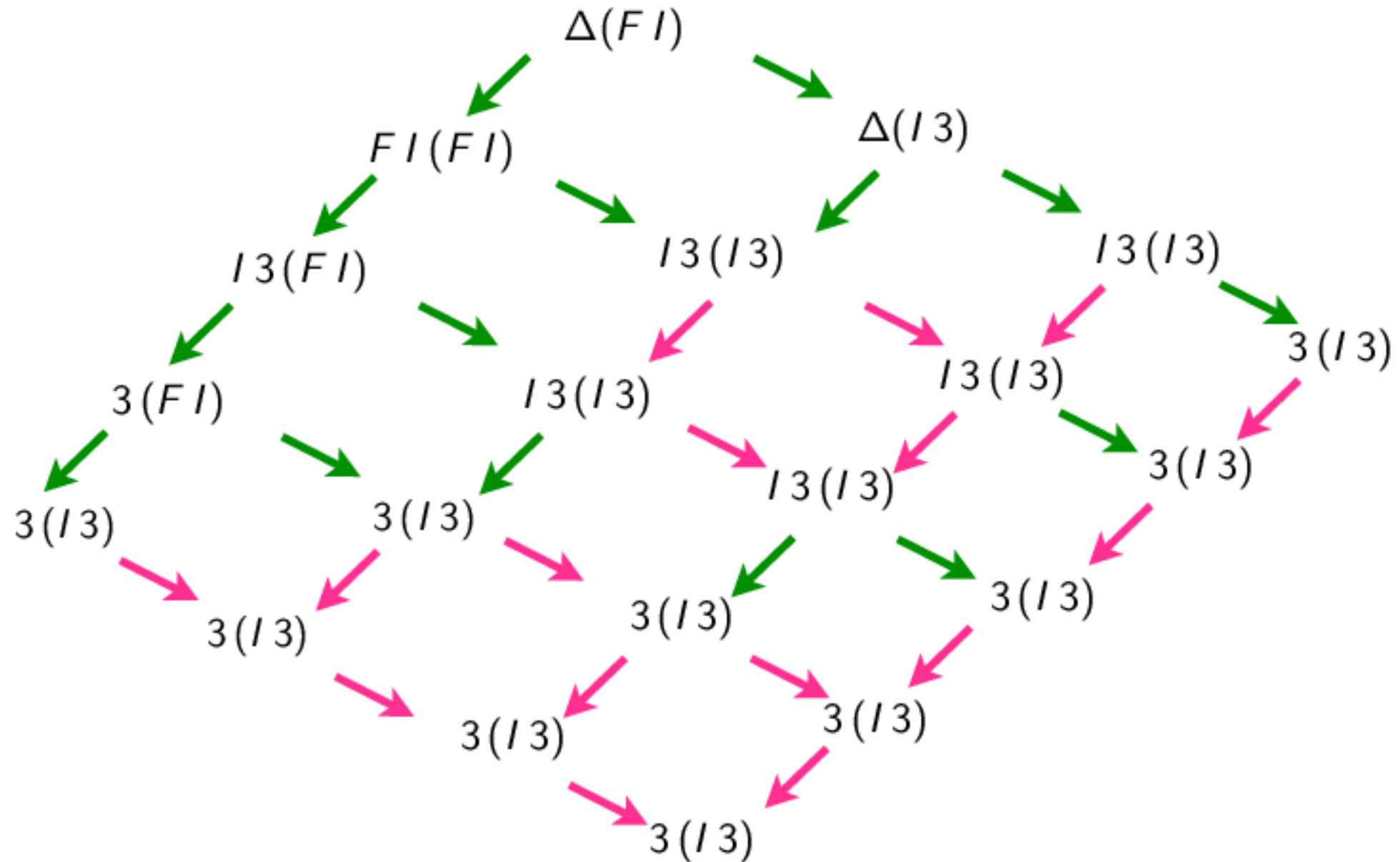
Let ρ and σ be 2 coinitial reductions. Then ρ is equivalent to σ by permutations, $\rho \simeq \sigma$, iff:

$$\rho/\sigma = \emptyset^m \quad \text{and} \quad \sigma/\rho = \emptyset^n$$

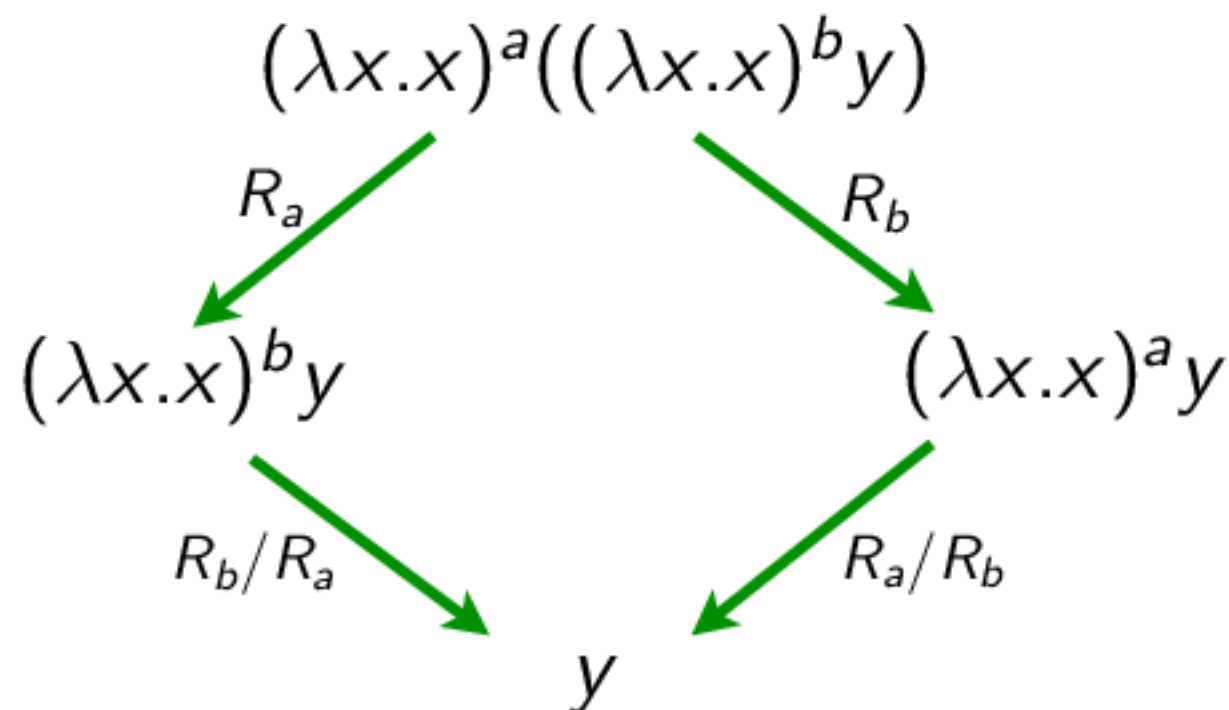


$\rho \simeq \sigma$ means that ρ and σ are coinitial and cofinal but converse is not true (see later)

Equivalence by permutations (2/4)

$$\Delta = \lambda x. xx$$
$$F = \lambda f.f\ 3$$
$$I = \lambda x.x$$


Equivalence by permutations (3/4)



$$\begin{aligned}
 \rho : M = I^a(I^b y) &\xrightarrow{R_a} I^b y \\
 \sigma : M = I^a(I^b y) &\xrightarrow{R_b} I^a y
 \end{aligned}$$

- Here $\rho \not\equiv \sigma$ while ρ and σ are coinitial and cofinal in the calculus with no labels

Equivalence by permutations (4/4)

- Same with $0 \not\approx \rho$ when $\rho : \Delta\Delta \xrightarrow{\text{green}} \Delta\Delta$

$$\Delta = \lambda x.xx$$

Exercise 1: Give other examples of non-equivalent reductions between same terms.

Exercise 2: Show following equalities

$$\rho/0 = \rho$$

$$\emptyset^n/\rho = \emptyset^n$$

$$0/\rho = 0$$

$$0 \simeq \emptyset^n$$

$$\rho/\emptyset^n = \rho$$

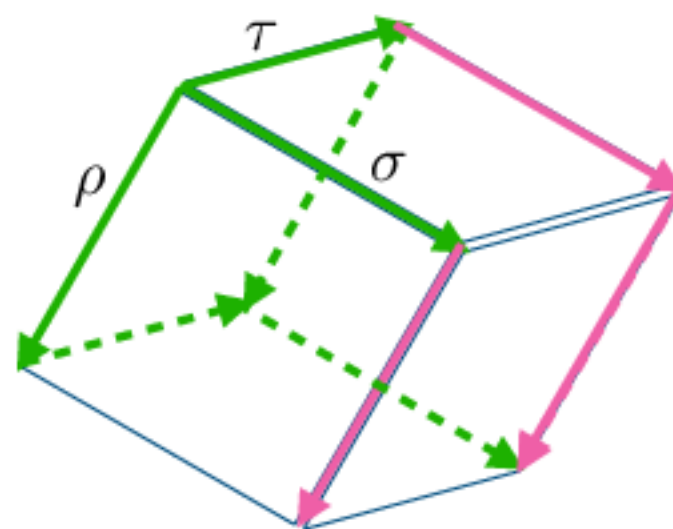
$$\rho/\rho = \emptyset^n$$

Equivalence by permutations (4/4)

Exercise 3: Show that $\simeq$ is an equivalence relation.

Proof

- (i) $\rho \simeq \rho$ obvious
- (ii) same with $\rho \simeq \sigma$ implies $\sigma \simeq \rho$
- (iii) $\rho \simeq \sigma \simeq \tau$ implies $\rho \simeq \tau$??



Properties of permutations (1/3)

- **Proposition**

$$(i) \quad \rho \simeq \sigma \text{ iff } \forall \tau. \tau/\rho = \tau/\sigma$$

$$(ii) \quad \rho \simeq \sigma \text{ implies } \rho/\tau = \sigma/\tau$$

$$(iii) \quad \rho \simeq \sigma \text{ iff } \tau\rho \simeq \tau\sigma$$

$$(iv) \quad \rho \simeq \sigma \text{ implies } \rho\tau \simeq \sigma\tau$$

$$(v) \quad \rho \sqcup \sigma \simeq \sigma \sqcup \rho$$

Proof

$$(i) \quad \rho \simeq \sigma \text{ implies } \sigma/\rho = \emptyset^n \text{ and } \rho/\sigma = \emptyset^m.$$

$$\text{Thus } \tau/(\rho \sqcup \sigma) = \tau/(\rho(\sigma/\rho)) = \tau/\rho/(\sigma/\rho) = \tau/\rho/\emptyset^m = \tau/\rho$$

$$\text{Similarly } \tau/(\sigma \sqcup \rho) = \tau/\sigma$$

$$\text{By cube lemma } \tau/\rho = \tau/\sigma$$

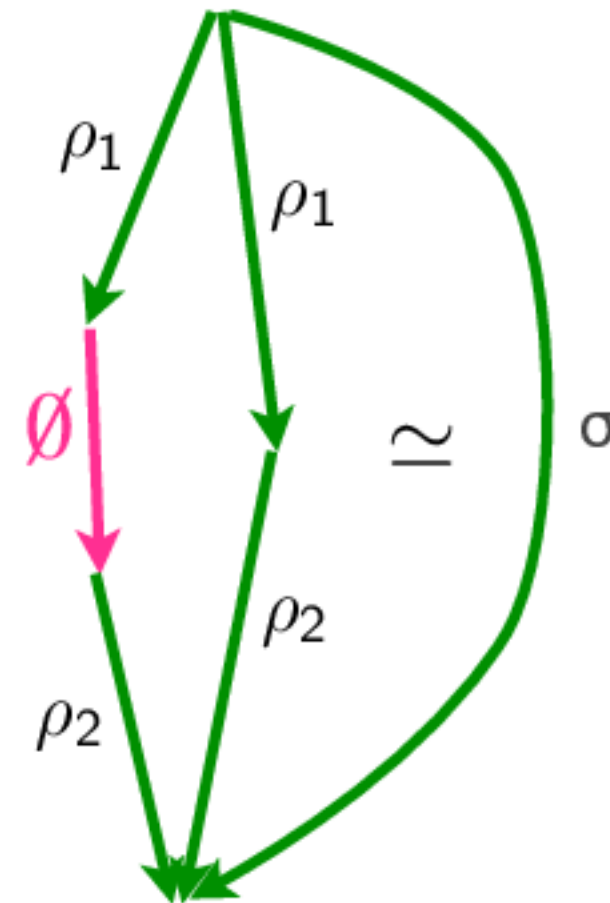
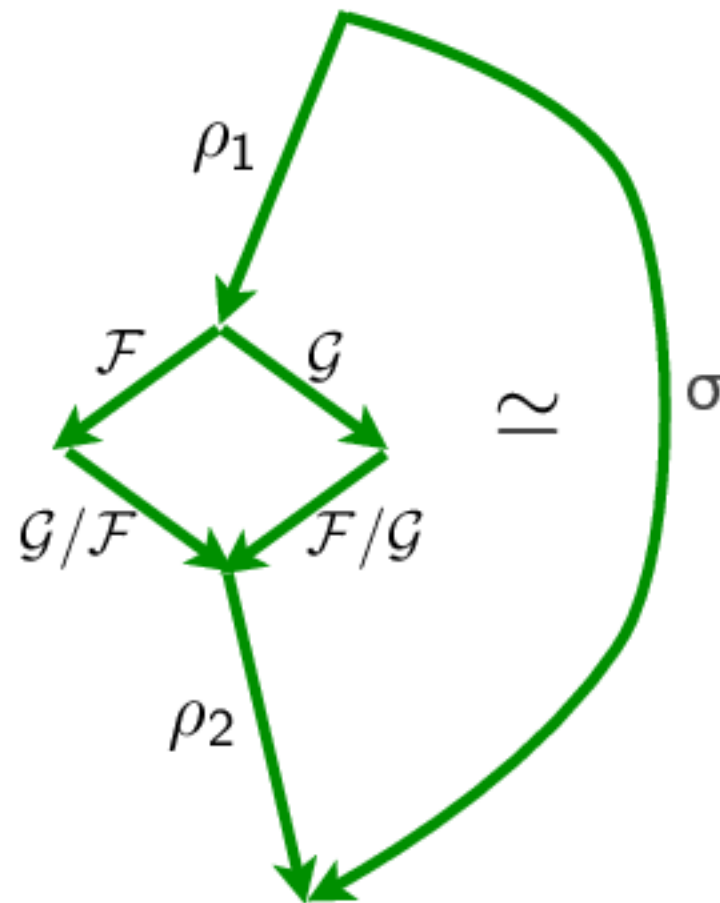
Conversely, take $\tau = \rho$ and $\tau = \sigma$.

Properties of permutations (2/3)

- **Proposition** $\simeq$ is the smallest congruence containing

$$\mathcal{F}(\mathcal{G}/\mathcal{F}) \simeq \mathcal{G}(\mathcal{F}/\mathcal{G})$$

$$0 \simeq \emptyset$$

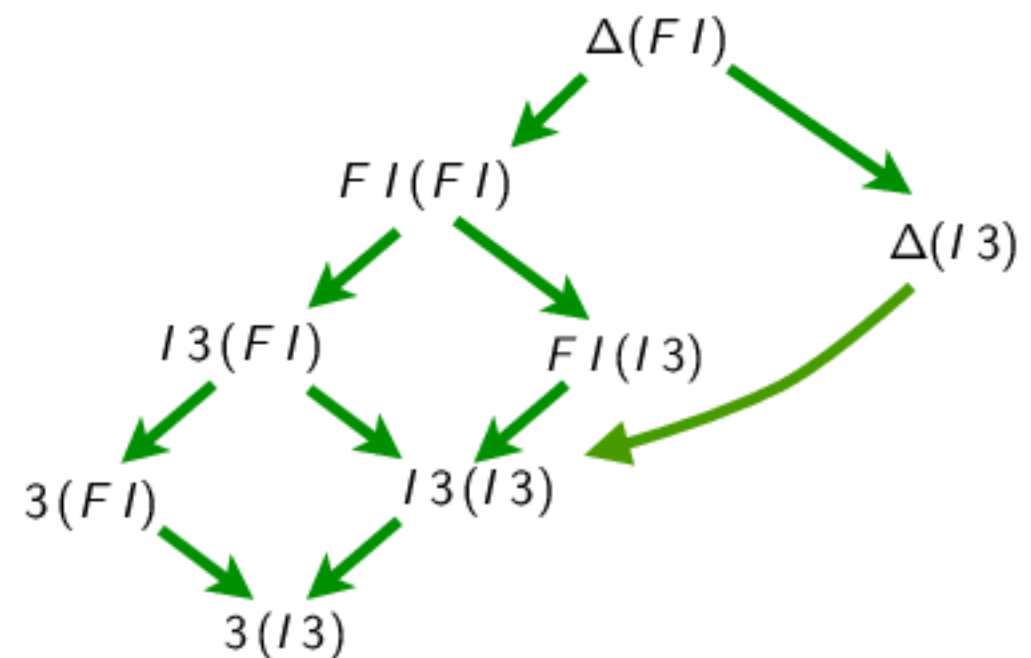
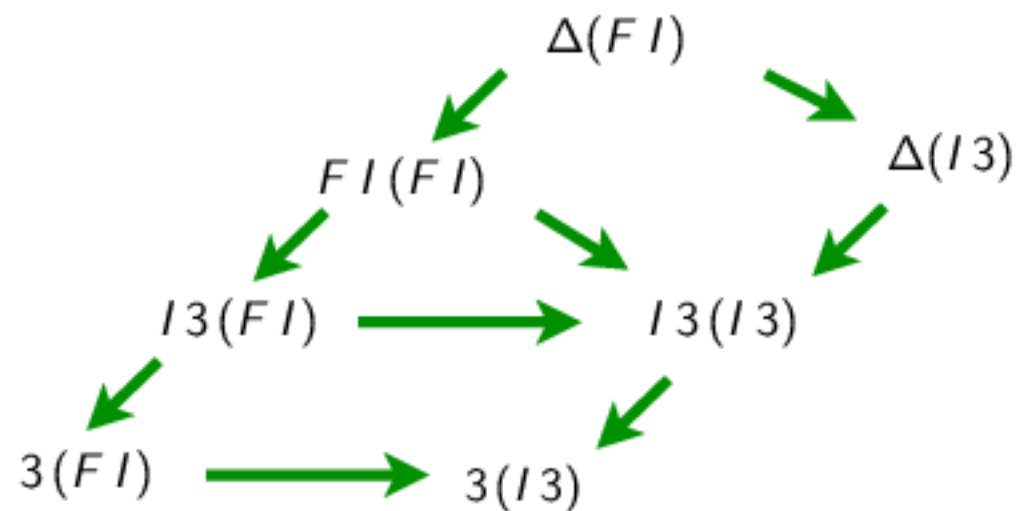
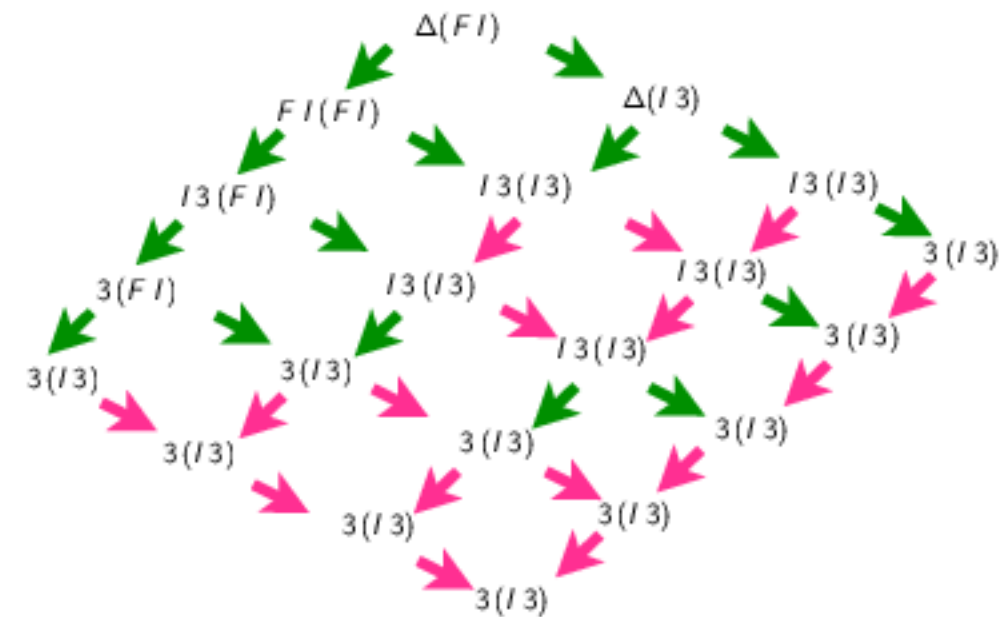


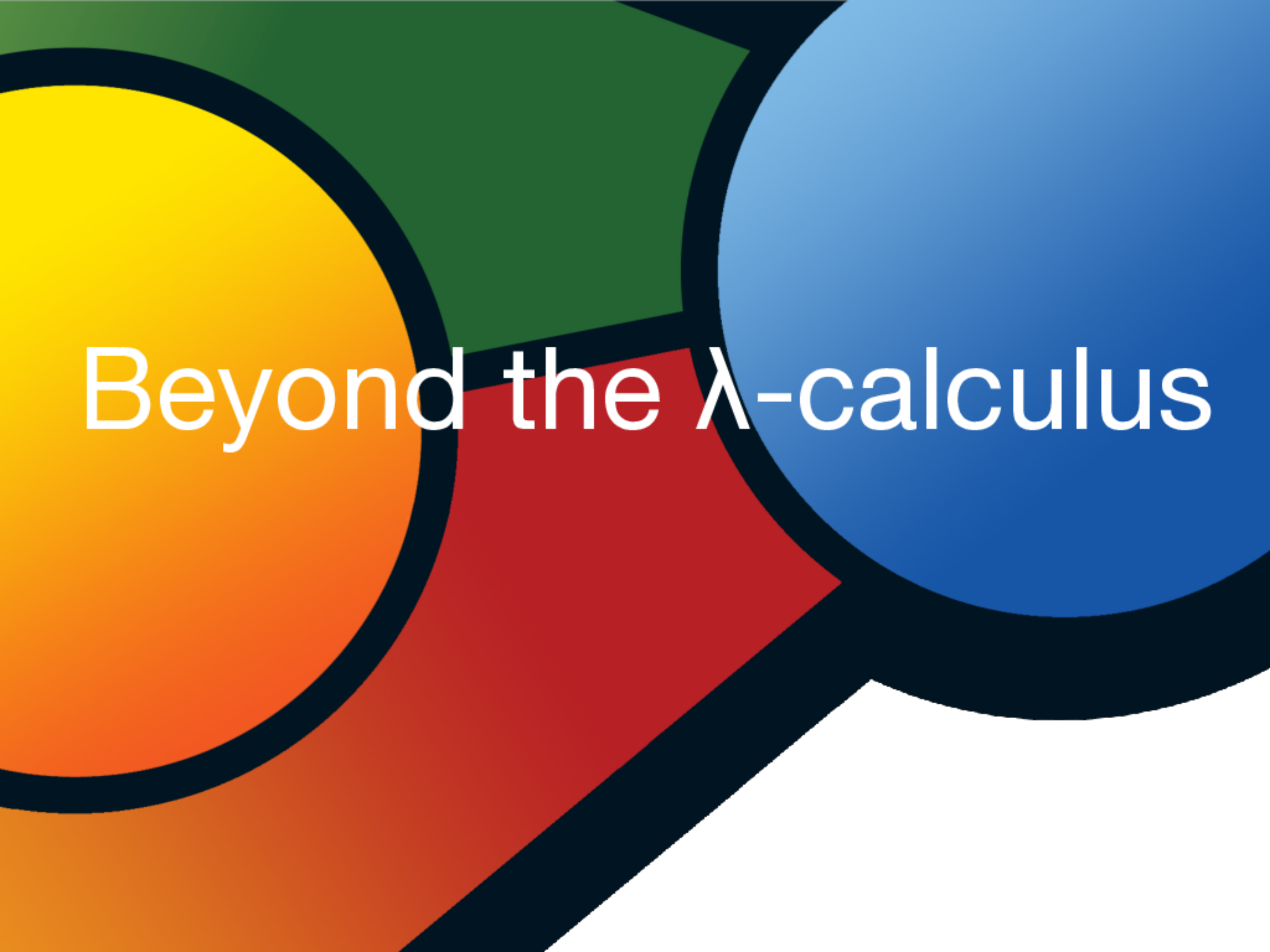
Properties of permutations (3/3)

$$\Delta = \lambda x.xx$$

$$F = \lambda f.f\ 3$$

$$I = \lambda x.x$$





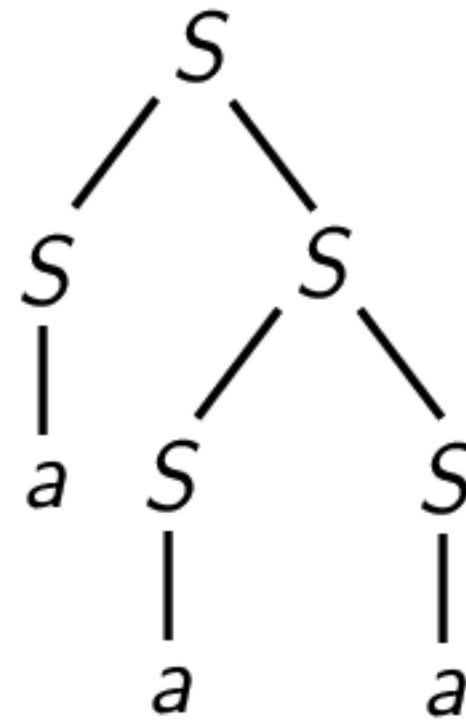
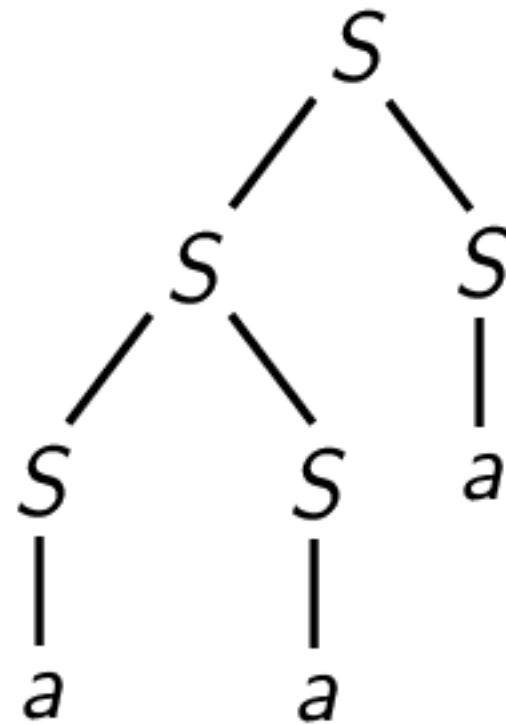
Beyond the λ -calculus

Context-free languages

- permutations of derivations in context-free languages

$S \rightarrow SS$

$S \rightarrow a$



- each parse tree corresponds to an equivalence class

Term rewriting

- recursive program schemes [Berry-JJL'77]
- permutations of derivations in orthogonal TRS [Huet-JJL'81]
- permutations of derivations are defined with critical pairs
- critical pairs make conflicts
- only 2nd definition of equivalence works [Boudol'82]
- interaction systems [Asperti-Laneve'93]

Process algebras

- similar to TRS [Boudol-Castellani'82]
- connection to event structures [Laneve'84]

PCF

- LCF considered as a programming language [Plotkin'74]

$M, N, P ::= x$	variable
$ \lambda x.M \mid M N$	abstraction application
$ \underline{n}$	integer constant
$ M \otimes N$	$\otimes \in \{+, -, \times, \div\}$
$ \text{ifz } M \text{ then } N \text{ then } N$	conditionnal
$ \mu x.M$	recursive definition

β $(\lambda x.M)N \rightarrow M \{x := N\}$

op $\underline{m} \otimes \underline{n} \rightarrow \underline{m \otimes n}$

$cond1$ $\text{ifz } \underline{0} \text{ then } M \text{ else } N \rightarrow M$

$cond2$ $\text{ifz } \underline{n+1} \text{ then } M \text{ else } N \rightarrow N$

μ $\mu x.M \rightarrow M \{x := \mu x.M\}$

Exemples de termes

$\text{Fact}(3)$

$\text{Fact} = Y(\lambda f. \lambda x. \text{ifz } x \text{ then } 1 \text{ else } x \star f(x - 1))$

$Y = \lambda f. (\lambda x. f(xx))(\lambda x. f(xx))$

s'écrit

$(\lambda \text{Fact} . \text{Fact}(3))$

$((\lambda Y. Y(\lambda f. \lambda x. \text{ifz } x \text{ then } 1 \text{ else } x \star f(x - 1)))$

$(\lambda f. (\lambda x. f(xx))(\lambda x. f(xx))))$

$(\lambda \text{Fact}. \text{Fact}3)(\lambda y. y(\lambda f. \lambda x. \text{ifz } x \text{ then } 1 \text{ else } x * f(x-1)))(\lambda f. Yf))$



$(\lambda y. y(\lambda f. \lambda x. \text{ifz } x \text{ then } 1 \text{ else } x * f(x-1)))(\lambda f. Yf)3$



$(\lambda f. Yf)(\lambda f. \lambda x. \text{ifz } x \text{ then } 1 \text{ else } x * f(x-1))3$



$(\lambda x. (\lambda f. \lambda x. \text{ifz } x \text{ then } 1 \text{ else } x * f(x-1))(xx))(\lambda x. (\lambda f. \lambda x. \text{ifz } x \text{ then } 1 \text{ else } x * f(x-1))(xx))3$



$(\lambda f. \lambda x. \text{ifz } x \text{ then } 1 \text{ else } x * f(x-1))((\lambda x. (\lambda f. \lambda x. \text{ifz } x \text{ then } 1 \text{ else } x * f(x-1))(xx))(\lambda x. (\lambda f. \lambda x. \text{ifz } x \text{ then } 1 \text{ else } x * f(x-1))(xx)))3$



$(\lambda x. \text{ifz } x \text{ then } 1 \text{ else } x * (\lambda x. (\lambda f. \lambda x. \text{ifz } x \text{ then } 1 \text{ else } x * f(x-1))(xx))(\lambda x. (\lambda f. \lambda x. \text{ifz } x \text{ then } 1 \text{ else } x * f(x-1))(xx))(x-1))3$



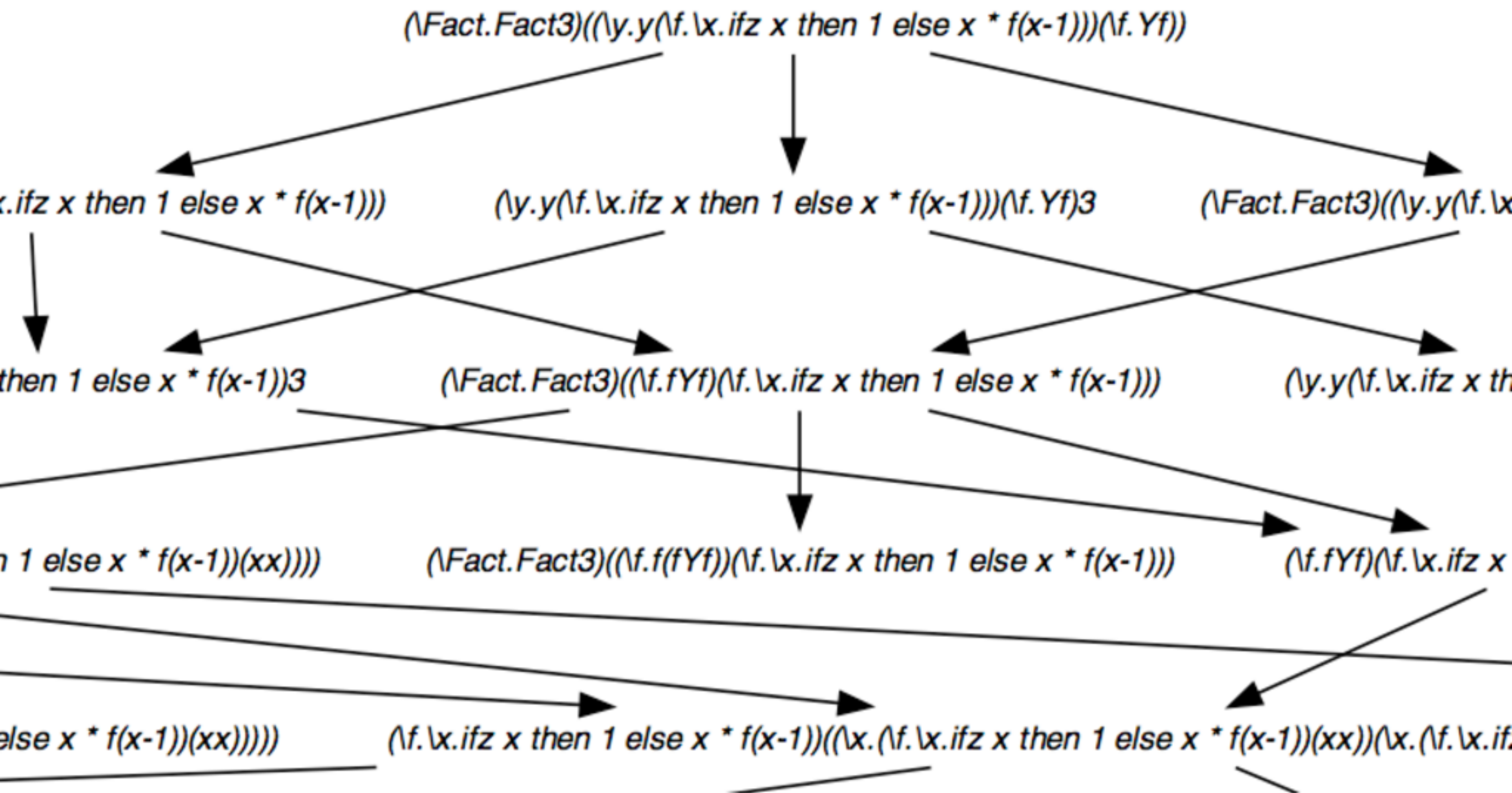
$\text{ifz } 3 \text{ then } 1 \text{ else } 3 * (\lambda x. (\lambda f. \lambda x. \text{ifz } x \text{ then } 1 \text{ else } x * f(x-1))(xx))(\lambda x. (\lambda f. \lambda x. \text{ifz } x \text{ then } 1 \text{ else } x * f(x-1))(xx))(3-1)$

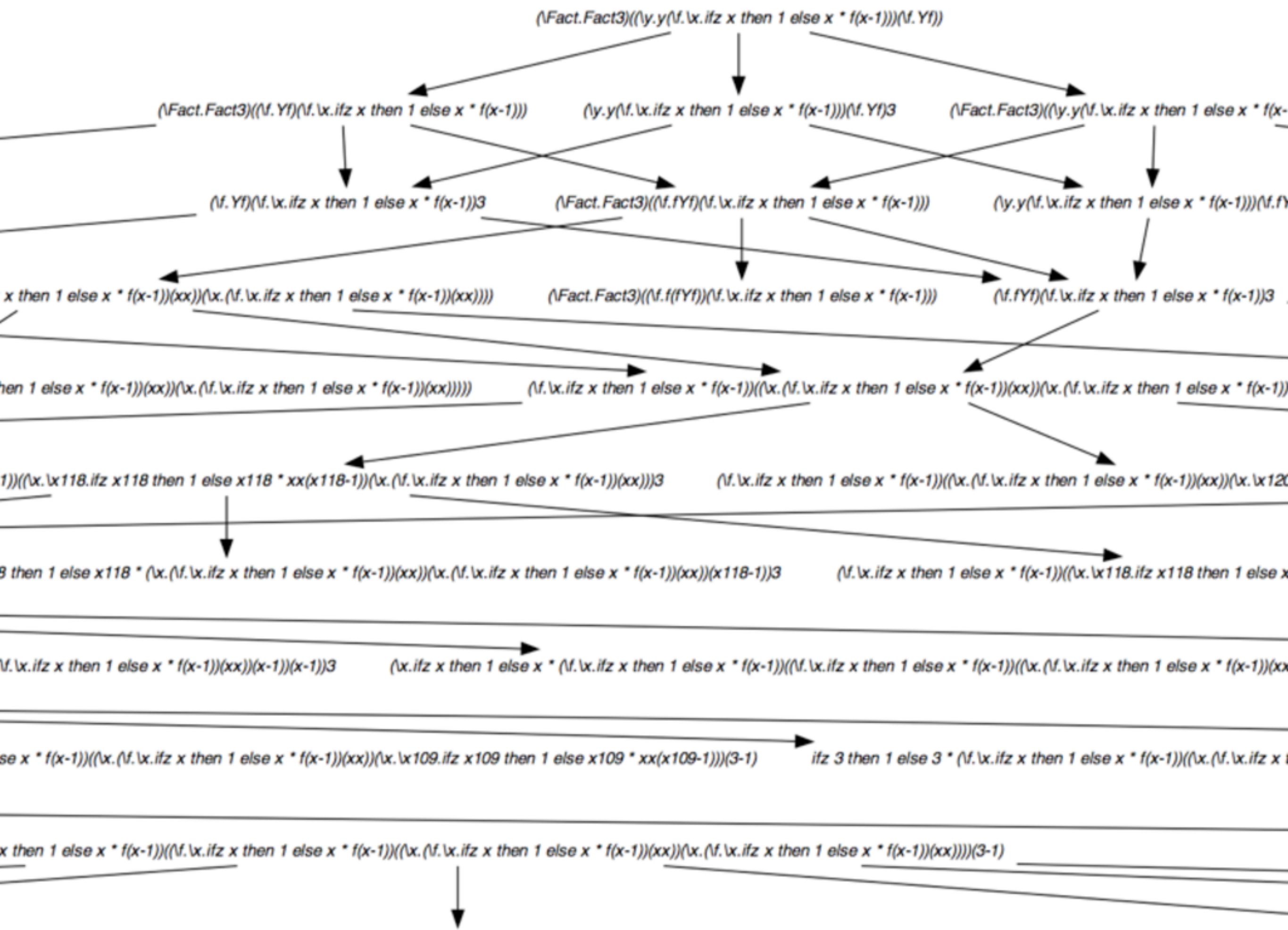


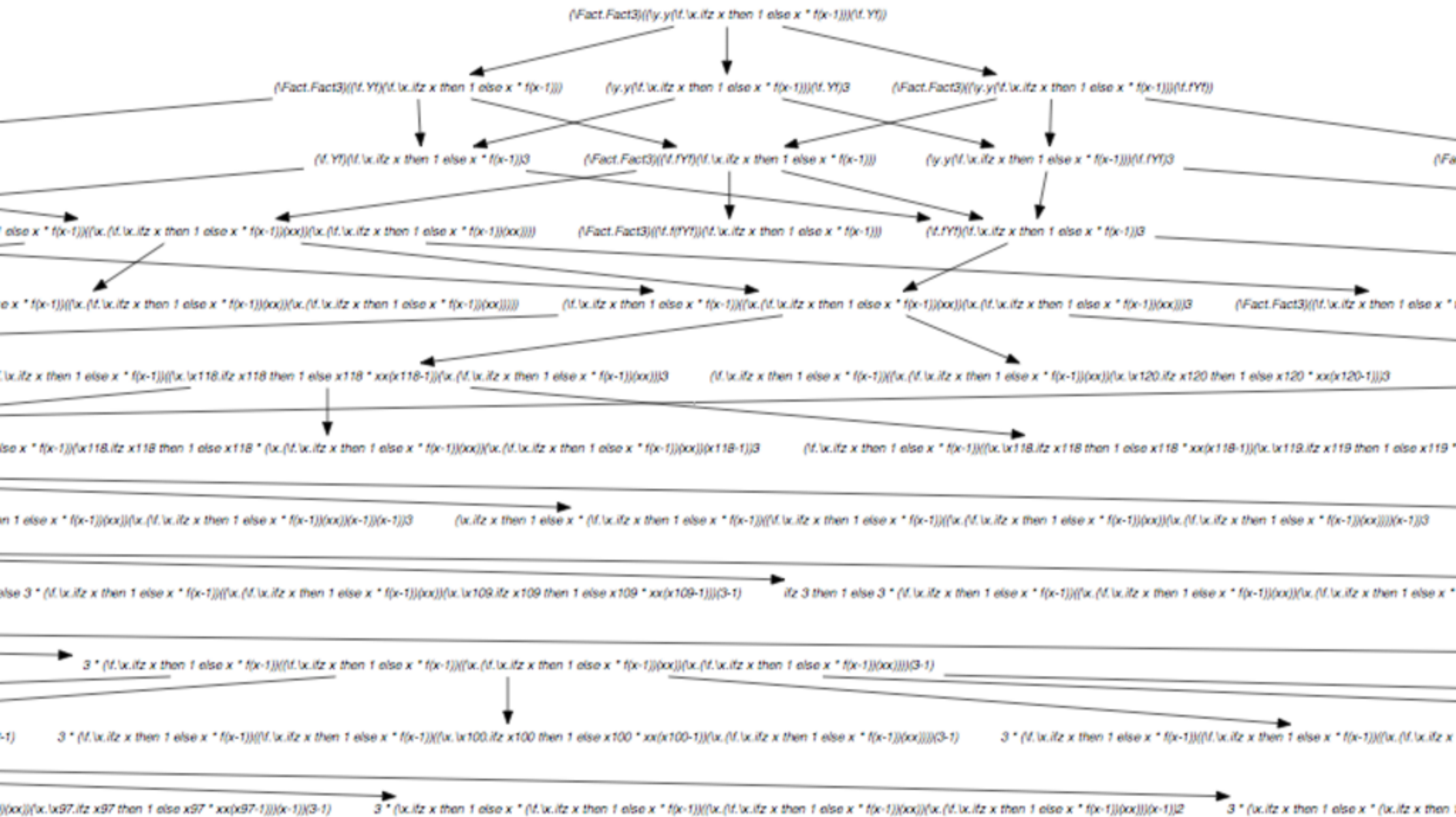
$3 * (\lambda x. (\lambda f. \lambda x. \text{ifz } x \text{ then } 1 \text{ else } x * f(x-1))(xx))(\lambda x. (\lambda f. \lambda x. \text{ifz } x \text{ then } 1 \text{ else } x * f(x-1))(xx))(3-1)$

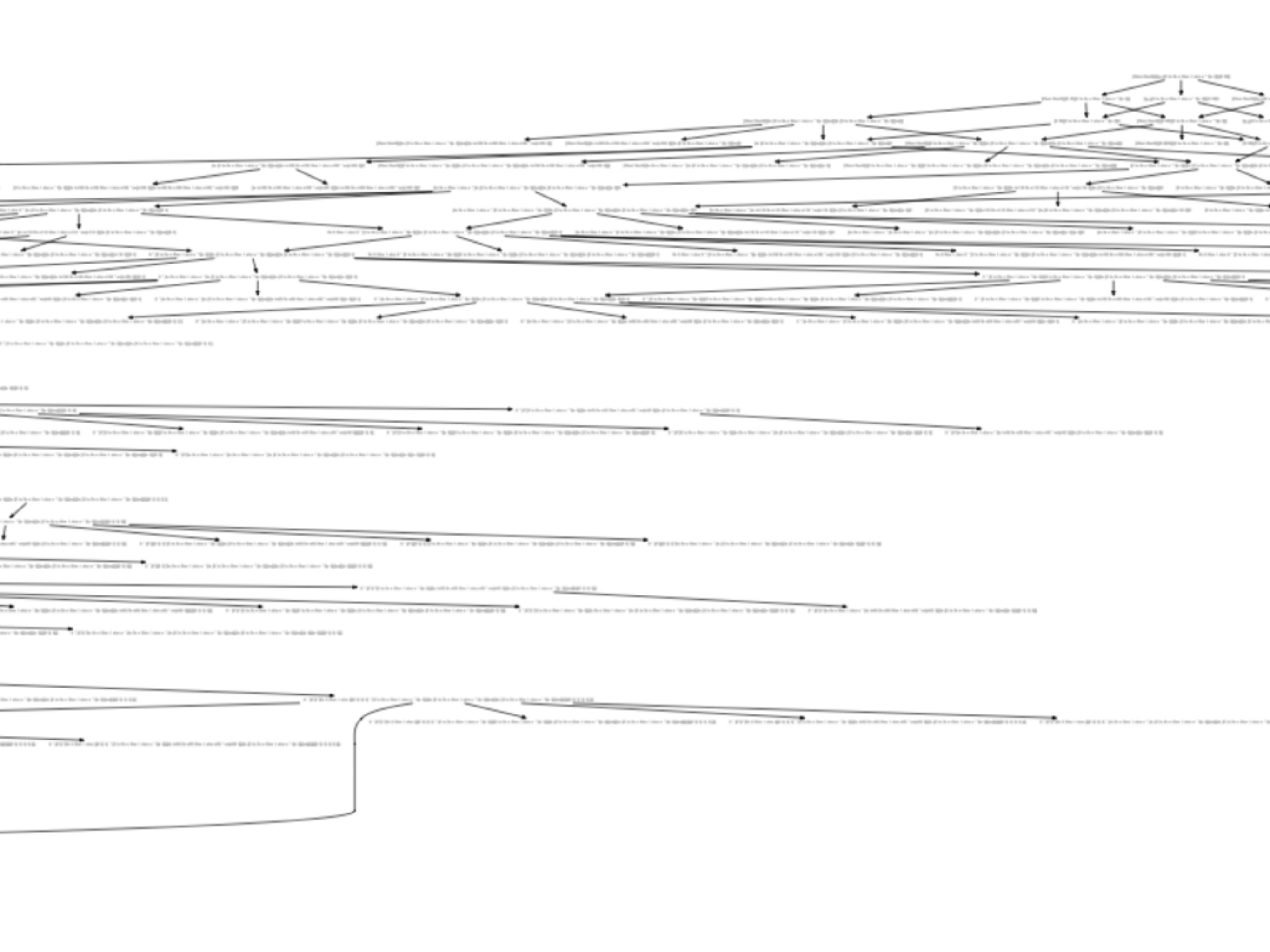


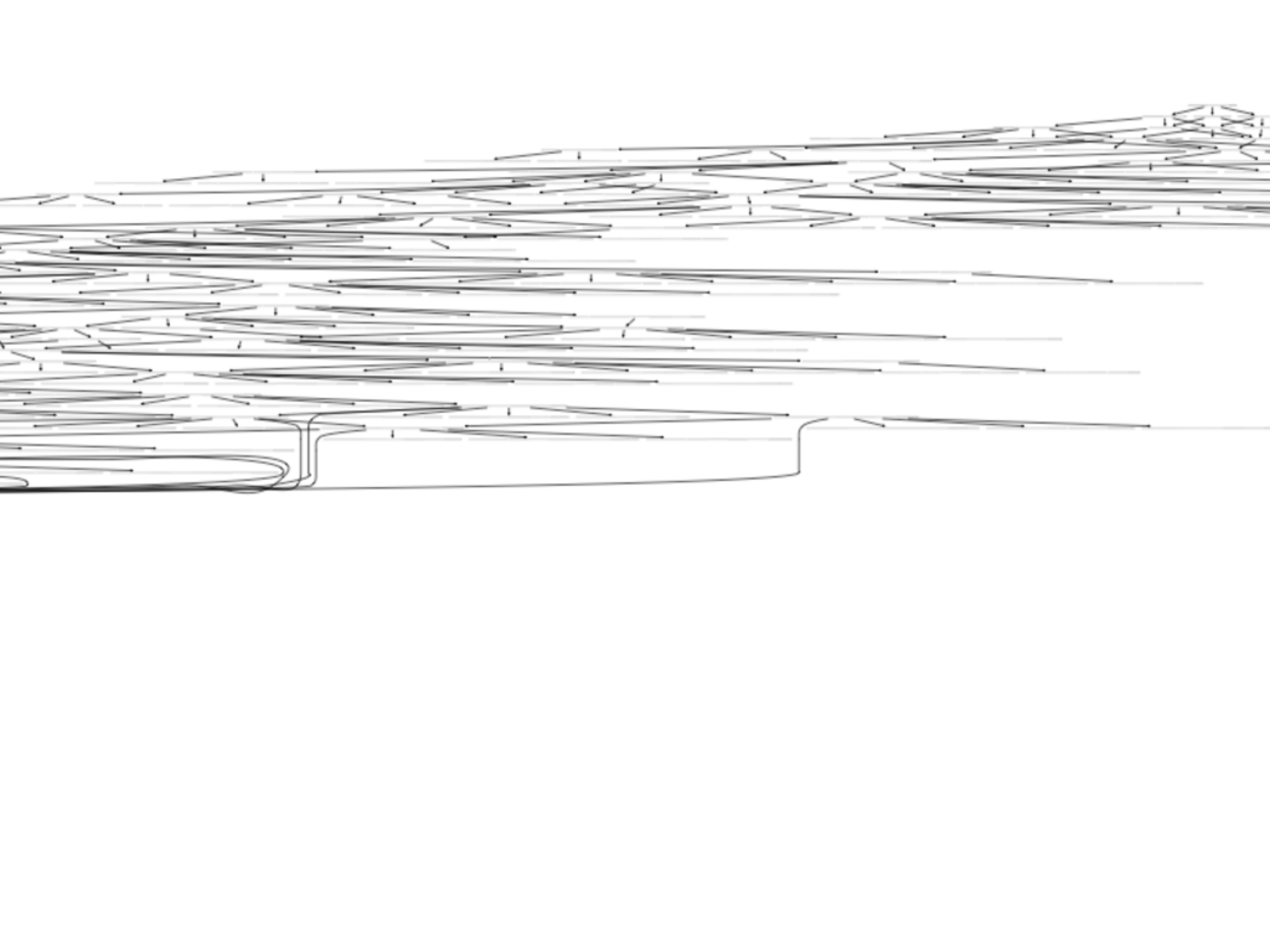
$3 * (\lambda f. \lambda x. \text{ifz } x \text{ then } 1 \text{ else } x * f(x-1))((\lambda x. (\lambda f. \lambda x. \text{ifz } x \text{ then } 1 \text{ else } x * f(x-1))(xx))(\lambda x. (\lambda f. \lambda x. \text{ifz } x \text{ then } 1 \text{ else } x * f(x-1))(xx)))(3-1)$

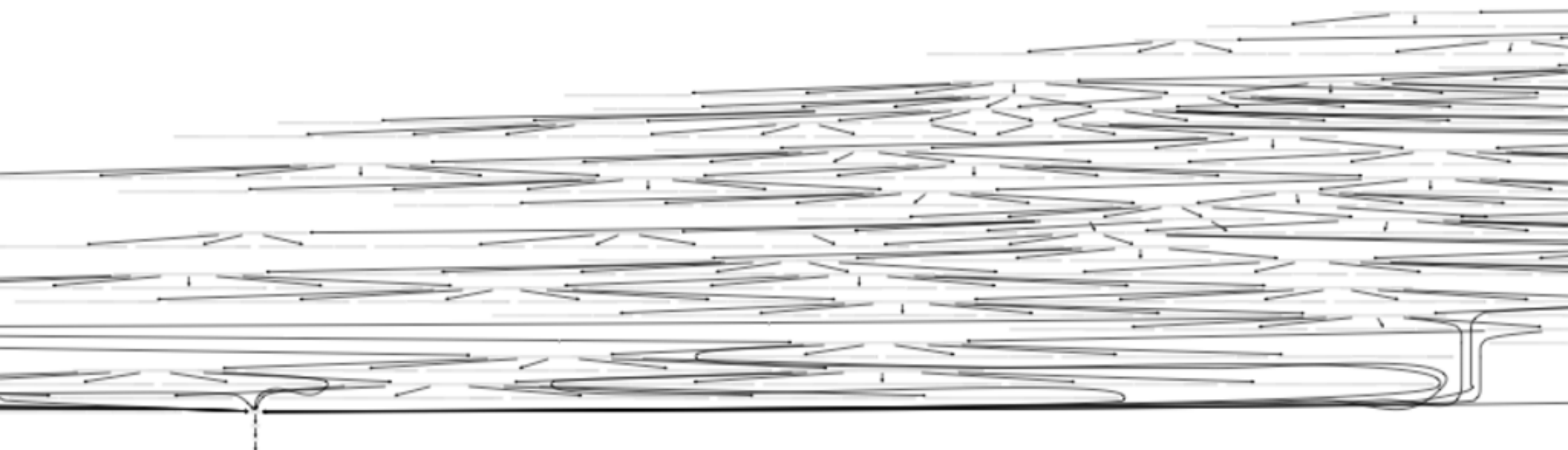


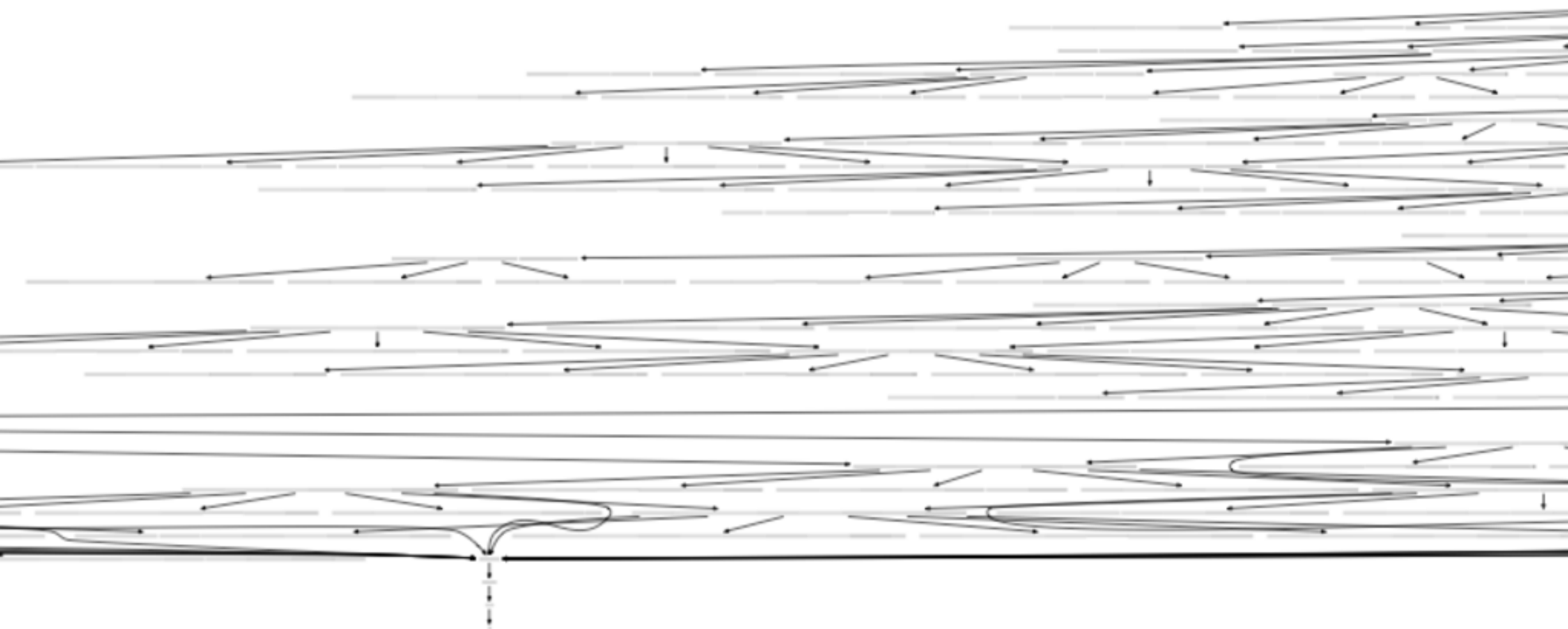


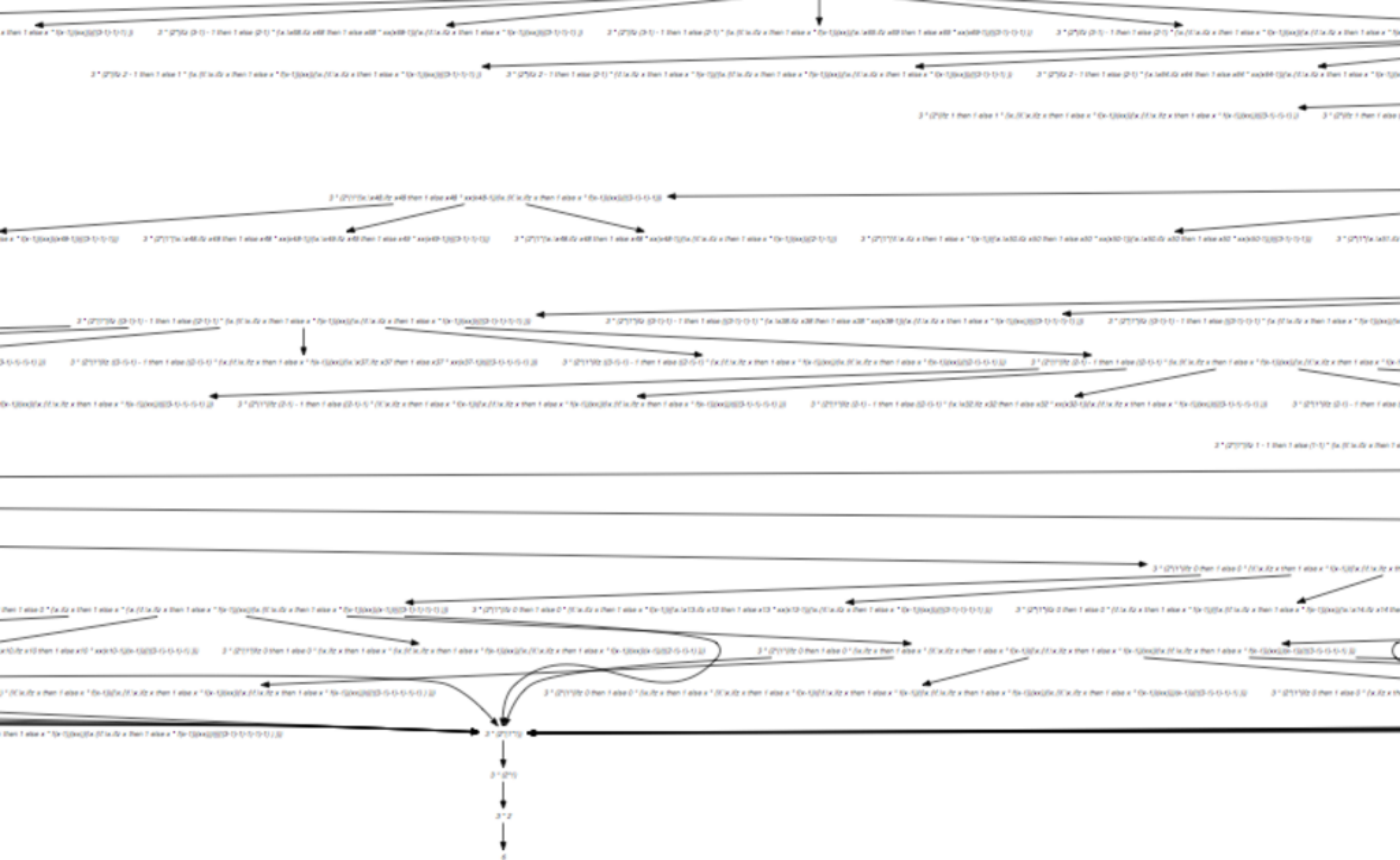


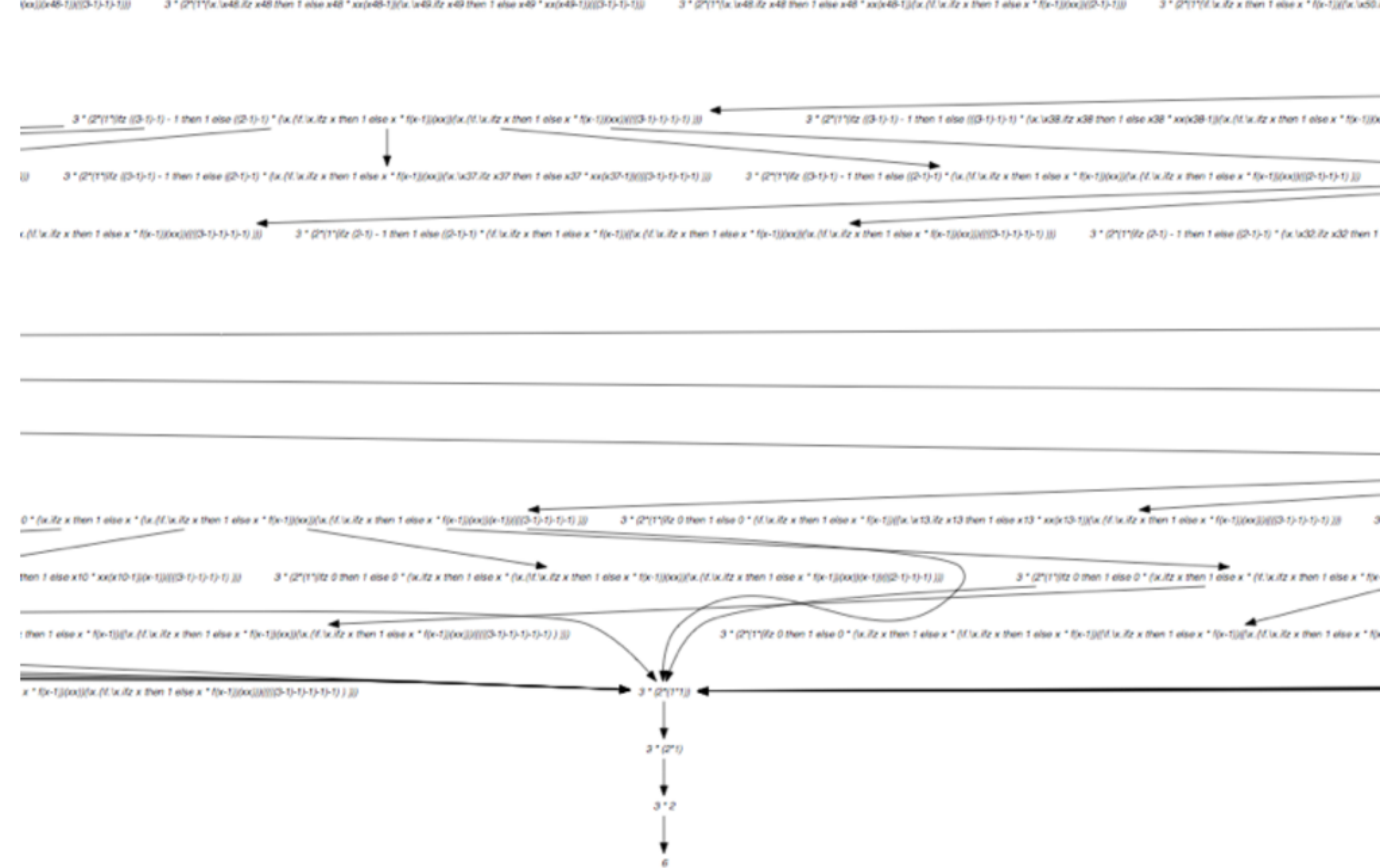


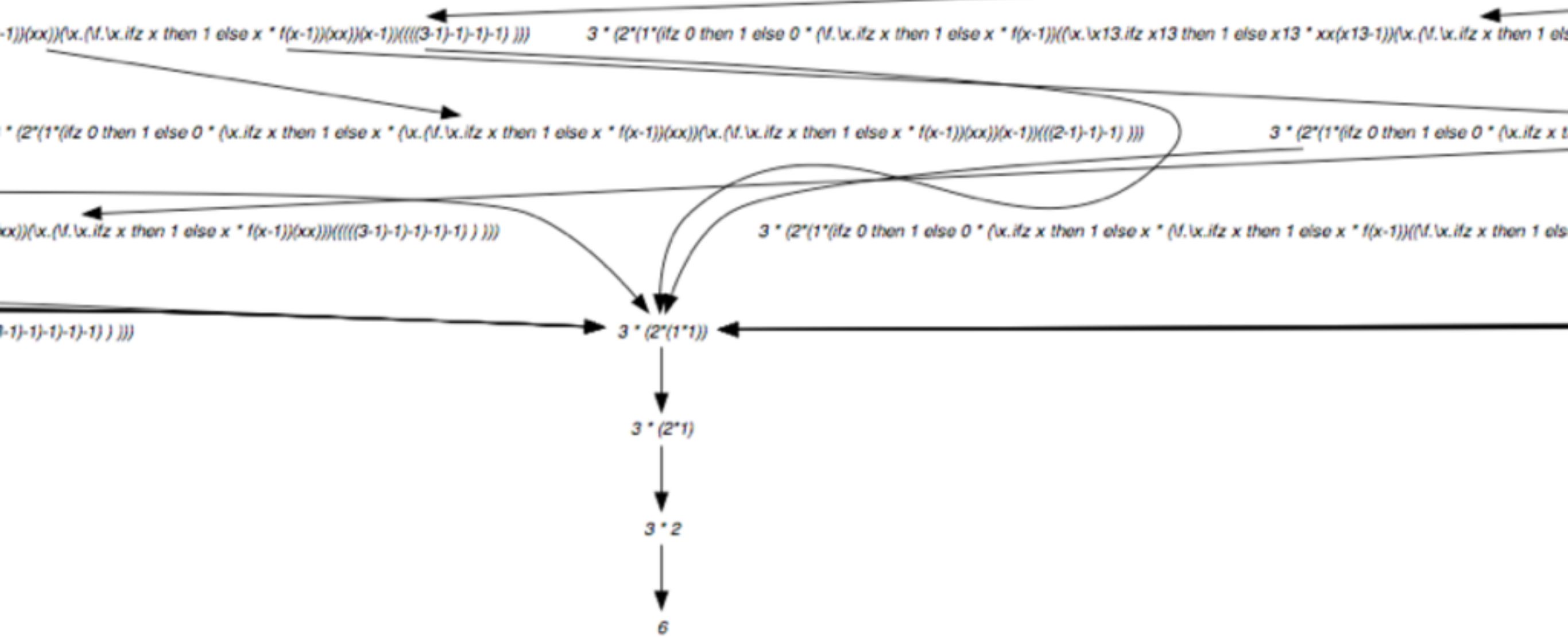












$\lambda x. \text{ifz } x \text{ then } 1 \text{ else } x * f(x-1))(xx))(x-1))(((3-1)-1)-1))))$

$3 * (2 * (1 * (\text{ifz } 0 \text{ then } 1 \text{ else } 0 * (\lambda f. \lambda x. \text{ifz } x \text{ then } 1 \text{ else } x * f(x-1))(\lambda x. \lambda x13. \text{ifz } x13 \text{ then } 1 \text{ else } 0 * (\lambda x. \text{ifz } x \text{ then } 1 \text{ else } x * (\lambda f. \lambda x. \text{ifz } x \text{ then } 1 \text{ else } x * f(x-1))(xx))(\lambda x. (\lambda f. \lambda x. \text{ifz } x \text{ then } 1 \text{ else } x * f(x-1))(xx))(x-1))(((2-1)-1)-1)))))$

$\text{then } 1 \text{ else } 0 * (\lambda x. \text{ifz } x \text{ then } 1 \text{ else } x * (\lambda x. (\lambda f. \lambda x. \text{ifz } x \text{ then } 1 \text{ else } x * f(x-1))(xx))(\lambda x. (\lambda f. \lambda x. \text{ifz } x \text{ then } 1 \text{ else } x * f(x-1))(xx))(x-1))(((2-1)-1)-1))))$

$\text{z } x \text{ then } 1 \text{ else } x * f(x-1))(xx))(((3-1)-1)-1)-1))))$

$3 * (2 * (1 * (\text{ifz } 0 \text{ then } 1 \text{ else } 0 * (\lambda x. \text{ifz } x \text{ then } 1 \text{ else } x * (\lambda f. \lambda x. \text{ifz } x \text{ then } 1 \text{ else } x * f(x-1))(xx))(\lambda x. (\lambda f. \lambda x. \text{ifz } x \text{ then } 1 \text{ else } x * f(x-1))(xx))(x-1))(((2-1)-1)-1)))))$

$)))$

$3 * (2 * (1 * 1))$

$3 * (2 * 1)$

$3 * 2$

6



Exercises

Parallel moves

- **Lemma** $M \xrightarrow{\mathcal{F}} N, M \xrightarrow{\mathcal{G}} P \Rightarrow N \xrightarrow{\mathcal{G}} Q, P \xrightarrow{\mathcal{F}} Q$

Proof

Case 1: $M = x = N = P = Q$. Obvious.

Case 2: $M = \lambda x.M_1, N = \lambda x.N_1, P = \lambda x.P_1$. Obvious by induction on M_1

Case 3: (App-App) $M = M_1M_2, N = N_1N_2, P = P_1P_2$. Obvious by induction on M_1, M_2 .

Case 4: (Red'-Red') $M = (\lambda x.M_1)^a M_2, N = (\lambda x.N_1)^a N_2, P = (\lambda x.P_1)^a P_2, a \notin \mathcal{F} \cup \mathcal{G}$

Then induction on M_1, M_2 .

Case 4: (beta-Red') $M = (\lambda x.M_1)^a M_2, N = N_1\{x := N_2\}, P = (\lambda x.P_1)^a P_2, a \in \mathcal{F}, a \notin \mathcal{G}$

By induction $N_1 \xrightarrow{\mathcal{G}} Q_1, P_1 \xrightarrow{\mathcal{F}} Q_1$. And $N_2 \xrightarrow{\mathcal{G}} Q_2, P_1 \xrightarrow{\mathcal{F}} Q_2$.

By lemma, $N_1\{x := N_2\} \xrightarrow{\mathcal{G}} Q_1\{x := Q_2\}$. And $(\lambda x.P_1)^a P_2 \xrightarrow{\mathcal{F}} Q_1\{x := Q_2\}$

Case 5: (beta-beta) $M = (\lambda x.M_1)^a M_2, N = N_1\{x := N_2\}, P = P_1\{x := P_2\}, a \in \mathcal{F} \cap \mathcal{G}$

As before with same lemma.

Parallel moves

- Lemma $M \xrightarrow{\mathcal{F}} N, P \xrightarrow{\mathcal{F}} Q \Rightarrow M\{x := P\} \xrightarrow{\mathcal{F}} N\{x := Q\}$

Proof: [exercise!](#)

- Lemma [\[subst\]](#) $M\{x := N\}\{y := P\} = M\{y := P\}\{x := N\{y := P\}\}$
when x not free in P

Proof: [exercise!](#)

this lemma about distribution of substitution is critical for the Church-Rosser property.

Finite Developments in the λ -calculus

Part 2

jean-jacques.levy@inria.fr

ISR 2021

Madrid

July 6, 2021



<http://jeanjacqueslevy.net/talks/21isr>



A labeled lambda-calculus ^(1/3)

- Give names to redexes and to (some) subterms
- make names consistent with permutation equivalence.

$$M, N, \dots ::= x \mid MN \mid \lambda x.M \mid M^\alpha$$

- Conversion rule is:

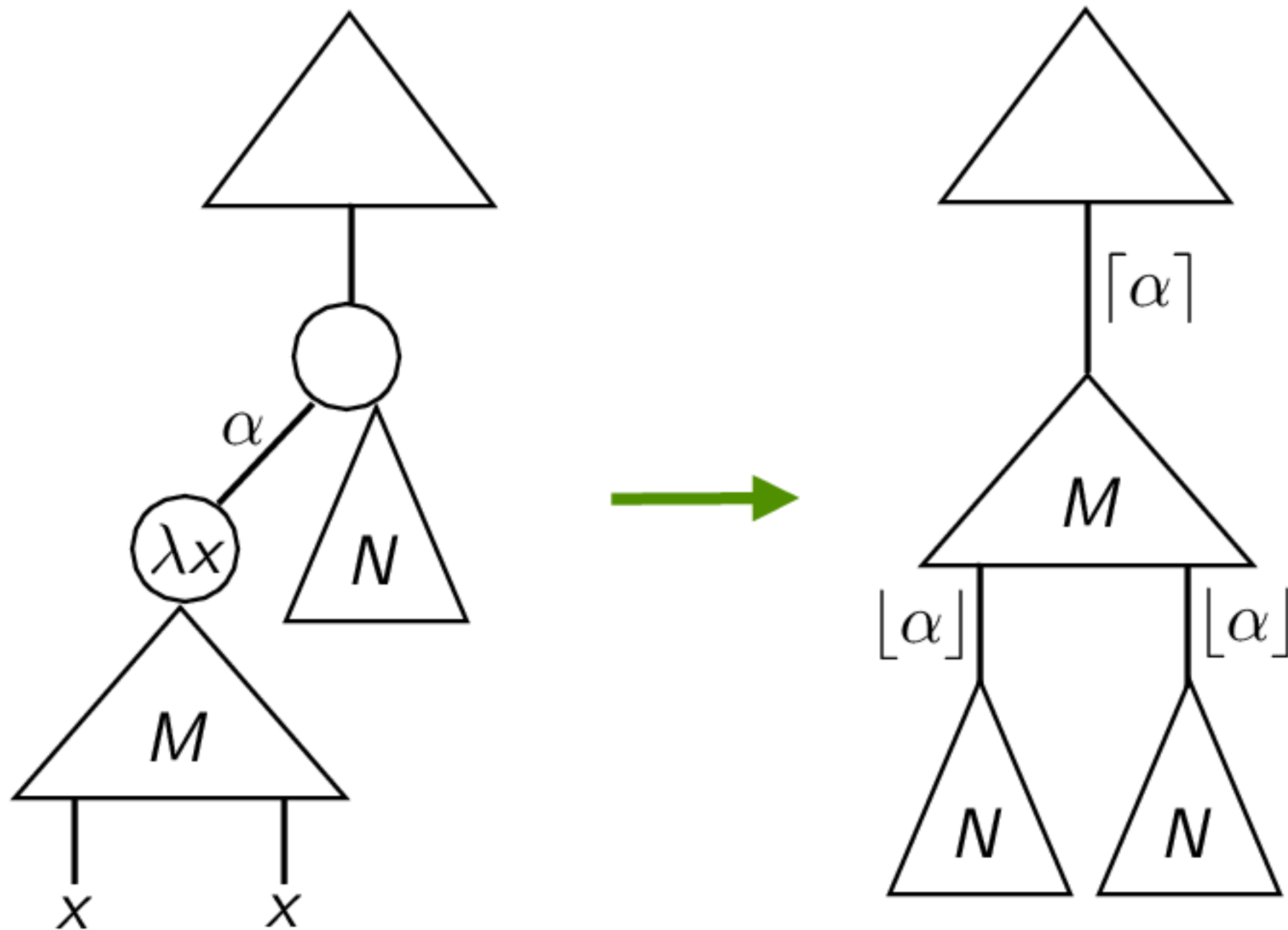
$$(\lambda x.M)^\alpha N \longrightarrow M^{\lceil \alpha \rceil} \{x := N^{\lfloor \alpha \rfloor}\}$$

α is the **name** of that redex

where

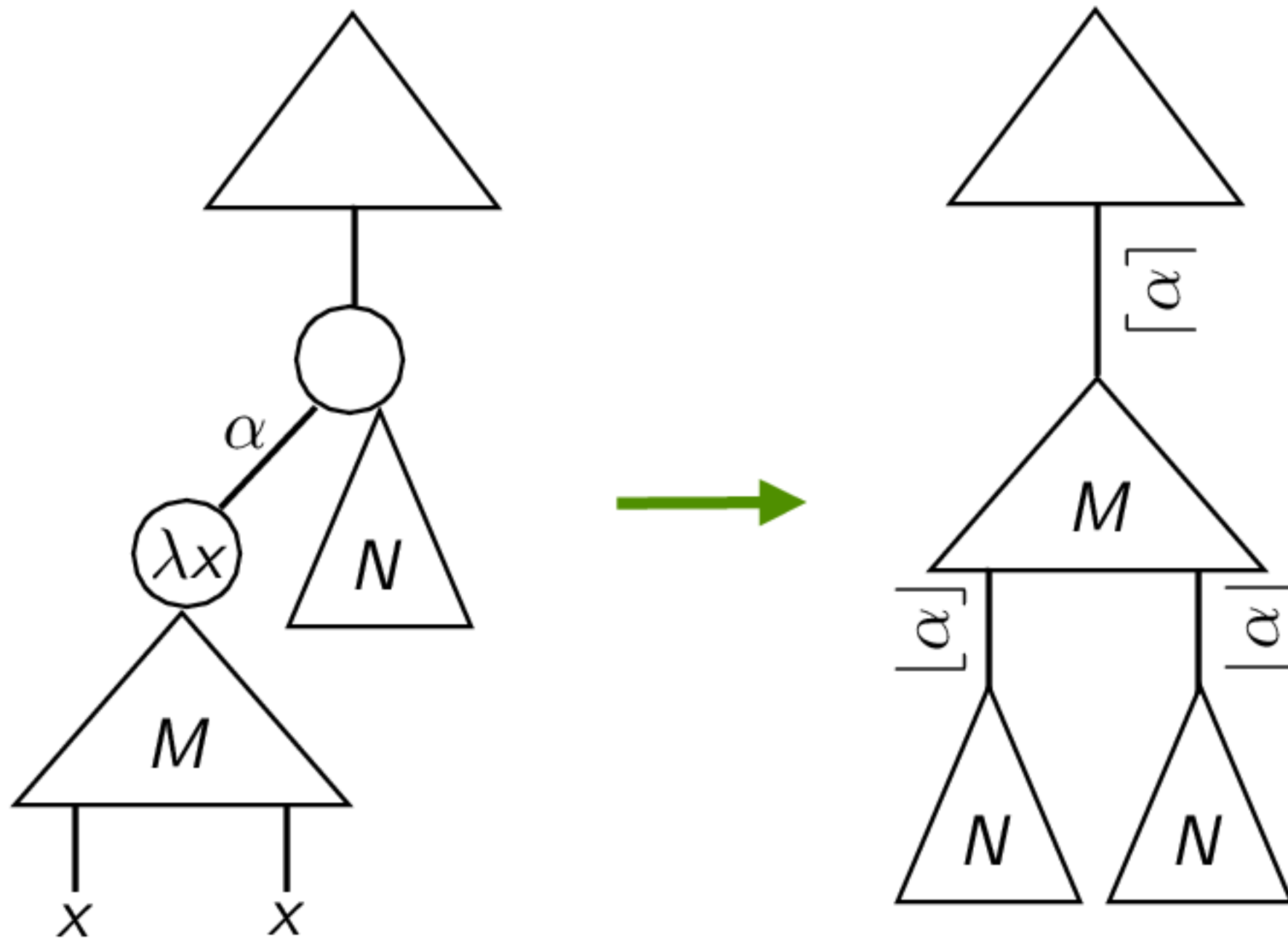
$$(M^\alpha)^\beta = M^{\alpha\beta} \quad \text{and} \quad (M^\alpha)\{x := N\} = (M\{x := N\})^\alpha$$

A labeled lambda-calculus ^(2/3)



abstract syntax trees of labeled λ -terms

A labeled lambda-calculus ^(2/3)



A labeled lambda-calculus (3/3)

- Labels are strings of atomic labels:

$$\alpha, \beta, \dots ::= \underbrace{a, b, c, \dots \mid [\alpha] \mid [\alpha]}_{\text{atomic labels}} \mid \alpha\beta \mid \epsilon$$

- Labels are strings of atomic labels:

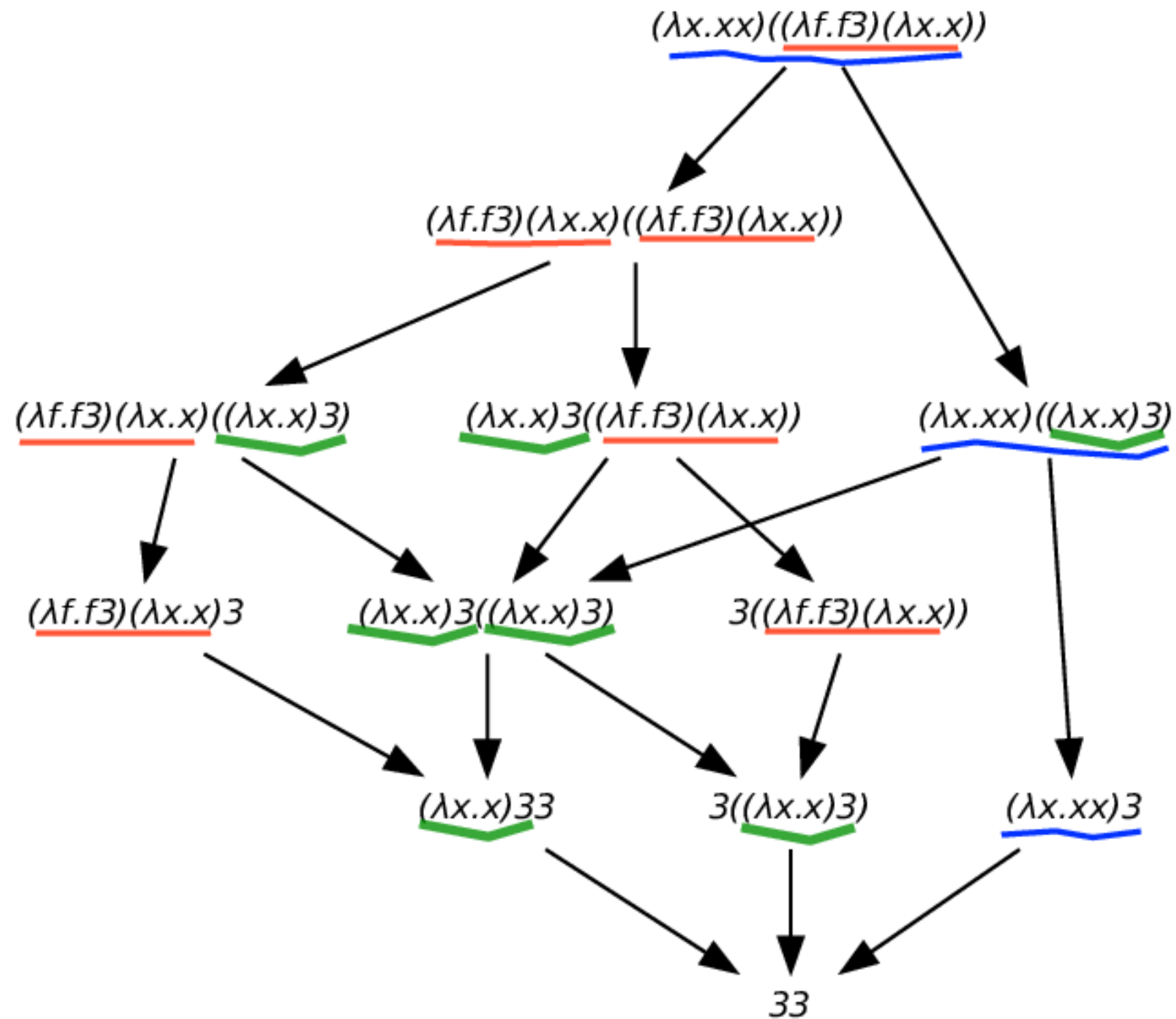
$a, b, c, \dots$ atomic letters

$[\alpha], [\alpha], \dots$ overlined, underlined labels

$\alpha\beta$ compound labels

$\epsilon = [\epsilon] = [\epsilon]$ empty label

Example



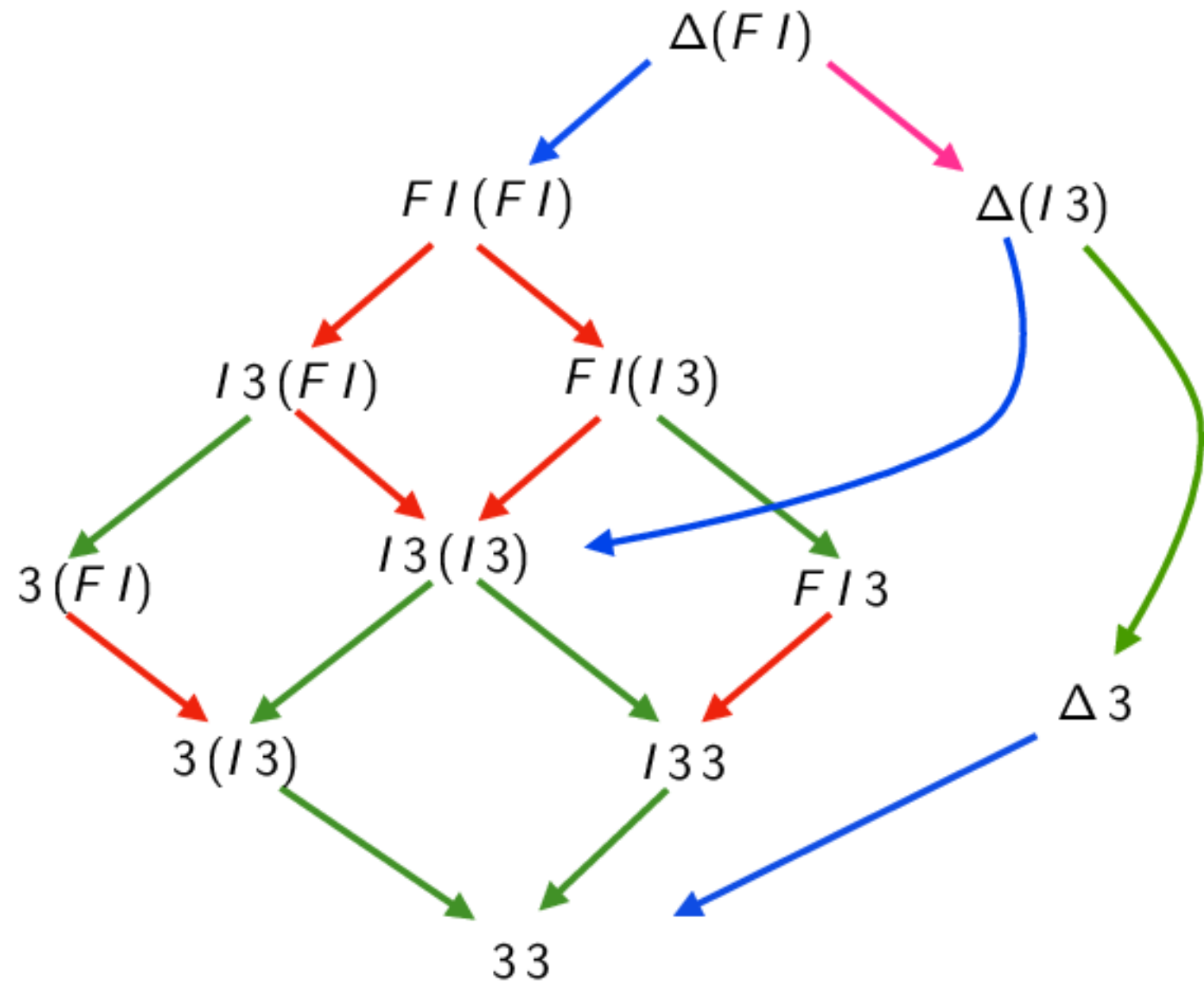
- 3 redex families: **red**, **blue**, **green**.

Example

$$\Delta = \lambda x. x x$$

$$F = \lambda f. f 3$$

$$I = \lambda x. x$$



Example

$$\Delta = \lambda x. (x^c x^d)^b$$

$$F = \lambda f. (f^k 3^\ell)^j$$

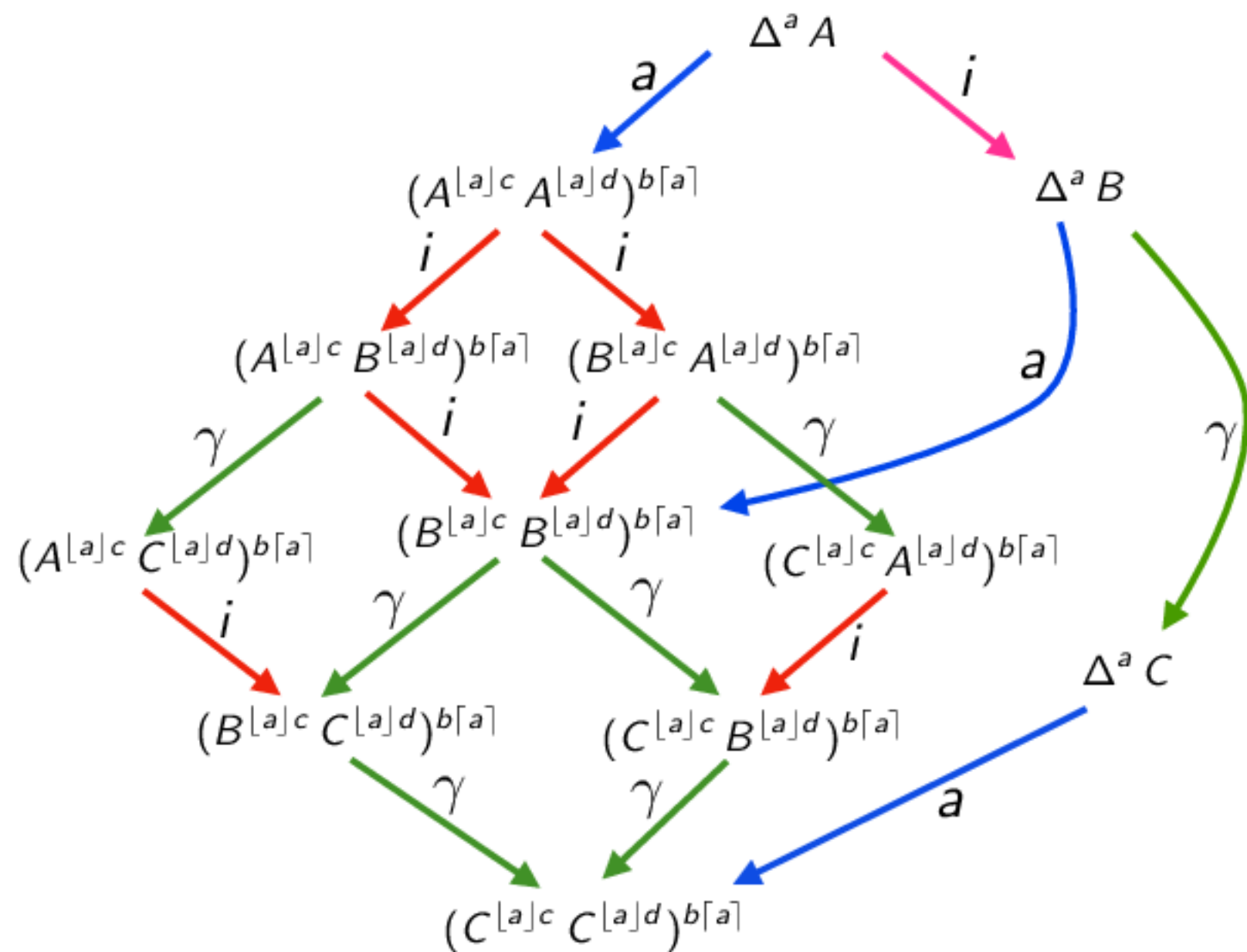
$$I = \lambda x. x^v$$

$$A = (F^i I^u)^q$$

$$B = (I^\gamma 3^\ell)^q$$

$$C = 3^\ell [\gamma] v [\gamma] q$$

$$\gamma = u[i]k$$



3 redexes names: $a, i, \gamma = u[i]k$

Example

$$\begin{array}{c}
 \Omega = D^a \Delta^e \\
 \downarrow a \\
 \Omega_1 = (\Delta^{\gamma_1} \Delta^{\delta_1})^{b[a]} \\
 \downarrow \gamma_1 \\
 \Omega_2 = (\Delta^{\gamma_2} \Delta^{\delta_2})^{f[\gamma_1]b[a]} \\
 \downarrow \gamma_2 \\
 \Omega_3 = (\Delta^{\gamma_3} \Delta^{\delta_3})^{f[\gamma_2]f[\gamma_1]b[a]} \\
 \downarrow \gamma_3 \\
 \Omega_4 = (\Delta^{\gamma_4} \Delta^{\delta_4})^{f[\gamma_3]f[\gamma_2]f[\gamma_1]b[a]} \\
 \downarrow \gamma_4
 \end{array}$$

redexes names: $a, \gamma_1, \gamma_2, \gamma_3, \dots$

$$D = \lambda x. (x^c x^d)^b$$

$$\Delta = \lambda x. (x^g x^h)^f$$

$$\gamma_1 = e[a]c$$

$$\gamma_2 = \delta_1[\gamma_1]g$$

$$\gamma_3 = \delta_2[\gamma_2]g$$

$$\gamma_4 = \delta_3[\gamma_3]g$$

$$\delta_1 = e[a]d$$

$$\delta_2 = \delta_1[\gamma_1]h$$

$$\delta_3 = \delta_2[\gamma_2]h$$

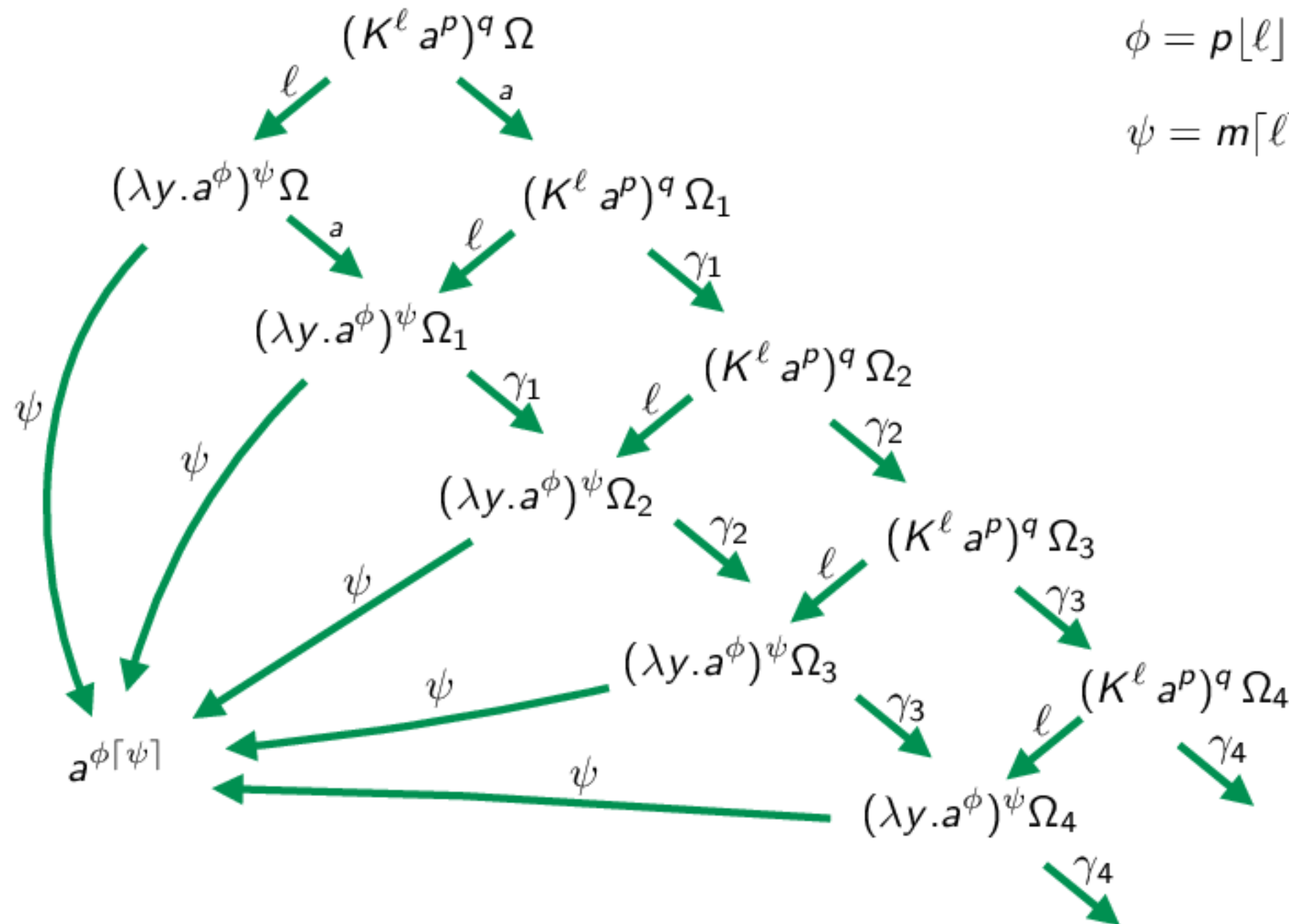
$$\delta_4 = \delta_2[\gamma_2]h$$

Example

$$K = \lambda x. (\lambda y. x^n)^m$$

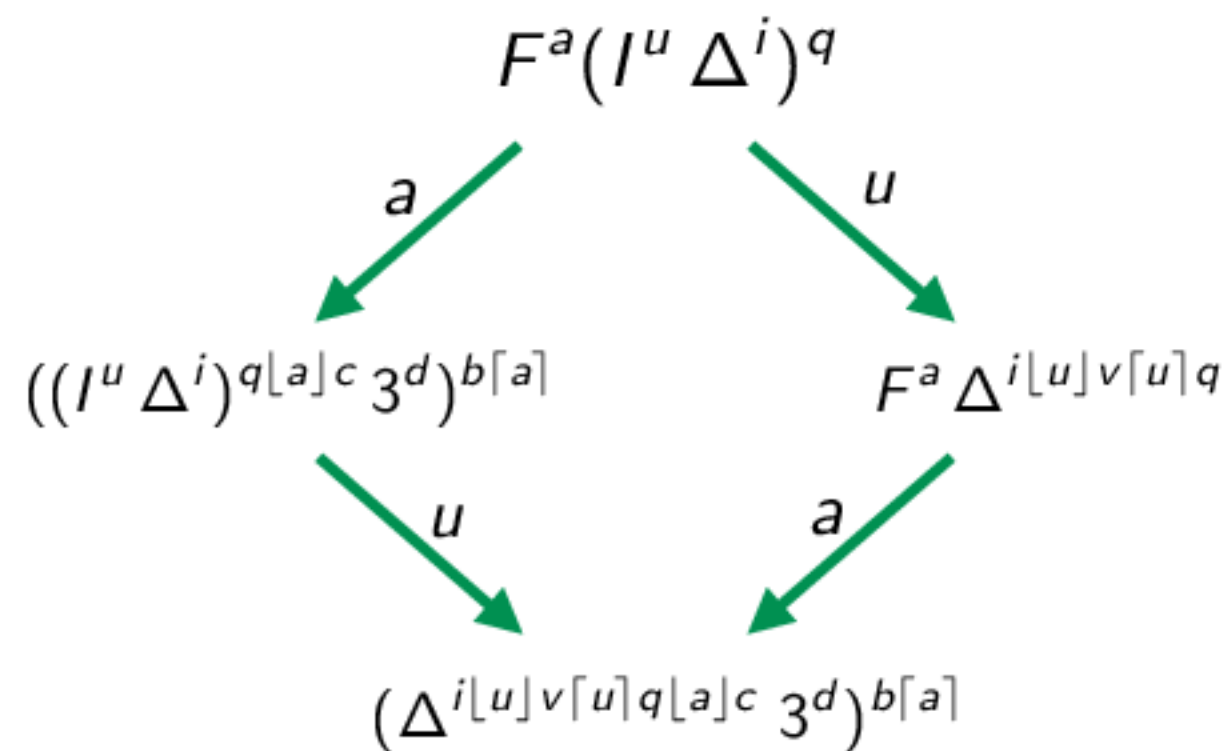
$$\phi = p[\ell]n$$

$$\psi = m[\ell]q$$



redexes names: $\ell, \psi, a, \gamma_1, \gamma_2, \gamma_3, \dots$

Example



$$F = \lambda f.(f^c \ 3^d)^b$$

$$I = \lambda x.x^v$$

$$\Delta = \lambda x.(x^k \ x^\ell)^j$$

2 independent redexes a and u creates the new one $i[u]v[u]q[a]c$

Empirical facts (bis)

- **deterministic** result when it exists

Church-Rosser

- multiple reduction strategies

- **terminating** strategy ?

- **efficient** reduction strategy ?

optimal reduction

- **worst** reduction strategy ?

- when all reductions are finite ?

strong normalisation

- when finite, the reduction graph has a **lattice** structure ?

YES!

Permutation equivalence (1/7)

- **Proposition** [residuals of labeled redexes]

$S \in R/\rho$ implies $\text{name}(R) = \text{name}(S)$

- **Definition** [created redexes] Let $\rho : M \xrightarrow{\star} N$
we say that ρ **creates** R in M when $\nexists R', R \in R'/\rho$.

- **Proposition** [created labeled redexes]

If S creates R , then $\text{name}(S)$ is strictly contained in $\text{name}(R)$.

Permutation equivalence (2/7)

Proof (cont'd) Created redexes contains names of creator

$$\underbrace{(\lambda x. \dots (x^\beta N) \dots)^\alpha (\lambda y. M)^\gamma}_{\alpha} \rightarrow \dots \underbrace{((\lambda y. M)^{\gamma[\alpha]\beta} N') \dots}_{\gamma[\alpha]\beta}$$

creates

$$\underbrace{((\lambda x. (\lambda y. M)^\gamma)^\alpha N)^\beta P}_{\alpha} \rightarrow \underbrace{(\lambda y. M')^{\gamma[\alpha]\beta} P}_{\gamma[\alpha]\beta}$$

creates

$$\underbrace{((\lambda x. x^\gamma)^\alpha (\lambda y. M)^\delta)^\beta N}_{\alpha} \rightarrow \underbrace{(\lambda y. M)^{\delta[\alpha]\gamma[\alpha]\beta} N}_{\delta[\alpha]\gamma[\alpha]\beta}$$

creates

Permutation equivalence (3/7)

- **Labeled laws** $M^\alpha \{x := N\} = (M\{x := N\})^\alpha \quad (M^\alpha)^\beta = M^{\alpha\beta}$

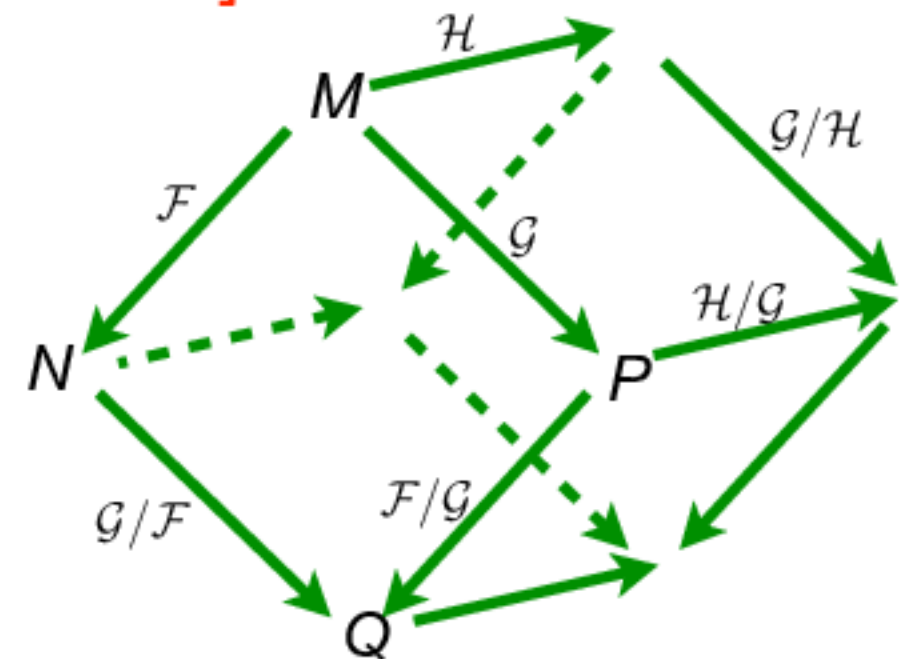
If $M \xrightarrow{\quad} N$, then $M^\alpha \xrightarrow{\quad} N^\alpha$

- **Labeled parallel moves lemma+** [74]

If $M \xrightarrow{\mathcal{F}} N$ and $M \xrightarrow{\mathcal{G}} P$, then $N \xrightarrow{\mathcal{G}/\mathcal{F}} Q$ and $P \xrightarrow{\mathcal{F}/\mathcal{G}} Q$ for some Q .

- **Parallel moves lemma++** [The Cube Lemma]

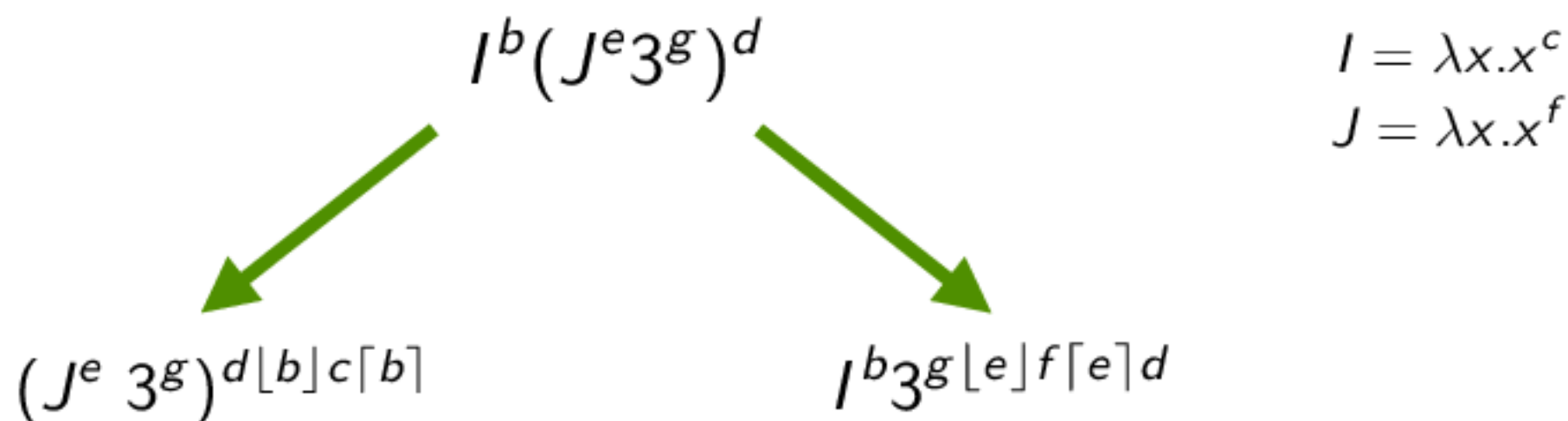
still holds.



Permutation equivalence (4/7)

- Labels do not break Church-Rosser, nor residuals
- Labels refine λ -calculus:
 - any unlabeled reduction can be performed in the labeled calculus
 - but two cofinal unlabeled reductions may no longer be cofinal

Take $I(I3)$ with $I = \lambda x.x$.



Permutation equivalence (5/7)

- **Definition** [pure labeled calculus]

Pure labeled terms are labeled terms where all subterms have non empty labels.

- **Theorem** [labeled permutation equivalence, 76]

Let ρ and σ be coinitial pure labeled reductions.

Then $\rho \simeq \sigma$ iff ρ and σ are labeled cofinal.

Proof Let $\rho \simeq \sigma$. Then obvious because of labeled parallel moves lemma.

Conversely, we apply standardization thm and following lemma.

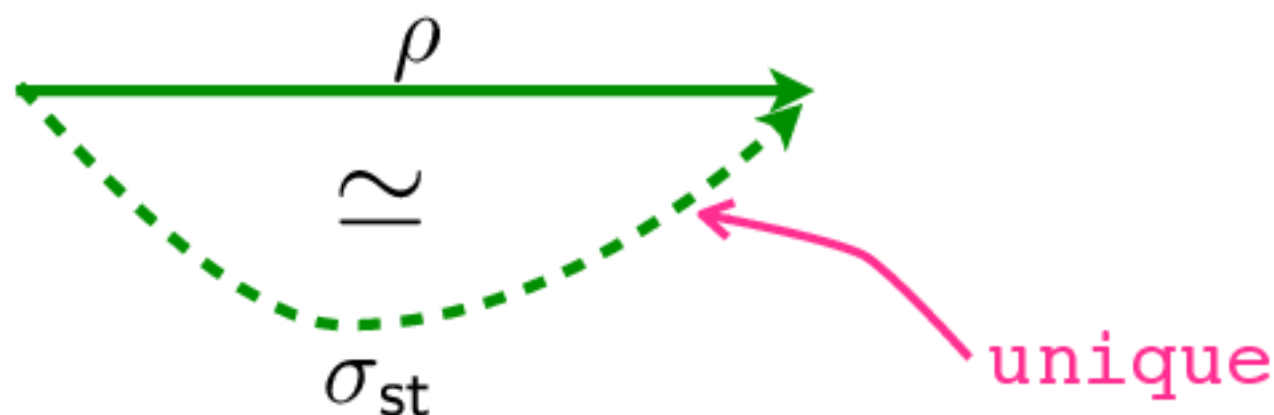
Permutation equivalence (6/7)

- **Definition:** The following reduction is **standard**

$$\rho : M = M_0 \xrightarrow{R_1} M_1 \xrightarrow{R_2} M_2 \cdots \xrightarrow{R_n} M_n = N$$

iff for all i and j , $i < j$, then R_j is not residual along ρ of some R'_j to the left of R_i in M_{i-1} .

- **Standardization** [Curry 50] Let $M \xrightarrow{\star} N$. Then $M \xrightarrow{\text{st}} N$.
- **Labeled standardization** $\forall \rho, \exists! \sigma_{\text{st}}, \rho \simeq \sigma_{\text{st}}$



Permutation equivalence (7/7)

- **Notation** [prefix ordering] $\rho \sqsubseteq \sigma$ for $\exists \tau. \rho \tau \simeq \sigma$
- **Corollary** [labeled prefix ordering]
Let $\rho : M \xrightarrow{\star} N$ and $\sigma : M \xrightarrow{\star} P$ be coinitial pure labeled reductions.
Then $\rho \sqsubseteq \sigma$ iff $N \xrightarrow{\star} P$.
- **Corollary** [lattice of labeled reductions]
Labeled reduction graphs are upwards semi lattices for any pure labeling.

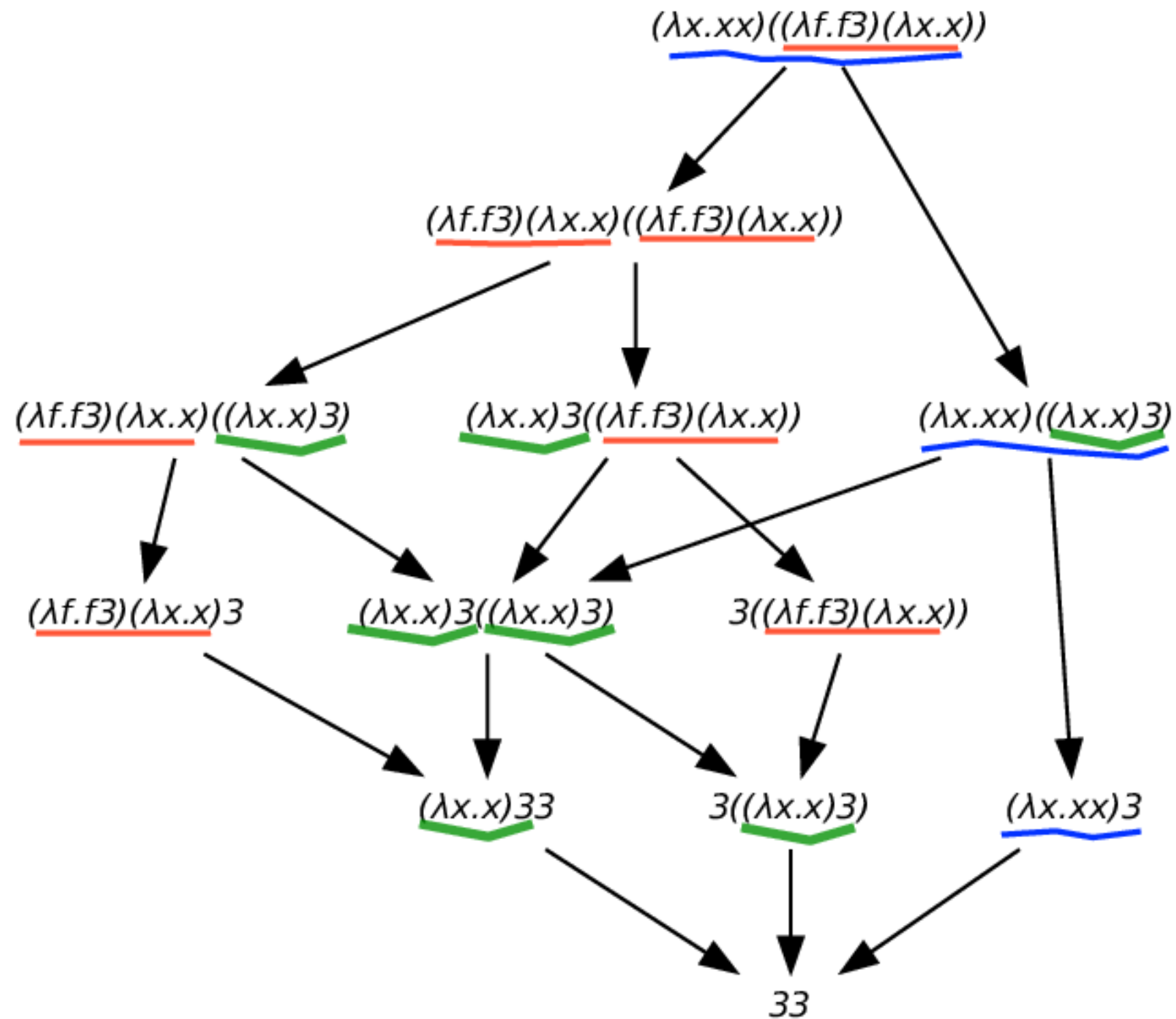
In other terms, reductions up-to permutation equivalence is a push-out category.

Exercise Try on $(\lambda x.x)((\lambda y.(\lambda x.x)a)b)$ or $(\lambda x.xx)(\lambda x.xx)$



Redex families

Example



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

hRedexes

- **Definition** [hRedex]

hRedex is a pair $\langle \rho, R \rangle$ where R is a redex in final term of ρ

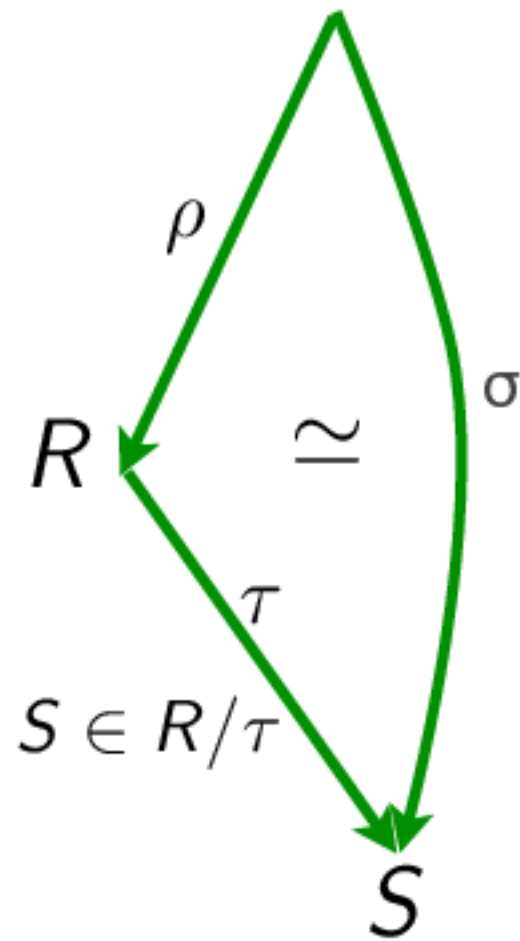
- **Definition** [copies of hRedex]

$\langle \rho, R \rangle \leq \langle \sigma, S \rangle$ when $\exists \tau. \rho\tau \simeq \sigma$ and $S \in R/\tau$

- **Definition** [families of hRedexes]

$\langle \rho, R \rangle \sim \langle \sigma, S \rangle$ for reflexive, symmetric, transitive closure of the copy relation.

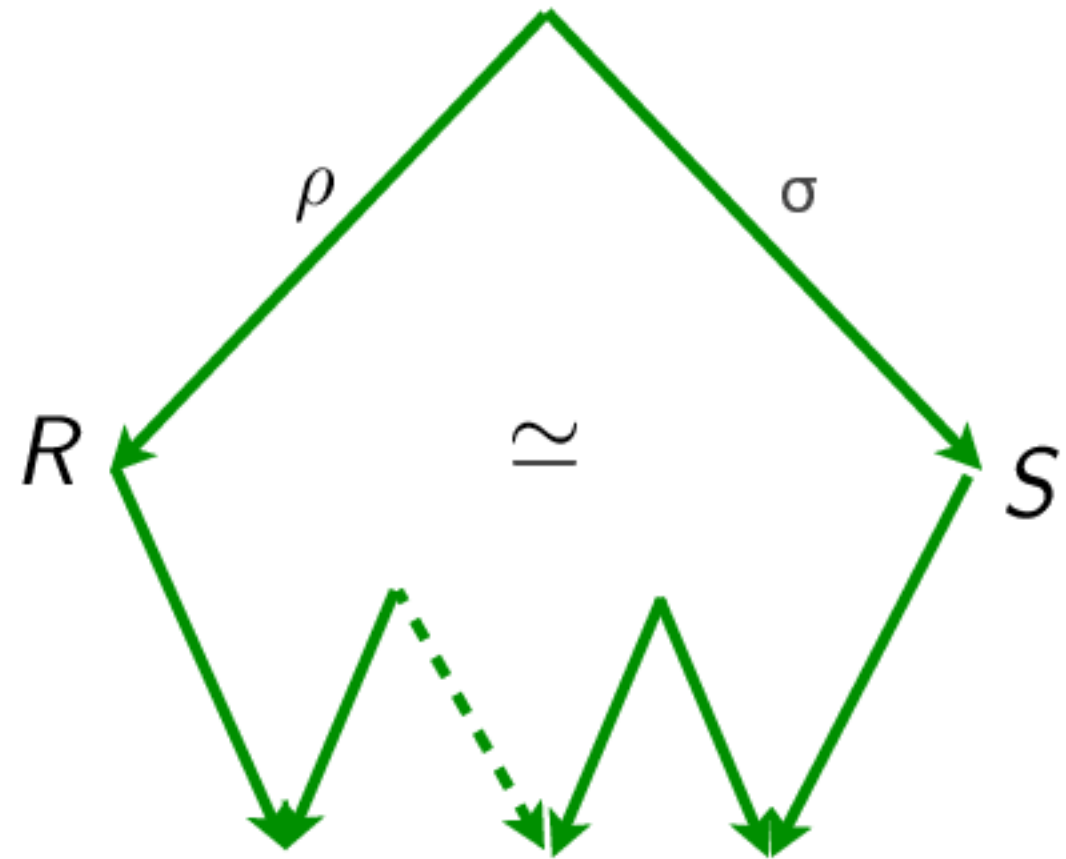
Labels and history (1/4)



$$\langle \rho, R \rangle \leq \langle \sigma, S \rangle$$



$$\text{name}(R) = \text{name}(S)$$



$$\langle \rho, R \rangle \sim \langle \sigma, S \rangle$$



$$\text{name}(R) = \text{name}(S)$$

Labels and history (2/4)

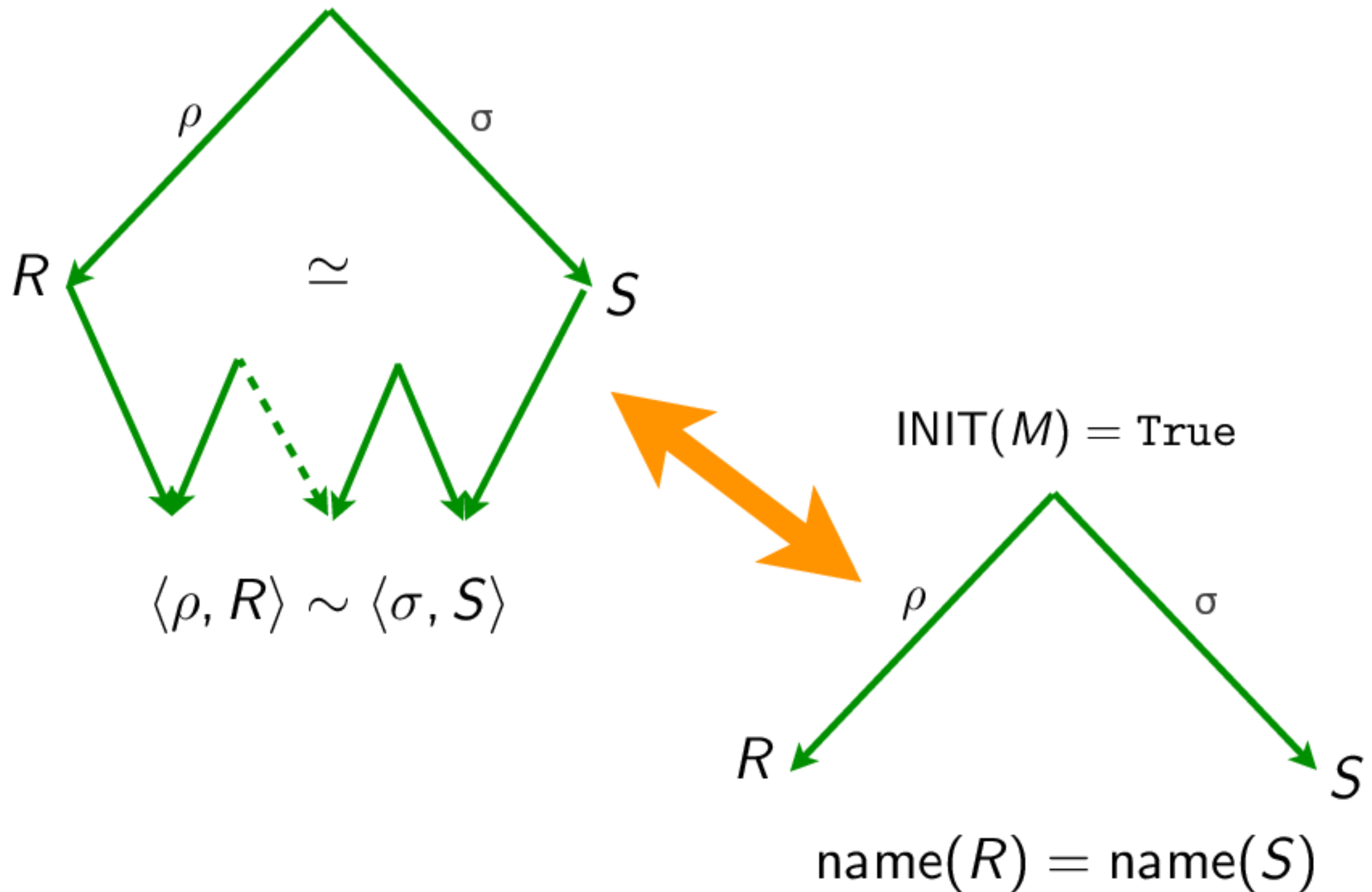
- **Proposition** [same history $\rightarrow$ same name]

In the labeled λ -calculus, for any labeling, we have:

$$\langle \rho, R \rangle \sim \langle \sigma, S \rangle \text{ implies } \text{name}(R) = \text{name}(S)$$

- The opposite direction is clearly not true for any labeling
(For instance, take all labels equal)
- But it is true when all labels are distinct atomic letters in the initial term.
- **Definition** [all labels distinct letters]
 $\text{INIT}(M) = \text{True}$ when all labels in M are distinct letters.

Labels and history (3/4)



Labels and history (4/4)

- **Theorem** [same history = same name, 76]

When $\text{INIT}(M)$ and reductions ρ and σ start from M :

$$\langle \rho, R \rangle \sim \langle \sigma, S \rangle \text{ iff } \text{name}(R) = \text{name}(S)$$

- **Corollary** [decidability of family relation]

The family relation is decidable (although complexity is proportional to length of standard reduction).



Finite developments

Parallel steps revisited (1/3)

- parallel steps were defined with inside-out strategy
[à la Martin-Löf]

Can we take any order as a reduction strategy ?

- **Definition** A **reduction relative** to a set $\mathcal{F}$ of redexes in M is any reduction contracting only residuals of $\mathcal{F}$.
A **development** of $\mathcal{F}$ is any maximal relative reduction of $\mathcal{F}$.

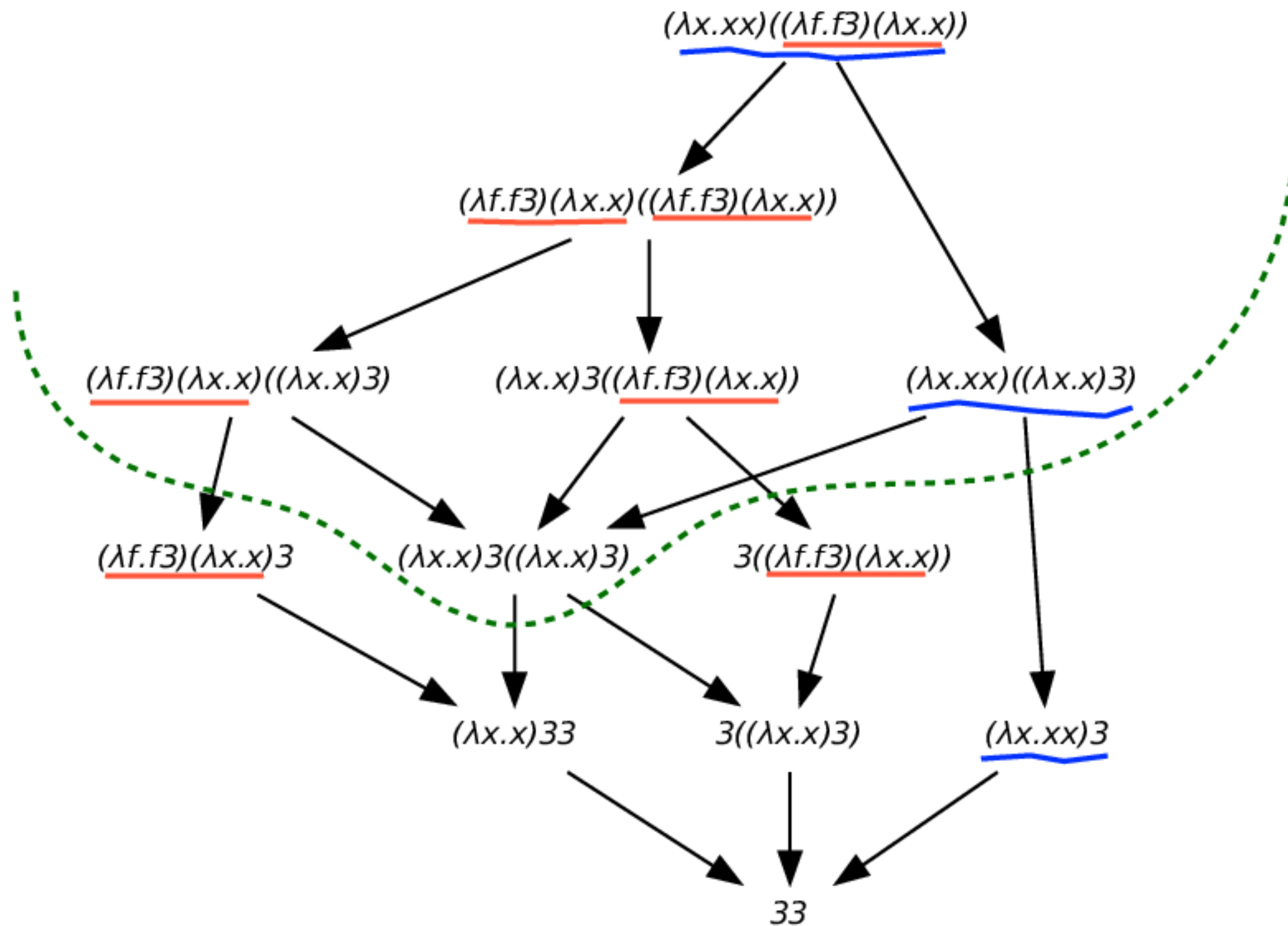
Parallel steps revisited (2/3)

- **Theorem** [Finite Developments, Curry, 50]

Let $\mathcal{F}$ be set of redexes in M .

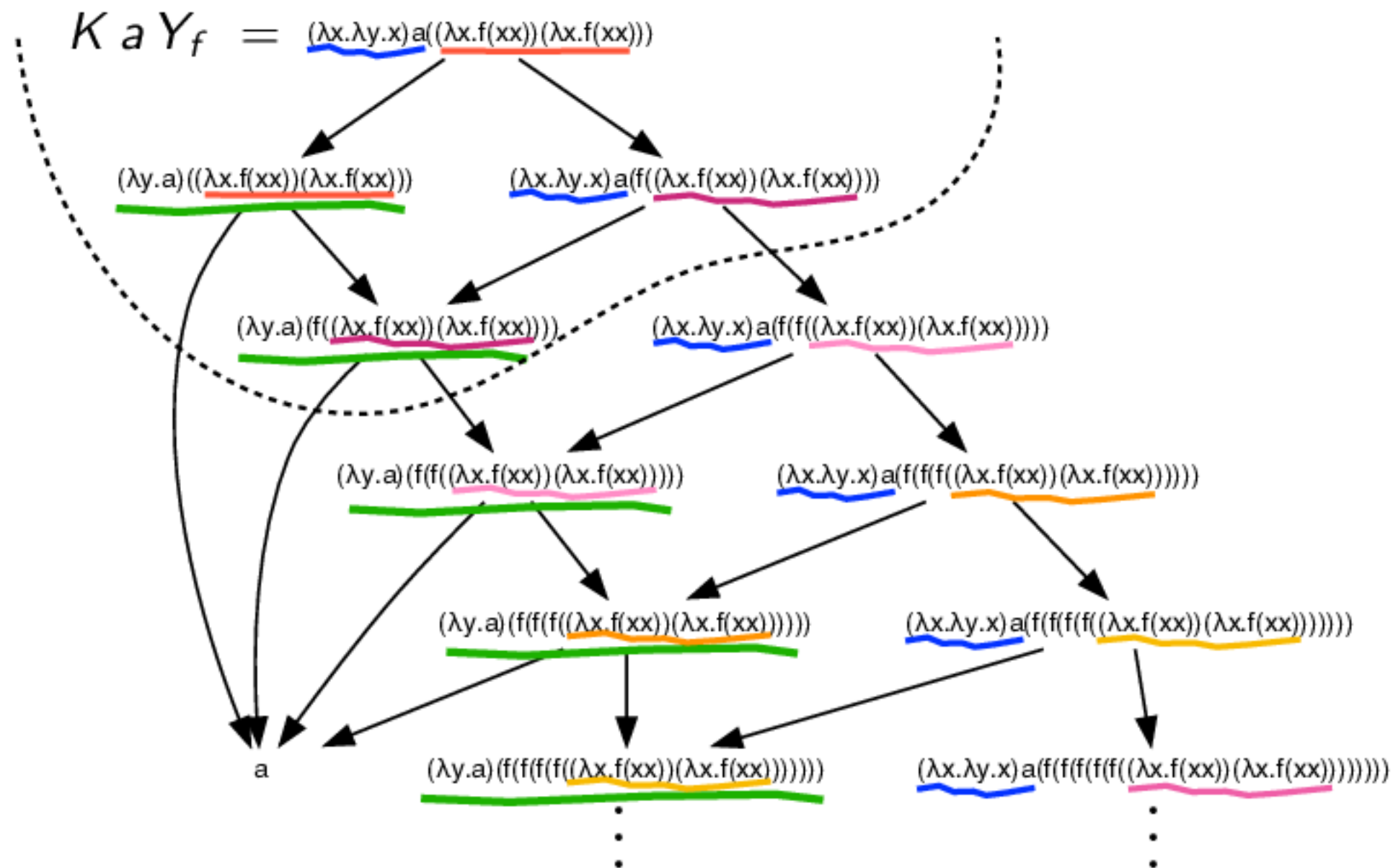
- (1) there are no infinite relative reductions of $\mathcal{F}$,
 - (2) they all finish on same term N
 - (3) Let R be redex in M . Residuals of R by all finite developments of $\mathcal{F}$ are the same.
- Similar to the parallel moves lemma, but we considered a particular inside-out reduction strategy.

Example



developments of **red**, **blue**.

Example



developments of red, blue.

Parallel steps revisited (3/3)

- **Notation** [parallel reduction steps]

Let $\mathcal{F}$ be set of redexes in M . We write $M \xrightarrow{\mathcal{F}} N$
if a development of $\mathcal{F}$ connects M to N .

- This notation is consistent with previous definition
(since inside-out parallel step is a particular development)
- Corollaries of FD thm are also parallel moves + cube lemmas

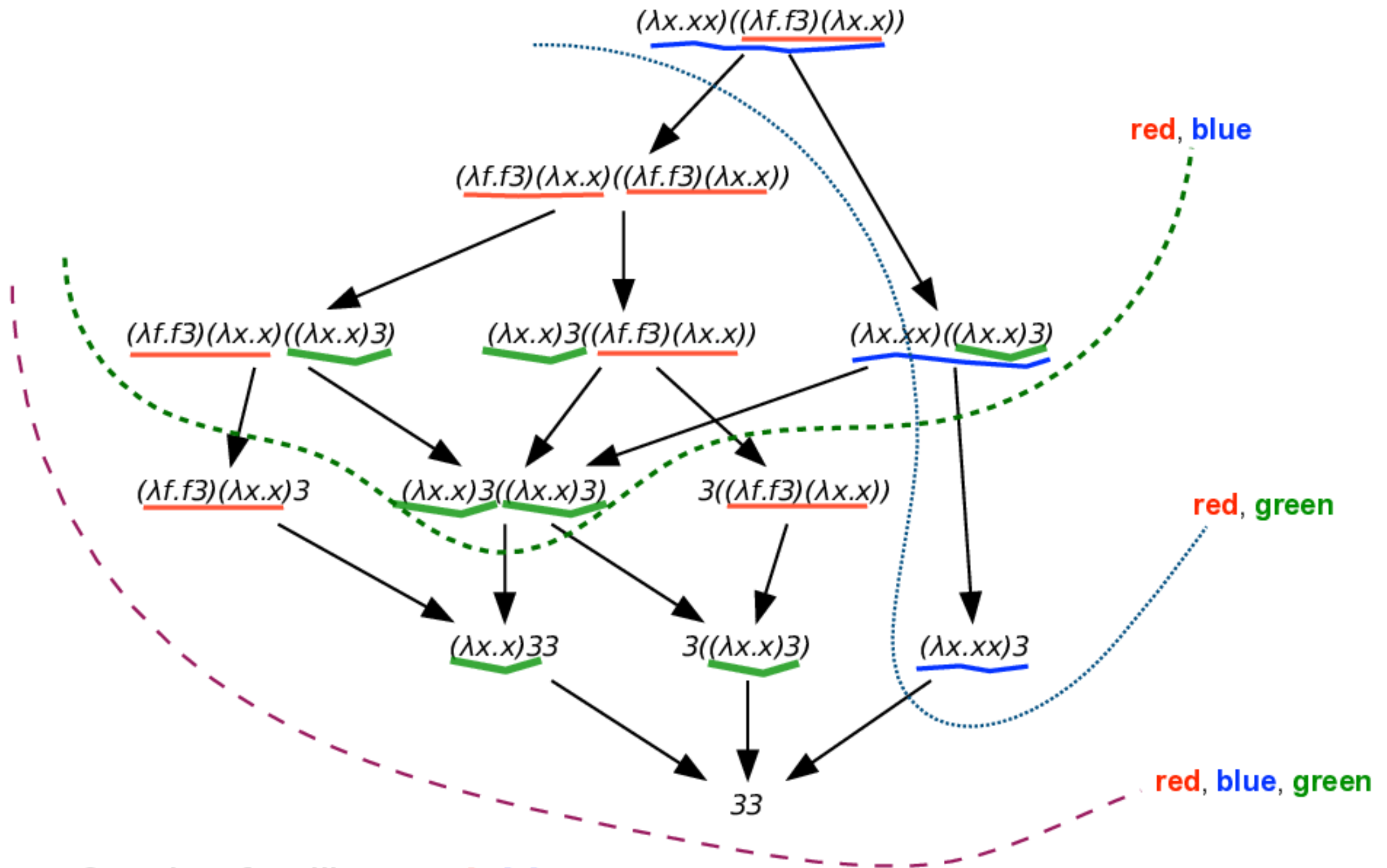
Finite and infinite reductions (1/3)

- **Definition** A **reduction relative** to a set $\mathcal{F}$ of redex families is any reduction contracting redexes in families of $\mathcal{F}$.

A **development** of $\mathcal{F}$ is any maximal relative reduction.

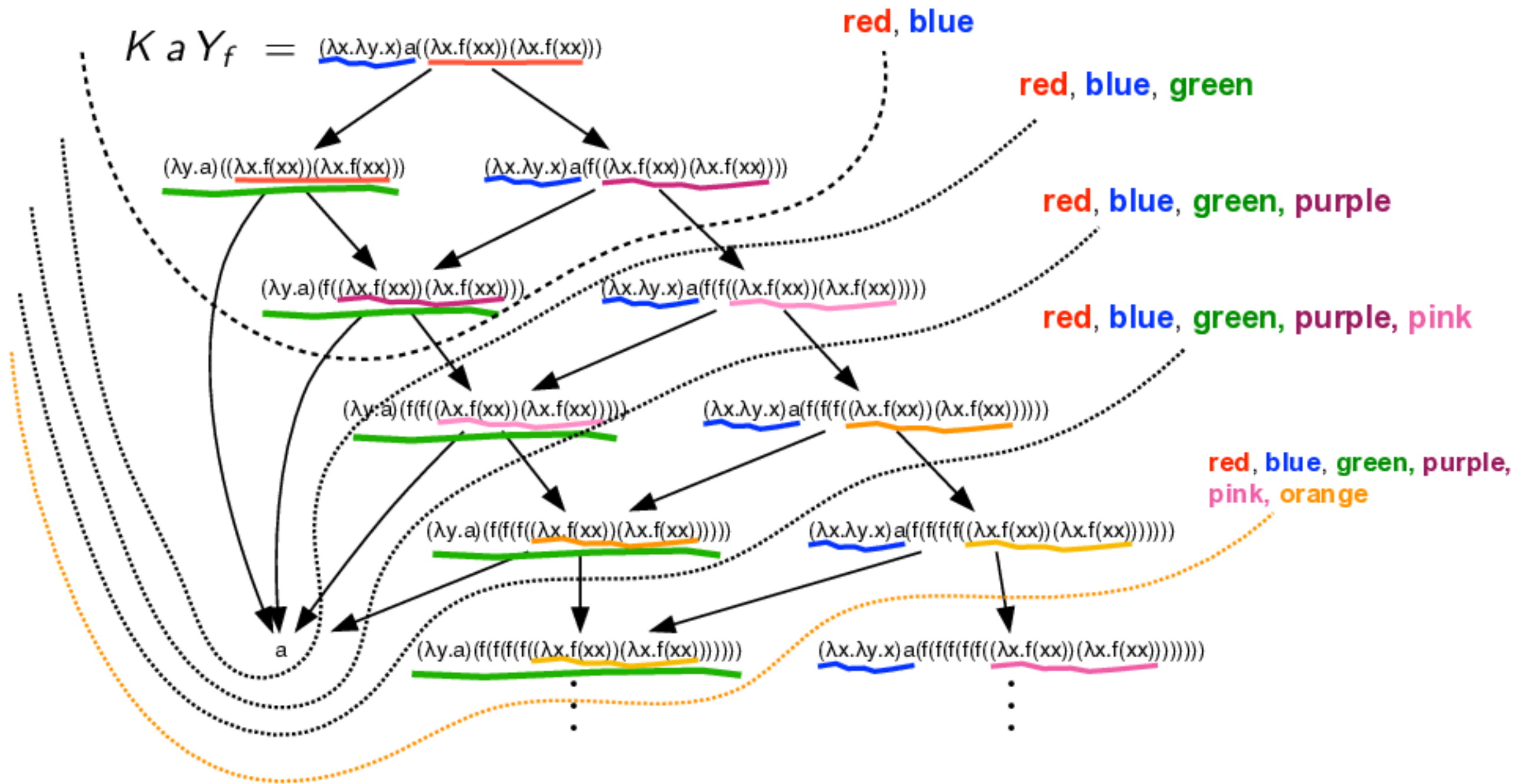
- **Theorem** [Generalized Finite Developments+, 76]
Let $\mathcal{F}$ be a finite set of redex families.
 - (1) there are no infinite reductions relative to $\mathcal{F}$,
 - (2) they all finish on same term N
 - (3) All developments are equivalent by permutations.

Example



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

Example



developments of families.

Finite and infinite reductions (2/3)

- **Corollary** An **infinite reduction** contracts an **infinite set of redex families**.
- **Corollary** Any term generating a finite number of redex families strongly normalizes

finite number of redex families



strong normalization



Proof of the GFD thm

Bound on heights of labels

- **Definition** The height of a label is its nesting of underlines and overlines

$$h(a) = 0$$

$$h(\lceil \alpha \rceil) = h(\lfloor \alpha \rfloor) = 1 + h(\alpha)$$

$$h(\alpha\beta) = \max\{h(\alpha), h(\beta)\}$$

- **Fact** Let $\mathcal{F}$ be a finite set of redex families, then there is an upper bound $H(\mathcal{F})$ on labels of subterms in reductions relative to $\mathcal{F}$.

When initial term is labeled with atomic letters, we have

$$H(\mathcal{F}) = \max \{h(\alpha) \mid \alpha \in \mathcal{F}\}$$

Proof of finite developments

- **Notation** $\tau(M^\alpha) = \alpha$ when M has an empty external label

- **Lemma 1** Let $M \xrightarrow{\star} M'$, then $h(\tau(M)) \leq h(\tau(M'))$

- **Lemma 2** Let $(\dots ((M M_1)^{\beta_1} M_2)^{\beta_2} \dots M_n)^{\beta_n} \xrightarrow{\star} (\lambda x.N)^\alpha$
Then $h(\tau(M)) \leq h(\alpha)$

- **Lemma 3** [Barendregt] Let $M\{x := N\} \xrightarrow{\star} (\lambda y.P)^\alpha$

There are 2 cases:

$$M \xrightarrow{\star} (\lambda y.M')^\alpha \text{ and } M'\{x := N\} \xrightarrow{\star} P$$

$$M \xrightarrow{\star} M' = (\dots ((x^\beta M_1)^{\beta_1} M_2)^{\beta_2} \dots M_n)^{\beta_n} \text{ and } M'\{x := N\} \xrightarrow{\star} (\lambda y.P)^\alpha$$

Proof of finite developments

- **Notation** Let $\mathcal{SN}_{\mathcal{F}}$ be the set of strongly normalizable terms w.r.t. reductions relative to $\mathcal{F}$.

- **Lemma [subst]** Let $\mathcal{F}$ be a finite set of redex families.

$M, N \in \mathcal{SN}_{\mathcal{F}}$ implies $M\{x := N\} \in \mathcal{SN}_{\mathcal{F}}$

Proof [van Daalen] by induction on $\langle H(\mathcal{F}) - h(\tau(N)), \text{depth}(M), \|M\| \rangle$

- **Theorem GFD** Let $\mathcal{F}$ be a finite set of redex families.

Then $M \in \mathcal{SN}_{\mathcal{F}}$ for all M .

Proof by induction on $\|M\|$



Strong normalization

1st-order typed λ -calculus (1/2)

Residuals of redexes keep their types (of names)

Created redexes have lower types



Finite number of redexes families



Strong normalization

$$\frac{(\lambda x. \dots xN \dots)}{s \rightarrow t} \quad \frac{(\lambda y. M)}{s} \xrightarrow{\text{green}} \dots (\lambda y. M)N' \dots$$

creates

$$\frac{(\lambda x. \lambda y. M)NP}{t} \xrightarrow{\text{green}} (\lambda y. M')P$$

creates

$$\frac{(\lambda x. x)}{s \rightarrow s} \quad \frac{(\lambda y. M)}{s} \xrightarrow{\text{green}} (\lambda y. M)N$$

creates

1st-order typed λ -calculus (2/2)

- **Typed λ -calculus** as a specific labeled calculus

$$s, t ::= \mathbb{N}, \mathbb{B} \mid s \rightarrow t$$

Decorate subterms with their types

$$(\lambda f. (f^{\mathbb{N} \rightarrow \mathbb{N}} 3^{\mathbb{N}})^{\mathbb{N}})^{(\mathbb{N} \rightarrow \mathbb{N}) \rightarrow \mathbb{N}} I^{\mathbb{N} \rightarrow \mathbb{N}} \xrightarrow{\quad} (I^{\mathbb{N} \rightarrow \mathbb{N}} 3^{\mathbb{N}})^{\mathbb{N}} \xrightarrow{\quad} 3^{\mathbb{N}}$$

Apply following rules to labeled λ -calculus

$$[s \rightarrow t] = t$$

$$[s \rightarrow t] = s$$

$$s t = s$$

Scott D-infinity model (1/2)

- Another labeled λ -calculus was considered to study Scott D-infinity model [Hyland-Wadsworth, 74]

- D-infinity projection functions on each subterm (n is any integer):

$$M, N, \dots ::= x^n \mid (MN)^n \mid (\lambda x.M)^n$$

- Conversion rule is:

$$((\lambda x.M)^{n+1} N)^p \longrightarrow M\{x := N_{[n]}\}_{[n][p]}$$

$n + 1$ is **degree** of redex

$$U_{[m][n]} = U_{[p]} \text{ where } p = \min\{m, n\}$$

$$x^n \{x := M\} = M_{[n]}$$

Scott D-infinity model (2/2)

- **Proposition** Hyland-Wadsworth calculus is derivable from labeled calculus by simple homomorphism on labels.

Proof Assign an integer to any atomic letter and take:

$$h(\alpha\beta) = \min\{h(\alpha), h(\beta)\}$$

$$h(\lceil\alpha\rceil) = h(\lfloor\alpha\rfloor) = h(\alpha) - 1$$

- Redex degrees are bounded by maximum of labels in initial term.
therefore a finite number of redex families
- **Proposition** Hyland-Wadsworth calculus strongly normalizes.

Conclusion

- **many** proofs of strong normalization for various calculi
- these proofs look often **magic**
- but intuition is

GFD theorem $\equiv$ strong normalization

- more properties on redex families + labeled calculus
 - standardization theorem
 - completeness of inside-out reductions
 - compactness of main theorems about syntax
 - stability of redexes and sequentiality
 - optimal reductions and relation to Girard's GOI

Finite Developments in the λ -calculus



Part 3

jean-jacques.levy@inria.fr

ISR 2021

Madrid

July 6, 2021

<http://jeanjacqueslevy.net/talks/21isr>



Labeled lambda-calculus

Exercise 1 Show that residuals of redexes keep same names by case inspection on occurrences of redexes.

Exercise 2 Show that $M \longrightarrow N$ implies $M^\alpha \longrightarrow N^\alpha$

Exercise 3 Show the parallel moves lemma (with Martin-Löf way)

If $M \xrightarrow{\mathcal{F}} N$ and $M \xrightarrow{\mathcal{G}} P$, then $N \xrightarrow{\mathcal{G}/\mathcal{F}} Q$ and $P \xrightarrow{\mathcal{F}/\mathcal{G}} Q$ for some Q .

Exercise 4 Label Y_f , draw its reduction graph and show redexes families when $Y_f = (\lambda x.f(xx))(\lambda x.f(xx))$

Exercise 5 Same with KaY_f

Inside-out reductions

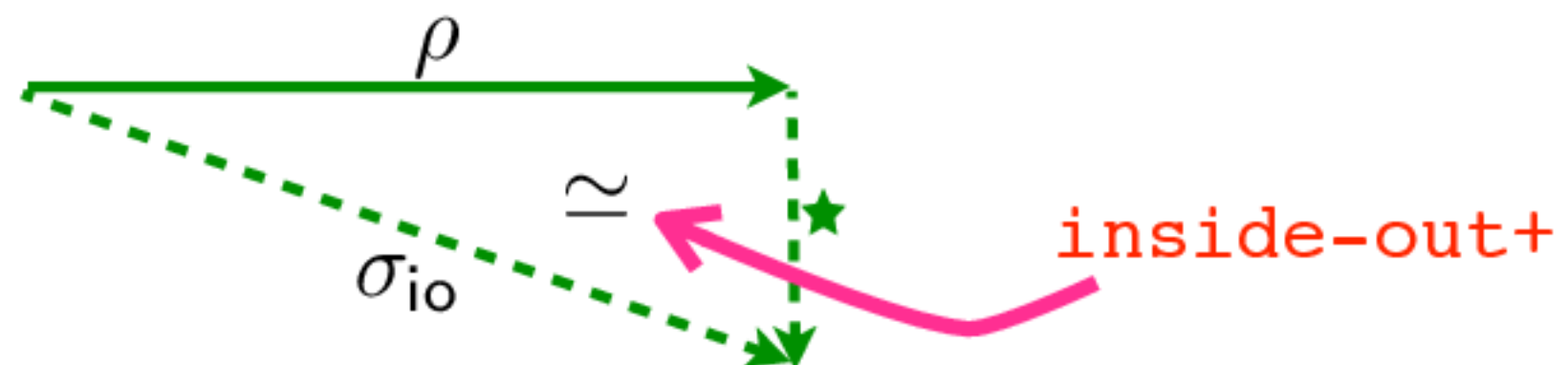
- **Definition:** The following reduction is **inside-out**

$$\rho : M = M_0 \xrightarrow{R_1} M_1 \xrightarrow{R_2} M_2 \cdots \xrightarrow{R_n} M_n = N$$

iff for all i and j , $i < j$, then R_j is not residual along ρ of some R'_j inside R_i in M_{i-1} .

- **Theorem** [Inside-out completeness, 74]

Let $M \xrightarrow{\star} N$. Then $M \xrightarrow[\text{io}]{\star} P$ and $N \xrightarrow{\star} P$ for some P .

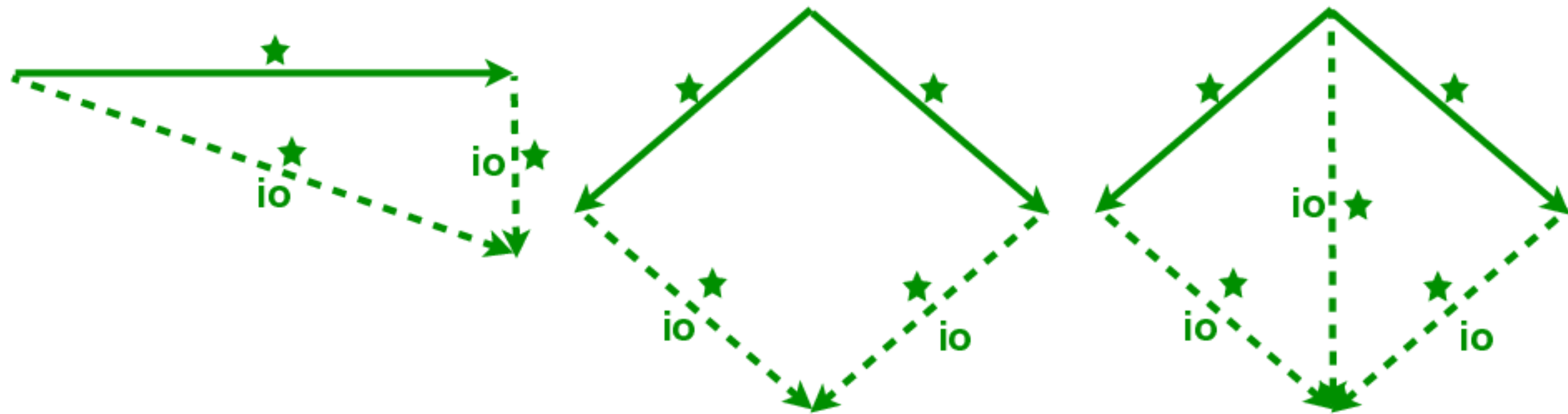


Exercises

Exercise 6 Prove inside-out completeness

Hint: use Finite Development theorem.

Exercise 7 Prove the following diagrams



Permutation equivalence

Proof [uniqueness of labeled standard]

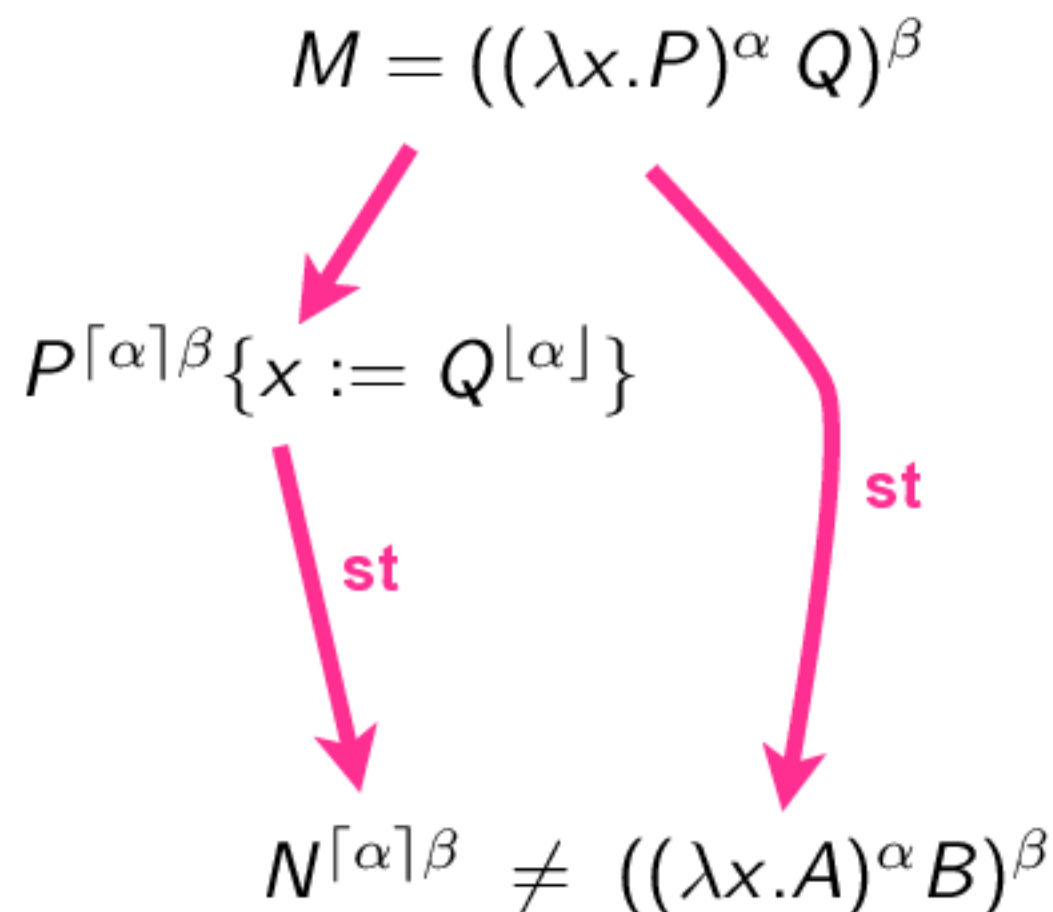
Let ρ and σ be 2 distinct coinitial pure labeled standard reductions.

Take first step when they diverge. Call M that term.

We make structural induction on M . Say ρ is more to the left.

If first step of ρ contracts an internal redex, we use induction.

If first step of ρ contracts an external redex, then:



Permutation equivalence

- **Corollary** [labeled prefix ordering]

Let $\rho : M \xrightarrow{\star} N$ and $\sigma : M \xrightarrow{\star} P$ be coinitial pure labeled reductions.
Then $\rho \sqsubseteq \sigma$ iff $N \xrightarrow{\star} P$.

- **Exercise 8** Show the following properties

(i) $\rho \sqsubseteq \rho$

(ii) $\rho \sqsubseteq \sigma \sqsubseteq \rho$ implies $\rho \simeq \sigma$

(iii) $\rho \sqsubseteq \sigma \sqsubseteq \tau$ implies $\rho \simeq \tau$

(iv) $\rho \sqsubseteq \sigma$ implies $\rho/\tau \sqsubseteq \sigma/\tau$

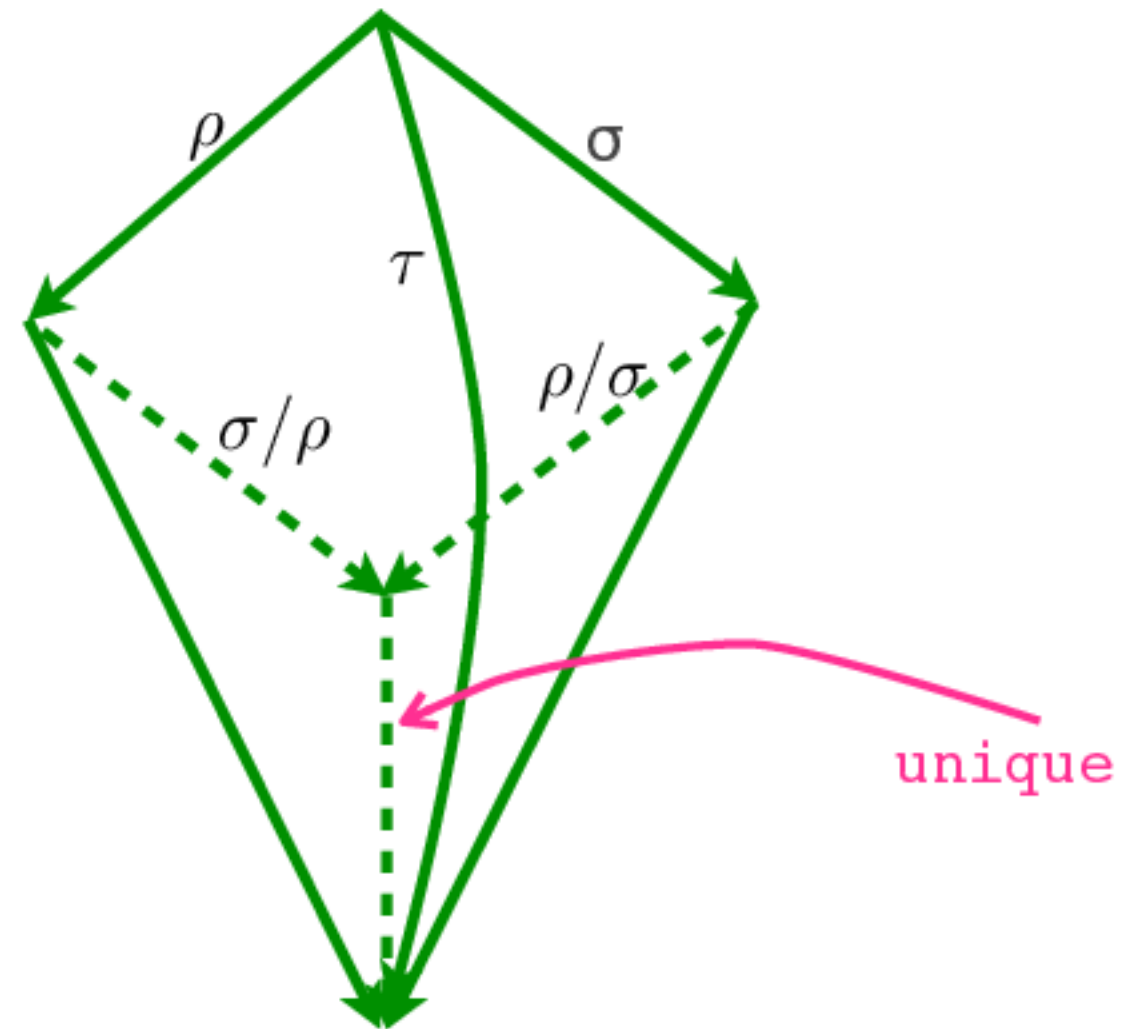
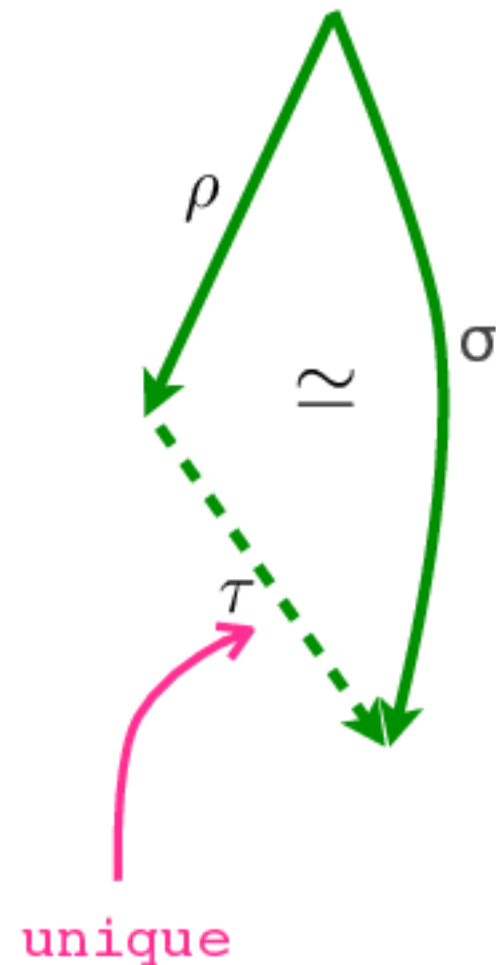
(v) $\rho \sqsubseteq \sigma$ iff $\exists \tau, \rho\tau \simeq \sigma$

(vi) $\rho \sqsubseteq \rho \sqcup \sigma, \sigma \sqsubseteq \rho \sqcup \sigma$

(vii) $\rho \sqsubseteq \tau, \sigma \sqsubseteq \tau$ implies $\rho \sqcup \sigma \sqsubseteq \tau$

Permutation equivalence

- **Exercise 9** Show the following diagrams



Permutation equivalence

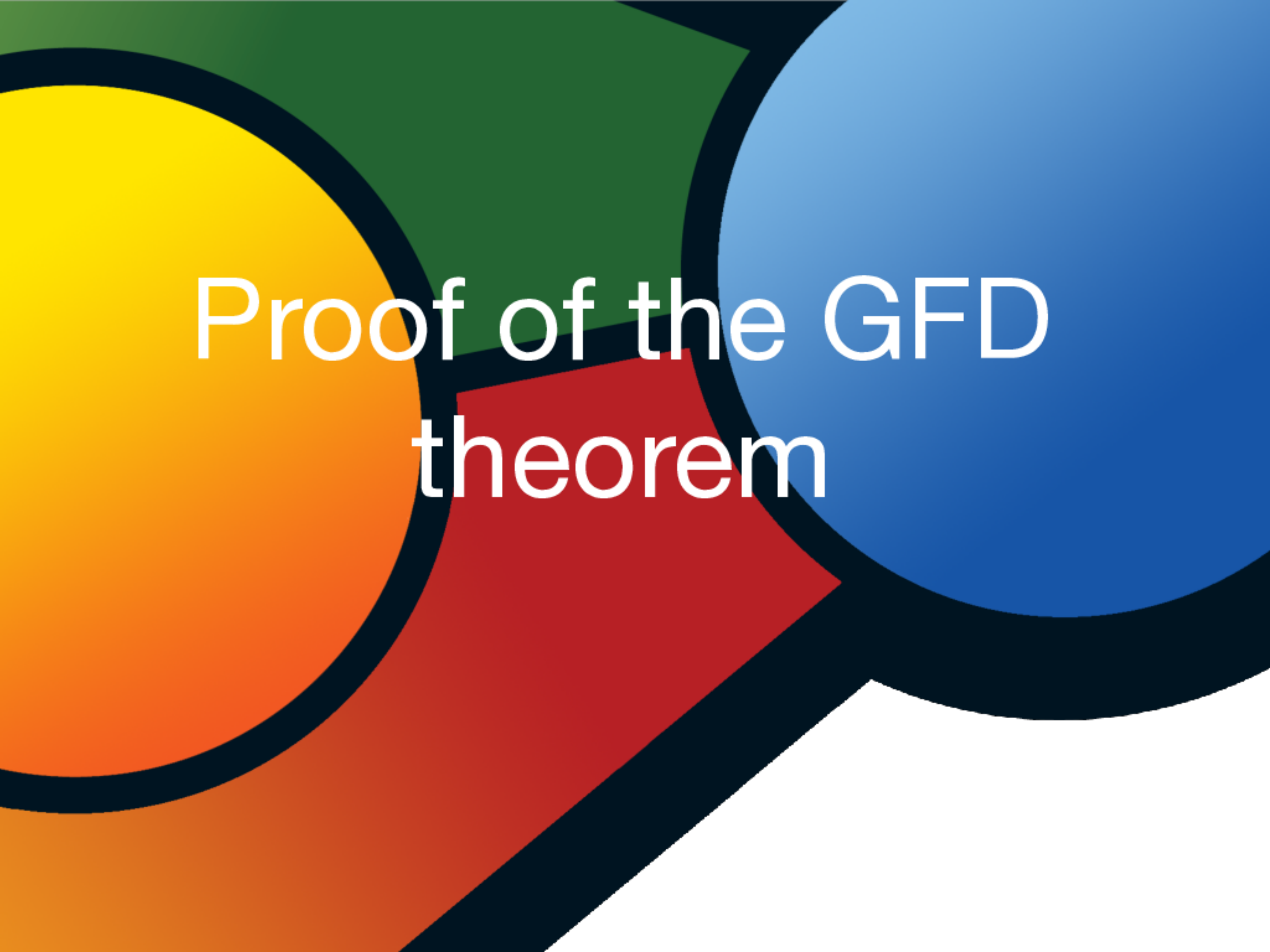
- **Corollary** [lattice of labeled reductions]

Labeled reduction graphs are upwards semi lattices for any pure labeling.

- **Corollary** [push-out category]

Prefix ordering on reductions is a push-out.

- **Exercise 10** Try on $(\lambda x.x)((\lambda y.(\lambda x.x)a)b)$ or $(\lambda x.xx)(\lambda x.xx)$
- **Exercise 11** Show that prefix ordering on reductions is not a pull-back.



Proof of the GFD theorem

Bound on heights of labels

- **Definition** The height of a label is its nesting of underlines and overlines

$$h(a) = 0$$

$$h(\lceil \alpha \rceil) = h(\lfloor \alpha \rfloor) = 1 + h(\alpha)$$

$$h(\alpha\beta) = \max\{h(\alpha), h(\beta)\}$$

- **Fact** Let $\mathcal{F}$ be a finite set of redex families, then there is an upper bound $H(\mathcal{F})$ on labels of subterms in reductions relative to $\mathcal{F}$.

When initial term is labeled with atomic letters, we have

$$H(\mathcal{F}) = \max \{h(\alpha) \mid \alpha \in \mathcal{F}\}$$

Proof of finite developments

- **Notation** $\tau(M^\alpha) = \alpha$ when M has an empty external label

- **Lemma 1** Let $M \xrightarrow{\star} M'$, then $h(\tau(M)) \leq h(\tau(M'))$

- **Lemma 2** Let $(\dots ((M M_1)^{\beta_1} M_2)^{\beta_2} \dots M_n)^{\beta_n} \xrightarrow{\star} (\lambda x.N)^\alpha$
Then $h(\tau(M)) \leq h(\alpha)$

- **Lemma 3** [Barendregt] Let $M\{x := N\} \xrightarrow{\star} (\lambda y.P)^\alpha$

There are 2 cases:

$$M \xrightarrow{\star} (\lambda y.M')^\alpha \text{ and } M'\{x := N\} \xrightarrow{\star} P$$

$$M \xrightarrow{\star} M' = (\dots ((x^\beta M_1)^{\beta_1} M_2)^{\beta_2} \dots M_n)^{\beta_n} \text{ and } M'\{x := N\} \xrightarrow{\star} (\lambda y.P)^\alpha$$

Proof of finite developments

- **Lemma 1** Let $M \xrightarrow{\star} N$, then $h(\tau(M)) \leq h(\tau(N))$

Proof by induction on length of reduction. Let $M \xrightarrow{R} N$, $R = ((\lambda x.A)^\alpha B)^\beta$

If R is internal in M , then $\tau(M) = \tau(N)$.

If $M = R = ((\lambda x.A)^\alpha B)^\beta \xrightarrow{\star} A\{x := B^{\lfloor \alpha \rfloor}\}^{\lceil \alpha \rceil \beta} = N$,
then $h(\tau(M)) = h(\beta) \leq h(\gamma\beta) = h(\tau(N))$ for some γ .

- **Lemma 2** Let $(\dots ((M M_1)^{\beta_1} M_2)^{\beta_2} \dots M_n)^{\beta_n} \xrightarrow{\star} (\lambda x.N)^\alpha$

Then $h(\tau(M)) \leq h(\alpha)$

Proof by induction on n .

When $n = 0$, obvious by lemma 1.

Otherwise $(\dots ((M M_1)^{\beta_1} M_2)^{\beta_2} \dots M_{n-1})^{\beta_{n-1}} \xrightarrow{\star} (\lambda y.P)^\gamma$

and $((\lambda y.P)^\gamma Q)^{\beta_n} \xrightarrow{\star} P\{y := Q^{\lfloor \gamma \rfloor}\}^{\lceil \gamma \rceil \beta_n} \xrightarrow{\star} (\lambda x.N)^\alpha$

So $h(\tau(M)) \leq h(\gamma) < h(\delta \lceil \gamma \rceil \beta_n) \leq h(\alpha)$ by induction and lemma 1.

Proof of finite developments

- **Lemma 3** [Barendregt] Let $M\{x := N\} \xrightarrow{\star} (\lambda y.P)^\alpha$

There are 2 cases:

$$M \xrightarrow{\star} (\lambda y.M')^\alpha \text{ and } M'\{x := N\} \xrightarrow{\star} P$$

$$M \xrightarrow{\star} M' = (\dots((x^\beta M_1)^{\beta_1} M_2)^{\beta_2} \dots M_n)^{\beta_n} \text{ and } M'\{x := N\} \xrightarrow{\star} (\lambda y.P)^\alpha$$

Proof Let $M^* = M\{x := N\}$. There are 3 cases on weak head reduction of M :
 it reaches an abstraction or a head variable which has to be x .
 More precisely, we consider the standard reduction from M^* to $(\lambda y.P)^\alpha$.

Case 1: $M = (\lambda y.M')^\alpha$ and we are done since $M^* = (\lambda y.M'^*)^\alpha$.

Case 2: $M = ((\dots((y^\beta M_1)^{\beta_1} M_2)^{\beta_2}) \dots M_n)^{\beta_n}$. Then $y = x$ and $M' = M$.

Case 3: $M = (\dots((((\lambda z.A)^\beta B)^\gamma C_1)^{\beta_1} C_2)^{\beta_2} \dots C_n)^{\beta_n}$

$$\text{Let } M_1 = (\dots((A\{z := B^{[\beta]}\}^{\lceil\beta\rceil} C_1)^{\beta_1} C_2)^{\beta_2} \dots C_n)^{\beta_n}$$

Then $M^* = (\dots((((\lambda z.A^*)^\beta B^*)^{\beta_1} C_1^*)^{\beta_1} C_2^*)^{\beta_2} \dots C_n^*)^{\beta_n} \xrightarrow{\star} M_1^*$ is the first step of the standard reduction from M^* to $(\lambda y.P)^\alpha$. By induction on its length, we are done.

Proof of finite developments

- **Notation** Let $\mathcal{SN}_{\mathcal{F}}$ be the set of strongly normalizable terms w.r.t. reductions relative to $\mathcal{F}$.

- **Lemma** [subst] Let $\mathcal{F}$ be a finite set of redex families.

$M, N \in \mathcal{SN}_{\mathcal{F}}$ implies $M\{x := N\} \in \mathcal{SN}_{\mathcal{F}}$

Proof [van Daalen] by induction on $\langle H(\mathcal{F}) - h(\tau(N)), \text{depth}(M), \|M\| \rangle$

- **Theorem** GFD Let $\mathcal{F}$ be a finite set of redex families.

Then $M \in \mathcal{SN}_{\mathcal{F}}$ for all M .

Proof by easy induction on $\|M\|$

Proof of finite developments

- **Lemma** [subst] Let $\mathcal{F}$ be a finite set of redex families.

$M, N \in \mathcal{SN}_{\mathcal{F}}$ implies $M\{x := N\} \in \mathcal{SN}_{\mathcal{F}}$

Proof [van Daalen] by induction on $\langle H(\mathcal{F}) - h(\tau(N)), \text{depth}(M), \|M\| \rangle$

Cases $M = x$, $M = y$, $M = \lambda y.M_1$ are obvious or easy by induction on $\|M\|$.

Write M^* for $M\{x := N\}$ and consider case $M = (M_1 M_2)^\alpha$.

If all reductions are internal to M_1^* and M_2^* , then easy induction on $\|M\|$.

Otherwise, let $M_1^* \xrightarrow{\star} (\lambda y.P)^\beta$ and $M_2^* \xrightarrow{\star} Q$ and $((\lambda y.P)^\beta Q)^\alpha \rightarrow P\{y := Q^{\lfloor \beta \rfloor}\}^{\lceil \beta \rceil \alpha}$

Then M_1^* and M_2^* are in $\mathcal{SN}_{\mathcal{F}}$ by induction on $\|M\|$,

and $M_1^* \xrightarrow{\star} (\lambda y.P)^\beta$ and $M_2^* \xrightarrow{\star} Q$. So P and Q are in $\mathcal{SN}_{\mathcal{F}}$.

How is $P\{y := Q^{\lfloor \beta \rfloor}\}^{\lceil \beta \rceil \alpha}$??

By lemma 3, we have 2 cases:

Proof of finite developments

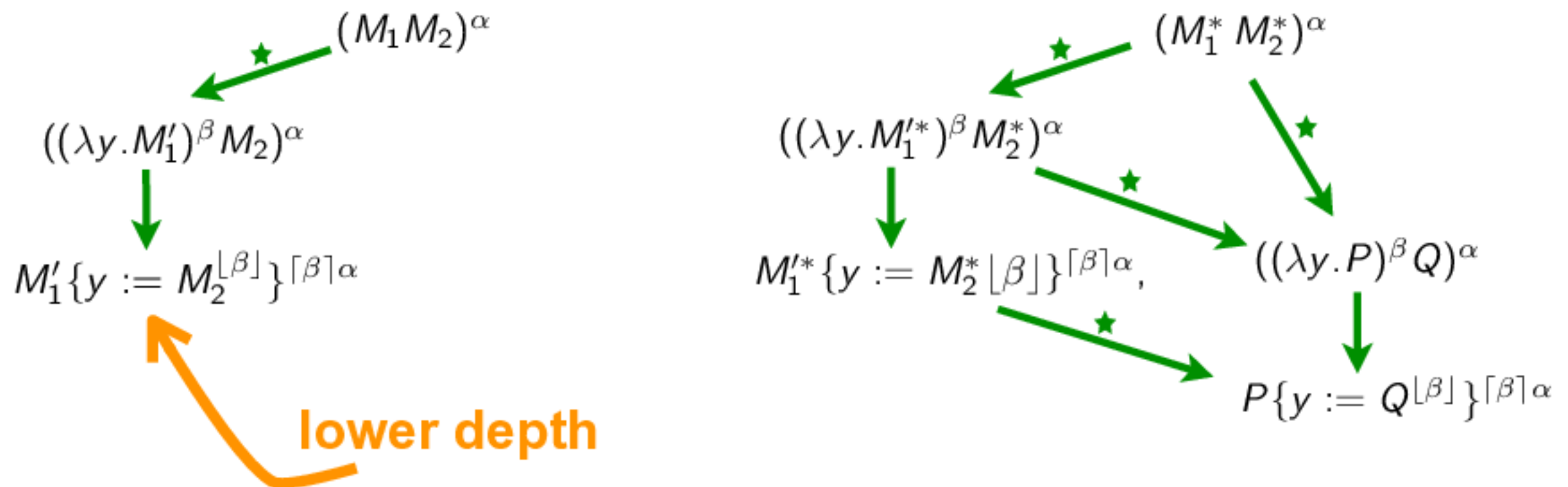
Case 1:

Then $M_1 \xrightarrow{\star} (\lambda y. M'_1)^\beta$ and $M_1'^* \xrightarrow{\star} P$.

Therefore $M_1'^* \{y := M_2^{*\lfloor\beta\rfloor}\}^{\lceil\beta\rceil\alpha} \xrightarrow{\star} P \{y := Q^{\lfloor\beta\rfloor}\}^{\lceil\beta\rceil\alpha}$.

But as $M = (M_1 M_2)^\alpha \xrightarrow{\star} ((\lambda y. M'_1)^\beta M_2)^\alpha \rightarrow M' = M'_1 \{y := M_2^{\lfloor\beta\rfloor}\}^{\lceil\beta\rceil\alpha}$,
we have $\text{depth}(M') < \text{depth}(M)$.

Thus by induction $M'^* = M_1'^* \{y := M_2^{*\lfloor\beta\rfloor}\}^{\lceil\beta\rceil\alpha} \in \mathcal{SN}_{\mathcal{F}}$
and $P \{y := Q^{\lfloor\beta\rfloor}\}^{\lceil\beta\rceil\alpha} \in \mathcal{SN}_{\mathcal{F}}$.



Proof of finite developments

Case 2:

$$M_1 \xrightarrow{\star} M'_1 = (\dots ((x^\gamma A_1)^{\gamma_1} A_2)^{\gamma_2}) \dots A_n)^{\gamma_n} \text{ and}$$

$$M'^*_1 = (\dots ((N^\gamma A_1^*)^{\gamma_1} A_2^*)^{\gamma_2}) \dots A_n^*)^{\gamma_n} \xrightarrow{\star} (\lambda y. P)^\beta$$

Therefore $h(\tau(N)) \leq h(\tau(N^\gamma)) \leq h(\beta)$ by lemma 2.

$$\text{So } M^* = (M_1^* M_2^*)^\alpha \xrightarrow{\star} ((\lambda y. P)^\beta Q)^\alpha \xrightarrow{\star} P\{y := Q^{\lfloor \beta \rfloor}\}^{\lceil \beta \rceil \alpha}$$

$$\text{and } h(\tau(N)) \leq h(\beta) < h(\lfloor \beta \rfloor) \leq h(\tau(Q^{\lfloor \beta \rfloor})).$$

We get by induction $P\{y := Q^{\lfloor \beta \rfloor}\}^{\lceil \beta \rceil \alpha} \in \mathcal{SN}_{\mathcal{F}}$.

