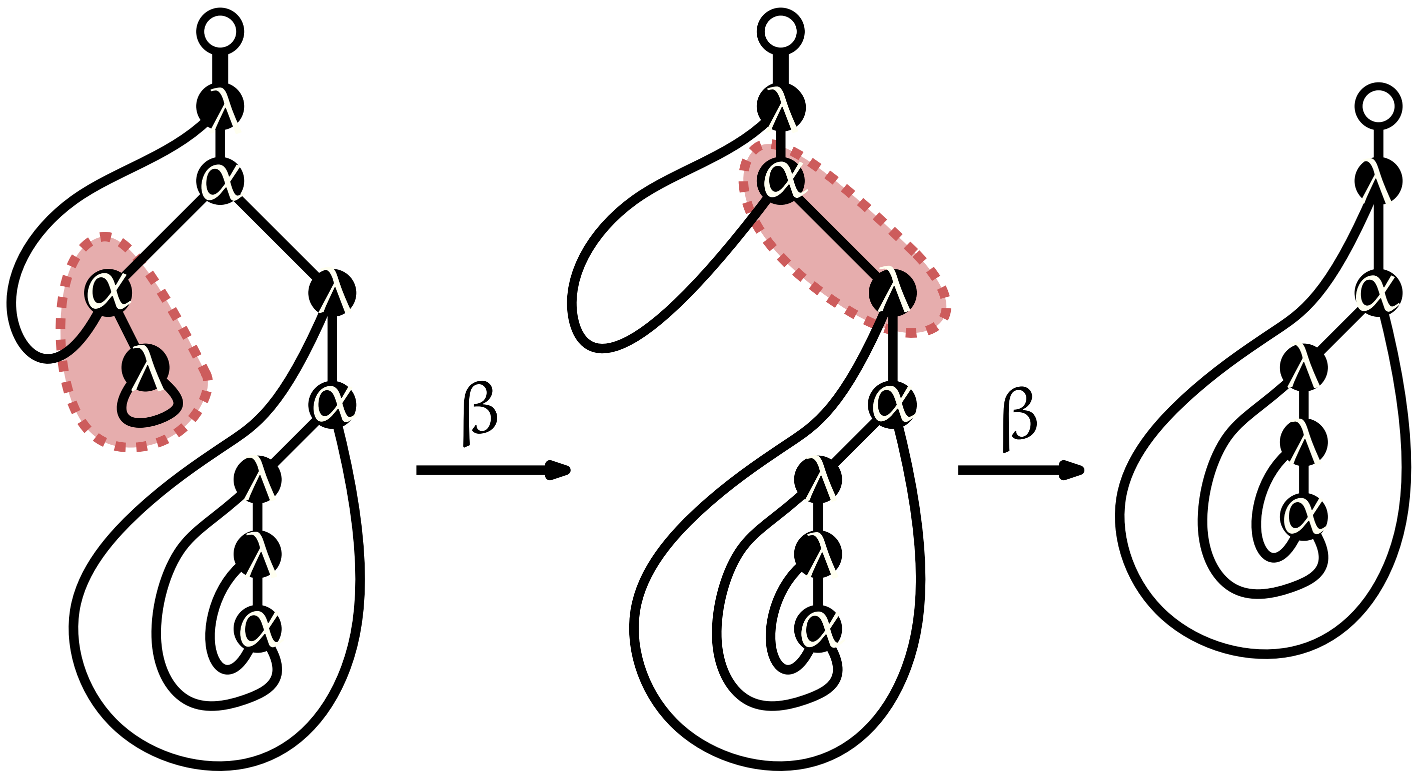
Normalisation of random linear λ-terms



Alexandros Singh

Based on joint work with Olivier Bodini, Bernhard Gittenberger
Michael Wallner, and Noam Zeilberger.

AofA 2023 - Taiwan
Friday, June 30 2023

What is the λ-calculus?

A calculus of functions taking a single argument:

$$f, t := x \mid \lambda x.t \mid (f\ t)$$

What is the λ-calculus?

A calculus of functions taking a single argument:

$$f, t := x \mid \lambda x.t \mid (f\ t)$$

variables

What is the λ-calculus?

A calculus of functions taking a single argument:

$$f, t := x \mid \lambda x.t \mid (f\ t)$$

variables

abstractions
represent functions "$x \mapsto t$"

What is the λ-calculus?

A calculus of functions taking a single argument:

$$f, t := x \mid \lambda x.t \mid (f\ t)$$

variables

abstractions
represent functions "$x \mapsto t$"

applications
represent "$f(t)$"

What is the λ-calculus?

A calculus of functions taking a single argument:

$$f, t := x \mid \lambda x.t \mid (f\ t)$$

variables

abstractions
represent functions "$x \mapsto t$"

applications
represent "$f(t)$"

- Introduced by Church around 1928, developed together with Kleene, Rosser.

What is the λ-calculus?

A calculus of functions taking a single argument:

$$f, t := x \mid \lambda x.t \mid (f\ t)$$

variables

abstractions represent functions "$x \mapsto t$"

applications represent "$f(t)$"

- Introduced by Church around 1928, developed together with Kleene, Rosser.

- It can encode: arithmetic, data structures, programming ...

Church-Turing thesis: "effectively computable" = definable in λ-calculus (or Turing machines, or recursive functions).

What is the λ-calculus?

A calculus of functions taking a single argument:

$$f, t := x \mid \lambda x.t \mid (f\ t)$$

variables

abstractions represent functions "$x \mapsto t$"

applications represent "$f(t)$"

- Introduced by Church around 1928, developed together with Kleene, Rosser.

- It can encode: arithmetic, data structures, programming ...

Church-Turing thesis: "effectively computable" = definable in λ-calculus (or Turing machines, or recursive functions).

- In its typed form: functional programming, proof theory,...

More on the λ-calculus

Examples of terms:

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

$$g(x, y) = y \rightsquigarrow \lambda x.\lambda y.y$$

(Currying: $X \times Y \to Z \leftrightsquigarrow X \to Y \to Z$)

$$f \circ g \rightsquigarrow (\lambda x.x)(\lambda x.\lambda y.y)$$

More on the λ-calculus

Examples of terms:

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

$g(x, y) = y \rightsquigarrow \lambda x.\lambda y.y$ (Currying: $X \times Y \to Z \leftrightsquigarrow X \to Y \to Z$)

$f \circ g \rightsquigarrow (\lambda x.x)(\lambda x.\lambda y.y)$

Some terminology:

$(\lambda x.(x\ y))$ open term (has *free* variables)

$(\lambda x.(x\ x))(\lambda z.z)$ closed term (no free variables)

$((\lambda x.\lambda y.(y\ x))\ a)$ linear term (bound vars. used once)

More on the λ-calculus

Examples of terms:

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

$$g(x, y) = y \rightsquigarrow \lambda x.\lambda y.y \qquad \text{(Currying: } X \times Y \to Z \leftrightsquigarrow X \to Y \to Z\text{)}$$

$$f \circ g \rightsquigarrow (\lambda x.x)(\lambda x.\lambda y.y)$$

Some terminology:

$(\lambda x.(x\ y))$ — open term (has *free* variables)

$(\lambda x.(x\ x))(\lambda z.z)$ — closed term (no free variables)

$((\lambda x.\lambda y.(y\ x))\ a)$ — linear term (bound vars. used once)

Terms are considered up to *careful* renaming of variables:

$$(\lambda x.\lambda y.(x\ y\ x)) \stackrel{\alpha}{=} (\lambda z.\lambda y.(z\ y\ z)) \stackrel{\alpha}{\neq} (\lambda x.\lambda y.(z\ y\ x))$$

Dynamics of the λ-calculus: β-reductions

$$((\lambda x.t_1)\ t_2) \xrightarrow{\beta} t_1[x := t_2]$$

Computing with the λ-calculus

Dynamics of the λ-calculus: β-reductions

redex $\quad ((\lambda x.t_1)\ t_2) \xrightarrow{\beta} t_1[x := t_2]$

What it means:
Given a function $f = x \mapsto t_1$ and an argument t_2,
to compute $f(t_2)$, replace x with t_2 inside t_1.

Computing with the λ-calculus

Dynamics of the λ-calculus: β-reductions

redex $\quad ((\lambda x.t_1)\ t_2) \xrightarrow{\beta} t_1[x := t_2]$

What it means:
Given a function $f = x \mapsto t_1$ and an argument t_2,
to compute $f(t_2)$, replace x with t_2 inside t_1.

Examples of reductions:

$$f \circ g = g \rightsquigarrow (\lambda x.x)(\lambda x.\lambda y.y) \xrightarrow{\beta} x[x := (\lambda x.\lambda y.y)] = (\lambda x.\lambda y.y)$$

A term with no redices is called a normal form

Computing with the λ-calculus

Dynamics of the λ-calculus: β-reductions

redex $\quad ((\lambda x.t_1)\ t_2) \xrightarrow{\beta} t_1[x := t_2]$

What it means:
Given a function $f = x \mapsto t_1$ and an argument t_2,
to compute $f(t_2)$, replace x with t_2 inside t_1.

Examples of reductions:

$$f \circ g = g \rightsquigarrow (\lambda x.x)(\lambda x.\lambda y.y) \xrightarrow{\beta} x[x := (\lambda x.\lambda y.y)] = (\lambda x.\lambda y.y)$$

$$((\lambda x.((\lambda y.(y\ x))\ z))\ (a\ b))$$

A term with no redices is called a normal form

Computing with the λ-calculus

Dynamics of the λ-calculus: β-reductions

redex $\quad ((\lambda x.t_1)\ t_2) \xrightarrow{\beta} t_1[x := t_2]$

What it means:
Given a function $f = x \mapsto t_1$ and an argument t_2,
to compute $f(t_2)$, replace x with t_2 inside t_1.

Examples of reductions:

$$f \circ g = g \rightsquigarrow (\lambda x.x)(\lambda x.\lambda y.y) \xrightarrow{\beta} x[x := (\lambda x.\lambda y.y)] = (\lambda x.\lambda y.y)$$

$$((\lambda x.((\lambda y.(y\ x))\ z))\ (a\ b))$$

A term with no redices is called a normal form

Computing with the λ-calculus

Dynamics of the λ-calculus: β-reductions

redex → $((\lambda x. t_1)\ t_2) \xrightarrow{\beta} t_1[x := t_2]$

What it means:
Given a function $f = x \mapsto t_1$ and an argument t_2,
to compute $f(t_2)$, replace x with t_2 inside t_1.

Examples of reductions:

$$f \circ g = g \rightsquigarrow (\lambda x. x)(\lambda x. \lambda y. y) \xrightarrow{\beta} x[x := (\lambda x. \lambda y. y)] = (\lambda x. \lambda y. y)$$

$$((\lambda x. ((\lambda y. (y\ x))\ z))\ (a\ b))$$

A term with no redices is called a normal form

Computing with the λ-calculus

Dynamics of the λ-calculus: β-reductions

redex → $((\lambda x.t_1)\ t_2) \xrightarrow{\beta} t_1[x := t_2]$

What it means:
Given a function $f = x \mapsto t_1$ and an argument t_2,
to compute $f(t_2)$, replace x with t_2 inside t_1.

Examples of reductions:

$$f \circ g = g \rightsquigarrow (\lambda x.x)(\lambda x.\lambda y.y) \xrightarrow{\beta} x[x := (\lambda x.\lambda y.y)] = (\lambda x.\lambda y.y)$$

$$((\lambda x.((\lambda y.(y\ x))\ z))\ (a\ b)) \to ((\lambda y.(y\ (a\ b))\ z) \to (z\ (a\ b))$$

$$((\lambda x.((\lambda y.(y\ x))\ z))\ (a\ b)) \to ((\lambda x.(z\ x))\ (a\ b)) \to (z\ (a\ b))$$

A term with no redices is called a normal form

Computing with the λ-calculus

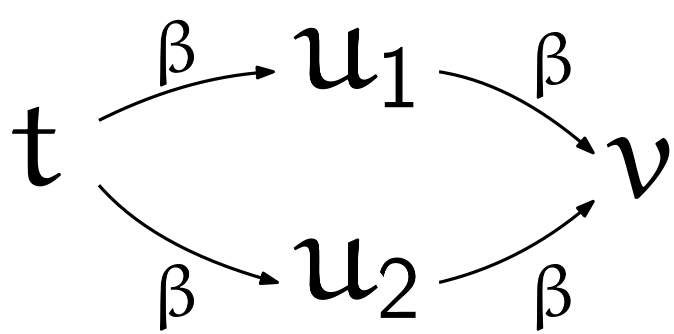
Dynamics of the λ-calculus: β-reductions

redex

$$((\lambda x.t_1)\ t_2) \xrightarrow{\beta} t_1[x := t_2]$$

What it means:
Given a function $f = x \mapsto t_1$ and an argument t_2,
to compute $f(t_2)$, replace x with t_2 inside t_1.

For linear terms: β-reduction is strongly normalising,
has strong diamond property.



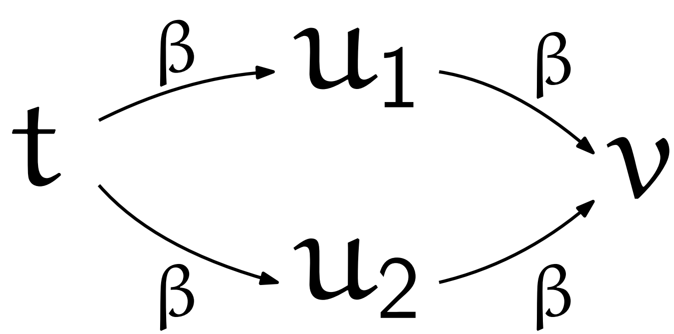
Computing with the λ-calculus

Dynamics of the λ-calculus: β-reductions

redex $\quad ((\lambda x.t_1)\ t_2) \xrightarrow{\beta} t_1[x := t_2]$

What it means:
Given a function $f = x \mapsto t_1$ and an argument t_2,
to compute $f(t_2)$, replace x with t_2 inside t_1.

For linear terms: β-reduction is strongly normalising,
has strong diamond property.



β-normalisation terminates in deterministic number of steps!

Normalisation of random closed linear terms

Q: How many steps does it take for a random closed linear term to reach normal form, on average?

Normalisation of random closed linear terms

well defined! (strong normalisation + diamond)

Q: How many steps does it take for a random closed linear term to reach normal form, on average?

uniform distribution on terms of size n

Size of a term t = # of subterms of t.
Equivalently, size defined via recursion:

$|x| = 1$ $\quad$ $|\lambda x.t| = 1 + |t|$ $\quad$ $|(f\ g)| = 1 + |f| + |g|$

$n = 2$

$\lambda x.x$

$n = 5$

$(\lambda x.x)(\lambda y.y)$ $\quad$ $\lambda x.((\lambda y.y)\ x)$

$\lambda x.\lambda y.(x\ y)$ $\quad$ $\lambda x.(x\ (\lambda y.y))$

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

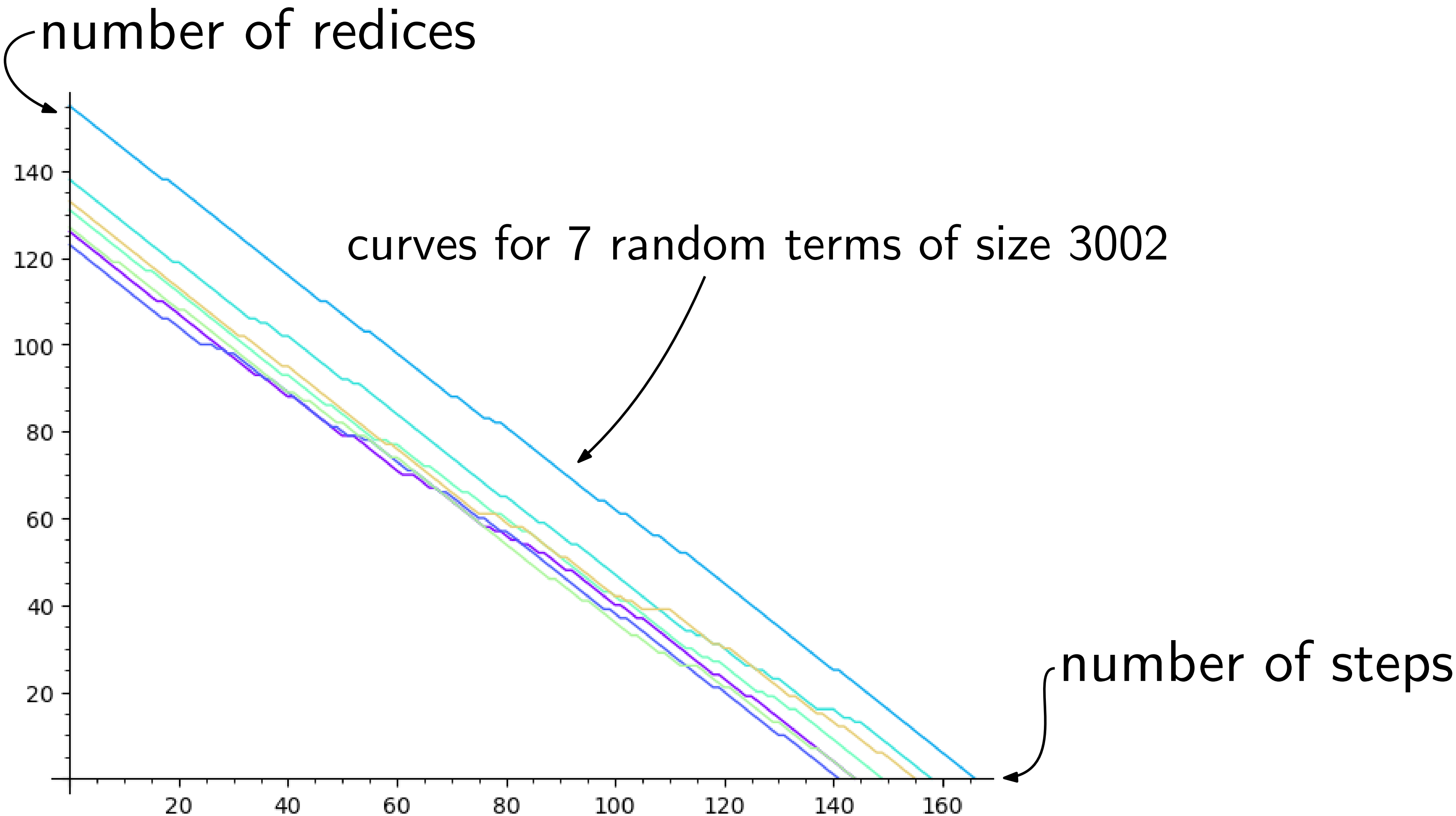
$n = 8$

60 terms

OEIS: A062980 (spoilers!)

Normalisation of random closed linear terms

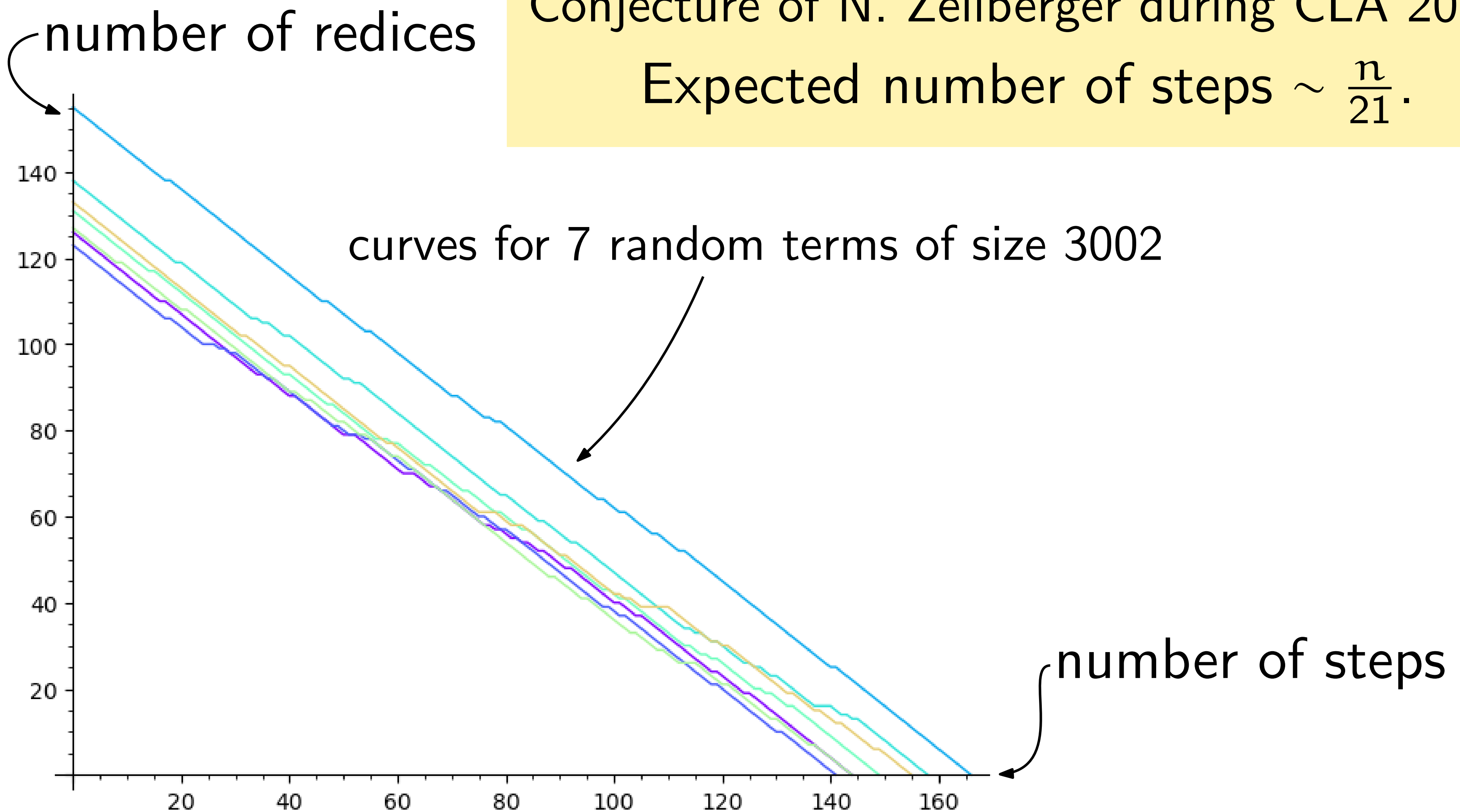
Q: How many steps does it take for a random closed linear term to reach normal form, on average?



Normalisation of random closed linear terms

Q: How many steps does it take for a random closed linear term to reach normal form, on average?

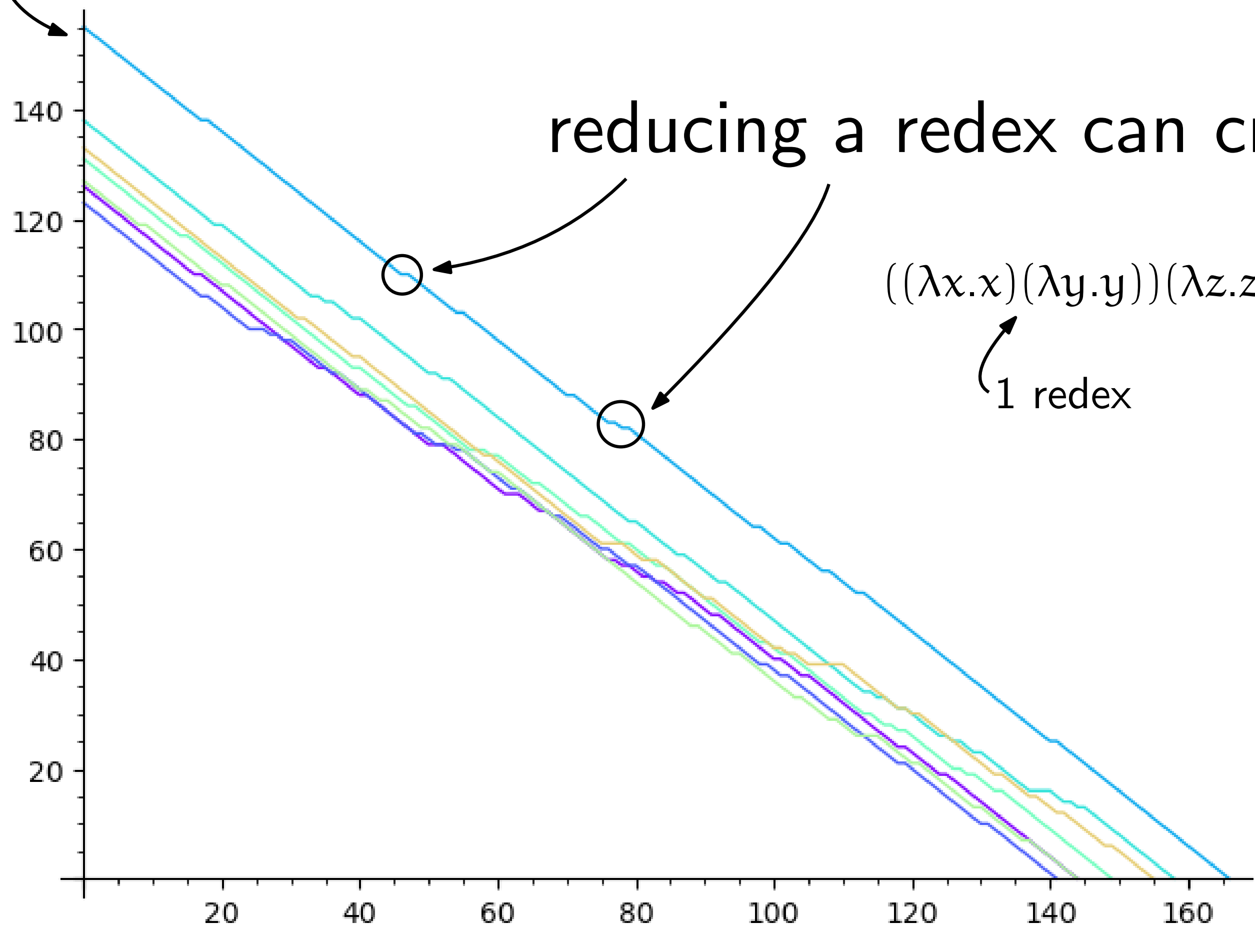
Conjecture of N. Zeilberger during CLA 2020:
Expected number of steps $\sim \frac{n}{21}$.



Normalisation of random closed linear terms

Q: How many steps does it take for a random closed linear term to reach normal form, on average?

number of redices



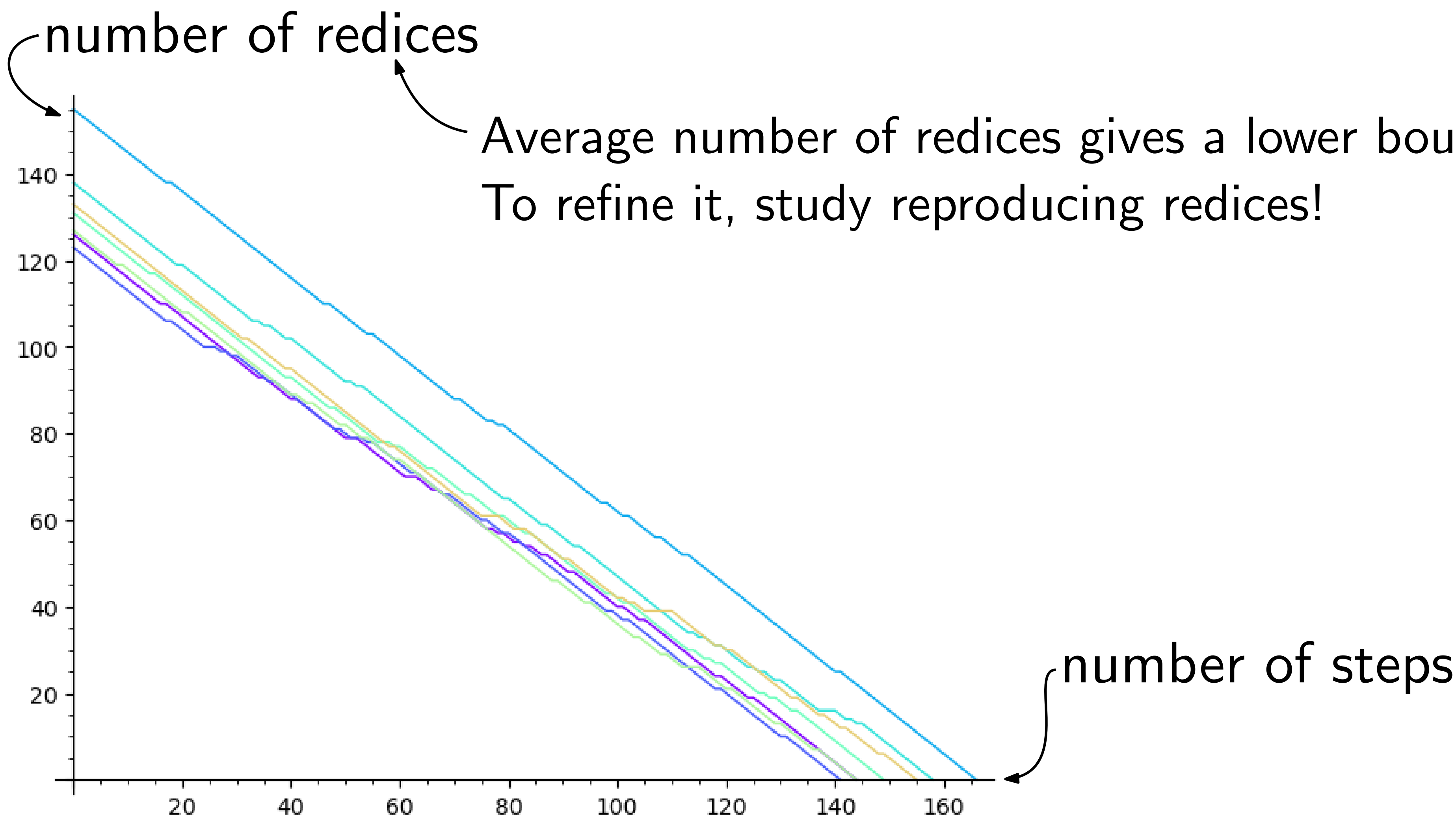
$$((\lambda x.x)(\lambda y.y))(\lambda z.z) \xrightarrow{\beta} (\lambda y.y)(\lambda z.z) \xrightarrow{\beta} \lambda z.z$$

1 redex   1 redex

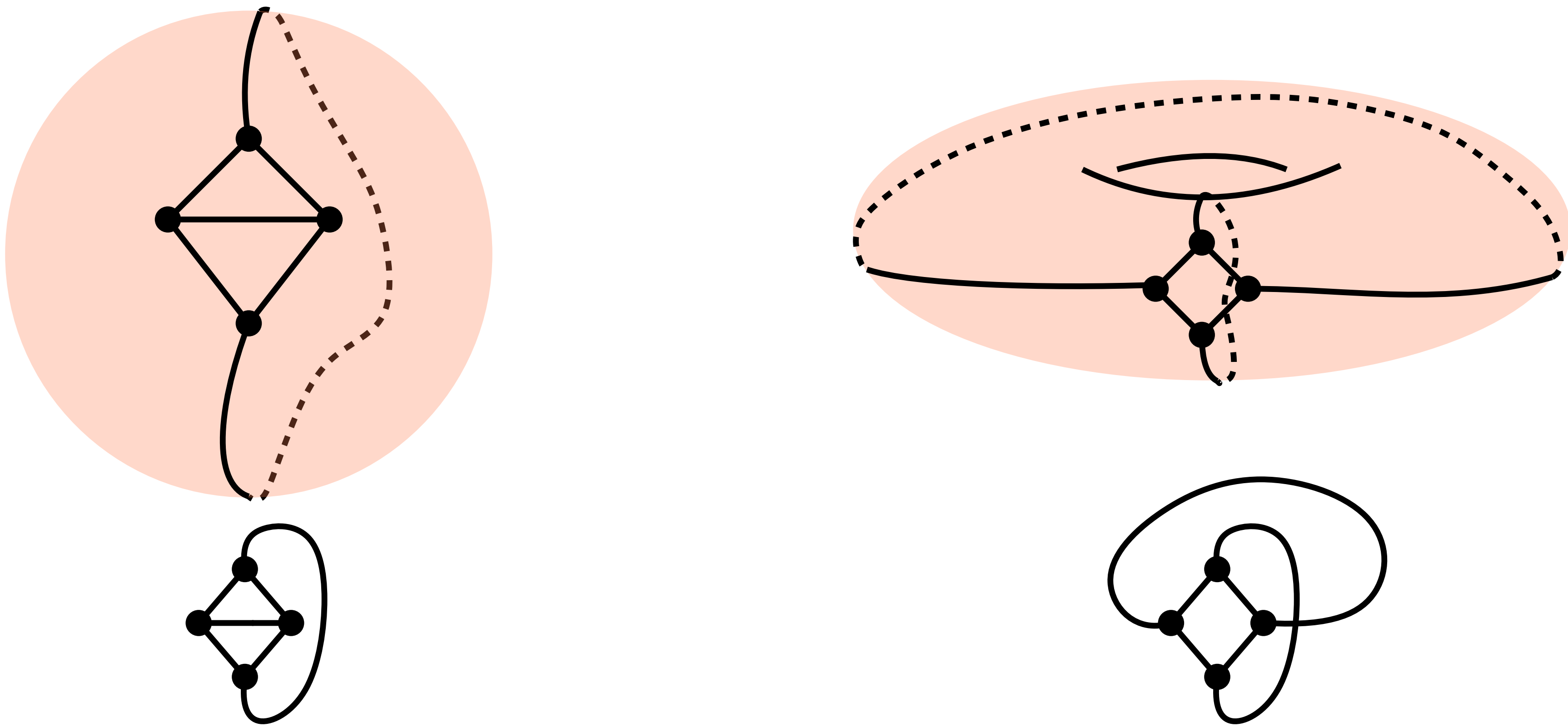
number of steps

Normalisation of random closed linear terms

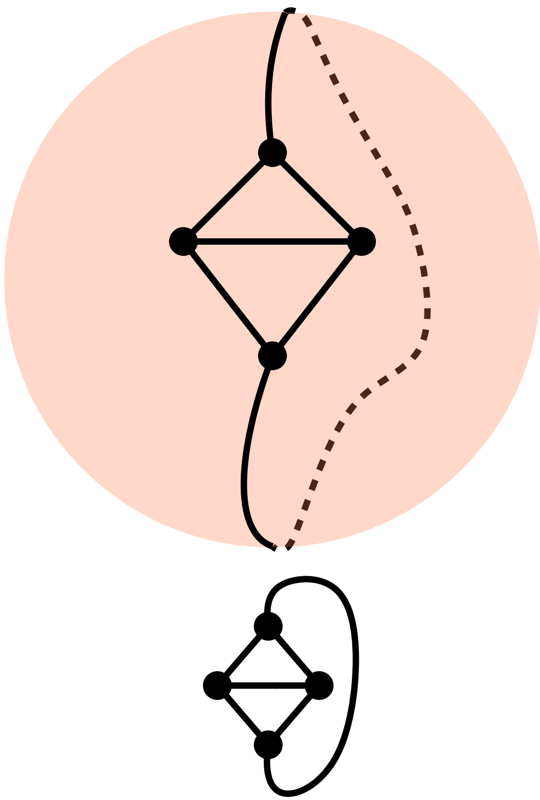
Q: How many steps does it take for a random closed linear term to reach normal form, on average?

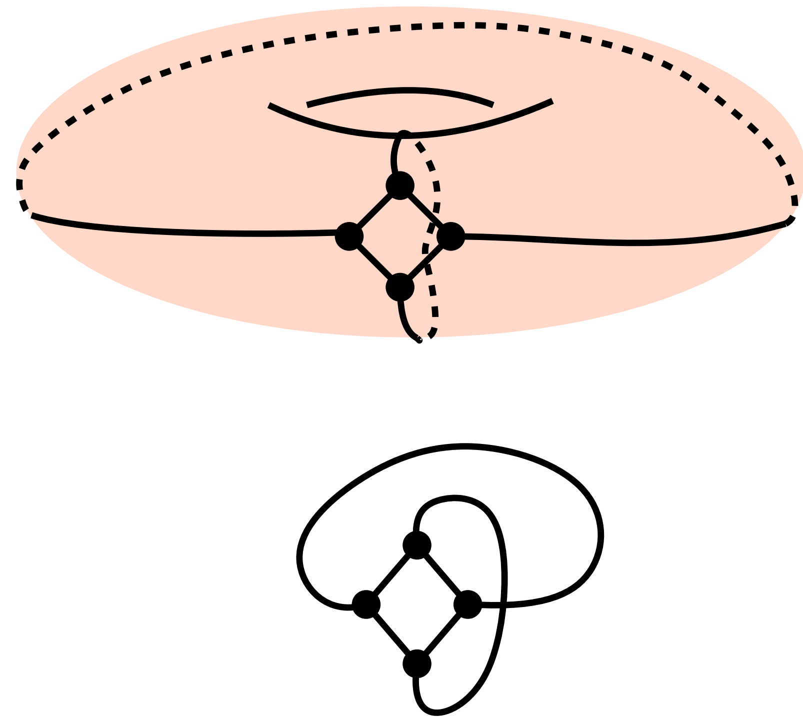


What are maps?



What are maps?



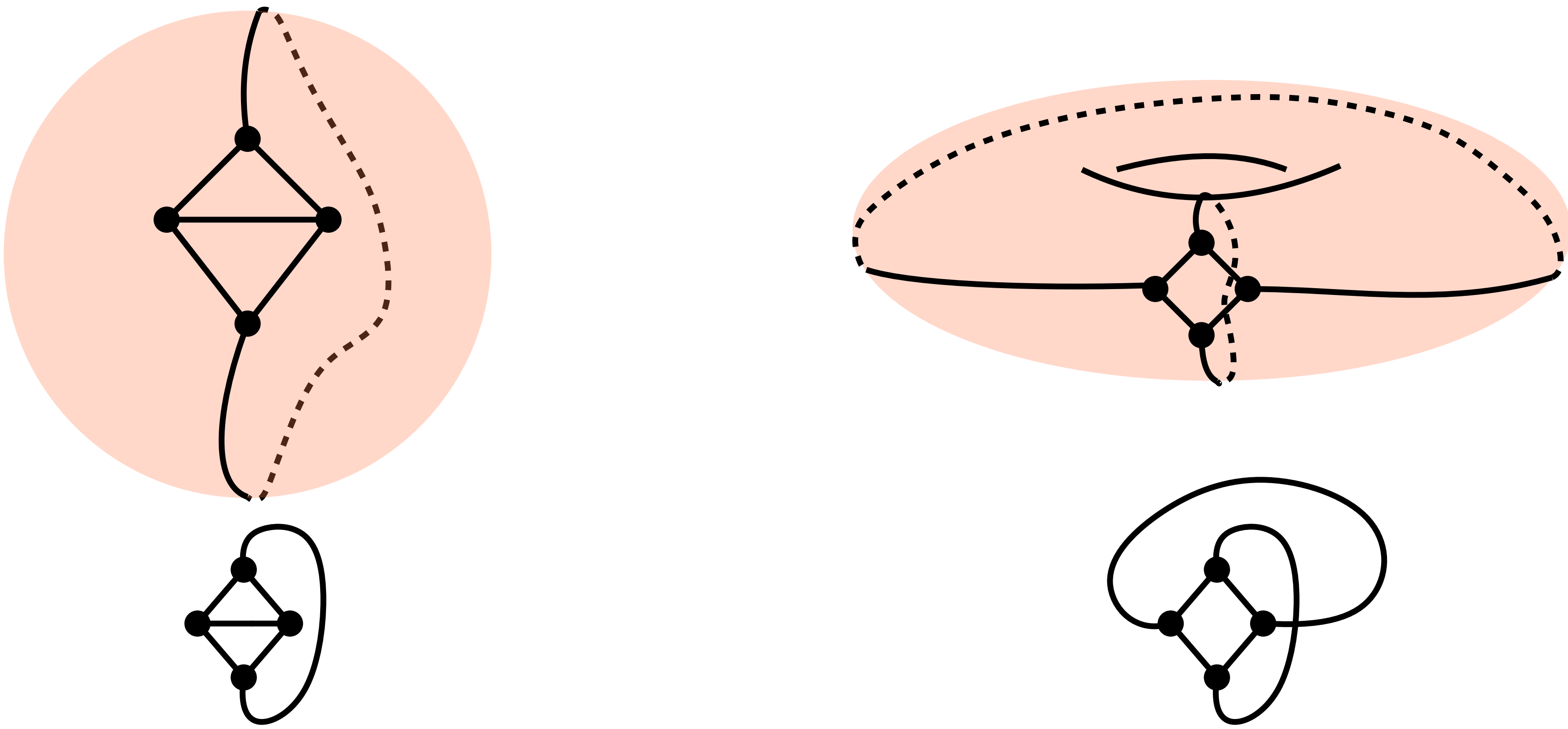


4CT...

- A central object in modern combinatorics, but not only that: probability, algebraic geometry, theoretical physics...

scaling limits... matrix integrals, Witten's conjecture, . . .

What are maps?

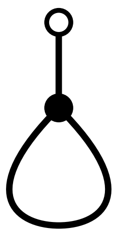


- A central object in modern combinatorics, but not only that: probability, algebraic geometry, theoretical physics...

- Their enumeration was pioneered by Tutte in the 60s, as part of his approach to the four colour theorem.

Decomposing rooted trivalent maps

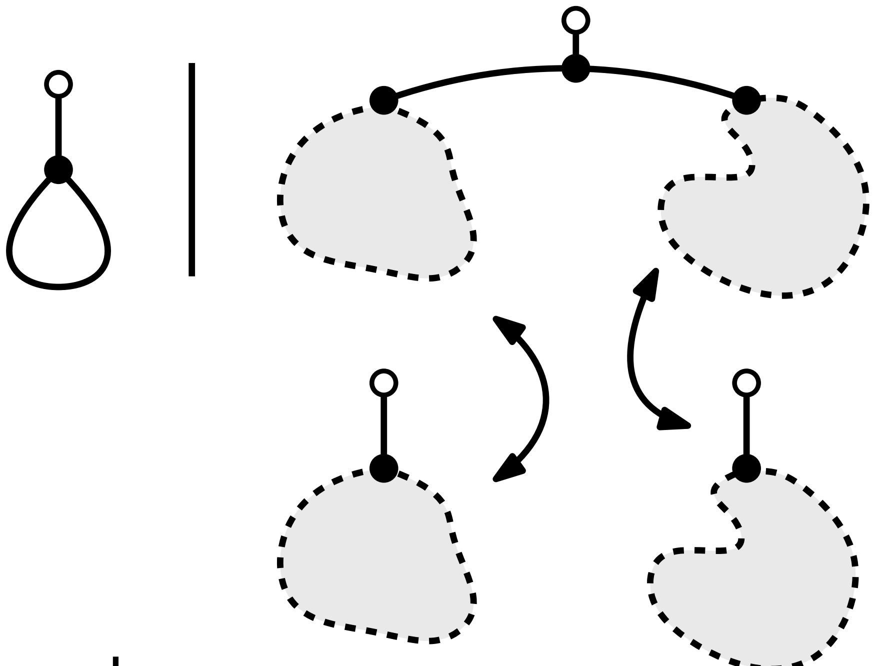
Decomposing rooted trivalent maps



edges

$$\mathsf{T}(z) = z^2$$

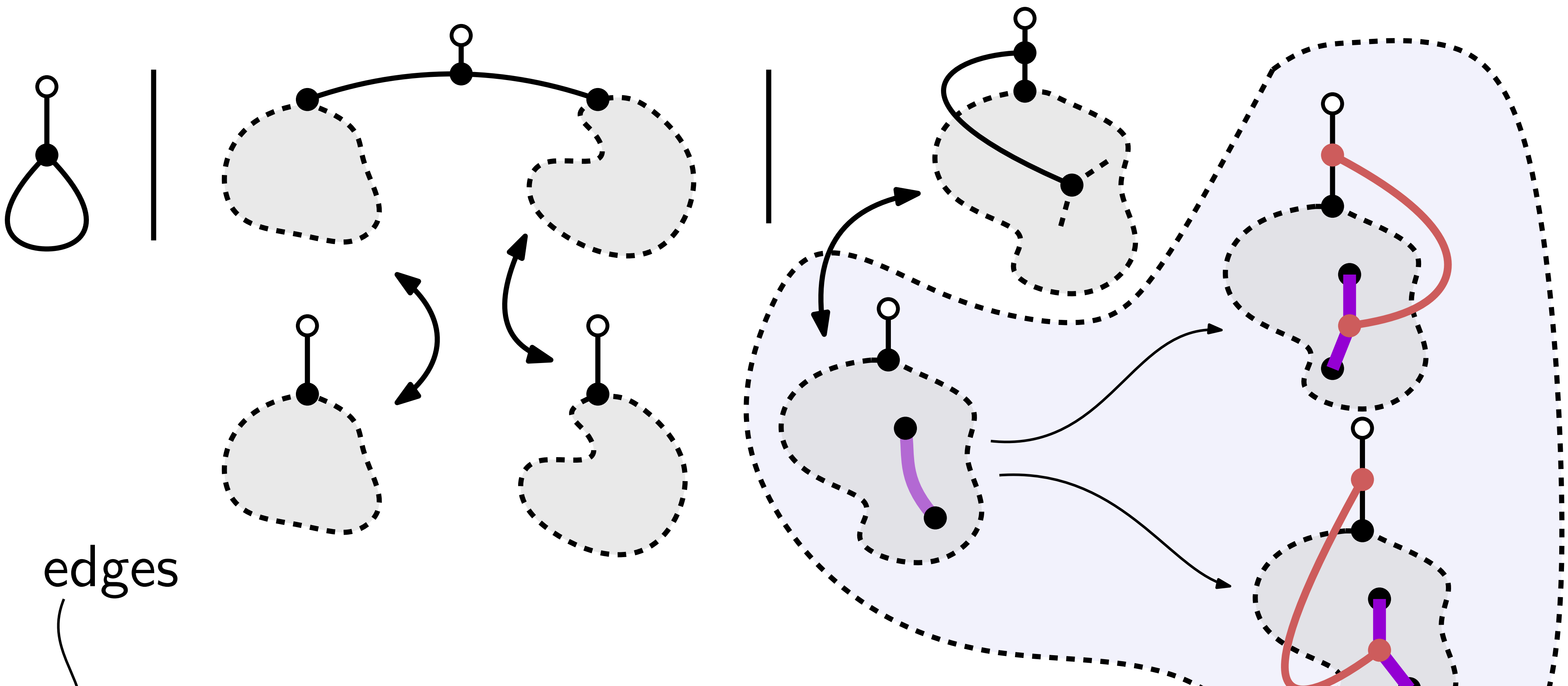
Decomposing rooted trivalent maps



edges

$$\mathsf{T}(z) = z^2 \; + \; z\mathsf{T}(z)^2$$

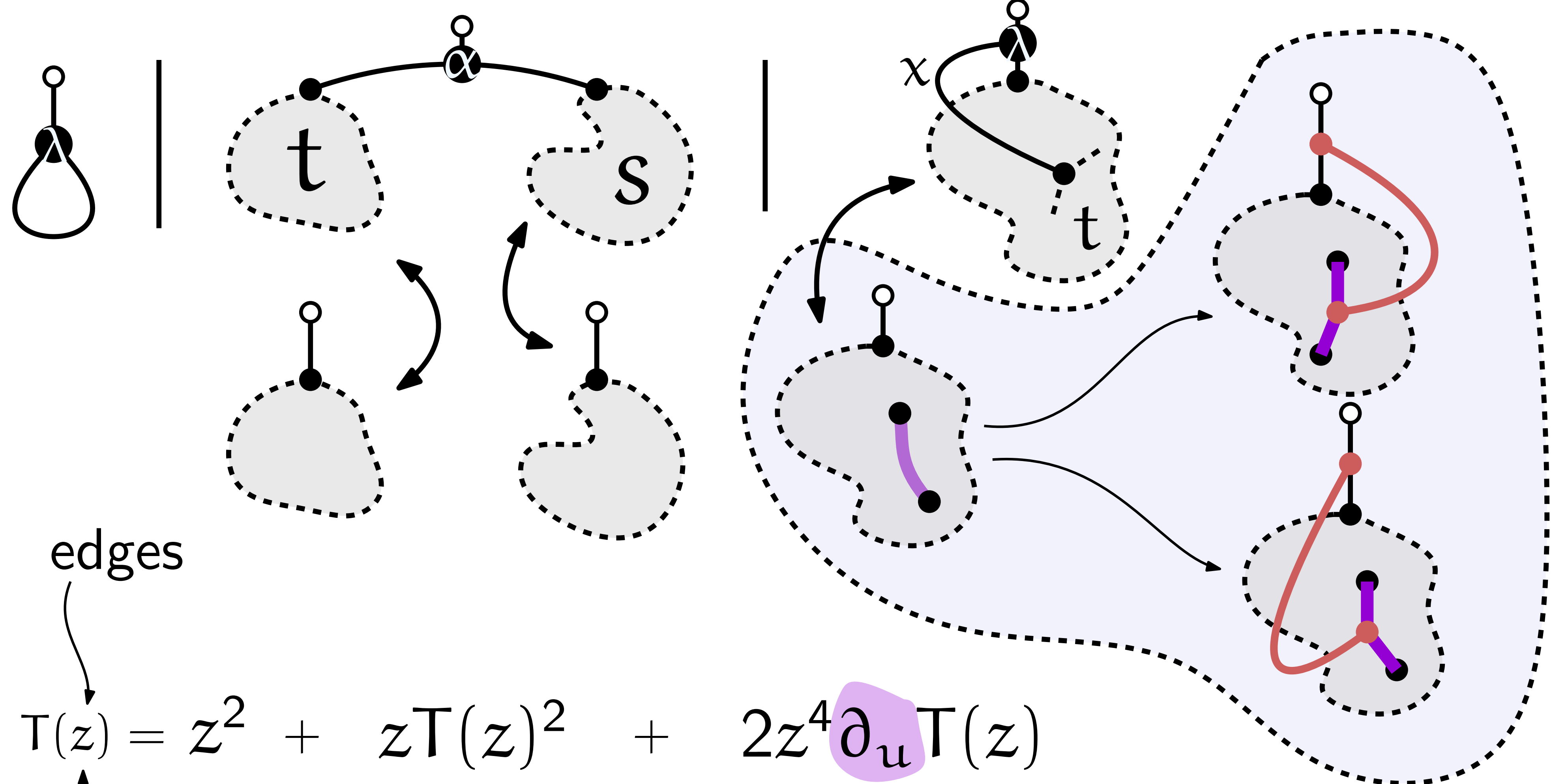
Decomposing rooted trivalent maps



$$\mathsf{T}(z) = z^2 + z\mathsf{T}(z)^2 + 2z^4\partial_u \mathsf{T}(z)$$

Decomposing rooted trivalent maps and closed linear terms!

$\lambda x.x$ | $(s\ t)$ | $\lambda x.t$ → $\lambda x.t[u := (x\ u)]$ or $\lambda x.t[u := (u\ x)]$

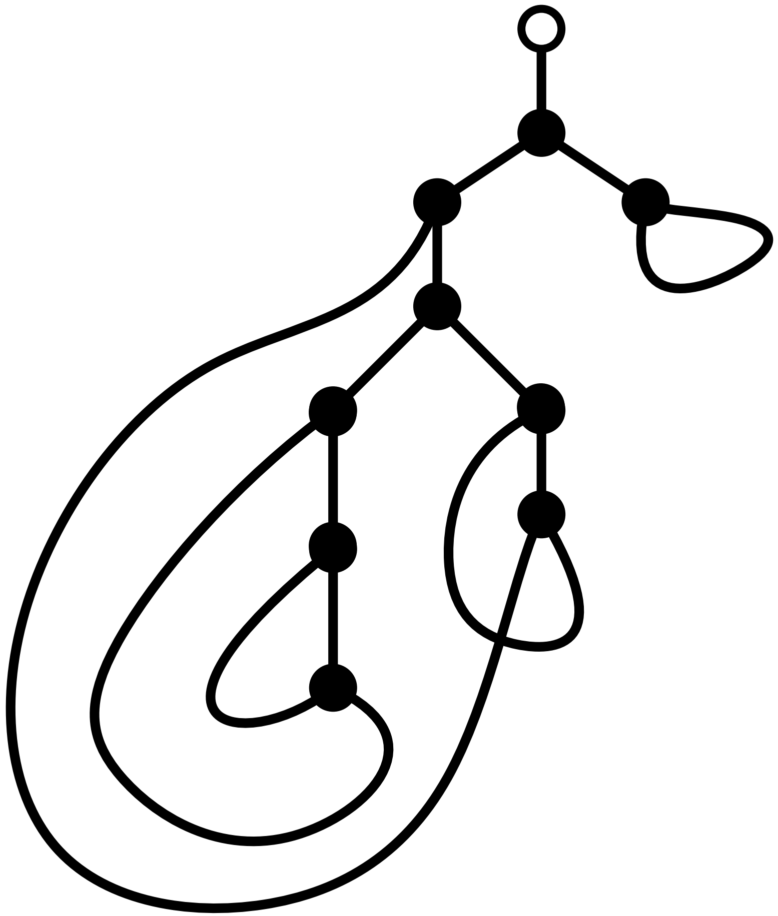


edges

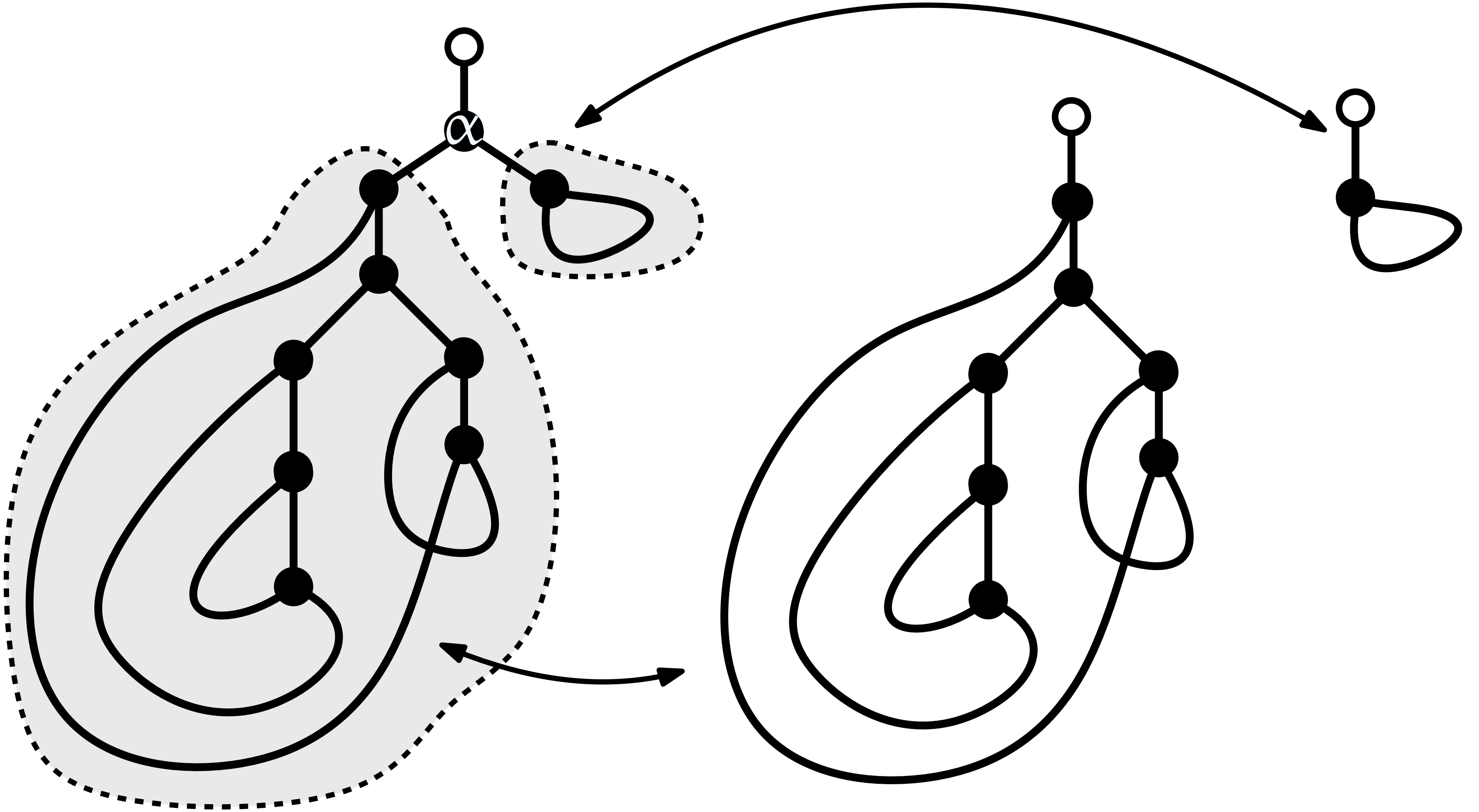
$$T(z) = z^2 + zT(z)^2 + 2z^4\partial_u T(z)$$

subterms

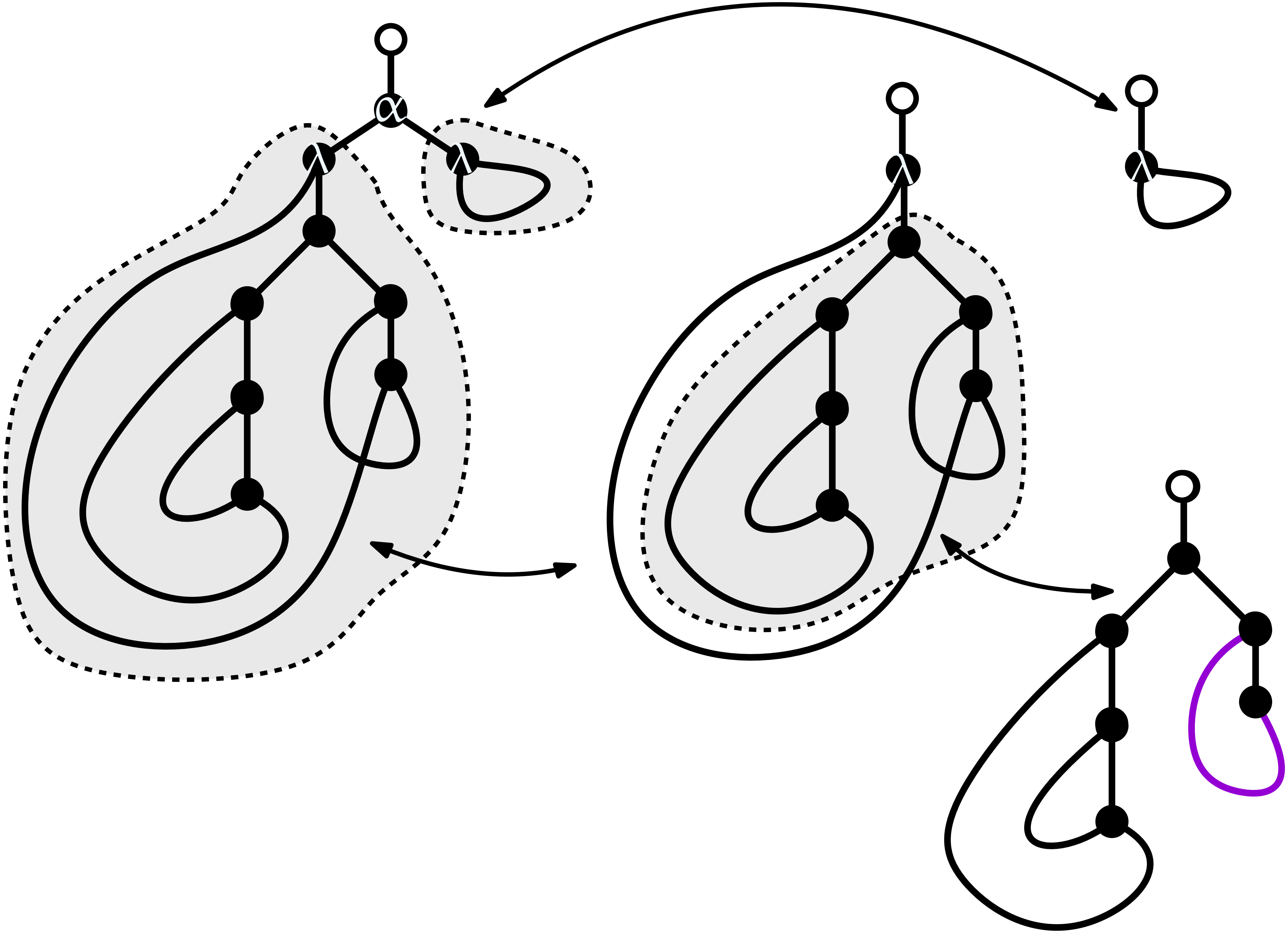
Decomposing rooted trivalent maps



Decomposing rooted trivalent maps

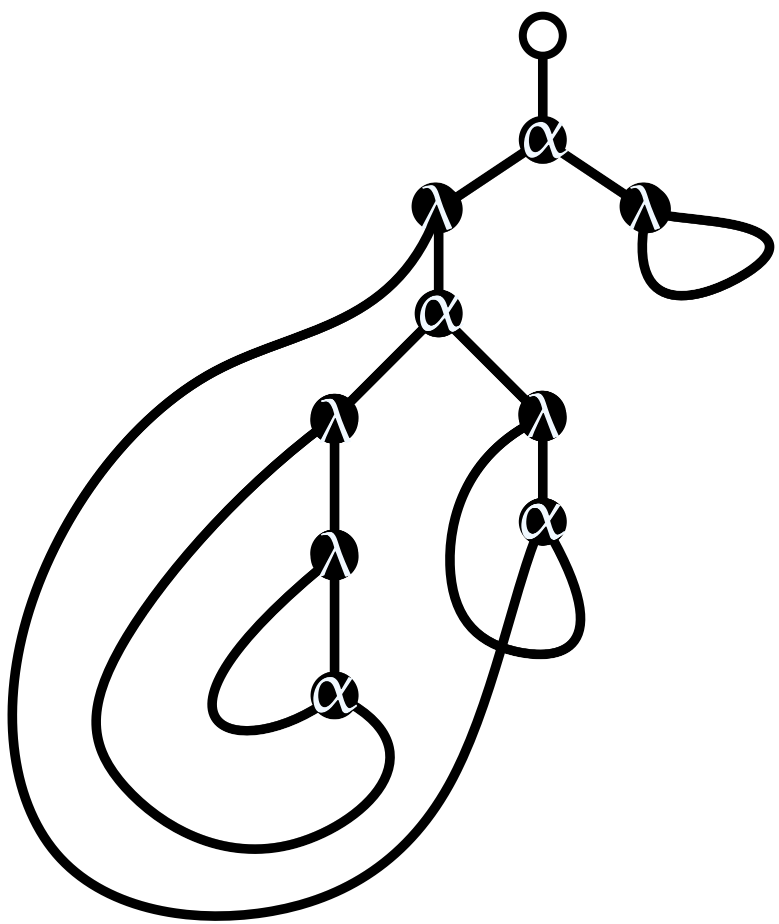


Decomposing rooted trivalent maps



Decomposing rooted trivalent maps

$(\lambda x.x)\ (\lambda y.(\lambda z.z\ y)\ (\lambda w.\lambda u.w\ u))$



Linking terms and maps

- In 2013, Bodini, Gardy, Jacquot, describe a series of bijections:

  rooted trivalent maps $\leftrightarrow$ closed linear terms
  rooted (2,3)-valent maps $\leftrightarrow$ closed affine terms

In the same year, together with Gittenberger, they study:

$\mathrm{BCI}(p)$ terms (each bound variable appears p times)

general closed λ-terms

Linking terms and maps

- In 2013, Bodini, Gardy, Jacquot, describe a series of bijections:

  rooted trivalent maps $\leftrightarrow$ closed linear terms

  rooted (2,3)-valent maps $\leftrightarrow$ closed affine terms

  In the same year, together with Gittenberger, they study:

  $\mathrm{BCI}(p)$ terms (each bound variable appears p times)

  general closed λ-terms

- In 2014, Zeilberger and Giorgetti describe a bijection:

  rooted planar maps $\leftrightarrow$ normal planar lambda terms

Linking terms and maps

- In 2013, Bodini, Gardy, Jacquot, describe a series of bijections:

  rooted trivalent maps $\leftrightarrow$ closed linear terms

  rooted (2,3)-valent maps $\leftrightarrow$ closed affine terms

  In the same year, together with Gittenberger, they study:

  $\mathrm{BCI}(\mathrm{p})$ terms (each bound variable appears p times)

  general closed λ-terms

- In 2014, Zeilberger and Giorgetti describe a bijection:

  rooted planar maps $\leftrightarrow$ normal planar lambda terms

Both make use of decompositions in the style of Tutte!
(cf. the approach of Arquès-Béraud in 2000)

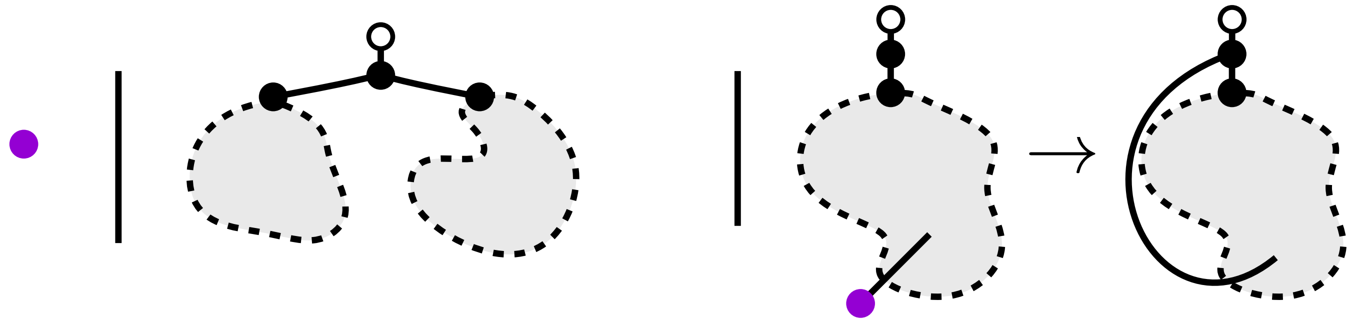
Our strategy:

1) Track evolution of patterns through decompositions of maps/λ-terms

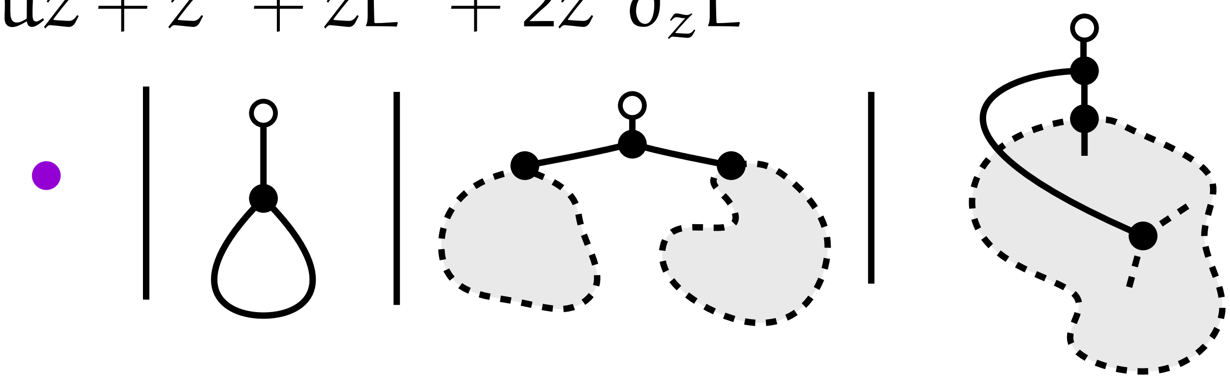
Our strategy:

1) Track evolution of patterns through decompositions of maps/λ-terms

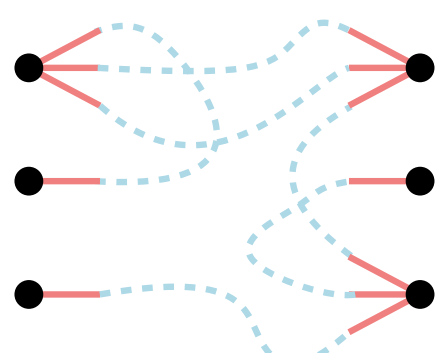
- $T = uz + zL^2 + z\partial_u L$



- $T = uz + z^2 + zL^2 + 2z^4 \partial_z L$



- $T = uz^2 + z^4 + z^5 \frac{\partial}{\partial z} \left(\ln \left(\exp(z^2/2) \odot \exp(z^3/3 + uz) \right) \right)$



Our strategy:

1) Track evolution of patterns through decompositions of maps/λ-terms

different decompositions $\rightsquigarrow$ differential equations, Hadamard products, ...

generating functions divergent away from 0

2) Develop tools for rapidly growing coefficients, based on:

- Moment pumping
- Bender's theorem for compositions $F(z, G(z))$
- Coefficient asymptotics of Cauchy products

$$[z^n](A(z) \cdot B(z)) \sim a_n b_0 + a_0 b_n + O(a_{n-1} + b_{n-1})$$

for A, B, G divergent and F analytic

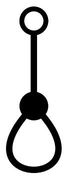
Mean number of β-redices in closed terms

Tracking redices: starts off easy...

Mean number of β-redices in closed terms

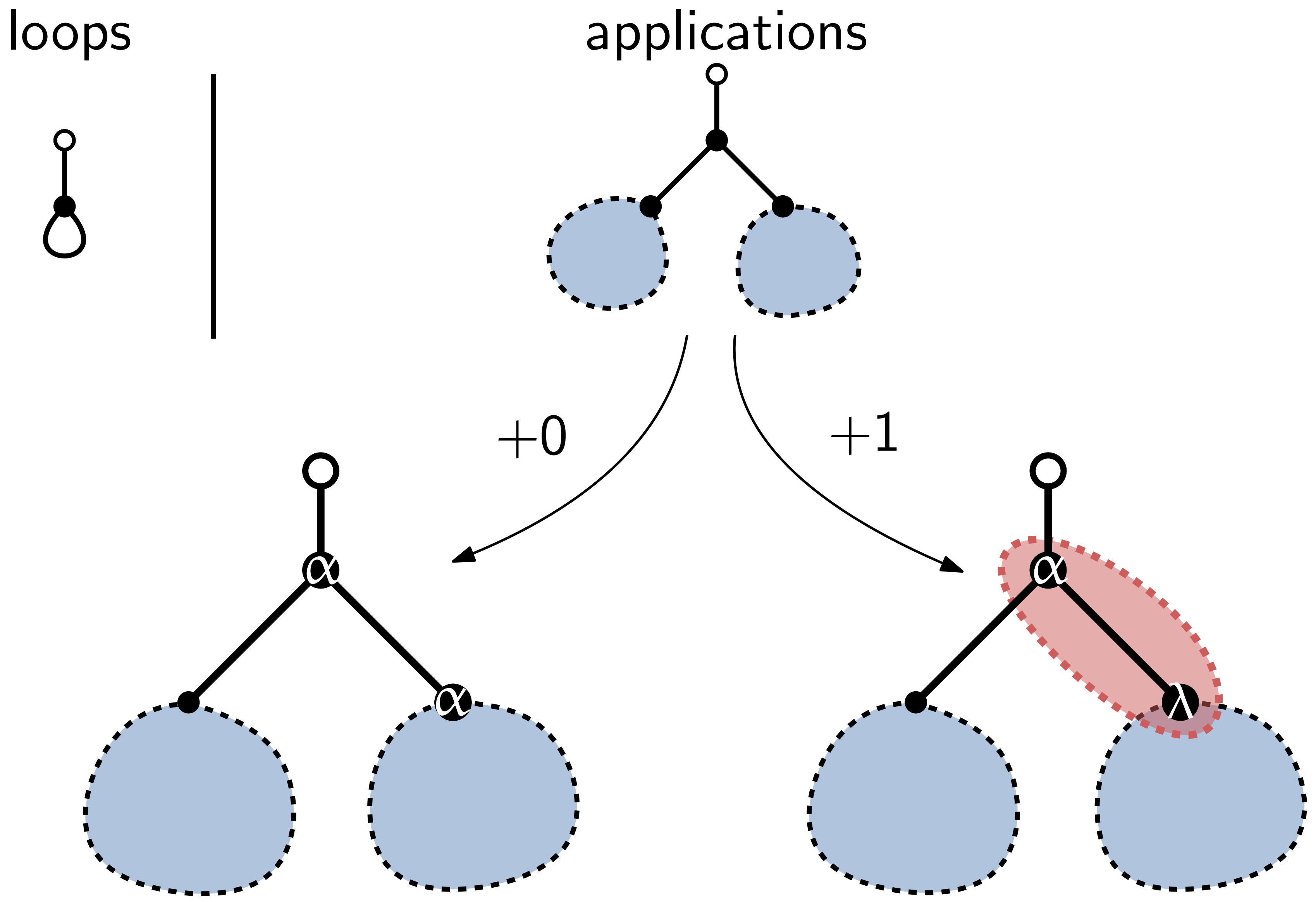
Tracking redices: starts off easy...

loops



Mean number of β-redices in closed terms

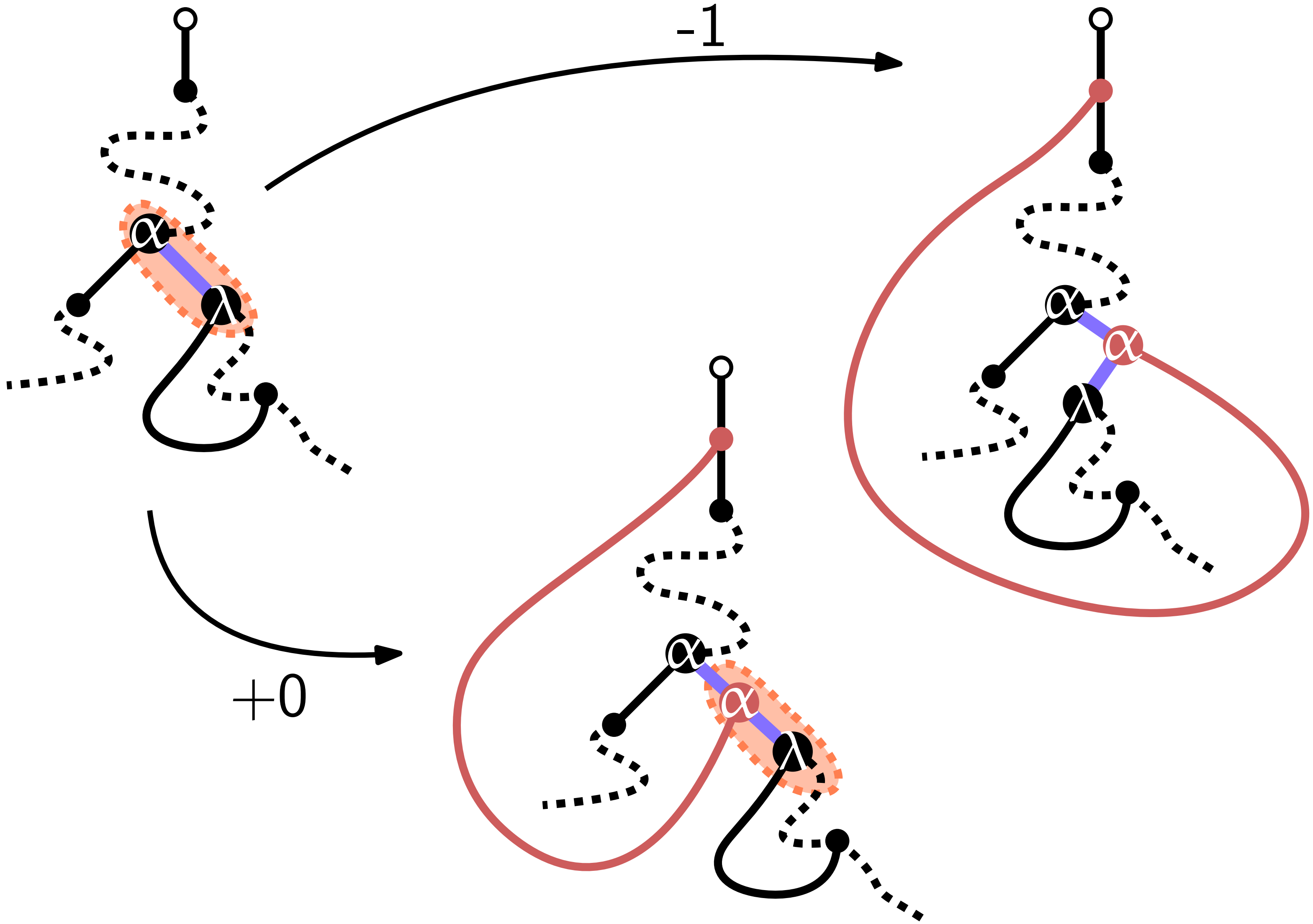
Tracking redices: starts off easy...



Mean number of β-redices in closed terms

Tracking redices: then gets harder!

Abstractions, subcase 1.1



Mean number of β-redices in closed terms

Translating to a differential equation and pumping

$$\mathsf{T} = z^2 + z\mathsf{T}^2 + z^3(1 + (\mathrm{r} - 1)z\mathsf{T})\left(\frac{z(\mathrm{r}+5)\partial_z \mathsf{T}}{3} - (\mathrm{r}^2 - 1)\partial_r \mathsf{T})\right) + \frac{z^4(\mathrm{r}-1)^2\mathsf{T}^2}{3} + \frac{4z^3(\mathrm{r}-1)\mathsf{T}}{3}$$

Mean number of β-redices in closed terms

Translating to a differential equation and pumping

$$T = z^2 + zT^2 + z^3(1 + (r-1)zT)\left(\frac{z(r+5)\partial_z T}{3} - (r^2-1)\partial_r T\right) + \frac{z^4(r-1)^2T^2}{3} + \frac{4z^3(r-1)T}{3}$$

Let X_n be the random variable given by number of redices in a closed linear term of size $n \in 3\mathbb{N} + 2$. Then

$$\mathbb{E}(X_n) \sim \frac{n}{24}$$

$$\mathbb{V}(X_n) \sim \frac{n}{24}$$

Pretty far from $\frac{n}{21}$!
Expect a linear number of reproducing ones.

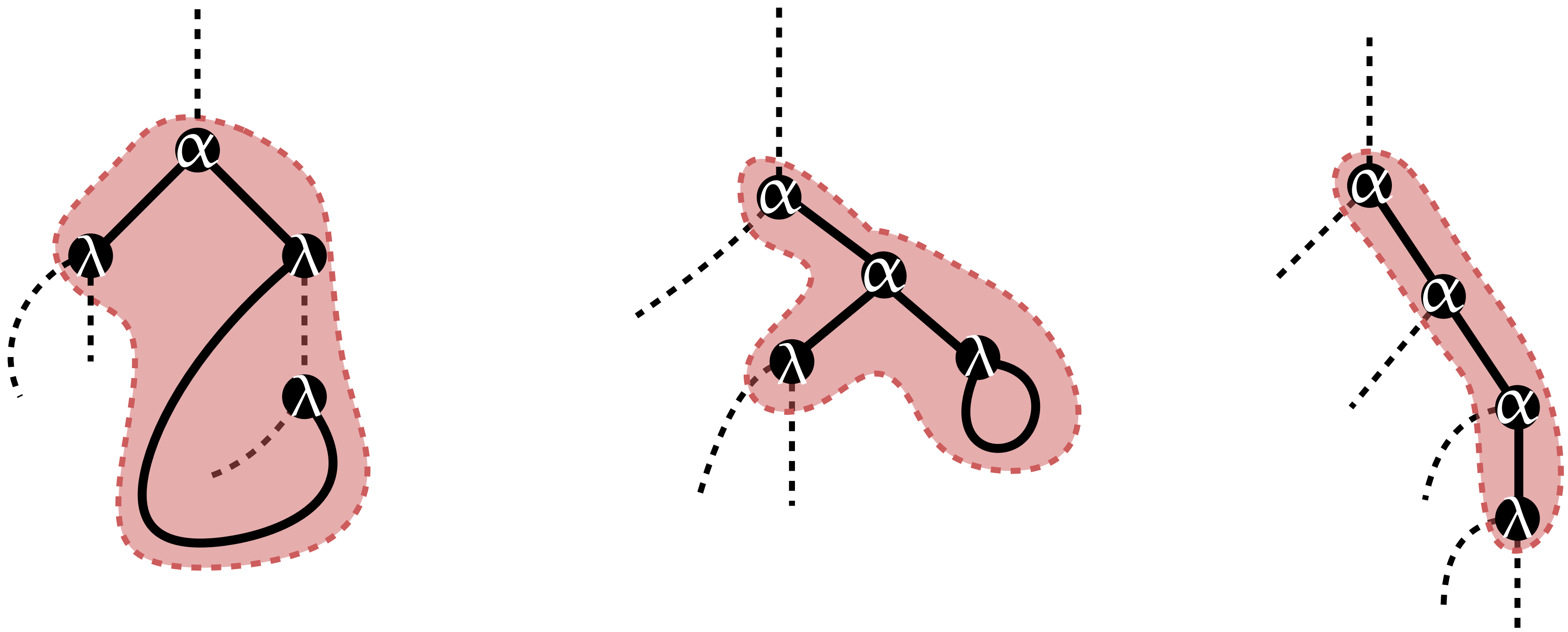
A lower bound for normalisation

Refining our counting to track reproducing redices:

A lower bound for normalisation

(see JJ Lévy's thesis)

Refining our counting to track reproducing redices:



$$p_1 = (\lambda x.C[(x\ u)])(\lambda y.t) \xrightarrow{\beta} C[((\lambda y.t)\ u)]$$

$$p_2 = (\lambda x.x)(\lambda y.t_1)t_2 \xrightarrow{\beta} (\lambda y.t_1)t_2$$

$$p_3 = ((\lambda x.\lambda y.t_1)\ t_2)\ t_3 \xrightarrow{\beta} (\lambda y.t_1[x := t_2])\ t_3$$

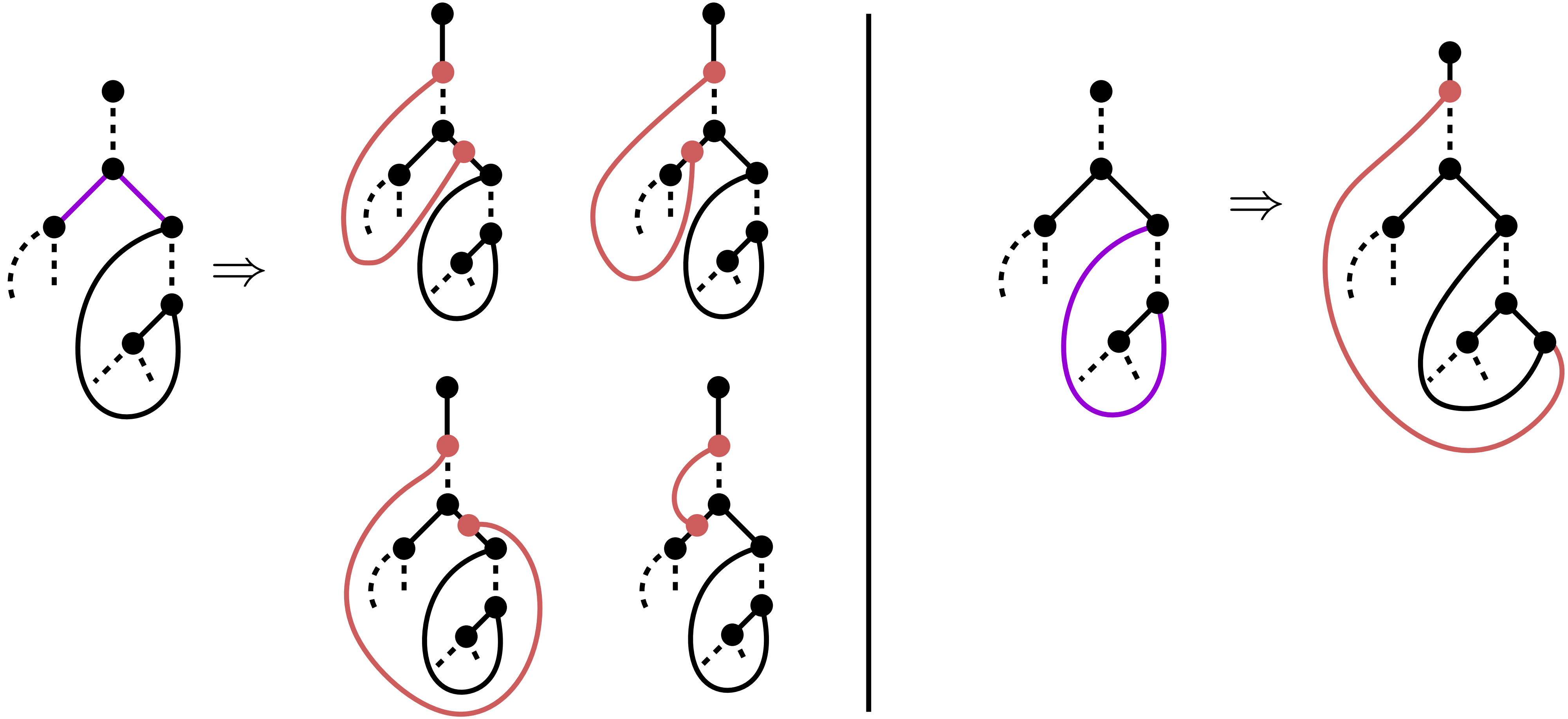
Enumerating $\mathfrak{p}_1$-patterns

- Tracking the creation/destruction of patterns during the recursive decomposition:

Enumerating $\mathfrak{p}_1$-patterns

- Tracking the creation/destruction of patterns during the recursive decomposition:

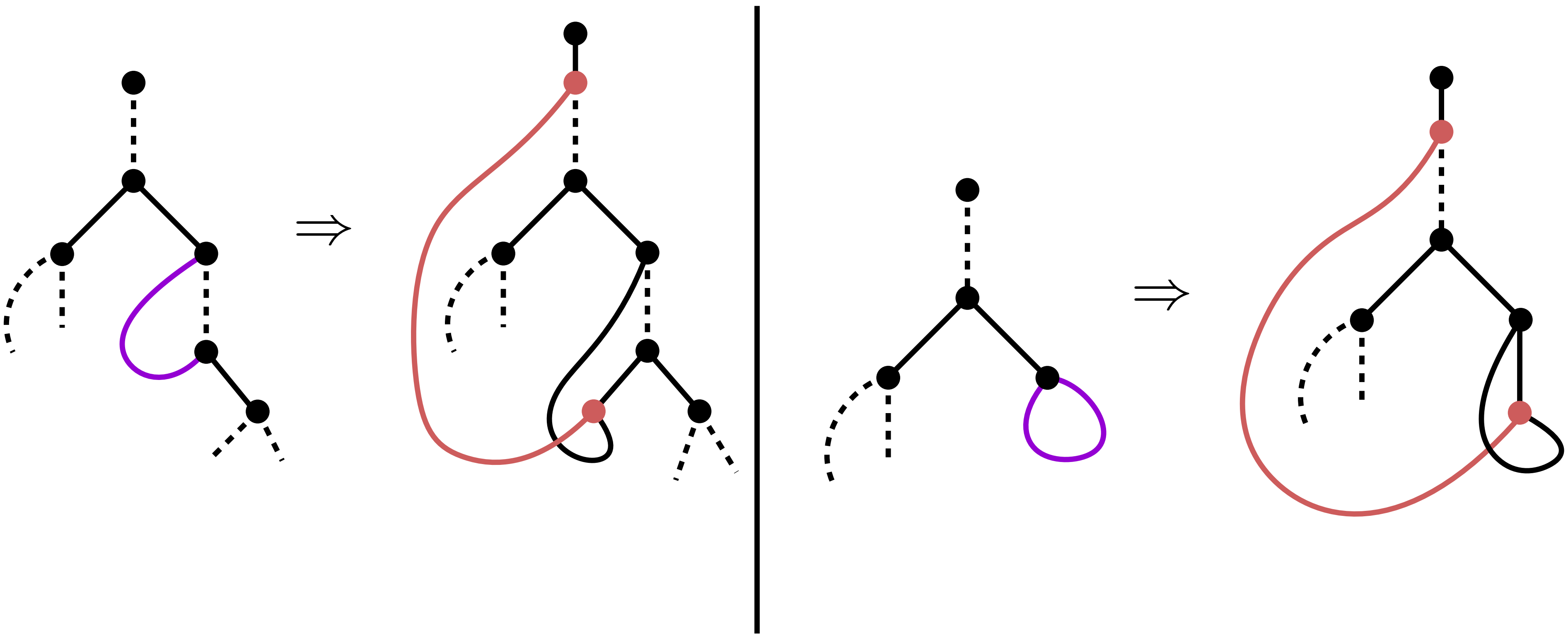
Cuts destroying a $\mathfrak{p}_1$-pattern:



Enumerating p_1-patterns

- Tracking the creation/destruction of patterns during the recursive decomposition:

Cuts creating a p_1-pattern:



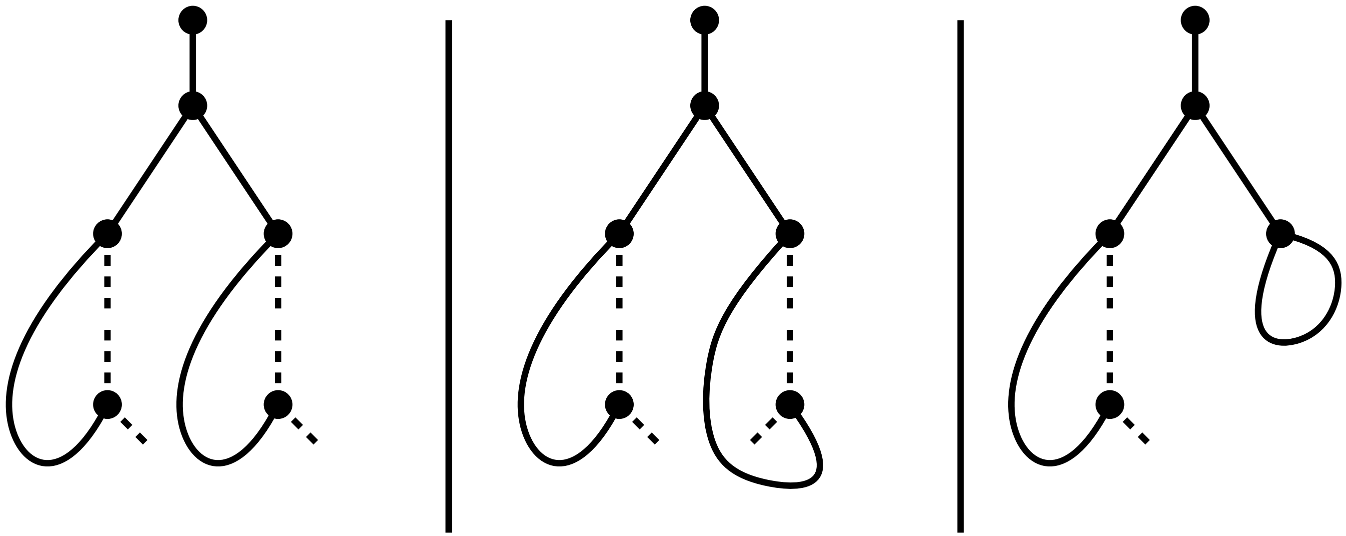
Thus we also need to keep track of:

$$C_1[\lambda x.C_2[(t_1\ x)]](\lambda y.t_2)] \qquad C_1[(\lambda x.x)(\lambda y.t_2)]$$

Enumerating p_1-patterns

- Tracking the creation/destruction of patterns during the recursive decomposition:

  Applications creating p_1 and auxilliary patterns:



Thus, for an app. of the form $(l_1\ \lambda y.t_1)$ we need to consider how l_1 was formed.

Enumerating p_1-patterns

- Thus we have the following equations:

$$S = \Lambda + A$$

$$\Lambda = z^2 + 2z^4 S_z + (v - u + 4(1-u))z^3 S_u + (u - v + 4(1-v))z^3 S_v$$

$$A = zS^2 + (u-1)z(z^4 S_z + (v - u + 2(1-u))z^3 S_u + 2(1-v)z^3 S_v) \cdot \Lambda$$
$$+(v-1)z(z^2 + z^4 S_z + (u - v + 2(1-v))z^3 S_u + 2(1-u)z^3 S_u) \cdot \Lambda$$

- Extracting the mean:

$$\partial_u S|_{u=1,v=1}$$
$$= \left(2zS\partial_u S + 2z^4 \partial_{z,u} S + z^7 \partial_z S + 2z^9 (\partial_z S)^2 - 5z^3 \partial_u S + z^3 \partial_v S\right)|_{u=1,v=1}$$

Enumerating p_1-patterns

- Thus we have the following equations:

$$S = \Lambda + A$$

$$\Lambda = z^2 + 2z^4 S_z + (v - u + 4(1-u))z^3 S_u + (u - v + 4(1-v))z^3 S_v$$

$$A = zS^2 + (u-1)z(z^4 S_z + (v - u + 2(1-u))z^3 S_u + 2(1-v)z^3 S_v) \cdot \Lambda$$
$$+(v-1)z(z^2 + z^4 S_z + (u - v + 2(1-v))z^3 S_u + 2(1-u)z^3 S_u) \cdot \Lambda$$

- Extracting the mean:

$$\partial_u S|_{u=1,v=1}$$
$$= \left(2zS\partial_u S + 2z^4 \partial_{z,u} S + z^7 \partial_z S + 2z^9 (\partial_z S)^2 - 5z^3 \partial_u S + z^3 \partial_v S\right)|_{u=1,v=1}$$

Enumerating p_1-patterns

- Thus we have the following equations:

$$S = \Lambda + A$$

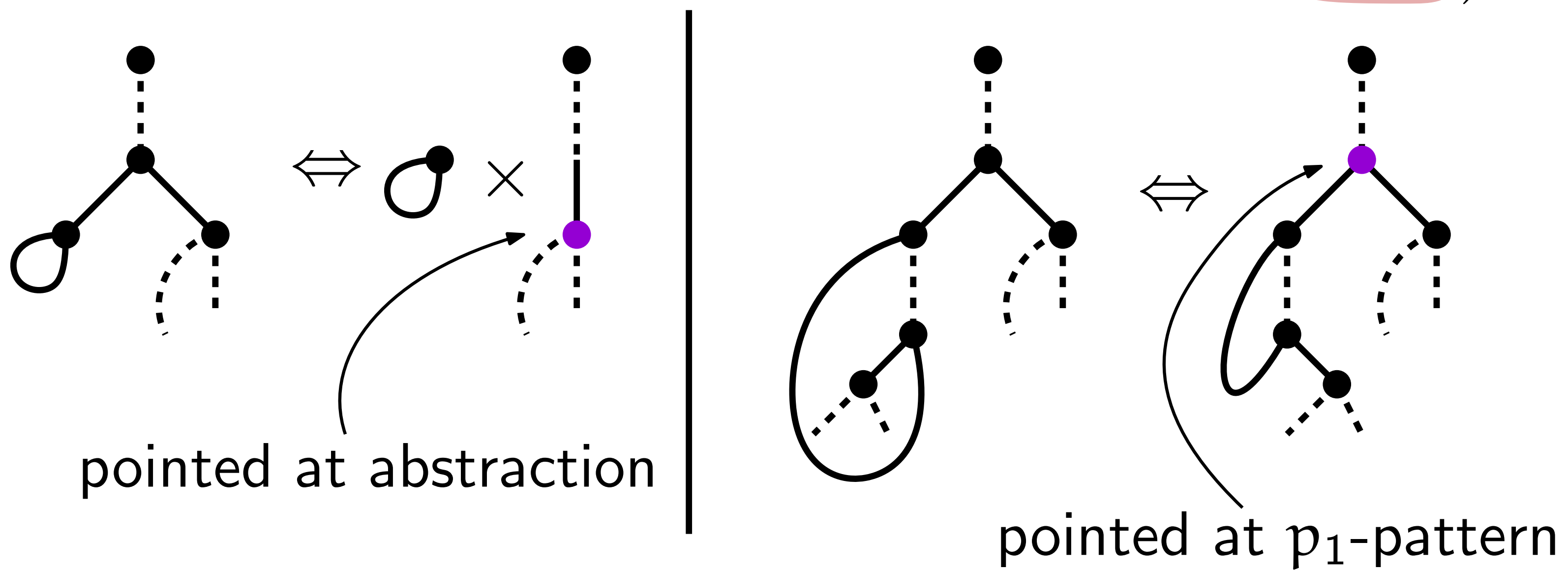
$$\Lambda = z^2 + 2z^4 S_z + (v - u + 4(1-u))z^3 S_u + (u - v + 4(1-v))z^3 S_v$$

$$A = zS^2 + (u-1)z(z^4 S_z + (v - u + 2(1-u))z^3 S_u + 2(1-v)z^3 S_v) \cdot \Lambda$$
$$+(v-1)z(z^2 + z^4 S_z + (u - v + 2(1-v))z^3 S_u + 2(1-u)z^3 S_u) \cdot \Lambda$$

- Extracting the mean:

$\partial_u S|_{u=1,v=1}$ bijection: $\partial_v \leftrightarrow \partial_u$

$$= \left(2zS\partial_u S + 2z^4 \partial_{z,u} S + z^7 \partial_z S + 2z^9 (\partial_z S)^2 - 5z^3 \partial_u S + z^3 \partial_v S\right)|_{u=1,v=1}$$



Enumerating $\mathfrak{p}_1$-patterns

- Finally we obtain a mean number of occurences:

$$\mathbb{E}[\#\ \mathfrak{p}_1 \text{ patterns}] \sim \frac{1}{6}$$

Enumerating $\mathfrak{p}_1$-patterns and $\mathfrak{p}_2$-patterns

- Finally we obtain a mean number of occurences:

$$\mathbb{E}[\#\ \mathfrak{p}_1 \text{ patterns}] \sim \frac{1}{6}$$

- Analogously, we have a mean number of occurences for $\mathfrak{p}_2$:

$$\mathbb{E}[\#\ \mathfrak{p}_2 \text{ patterns}] \sim \frac{1}{48}$$

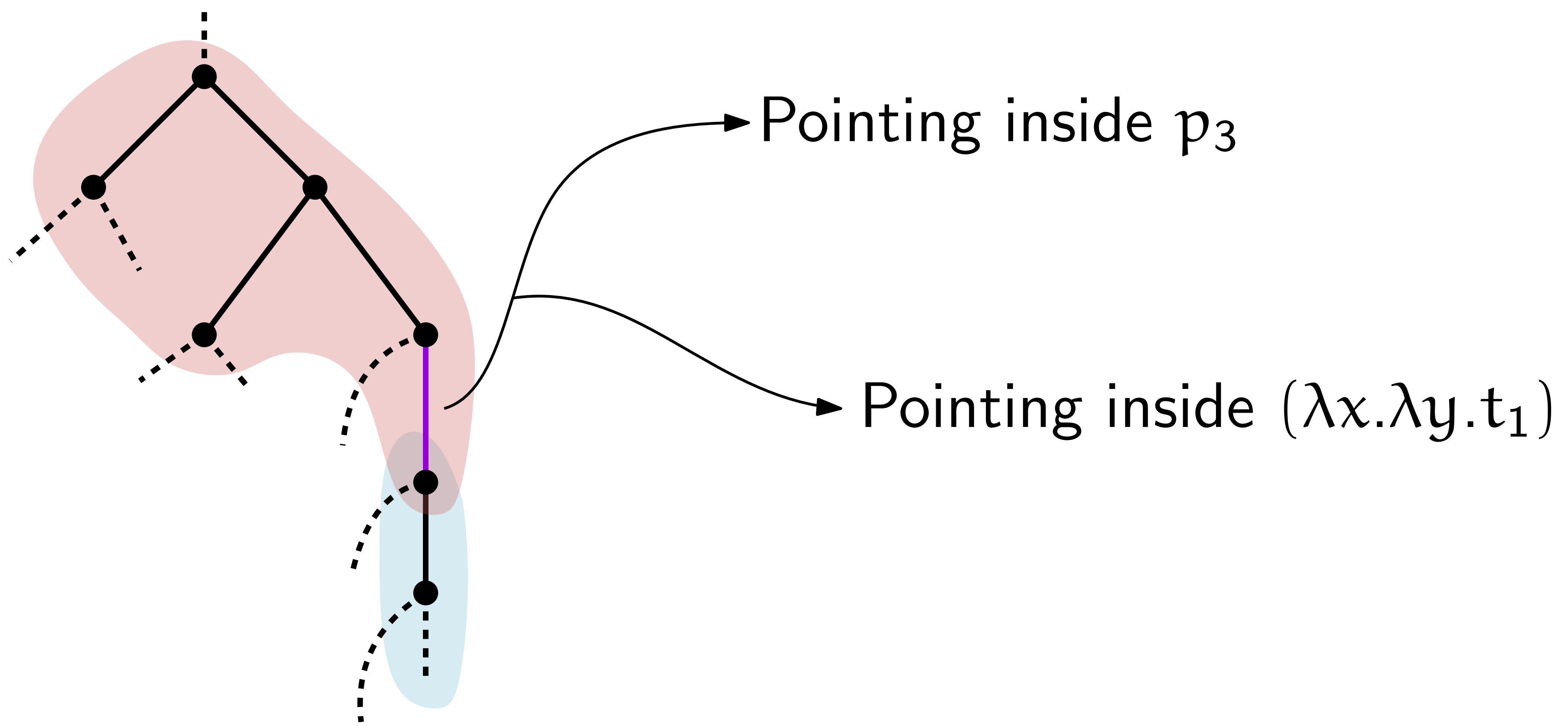
Both are asymptotically constant in expectation!

Enumerating p_3-patterns

- As before, we'll also need to enumerate auxilliary patterns:

$$(\lambda x.\lambda y.t_1) \qquad (\lambda x.\lambda y.t_1)\ t_2 \qquad (\lambda x.\lambda y.t_1)\ t_2\ t_3\ \ (p_3)$$

- However we run into a problem:



Enumerating $\mathfrak{p}_3$-patterns

- Generatingfunctionology fails, we revert to more elementary methods:

$$\mathbb{E}(V_n) = \mathbb{E}(V_n|\Lambda_n) \cdot \frac{|\Lambda_n|}{|L_n|} + \mathbb{E}(V_n|A_n) \cdot \frac{|A_n|}{|L_n|}$$

Enumerating p_3-patterns

- Generatingfunctionology fails, we revert to more elementary methods:

asymptotic contribution $\approx \frac{\mathbb{E}(V_{n-3})}{n}$

$$\mathbb{E}(V_n) = \mathbb{E}(V_n|\Lambda_n) \cdot \frac{|\Lambda_n|}{|L_n|} + \mathbb{E}(V_n|A_n) \cdot \frac{|A_n|}{|L_n|}$$

Enumerating p_3-patterns

- Generatingfunctionology fails, we revert to more elementary methods:

asymptotic contribution $\approx \frac{\mathbb{E}(V_{n-3})}{n}$

$$\mathbb{E}(V_n) = \mathbb{E}(V_n | \Lambda_n) \cdot \frac{|\Lambda_n|}{|L_n|} + \mathbb{E}(V_n | A_n) \cdot \frac{|A_n|}{|L_n|}$$

Magic: linear over *families* of all possible abstractions created via cuts from a fixed term!

$$\overline{X}_n = (2n-12)\overline{X}_{n-3} 2\overline{Y}_{n-3}$$
$$\overline{Y}_n = (2n-6)Y_{n-3} - 6Y_{n-3}$$
$$\overline{Z}_n = 2(n-4)(Z + \mathbf{1}_{\Lambda_n})$$

where: X_n counts # of p_3 patt. over terms of size n

Y_n is the same for the pattern $(\lambda x.\lambda y.t_1)\ t_2$, and

Z is the same for the pattern $(\lambda x.\lambda y.t_1)$

The $\overline{V}$ for $V \in \{X_n, Y_n, Z_n\}$ are cummulatives over families of abstractions

The lower bound

Theorem

Let W_n be the random variable given by number of steps required to normalise a linear term of size $n \in 3\mathbb{N} + 2$. Then

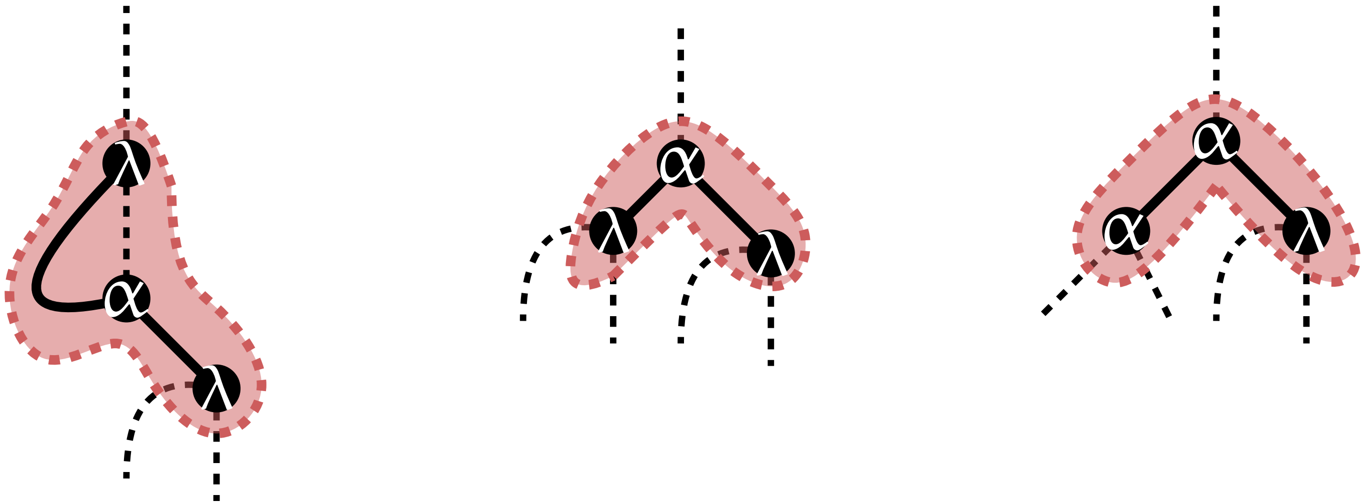
$$\mathbb{E}(W_n) \geqslant \frac{11n}{240}, \text{ for } n \text{ large enough}$$

Open problems

- Precise asymptotics for mean number of steps.

Open problems

- Precise asymptotics for mean number of steps.
- Classify patterns according to their expected number of occurences: constant or linear in n? Other behaviours?



Theorem

$\mathbb{E}(\#(\lambda x.t)\ y) \sim \frac{n}{30}$

$\mathbb{E}(\#(\lambda x.t)\ (\lambda y.t')) \sim \frac{1}{20}$

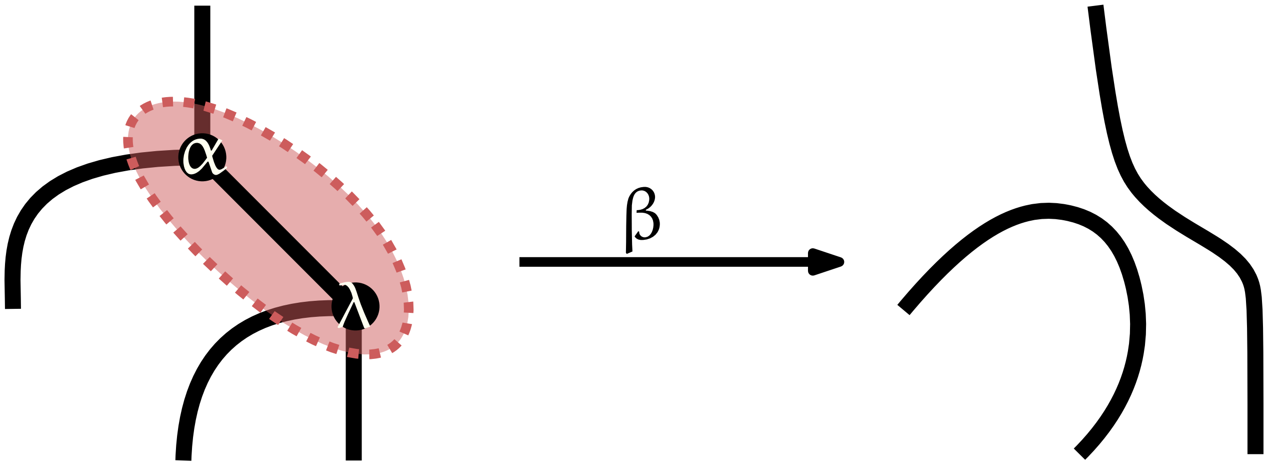
$\mathbb{E}(\#(a\ b)\ (\lambda y.t')) \sim \frac{n}{120}$

Open problems

- Precise asymptotics for mean number of steps.
- Classify patterns according to their expected number of occurences: constant or linear in n? Other behaviours?
- Automatise the process of obtaining specifications tracking occurences of our desired patterns (differential algebra?).

Open problems

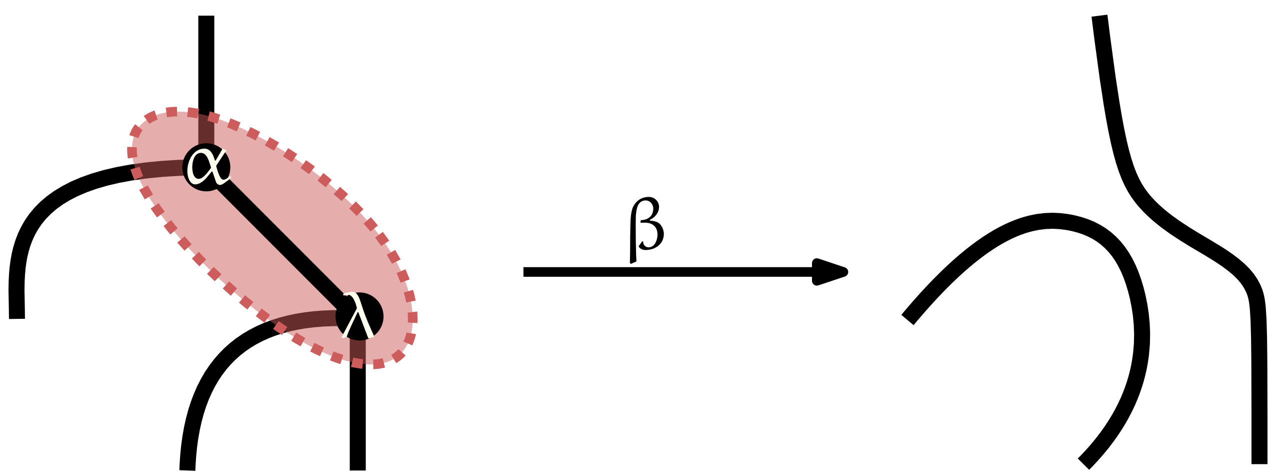
- Precise asymptotics for mean number of steps.
- Classify patterns according to their expected number of occurences: constant or linear in n? Other behaviours?
- Automatise the process of obtaining specifications tracking occurences of our desired patterns (differential algebra?).
- Explore the meaning of β-reduction on maps. Connections to knot theory?



Open problems

- Precise asymptotics for mean number of steps.
- Classify patterns according to their expected number of occurences: constant or linear in n? Other behaviours?
- Automatise the process of obtaining specifications tracking occurences of our desired patterns (differential algebra?).
- Explore the meaning of β-reduction on maps. Connections to knot theory?

Thank you!



Bibliography

[BGGJ13] Bodini, O., Gardy, D., Gittenberger, B., & Jacquot, A. (2013).
Enumeration of Generalized BCI Lambda-terms.
The Electronic Journal of Combinatorics, P30-P30.

[Z16] Zeilberger, N. (2016).
Linear lambda terms as invariants of rooted trivalent maps.
Journal of functional programming, 26.

[AB00] Arques, D., & Béraud, J. F. (2000).
Rooted maps on orientable surfaces, Riccati's equation and continued fractions
Discrete mathematics, 215(1-3), 1-12.

[BFSS01] Banderier, C., Flajolet, P., Schaeffer, G., & Soria, M. (2001).
Random maps, coalescing saddles, singularity analysis, and Airy phenomena.
Random Structures & Algorithms, 19(3-4), 194-246.

Bibliography

[BR86] Bender, E. A., & Richmond, L. B. (1986).
A survey of the asymptotic behaviour of maps.
Journal of Combinatorial Theory, Series B, 40(3), 297-329.

[BGLZ16] Bendkowski, M., Grygiel, K., Lescanne, P., & Zaionc, M. (2016).
A natural counting of lambda terms.
In International Conference on Current Trends in Theory and Practice of Informatics (pp. 183-194). Springer, Berlin, Heidelberg.

[BBD19] Bendkowski, M., Bodini, O., & Dovgal, S. (2019).
Statistical Properties of Lambda Terms.
The Electronic Journal of Combinatorics, P4-1.

[BCDH18] Bodini, O., Courtiel, J., Dovgal, S., & Hwang, H. K. (2018, June).
Asymptotic distribution of parameters in random maps.
In 29th International Conference on Probabilistic, Combinatorial and Asymptotic Methods for the Analysis of Algorithms (Vol. 110, pp. 13-1)

Bibliography

[B75] Bender, E. A. (1975).
An asymptotic expansion for the coefficients of some formal power series.
Journal of the London Mathematical Society, 2(3), 451-458.

[FS93] Flajolet, P., & Soria, M. (1993).
General combinatorial schemas: Gaussian limit distributions and exponential tails.
Discrete Mathematics, 114(1-3), 159-180.

[B18] Borinsky, M. (2018).
Generating Asymptotics for Factorially Divergent Sequences.
The Electronic Journal of Combinatorics, P4-1.

[BKW21] Banderier, C., Kuba, M., & Wallner, M. (2021).
Analytic Combinatorics of Composition schemes and phase transitions
mixed Poisson distributions.
arXiv preprint arXiv:2103.03751.

Bibliography

[BGJ13] Bodini, O., Gardy, D., & Jacquot, A. (2013).
Asymptotics and random sampling for BCI and BCK lambda terms
Theoretical Computer Science, 502, 227-238.

[M04] Mairson, H. G. (2004).
Linear lambda calculus and PTIME-completeness
Journal of Functional Programming, 14(6), 623-633.

[DGKRTZ13] Zaionc, M., Theyssier, G., Raffalli, C., Kozic, J.,
J., Grygiel, K., & David, R. (2013)
Asymptotically almost all λ-terms are strongly normalizing
Logical Methods in Computer Science, 9

[SAKT17] Sin'Ya, R., Asada, K., Kobayashi, N., & Tsukada, T. (2017)
Almost Every Simply Typed λ-Term Has a Long β-Reduction Sequence
In International Conference on Foundations of Software Science and
and Computation Structures (pp. 53-68). Springer, Berlin, Heidelberg.

Bibliography

[B19] Baptiste L. (2019).
A new family of bijections for planar maps
Journal of Combinatorial Theory, Series A.