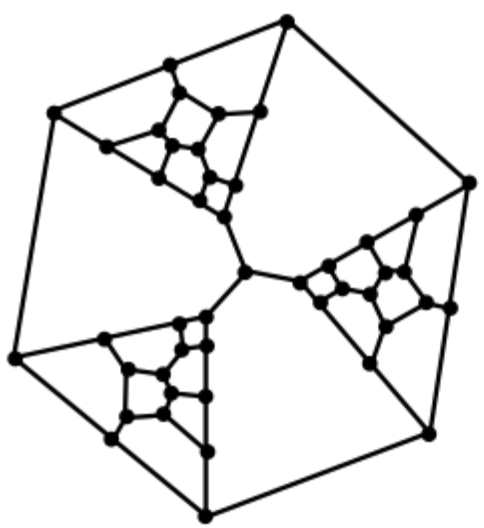
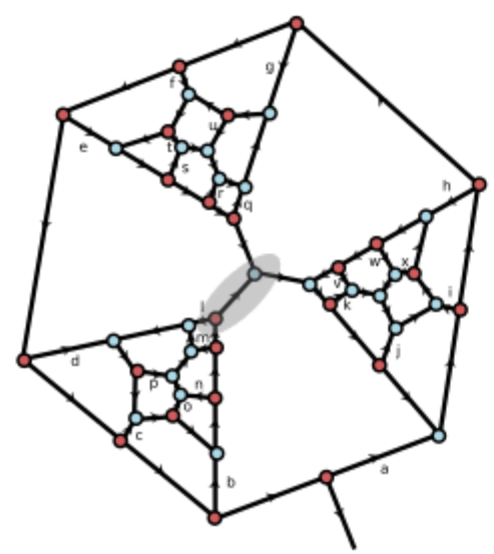
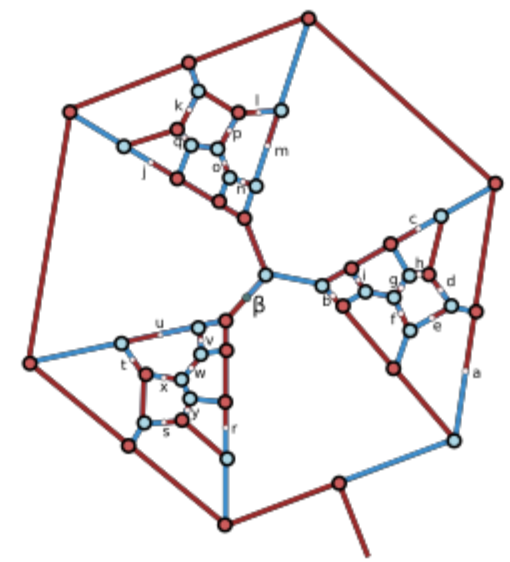
# *Untyped linear λ-calculus and the combinatorics of 3-valent ~~graphs~~ maps*



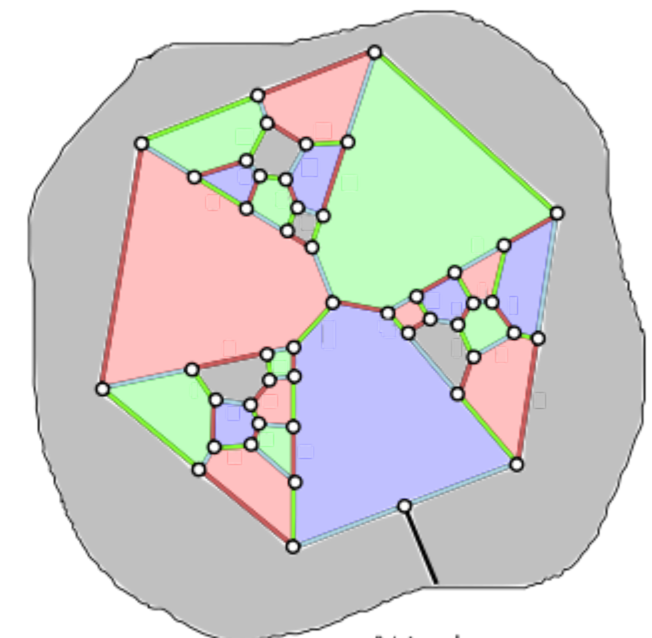
3-valent map



linear lambda term



principal typing



𝕍-typing

Noam Zeilberger (Ecole Polytechnique)

*Combinatorics & Arithmetic for Physics: special days*

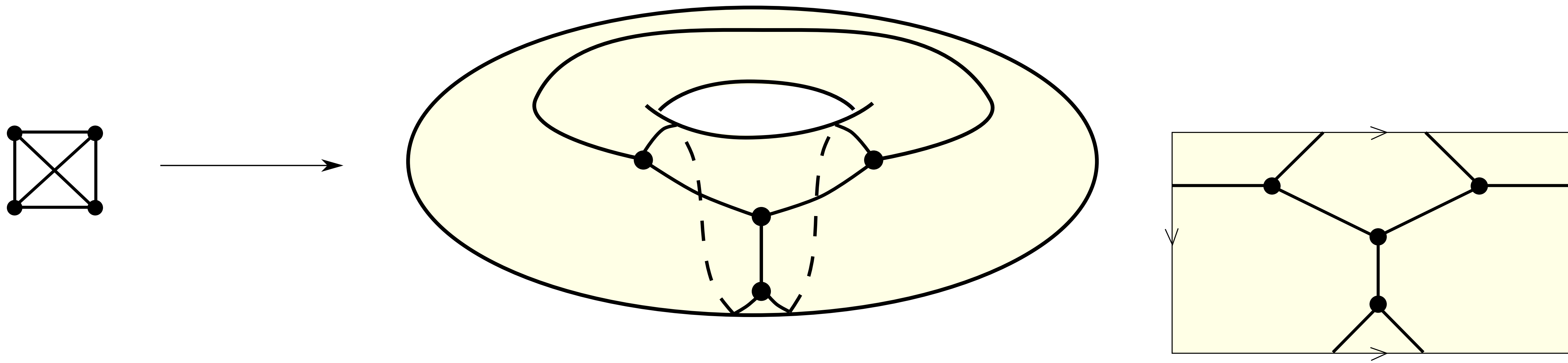
2-3 December 2020 @ IHES (virtually!)

# 1. What is a "map"?

## (And how many are there?)

# Topological definition

**map** = 2-cell embedding of a graph into a surface*



considered up to deformation of the underlying surface.

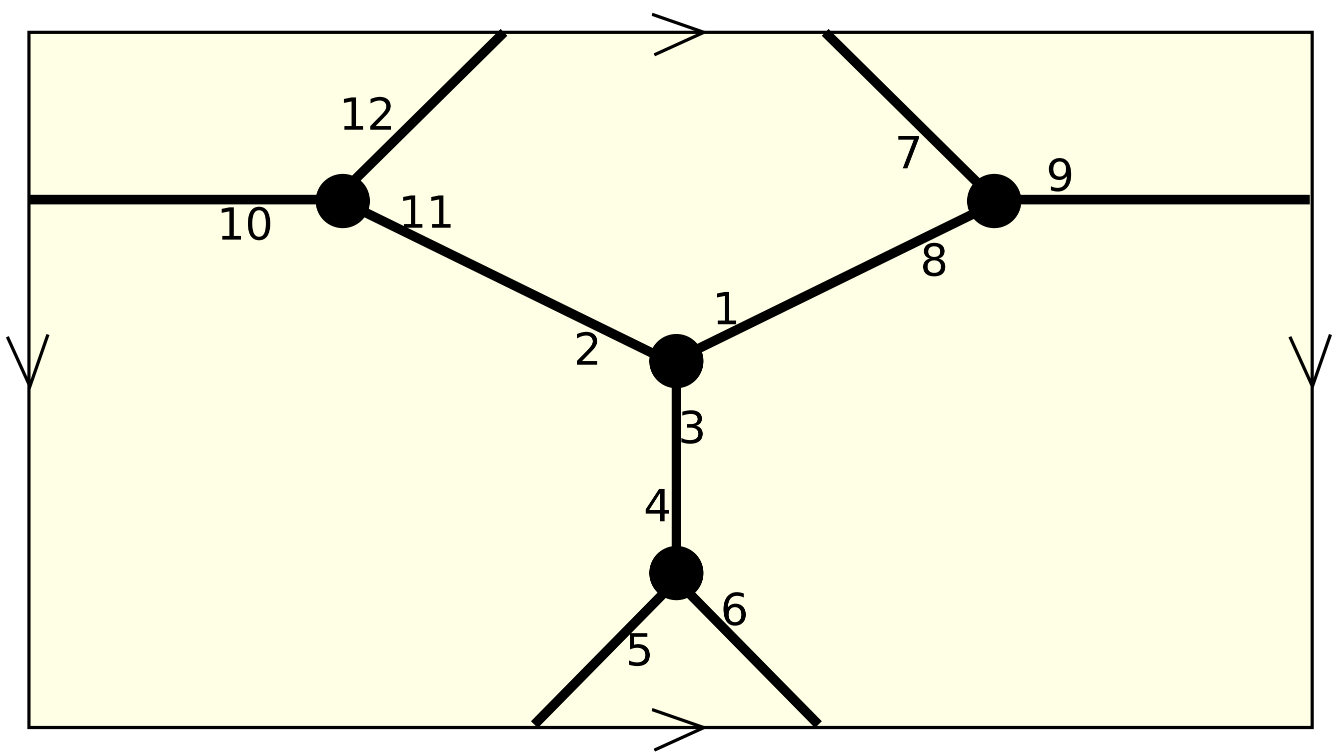
*All surfaces are assumed to be connected and oriented throughout this talk

# Algebraic definition

**map** = transitive permutation representation of the group

$$\text{G} = \langle v, e, f \mid e^2 = vef = 1 \rangle$$

considered up to G-equivariant isomorphism.




$$v = (1\ 2\ 3)(4\ 5\ 6)(7\ 8\ 9)(10\ 11\ 12)$$

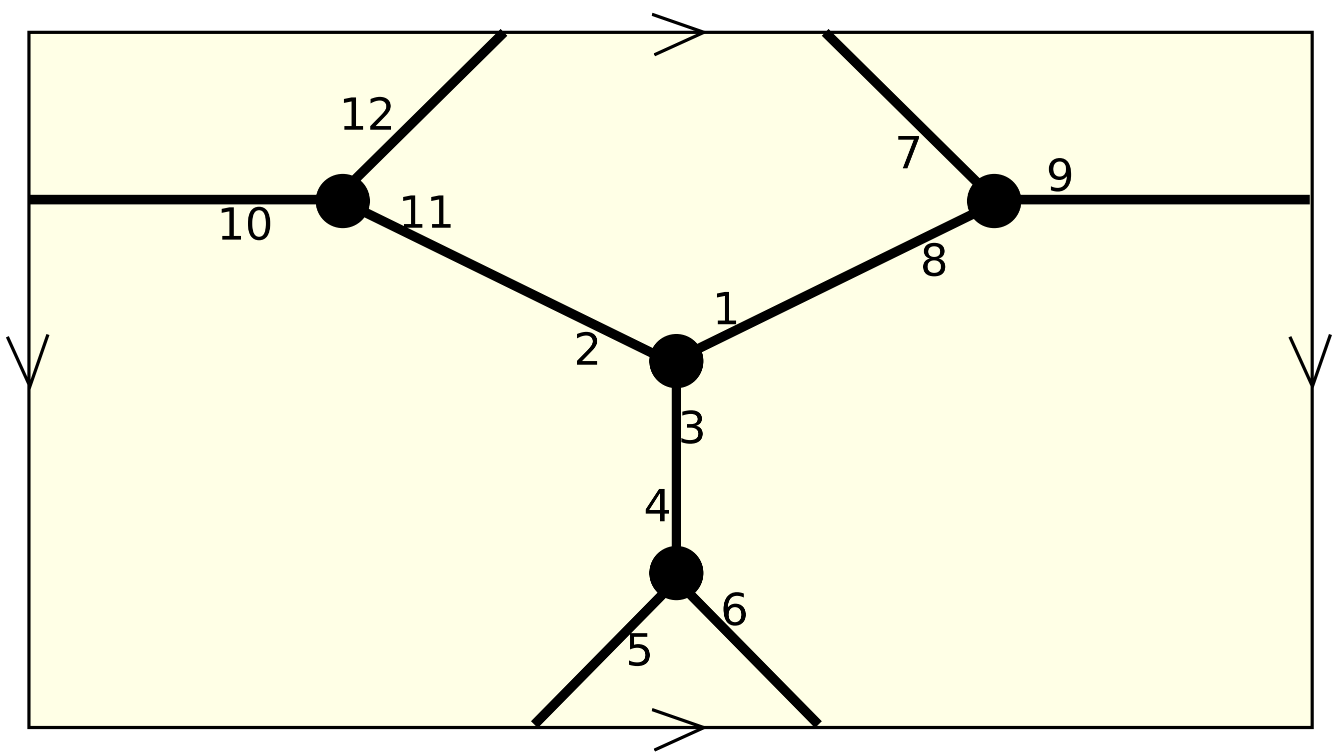
$$e = (1\ 8)(2\ 11)(3\ 4)(5\ 12)(6\ 7)(9\ 10)$$

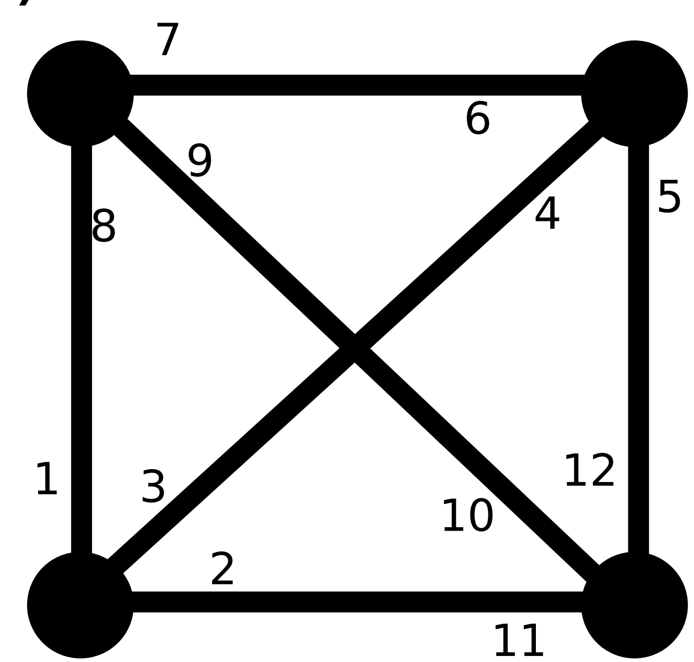
$$f = (1\ 7\ 5\ 11)(2\ 10\ 8\ 3\ 6\ 9\ 12\ 4)$$

$$c(v) - c(e) + c(f) = 2 - 2g$$

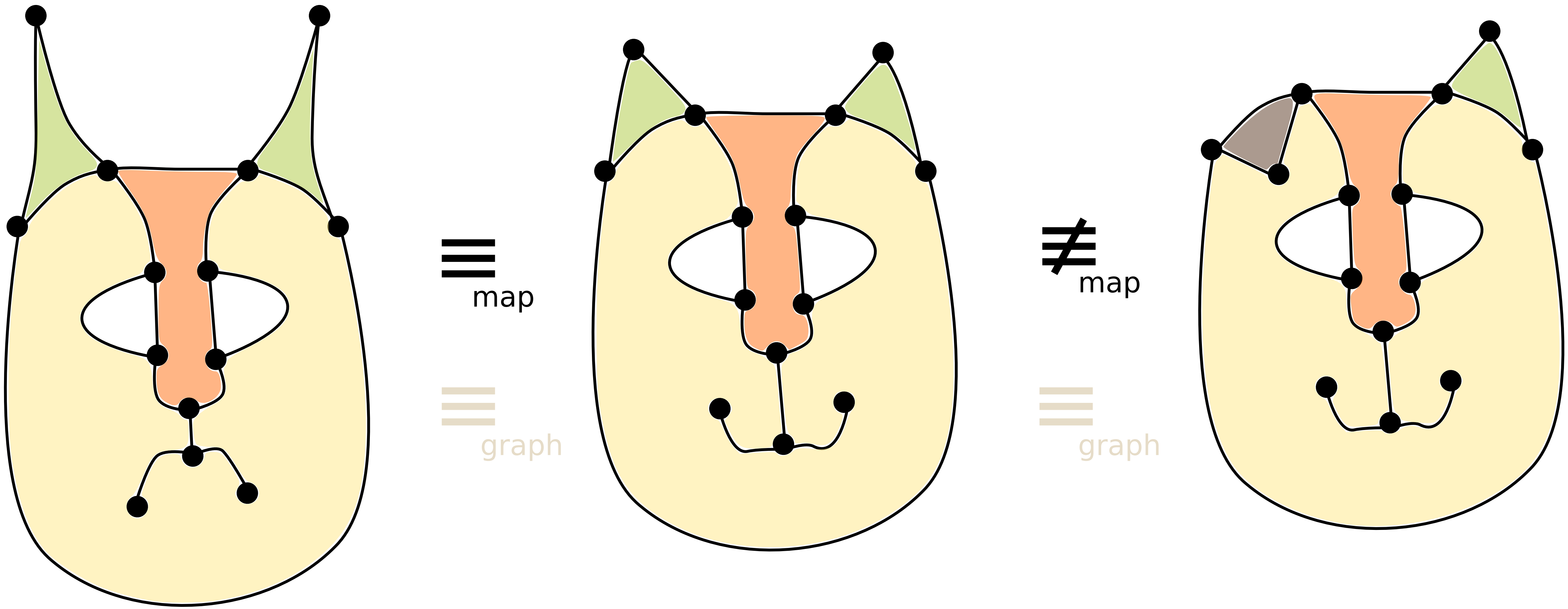
# Combinatorial definition

**map** = connected graph + cyclic ordering of the half-edges around each vertex (say, as given by a planar drawing with "virtual crossings").

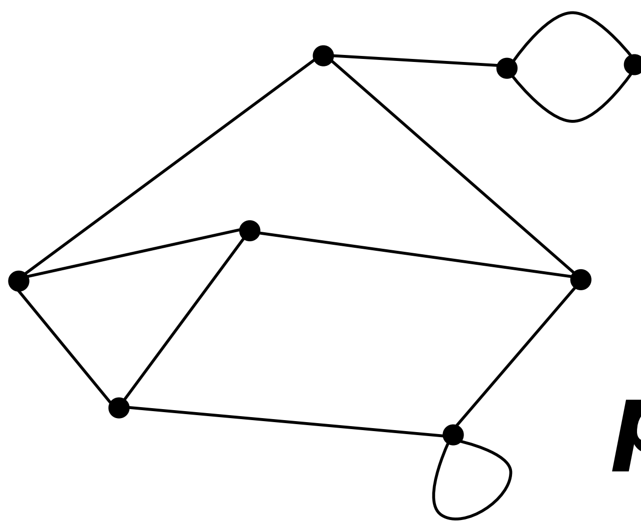


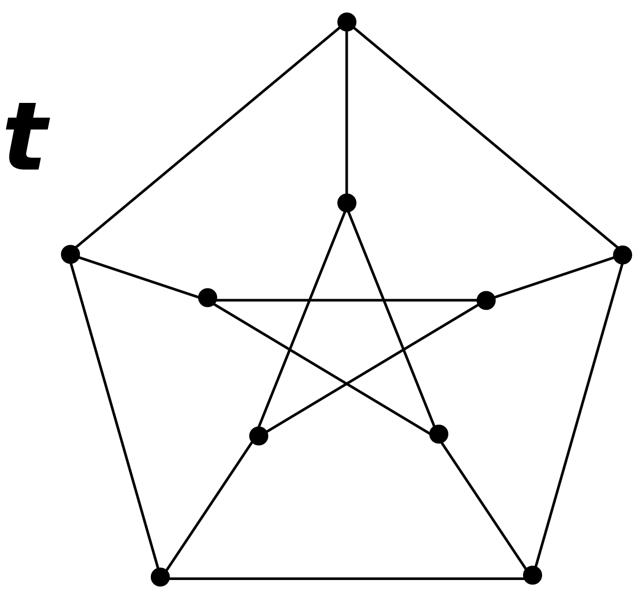
# Graph versus Map



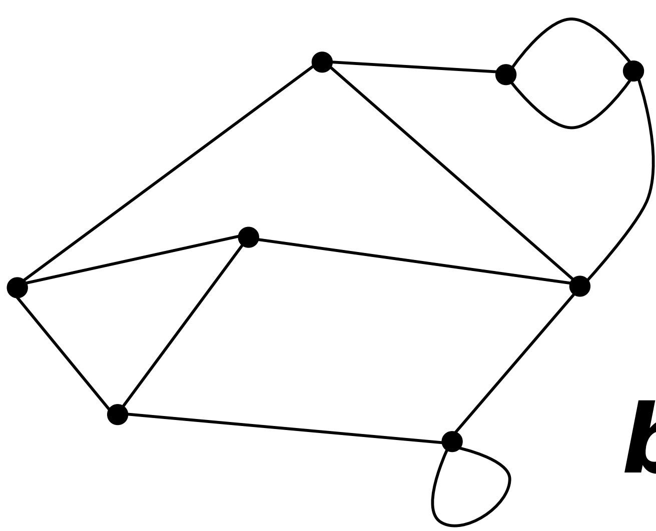
# Some special kinds of maps



***planar***



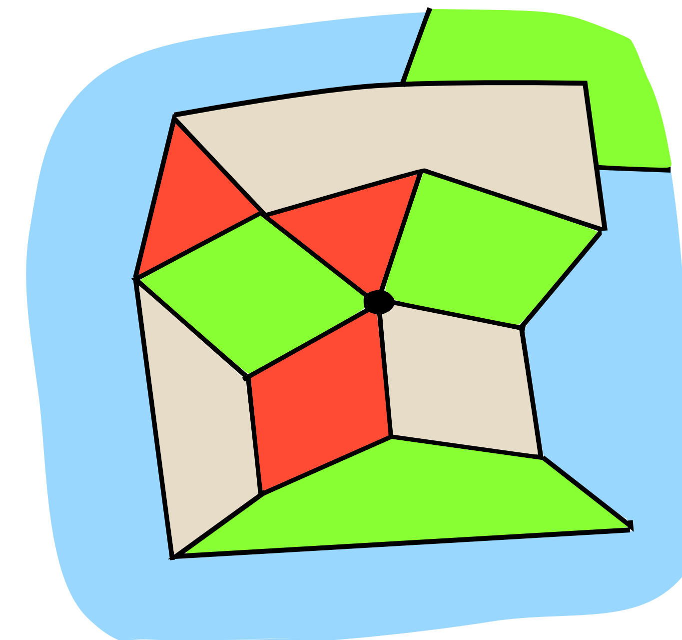
***3-valent***

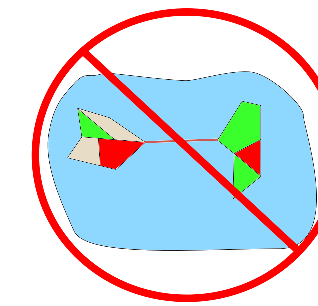


***bridgeless***

# Four Color Theorem

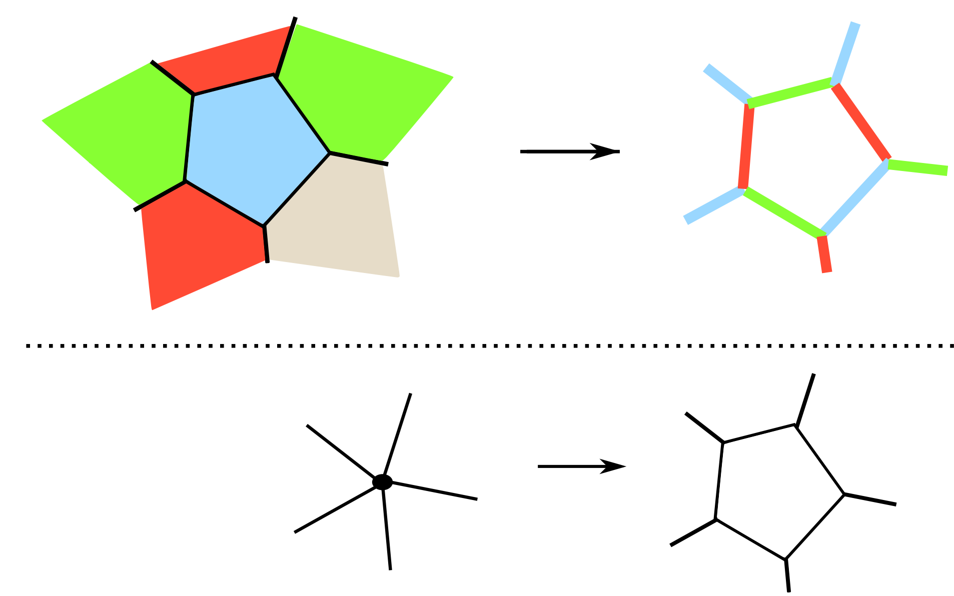
The 4CT is a statement about maps.





***every bridgeless planar map has a proper face 4-coloring***

By a well-known reduction (Tait 1880), 4CT is equivalent to a statement about 3-valent maps



***every bridgeless planar 3-valent map has a proper edge 3-coloring***

# Map enumeration

*From time to time in a graph-theoretical career one's thoughts turn to the Four Colour Problem. It occurred to me once that it might be possible to get results of interest in the theory of map-colourings without actually solving the Problem. For example, it might be possible to find the average number of colourings on vertices, for planar triangulations of a given size.*

*One would determine the number of triangulations of 2n faces, and then the number of 4-coloured triangulations of 2n faces. Then one would divide the second number by the first to get the required average. I gathered that this sort of retreat from a difficult problem to a related average was not unknown in other branches of Mathematics, and that it was particularly common in Number Theory.*

W. T. Tutte, Graph Theory as I Have Known It

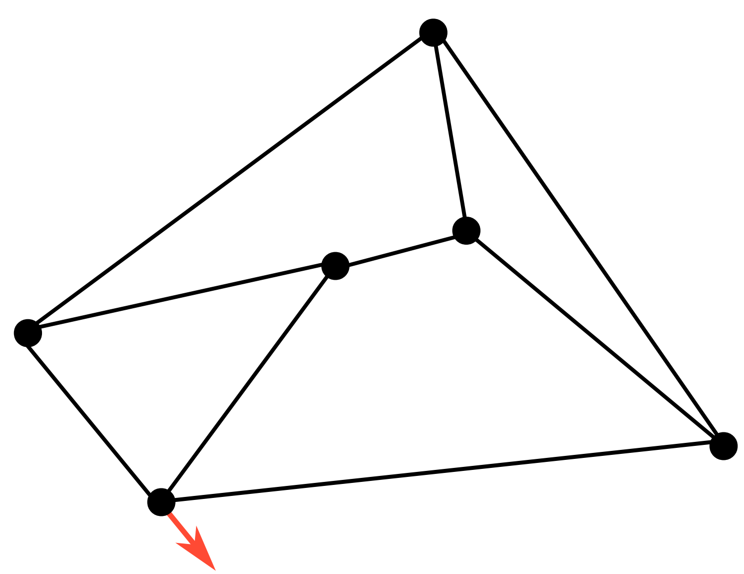
# Map enumeration

Tutte wrote a germinal series of papers (1962-1969)

bust by Gabriella Bollobás


W. T. Tutte (1962), A census of planar triangulations. Canadian Journal of Mathematics 14:21–38
W. T. Tutte (1962), A census of Hamiltonian polygons. Can. J. Math. 14:402–417
W. T. Tutte (1962), A census of slicings. Can. J. Math. 14:708–722
W. T. Tutte (1963), A census of planar maps. Can. J. Math. 15:249–271
W. T. Tutte (1968), On the enumeration of planar maps. Bulletin of the American Mathematical Society 74:64–74
W. T. Tutte (1969), On the enumeration of four-colored maps. SIAM Journal on Applied Mathematics 17:454–460


One of his insights was to consider ***rooted*** maps



*Key property: rooted maps have no non-trivial automorphisms*

# Map enumeration

Ultimately, Tutte obtained some remarkably simple formulas for counting different families of rooted planar maps, e.g.:

*The number $a_n$ of rooted maps with $n$ edges is*

$$\frac{2(2n)!\,3^n}{n!\,(n+2)!}.$$

For more on map-counting see:

Mireille Bousquet-Mélou, Enumerative Combinatorics of Maps (recorded lecture series)

Gilles Schaeffer, "Planar maps", in Handbook of Enumerative Combinatorics (ed. Bóna)

Bertrand Eynard, *Counting Surfaces*, Birkhäuser, 2016

# 2. A crash course in (linear) λ-calculus

# Lambda calculus: a very brief history*


Invented by Alonzo Church in late 20s, published in 1932

Original goal: foundation for logic without free variables

Minor defect: *inconsistent!*

Resolution: separate into an **untyped calculus** for computation, and a **typed calculus** for logic.

(Both have since found many uses.)

*Source: Cardone & Hindley's "History of Lambda-calculus and Combinatory Logic"

# Fixpoints and non-linearity


Turing published first *fixed-point combinator* (1937)

(key to Turing-completeness of λ-calculus)

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

Observe *doubled uses* of variables x and y.

By restricting to terms where every variable is used exactly once, one gets a well-behaved **linear** subsystem of lambda calculus.

(no longer Turing-complete...actually P-complete)

# Untyped linear lambda terms (defn.)

basic judgment $x_1, \ldots, x_n \vdash t$

*t is a linear term with free variables $x_1, \ldots, x_n$*

inductive definition

$$\frac{}{x \vdash x}\ \textit{var} \qquad \frac{\Gamma \vdash t \qquad \Delta \vdash u}{\Gamma, \Delta \vdash t(u)}\ \textit{app} \qquad \frac{\Gamma, x \vdash t}{\Gamma \vdash \lambda x.t}\ \textit{abs}$$

$$\frac{\Gamma, x, y, \Delta \vdash t}{\Gamma, y, x, \Delta \vdash t}\ \textit{exc}$$

define: subterms, bound variables, α-equivalence, closed subterms, ordered terms

# Untyped linear lambda terms (ex.)

| | |
|---|---|
| $\vdash \lambda x.\lambda y.\lambda z.x(yz)$ | ordered term (**B**) |
| $\vdash \lambda x.\lambda y.\lambda z.(xz)y$ | non-ordered term (**C**) |
| $x \vdash \lambda y.\lambda z.x(yz)$ | open term |
| $x \vdash x(\lambda y.y)$ | term w/closed subterm |

# Term rewriting

Computation through ***the rule of β-reduction:***

$$(\lambda x.t)(u) \;\to^{\beta}\; t[u/x]$$

*can apply to any matching subterm*
*(confluent and strongly normalizing)*

Sometimes paired with ***the rule of η-expansion:***

$$t \to^{\eta} \lambda x.t(x)$$

Example:

$$\begin{aligned} &(\lambda x.\lambda y.\lambda z.x(yz))(\lambda a.a)(t) \\ \to^{\beta}\; &(\lambda y.\lambda z.(\lambda a.a)(yz))(t) \\ \to^{\beta}\; &(\lambda y.\lambda z.yz)(t) \\ \to^{\beta}\; &\lambda z.t(z) \;{}^{\eta}\!\!\leftarrow t \end{aligned}$$

# Typing

types $A,B ::= X,Y,\ldots \mid A \multimap B$

basic judgment $x_1{:}A_1 , \ldots , x_n{:}A_n \vdash t{:}B$

*t is a proof that* $A_1 , \ldots , A_n$ *(linearly) entail B*

inductive definition

$$\frac{}{x{:}A \vdash x{:}A} \qquad \frac{\Gamma \vdash t{:}A\multimap B \quad \Delta \vdash u{:}A}{\Gamma , \Delta \vdash t(u){:}B} \qquad \frac{\Gamma , x{:}A \vdash t{:}B}{\Gamma \vdash \lambda x.t{:}A\multimap B}$$

$$\frac{\Gamma , x{:}A, y{:}B , \Delta \vdash t{:}C}{\Gamma , y{:}B, x{:}A , \Delta \vdash t{:}C}$$

typed linear terms modulo βη present the free sym. closed multicategory!

## 3. how on earth are these topics related??

# An innocent idea

In May 2014, I thought it could be fun* to count untyped *closed β-normal ordered linear terms* by size (#λs)...

*for reasons related to certain categorical models of typing, cf. Melliès & Zeilberger POPL 2015

$$\lambda x.x$$

1

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

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

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

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

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

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

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

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

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

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

9

λx.x(λy.y(λz.z(λw.w)))
λx.x(λy.y(λz.λw.z(w)))
λx.x(λy.y(λz.z)(λw.w))
λx.x(λy.λz.y(z(λw.w)))
λx.x(λy.λz.y(λw.z(w)))
λx.x(λy.λz.y(λw.w)(z))
λx.x(λy.λz.y(z)(λw.w))
λx.x(λy.λz.λw.y(z(w)))
λx.x(λy.λz.λw.y(z)(w))

λx.x(λy.y)(λz.z(λw.w))
λx.x(λy.y)(λz.λw.z(w))
λx.x(λy.y(λz.z))(λw.w)
λx.x(λy.λz.y(z))(λw.w)
λx.x(λy.y)(λz.z)(λw.w)
λx.λy.x(y(λz.z(λw.w)))
λx.λy.x(y(λz.λw.z(w)))
λx.λy.x(y(λz.z)(λw.w))
λx.λy.x(λz.y(z(λw.w)))

λx.λy.x(λz.y(λw.z(w)))
λx.λy.x(λz.y(λw.w)(z))
λx.λy.x(λz.y(z)(λw.w))
λx.λy.x(λz.λw.y(z(w)))
λx.λy.x(λz.λw.y(z)(w))
λx.λy.x(λz.z)(y(λw.w))
λx.λy.x(λz.z)(λw.y(w))
λx.λy.x(λz.z(λw.w))(y)
λx.λy.x(λz.λw.z(w))(y)

λx.λy.x(λz.z)(λw.w)(y)
λx.λy.x(y)(λz.z(λw.w))
λx.λy.x(y)(λz.λw.z(w))
λx.λy.x(y(λz.z))(λw.w)
λx.λy.x(λz.y(z))(λw.w)
λx.λy.x(λz.z)(y)(λw.w)
λx.λy.x(y)(λz.z)(λw.w)
λx.λy.λz.x(y(z(λw.w)))
λx.λy.λz.x(y(λw.z(w)))

λx.λy.λz.x(y(λw.w)(z))
λx.λy.λz.x(y(z)(λw.w))
λx.λy.λz.x(λw.y(z(w)))
λx.λy.λz.x(λw.y(z)(w))
λx.λy.λz.x(λw.w)(y(z))
λx.λy.λz.x(y)(z(λw.w))
λx.λy.λz.x(y)(λw.z(w))
λx.λy.λz.x(y(λw.w))(z)
λx.λy.λz.x(λw.y(w))(z)

λx.λy.λz.x(λw.w)(y)(z)
λx.λy.λz.x(y)(λw.w)(z)
λx.λy.λz.x(y(z))(λw.w)
λx.λy.λz.x(y)(z)(λw.w)
λx.λy.λz.λw.x(y(z(w)))
λx.λy.λz.λw.x(y(z)(w))
λx.λy.λz.λw.x(y)(z(w))
λx.λy.λz.λw.x(y(z))(w)
λx.λy.λz.λw.x(y)(z)(w)

# 54

# THE ON-LINE ENCYCLOPEDIA OF INTEGER SEQUENCES®

founded in 1964 by N. J. A. Sloane

1,2,9,54,378,2916,24057 Search Hints

(Greetings from The On-Line Encyclopedia of Integer Sequences!)

Search: **seq:1,2,9,54,378,2916,24057**

Displaying 1-1 of 1 result found. page 1

Sort: relevance | references | number | modified | created Format: long | short | data

A000168 2*3^n*(2*n)!/(n!*(n+2)!). +20 18

(Formerly M1940 N0768)

**1, 2, 9, 54, 378, 2916, 24057**, 208494, 1876446, 17399772, 165297834, 1602117468, 15792300756, 157923007560, 1598970451545, 16365932856990, 169114639522230, 1762352559231660, 18504701871932430, 195621134074714260, 2080697516976506220, 22254416920705240440, 239234981897581334730, 2583737804493878415084 (list; graph; refs; listen; history; text; internal format)

OFFSET 0,2

COMMENTS Number of rooted planar maps with n edges. - Don Knuth, Nov 24 2013

Number of rooted 4-regular planar maps with n vertices.

Also, number of doodles with n crossings, irrespective of the number of loops.

# THE ON-LINE ENCYCLOPEDIA OF INTEGER SEQUENCES®

founded in 1964 by N. J. A. Sloane

1,2,9,54,378,2916,24057 Search Hints

(Greetings from The On-Line Encyclopedia of Integer Sequences!)

Search: **seq:1,2,9,54,378,2916,24057**

Displaying 1-1 of 1 result found. page 1

Sort: relevance | references | number | modified | created Format: long | short | data

A000168 2*3^n*(2*n)!/(n!*(n+2)!). +20 18

(Formerly M1940 N0768)

**1, 2, 9, 54, 378, 2916**

15792300756, 15792300756

176235259231660, 185047

222544169207052440, 23

text; internal format)

OFFSET 0,2

COMMENTS Number of rooted planar maps with n edges. - Don Knuth

Number of rooted 4-regular planar maps with n vertices.

Also, number of doodles with n crossings, irrespective of the number of loops.

*The number $a_n$ of rooted maps with $n$ edges is*

$$\frac{2(2n)!\,3^n}{n!\,(n+2)!}.$$

# One piece of a larger puzzle

| family of rooted maps | family of lambda terms | sequence | OEIS |
|---|---|---|---|
| planar maps | normal ordered terms | 1,2,9,54,378,2916,... | A000168 |

Z, A. Giorgetti (2015), A correspondence between rooted planar maps and normal planar lambda terms, LMCS 11(3:22): 1-393.

# One piece of a larger puzzle

| family of rooted maps | family of lambda terms | sequence | OEIS |
|---|---|---|---|
| trivalent maps (genus g≥0) | linear terms | 1,5,60,1105,27120,... | A062980 |
| planar maps | normal ordered terms | 1,2,9,54,378,2916,... | A000168 |

O. Bodini, D. Gardy, A. Jacquot (2013), Asymptotics and random sampling for BCI and BCK lambda terms, TCS 502: 227-238.
Z, A. Giorgetti (2015), A correspondence between rooted planar maps and normal planar lambda terms, LMCS 11(3:22): 1-393.

# One piece of a larger puzzle

| family of rooted maps | family of lambda terms | sequence | OEIS |
|---|---|---|---|
| trivalent maps (genus g≥0) | linear terms | 1,5,60,1105,27120,... | A062980 |
| planar trivalent maps | ordered terms | 1,4,32,336,4096,... | A002005 |
| bridgeless trivalent maps | unitless linear terms | 1,2,20,352,8624,... | A267827 |
| bridgeless planar trivalent maps | unitless ordered terms | 1,1,4,24,176,1456,... | A000309 |
| maps (genus g≥0) | normal linear terms (mod ~) | 1,2,10,74,706,8162,... | A000698 |
| planar maps | normal ordered terms | 1,2,9,54,378,2916,... | A000168 |
| bridgeless maps | normal unitless linear terms (mod ~) | 1,1,4,27,248,2830,... | A000699 |
| bridgeless planar maps | normal unitless ordered terms | 1,1,3,13,68,399,... | A000260 |


O. Bodini, D. Gardy, A. Jacquot (2013), Asymptotics and random sampling for BCI and BCK lambda terms, TCS 502: 227-238.

Z, A. Giorgetti (2015), A correspondence between rooted planar maps and normal planar lambda terms, LMCS 11(3:22): 1-393.

Z (2015), Counting isomorphism classes of beta-normal linear lambda terms, arXiv:1509.075964.

Z (2016), Linear lambda terms as invariants of rooted trivalent maps, J. Functional Programming 26(e21).

J. Courtiel, K. Yeats, Z (2017), Connected chord diagrams and bridgeless maps, Electronic Journal of Combinatorics, 26:4.

Z (2017), A sequent calculus for a semi-associative law, FSCD 2017.

Z (2018), A theory of linear typings as flows on 3-valent graphs, LICS 2018.

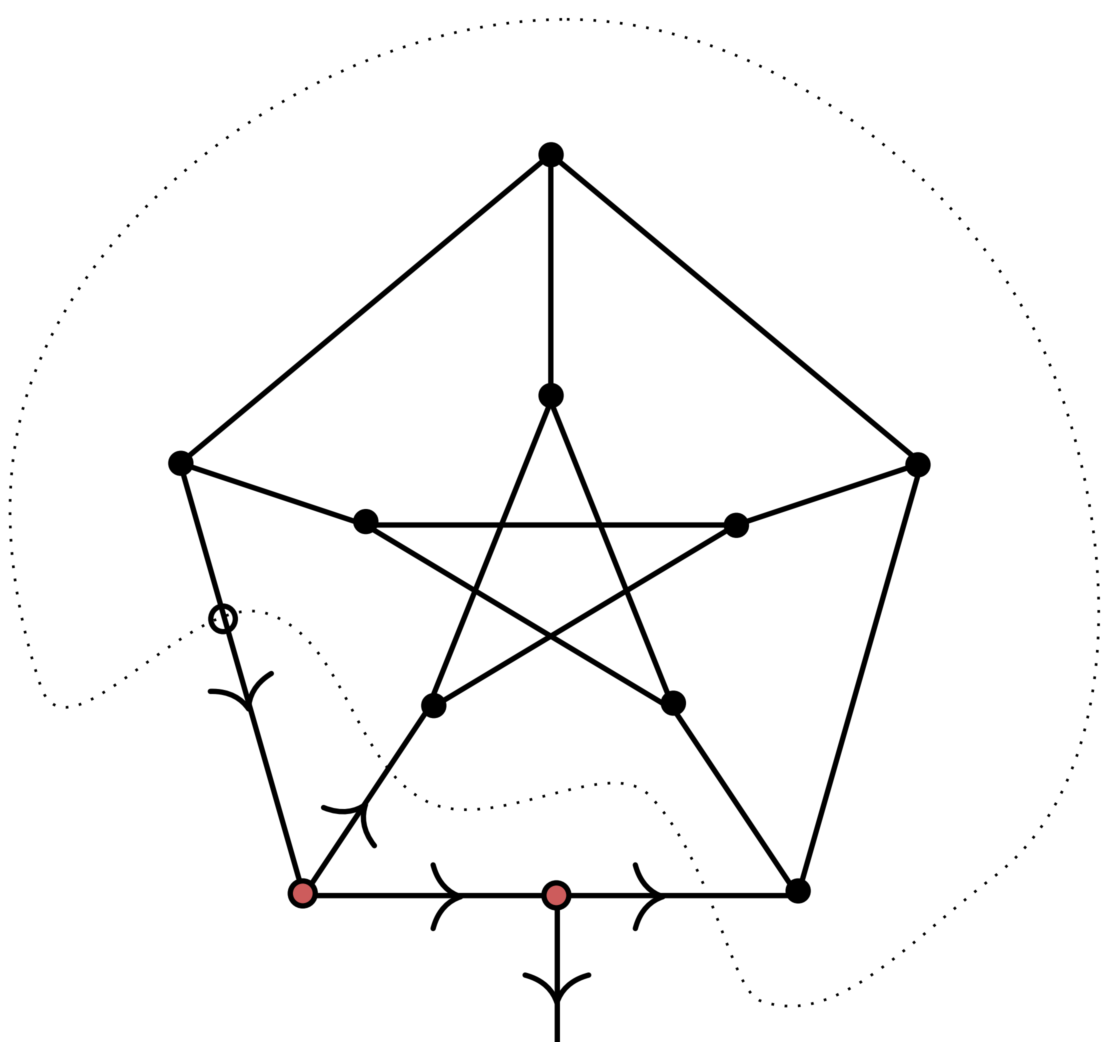

interdisciplinary between combinatorics, physics, and λ-calculus!

(unitless = no closed subterms)

(λx.λy.t ~ λy.λx.t)

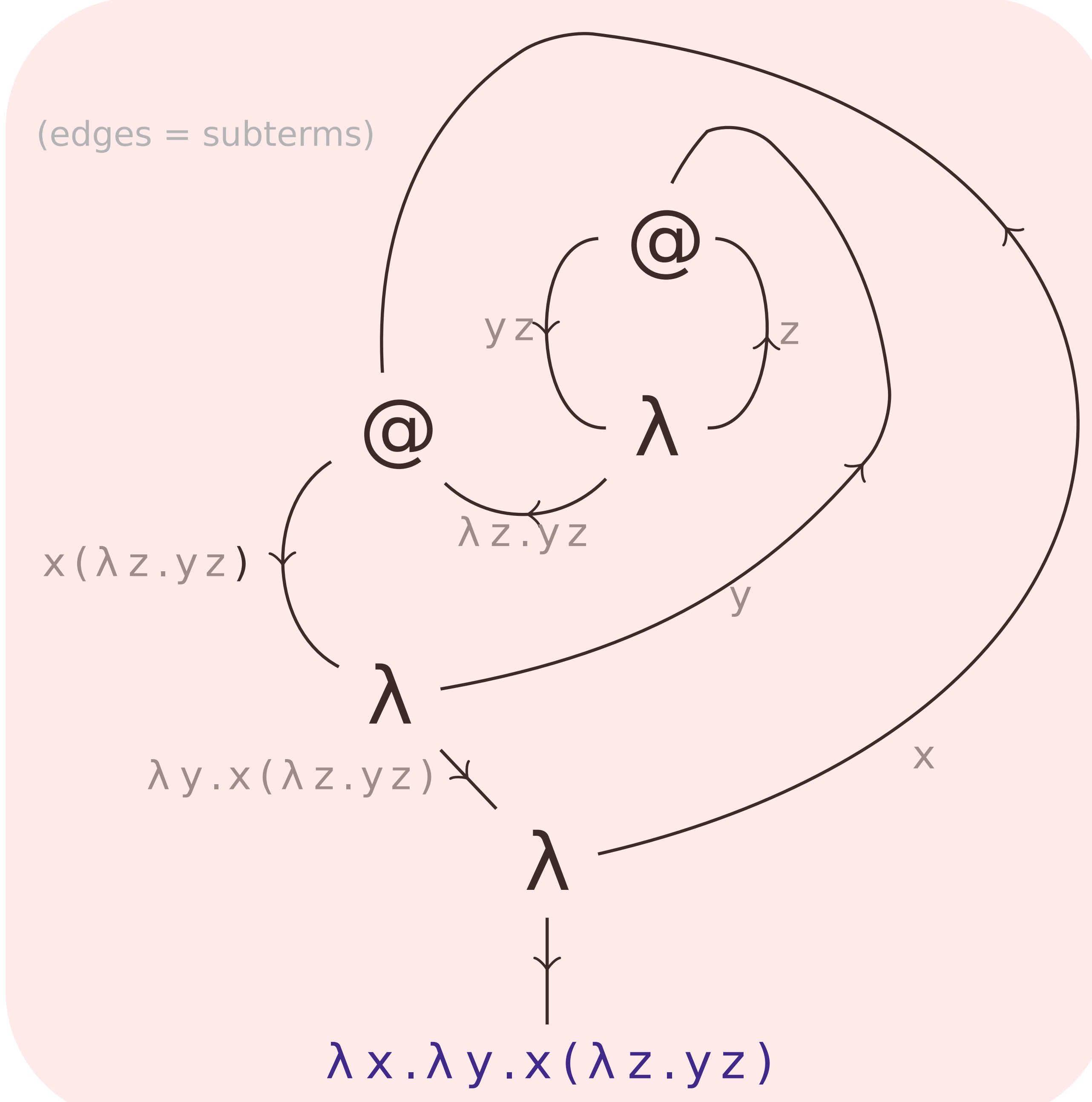
# 4. Between linear λ-terms and rooted 3-valent maps

(a bijection by Bodini et al 2013, as analyzed by Z 2016)



# Idea (folklore*): representing λ-terms as graphs

Can represent a term as tree w/two kinds of nodes (@/λ), with "pointers" from λ-nodes to bound variables. This idea is especially natural for linear terms.



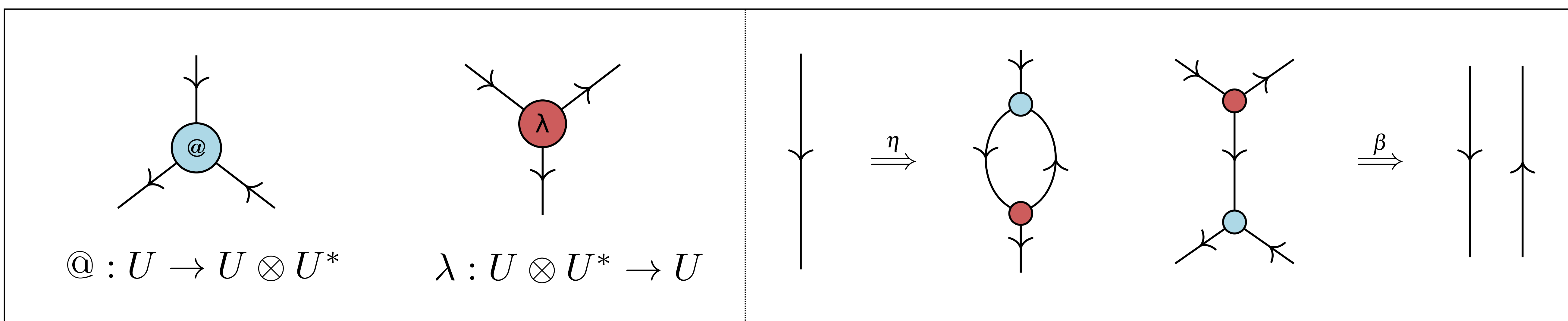

*The idea itself is natural and should probably be called folklore. The earliest explicit description I know of (currently) is in Knuth's "Examples of Formal Semantics" (1970), but it was developed more deeply and independently from different perspectives in the PhD theses of C. P. Wadsworth (1971) and R. Statman (1974).

# λ-graphs as string diagrams

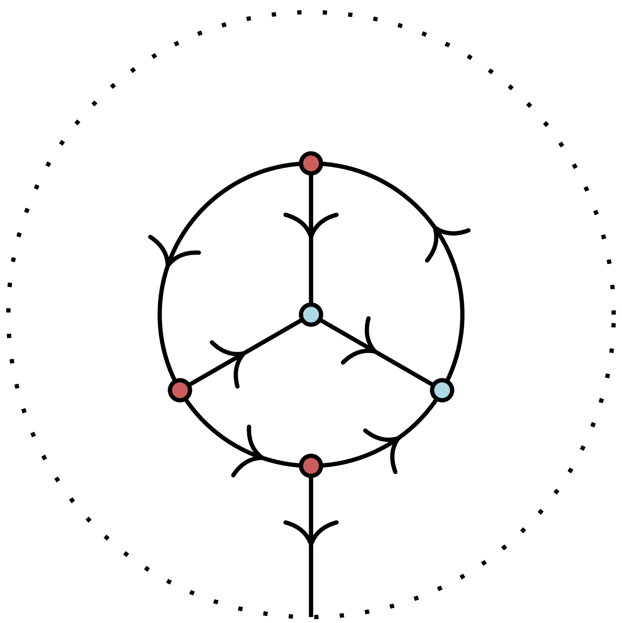
This idea can also be understood within the categorical framework of "string diagrams", by interpreting λ-terms (after D. Scott) as *endomorphisms of a reflexive object*

$$U \underset{\lambda}{\overset{@}{\rightleftarrows}} U \multimap U \quad (\lambda \dashv @)$$

in a symmetric monoidal closed bicategory.

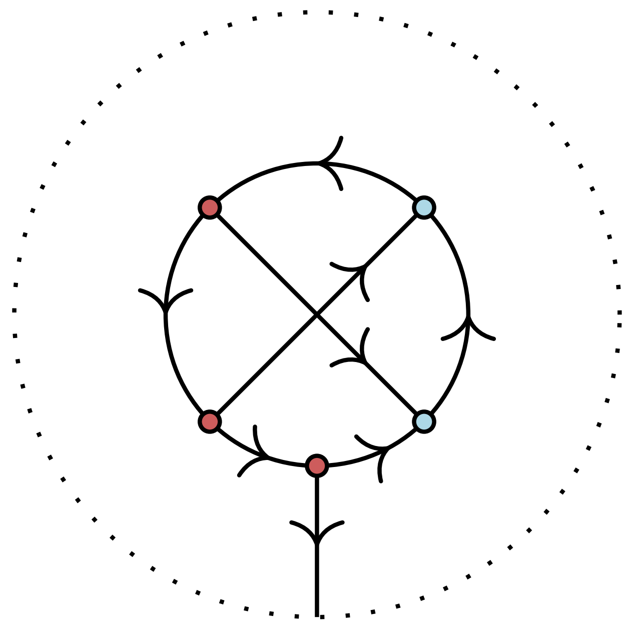


# From linear **λ**-terms to rooted 3-valent maps



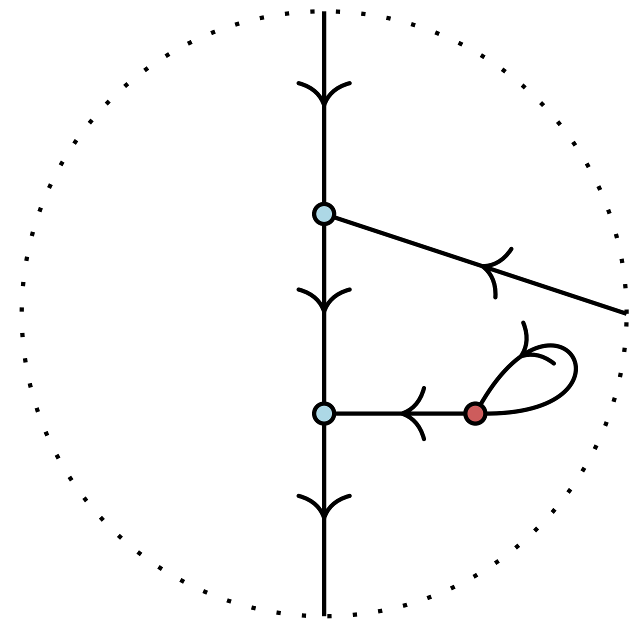
λx.λy.λz.x(yz)

(**B**)

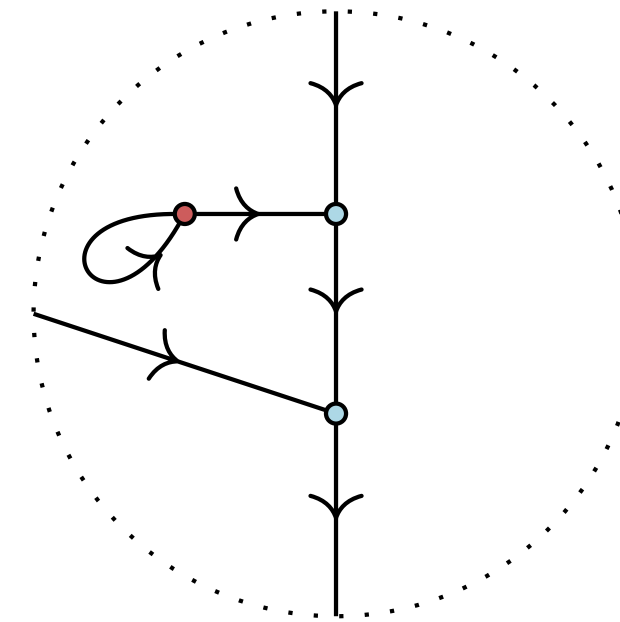


λx.λy.λz.(xz)y

(**C**)

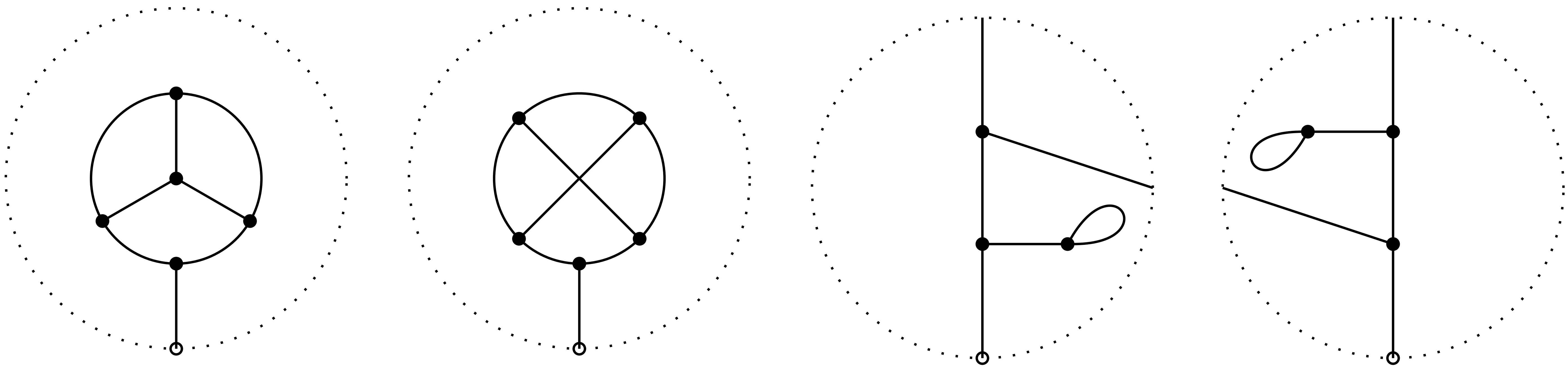


x,y ⊢ (xy)(λz.z)



x,y ⊢ x((λz.z)y)

# From linear **λ**-terms to rooted 3-valent maps



λx.λy.λz.x(yz) — (**B**)

λx.λy.λz.(xz)y — (**C**)

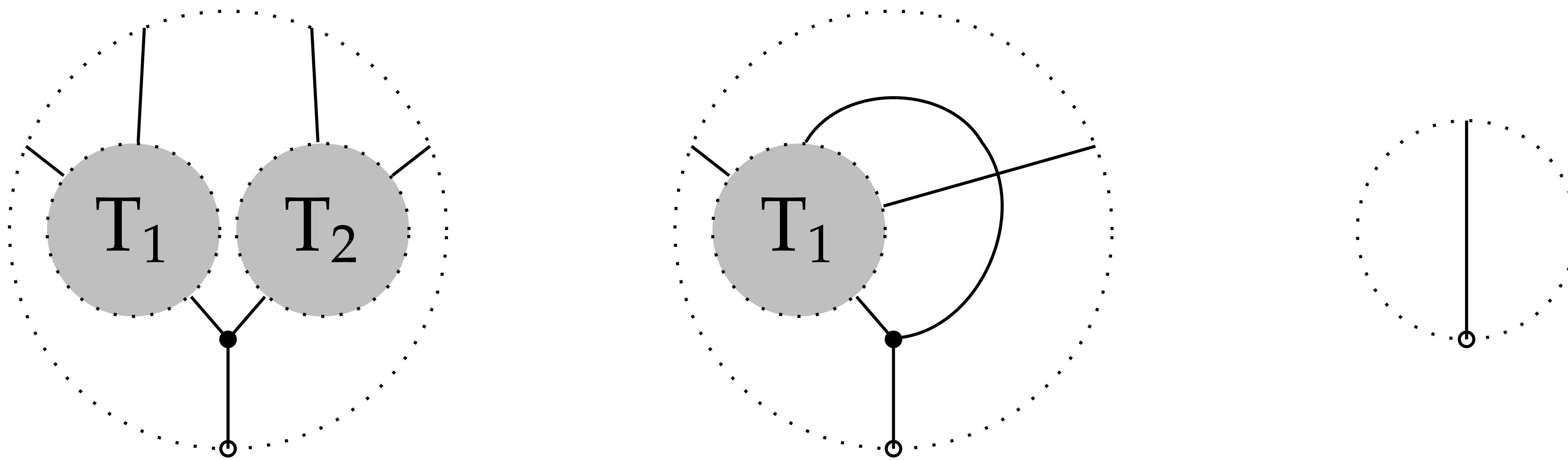
x,y ⊢ (xy)(λz.z)

x,y ⊢ x((λz.z)y)

# From rooted 3-valent maps to linear λ-terms

Step #1: generalize to 3-valent maps w/∂ of "free" edges, one marked as root.

Step #2: observe any such map must have one of the following forms:

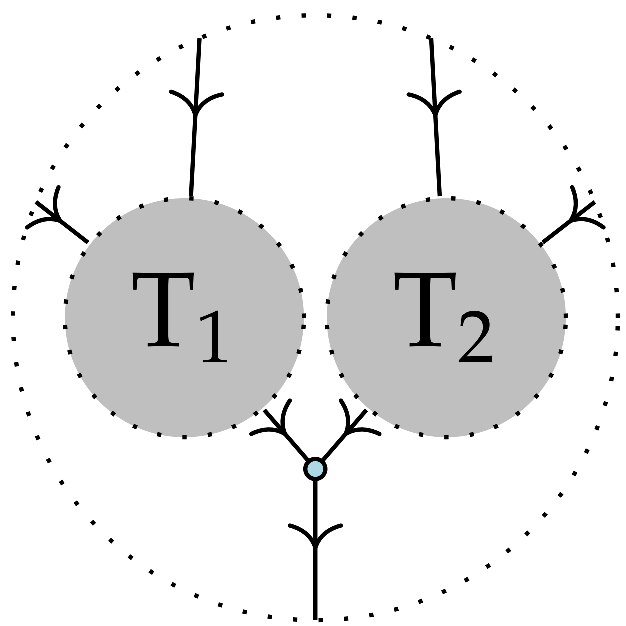



| disconnecting root vertex | connecting root vertex | no root vertex |
|---|---|---|

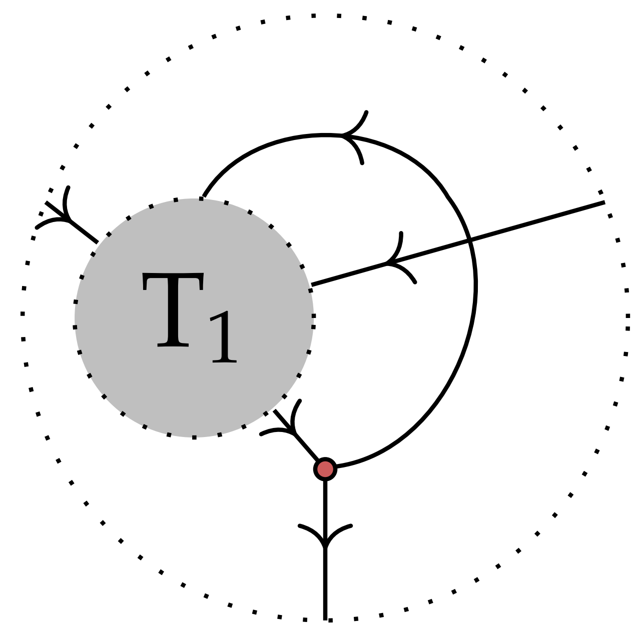
# From rooted 3-valent maps to linear λ-terms

⋮

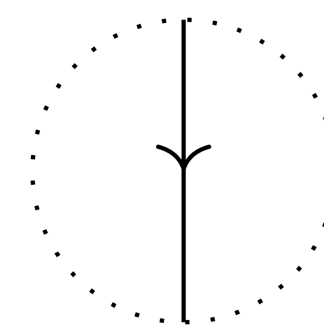
Step #3: observe this is exactly the inductive definition of linear λ-terms!




application

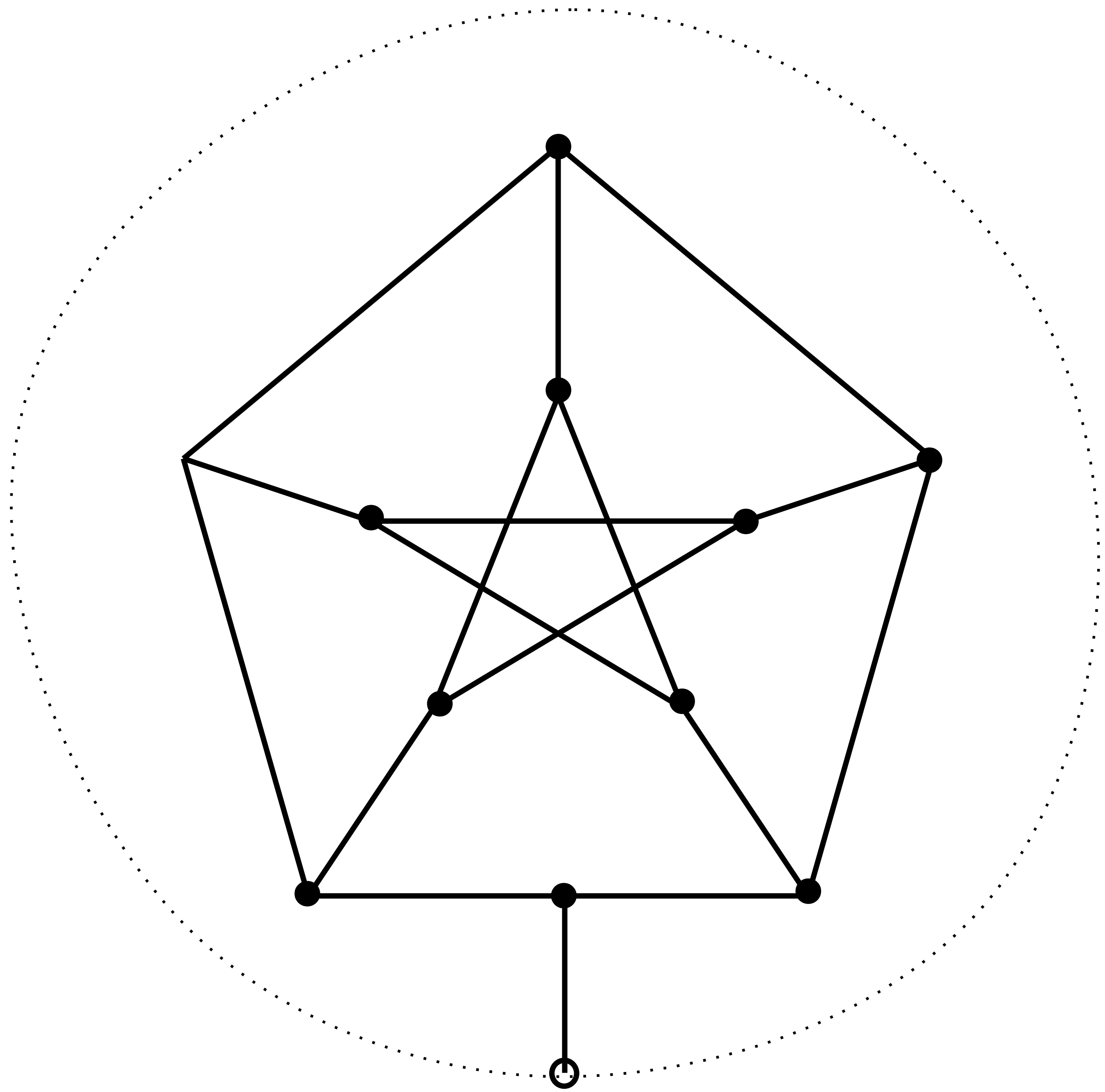



abstraction

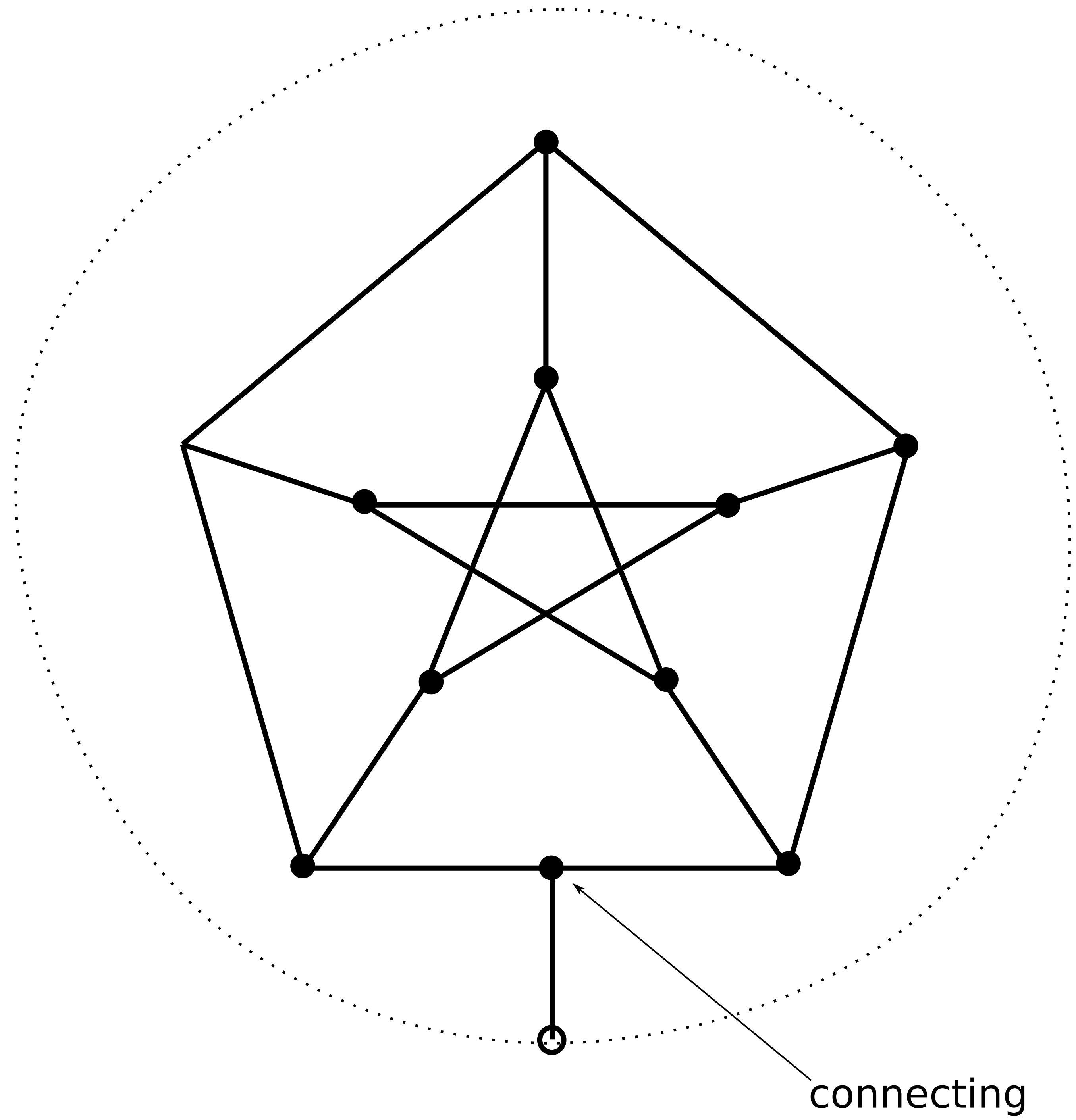


variable

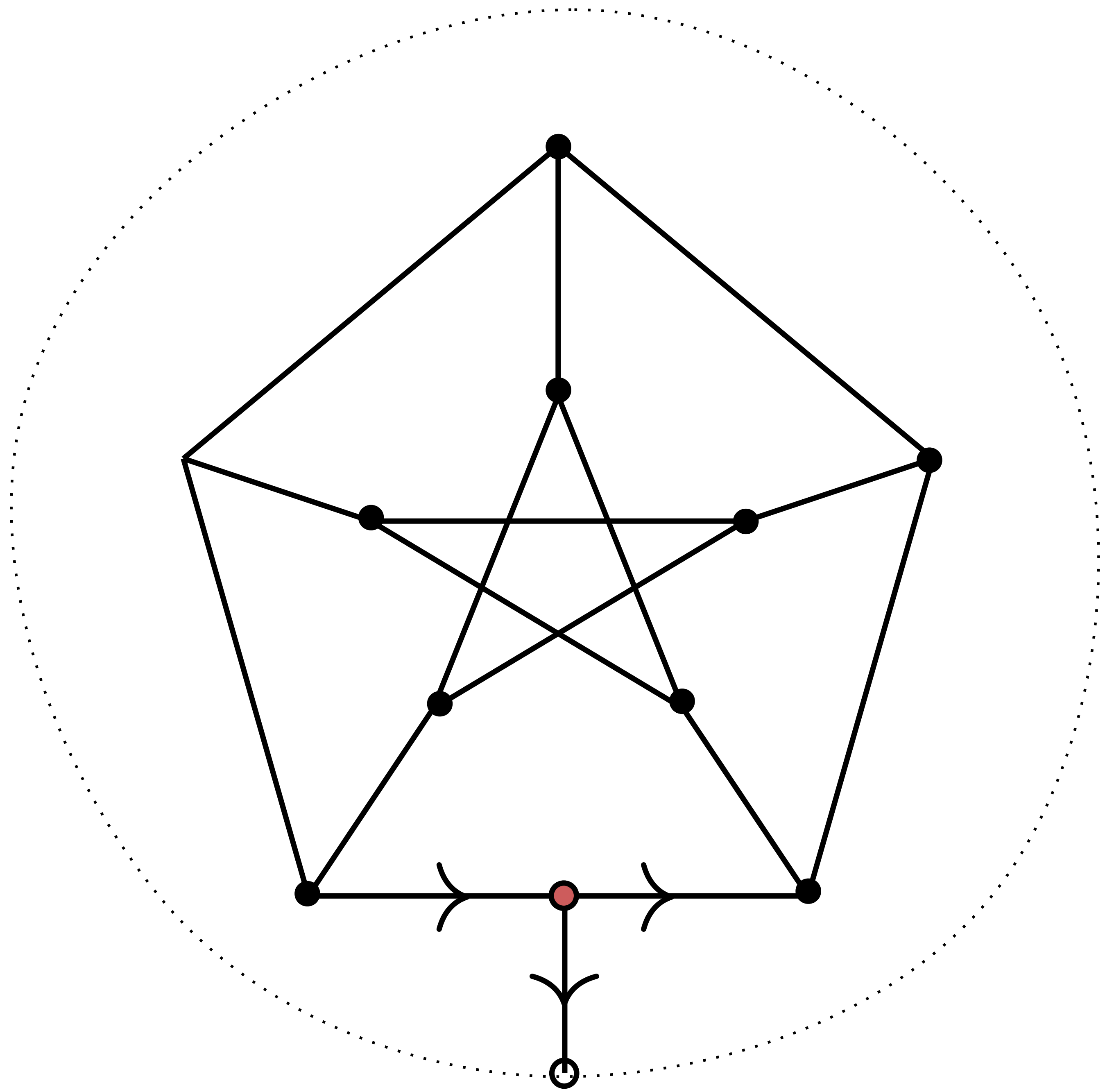
# An example



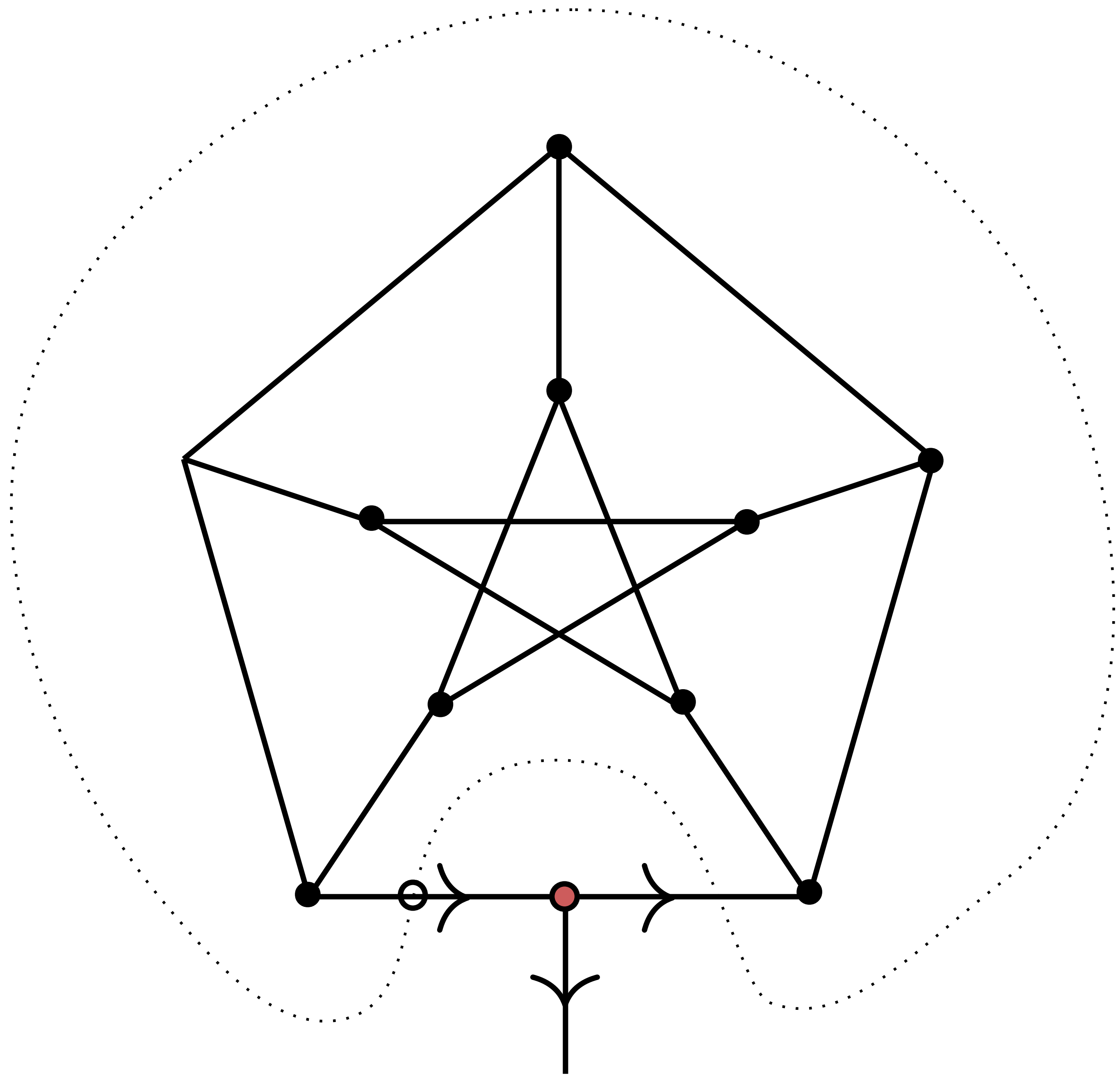
# An example



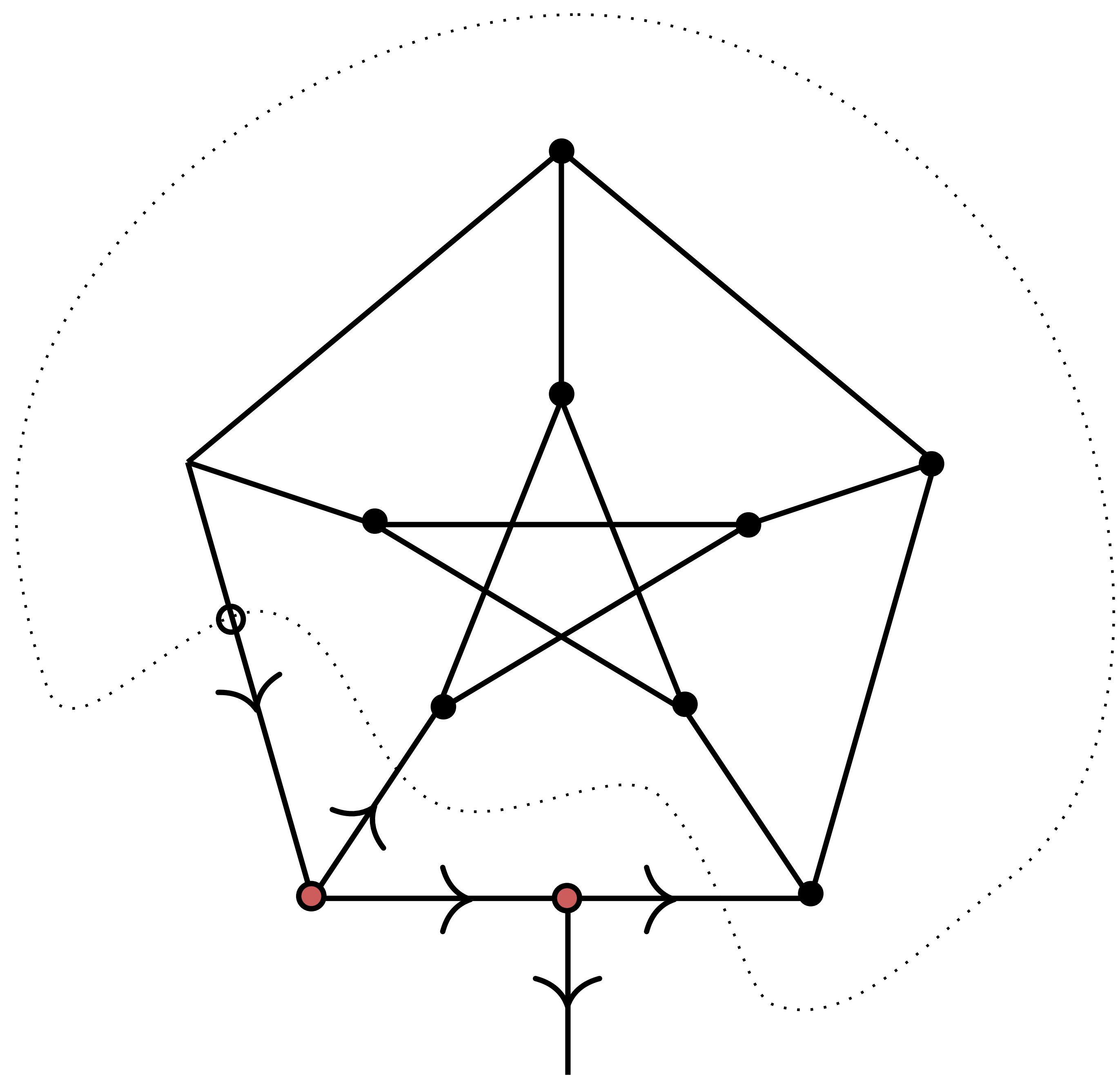
# An example



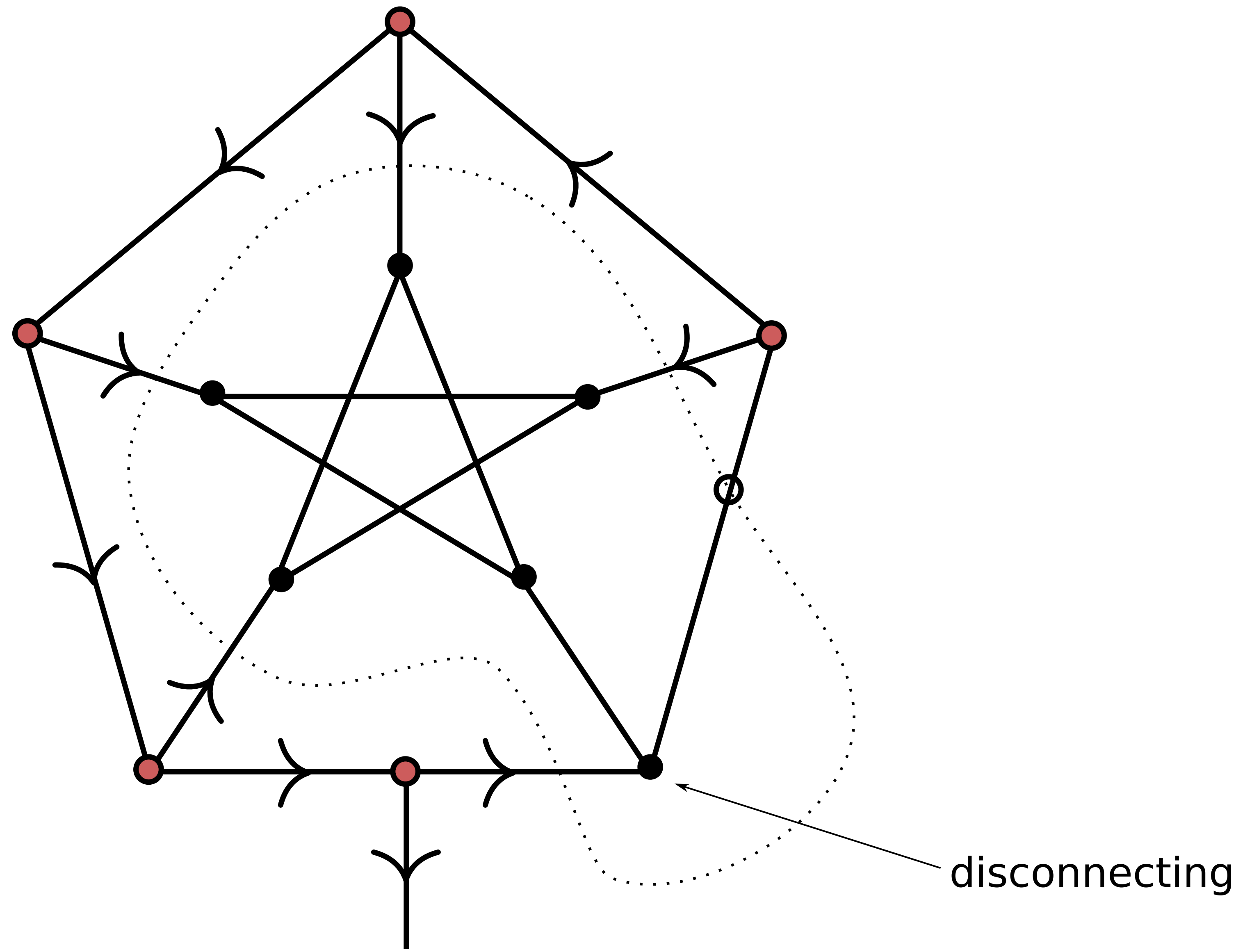
# An example



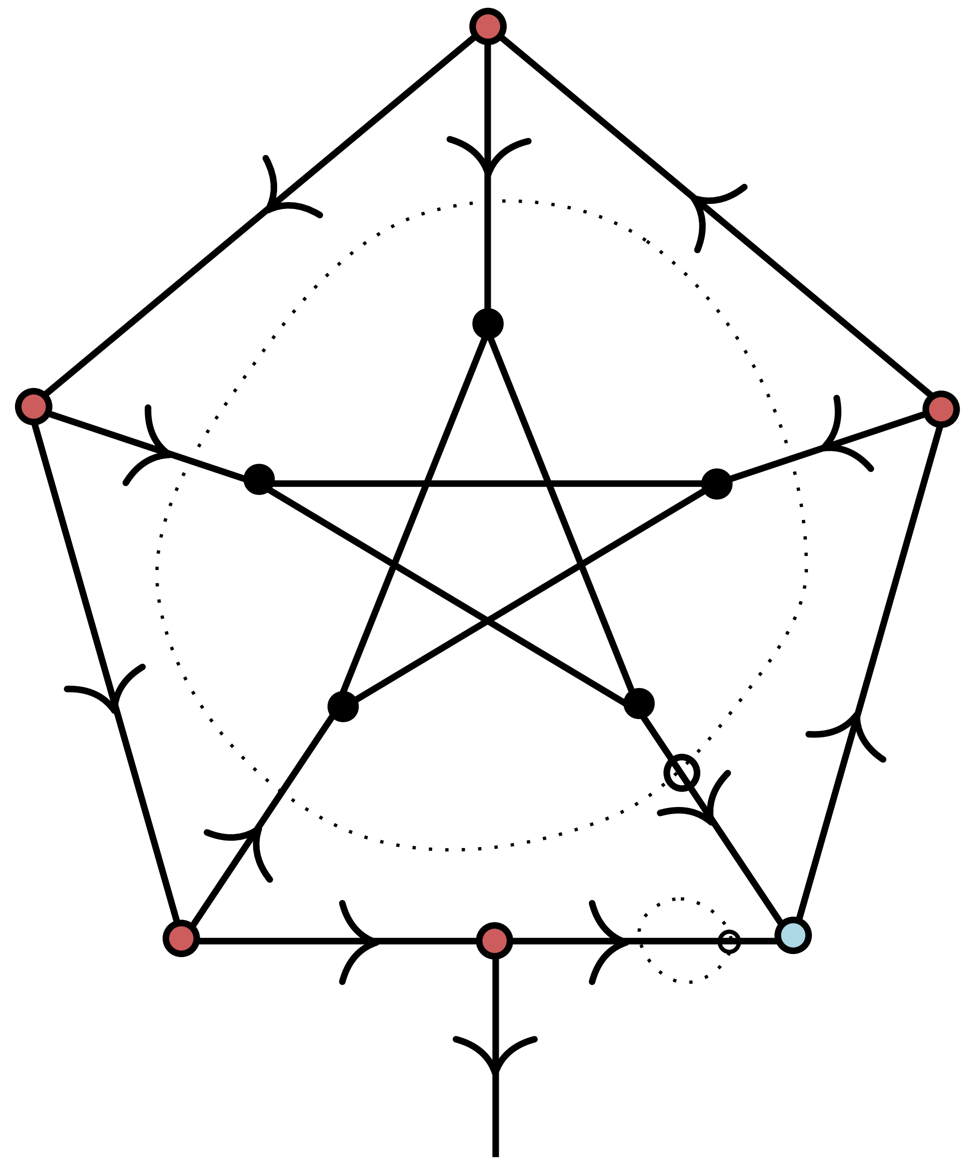
# An example



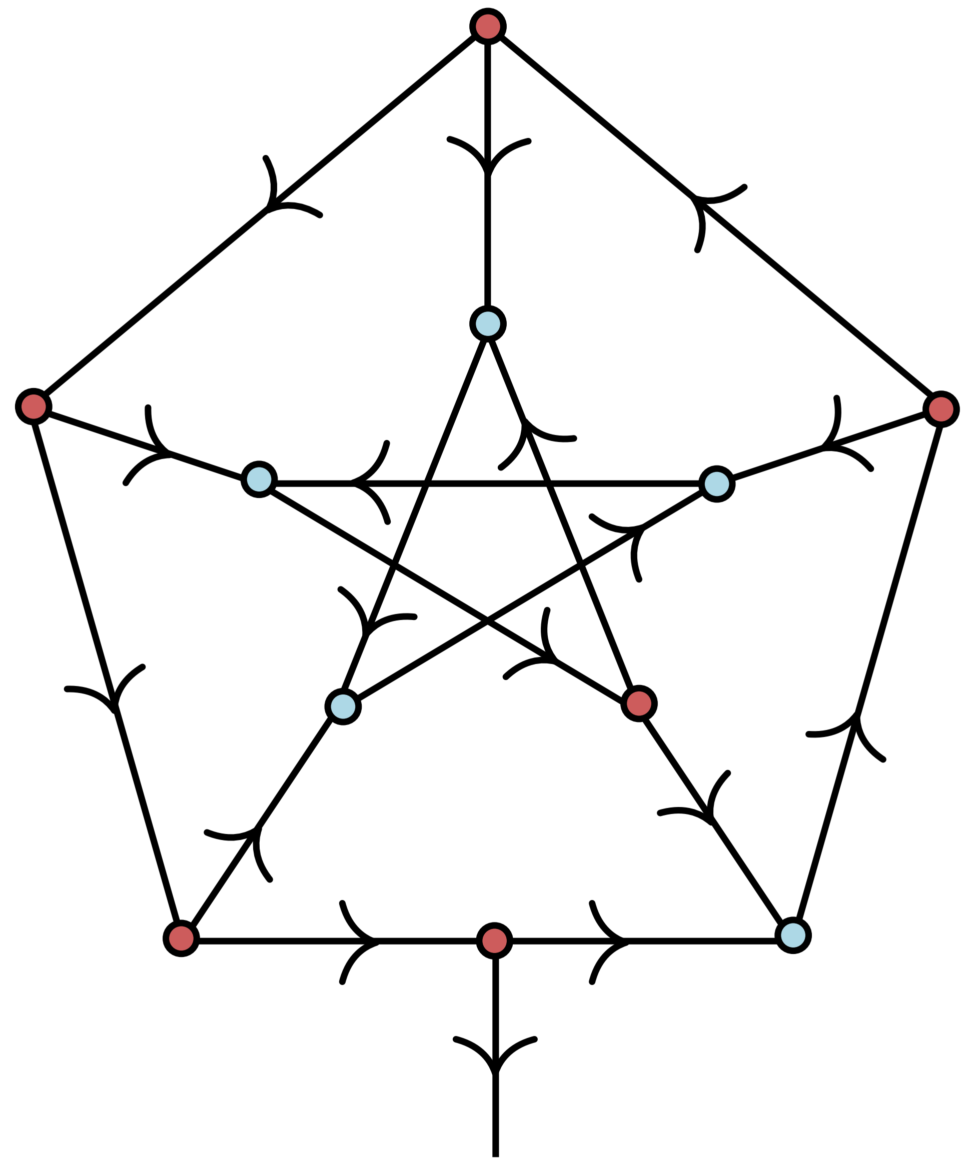
# An example



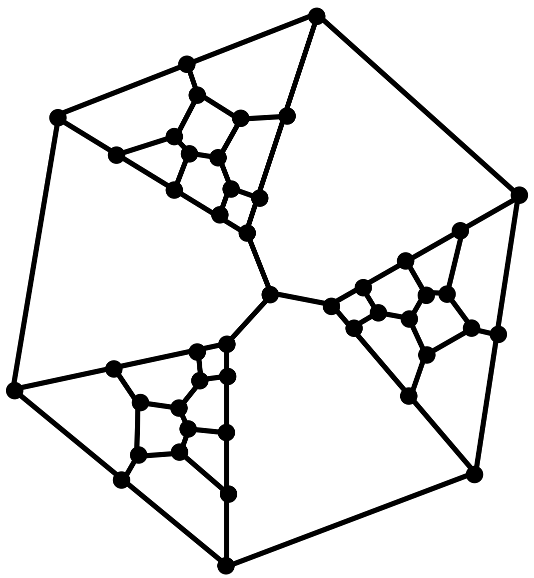
# An example



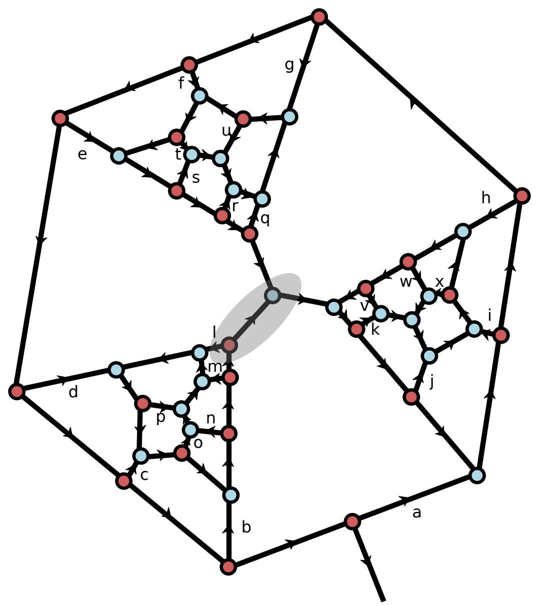
# An example



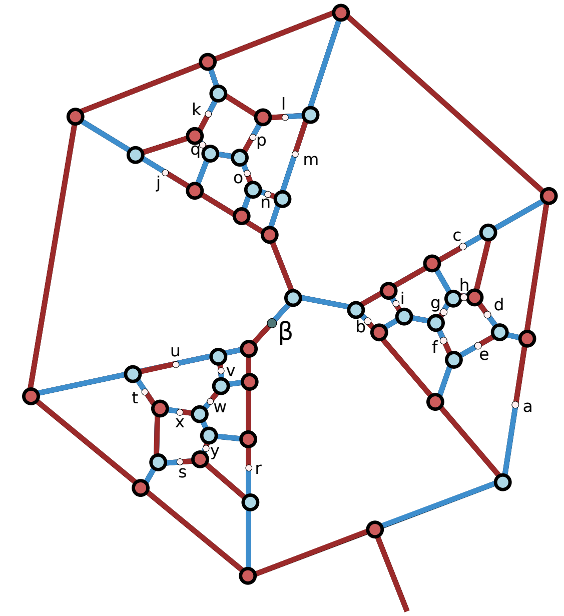
λa.λb.λc.λd.λe.a(λf.c(e(b(df))))



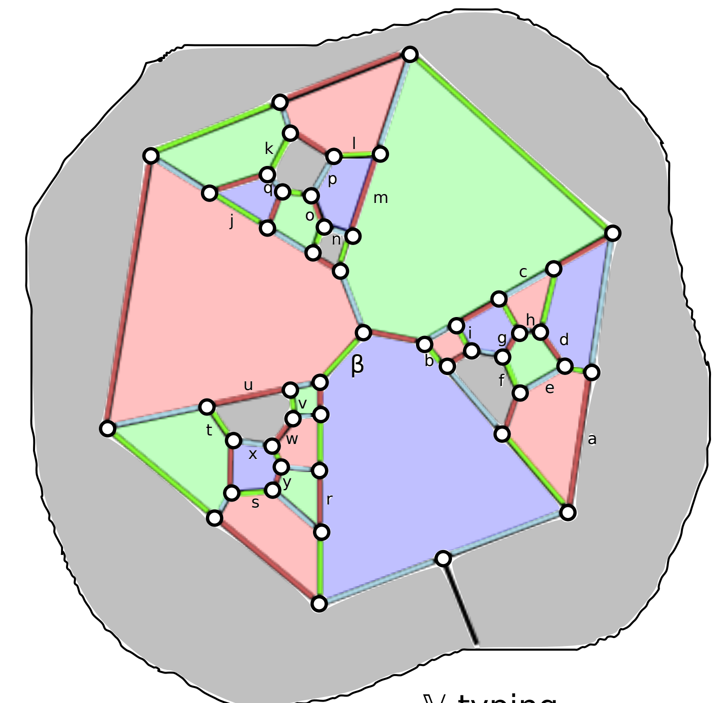
3-valent map



linear lambda term



principal typing



𝕍-typing

# 5. Coda

## (From Lambda Calculus to the Four Color Theorem...and beyond?)

# Typing as coloring

recall the typing rules:

$$\frac{}{x{:}A \vdash x{:}A} \qquad \frac{\Gamma \vdash t{:}A{\multimap}B \quad \Delta \vdash u{:}A}{\Gamma , \Delta \vdash t(u){:}B} \qquad \frac{\Gamma , x{:}A \vdash t{:}B}{\Gamma \vdash \lambda x.t{:}A{\multimap}B}$$

$$\frac{\Gamma , x{:}A, y{:}B , \Delta \vdash t{:}C}{\Gamma , y{:}B, x{:}A , \Delta \vdash t{:}C}$$

we can interpret types in any ab gp G, taking $A \multimap B := B - A$.

claim: any *ordered λ-term t has a typing in* $G = \mathbb{Z}_2 \times \mathbb{Z}_2$, *such that for every subterm u of t, u has type (0,0) iff u is closed.*

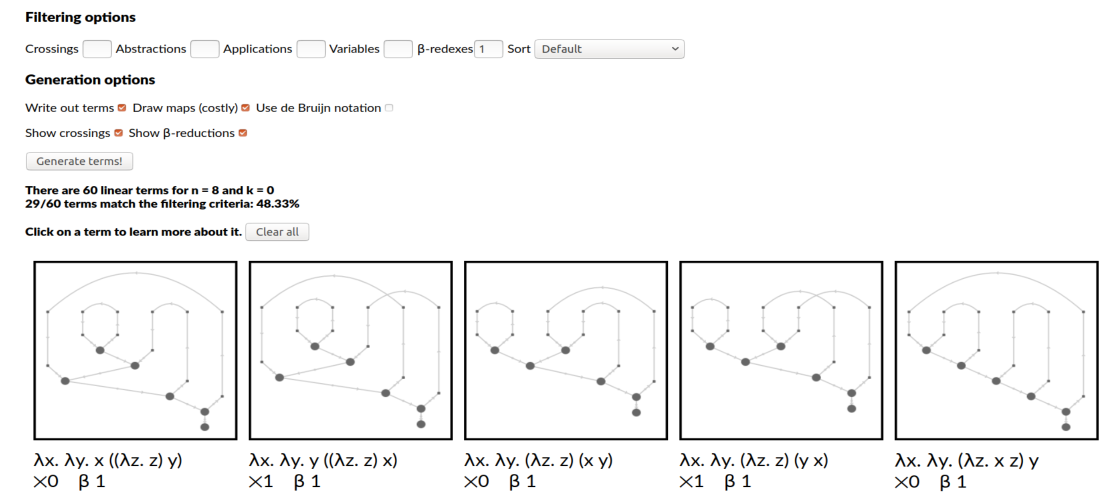
challenge problem: find a direct proof!

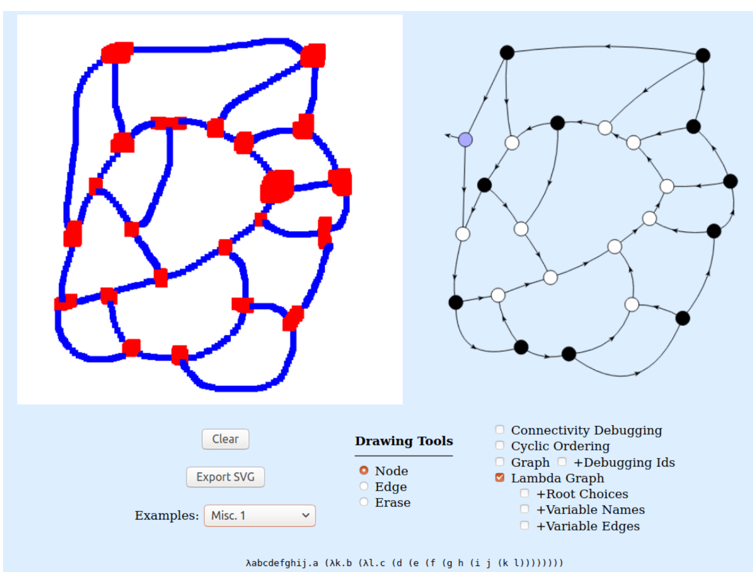
# Some tools for further exploration

George Kaye's λ-term visualiser and gallery

https://www.georgejkaye.com/lambda-visualiser/visualiser.html

https://www.georgejkaye.com/lambda-visualiser/gallery



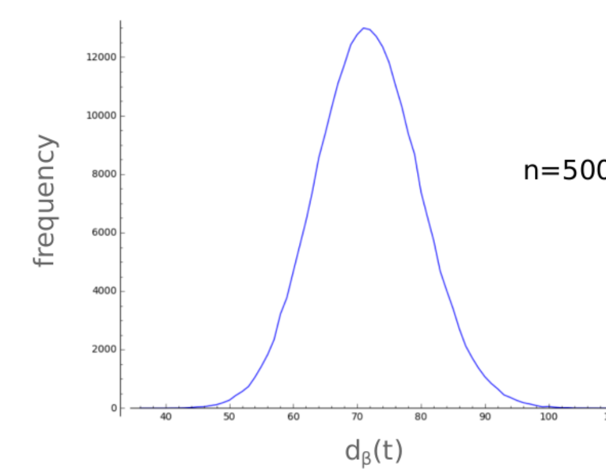




Jason Reed's Interactive Lambda Maps Toy

https://jcreedcmu.github.io/demo/lambda-map-drawer/public/index.html

LinLam: a library for experimental linear lambda calculus

https://github.com/noamz/linlam

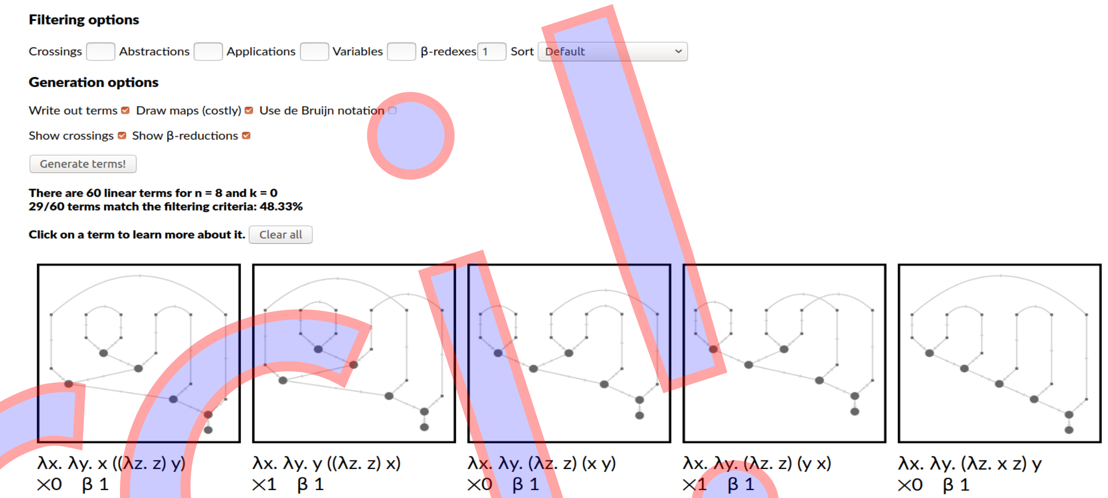


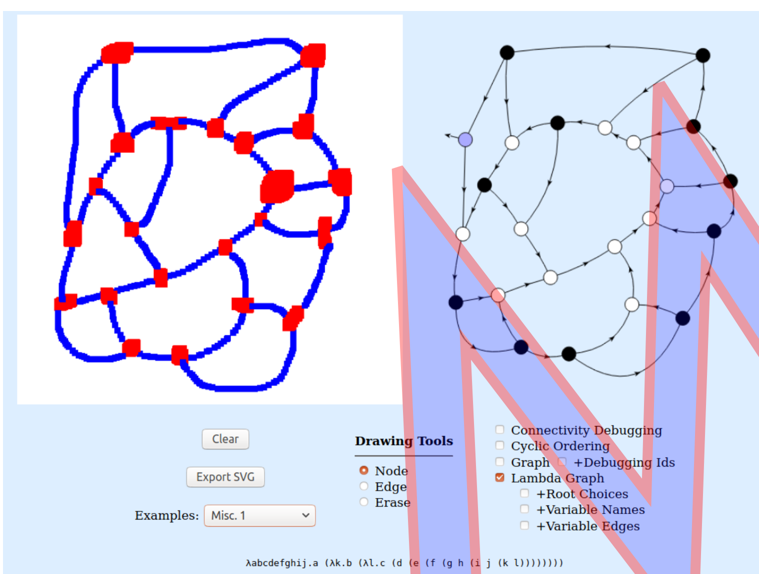
# Some tools for further exploration

George Kaye's λ-term visualiser and gallery

https://www.georgejkaye.com/lambda-visualiser/visualiser.html

https://www.georgejkaye.com/lambda-visualiser/gallery




Jason Reed's Interactive Lambda Maps Toy

https://jcreedcmu.github.io/demo/lambda-map-drawer/public/index.html

Merci!

LinLam: a library for experimental linear lambda calculus

https://github.com/noamz/linlam