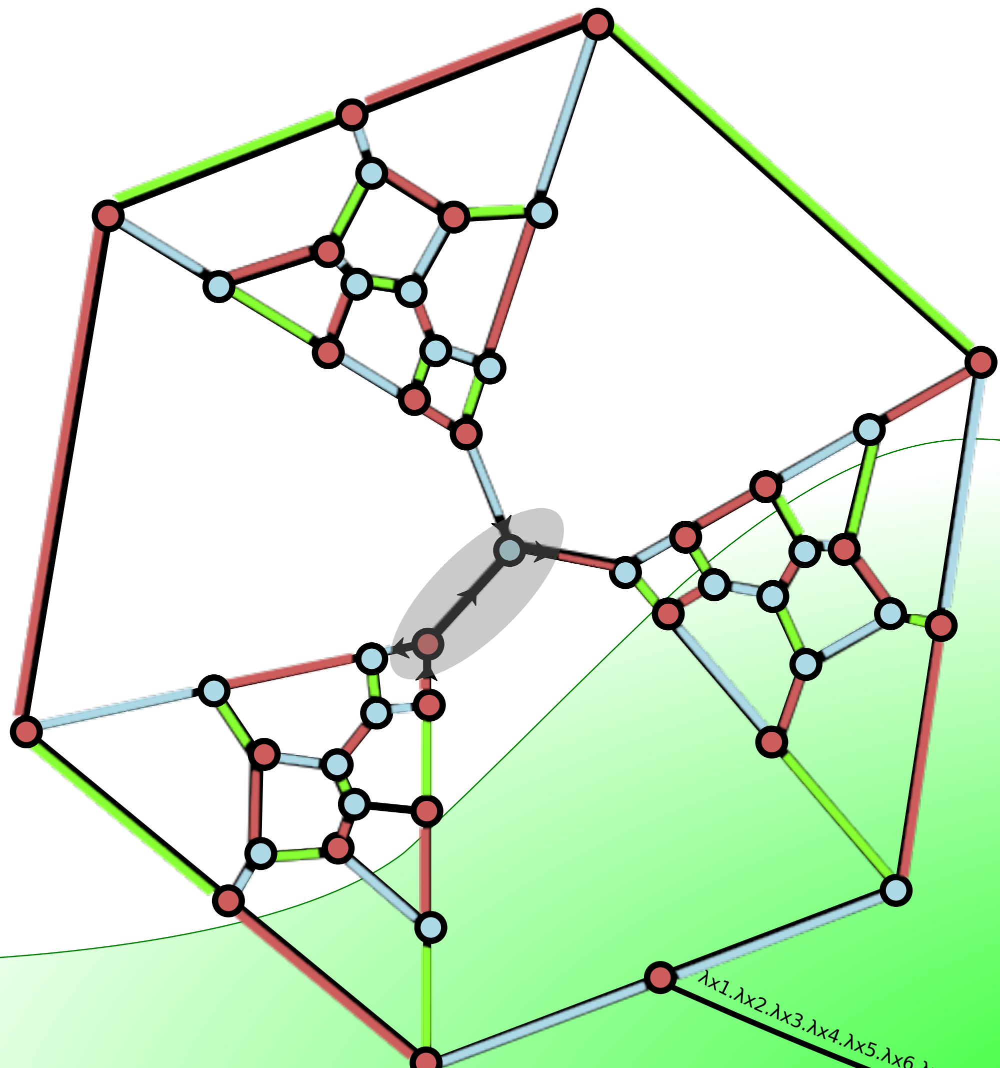
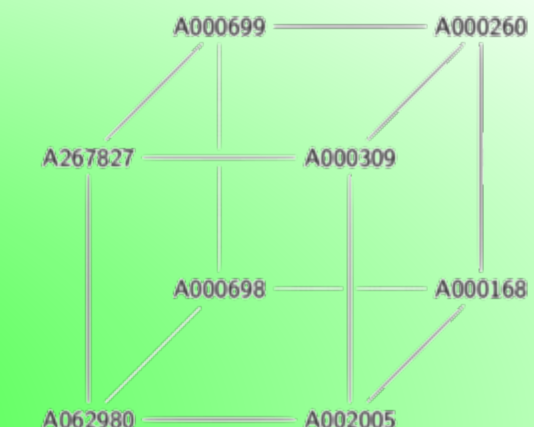
[this talk is being recorded]

slides: noamz.org/talks/expmath.2020.06.18.pdf

From ***Lambda Calculus*** *to the* ***Four Color Theorem*** *(via experimental math)*





Noam Zeilberger
Rutgers ExpMath Seminar

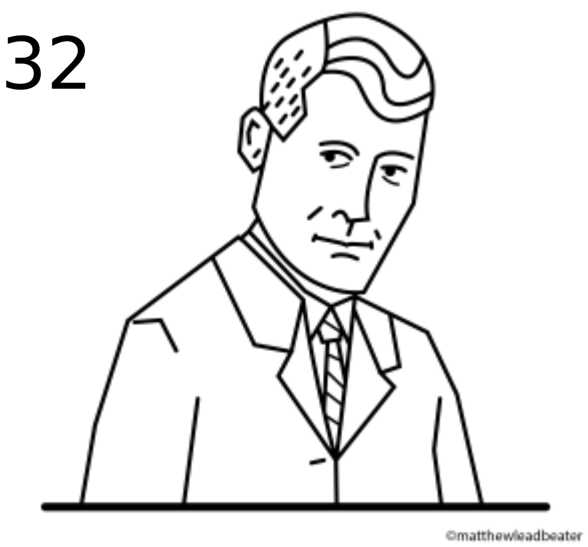
18 June 2020 @ 6PM UTC

1. Lambda calculus

x6.λx7.λx8.λx9.x1(λx10.λx11.(λx12.λx13.λx14.x2(λx15.x3(λx16.x4(x12(x13(x14(x15)(x16)))))))(λx17.λx18.λx19.x5(λx20.x6(λx21.x7(x17(x18(x19(x20)

Lambda calculus: a very brief history*

Invented by Alonzo Church around 1928, published in 1932



Original goal: a foundation for logic, more natural than Russell's type theory and Zermelo's set theory

One small problem: *inconsistency*!

Resolution: isolate an **untyped λ-calculus** for computation, and a **typed λ-calculus** for logic

Since then: λ-calculus (both typed and untyped) has served as the foundation and inspiration for countless PLs and proof assistants

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

Untyped lambda calculus: definition sketch (part 1)

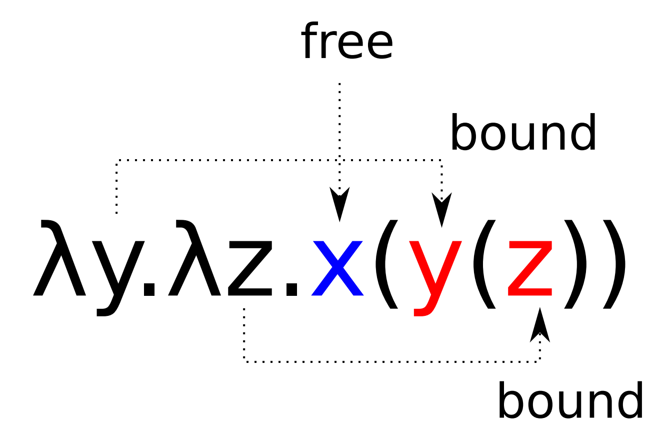
Formally, the set of **lambda terms** is closed under three constructs:

$t ::= x, y, \ldots$	variables
$\mid t(u)$	application
$\mid \lambda x.t$	abstraction

Quotient terms by "alpha-equivalence" (renaming of variables), e.g.:

$$\lambda x.\lambda y.x(y)(y) \equiv_\alpha \lambda f.\lambda a.f(a)(a)$$

Distinguish **free variables** from **bound variables**:



Untyped lambda calculus: definition sketch (part 2)

Computation is encoded in the rule of **beta-reduction**:

$$(\lambda x.t)(u) \;\to_\beta\; t[u/x]$$

Example:

$$
\begin{aligned}
& (\lambda f.\lambda a.f(a)(a))\ (\lambda x.x)\ y \\
\to_\beta\ & (\lambda a.(\lambda x.x)(a)(a))\ y \\
\to_\beta\ & (\lambda x.x)(y)(y) \\
\to_\beta\ & y(y)
\end{aligned}
$$

Fact: the beta-normal form of a term is unique if it exists, but in general it is uncomputable (Church, 1936).

Fixed-point combinators and (non-)linearity

A **fixed-point combinator** is a closed term Y such that $Yx \equiv_\beta x(Yx)$.

The first FP combinator in print (1937) was Turing's:

$$Y := (\lambda x.\lambda y.y(xxy))(\lambda x.\lambda y.y(xxy))$$

Observe the doubled uses of λ-bound variables x and y. By contrast, a term is said to be **linear** if every variable is used exactly once.

Fact: determining the beta-normal form of a linear term is complete for polynomial time (Mairson, 2004).

Combinatory logic: an ancestor to λ-calculus

Moses Schönfinkel

A var-free system,
invented by Schönfinkel
rebirthed by Curry.

Haskell Curry

Photos of logicians obtained from the Open Logic Project (see credits on page)

Combinatory logic and untyped λ-calculus

There exists a basis of *particularly powerful terms:*

B := λx.λy.λz.x(yz)

C := λx.λy.λz.(xz)y

I := λx.x

K := λx.λy.x

W := λx.λy.(xy)y

The set {B,C,K,W,I} forms a *basis* meaning any (closed) λ-term is in the closure of these terms under application and β-reduction.*

*In fact, I is redundant since W(K) $\to_\beta$ I

The set {B,C,I} is a basis in the same sense for linear λ-terms.

Combinatory logic and types

The combinators can be assigned *types* corresponding to basic axioms of implication:

B : $(\beta \supset \gamma) \supset ((\alpha \supset \beta) \supset (\alpha \supset \gamma))$

C : $(\alpha \supset (\beta \supset \gamma)) \supset (\beta \supset (\alpha \supset \gamma))$

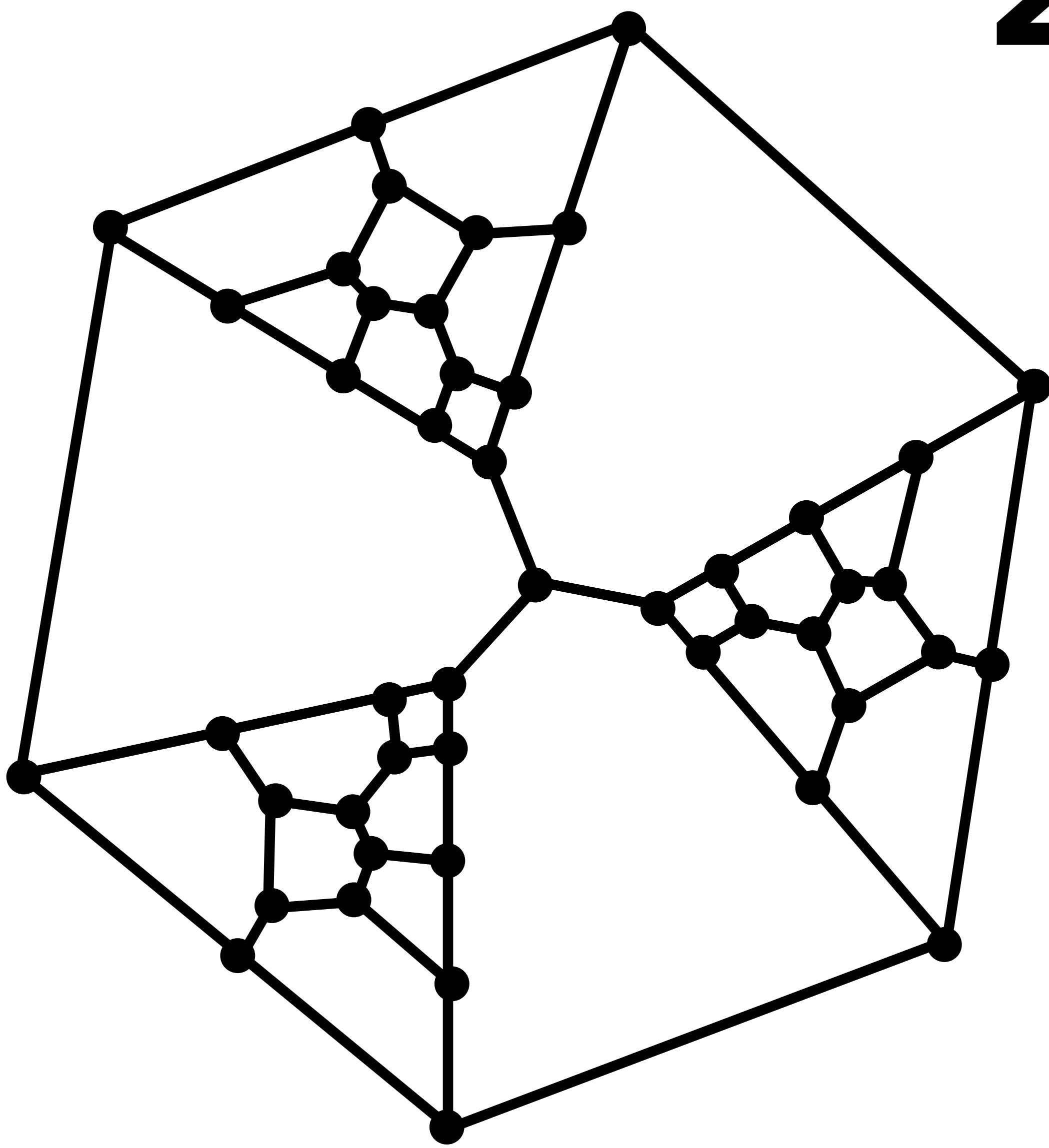
I : $\alpha \supset \alpha$

K : $\alpha \supset (\beta \supset \alpha)$

W : $(\alpha \supset (\alpha \supset \beta)) \supset (\alpha \supset \beta)$

More generally, terms can be seen as proofs in (purely implicative, intuitionistic) logic, although not every term can be assigned a type or the logic would be inconsistent (Y : $(\alpha \supset \alpha) \supset \alpha$).

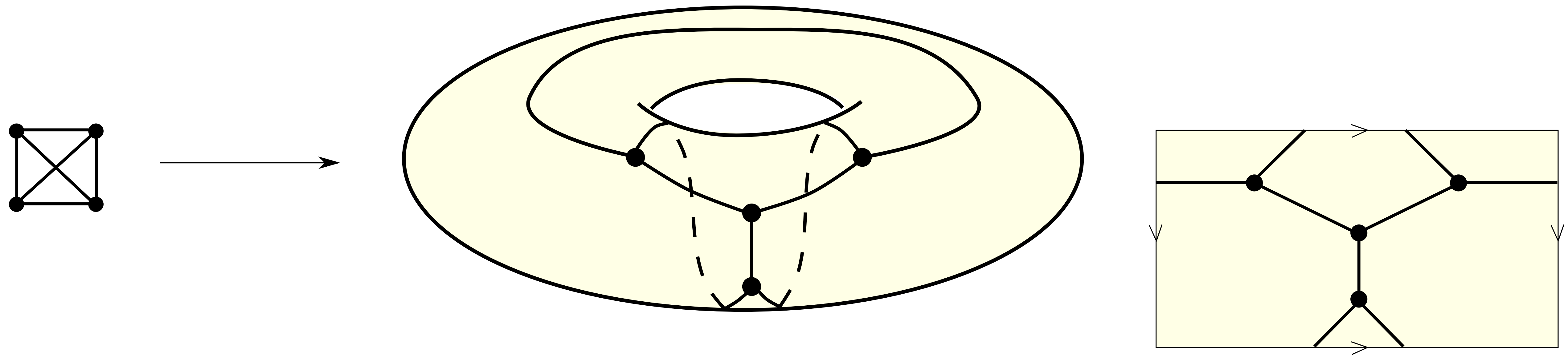
This forms part of the so-called "Curry-Howard correspondence".



2. What is a map?

Topological definition

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



considered up to deformation of the underlying surface.

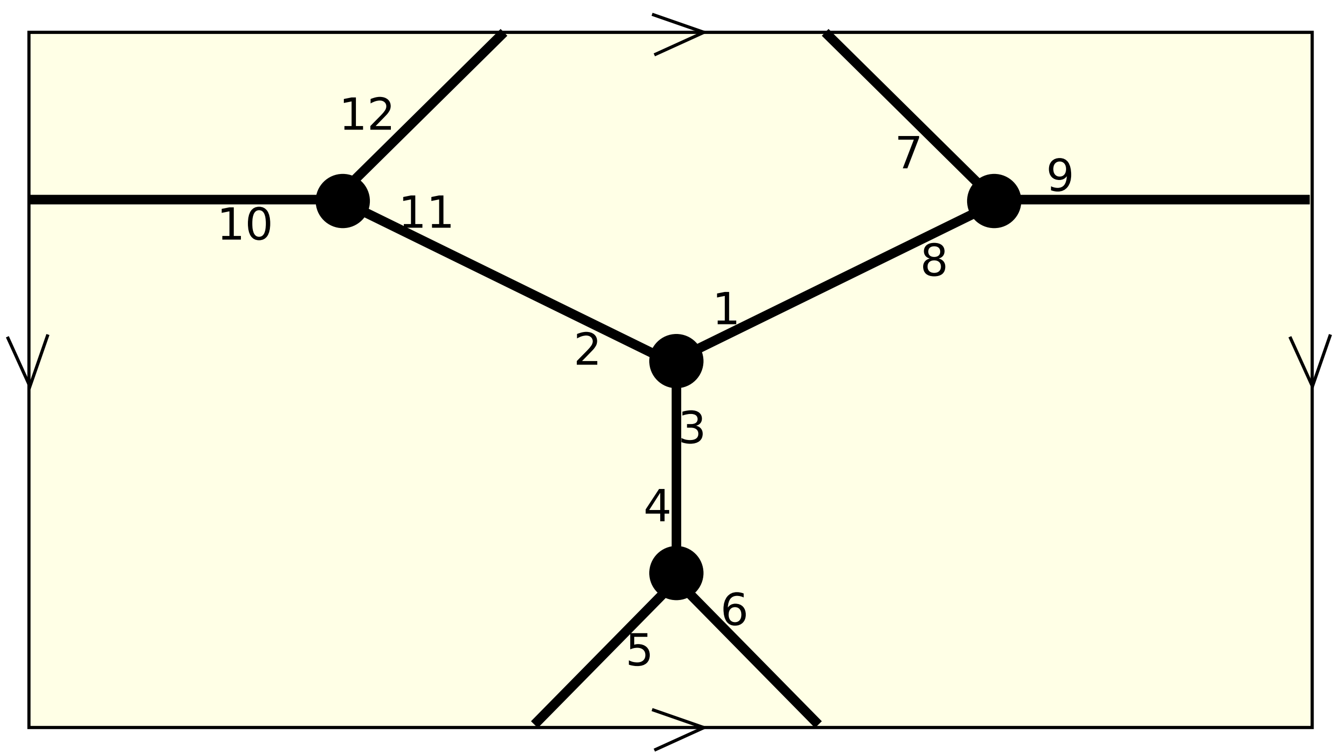
*All surfaces are assumed to be connected and oriented throughout this talk

Algebraic definition

map = transitive permutation representation of the group

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

considered up to G-equivariant isomorphism.



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

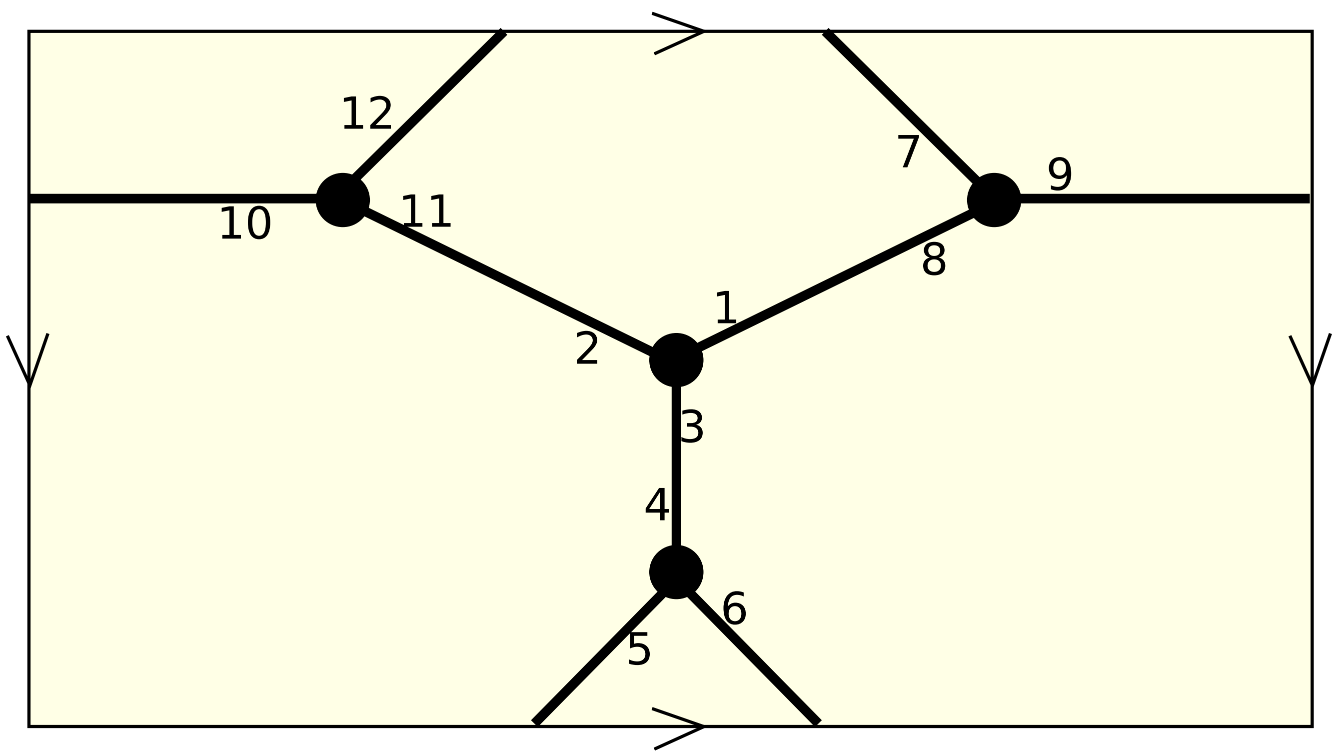
$$e = (1\ 8)(2\ 11)(3\ 4)(5\ 12)(6\ 7)(9\ 10)$$

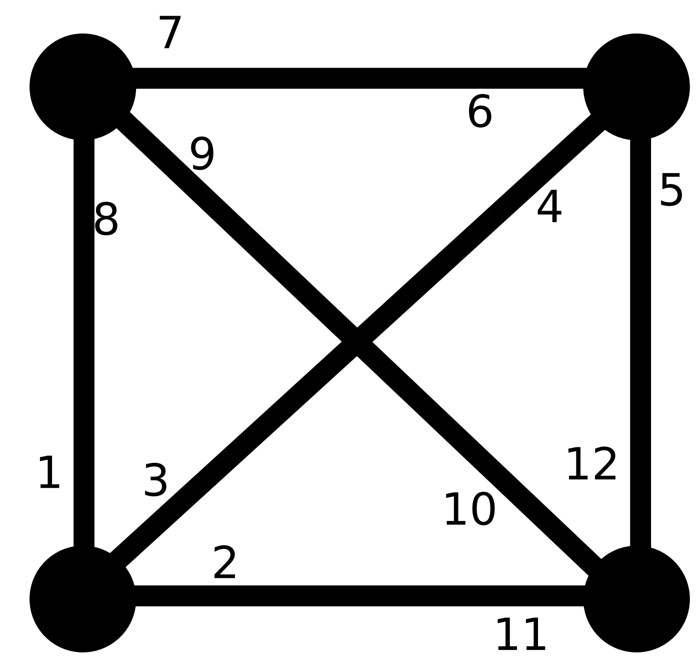
$$f = (1\ 7\ 5\ 11)(2\ 10\ 8\ 3\ 6\ 9\ 12\ 4)$$

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

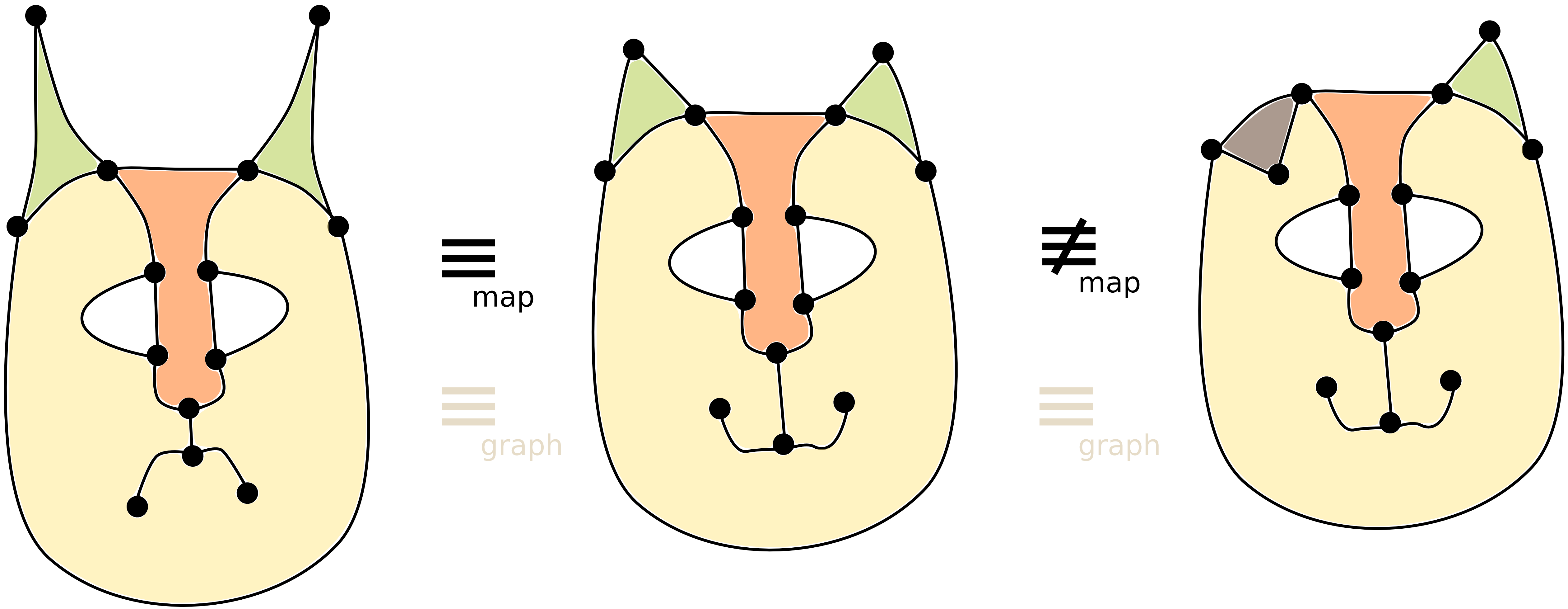
Combinatorial definition

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

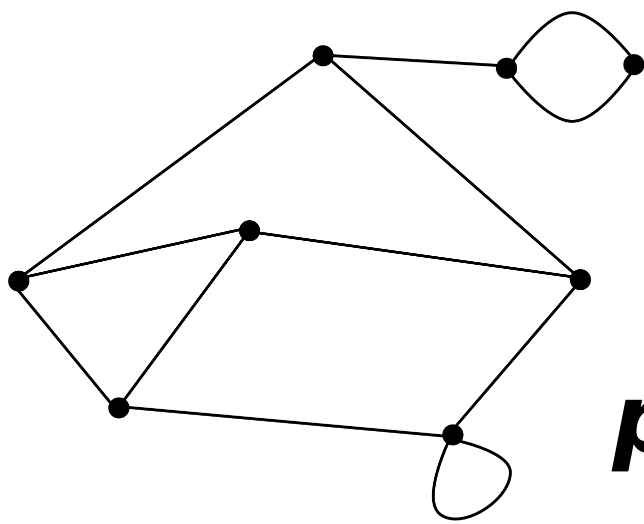




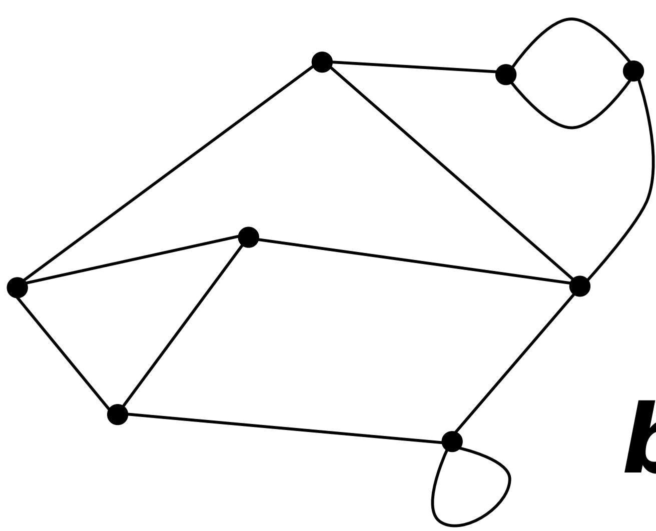
Graph versus Map



Some special kinds of maps

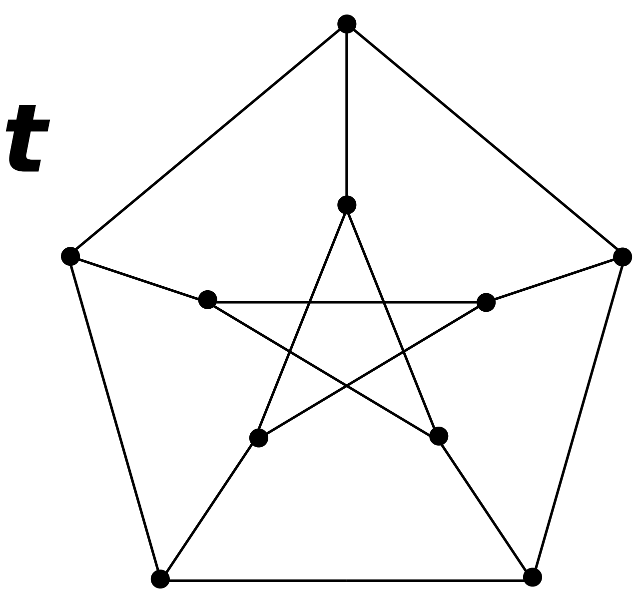


planar



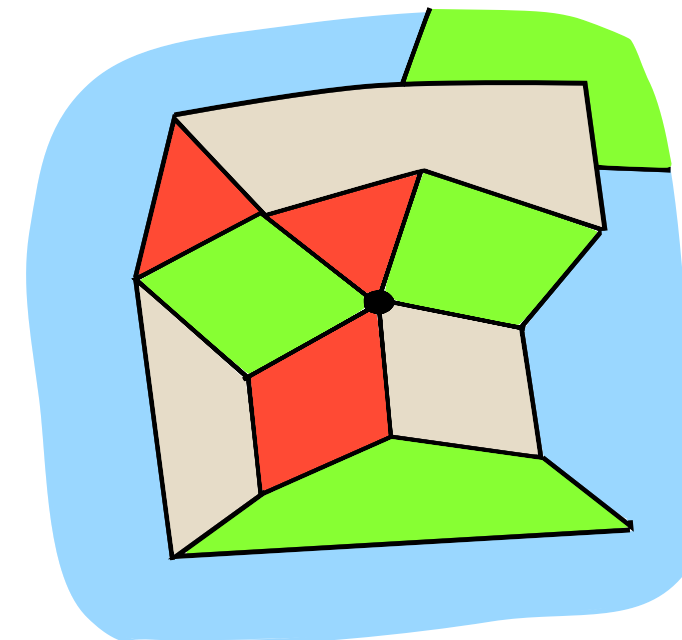
bridgeless

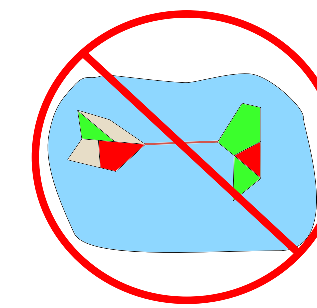
3-valent



Four Color Theorem

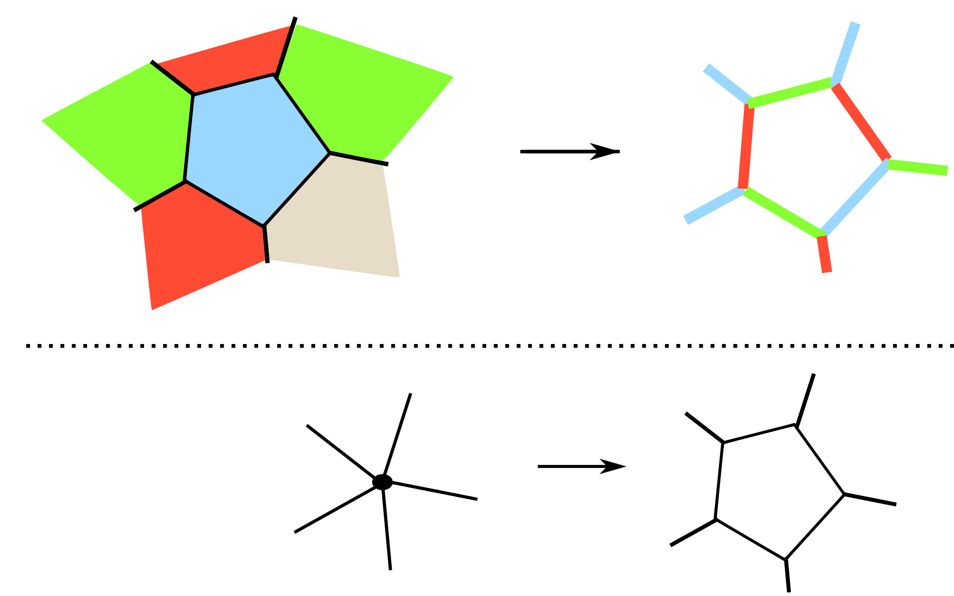
The 4CT is a statement about maps.





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

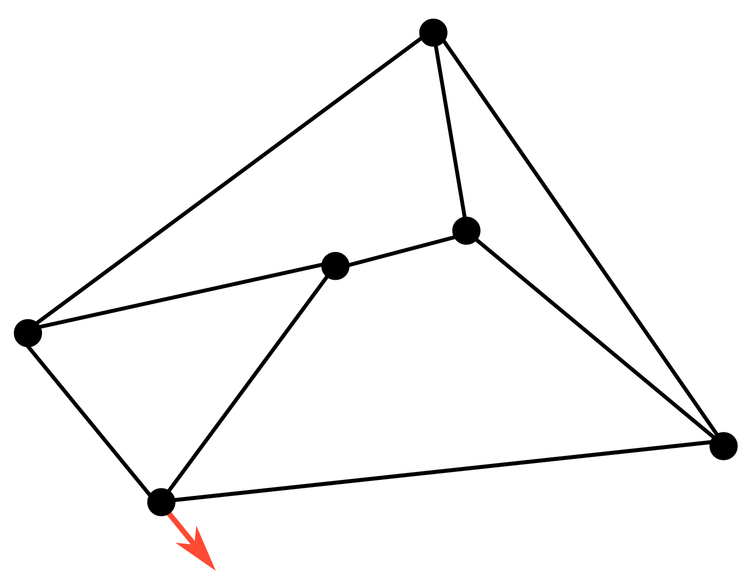
Map enumeration

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*

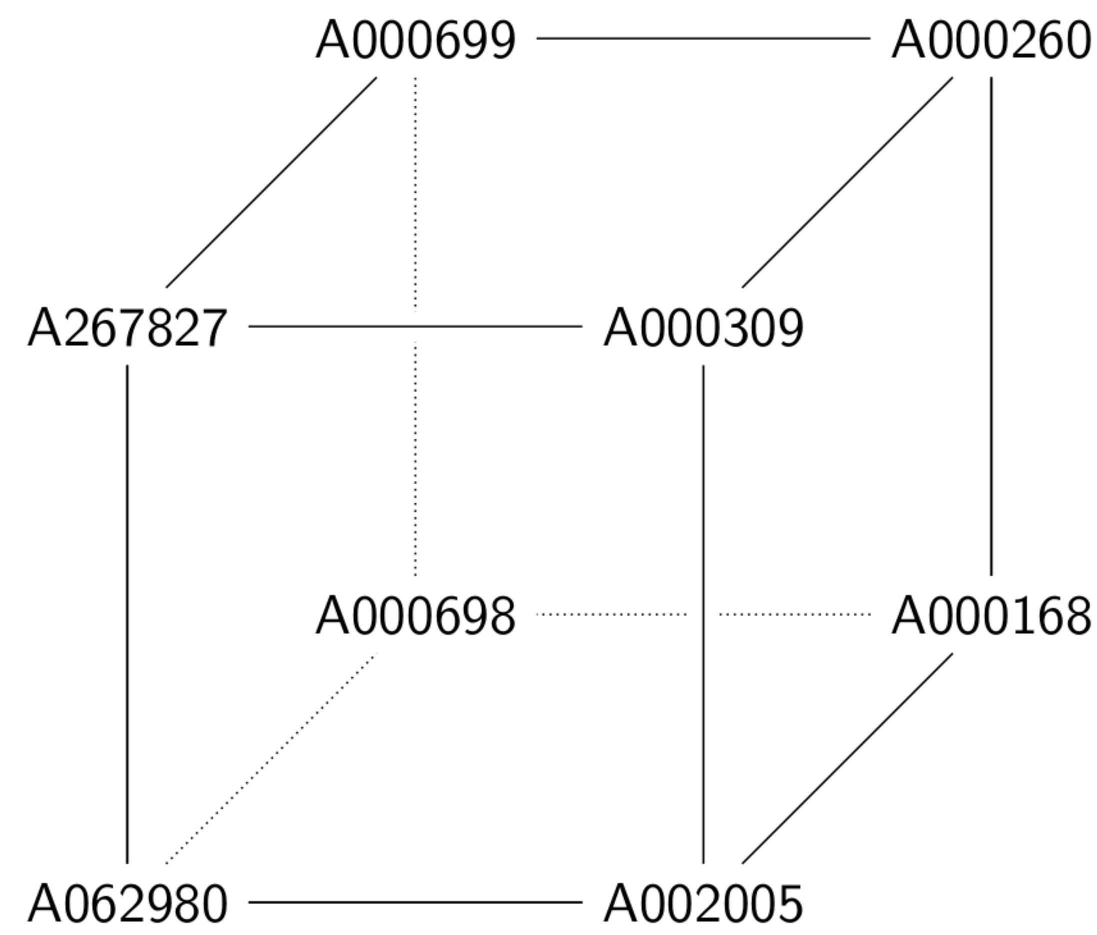
$$\frac{2(2n)!\,3^n}{n!\,(n+2)!}.$$

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



3. How on Earth are these topics related??

An innocent idea

In May 2014, for some reason* I thought it could be interesting to count linear lambda terms with the added restriction that terms are both *β-normal* and *ordered* (variables used not just once, but also in the order they are bound).

*related to certain categorical models of typing.

Counting β-normal ordered linear λ-terms

Let F(n,k) = # β-normal ordered terms w/*n subterms* and *k free vars.*
Let G(n,k) = same thing but restricting to terms that are not λs.

F and G satisfy the following mutual recurrence:

$$F(n,k) = F(n-1,k+1) + G(n,k)$$

$$G(n,k) = [n=1 \,\&\, k=1] + \sum_{m,j} G(m,j) * F(n-1-m,k-j)$$

Consider the sequence F(3*n+2,0) for n=0,1,...

$\lambda x.x$

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

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

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

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

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

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

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

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

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

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

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

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.

One piece of a larger puzzle

With Alain Giorgetti, we gave a bijection between β-normal ordered* linear λ-terms and rooted planar maps, albeit not so easy to interpret.

Independently (and a few years earlier), another group of people (Olivier Bodini, Danièle Gardy, and Alice Jacquot) studied linear lambda calculus coming from the completely different angle of analytic combinatorics, and found a natural bijection between general linear λ-terms and rooted *3-valent maps* of arbitrary genus.

Looking at this pair of connections, one may wonder whether they form part of a bigger picture...and it turns out they do!

*our bijection went via "skew-ordered" terms, which is part of what made it difficult to interpret.

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

ordered *vs.* non-ordered

λx.λy.λz.x(yz) λx.λy.λz.(xz)y

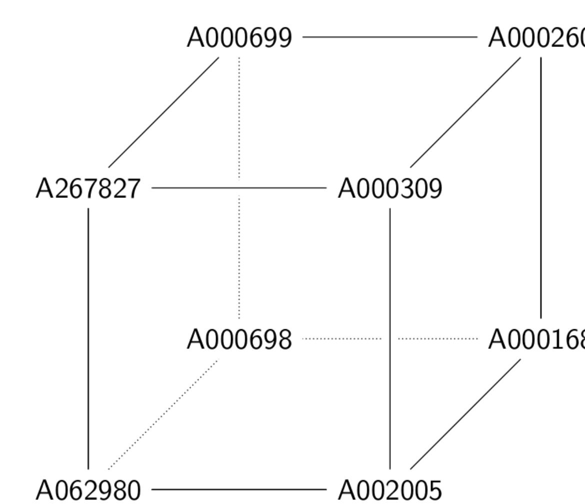
unitless *vs.* non-unitless

λx.λy.x(y) λx.x(λy.y)

normal *vs.* non-normal

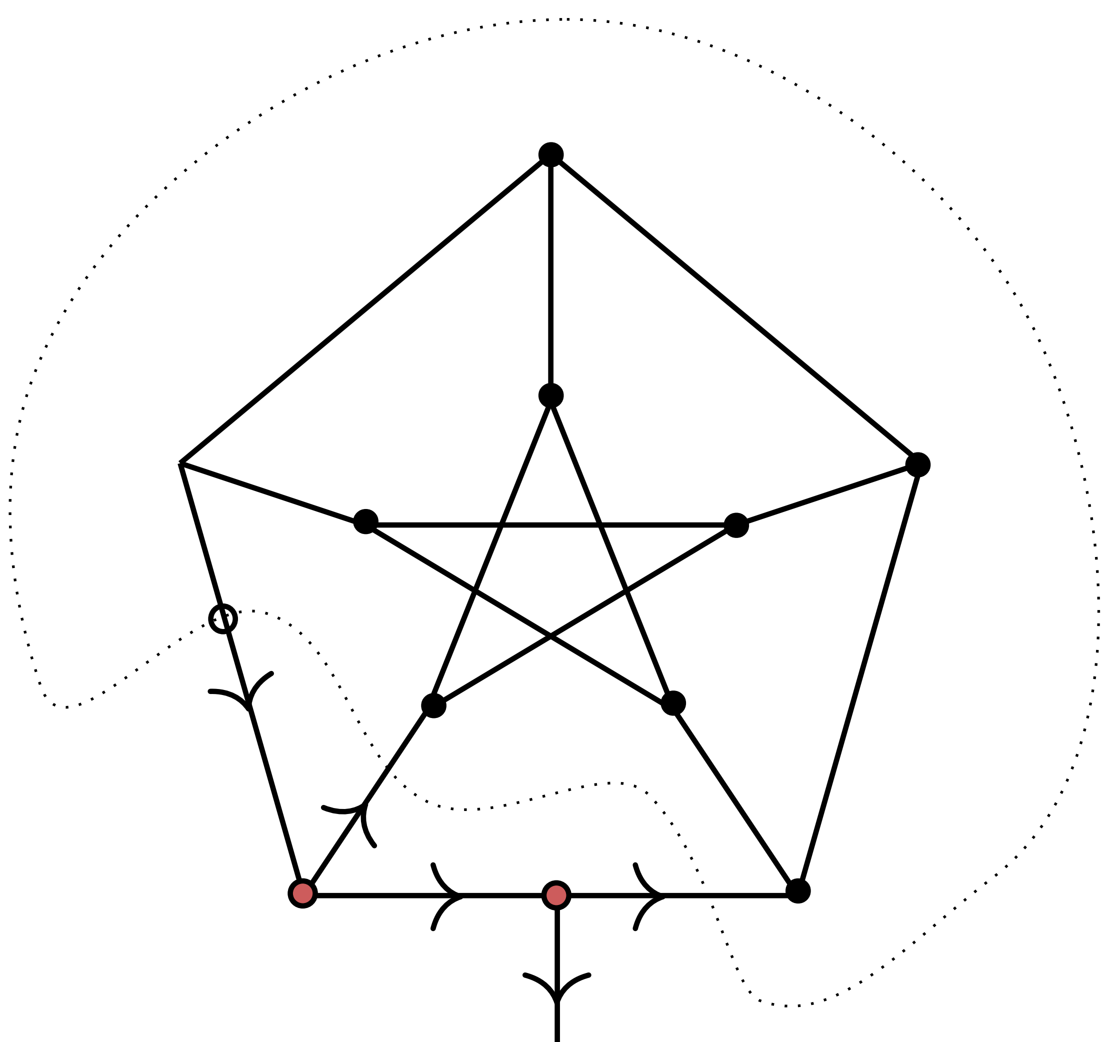
λx.λy.x(λz.yz) λx.λy.(λz.xz)y

λx.λy.t ~ λy.λx.t



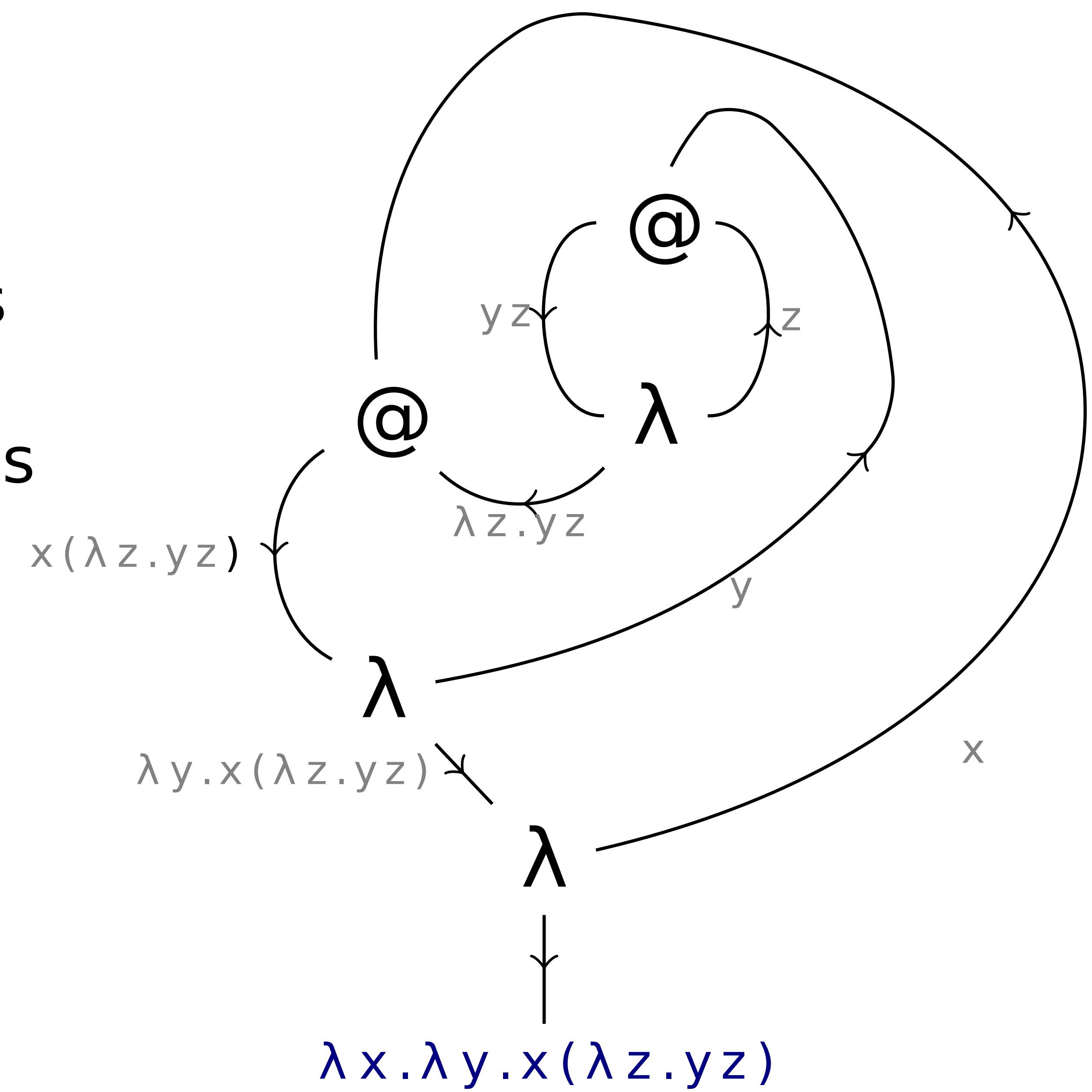
4. Between linear λ-terms and rooted 3-valent maps

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



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

A λ-term can be represented as a "tree w/pointers", either with λ-nodes pointing to the occurrences of bound variables, or conversely with variables pointing to their binders. This idea is especially natural for linear terms.



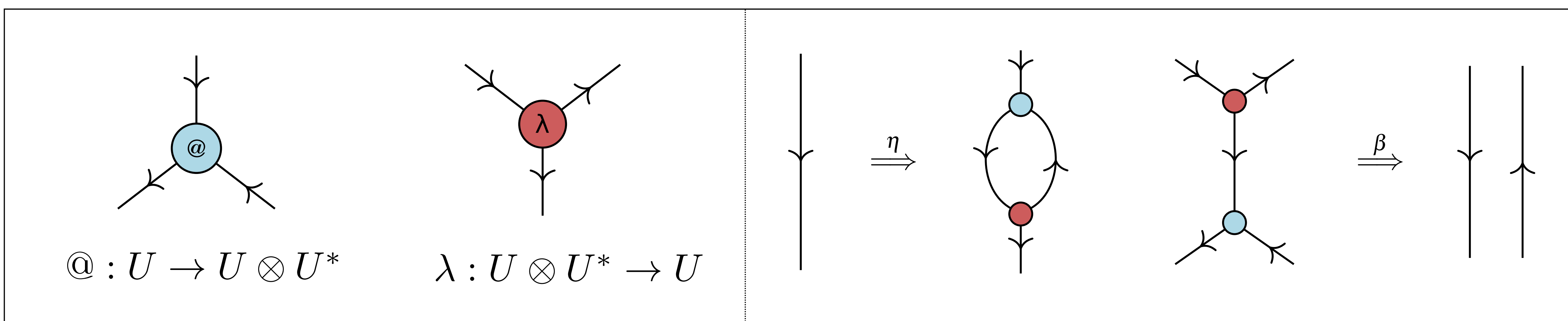
*The idea itself is natural and should probably be called folklore. The earliest explicit description I know of (currently) is in Knuth's "Examples of Formal Semantics" (1970), but it was developed more deeply and independently from different perspectives in the PhD theses of C. P. Wadsworth (1971) and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

λ-graphs as string diagrams

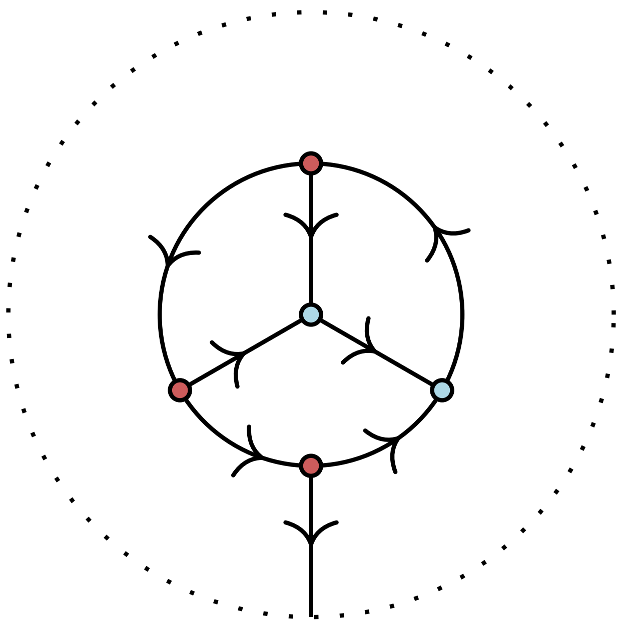
This idea can also be understood within the categorical framework of "string diagrams", by interpreting λ-terms (after D. Scott) as *endomorphisms of a reflexive object*

$$U \underset{\lambda}{\overset{@}{\underset{\perp}{\rightleftarrows}}} U \multimap U$$

in a symmetric monoidal closed bicategory.

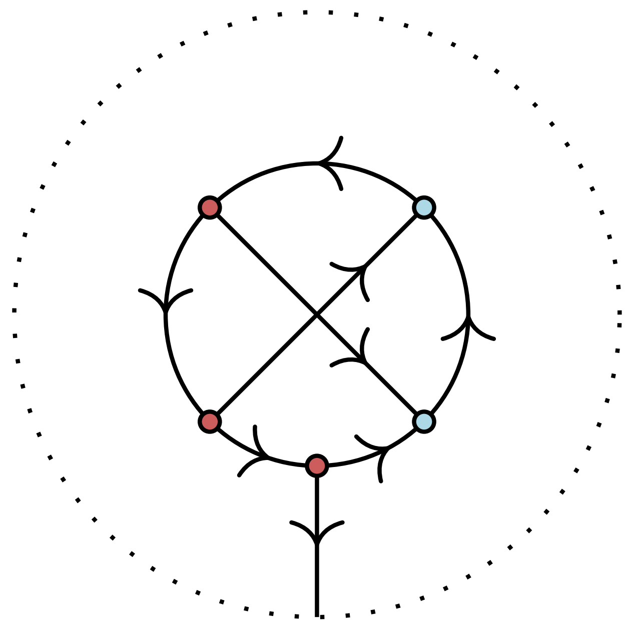


From linear λ-terms to rooted 3-valent maps



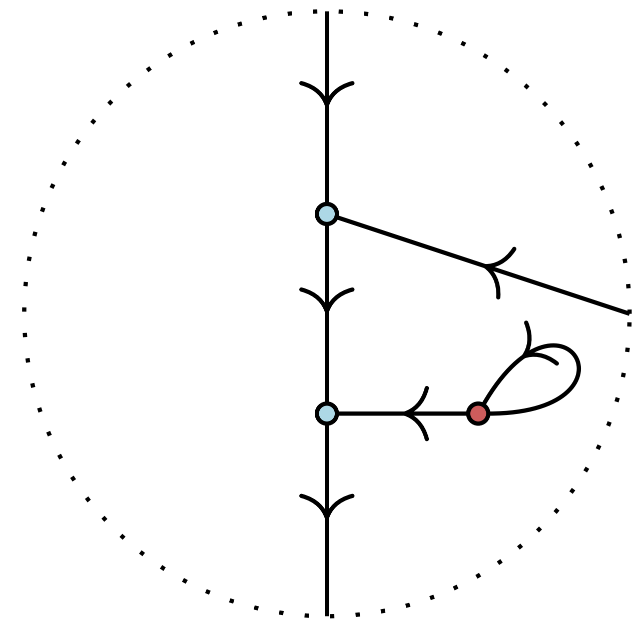
λx.λy.λz.x(yz)

(**B**)

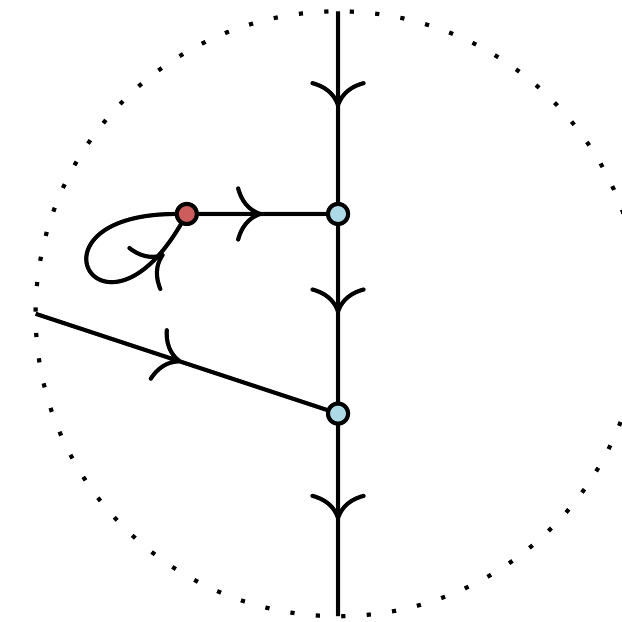


λx.λy.λz.(xz)y

(**C**)

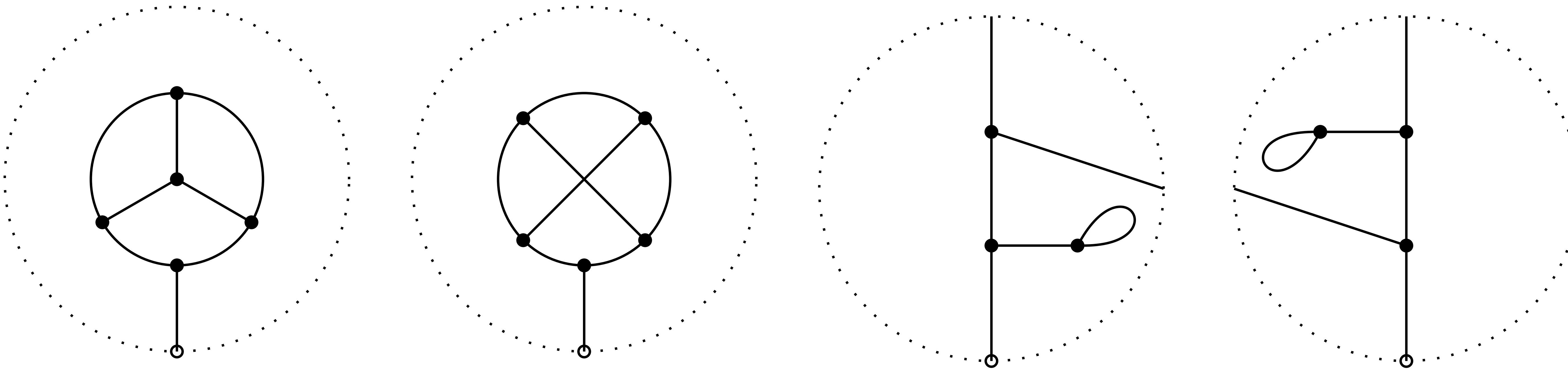


x,y ⊢ (xy)(λz.z)



x,y ⊢ x((λz.z)y)

From linear λ-terms to rooted 3-valent maps

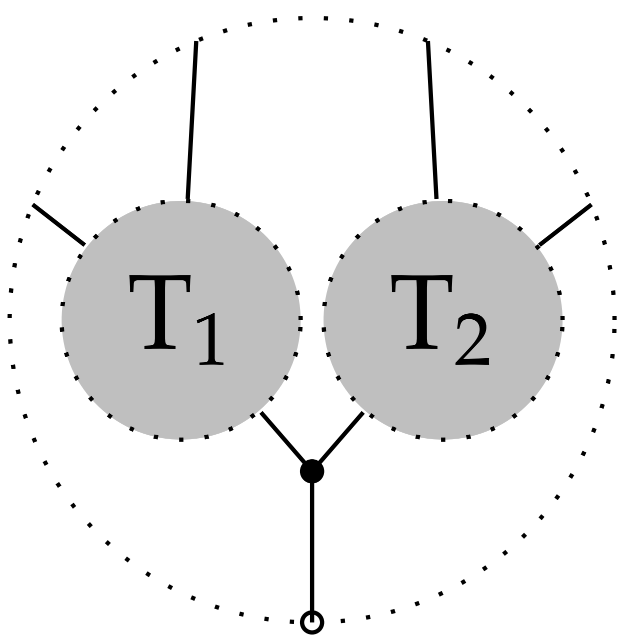


$\lambda x.\lambda y.\lambda z.x(yz)$ (**B**) | $\lambda x.\lambda y.\lambda z.(xz)y$ (**C**) | $x,y \vdash (xy)(\lambda z.z)$ | $x,y \vdash x((\lambda z.z)y)$

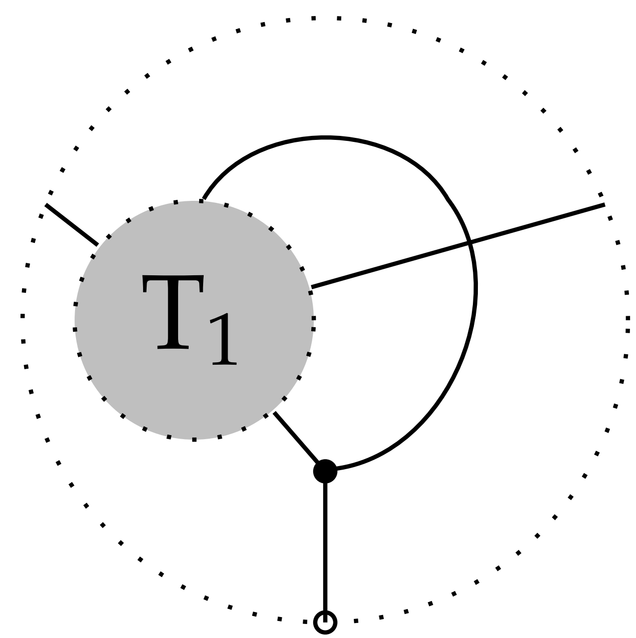
From rooted 3-valent maps to linear λ-terms

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

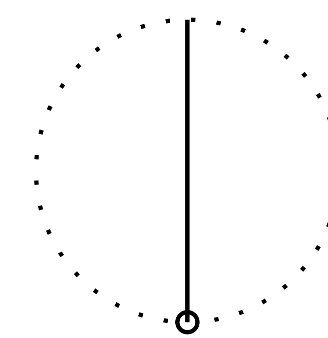
Step #2: observe any such map must have one of the following forms:



disconnecting
root vertex



connecting
root vertex

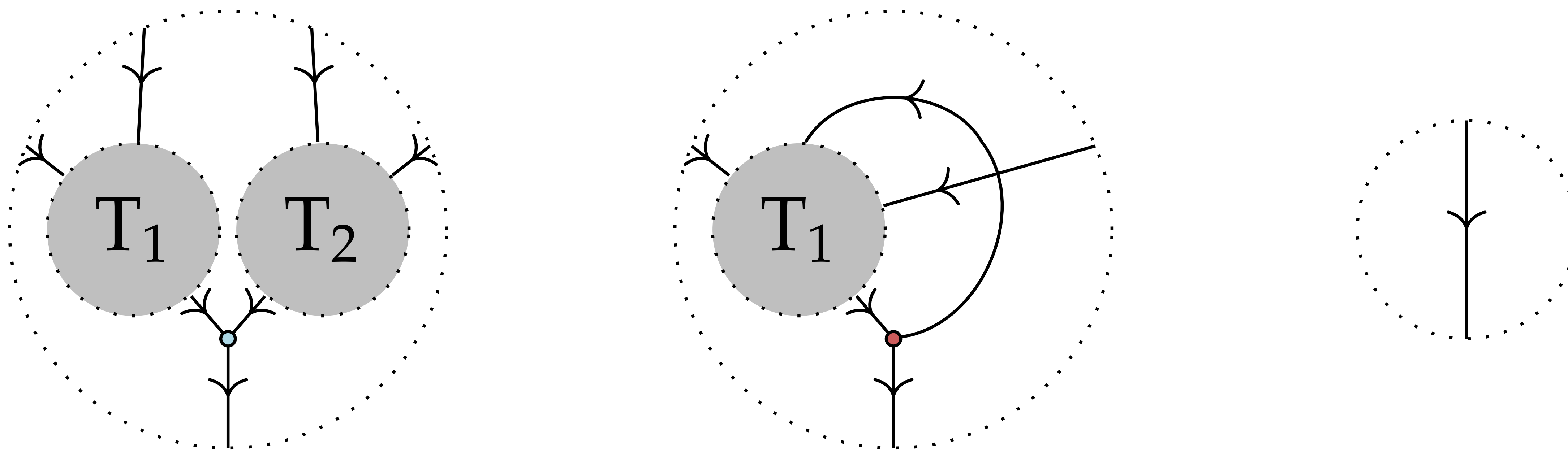


no
root vertex

From rooted 3-valent maps to linear λ-terms

⋮

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

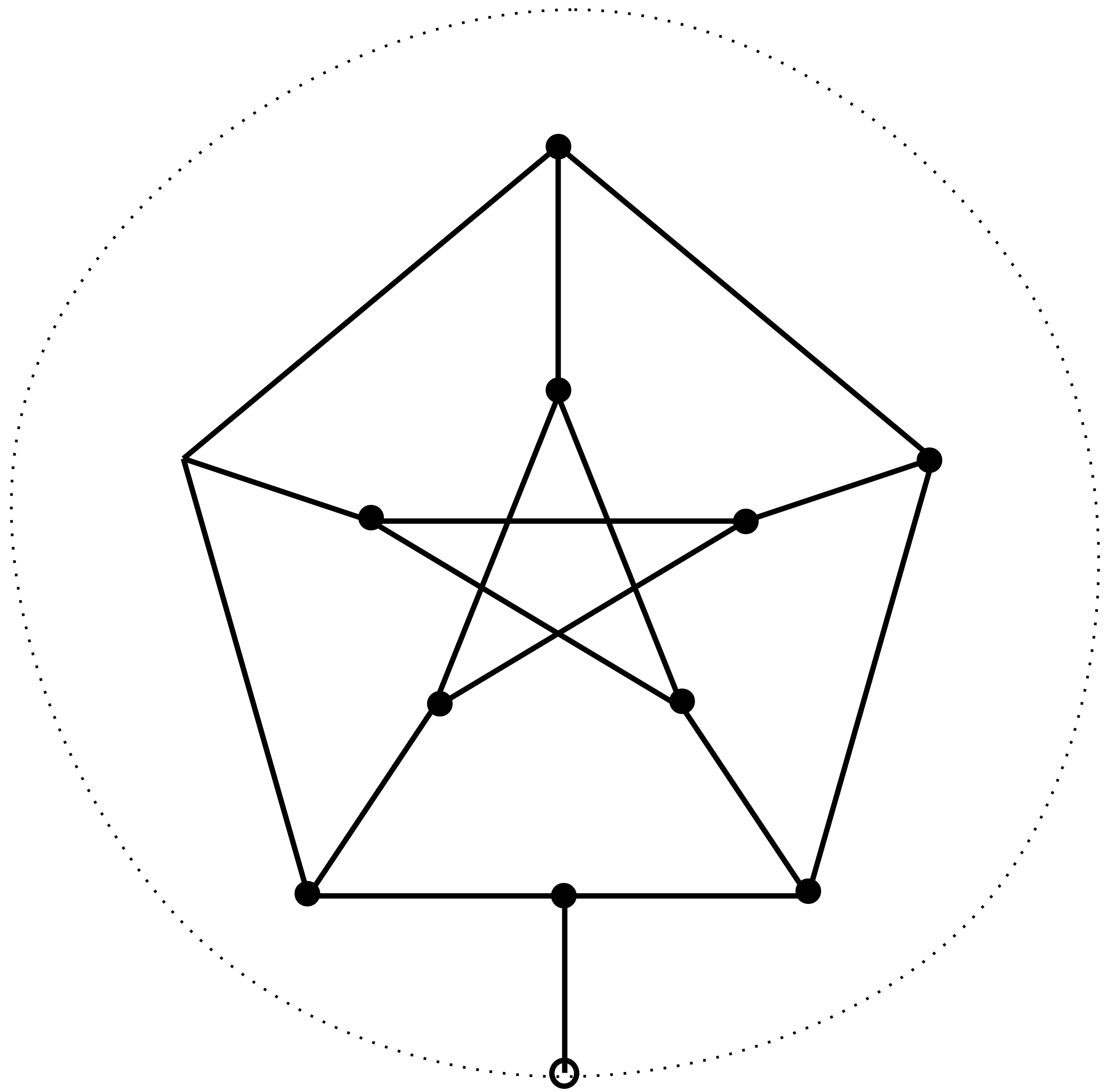


application

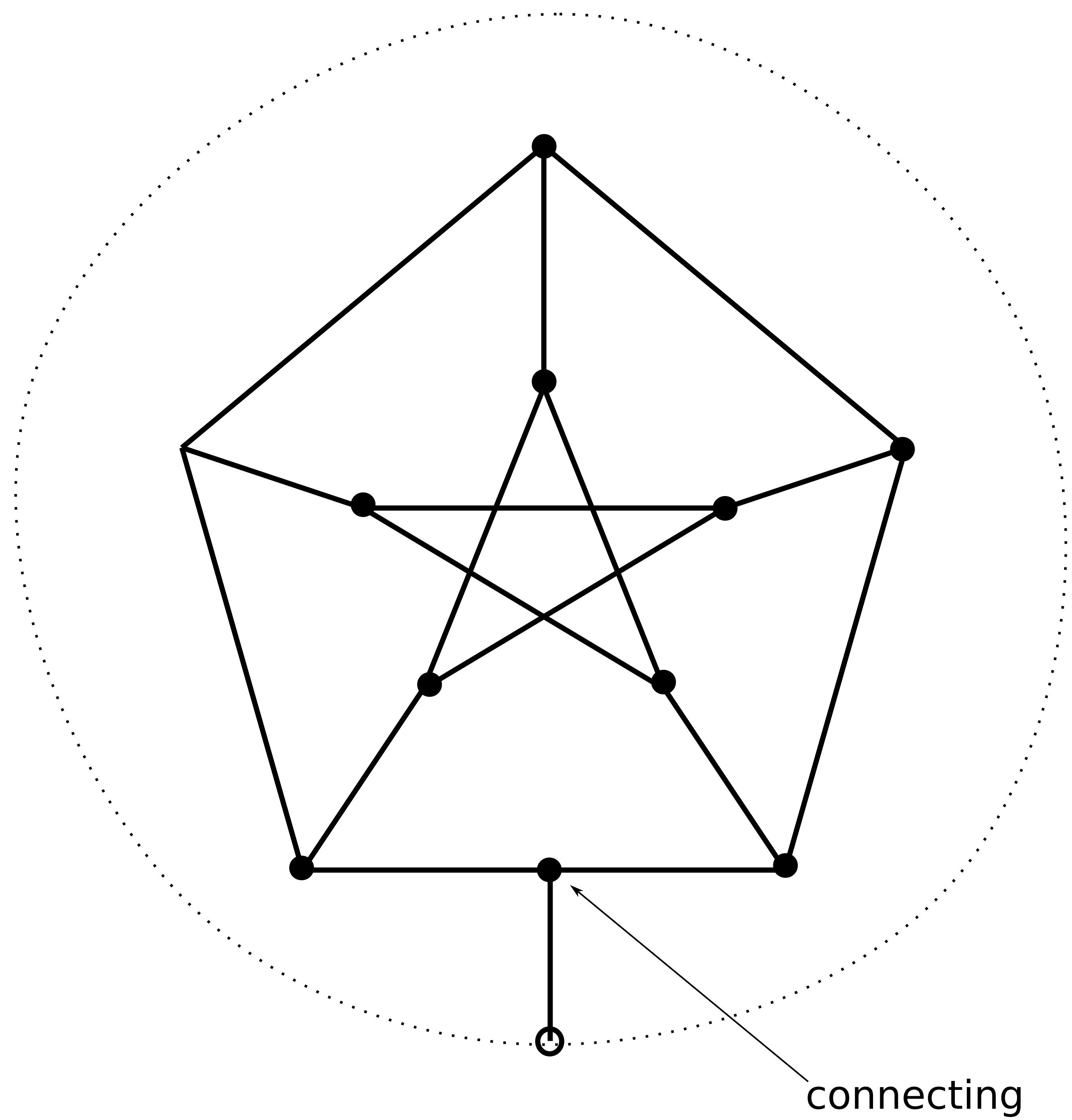
abstraction

variable

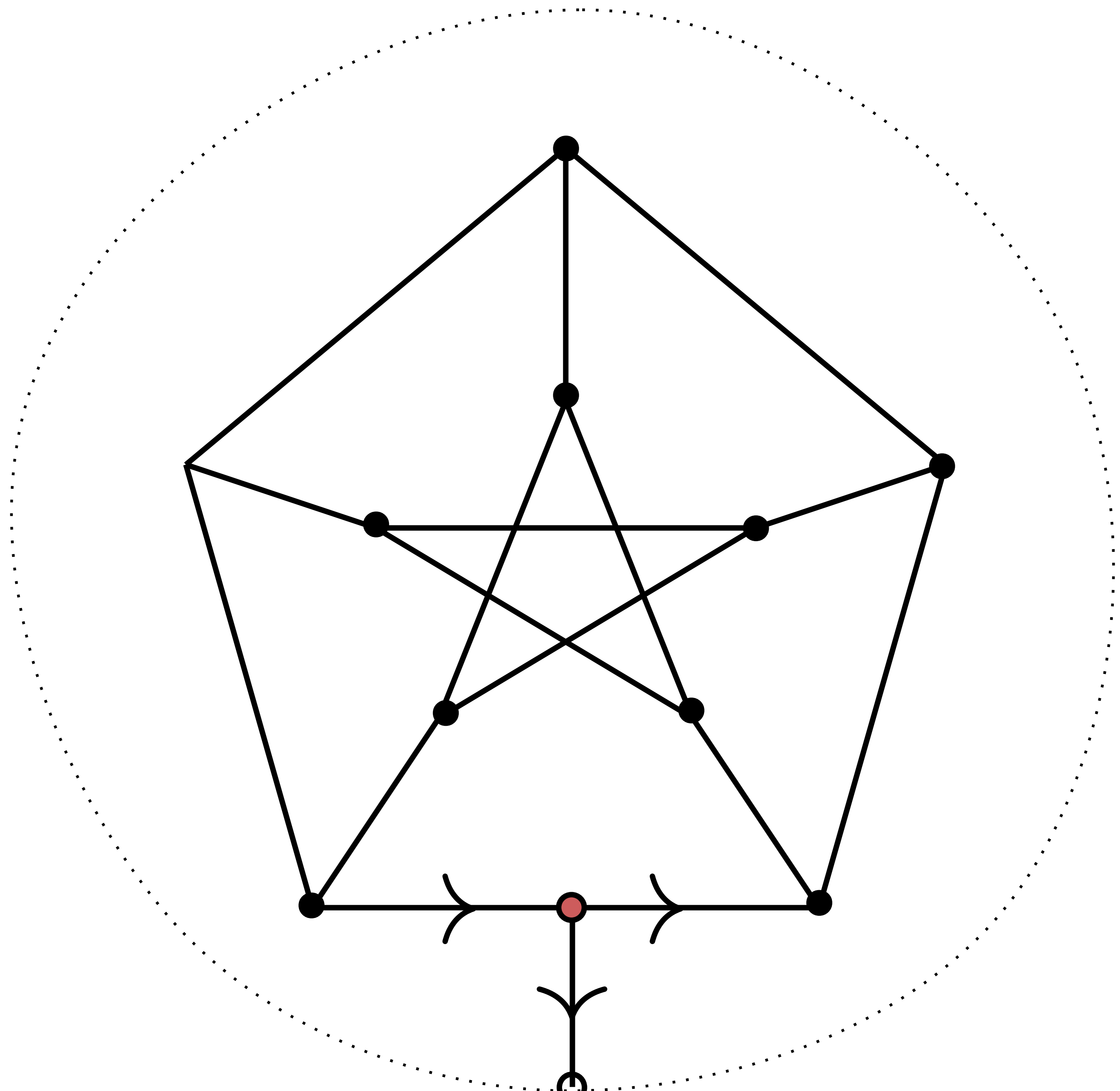
An example



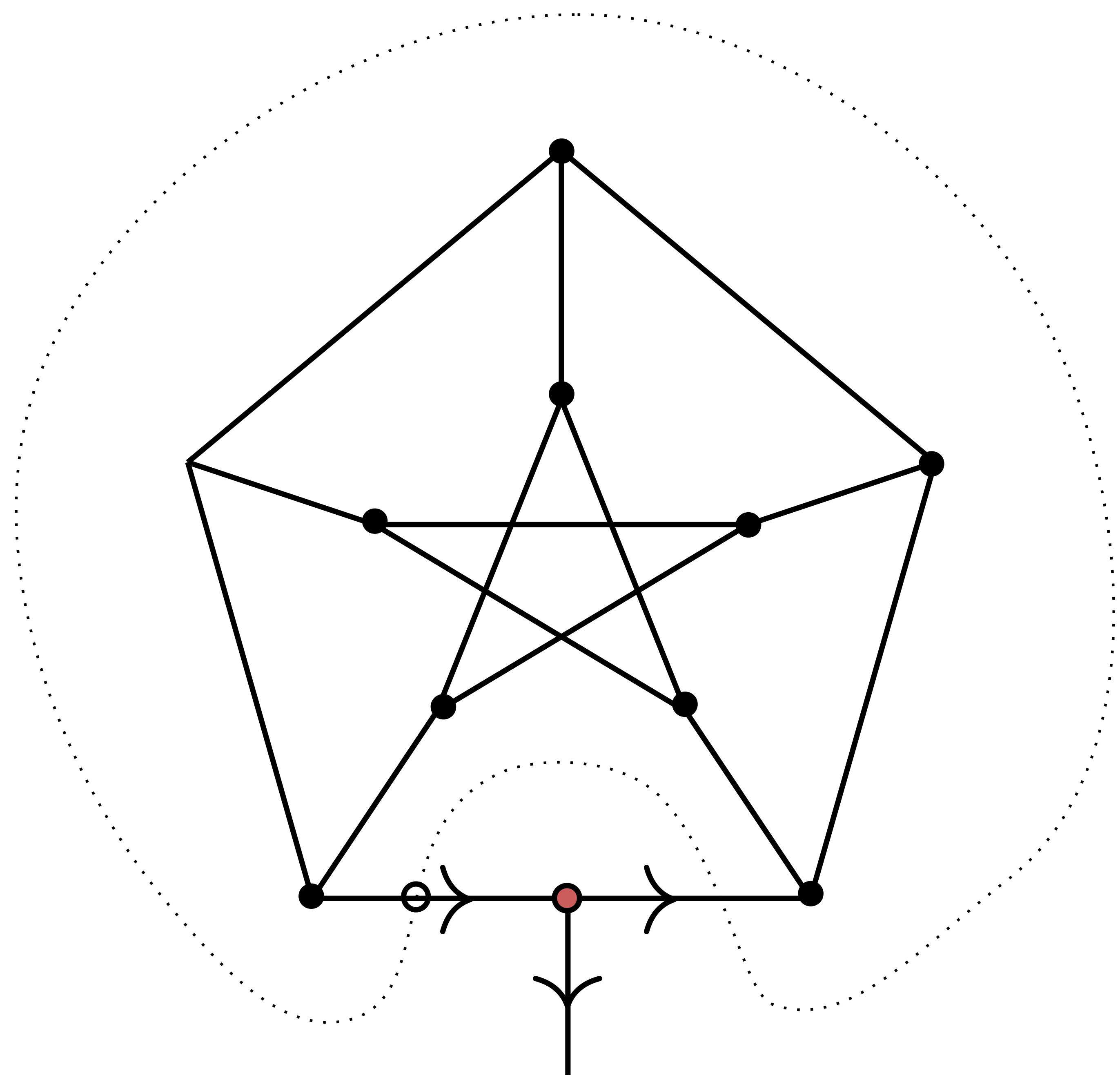
An example



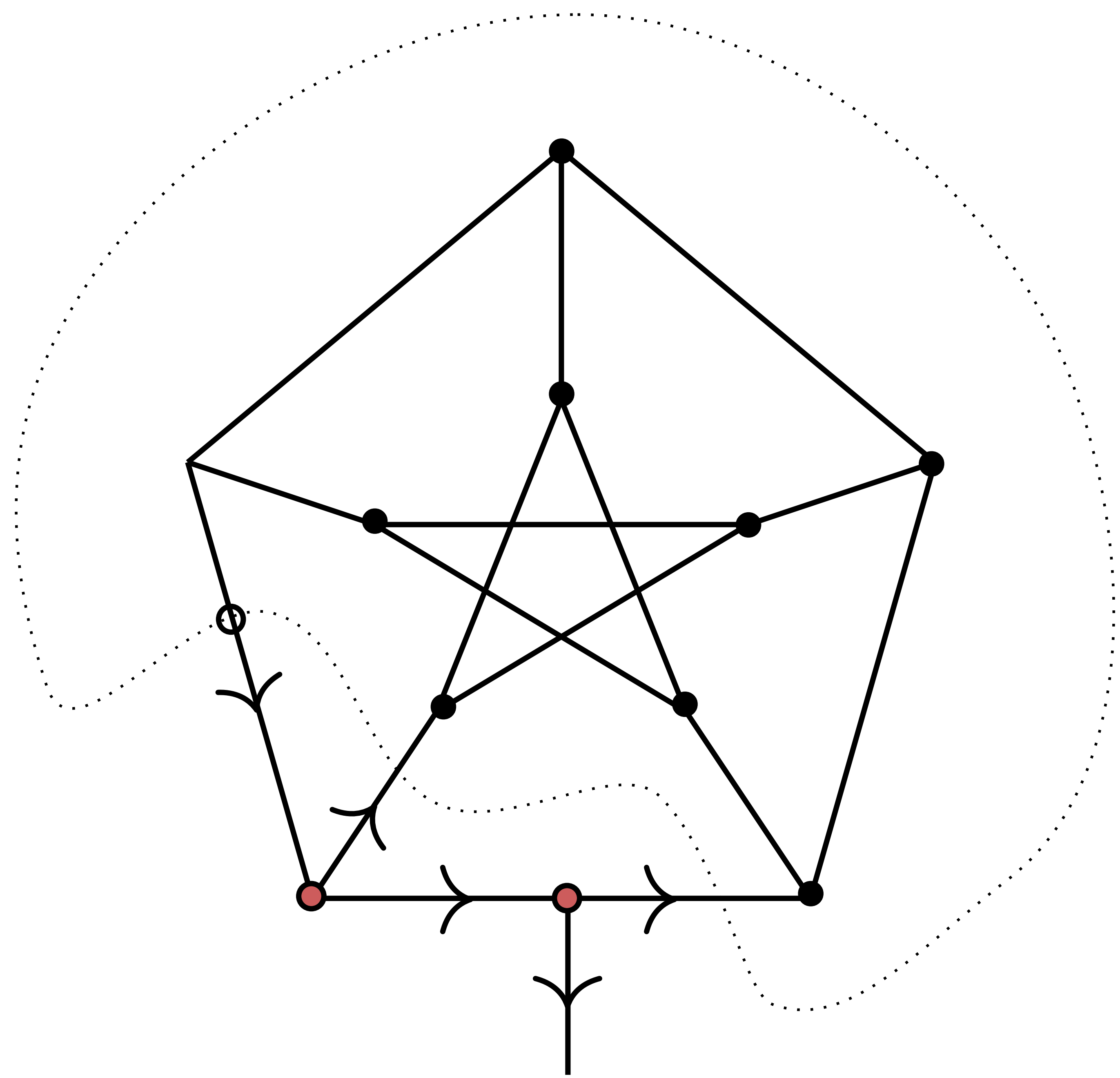
An example



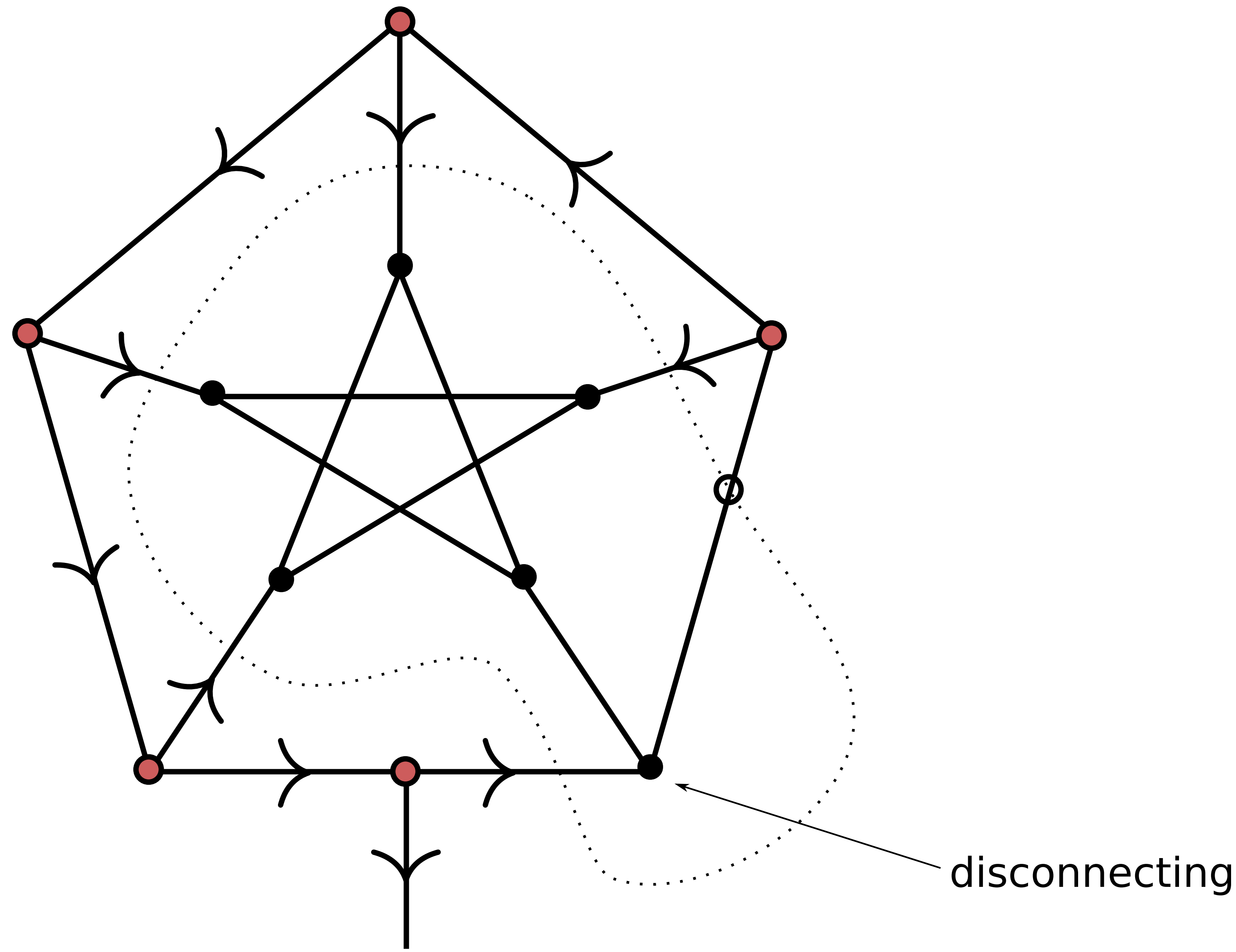
An example



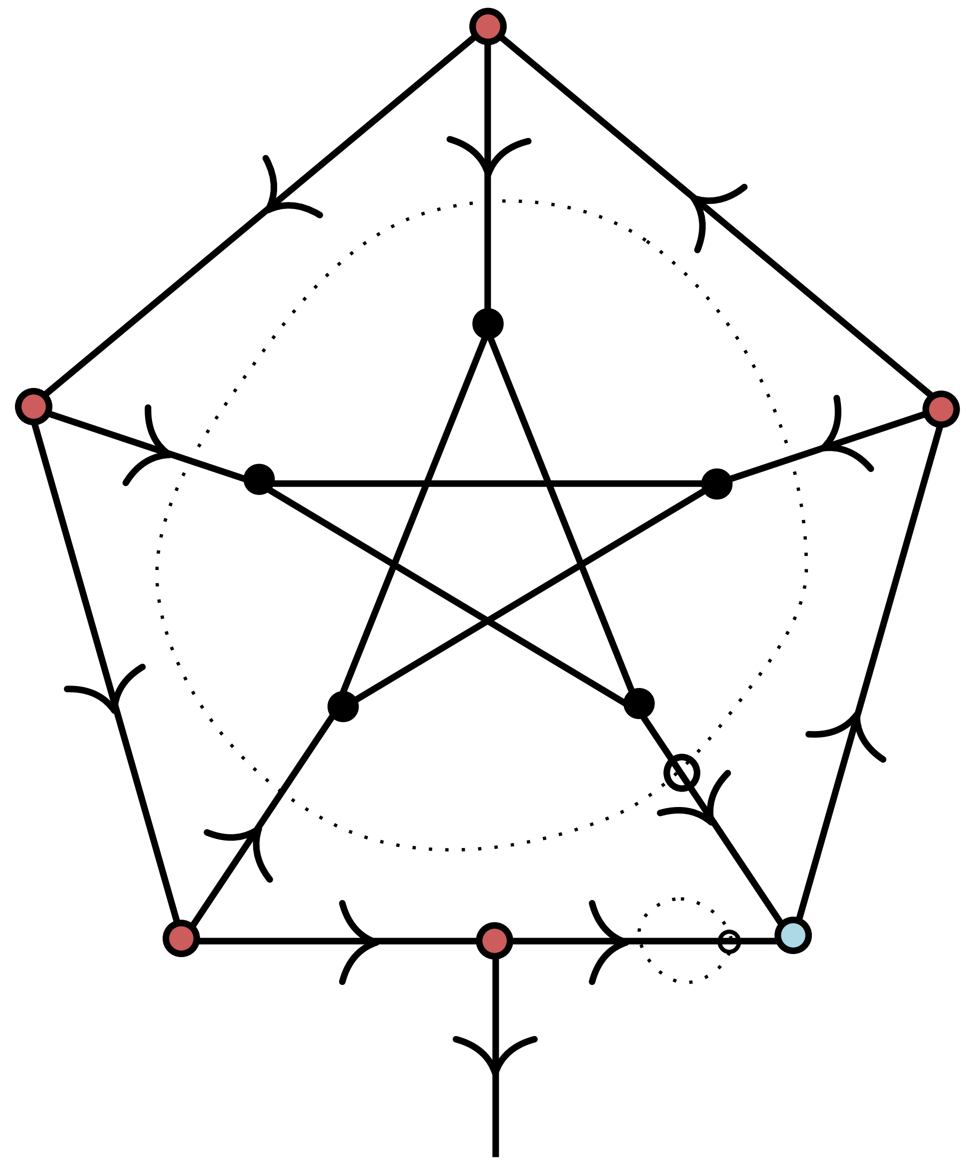
An example



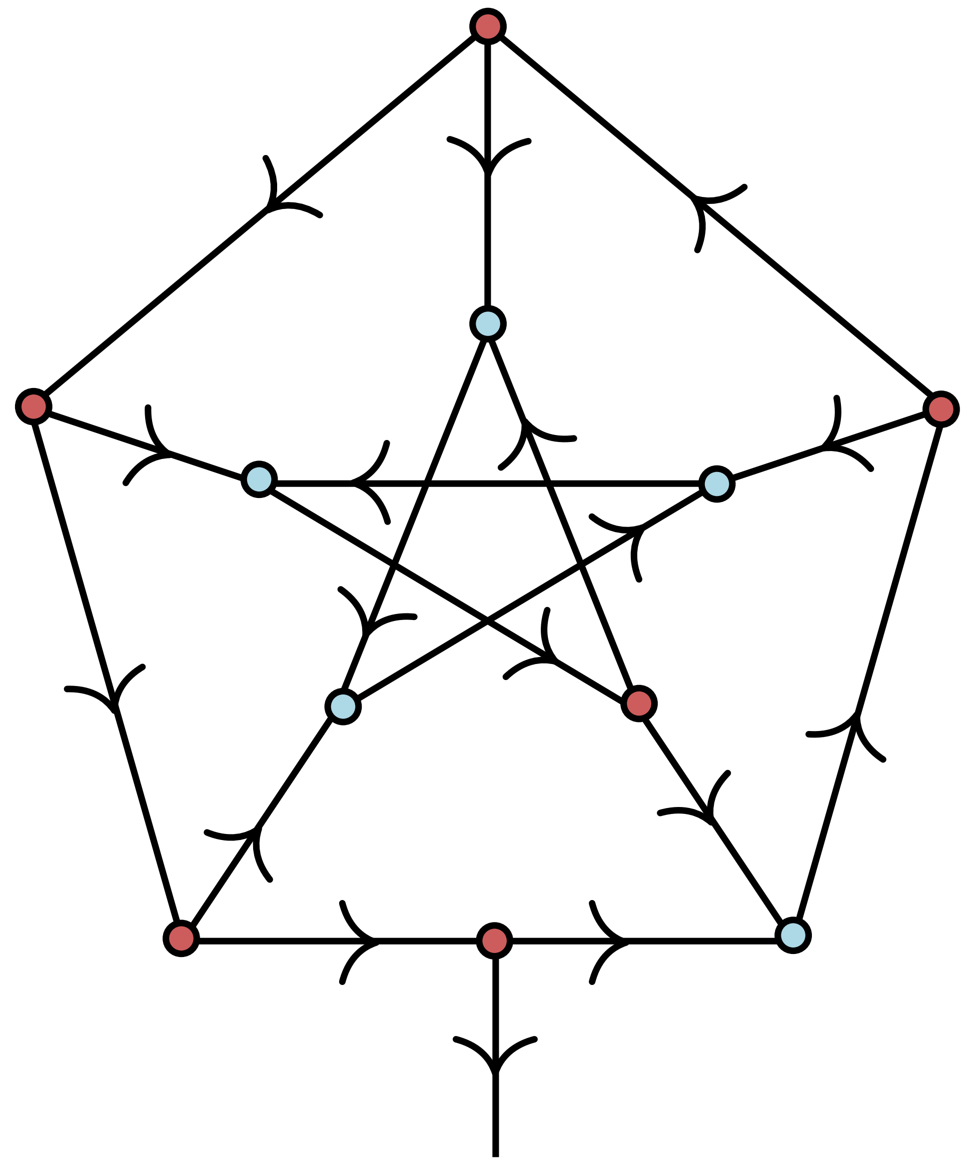
An example



An example



An example



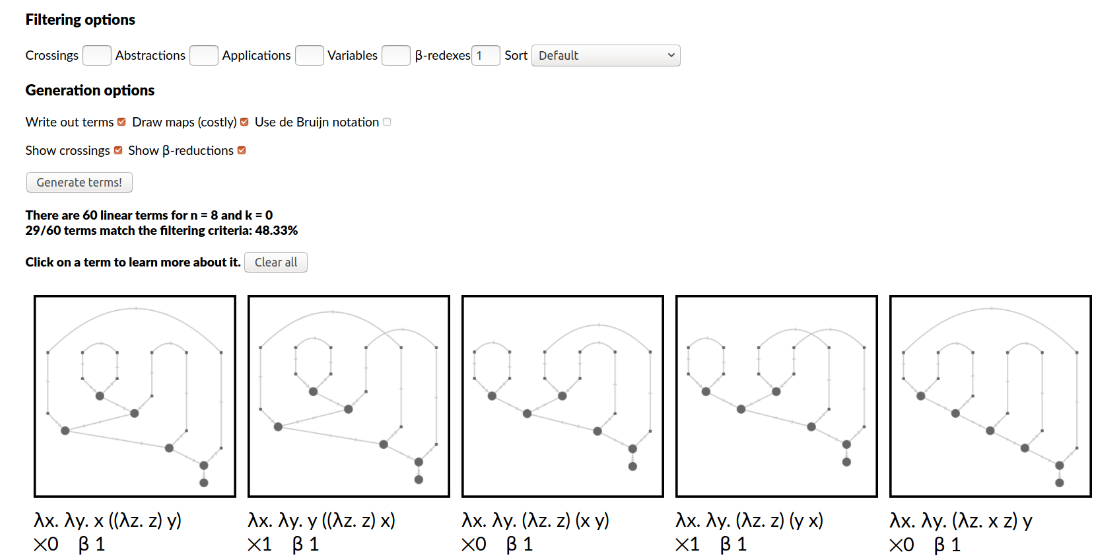
λa.λb.λc.λd.λe.a(λf.c(e(b(df))))

Some tools for further exploring the bijection

George Kaye's λ-term visualiser and gallery

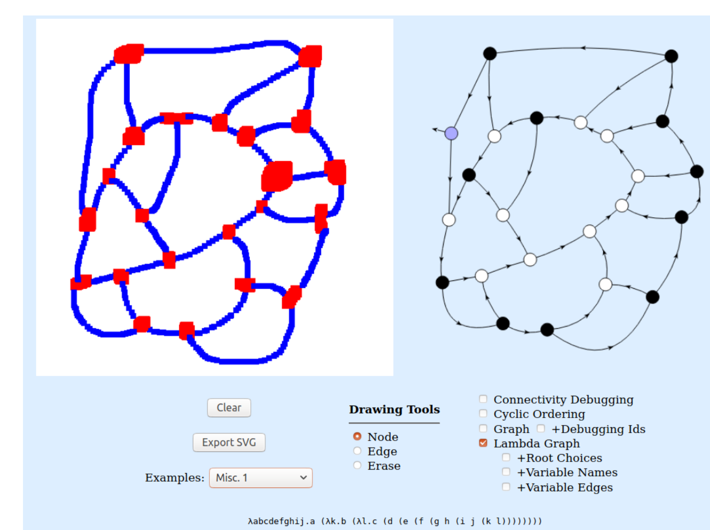
https://www.georgejkaye.com/lambda-visualiser/visualiser.html

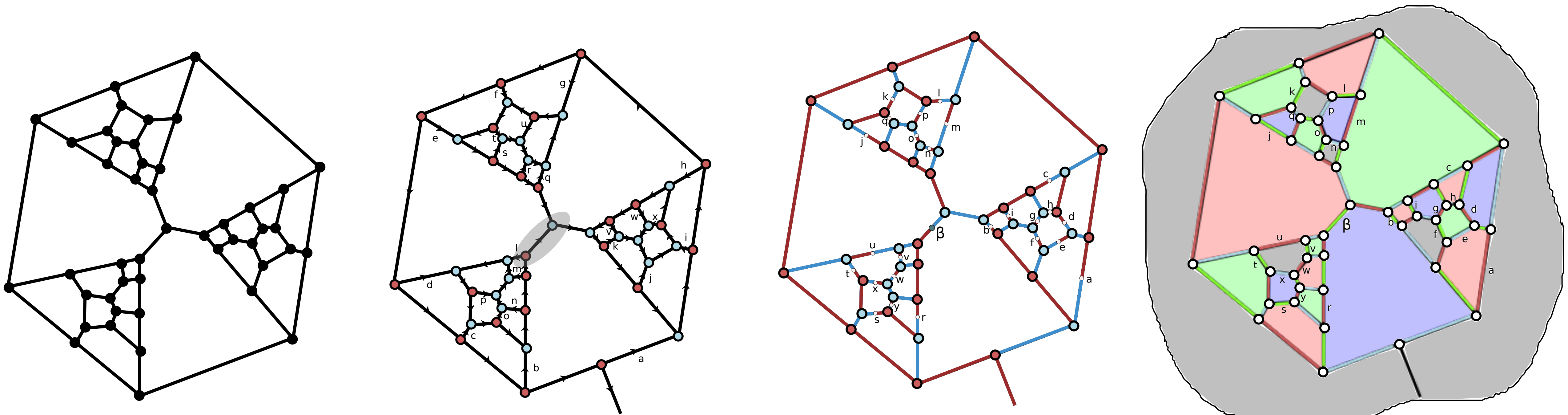
https://www.georgejkaye.com/lambda-visualiser/gallery



Jason Reed's Interactive Lambda Maps Toy

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



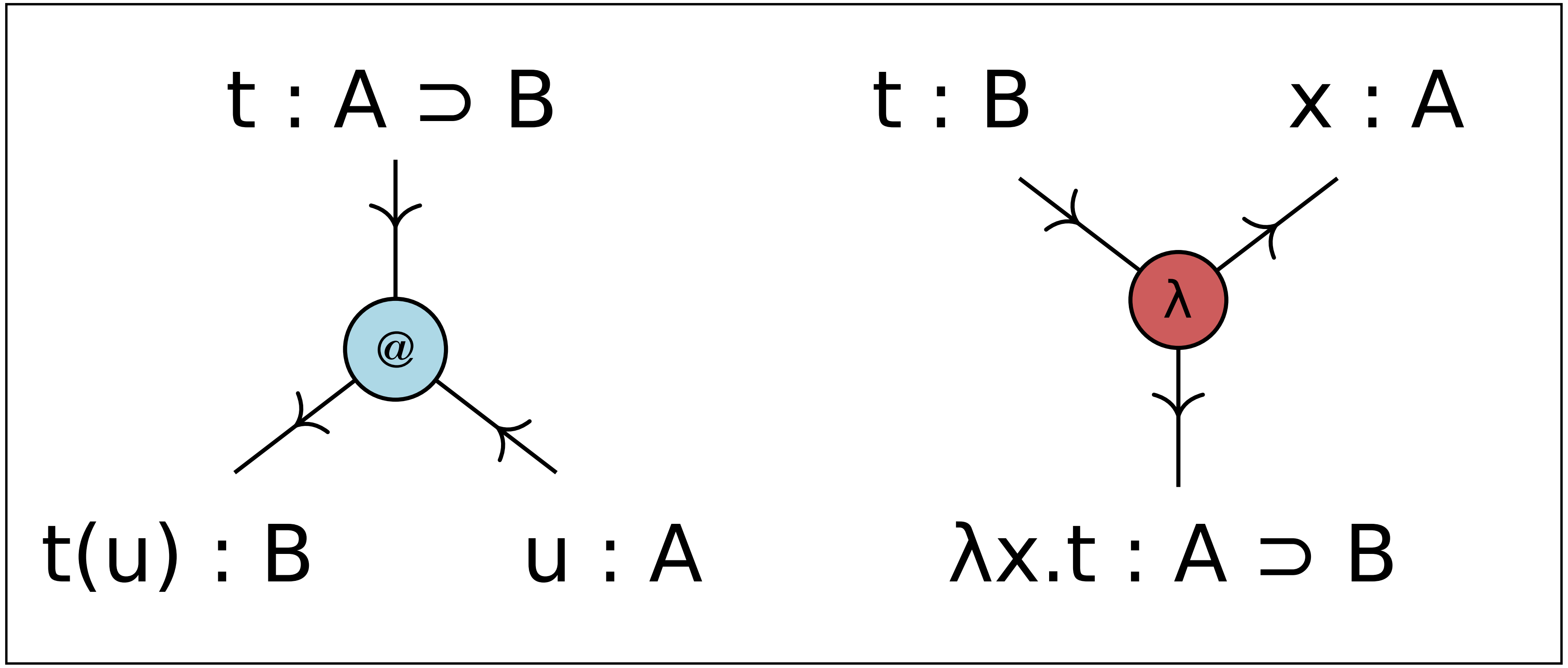


3-valent map — linear lambda term — principal typing — 𝕍-typing

5. From Lambda Calculus to the Four Color Theorem

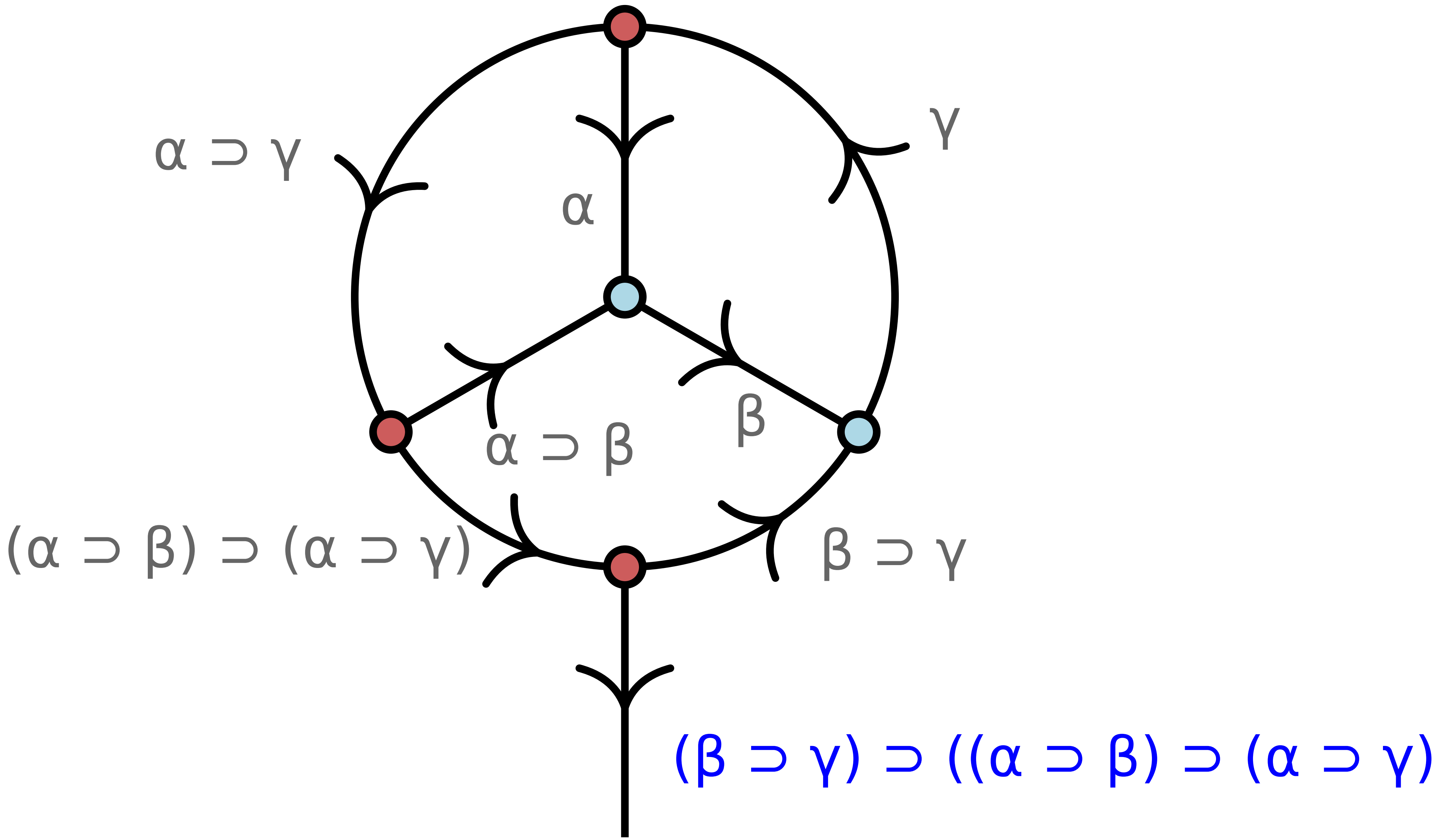
Typing as edge-coloring

Under this correspondence, typing is just an *edge-coloring problem*...

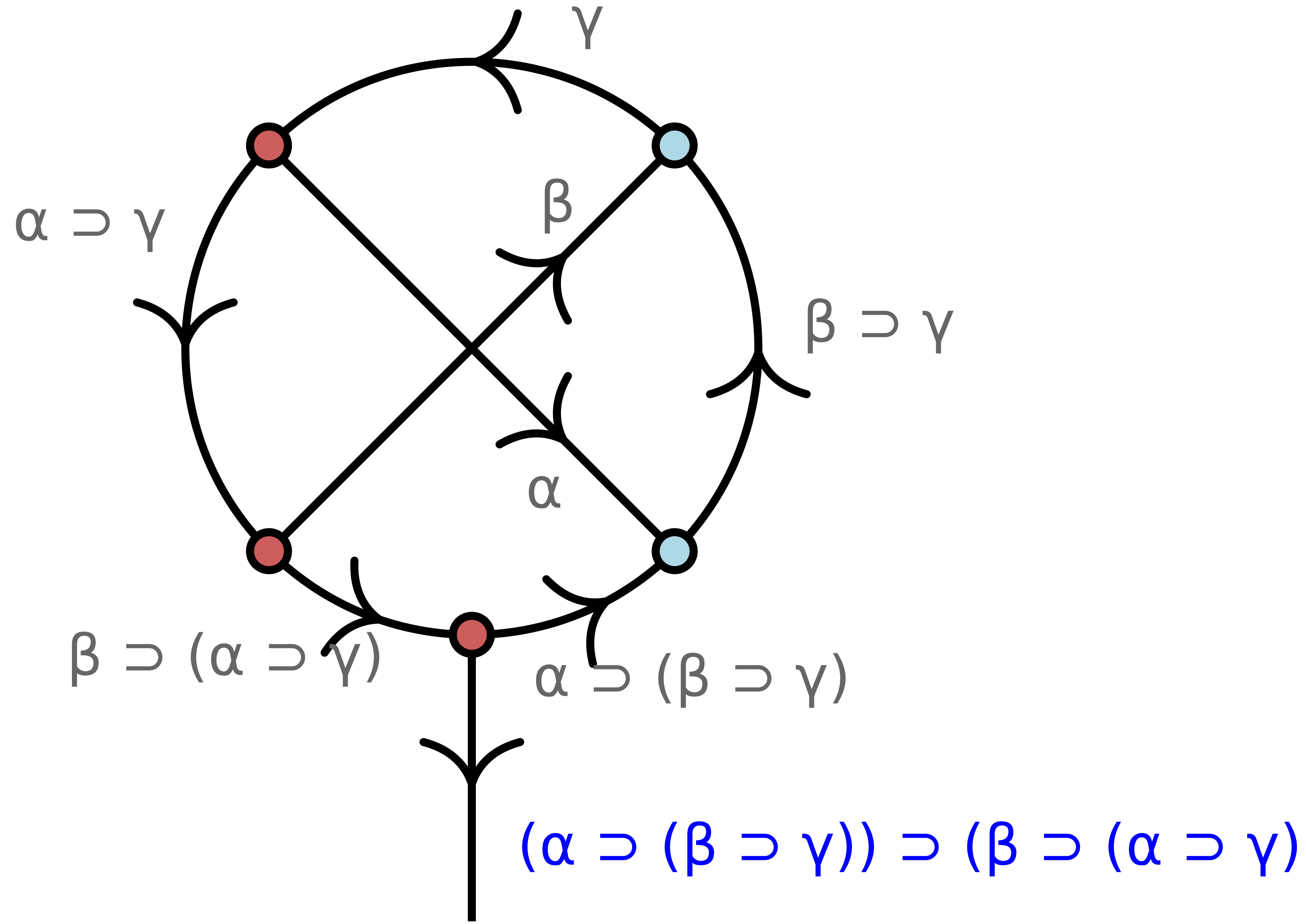


...for a liberal notion of "color"

Typing as edge-coloring



Typing as edge-coloring

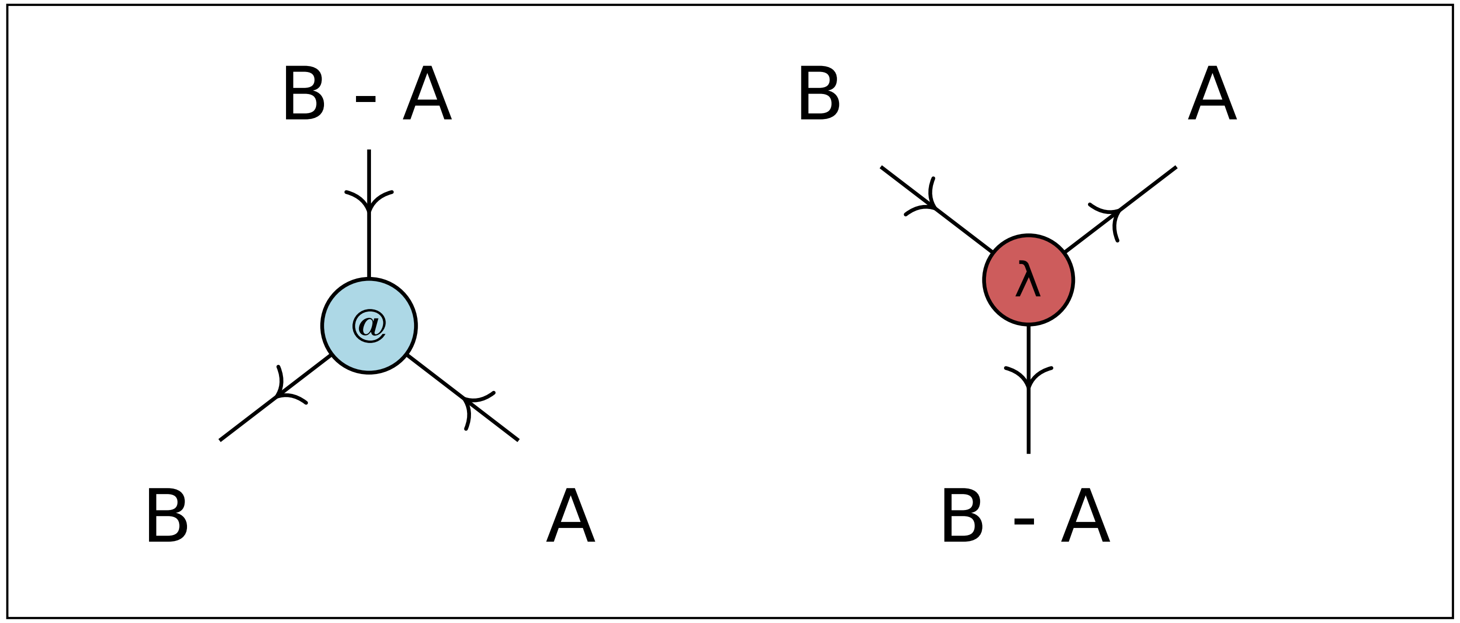


Typing in a group

Abstractly, types can be drawn from any algebraic structure satisfying the laws of linear implication (essentially, the BCI axioms).

In particular, an abelian group! $A \supset B := -A + B$

In this case, a typing of a linear λ-term is the same thing as a *flow** over the corresponding 3-valent graph.



*Tutte, "A contribution to the theory of chromatic polynomials", 1954

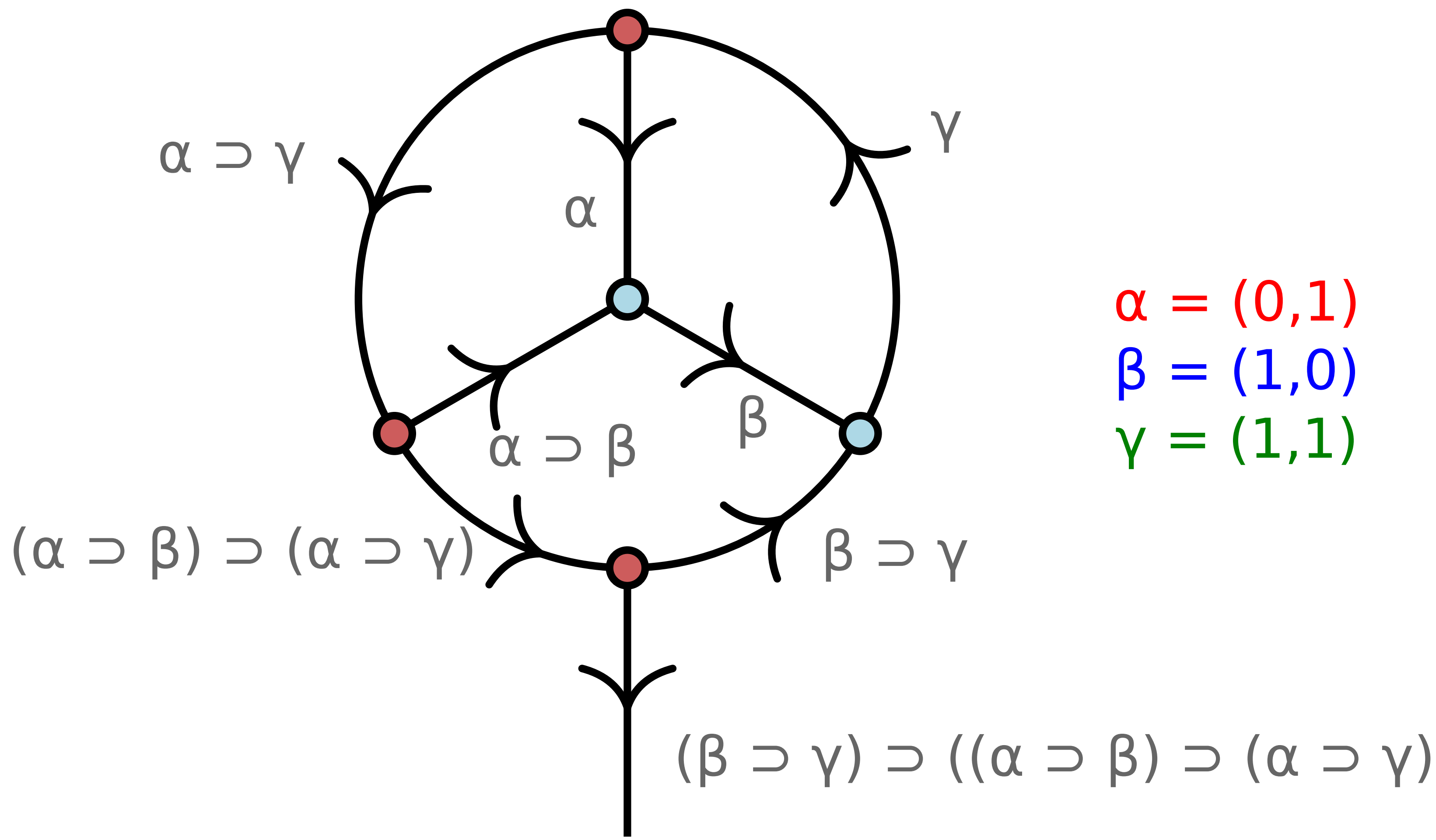
Exercise

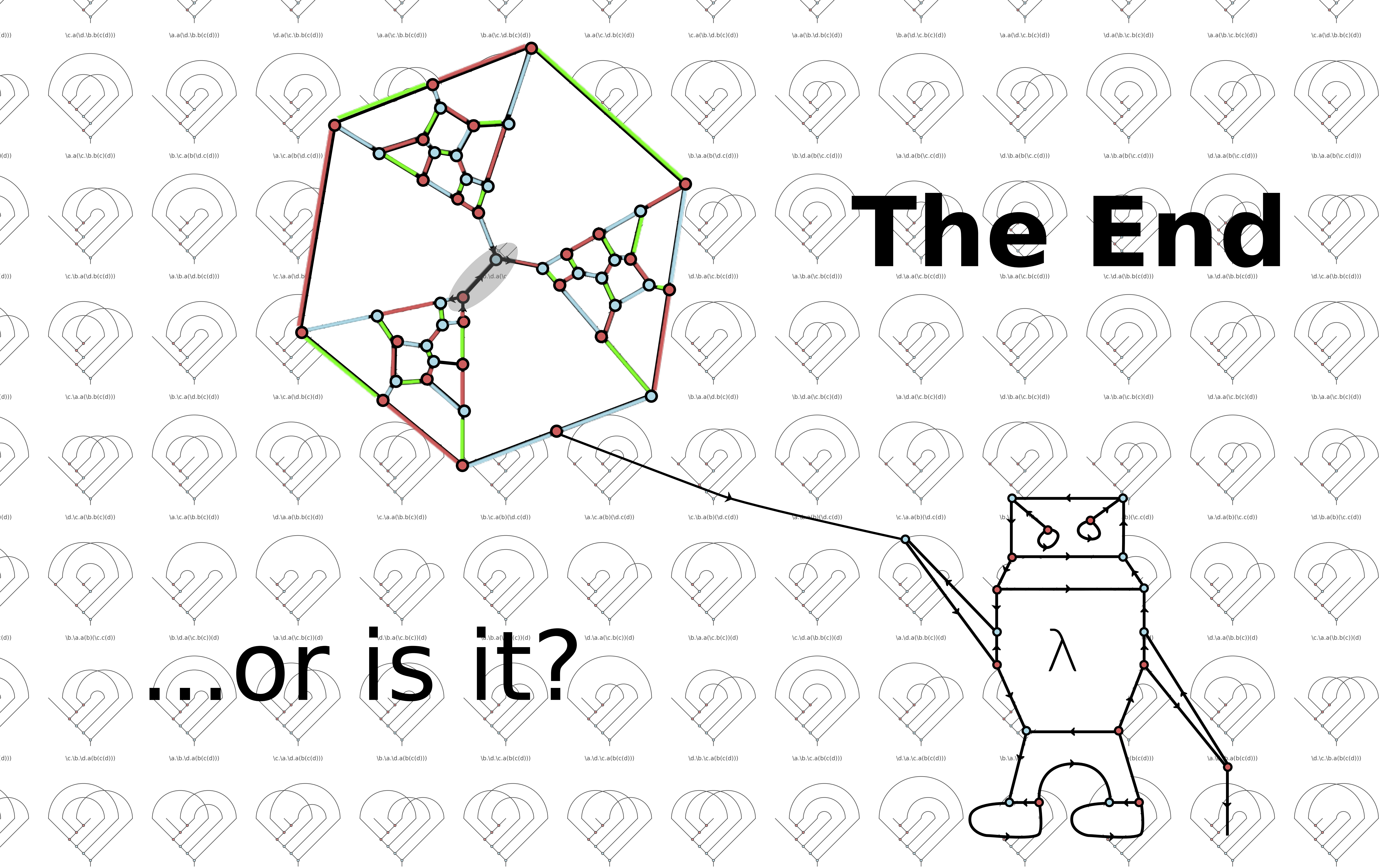
Let $\mathbb{V} = \mathbb{Z}_2 \times \mathbb{Z}_2$ be the Klein Four Group.

Prove: every ordered linear λ-term with no closed subterms has a $\mathbb{V}$-typing such that no subterm gets the type (0,0).

Exercise

Example:





The End
…or is it?
λ