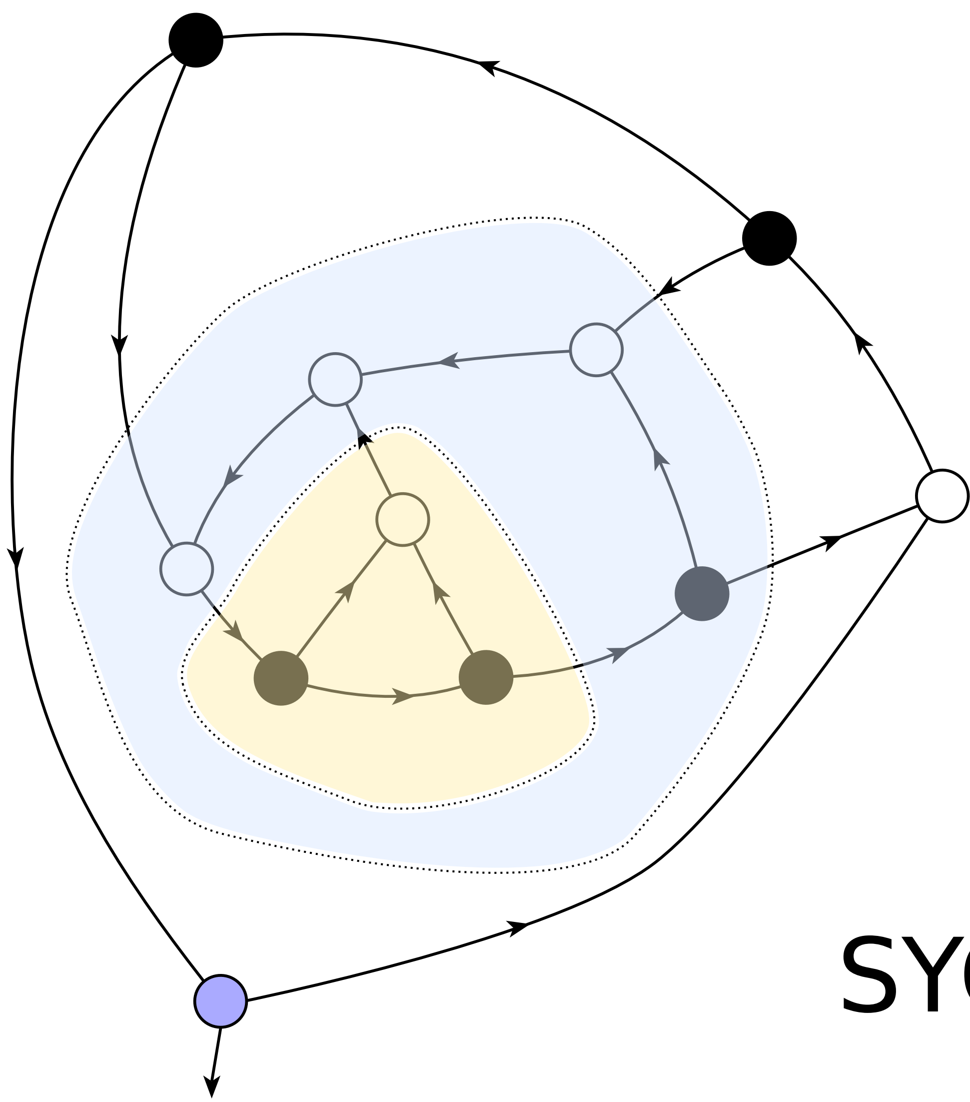
Higher connectivity in linear λ-terms as 3-valent graphs

Noam Zeilberger

an update on
work-in-progress w/Jason Reed

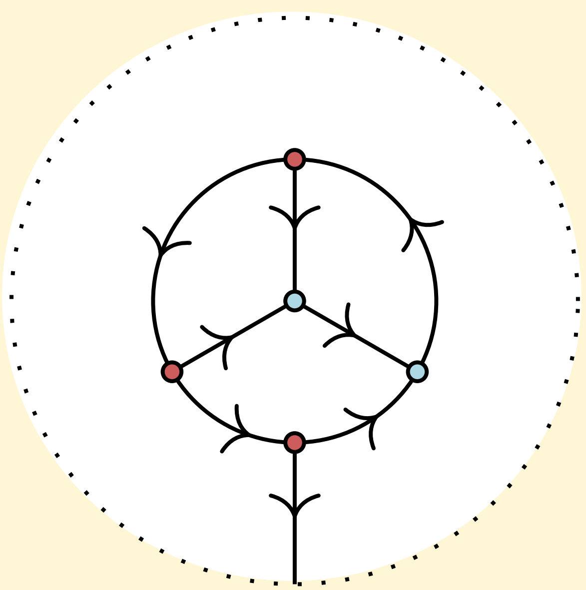
also showcasing some tools
by George Kaye



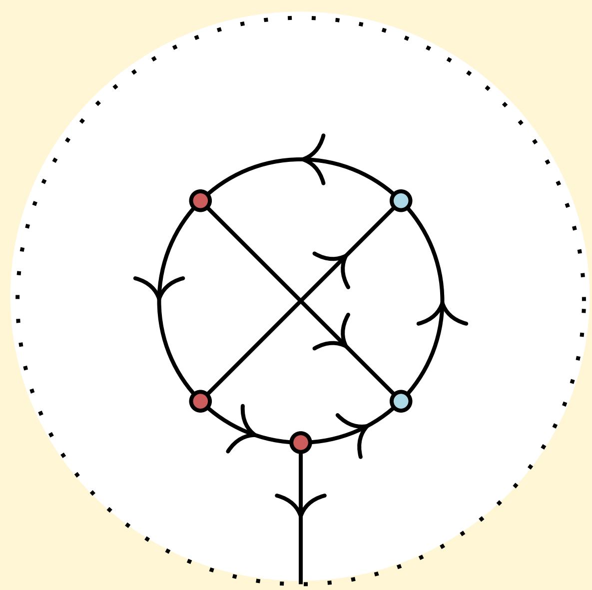
SYCO 5 @ bham!
4-sep-2019

[Background]

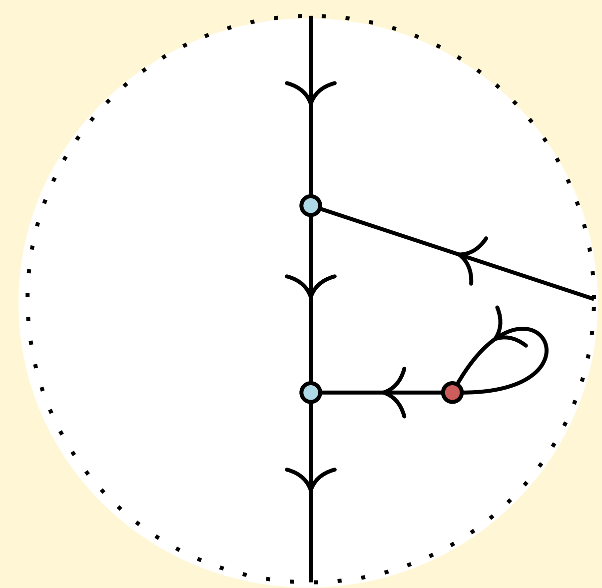
A few views on linear & planar λ-calculus



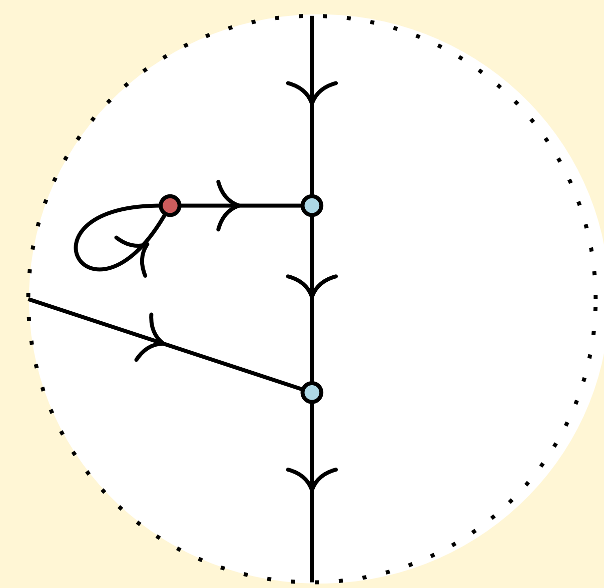
λx.λy.λz.x(yz)



λx.λy.λz.(xz)y



x,y ⊢ (xy)(λz.z)



x,y ⊢ x((λz.z)y)

Classical lambda calculus

Raw syntax:

$$t ::= x \quad | \quad t_1\ t_2 \quad | \quad \lambda x.t_1$$

variable | application | abstraction

Rewriting rules:

$$(\lambda x.t_1)\ t_2 \to_\beta t_1[t_2/x]$$

$$t \to_\eta \lambda x.(t\ x)$$

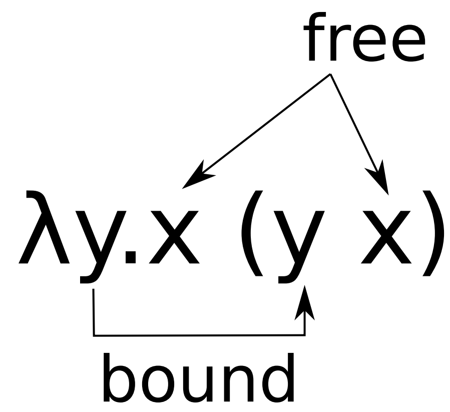
α-equivalence: names are just placeholders

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

Linear lambda calculus

An abstraction λx.t is said to **bind** the occurrences of x in t

A variable which is not bound by any λ is said to be **free**



A term is called **linear** if every free or bound variable occurs exactly once

λx.λy.λz.x (y z)
λx.λy.λz.(x z) y

linear!

λx.λy.x (y x)
λx.λy.x

non-linear!

Fun fact: β-normalization of linear terms is PTIME-complete (Mairson 2004)

Planar lambda calculus

(cf. Abramsky, "Temperley-Lieb Algebra: From Knot Theory to Logic & Computation via QM")

A (closed) linear term is called **ordered** (or **planar**) if every variable is used in the order it is bound...

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

ordered! *non-ordered!*

(The reason why ordered=planar will become clear later.)

Open problem: how hard is β-normalization of ordered linear terms?

Linear lambda calculus, take #2

(cf. Hyland, "Classical lambda calculus in modern dress")

Untyped linear terms may be naturally organized into a *symmetric operad*

- $\Lambda(n)$ = set of α-equivalence classes of linear terms in context $x_1,\ldots,x_n \vdash t$

$$\frac{}{x \vdash x} \qquad \frac{\Gamma \vdash t_1 \quad \Delta \vdash t_2}{\Gamma, \Delta \vdash t_1\, t_2} \qquad \frac{\Gamma, x \vdash t_1}{\Gamma \vdash \lambda x.t_1}$$

- $\circ_i : \Lambda(m+1) \times \Lambda(n) \to \Lambda(m+n)$ defined by (linear) substitution

$$\frac{\Theta \vdash t_2 \quad \Gamma,x,\Delta \vdash t_1}{\Gamma,\Theta,\Delta \vdash t_1[t_2/x]}$$

- symmetric action $S_n \times \Lambda(n) \to \Lambda(n)$ defined by permuting the context

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

Untyped ordered terms form a plain operad: just drop the symmetric action

Linear lambda calculus, take #3

(cf. Lambek, "Deductive systems and categories")

Typed linear terms modulo βη may also be seen as a presentation of the *free closed symmetric multicategory* over a set of atomic types

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

(A multicategory M is **closed** if for any pair of objects A,B there is a binary map

$$A \multimap B, A \xrightarrow{\text{eval}} B$$

together with a family of bijections on multi-hom-sets

$$\lambda : M(\Gamma, A ; B) \cong M(\Gamma ; A \multimap B)$$

whose inverse is the operation of post-composition with eval.)

Linear lambda calculus, take #4

(cf. Scott, "Relating theories of the λ-calculus")

Combining takes #2 and #3, untyped linear terms may be interpreted as *endomorphisms of a reflexive object* in a closed symmetric (2-)multicategory.

By "reflexive object" we mean (with a bit of ambiguity) an object U equipped with an isomorphism/section/adjunction to its space of endomorphisms:

$$U \underset{\text{lam}}{\overset{\text{app}}{\rightleftarrows}} U \multimap U$$

With the most liberal definition, the 2-cells app ∘ lam ⇒ id and id ⇒ lam ∘ app model β-reduction and η-expansion.

From reflexive objects to HOAS

Representation of untyped terms using higher-order abstract syntax (in Twelf):

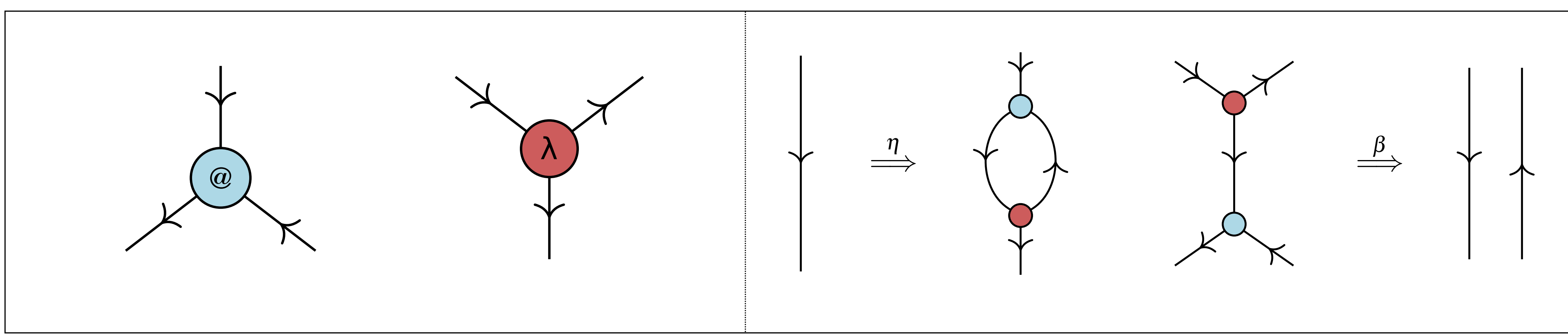
```
u : type.

app : u -> (u -> u).
lam : (u -> u) -> u.

t1 : u = lam [x] lam [y] lam [z] app x (app y z).
t2 : u = lam [x] lam [y] lam [z] app (app x z) y.
t3 : u -> u -> u = [x] [y] app (app x y) (lam [z] z).
t4 : u = lam [x] lam [y] app x (app y x).
t5 : u -> u = [x] lam [y] x.
```

From reflexive objects to string diagrams

A *compact closed* category is a particular kind of closed category in which

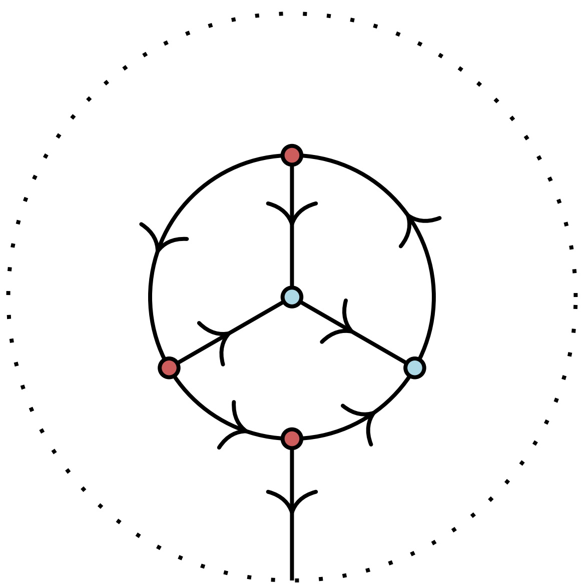
$$A \multimap B \approx B \otimes A^*.$$

By interpreting reflexive objects in the graphical language of compact closed (2-)categories, we derive a graphical representation for linear terms.

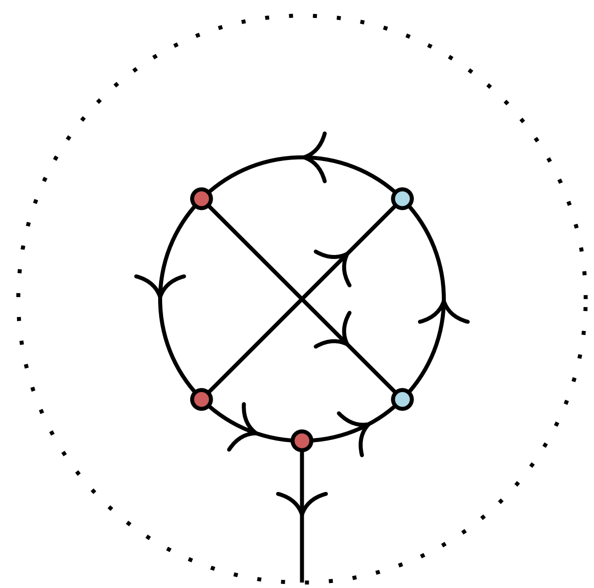


From reflexive objects to string diagrams

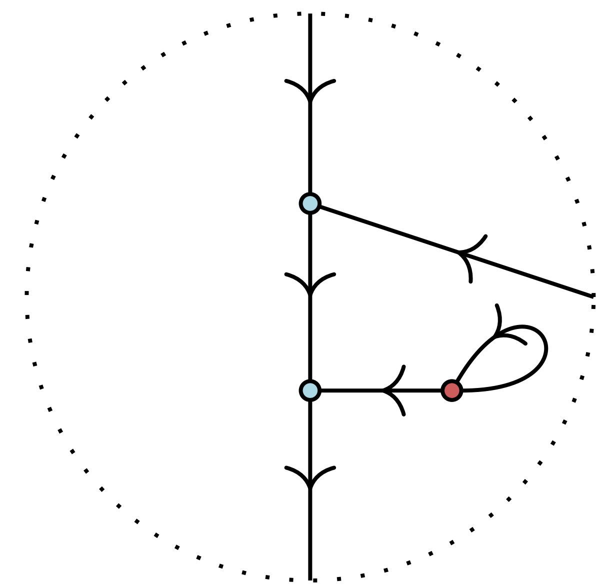
Some examples:



lam [x] lam [y] lam [z] app x (app y z)



lam [x] lam [y] lam [z] app (app x z) y



[x] [y] app (app x y) (lam [z] z)

To play more with these kinds of diagrams, try:

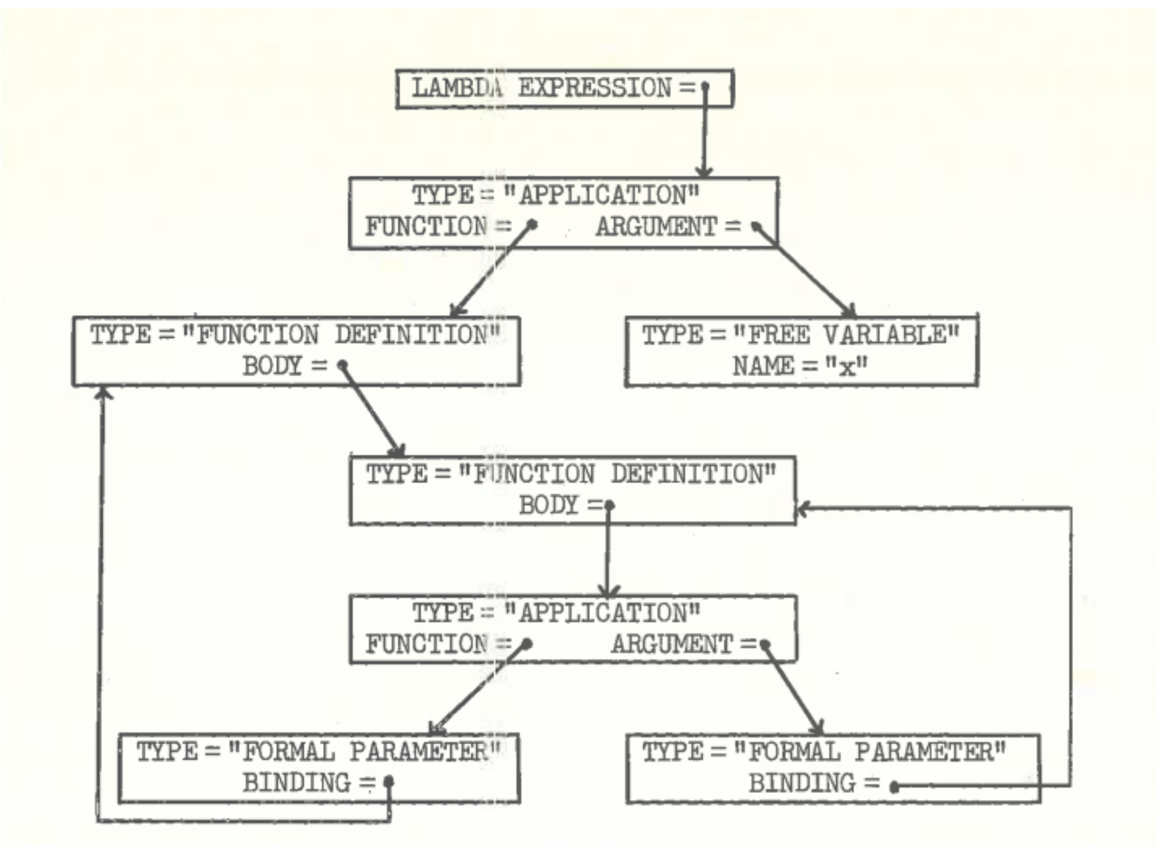
https://www.georgejkaye.com/fyp/visualiser.html

https://www.georgejkaye.com/fyp/gallery

An idea from the folklore

Representing λ-terms this way is an old idea (just under different names)...

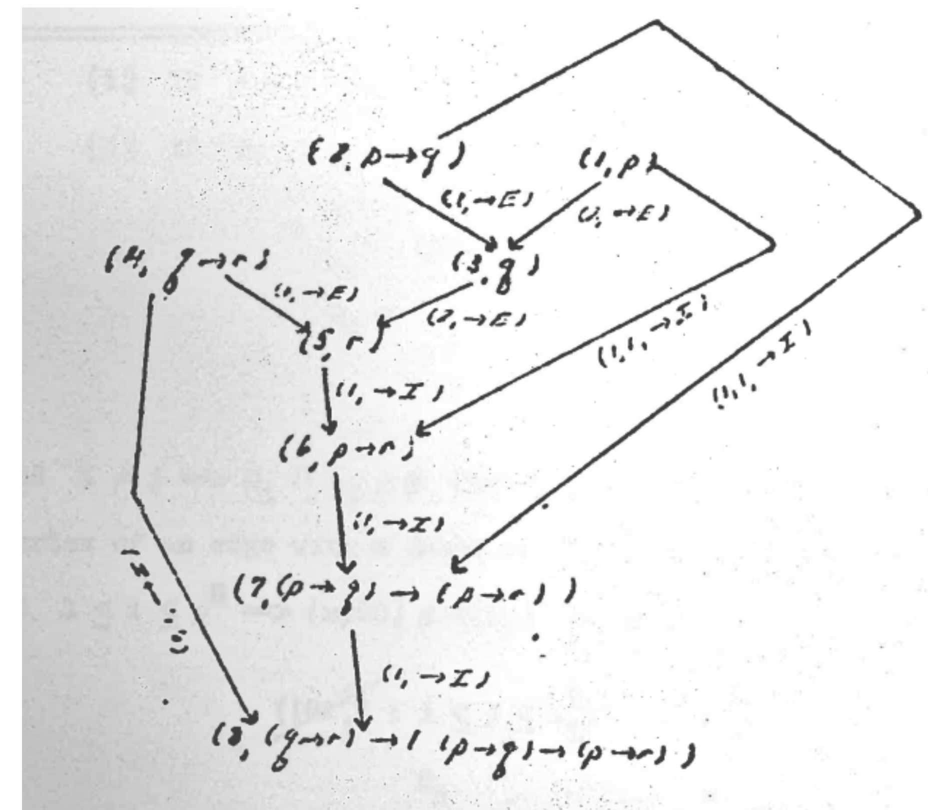
Knuth (1970), "Examples of Formal Semantics"



corresponding HOAS:

[x] app (lam [y] lam [z] app y z) x

Statman (1974), "Structural complexity of proofs"



corresponding HOAS:

lam [x] lam [y] lam [z] app x (app y z)

The surprising combinatorics of linear λ-terms

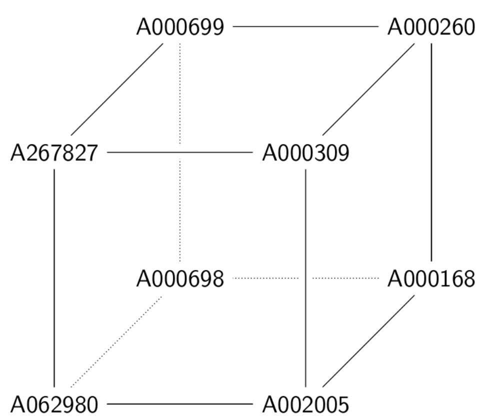
[Background]

1. λa.a
2. (λa.a) (λb.b)
3. λa.a (λb.b)
4. λa.(λb.b) a
5. λa.λb.a b
6. λa.λb.b a
7. (λa.a) ((λb.b) (λc.c))
8. (λa.a) (λb.b (λc.c))
9. (λa.a) (λb.(λc.c) b)
10. (λa.a) (λb.λc.b c)
11. (λa.a) (λb.λc.c b)
12. ((λa.a) (λb.b)) (λc.c)
13. (λa.a (λb.b)) (λc.c)
14. [illegible]a.(λb.[illegible] a) (λ[illegible]c)
15. (λa.λb.a b) [illegible]
16. (λa.λb.b a) (λc.c)
17. λa.a ((λb.b) (λc.c))
18. λa.a (λb.b (λc.c))
19. λa.a (λb.(λc.c) b)
20. λa.a (λb.λc.b c)
21. λa.a (λb.λc.c b)
22. λa.(a (λb.b)) (λc.c)
23. λa.((λb.b) a) (λc.c)
24. λa.(λb.a b) (λc.c)
25. λa.(λb.b a) (λc.c)
26. λa.(λb.b) (a (λc.c))
27. λa.(λb.b) ((λc.c) a)
28. λa.(λb.b) (λc.a c)
29. λa.(λb.b) (λc.c a)
30. λa.((λb.b) (λc.c)) a
31. λa.(λb.b (λc.c)) a
32. λa.(λb.(λc.c) b) a
33. λa.(λb.λc.b c) a
34. λa.(λb.λc.c b) a
35. λa.λb.([illegible] b) (λ[illegible].c)
36. λa.λb.([illegible] a) (λ[illegible].c)
37. λa.λb.a (b (λc.c))
38. λa.λb.a ((λc.c) b)
39. λa.λb.a (λc.b c)
40. λa.λb.a (λc.c b)
41. λa.λb.(a (λc.c)) b
42. λa.λb.((λc.c) a) b
43. λa.λb.(λc.a c) b
44. λa.λb.(λc.c a) b
45. λa.λb.b (a (λc.c))
46. λa.λb.b ((λc.c) a)
47. [illegible]a[illegible]b.b (λ[illegible].a
48. [illegible]a.λb[illegible] (λ[illegible].c[illegible])
49. λa.λb.(b (λc.c)) a
50. λa.λb.((λc.c) b) a
51. λa.λb.(λc.b c) a
52. λa.[illegible]b.[illegible]c[illegible])
53. [illegible]a.λb.(λc.c) ([illegible] b)
54. λa.λb.(λc.c) (b a)
55. λa.λb.λc.(a b) c
56. λa.λb.λc.(b a) c
57. λa.λb.λc.(a c) b
58. λa.λb.λc.(c a) b
59. λa.λb.λc.a (b c)
60. λa.λb.λc.a (c b)
61. λa.λb.λc.(b c) a
62. λa.λb.λc.(c b) a
63. λa.λb.λc.b (a c)
64. λa.λb.λc.b (c a)
65. λa.λb.λc.c (a b)
66. λa.λb.λc.c (b a)
67. (λa.a) ((λb.b) ((λc.c) (λd.d)))
68. [illegible]a[illegible]) ((λ[illegible].b) (λc.[illegible] (λd.[illegible]))
69. [illegible]a.a) ((λ[illegible].b) (λc.(λd.d) [illegible]))
70. (λa.a) ((λb.b) (λc.λd.c d))
71. (λa.a) ((λb.b) (λc.λd.d c))
72. (λa.a) (((λb.b) (λc.c)) (λd.d))
73. (λa.a) ((λb.b (λc.c)) (λd.d))
74. (λa.a) ((λb.(λc.c) b) (λd.d))
75. (λa.a) ((λb.λc.b c) (λd.d))
76. (λa.a) ((λb.λc.c b) (λd.d))
77. (λa.a) (λb.b ((λc.c) (λd.d)))
78. (λa.a) (λb.b (λc.c (λd.d)))
79. (λa.a) (λb.b (λc.(λd.d) [illegible]))
80. [illegible]
81. [illegible]
82. (λa.a) (λb.(b (λc.c)) (λd.d))
83. (λa.a) (λb.((λc.c) b) (λd.d))
84. (λa.a) (λ[illegible](λc.b c) (λd.d))
85. [illegible]
86. [illegible]a.a) (λ[illegible]c.[illegible] ([illegible](λ[illegible]d)))
87. (λa.a) (λb.(λc.c) ((λd.d) b))
88. (λa.a) (λb.(λc.c) (λd.b d))
89. (λa.a) (λb.(λc.c) (λd.d b))
90. (λa.a) (λb.((λc.c) (λd.d)) b)
91. (λa.a) (λb.(λc.c (λd.d)) b)
92. (λa.a) (λb.(λc.(λd.d) c) b)
93. (λa.a) (λb.(λc.λd.c d) b)
94. (λa.a) (λb.(λc.λd.d c) b)
95. (λa.a) (λb.λc.(b c) (λd.d))
96. (λa.a) (λb.λc.(c b) (λd.d))
97. (λa.a) (λb.λc.b (c (λd.d)))
98. (λa.a) (λb.λc.b ((λd.d) c))
99. (λa.a) (λb.λc.b (λd.c d))
100. (λa.a) (λb.λc.b (λd.d c))
101. (λa.a) (λb.λc.(b (λd.d)) c)
102. (λa.a) (λb.λc.((λd.d) b) c)
103. (λa.a) (λb.λc.(λd.b d) c)
104. (λa.a) (λb.λc.(λd.d b) c)
105. (λa.a) (λb.λc.c (b (λd.d)))
106. (λa.a) (λb.λc.c ((λd.d) b))
107. (λa.a) (λb.λc.c (λd.b d))
108. (λa.a) (λb.λc.c (λd.d b))
109. (λa.a) (λb.λc.(c (λd.d)) b)
110. (λa.a) (λb.λc.((λd.d) c) b)
111. (λa.a) (λb.λc.(λd.c d) b)
112. [illegible]a.a) (λb.λc.(λd.d [illegible]b)
113. [illegible]
114. [illegible]
115. (λa.a) (λb.λc.λd.(b c) d)
116. (λa.a) (λb.λc.λd.(c b) d)
117. (λa.a) (λb.λc.λd.(b d) c)
118. (λa.a) (λb.λc.λd.(d b) c)
119. (λa.a) (λb.λc.λd.b (c d))
120. (λa.a) (λb.λc.λd.b (d c))
121. (λa.a) (λb.λc.λd.(c d) b)
122. (λa.a) (λb.λc.λd.(d c) b)
123. (λa.a) (λb.λc.λd.c (b d))
124. (λa.a) (λb.λc.λd.c (d b))
125. (λa.a) (λb.λc.λd.d (b c))
126. (λa.a) (λb.λc.λd.d (c b))
127. ((λa.a) (λb.b)) ((λc.c) (λd.d))
128. ((λa.a) (λb.b)) (λc.c (λd.d))
129. ((λa.a) (λb.b)) (λc.(λd.d) c)
130. ((λa.a) (λb.b)) (λc.λd.c d)
131. ((λa.a) (λb.b)) (λc.λd.d c)
132. (λa.a (λb.b)) ((λc.c) (λd.d))

Some enumerative connections

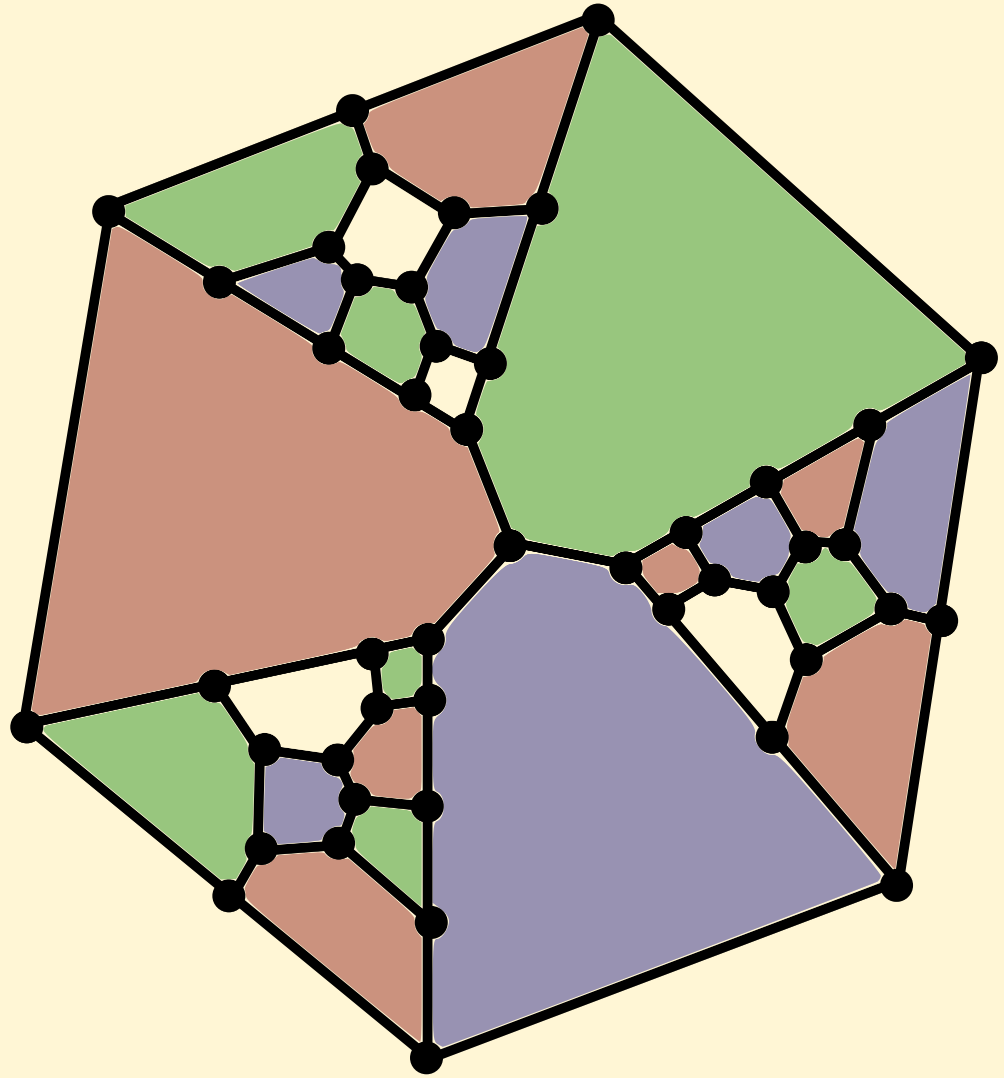
family of rooted maps	family of lambda terms	sequence	OEIS
trivalent maps (genus g≥0)	linear terms	1,5,60,1105,27120,...	A062980
planar trivalent maps	ordered terms	1,4,32,336,4096,...	A002005
bridgeless trivalent maps	unitless linear terms	1,2,20,352,8624,...	A267827
bridgeless planar trivalent maps	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

1. O. Bodini, D. Gardy, A. Jacquot (2013), Asymptotics and random sampling for BCI and BCK lambda terms, TCS 502: 227-238
2. Z, A. Giorgetti (2015), A correspondence between rooted planar maps and normal planar lambda terms, LMCS 11(3:22): 1-39
3. Z (2015), Counting isomorphism classes of beta-normal linear lambda terms, arXiv:1509.07596
4. Z (2016), Linear lambda terms as invariants of rooted trivalent maps, J. Functional Programming 26(e21)
5. J. Courtiel, K. Yeats, Z (2016), Connected chord diagrams and bridgeless maps, arXiv:1611.04611
6. Z (2017), A sequent calculus for a semi-associative law, FSCD



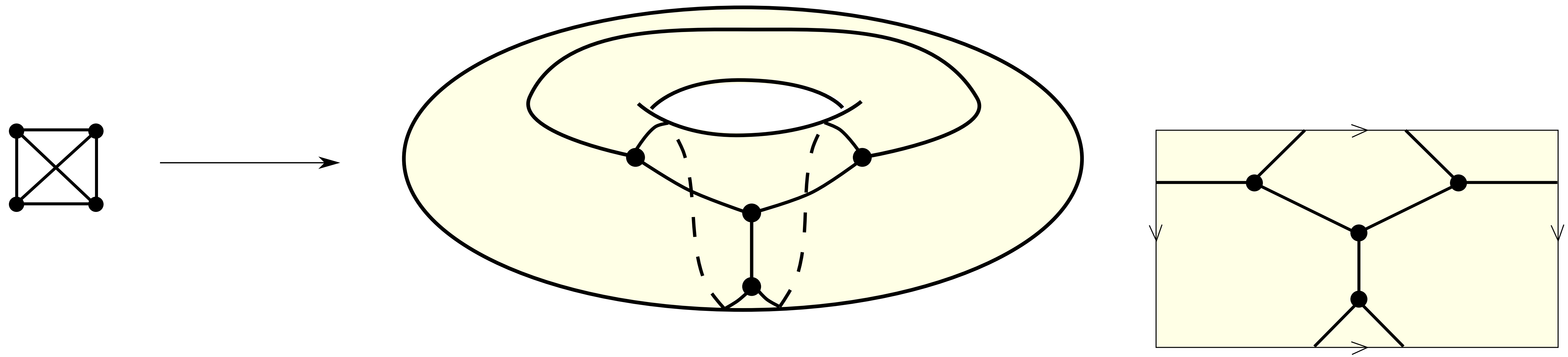
[Background]

A few views on maps



Topological definition

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



considered up to deformation of the underlying surface.

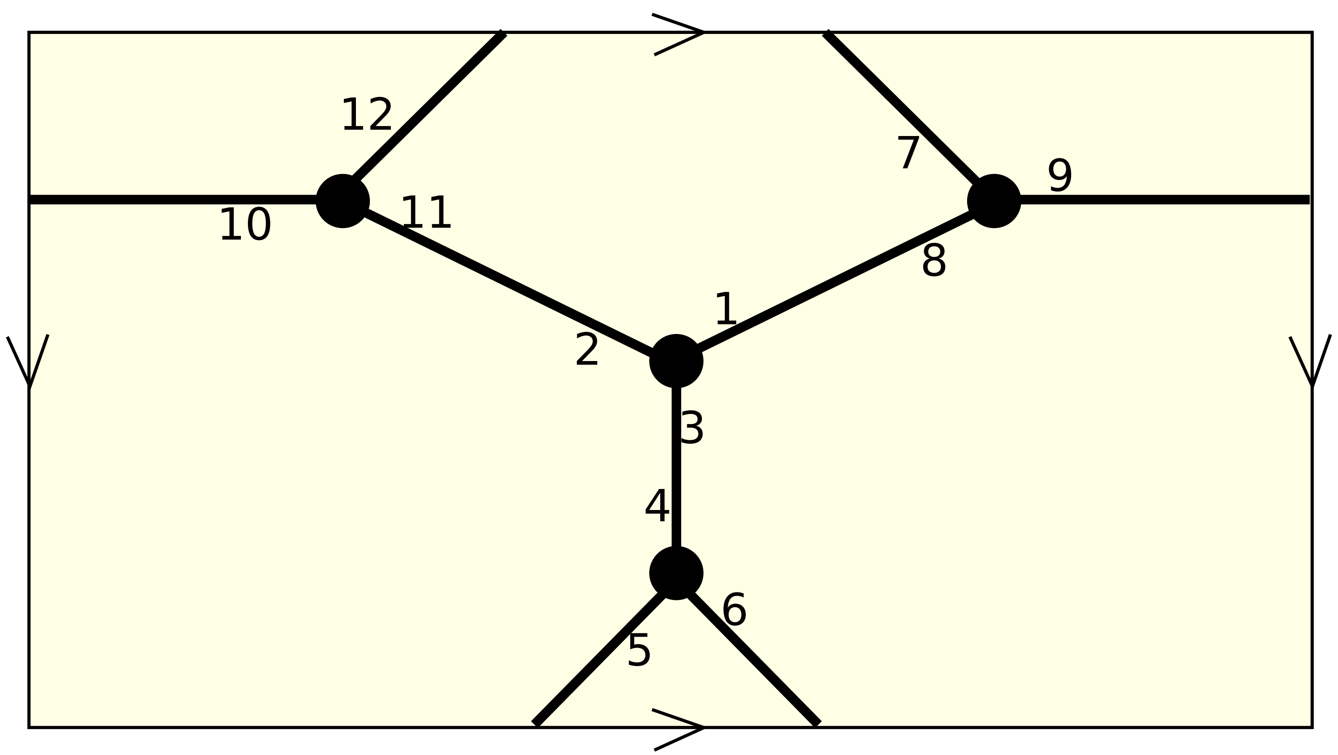
*All surfaces are assumed to be connected and oriented throughout this talk

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.



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

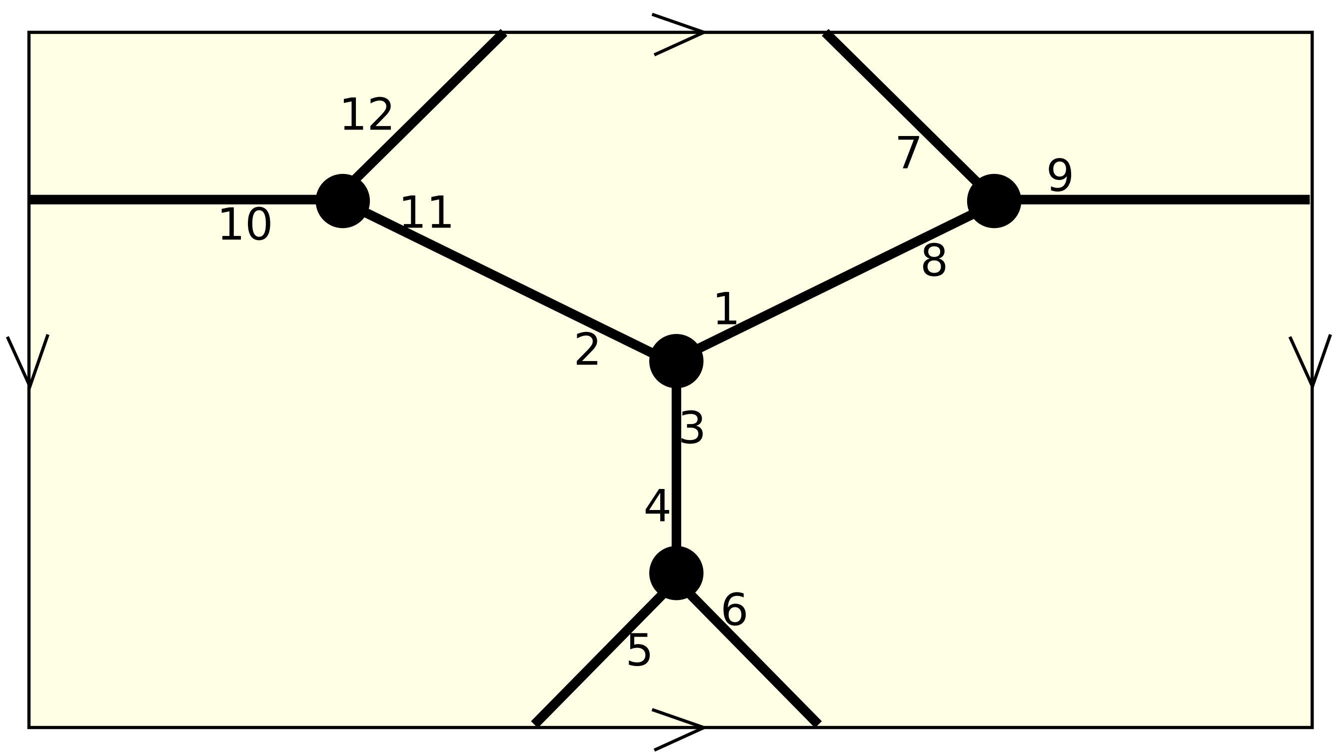
$$e = (1\ 8)(2\ 11)(3\ 4)(5\ 12)(6\ 7)(9\ 10)$$

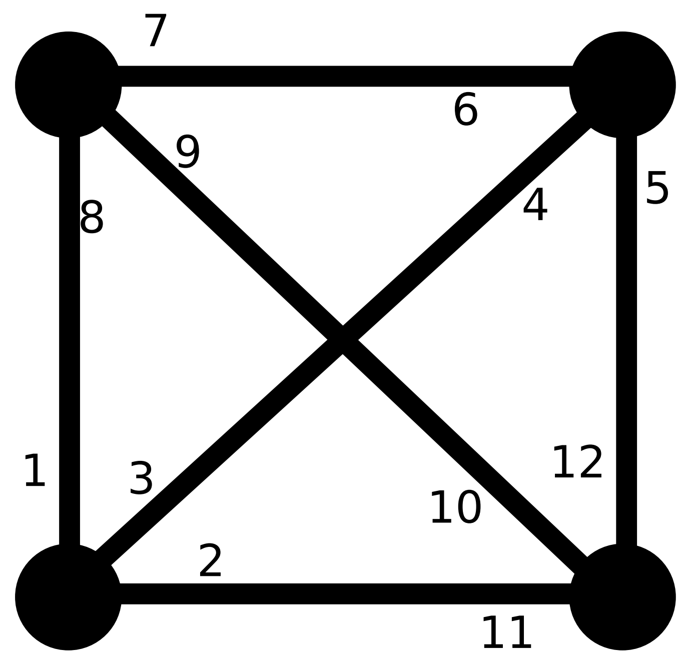
$$f = (1\ 7\ 5\ 11)(2\ 10\ 8\ 3\ 6\ 9\ 12\ 4)$$

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

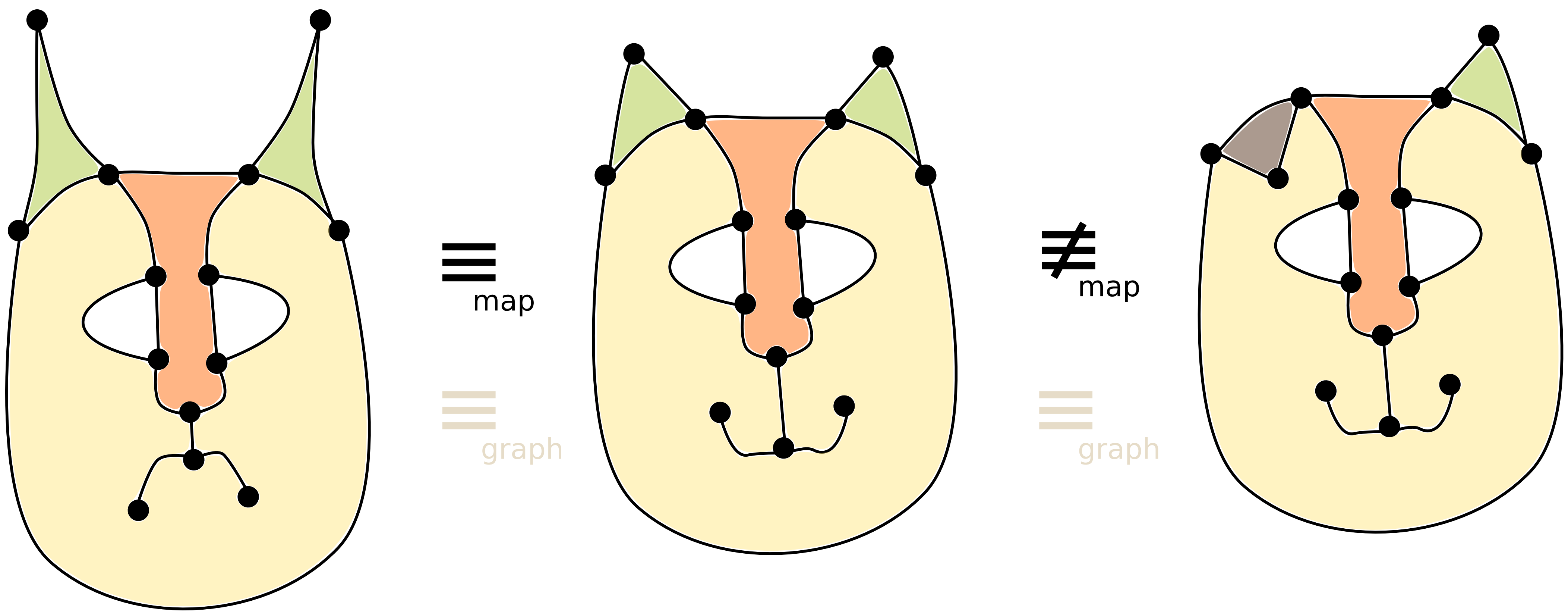
Combinatorial definition

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

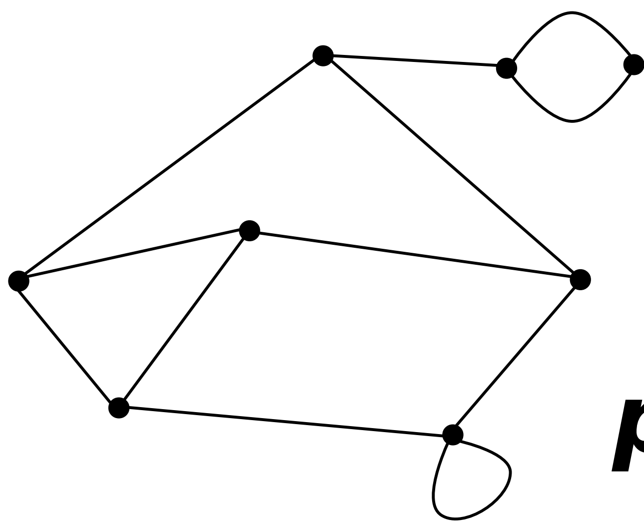




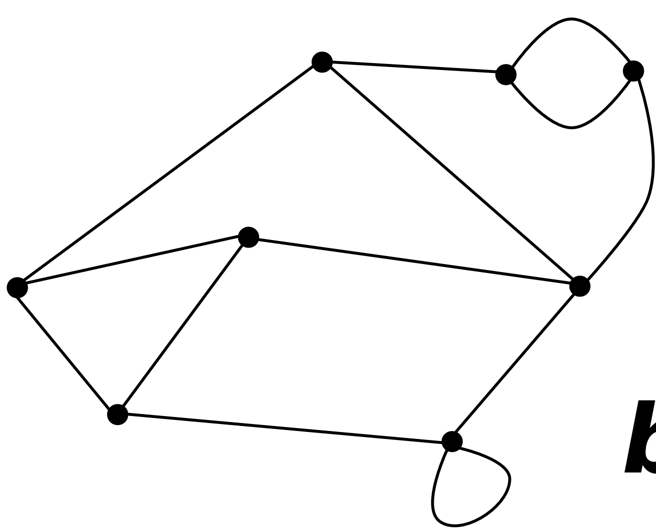
Graph versus Map



Some special kinds of maps

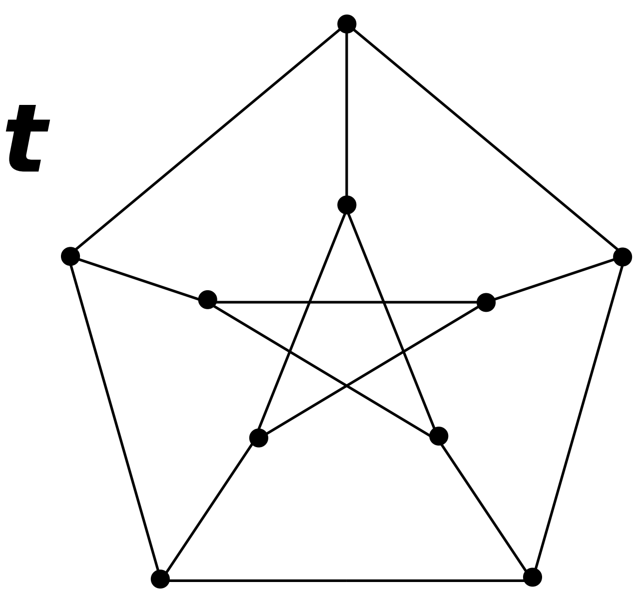


planar



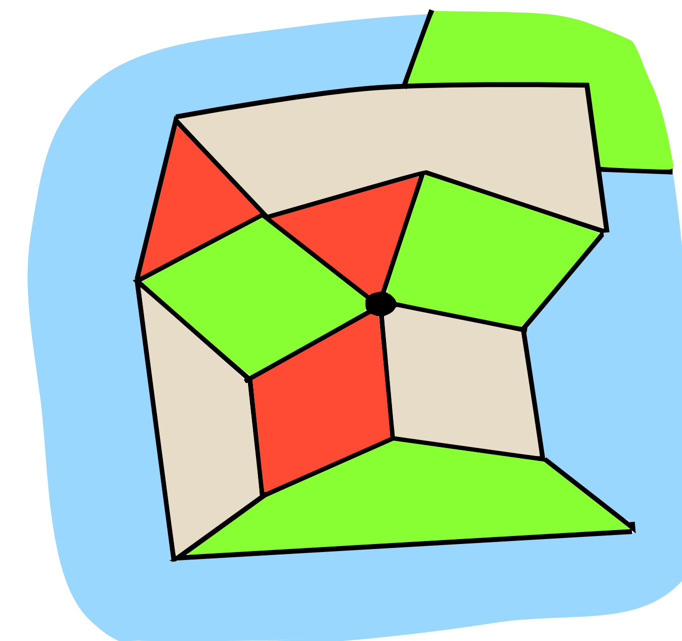
bridgeless

3-valent



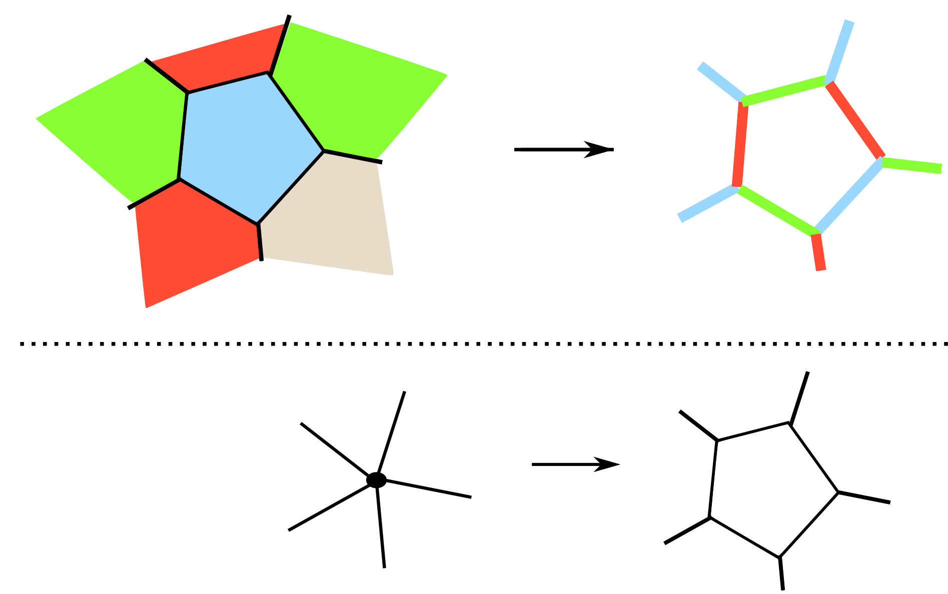
Four Colour Theorem

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.

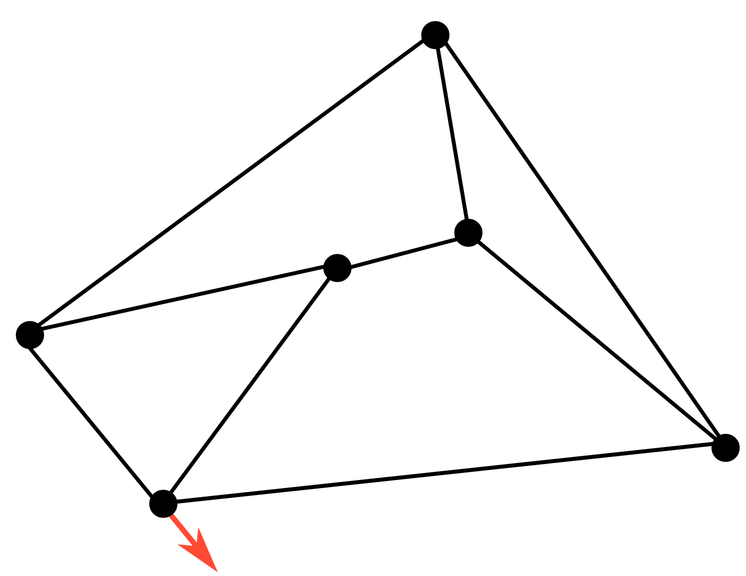
W. T. Tutte, Graph Theory as I Have Known It

Map enumeration

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

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.

(5.1) *The number a_n of rooted maps with n edges is*

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

We write

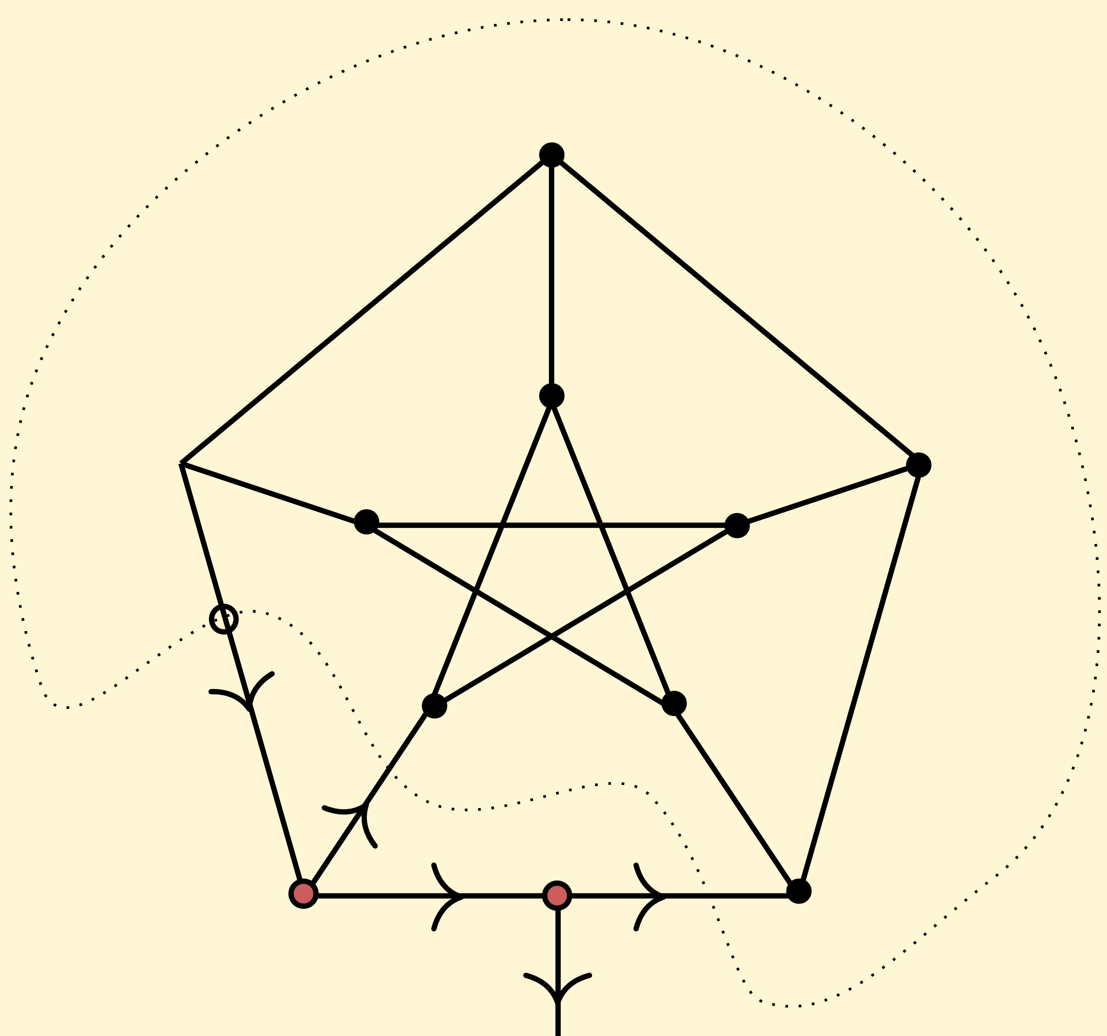
$$A(x) = \sum_{n=1}^{\infty} a_n x^n.$$

Thus $A(x) = 2x + 9x^2 + 54x^3 + 378x^4 + \ldots$. Figure 2 shows the 2 rooted maps with 1 edge, and Figure 3 the 9 rooted maps with 2 edges.

[Background]

A bijection between linear λ-terms and rooted 3-valent maps

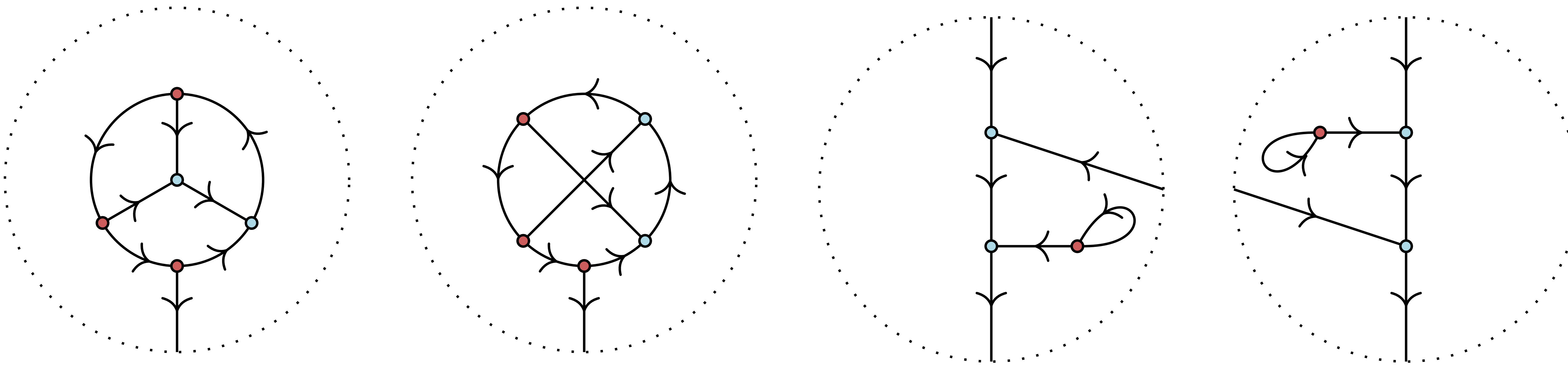
(cf. Bodini et al 2013, Z 2016)



From linear terms to rooted 3-valent maps via string diagrams

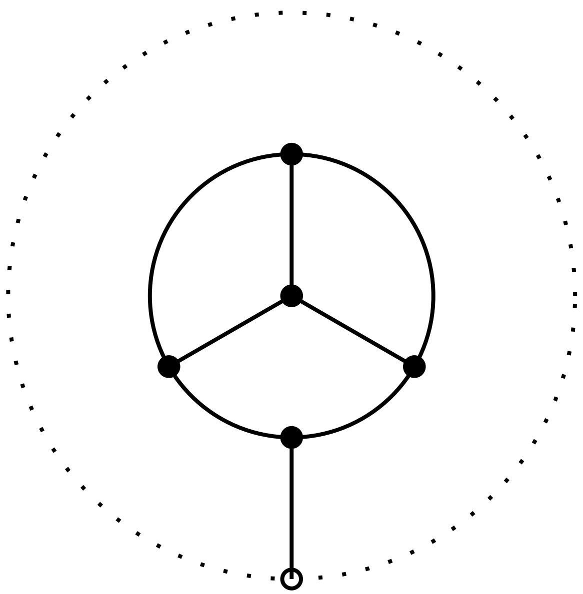
$\lambda x.\lambda y.\lambda z.x(yz)$ $\lambda x.\lambda y.\lambda z.(xz)y$ $x,y \vdash (xy)(\lambda z.z)$ $x,y \vdash x((\lambda z.z)y)$

From linear terms to rooted 3-valent maps via string diagrams

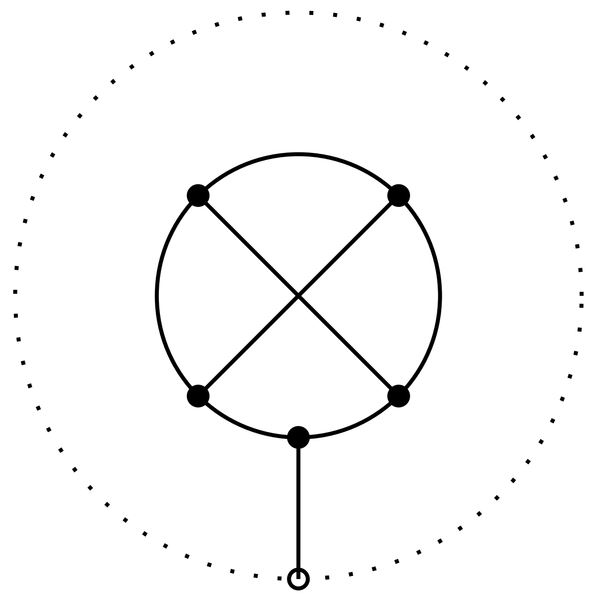


λx.λy.λz.x(yz) | λx.λy.λz.(xz)y | x,y ⊢ (xy)(λz.z) | x,y ⊢ x((λz.z)y)

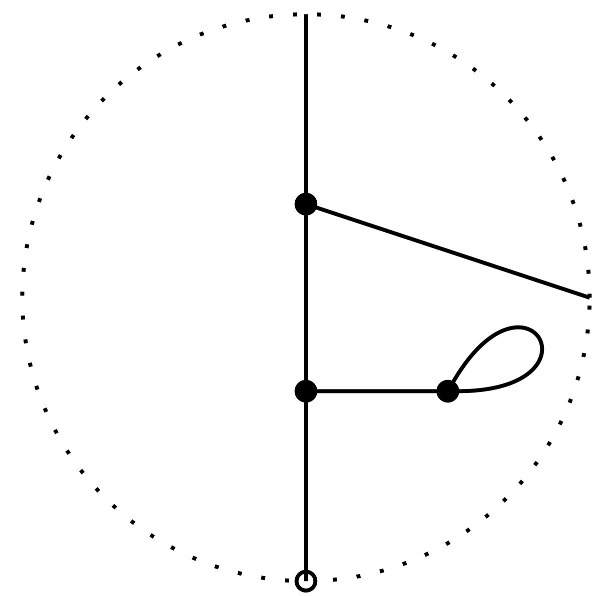
From linear terms to rooted 3-valent maps via string diagrams



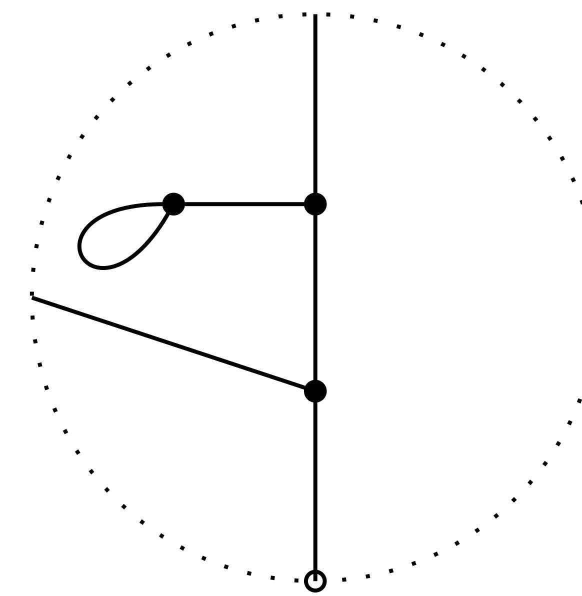
$\lambda x.\lambda y.\lambda z.x(yz)$



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



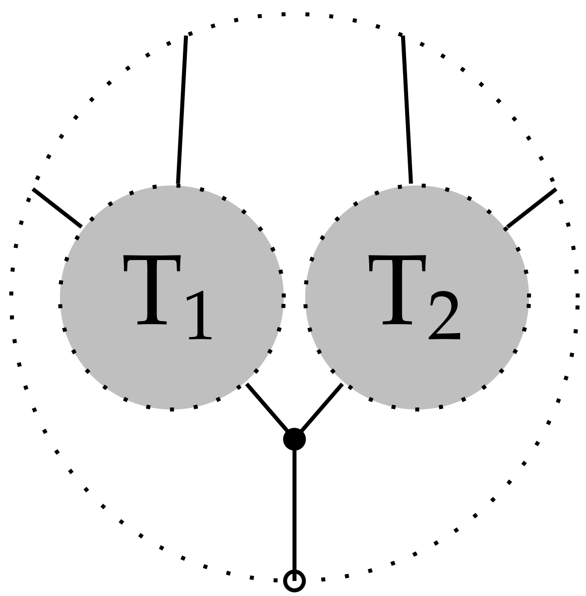
$x,y \vdash (xy)(\lambda z.z)$



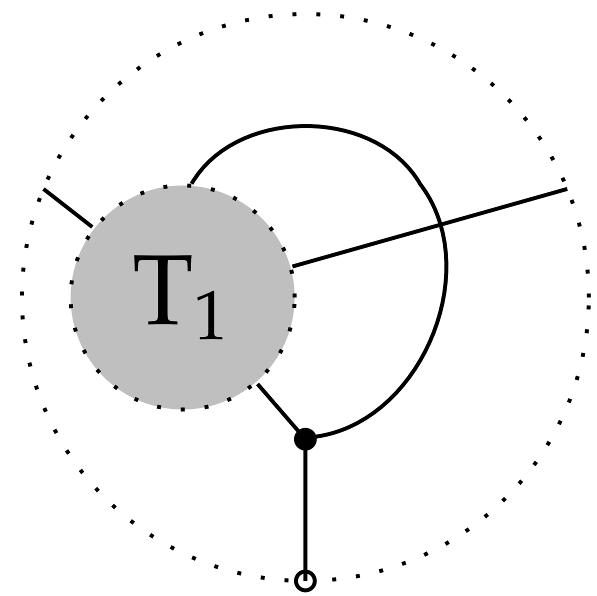
$x,y \vdash x((\lambda z.z)y)$

From rooted 3-valent maps to linear terms by induction

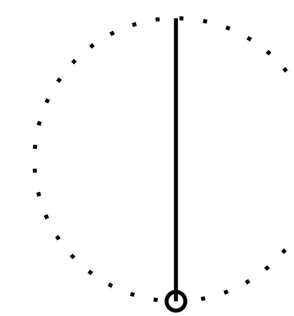
Observation: any rooted 3-valent map must have one of the following forms.



disconnecting root vertex



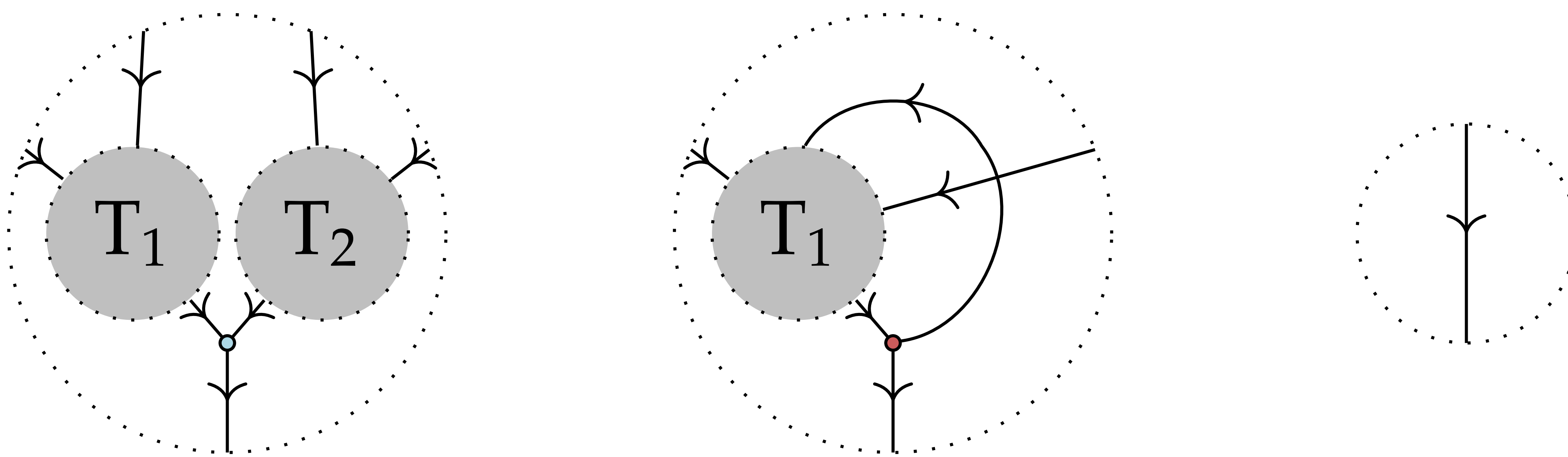
connecting root vertex



no root vertex

From rooted 3-valent maps to linear terms by induction

...but this exactly mirrors the inductive structure of linear lambda terms!

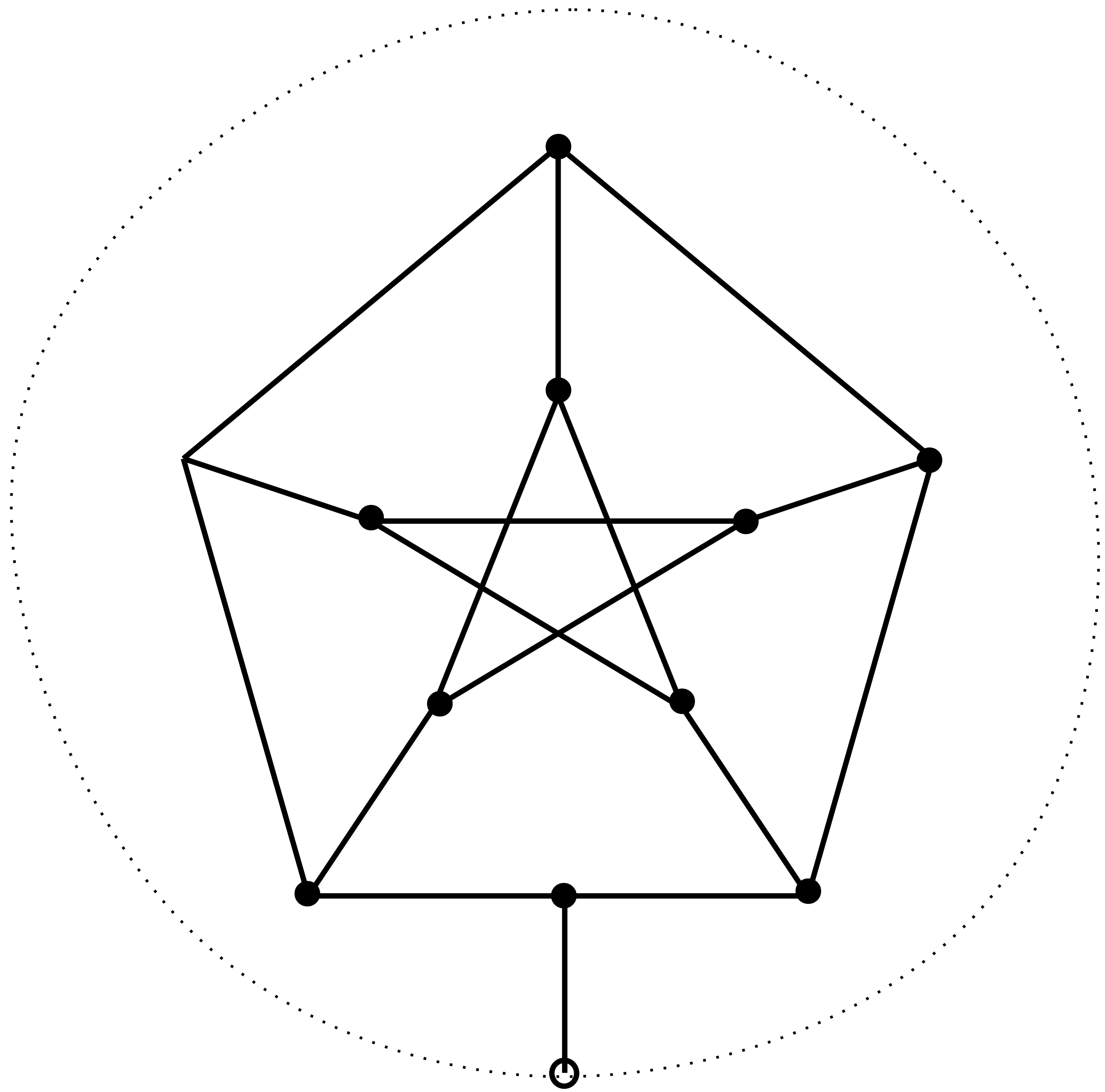


application

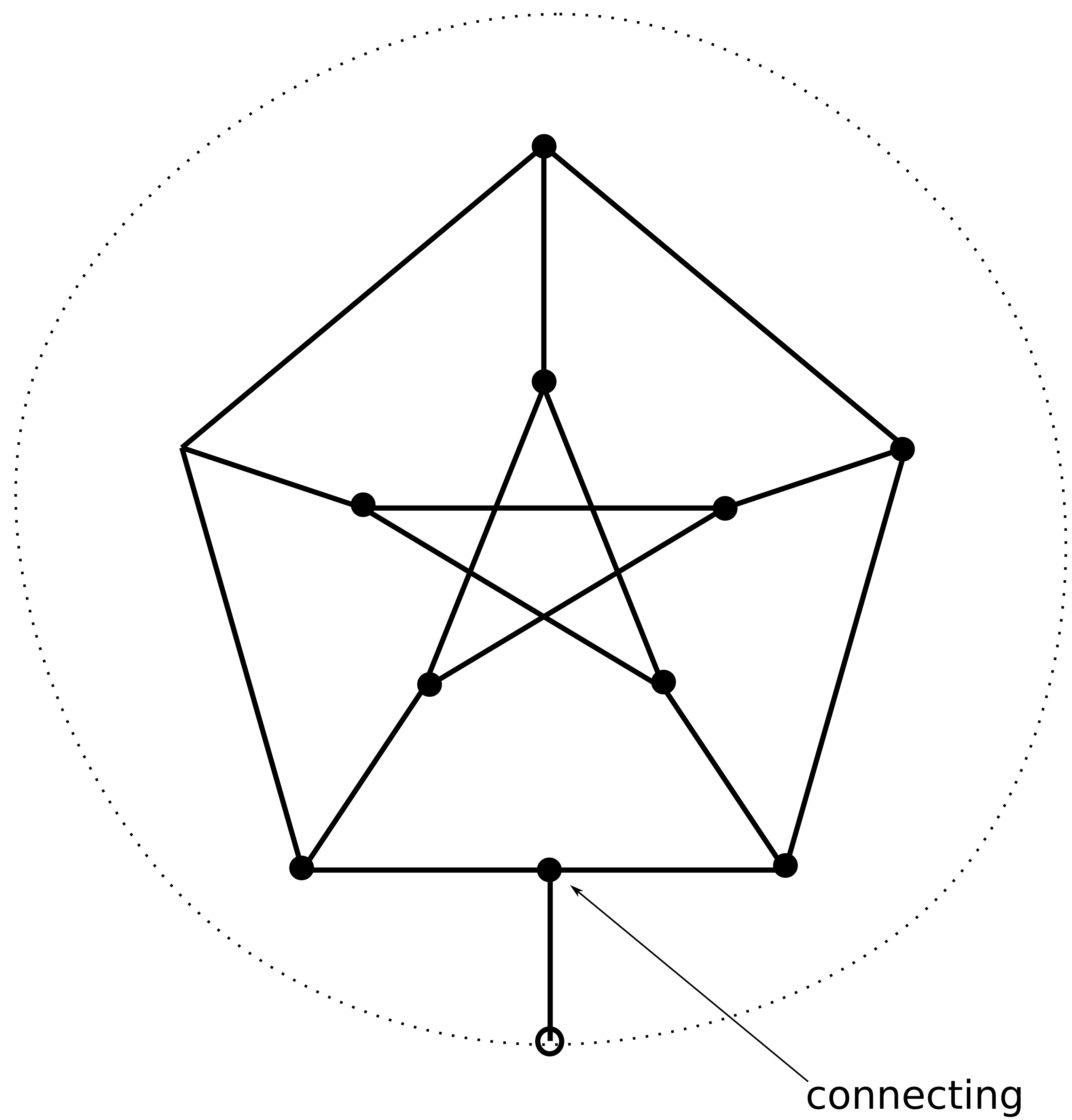
abstraction

variable

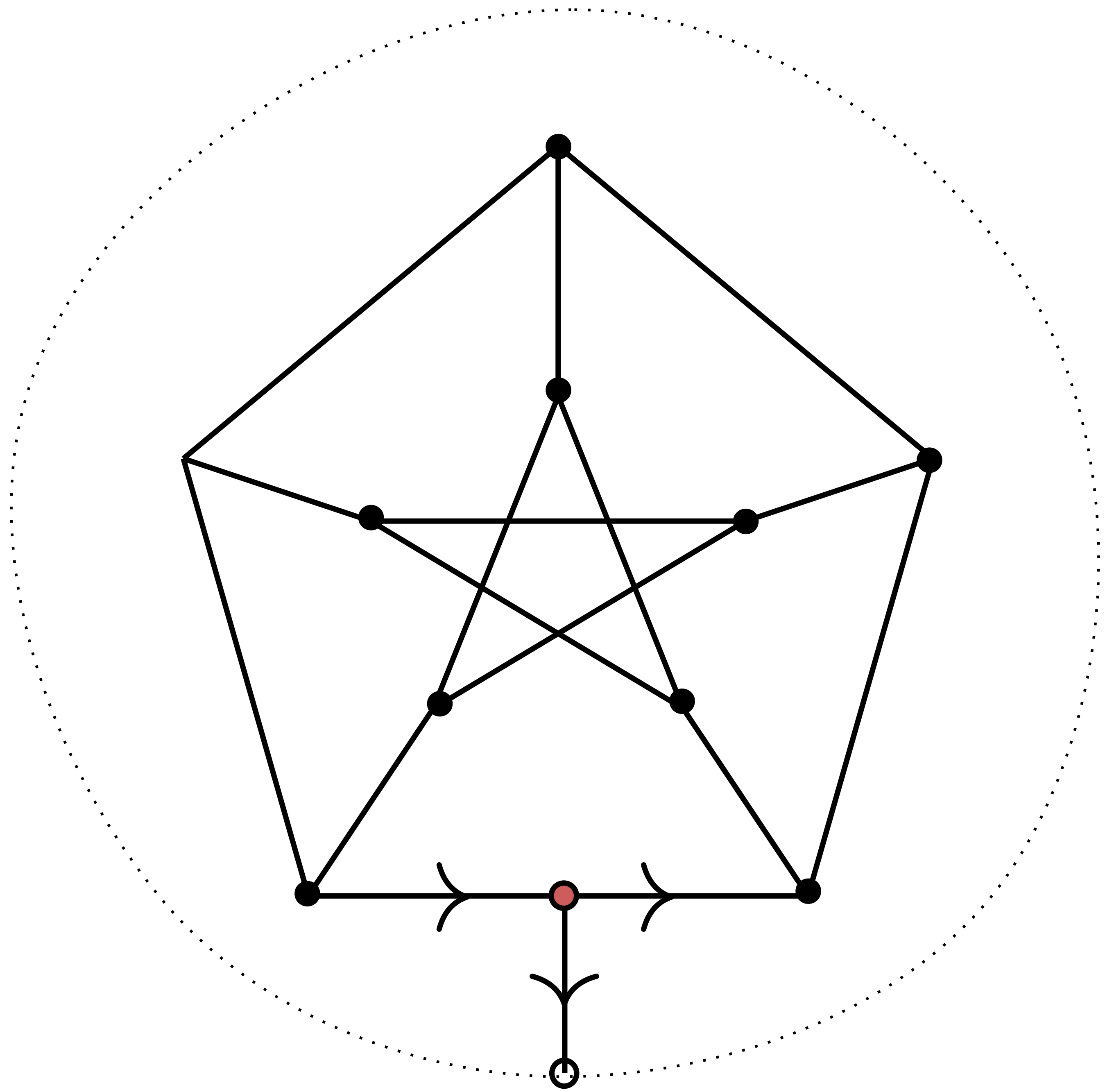
An example



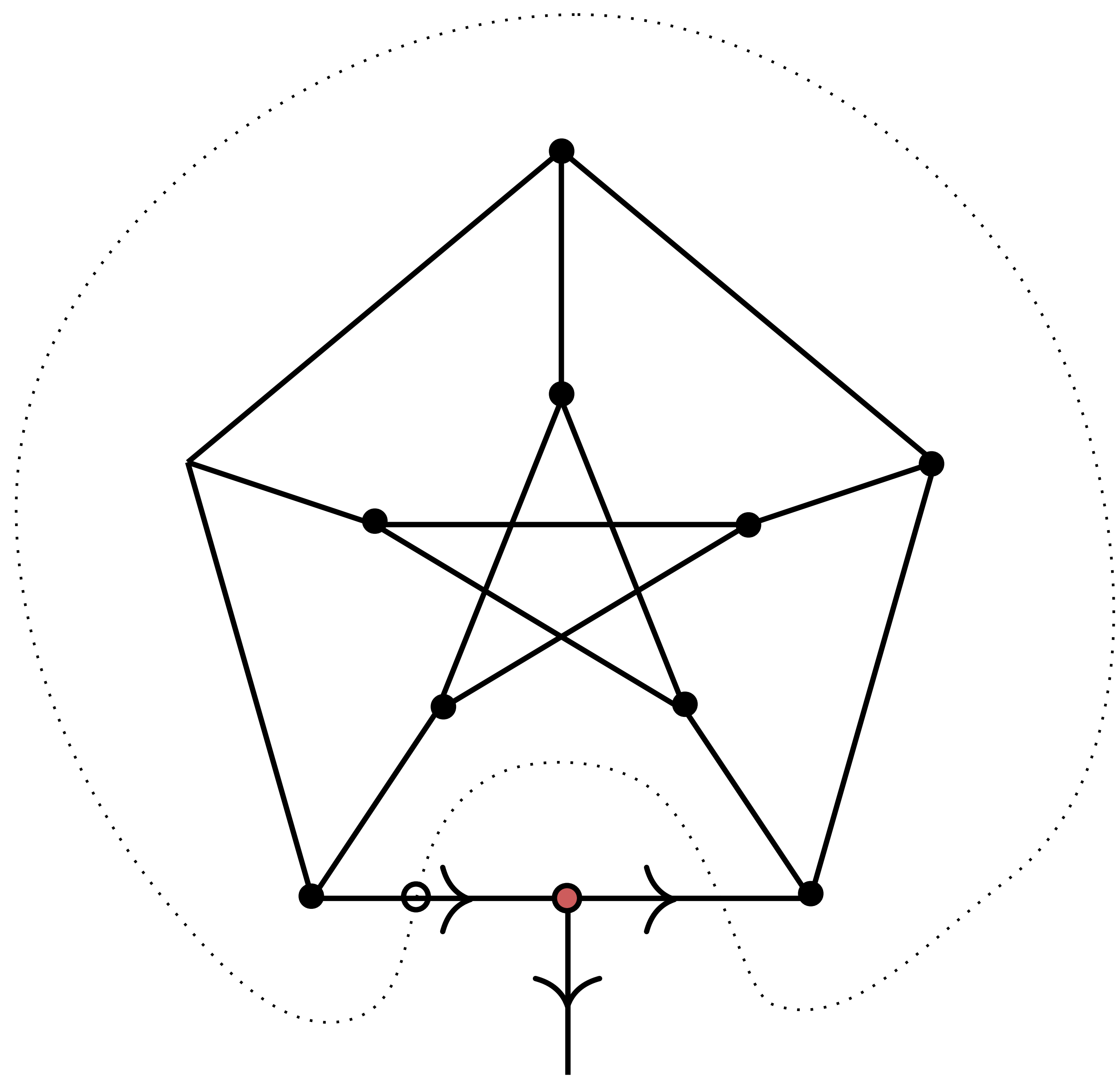
An example



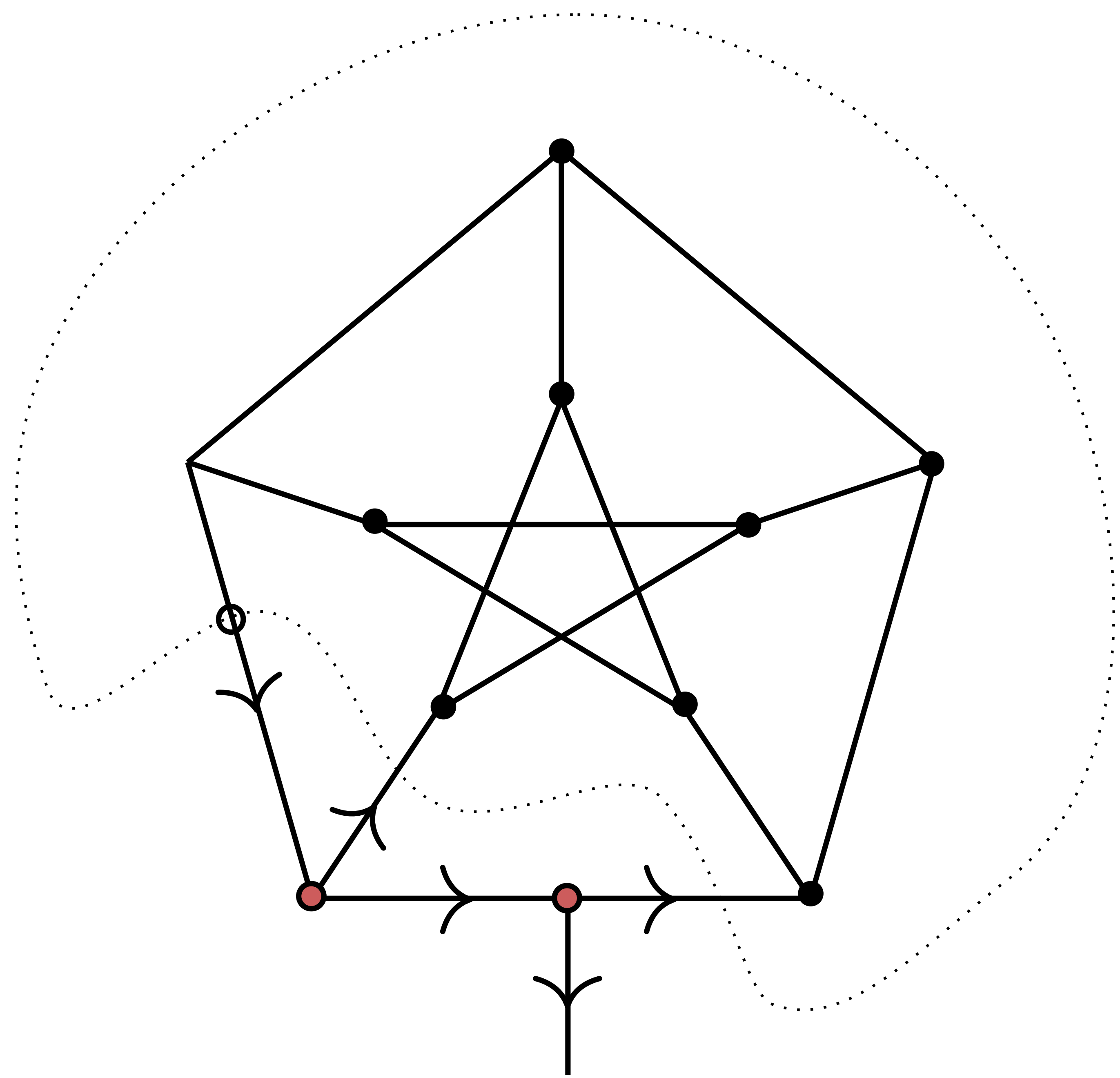
An example



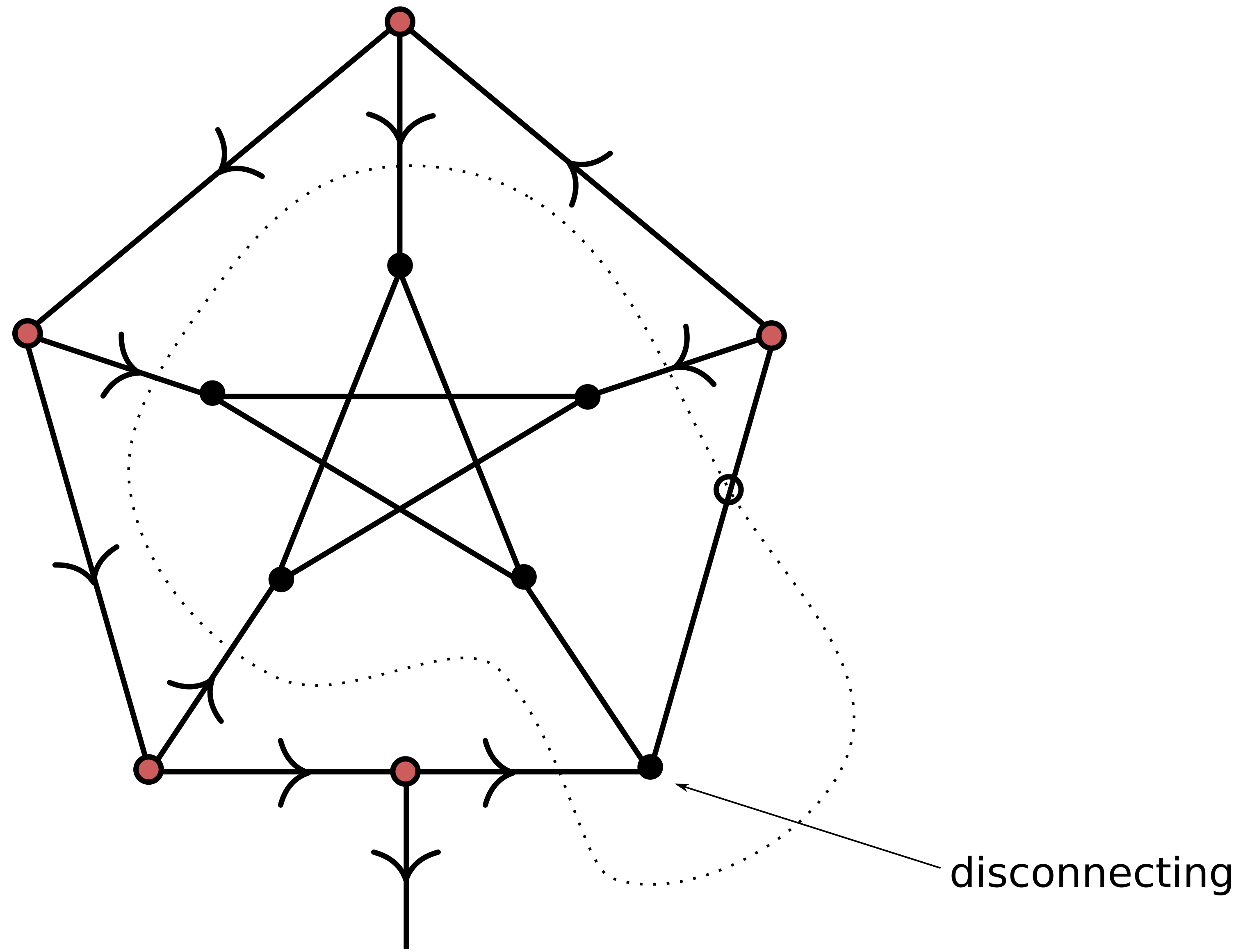
An example



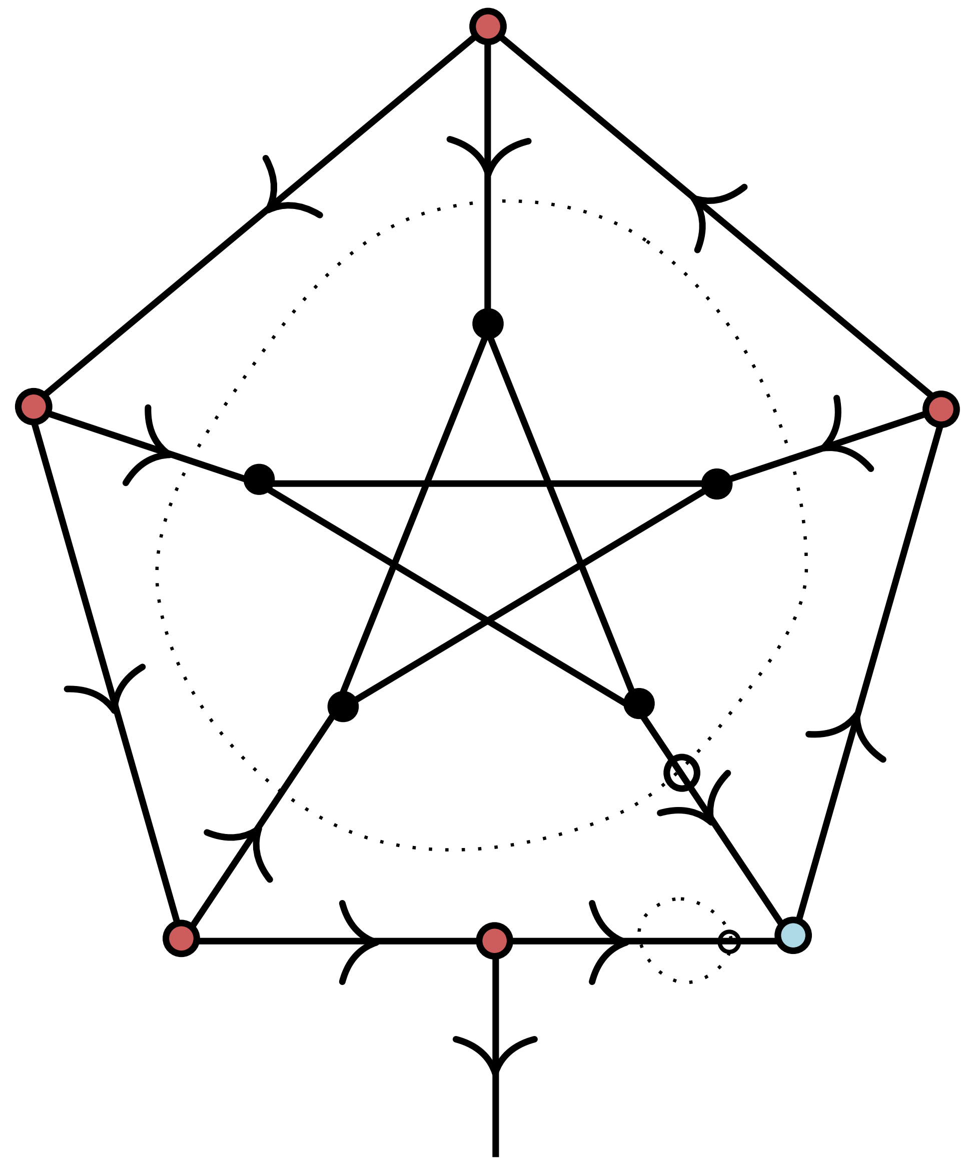
An example



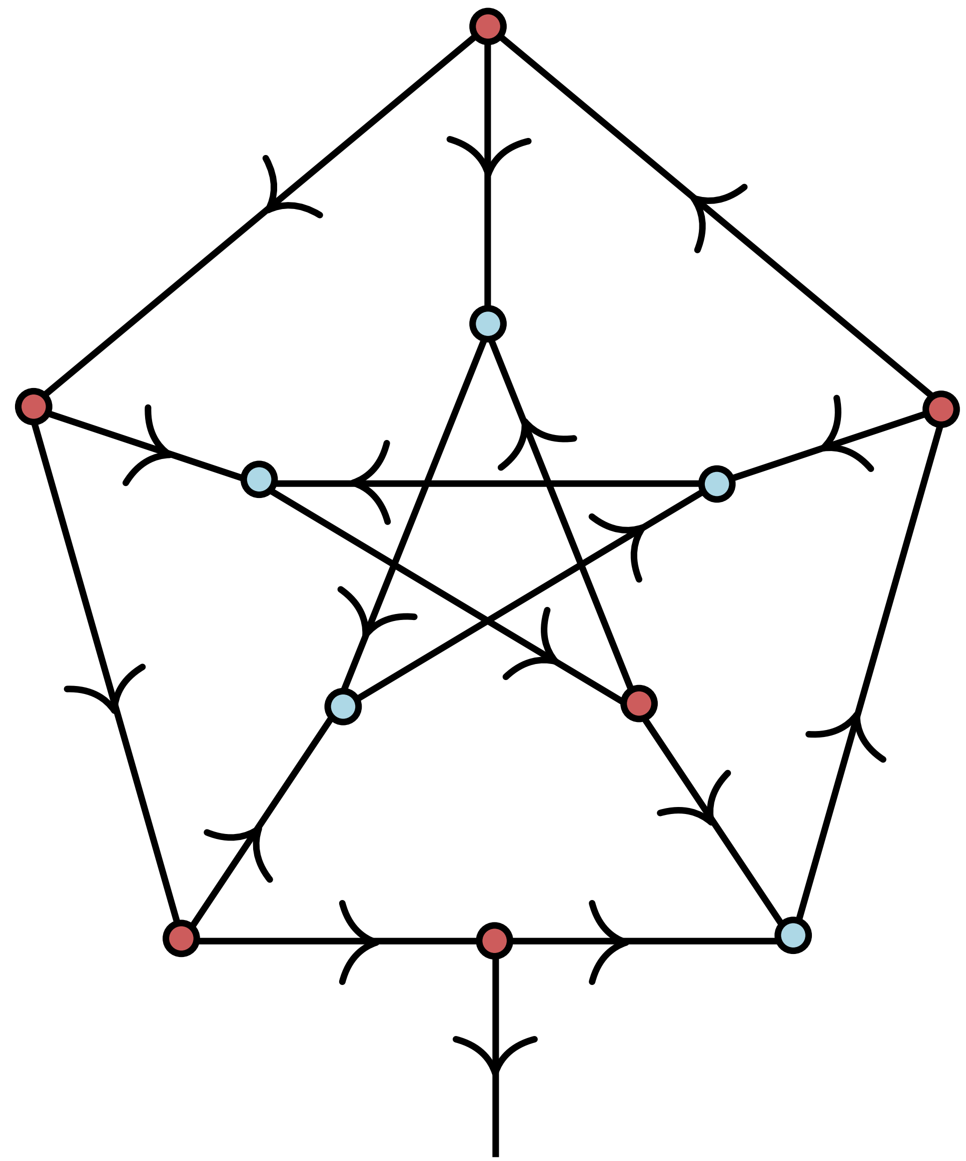
An example



An example

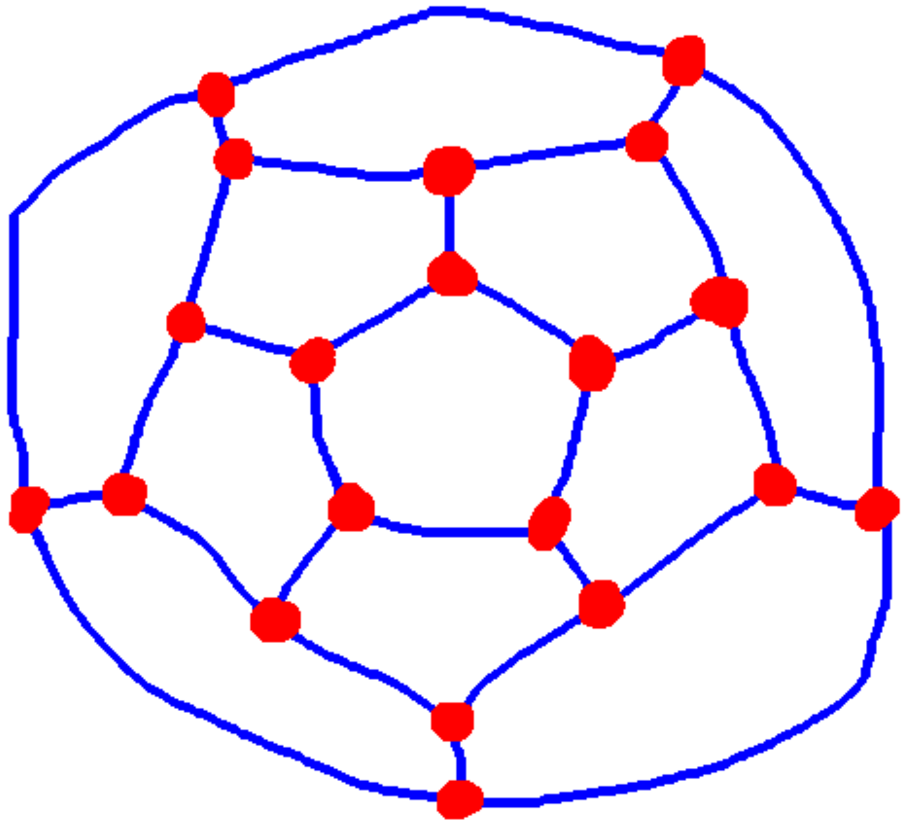


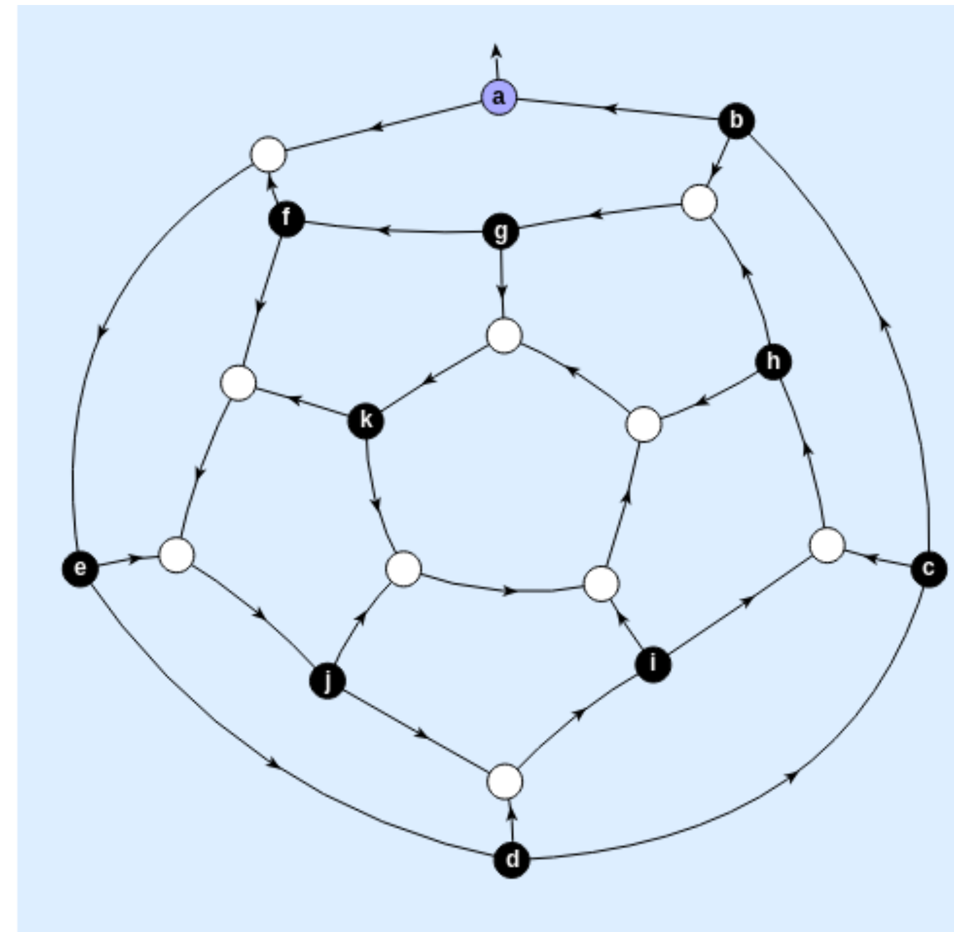
An example



λa.λb.λc.λd.λe.a(λf.c(e(b(df))))

Some more examples*

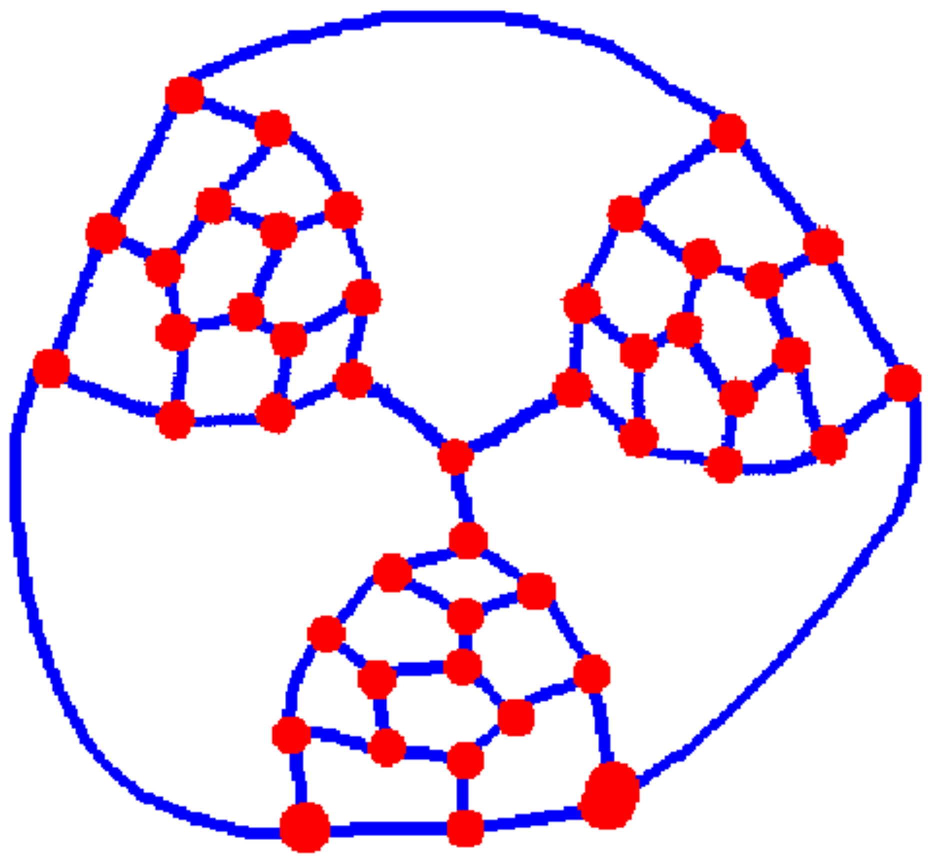


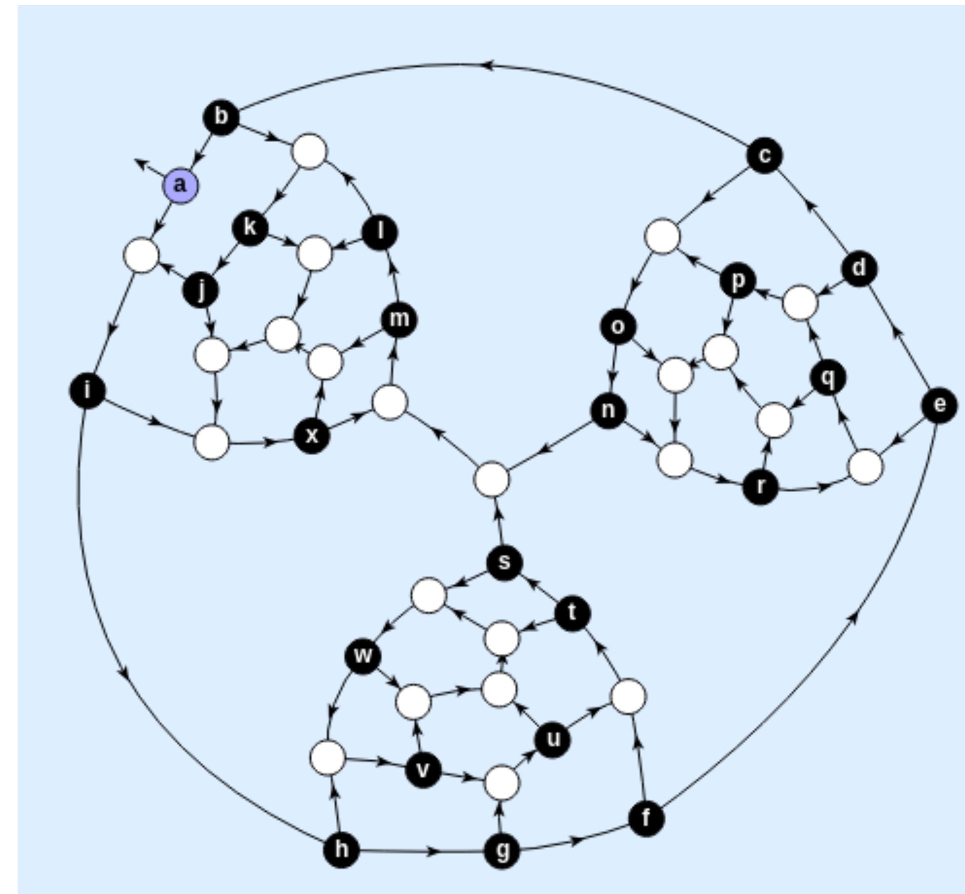


λabcde.a (λfg.b (λh.c (λi.d (λj.e (f (λk.g (h (i (j k))))))))))

*computed with the help of https://jcreedcmu.github.io/demo/lambda-map-drawer/public/index.html

Some more examples*

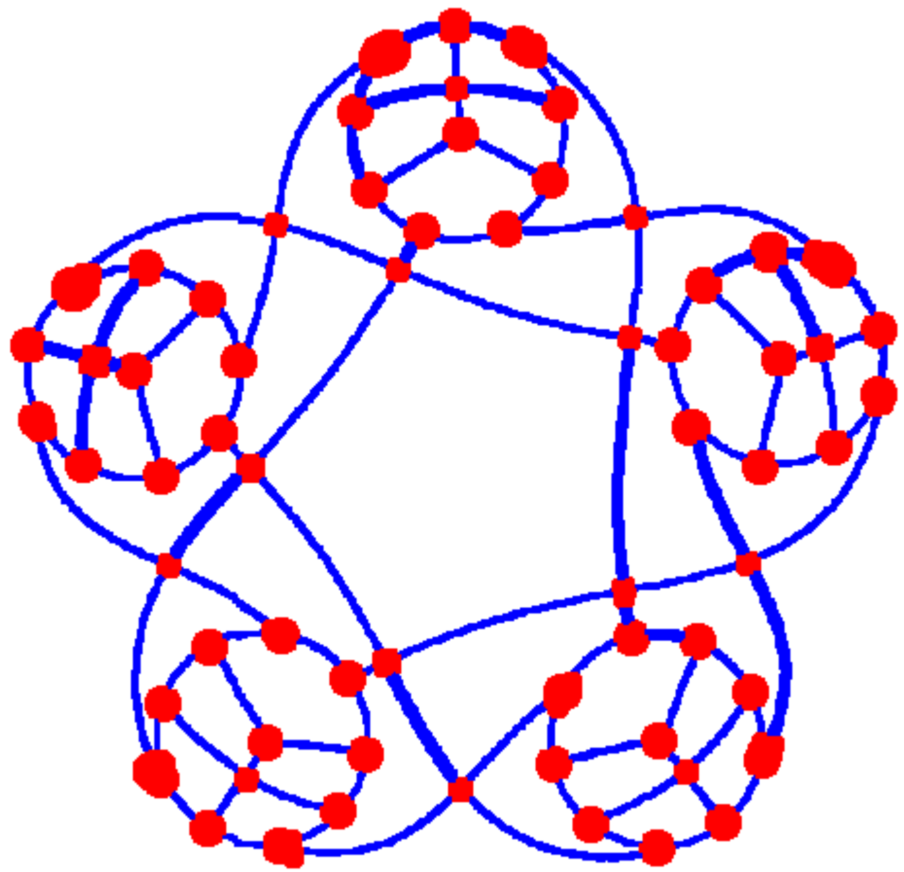


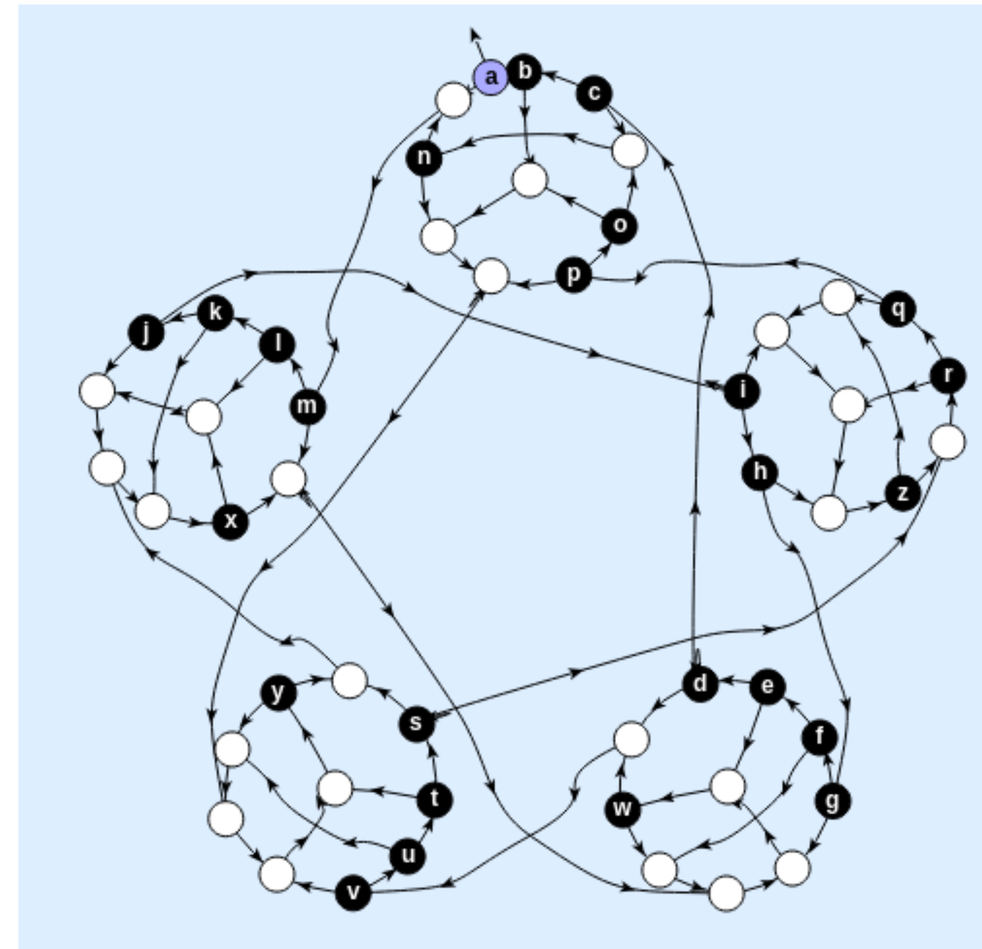


λabcdefghi.a (λjk.b (λlm.(λno.c (λp.d (λq.e (λr.n (o (p (q r))))))) (λst.f (λu.g (λv.h (λw.s (t (u (v w))))))) (λx.i (j (k l (m x))))))

*computed with the help of https://jcreedcmu.github.io/demo/lambda-map-drawer/public/index.html

Some more examples*



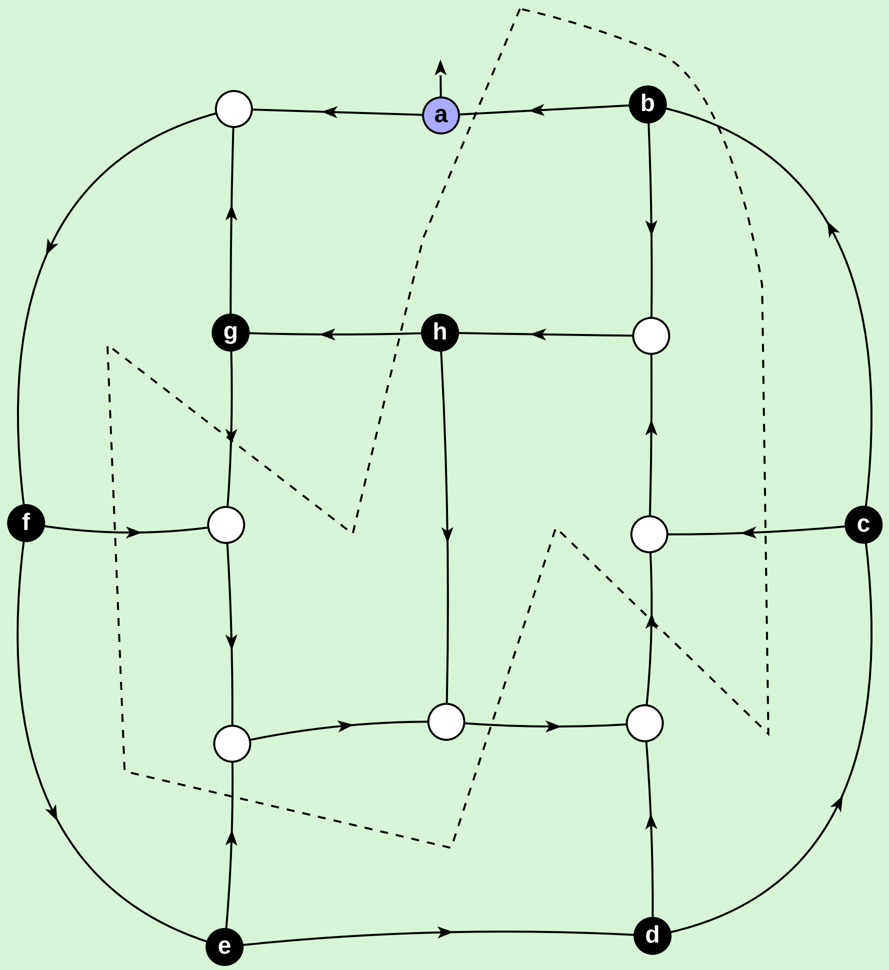


λabcdefghijklm.a (λn.c (λopqr.(λstuv.d (λw.e (g ((λx.s (λy.t (v (n (b o) p (y u)))) (j (l x)) k) m (w f))))) (λz.h (i (q z) r))))

*computed with the help of https://jcreedcmu.github.io/demo/lambda-map-drawer/public/index.html

[work-in-progress]

Higher connectivity of linear λ-terms

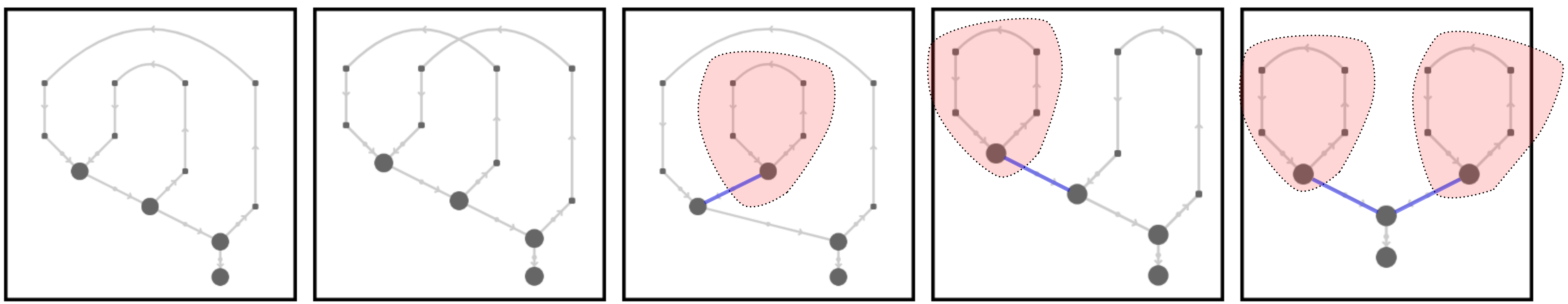


characterization of bridgeless terms

*reminder: bridgeless = stays connected after removing any edge.

from the description of the bijection φ, it's not hard to prove that...

M bridgeless ⇔ φ(M) has no closed subterms



λx. λy. x y — λx. λy. y x — λx. x (λy. y) — λx. (λy. y) x — (λx. x) (λy. y)

one corollary: equivalent λ-calculus reformulation of 4CT!

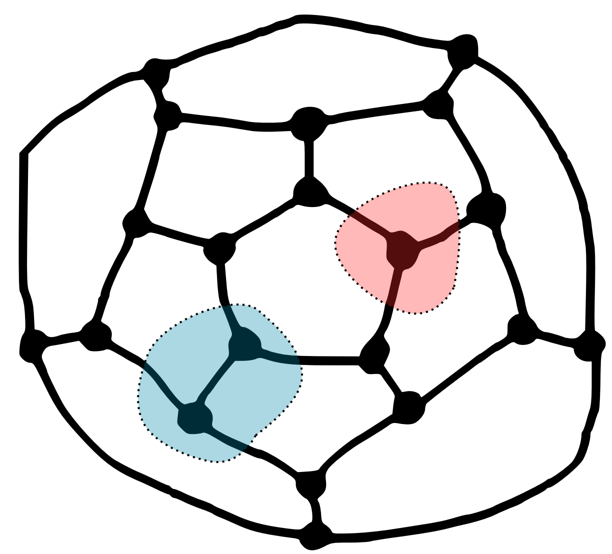
(cf. JFP 2016, LICS 2018)

k-edge-connection

a graph is **k-edge-connected** if it stays connected after cutting any j < k edges

(e.g., 1-edge-connected = connected, 2-edge-connected = bridgeless)

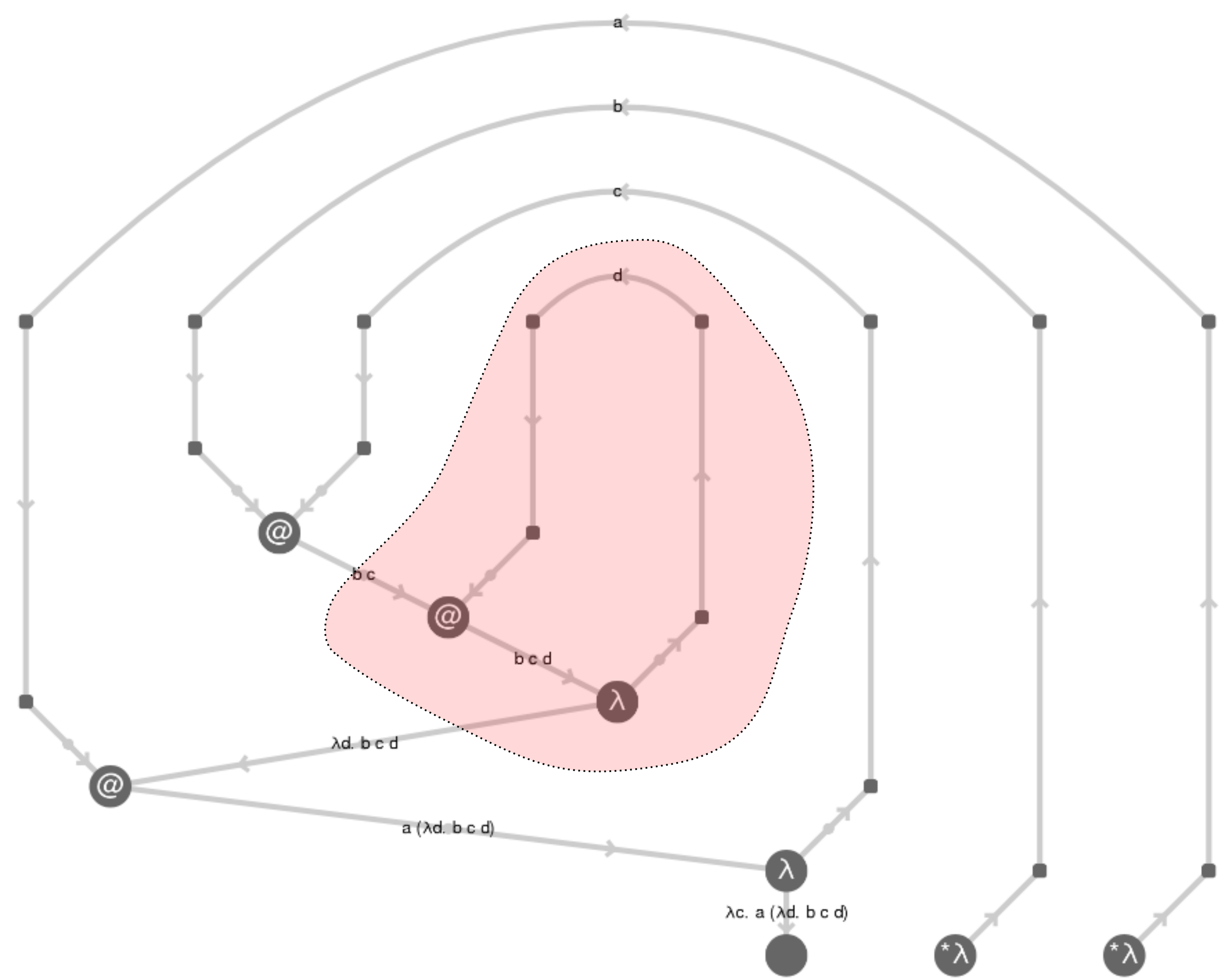
turns out useful to weaken to "internal" k-edge-connection (only trivial j-cuts)



internally 4-edge-connected
(trivial 3-cut, non-trivial 4-cut)

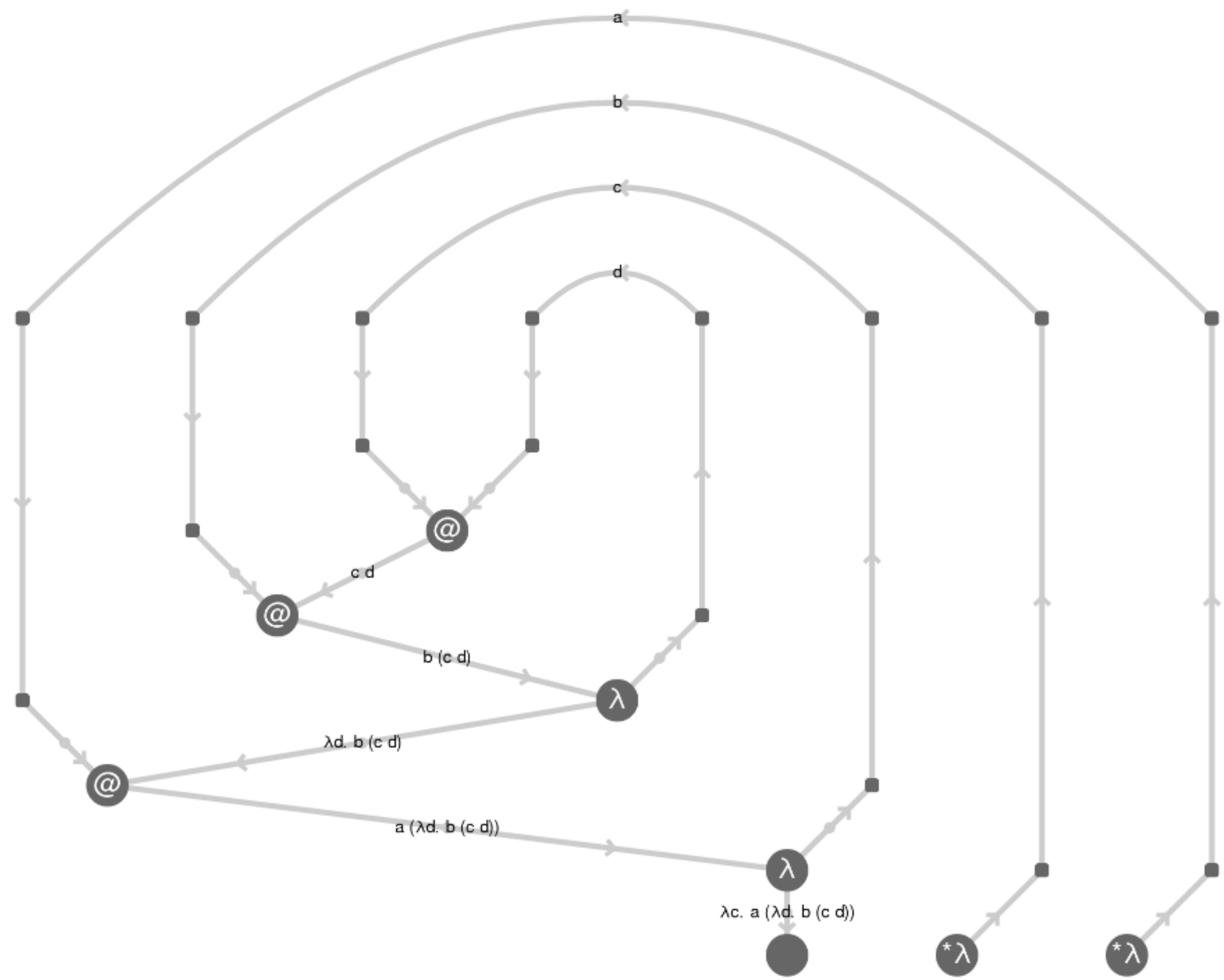
What does it mean for a λ-term to be internally k-edge-connected?

a term which is 2- but not 3-edge-connected



a, b ⊢ λc.a (λd.(b c) d)

a 3-edge-connected term



a, b ⊢ λc.a (λd.b (c d))

towards a logical characterization

A **cut** is a decomposition

$$t_1 = C\{t_2\}$$

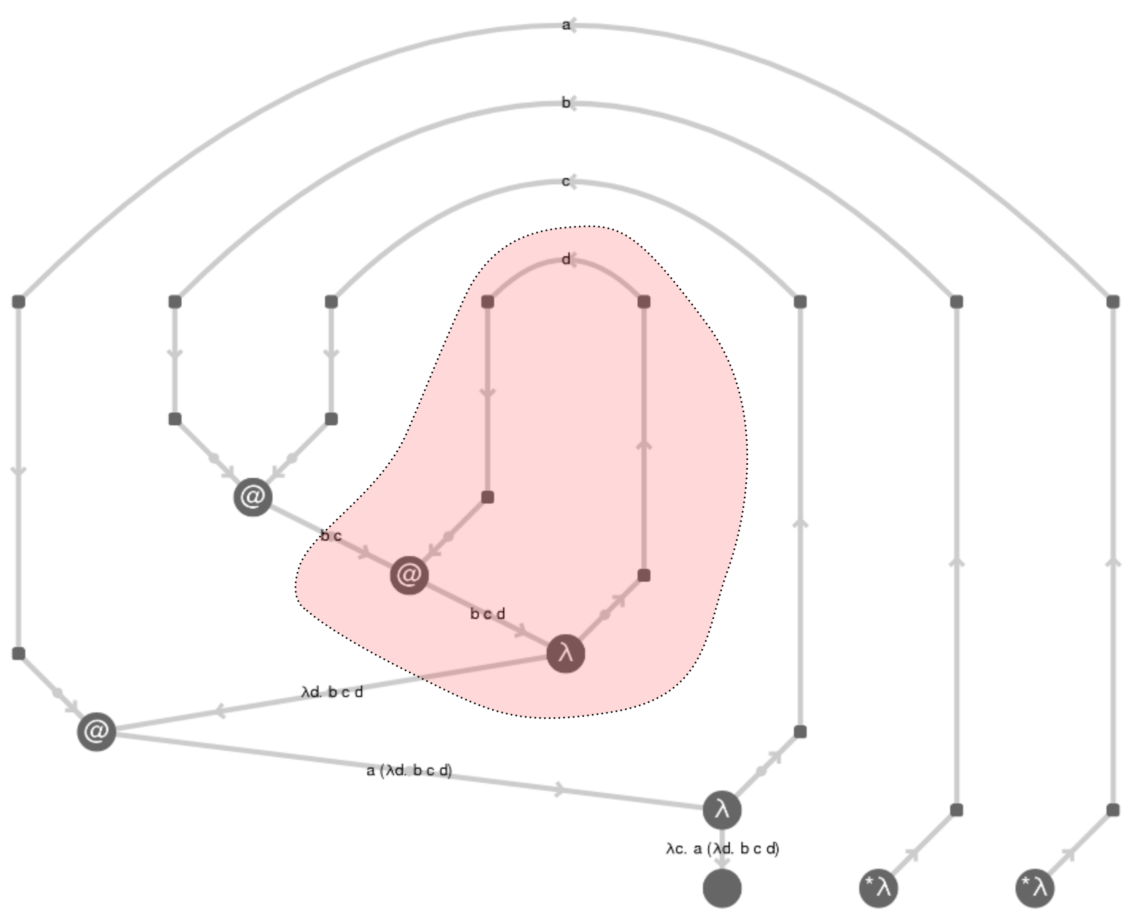
of a term t_1 into a *subterm* t_2 together with its surrounding *context* C.
Roughly speaking, a "context" is just a term with a hole/metavariable.

This definition gets a lot more interesting if we represent terms using HOAS and allow ("generalized") subterms to have higher type.

Then we say that the **type** of a cut $t_1 = C\{t_2\}$ is the type of t_2.

towards a logical characterization

For example, a few slides ago, we saw a term with a cut of type U ⊸ U



t_1 : U ⊸ (U ⊸ U)

t_1 = [a] [b] lam [c] app a (lam [d] app (app b c) d)

t_2 : U ⊸ U

t_2 = [x] lam [d] app x d

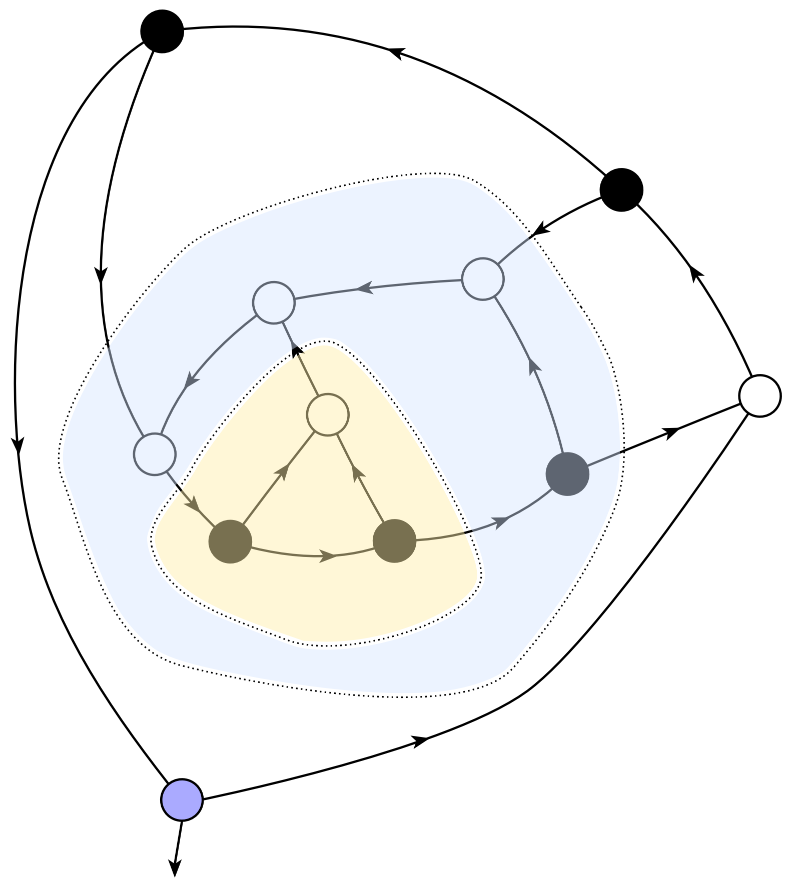
C : (U ⊸ U) ⇒ (U ⊸ (U ⊸ U))

C = {X} [a] [b] lam [c] app a (X (app b c))

a, b ⊢ λc.a (λd.(b c) d)

towards a logical characterization

Here is an example of a term with a yellow cut of type (U ⊸ U) ⊸ U
and a blue cut of type U ⊸ (U ⊸ U)



λa.λb.λc.a (λd.λe.λf.(b (c d)) (e f))

t_1 : U
t_1 = lam [a] lam [b] lam [c] app a (lam [d] lam [e] lam [f]
app (app b (app c d)) (app e f))

t_2 : (U ⊸ U) ⊸ U
t_2 = [G] lam [e] lam [f] G (app e f)

C : (U ⊸ U) ⊸ U ⇒ U
C = {X}lam [a] lam [b] lam [c]
app a (lam [d]
X ([y] app (app b (app c d)) y))

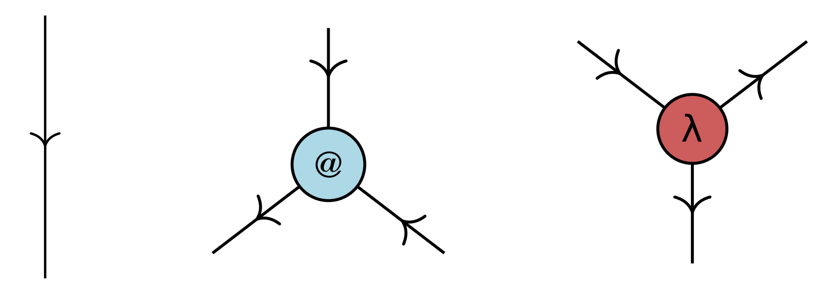
t_2' : U ⊸ (U ⊸ U)
t_2' = [b] [c] lam [d] lam [e] lam [f]
app (app b (app c d)) (app e f))

C' : U ⊸ (U ⊸ U) ⇒ U
C' = {X} lam [a] lam [b] lam [c] app a (X b c)

towards a logical characterization

Let us say that a cut $t_1 = C\{t_2\}$ is **trivial** if either C is the identity context or t_2 is one of the following **elementary** terms:

$$\begin{aligned} \lambda x.x &: U \multimap U \\ \mathrm{app} &: U \multimap (U \multimap U) \\ \mathrm{lam} &: (U \multimap U) \multimap U \end{aligned}$$



Define the **size** of a type as the number of occurrences of "U" (e.g., $|U \multimap U| = 2$)

Definition: a term is **k-indecomposable** if it has no non-trivial τ-cuts for $|\tau| < k$

Claim (conjecture): *t is k-indecomposable iff t is internally k-edge-connected.*

motivations & questions

Internally 3- and 4-edge-connected planar 3-valent maps were first enumerated by Tutte (1961) who found some nice counting formulas.

Surprisingly, Tutte's formula for 3-edge-connected planar 3-valent maps also counts β-normal 2-indecomposable ordered terms (A000260).

Indeed, there is a simple bijection

[3-ind ordered terms] ↔ [β-normal 2-ind ordered terms]

the bijection goes by way of open "neutral" terms, although it is not obviously meaningful... here is the graph of the bijection at n=3 apps:

a (b (c d)) ↔ a (b (c d))
a ((b c) d) ↔ a ((b c) d)
(a b) (c d) ↔ (a b) (c d)
(a (b c)) d ↔ (a (b c)) d
((a b) c) d ↔ ((a b) c) d

λc.a (λd.b (c d)) ↔ a (λc.b (λd.c d))
λc.λd.a (b (c d)) ↔ a (λc.λd.b (c d))
λc.λd.a ((b c) d) ↔ a (λc.λd.(b c) d)
a (λd.b (c d)) ↔ a (b (λd.c d))
λd.a (b (c d)) ↔ a (λd.b (c d))

λd.a ((b c) d) ↔ a (λd.(b c) d)
λd.(a b) (c d) ↔ (a b) (λd.c d)
(λd.a (b d)) c ↔ (a (λd.b d)) c

Conjecture: β-normal 3-ind ordered terms are counted by A000257.

motivations & questions

one of our original motivations was to revisit some old results in graph theory, such as Whitney's theorem (1931) that *every internally 4-edge-connected planar 3-valent map has a Hamiltonian cycle on its faces.*

4. A theorem on maps deducible immediately from Theorem I is the following, as we shall see later:

THEOREM II. *Given a map on the surface of a sphere containing at least three regions in which:*

(A_1) *The boundary of each region is a single closed curve without multiple point,*

(B) *Exactly three boundary lines meet at each vertex,*

(A_2) *No pair of regions taken together with any boundary lines separating them form a multiply connected region,*

(A_3) *No three regions taken together with any boundary lines separating them form a multiply connected region,*

we may draw a closed curve which passes through each region of the map once and only once, and touches no vertex.

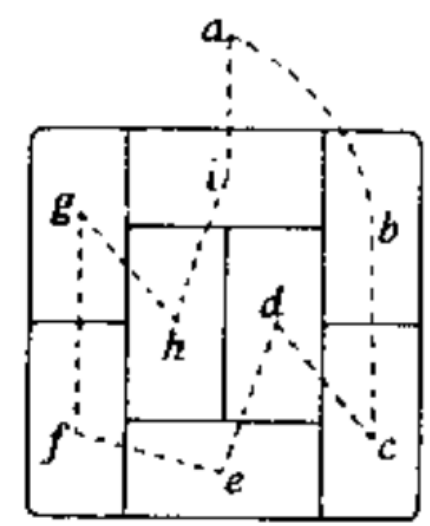
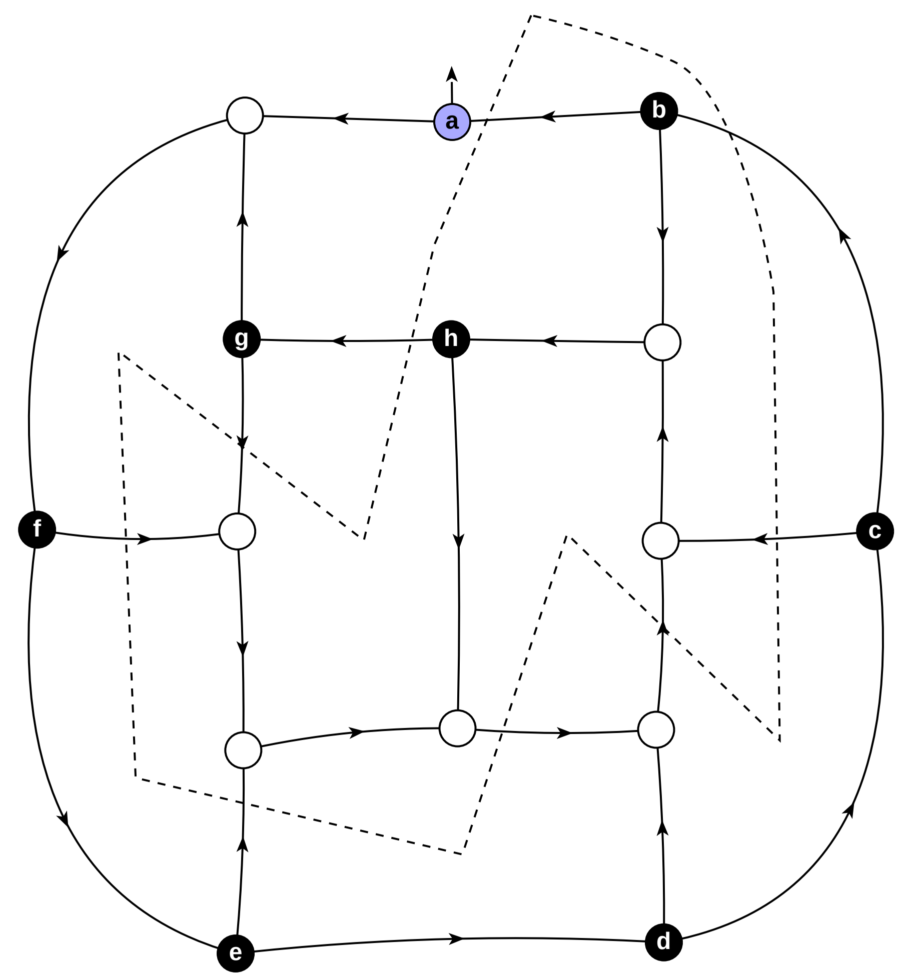


Fig. 3.

motivations & questions

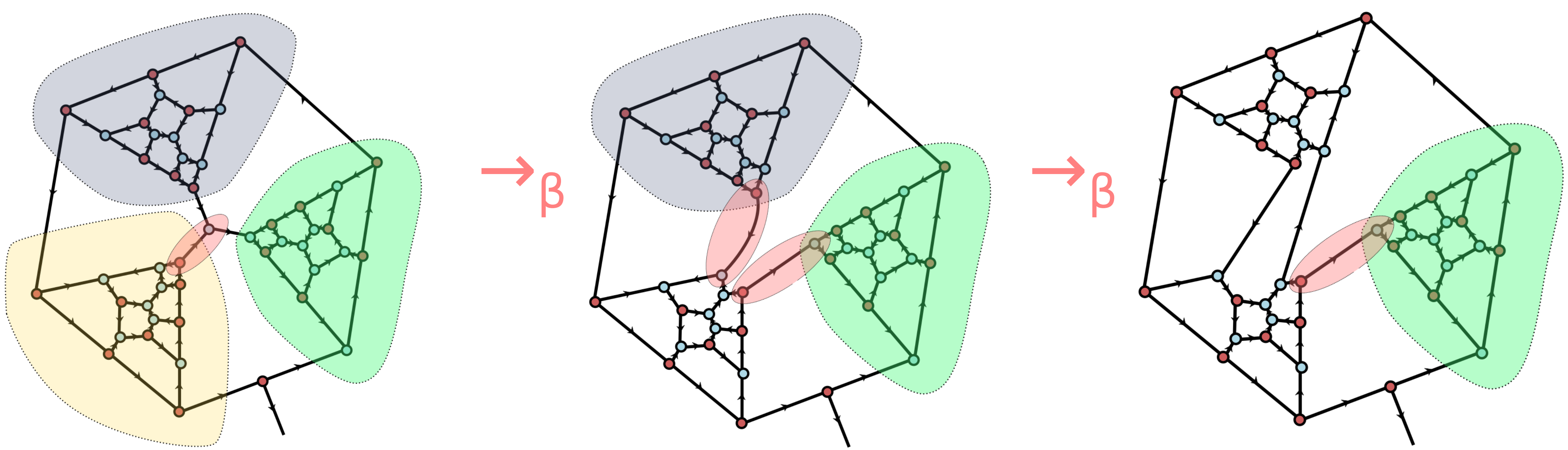
Question: is there a nice/new proof of Whitney's theorem as a statement about 4-indecomposable ordered λ-terms?



motivations & questions

More broadly speaking, would like to better understand the relationships between a term and its (generalized) subterms.

How do cuts evolve over the course of evaluation?



What are the λ-analogues of graph minor theorems?