Graphical Linear Algebra

QPL ’15 Tutorial

Pawel Sobocinski
University of Southampton
(joint work with F. Bonchi and F. Zanasi, ENS Lyon)

graphicallinearalgebra.net

5 stages of addiction denial

(Kubler Ross Model)

- Petri nets, compositionally, with string diagrams: *Representations of Petri net interactions*, CONCUR `10 (2010)
- **Denial** (2011)
  - these proofs are really cute, but I have more important things to do with my life
- **Anger** (2012)
  - why can't I stop drawing them?
- **Grief** (2013)
  - they are taking over :(
- **Bargaining** (2014)
  - I will try to keep other research side-interests… but let me just try to understand what's going on here…
- **Acceptance** (2015)
  - blog, QPL tutorial

Plan

• **maths of string diagrams** • theory of natural number matrices (bimonoids) and integer matrices (Hopf monoids) • theory of linear relations (interacting Hopf monoids)	Monday
• distributive laws • linear algebra, diagrammatically • an application: generating functions and signal flow graphs	Tuesday

Plan

- **maths of string diagrams**
  - setup is slightly different to the usual Oxford lore
  - a “formal semantics/computer science” bent
- theory of natural number matrices (bimonoids) and integer matrices (Hopf monoids)
- theory of linear relations (interacting Hopf monoids)
- distributive laws
- linear algebra, diagrammatically
- an application: generating functions and signal flow graphs

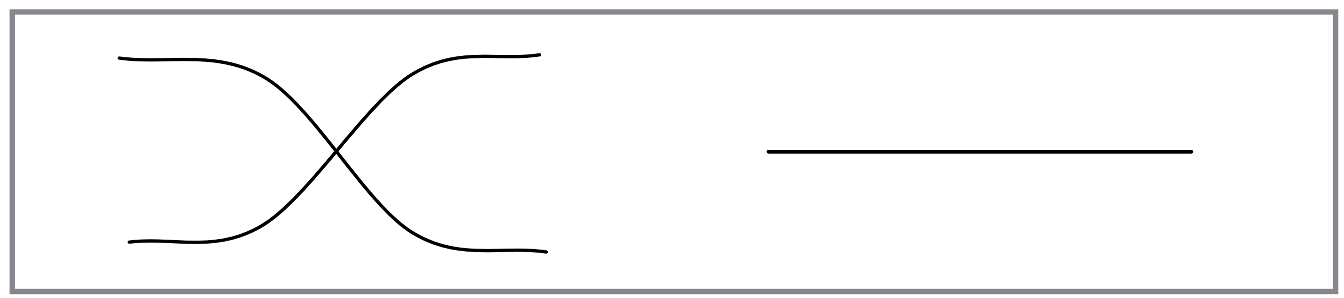
Maths of string diagrams

- PROPs (product and permutation categories)
  - strict symmetric monoidal
  - objects = natural numbers
  - monoidal product on objects = addition
- e.g. the PROP **F** where arrows from m to n are the functions from [m] = {0,1,…, m-1} to [n]
  - equivalent to **FinSet**

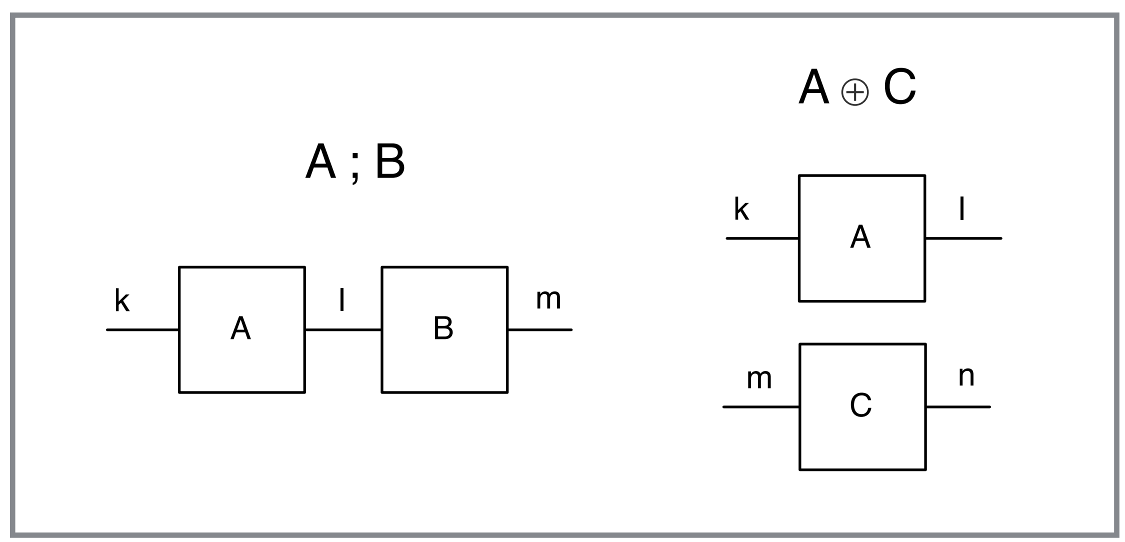
Symmetric monoidal theories

- generators (e.g.)

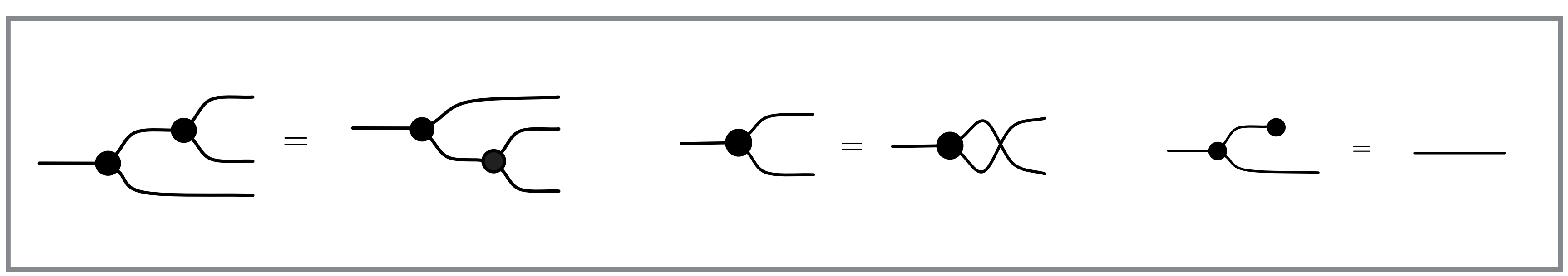
- basic tiles



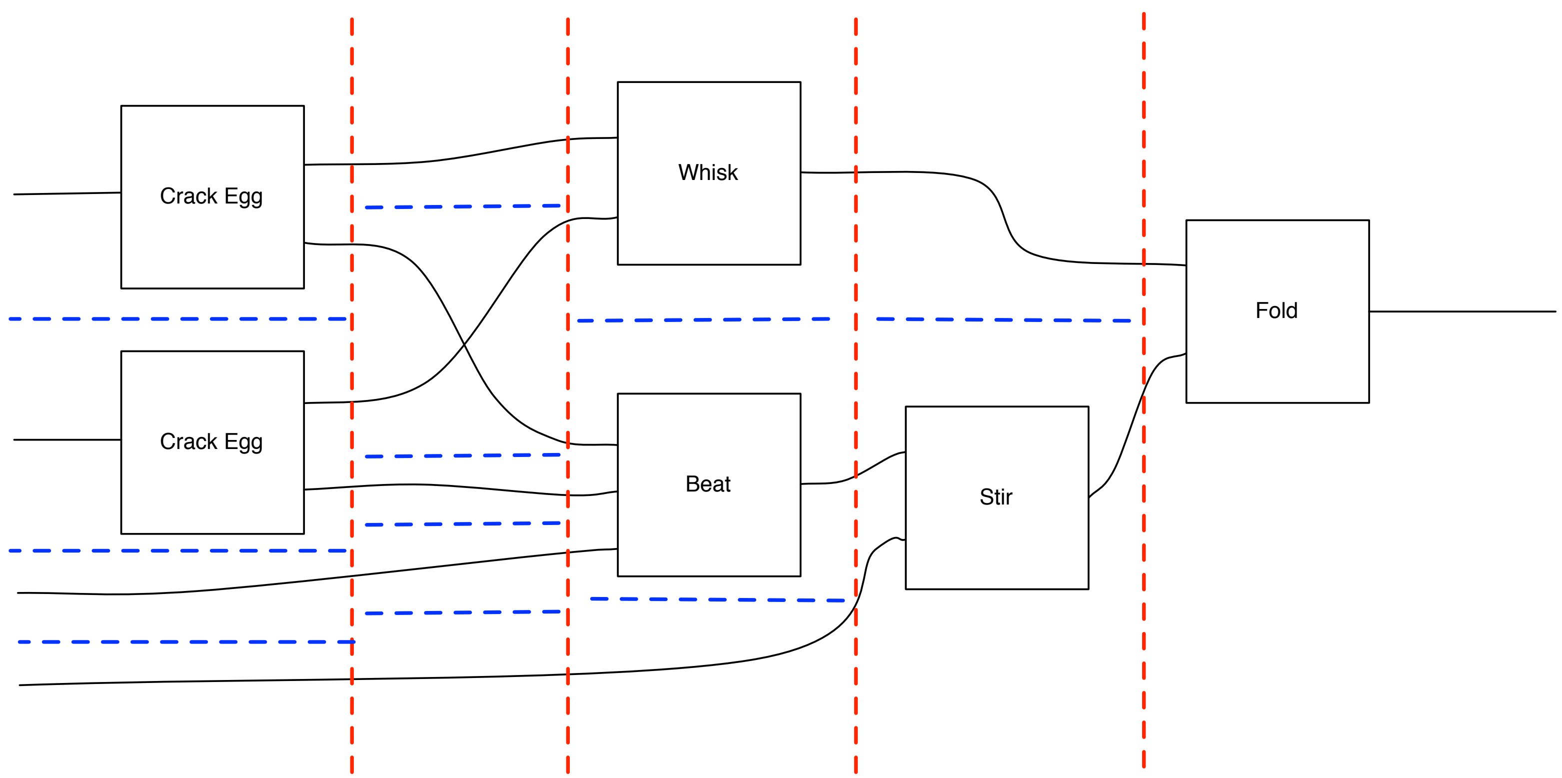
- algebra

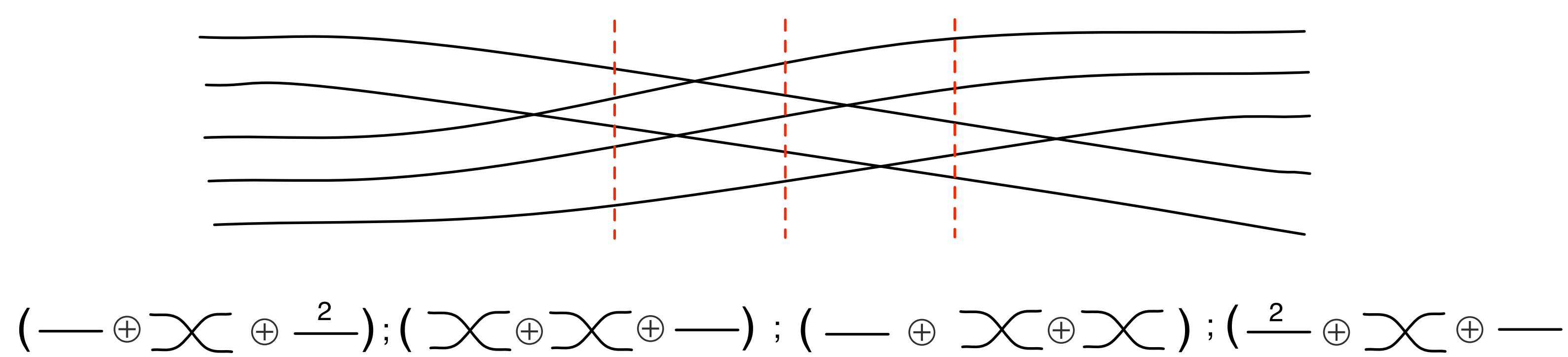


- equations (e.g.)



Drawing convention

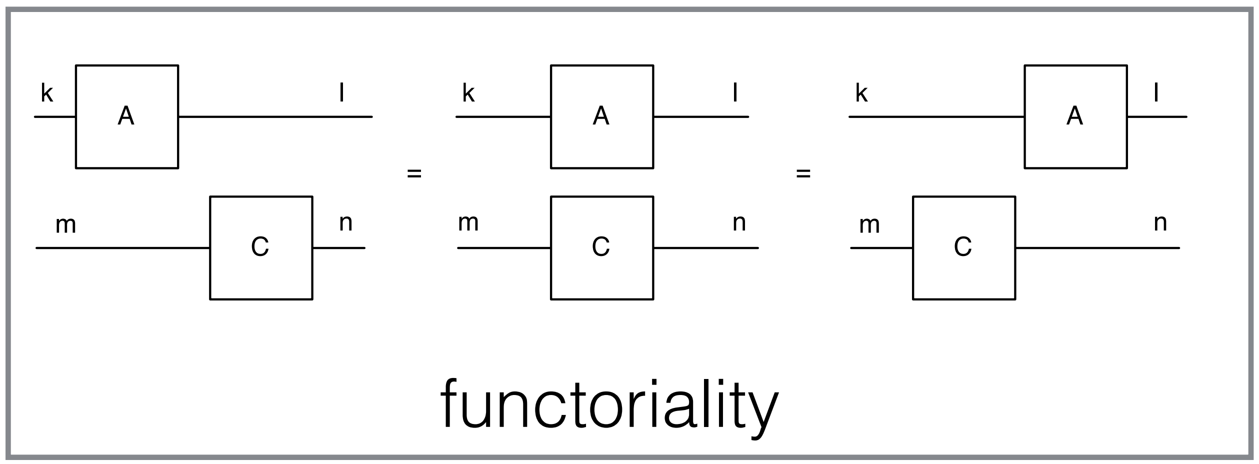


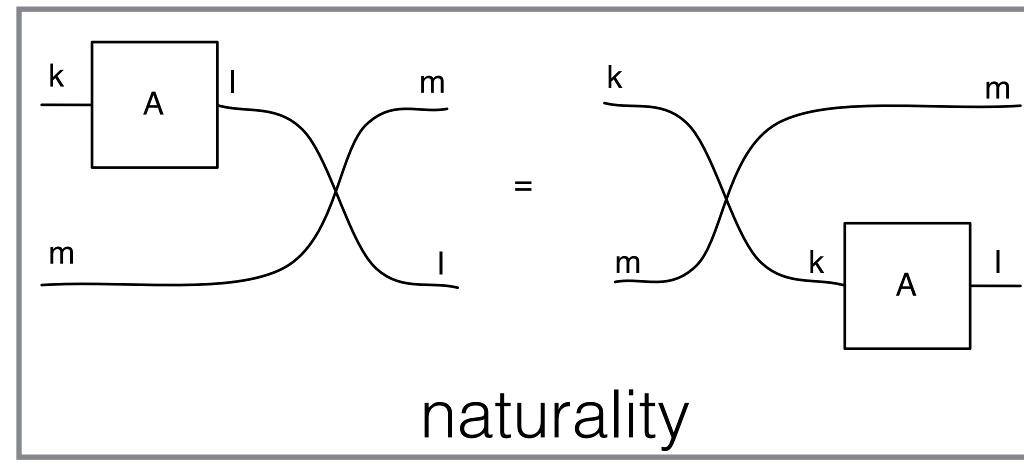


we want to have our cake (diagrams, useful for proofs) and eat it too (direct connection with terms)

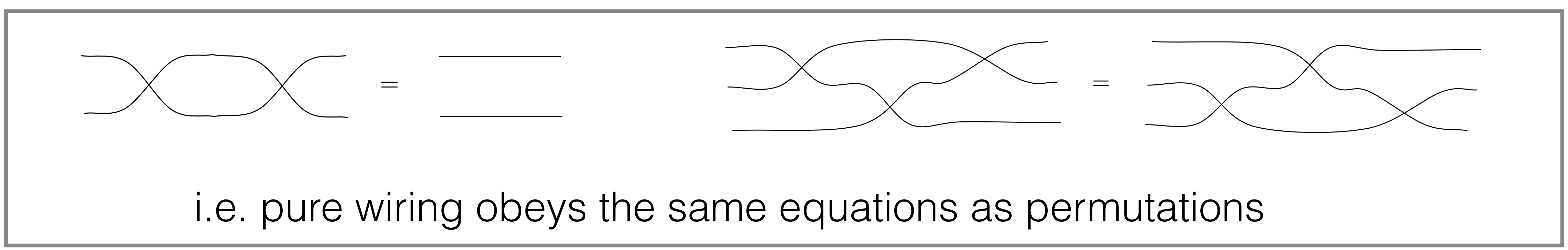
Diagrammatic Reasoning

- diagrams can slide along wires





- wires don't tangle, i.e.

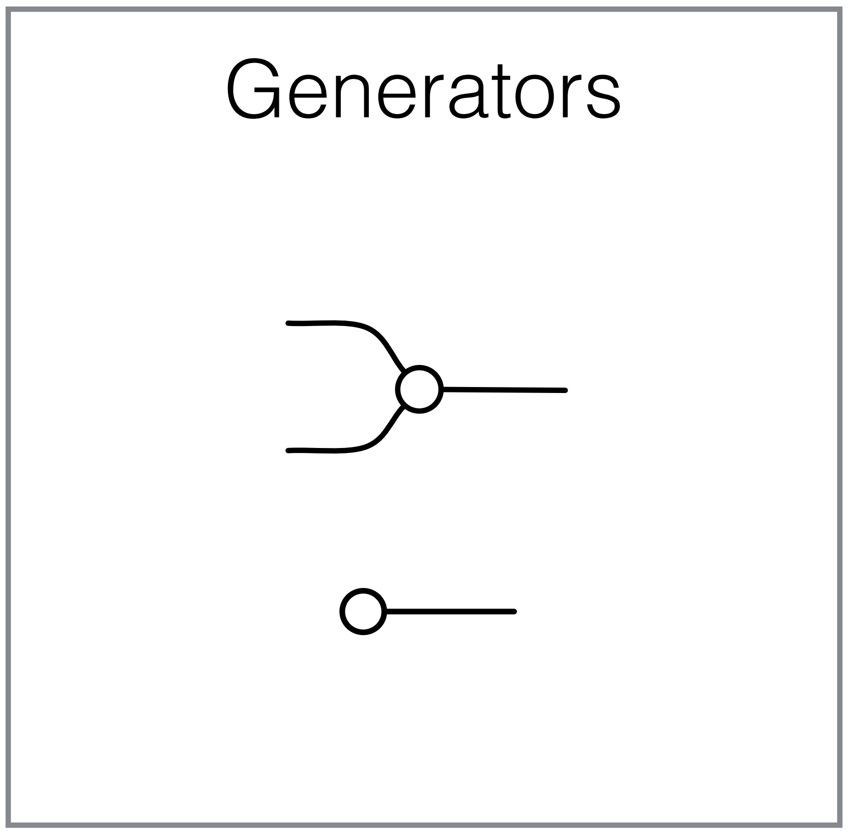


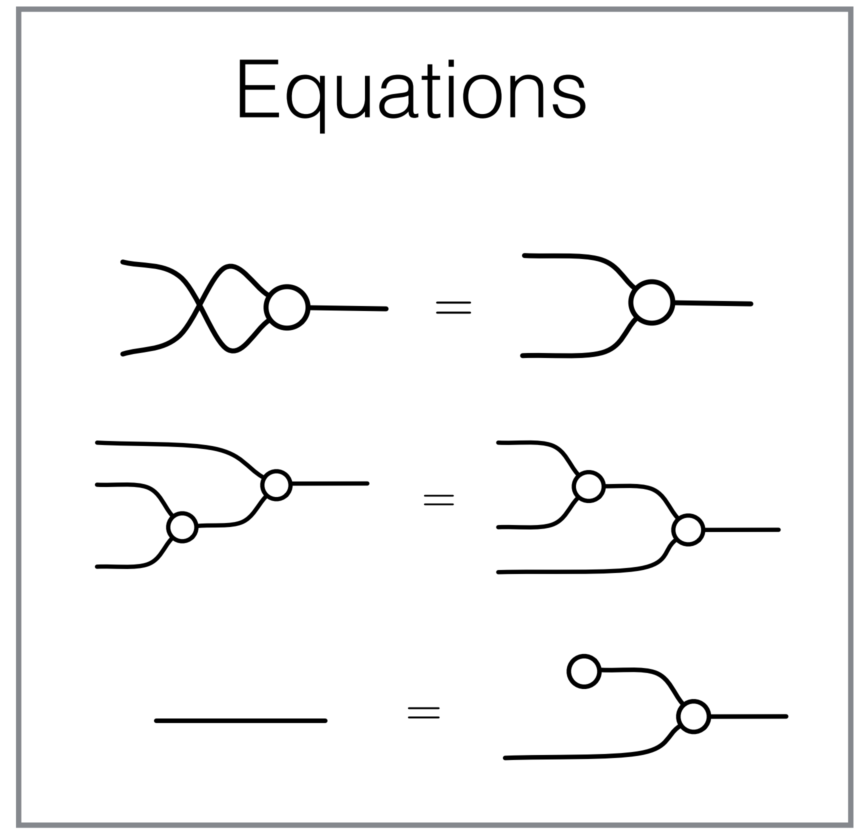
- sub-diagrams can be replaced with equal diagrams (compositionality)

PROPs and SMTs

- diagrammatic reasoning gives notion of equality on diagrams in an SMT
- in this way, every SMT is a PROP
  - natural to think of SMTs as **syntax**
  - other PROPs (like **F**) are **semantic domains**
  - homomorphisms assign semantics to syntax
- A **homomorphism of PROPs** is an identity-on-objects strict symmetric monoidal functor
  - the SMT with no generators and no equations is is isomorphic to the **initial PROP P** where arrows n to n are the **permutations** on [n]
  - the final PROP **1** has exactly one arrow from each m to n

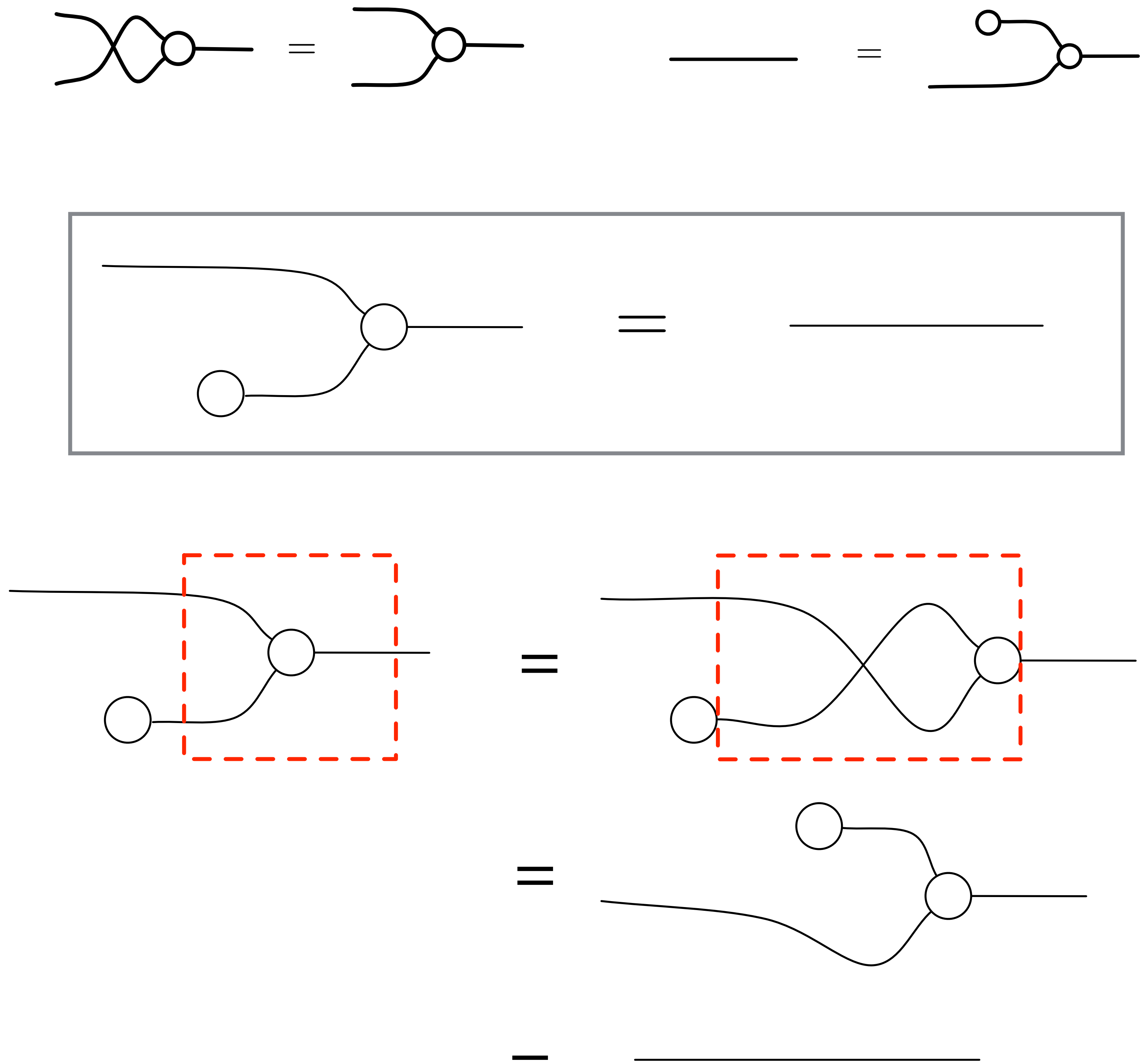
Example: commutative monoids



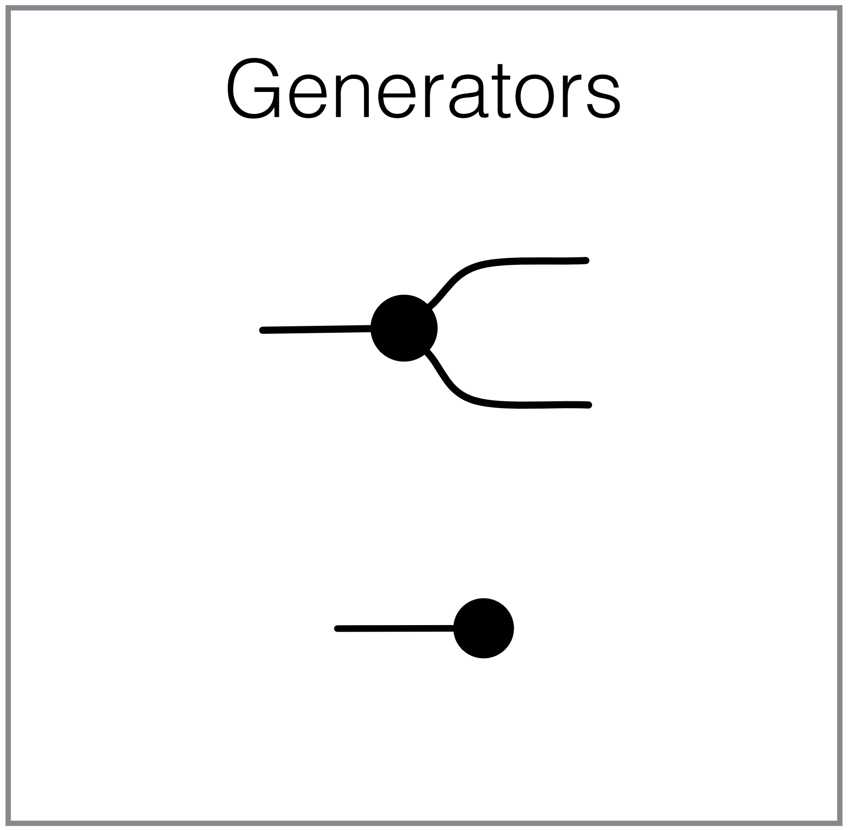


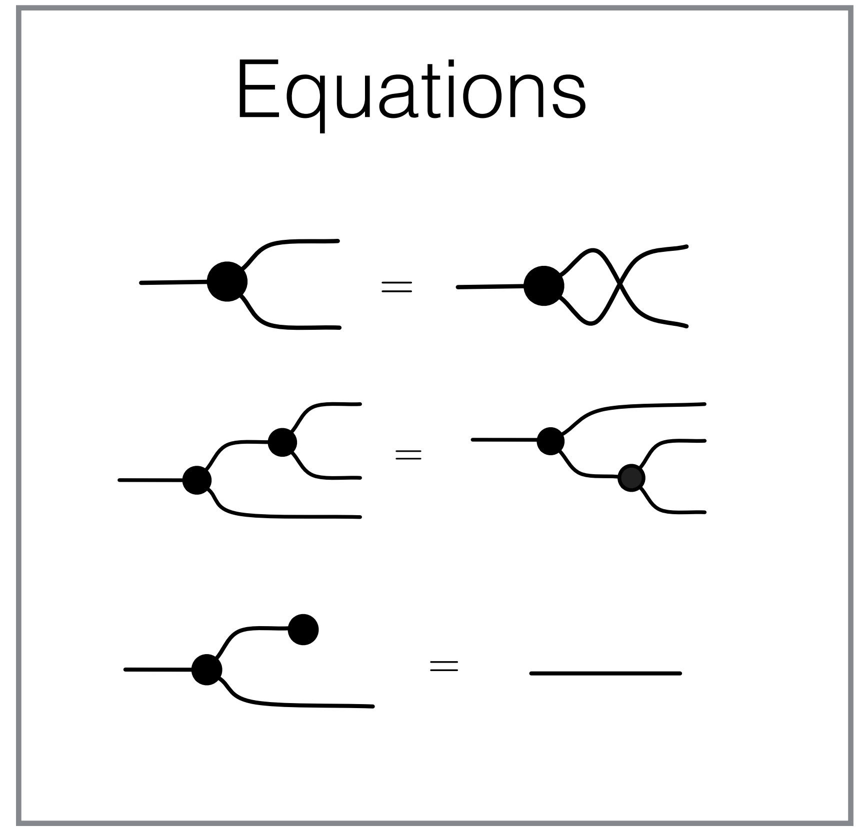
- SMT **M** on this data isomorphic to the PROP **F** of functions
- i.e. the “commutative monoids are the theory of functions”

Diagrammatic reasoning example



Example: commutative comonoids





- Isomorphic to $\mathbf{F}^{op}$
- NB departure from operads at this point: in an SMT generators of arbitrary arities and coarities are allowed

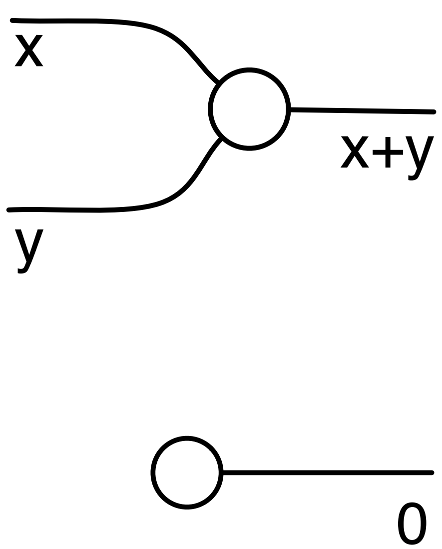
Plan

- basic theory of string diagrams
    - setup is slightly different to the usual Oxford lore
- **theory of natural number matrices (bimonoids) and integer matrices (Hopf monoids)**
    - intuition
    - bimonoids and matrices of natural numbers
    - Hopf monoids and matrices of integers
    - maths with diagrams
- theory of linear relations (interacting Hopf monoids)
- distributive laws
- linear algebra, diagrammatically
- an application: signal flow graphs

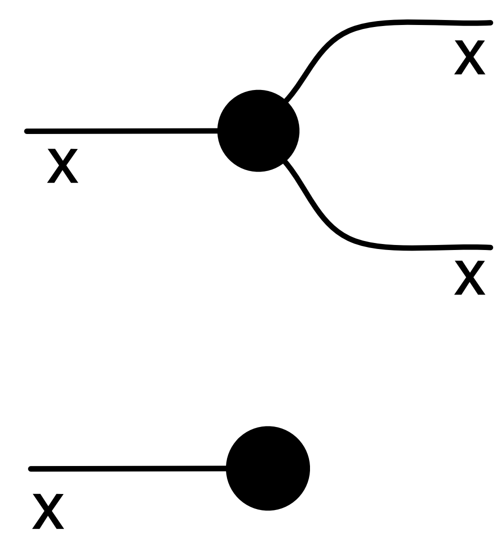
Useful intuition

- "numbers" travel on wires from left to right

The monoid structure acts as addition/zero

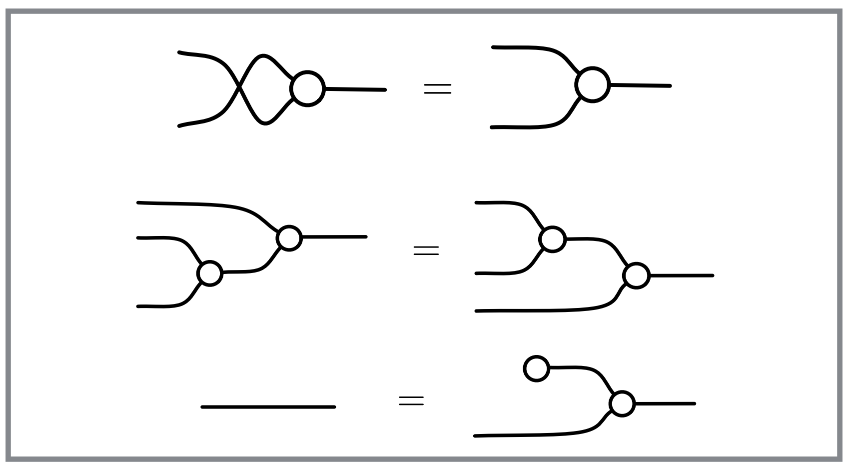


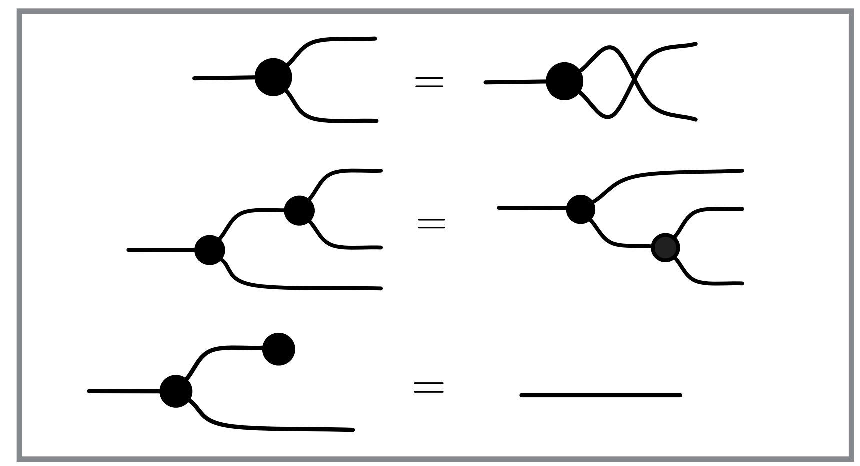
The comonoid structure acts as copying/discarding



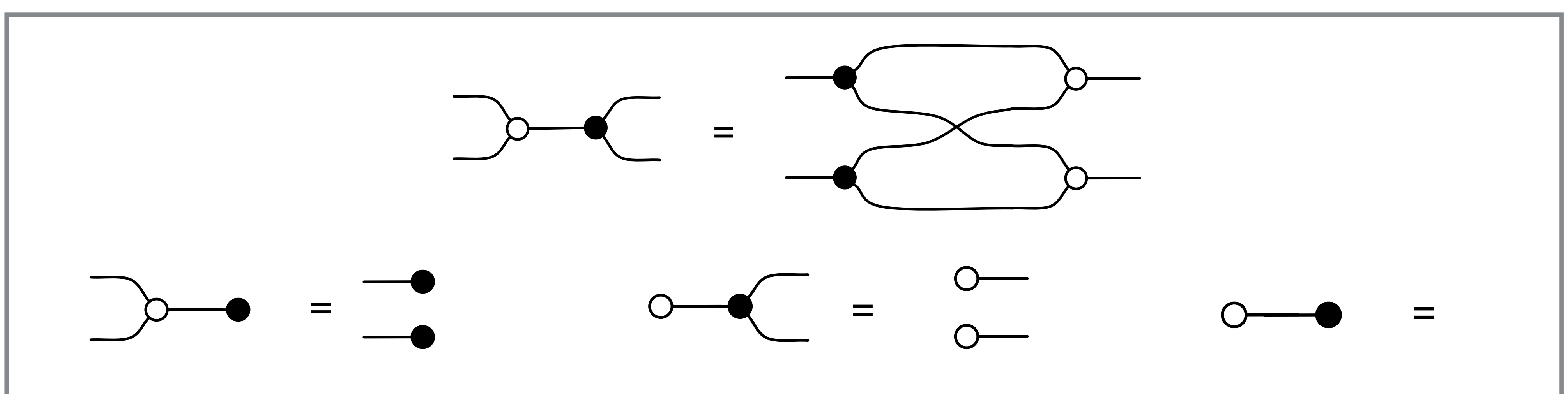
Bimonoids

- all the generators we have seen so far
- monoid and comonoid equations





- "adding meets copying" - equations compatible with intuition



Mat

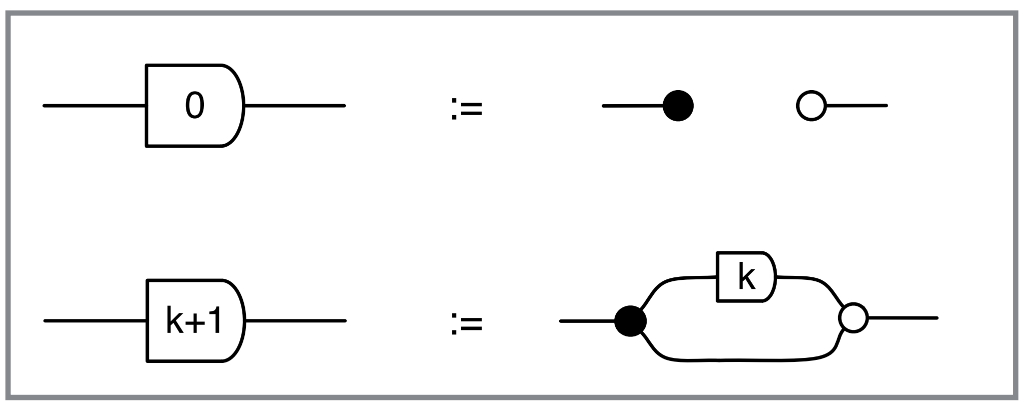
- A PROP where arrows m to n are n×m matrices of natural numbers
  - e.g. $\begin{pmatrix} 0 & 5 \end{pmatrix} : 2 \to 1$ $\begin{pmatrix} 3 \\ 15 \end{pmatrix} : 1 \to 2$ $\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} : 2 \to 2$
- Composition is matrix multiplication
- Monoidal product is direct sum

$$A_1 \oplus A_2 = \begin{pmatrix} A_1 & 0 \\ 0 & A_2 \end{pmatrix}$$

- Symmetries are permutation matrices

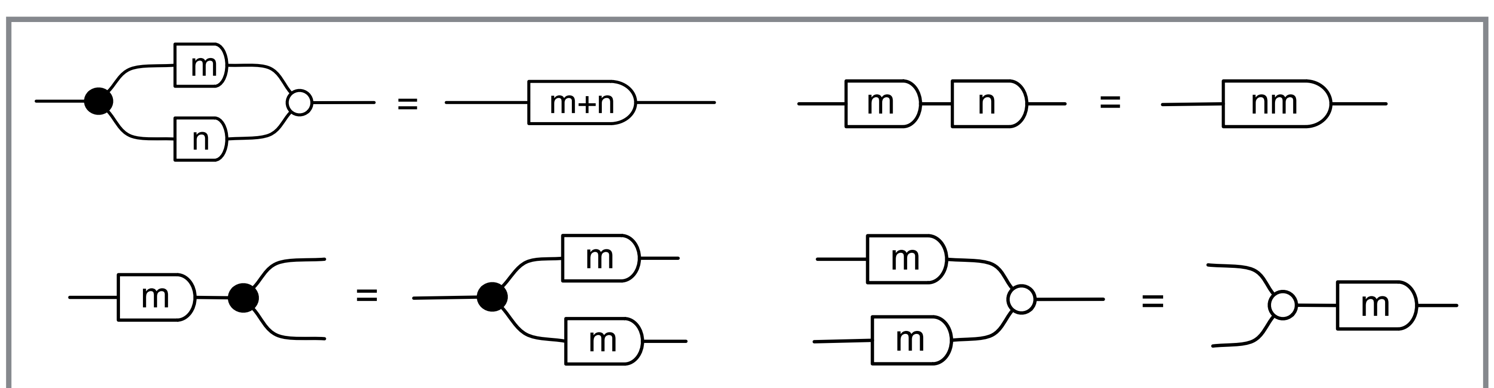
B and **Mat**

- **B** is isomorphic to the **Mat**
  - ie. bimonoids is the theory of natural number matrices
- natural numbers can be seen as certain (1,1) diagrams, with recursive defn



+1 is “add one path”

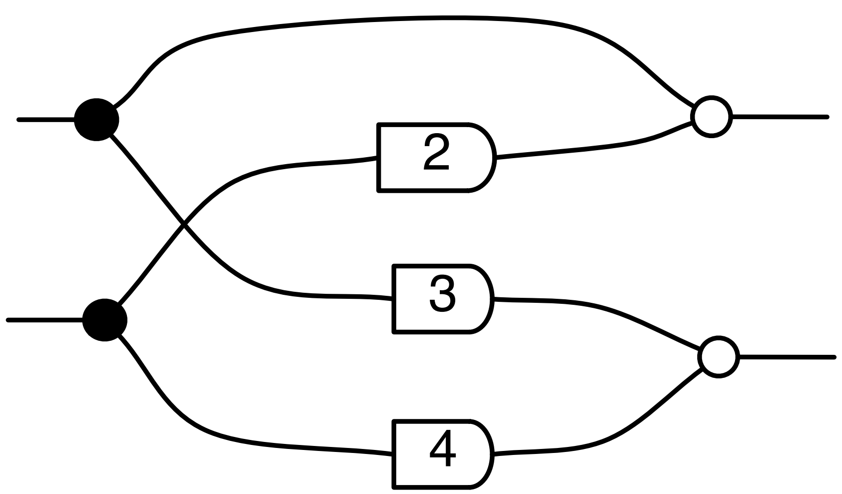
- the algebra (rig) of natural numbers follows; the following are easy inductions



Matrices

- To get the ij*th* entry in the matrix, count the paths from the j*th* port on the left to the i*th* port on the right

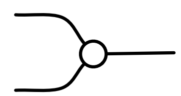
- Example:

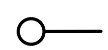


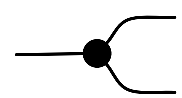
$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$$

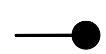
Proof **B** ≅ **Mat**

Since **B** is an SMT, suffices to say where generators go
(and check that equations hold in the codomain)

 $\mapsto \quad \begin{pmatrix} 1 & 1 \end{pmatrix} : 2 \to 1$

 $\mapsto \quad () : 0 \to 1$

 $\mapsto \quad \begin{pmatrix} 1 \\ 1 \end{pmatrix} : 1 \to 2$

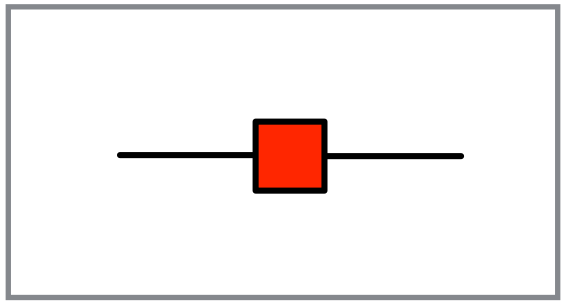
 $\mapsto \quad () : 1 \to 0$

Full - easy!
Recursively define a syntactic sugar for matrices

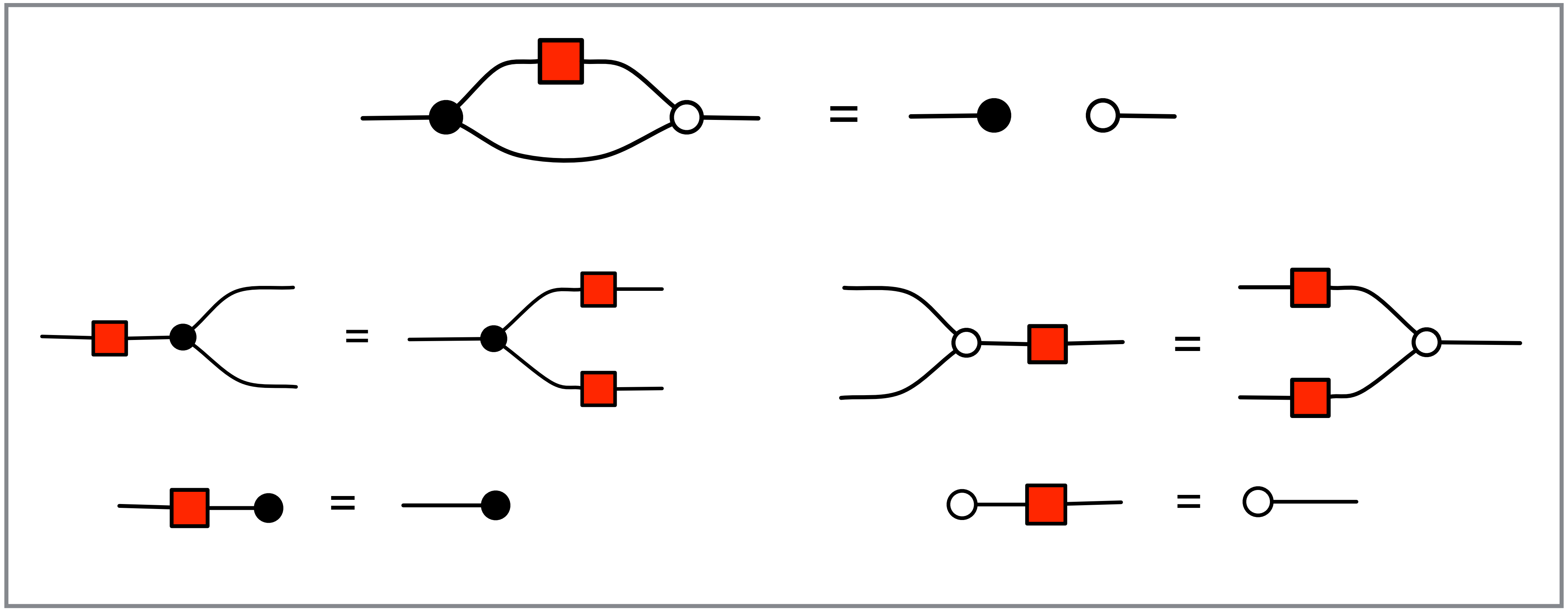
Faithful - little bit harder
Use the fact that equations are a presentation of a distributive law, obtain factorisation of diagrams as comonoid structure followed by monoid structure

Putting the n in ring: Hopf monoids

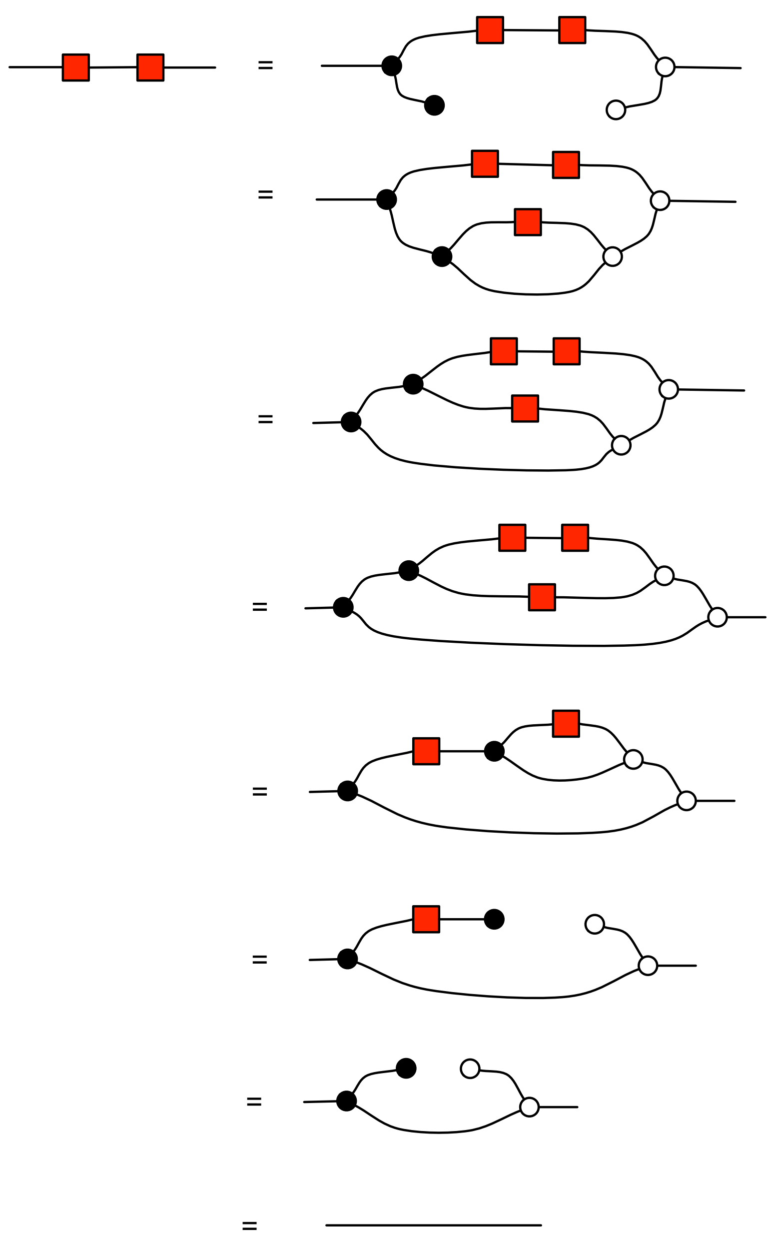
- generators of bimonoids + **antipode**



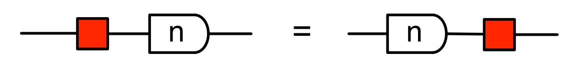
- equations of bimonoids + the following



The ring of integers

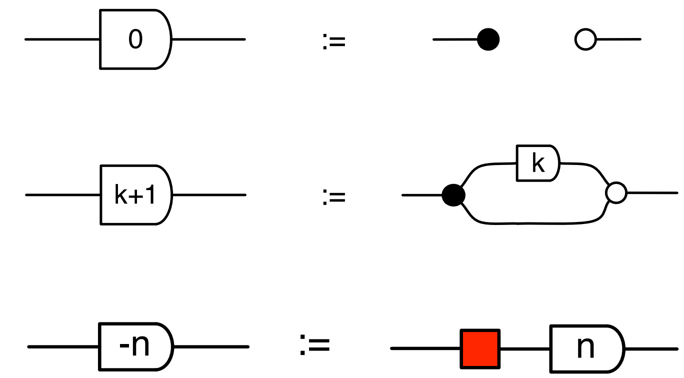


-1 · -1 = 1

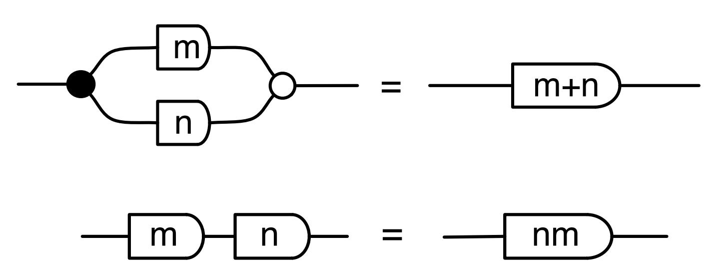


simple induction

in **B**, the naturals were (1,1) diagrams
in **H**, the integers are the (1,1) diagrams



Just as for nats, we have



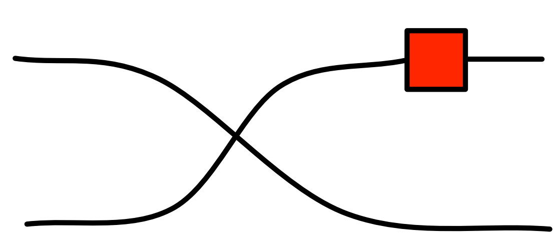
etc.

$\mathbf{Mat}_{\mathbf{Z}}$

- Arrows m to n are n×m matrices of integers
  - composition is matrix multiplication
  - monoidal product is direct sum
- $\mathbf{Mat}_{\mathbf{Z}}$ is equivalent to the category of finite dimensional free **Z**-modules
- SMT **H** is isomorphic to the PROP $\mathbf{Mat}_{\mathbf{Z}}$

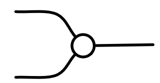
Path counting in MatZ

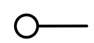
- To get the ij*th* entry in the matrix, count the
  - positive paths from the j*th* port on the left to the i*th* port on the right (where antipode appears an even number of times)
  - negative paths between these two ports (where antipode appears an odd number of times)
  - subtract the negative paths from the positive paths
- Example:

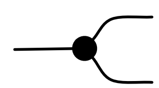


$$\begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$$

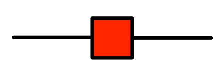
Proof $\mathbf{H} \cong \mathbf{Mat}_{\mathbf{Z}}$

 $\mapsto \begin{pmatrix} 1 & 1 \end{pmatrix} : 2 \to 1$

 $\mapsto () : 0 \to 1$

 $\mapsto \begin{pmatrix} 1 \\ 1 \end{pmatrix} : 1 \to 2$

 $\mapsto () : 1 \to 0$

 $\mapsto (-1) : 1 \to 1$

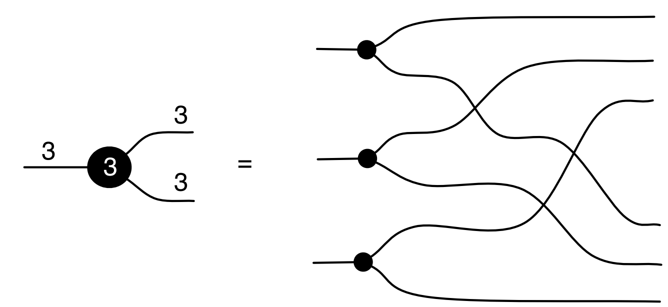
- Fullness easy
- Faithfulness more challenging: put diagrams in the form

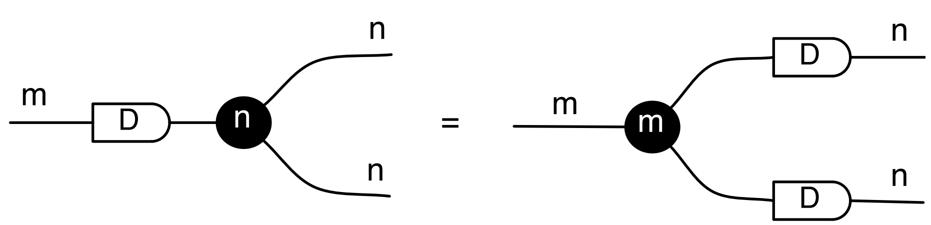
copying ; antipode ; adding

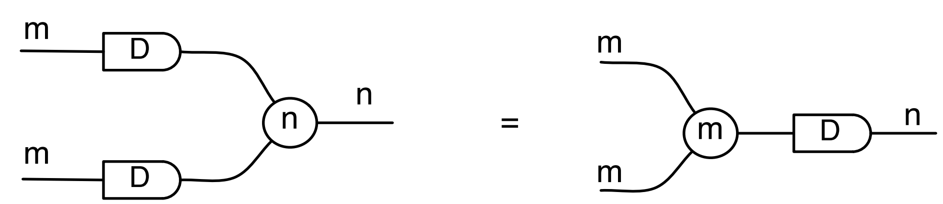
Maths with diagrams

- we focussed on (1,1) for historical reasons

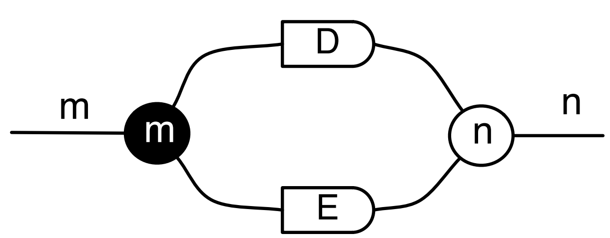
eg



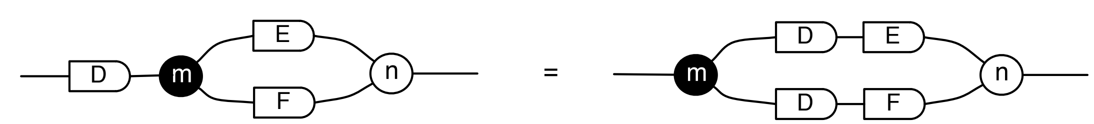




D + E :=



associative, commutative with unit
has additive inverse in **H**



multiplication through composition,
addition distributes on both sides

Plan

- basic theory of string diagrams
- theory of natural number matrices (bimonoids) and integer matrices (Hopf monoids)
- **theory of linear relations (interacting Hopf monoids)**
  - intuition upgrade
  - the equations of **IH**
  - linear relations
  - rational numbers, diagrammatically
- distributive laws
- linear algebra, diagrammatically
- an application: signal flow graphs

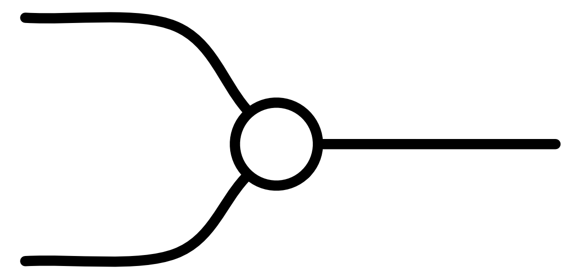
Intuition upgrade

- We have been saying that numbers go from left to right in diagrams
  - this is a **functional**, input/output interpretation

> The input/output framework is totally inappropriate for dealing with all but the most special system interconnections. [The input/output representation] often needlessly complicates matters, mathematically and conceptually. A good theory of systems takes the behavior as the basic notion.
>
> J.C. Willems, *Linear systems in discrete time,* 2009

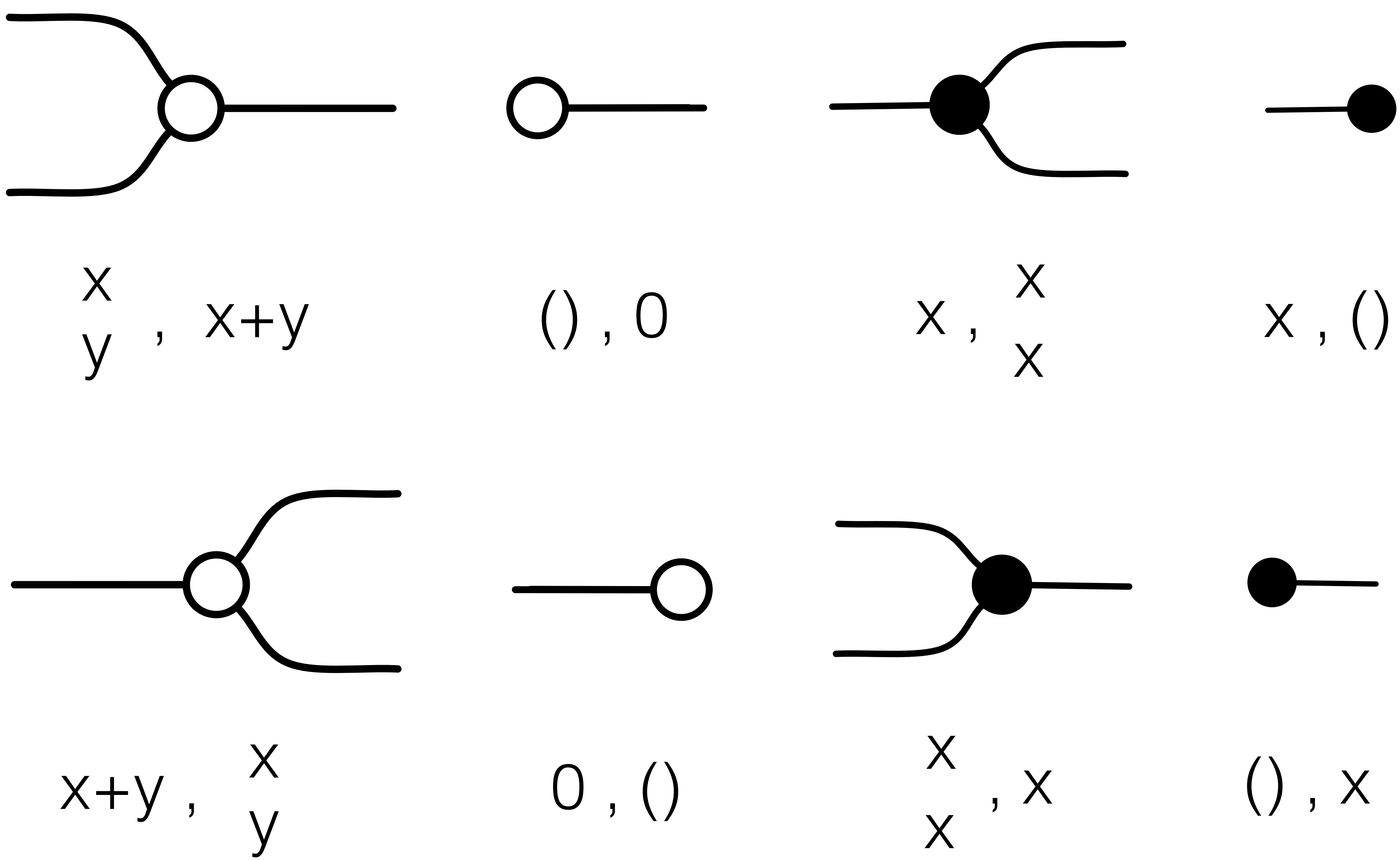
- From now on, we will take a **relational** point of view, a diagram is a contract that allows certain numbers to appear on the left and on the right

Intuition upgrade

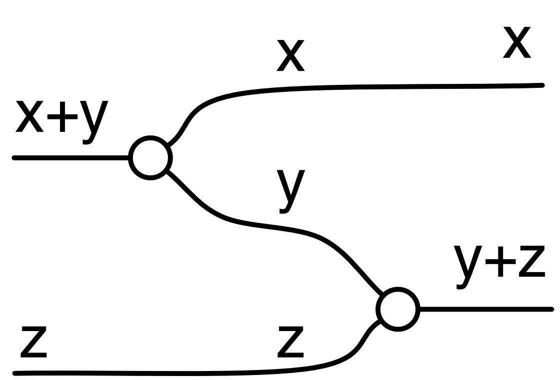


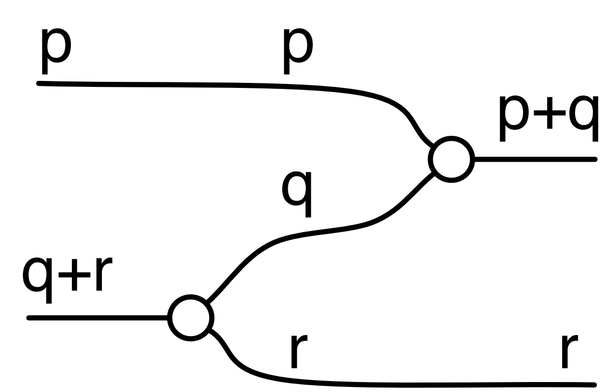
- Intuition so far is this as a function +: $D \times D \to D$
- From now it will be as a relation of type $D \times D \to D$
- Composition is relational composition

Example



Adding meets adding





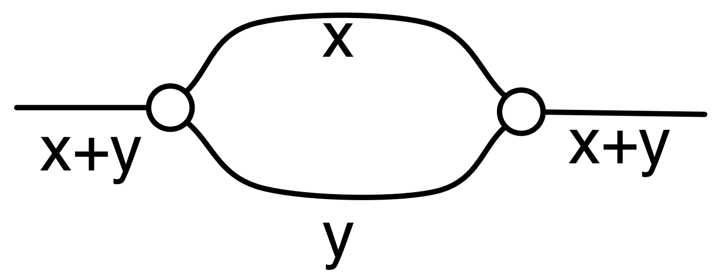
$x = p+q$ $p=x+y$

$z = q+r$ $r=y+z$

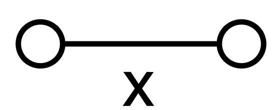
⇒ $y=-q$

Provided addition yields abelian group (i.e. there are additive inverses), the two are **the same** relation

More adding meets adding

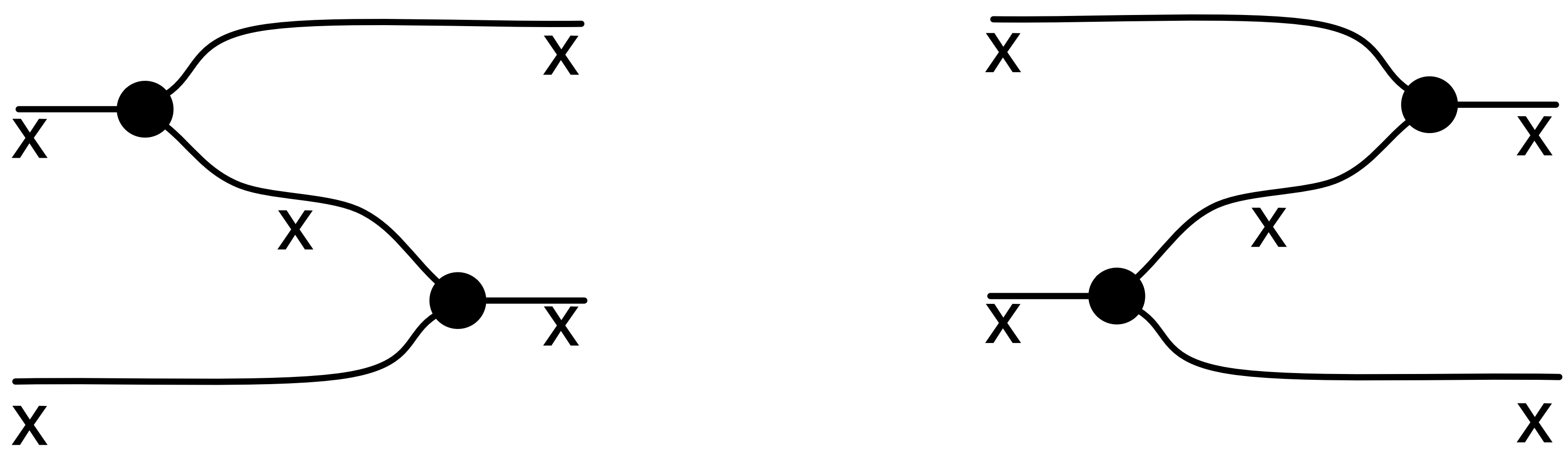


since x and y are free, this is the identity relation

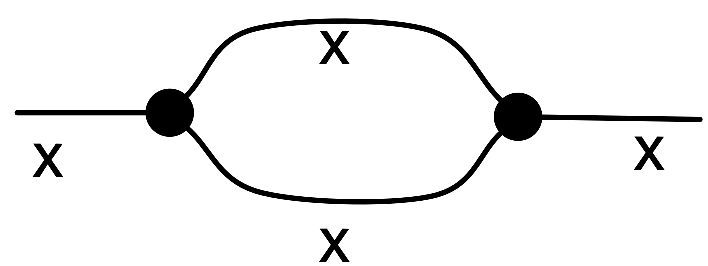


empty relation

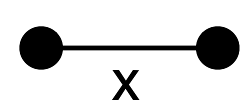
Copying meets copying



clearly both give the same relation

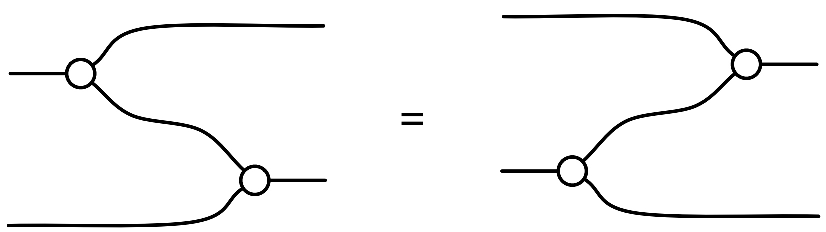


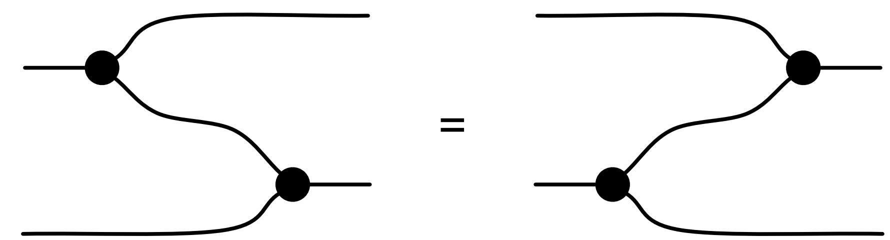
identity relation



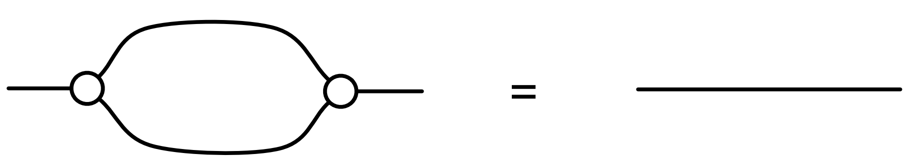
empty relation

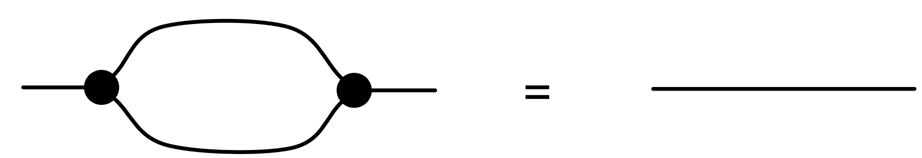
Two Frobenius structures



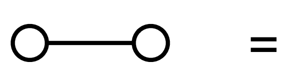


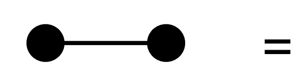
+ special / strongly separable equations





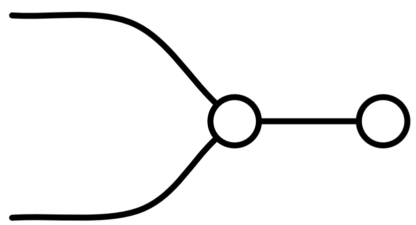
+ “bone” equations

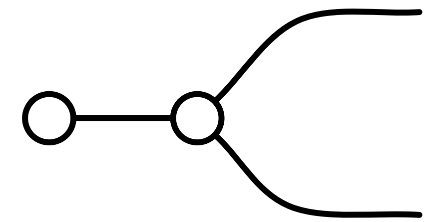




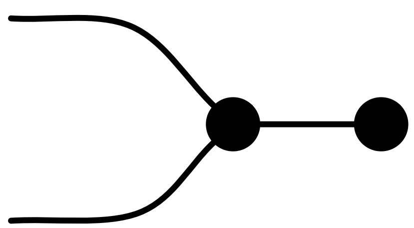
Two self-dual compact closed structures

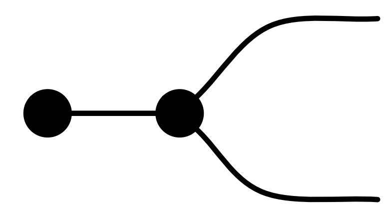
(*cf.* cups and caps)



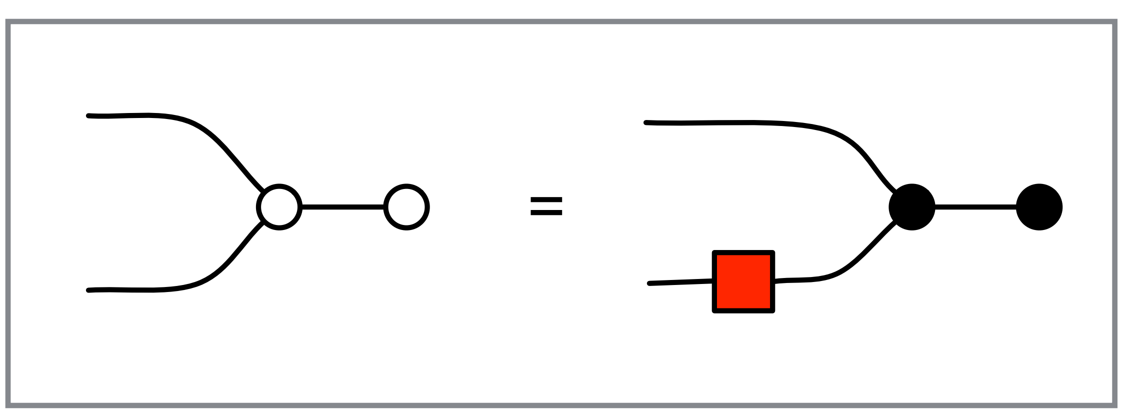


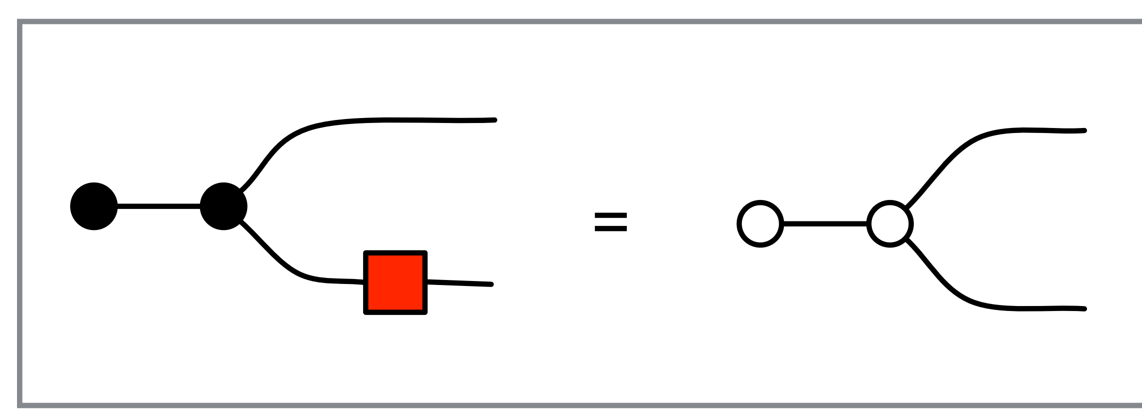
$$\{(\begin{pmatrix} x \\ y \end{pmatrix}, ()) \mid x + y = 0\}$$



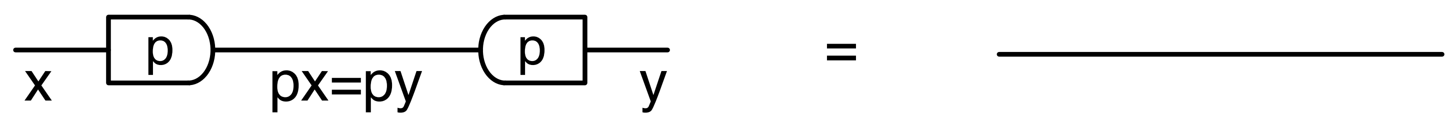


$$\{(\begin{pmatrix} x \\ x \end{pmatrix}, ())\}$$

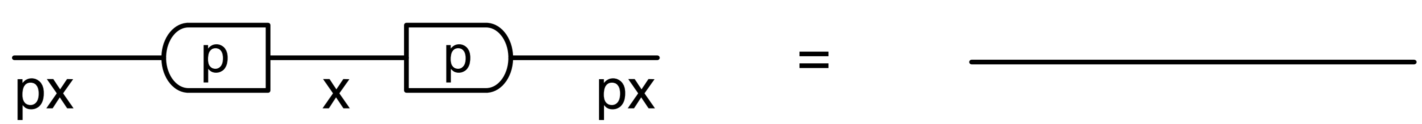




Scalars meet scalars



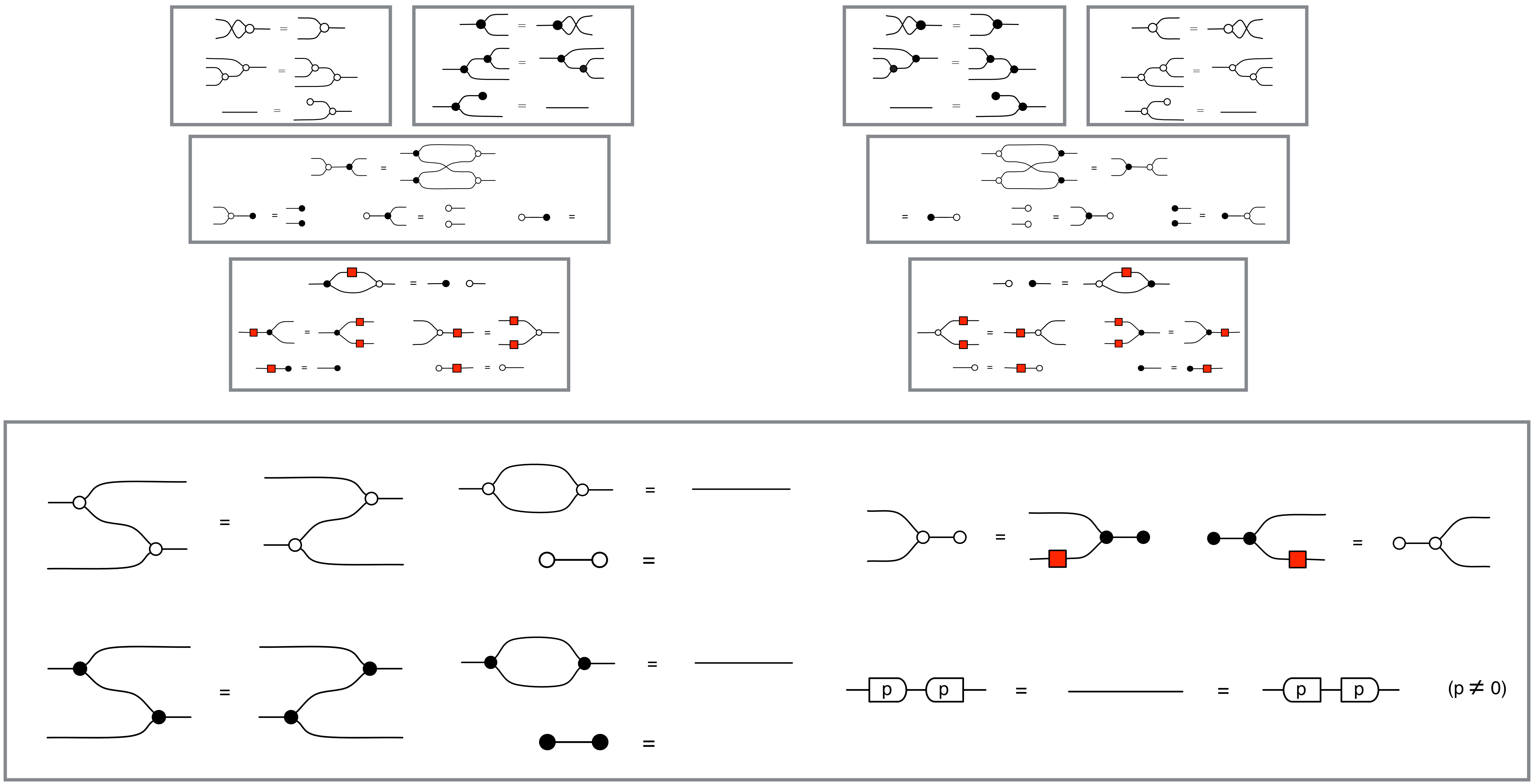
if multiplication on the left by p is injective
(e.g. if p ≠ 0 in a field)



if multiplication on the left by p is surjective
(e.g. if p ≠ 0 in a field)

Interacting Hopf Monoids

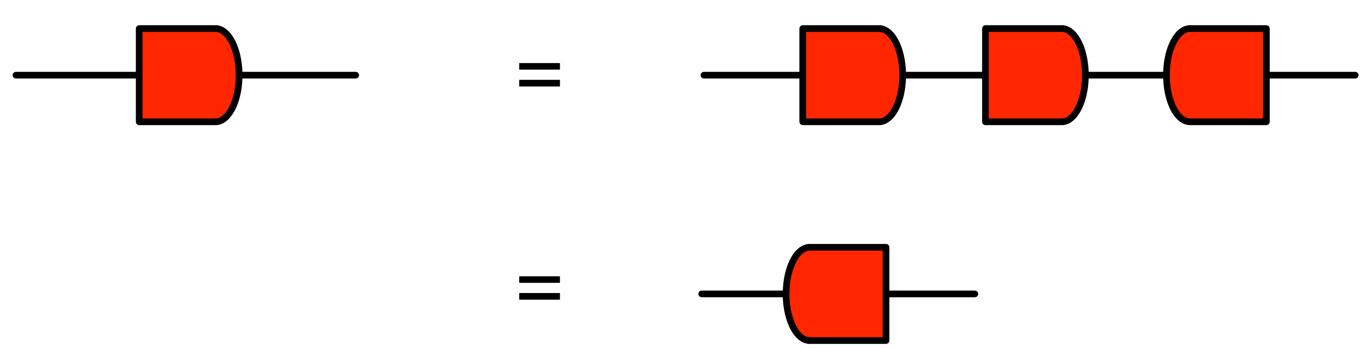
(Bonchi, S., Zanasi, '13, '14)



(*cf.* ZX-calculus, Coecke and Duncan '08, Baez and Erbele '14)

The antipode cheat

The antipodes in **H** and **H**^{op} are formally different but we were slightly naughty with notation.



Two daggers

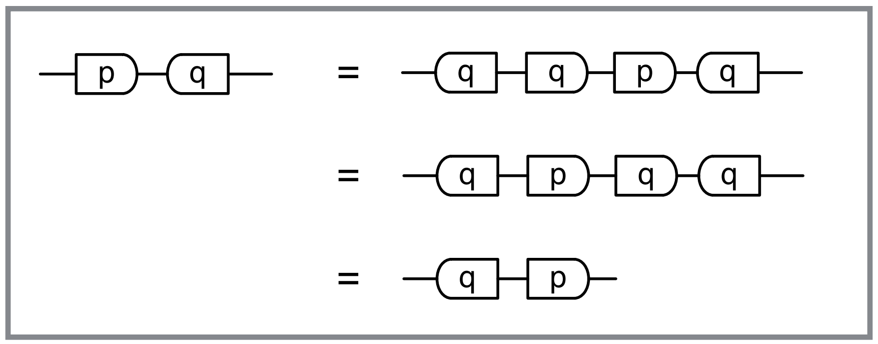
- 1. “**opposite**”
  - left goes to right
  - takes matrix (diagram in **H** or $\mathbf{H}^{op}$) to its opposite
  - takes a linear relation to its opposite
- 2. “**bizarro**”
  - left goes to right and
  - black goes to white
  - takes matrix (diagram in **H** or $\mathbf{H}^{op}$) to its transpose
  - On diagrams (n,0) it gives the orthogonal space (but type is (0,n))

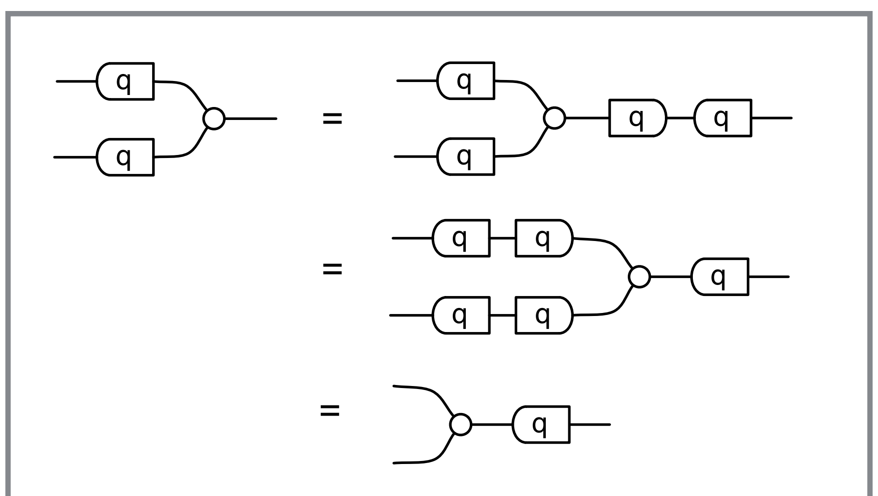
LinRel

- PROP of linear relations over the rationals
  - arrows m to n are subspaces of $\mathbf{Q}^m \times \mathbf{Q}^n$
  - composed **as relations**
  - monoidal product is direct sum
- **IH** is isomorphic to **LinRel**
  - we will prove this tomorrow

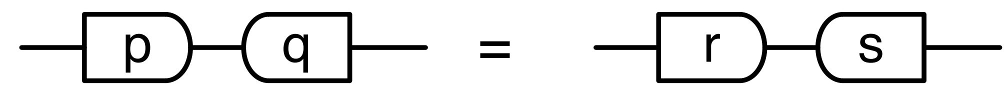
Where did the rationals come from?

if $q \neq 0$:



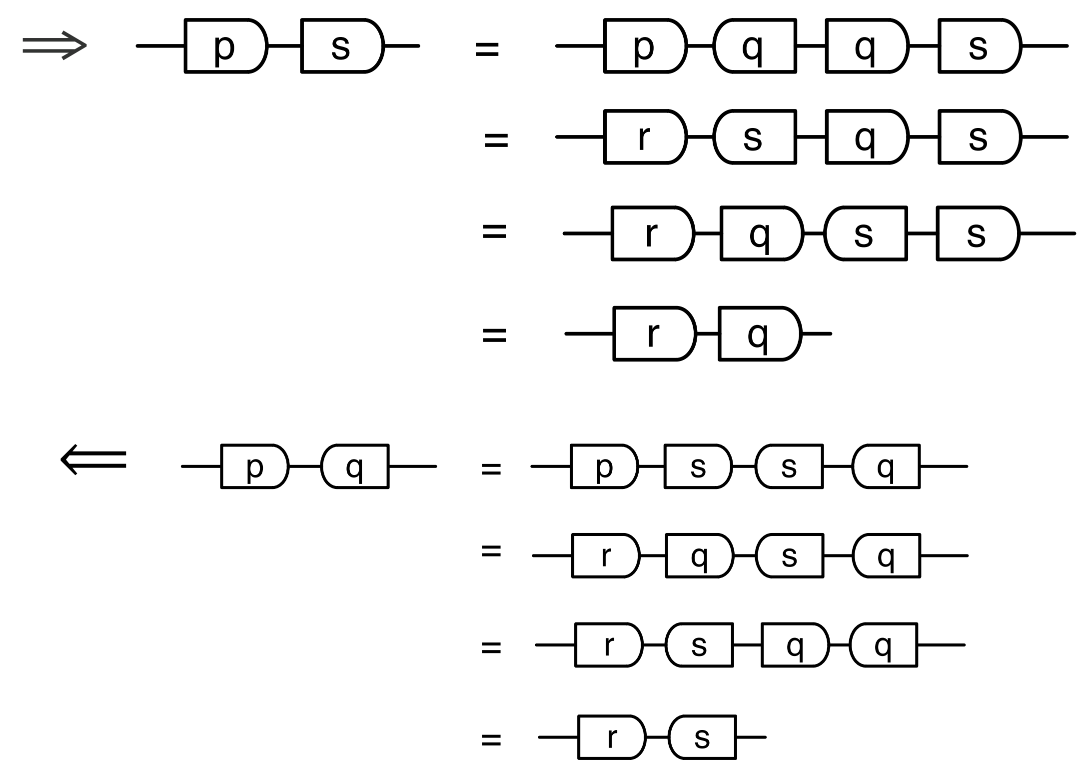


suppose $q, s \neq 0$:



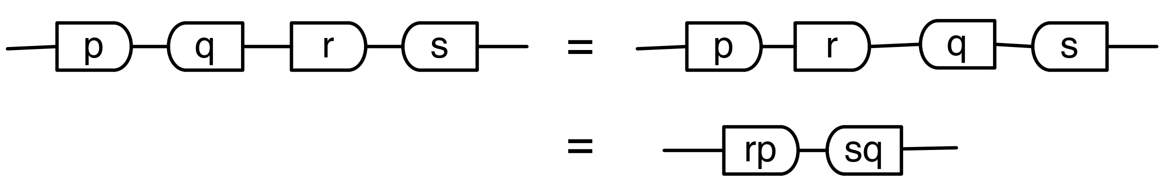
$\Leftrightarrow$

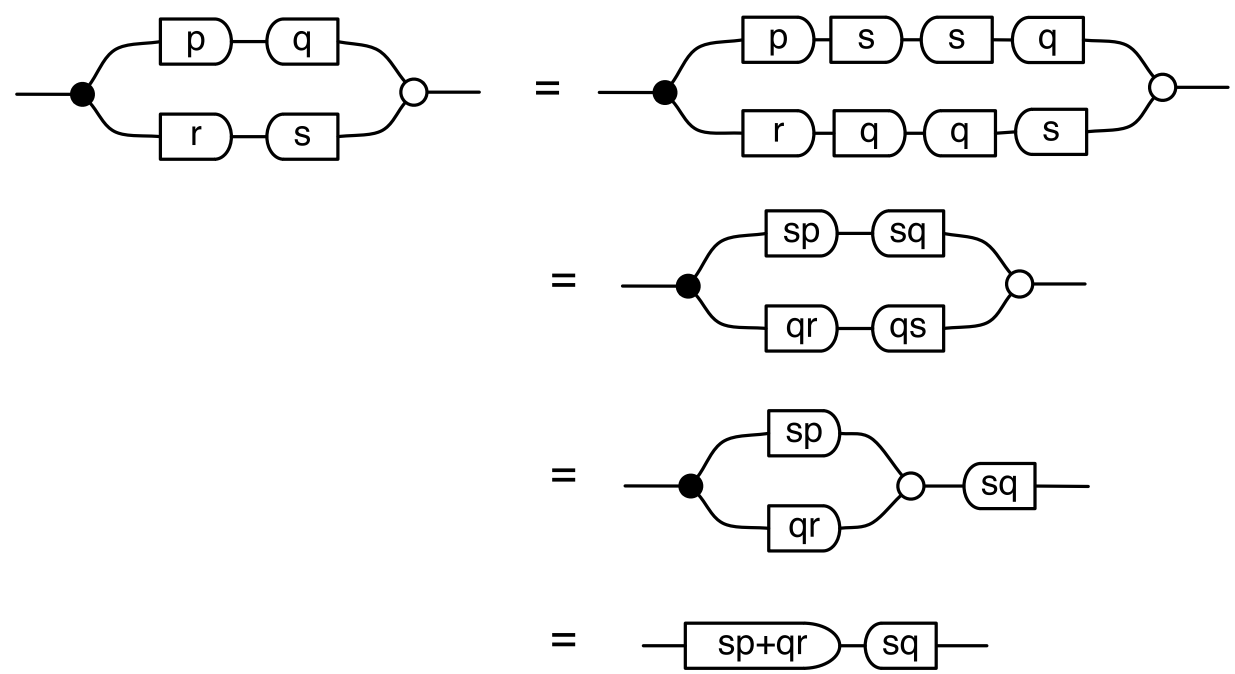
sp = qr



Rational arithmetic

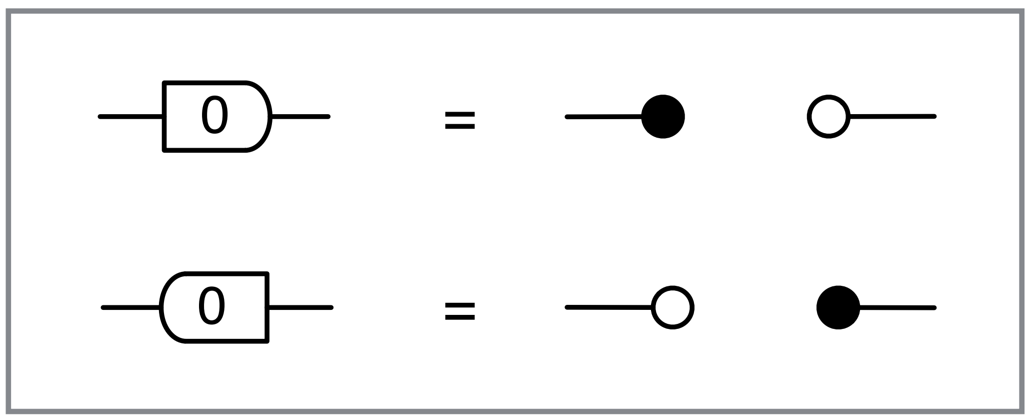
(q,s ≠ 0)





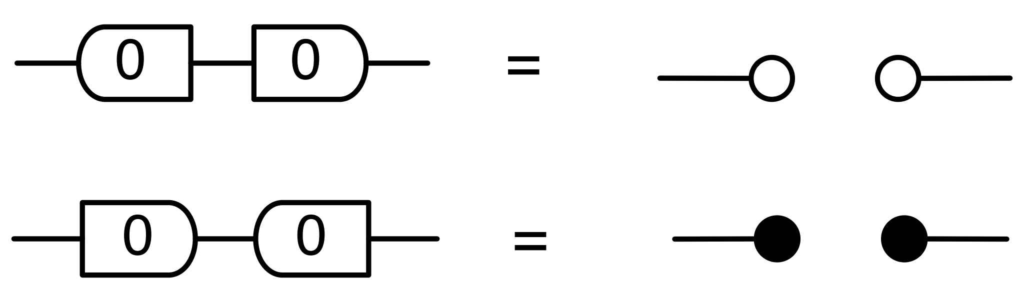
Keep calm and divide by zero

- it's ok, nothing blows up



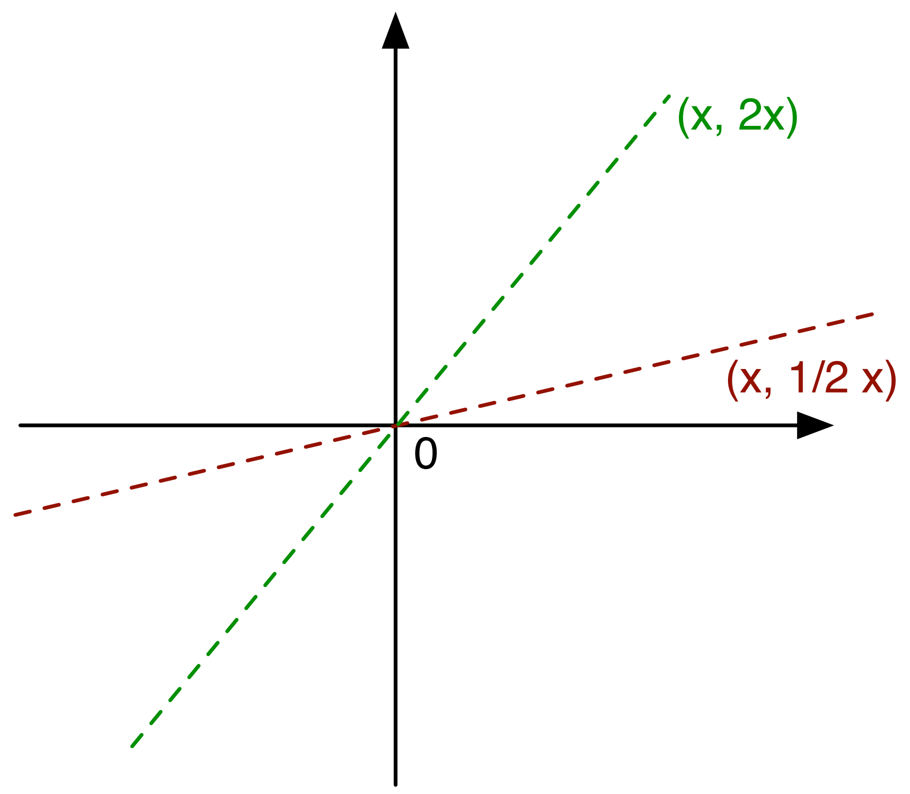
- of course, arithmetic with 1/0 is not quite as nice as with proper rationals.

- two ways of interpreting 0/0 (0 · /0 or /0 · 0)



Projective arithmetic++

- Projective arithmetic identifies numbers with one-dimensional spaces (lines) of $\mathbf{Q}^2$
    - one for each rational p : $\{ (x,px) \mid x \in \mathbf{Q} \}$
    - and "infinity" : $\{ (0, x) \mid x \in \mathbf{Q} \}$
- The extended system includes all the subspaces of $\mathbf{Q}^2$, in particular:
    - the unique zero dimensional space $\{ (0, 0) \}$
    - the unique two dimensional space $\{ (x,y) \mid x,y \in \mathbf{Q} \}$

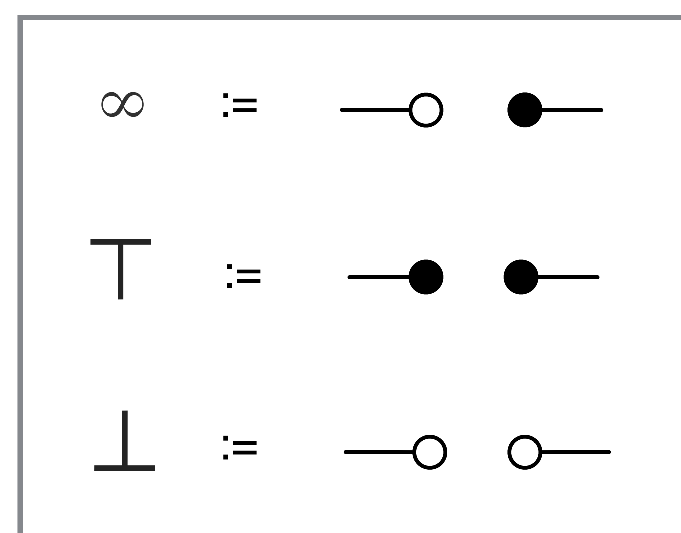


Dividing by zero

Edalat and Potts suggested that two extra ‘numbers’, $\infty = 1/0$ and $\perp = 0/0$, be adjoined to the set of real numbers (thus obtaining what in domain theory is called the ‘lifting’ of the real projective line) in order to make division always possible. In a seminar, Martin-Löf proposed that **one should try to include these ‘numbers’ already in the construction of the rationals from the integers, by allowing not only non-zero denominators, but arbitrary denominators**, thus ending up not with a field, but with a field with two extra elements.

Jesper Carlström, *Wheels, On Division by Zero*, 2001

Here we have three extra elements!

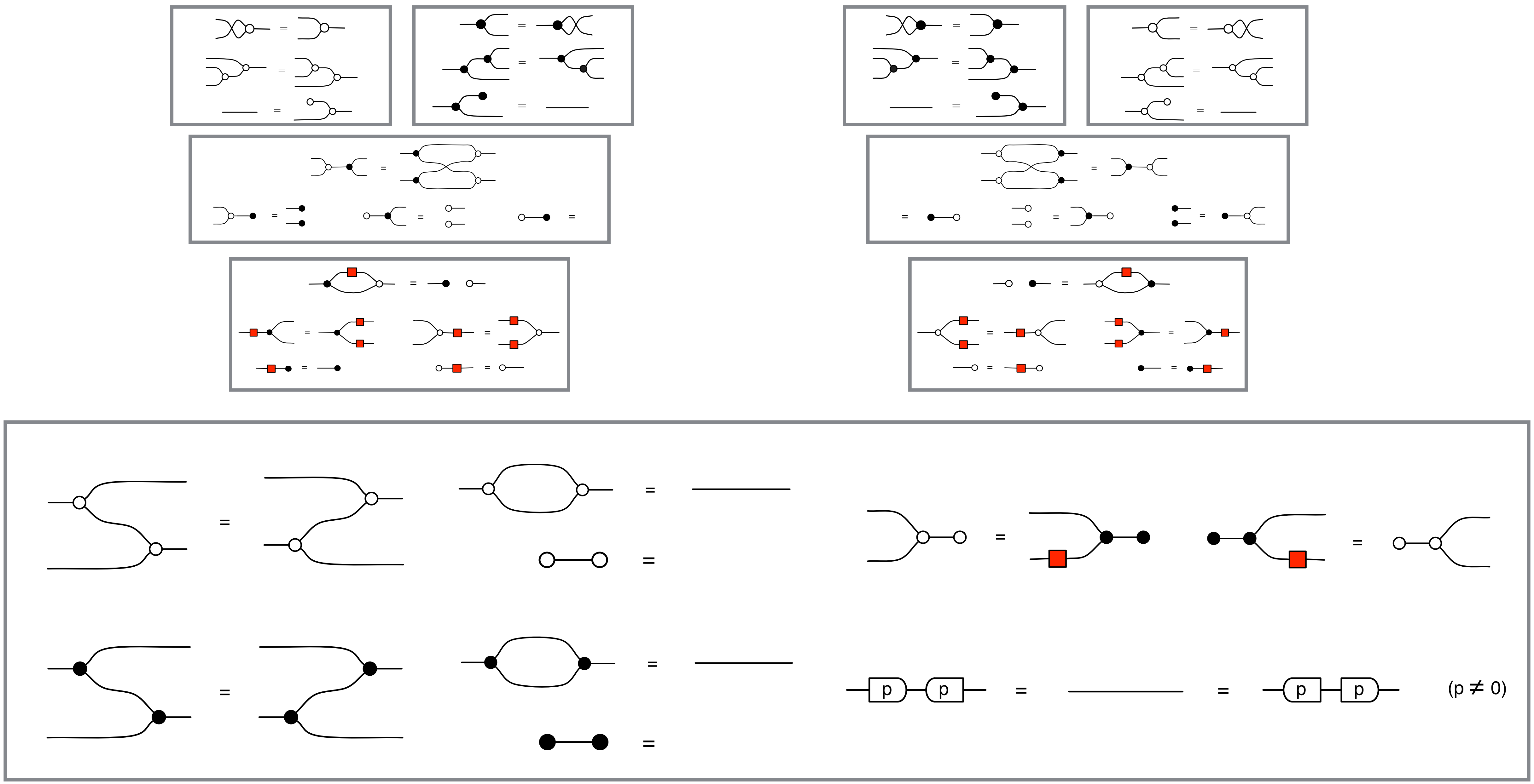


Plan

- basic theory of string diagrams
- theory of natural number matrices (bimonoids) and integer matrices (Hopf monoids)
- theory of linear relations (interacting Hopf monoids)
- **distributive laws**
- linear algebra, diagrammatically
- an application: signal flow graphs

Interacting Hopf Monoids

(Bonchi, S., Zanasi, ’13, ’14)



(*cf.* ZX-calculus, Coecke and Duncan ’08, Baez and Erbele ‘14)

Distributive laws of PROPs

- Proof **IH** ≅ **LinRel** relies on the notion of **distributive law** of PROPs (Lack, *Composing PROPs*, 2004)
  - a variant of distributive laws of **monads**
  - monads can be considered in any 2-category (R. Street, *Formal Theory of Monads,* 1972)
    - categories = monads in Span(**Set**)
    - strict monoidal categories = monads in Span(**Mon**)
    - small technical complications for PROPs because of symmetries

Categories = Monads??

- What is a monad in Span(**Set**)?
  - endo 1-cell $O \xleftarrow{\delta_0} A \xrightarrow{\delta_1} O$
  - multiplication

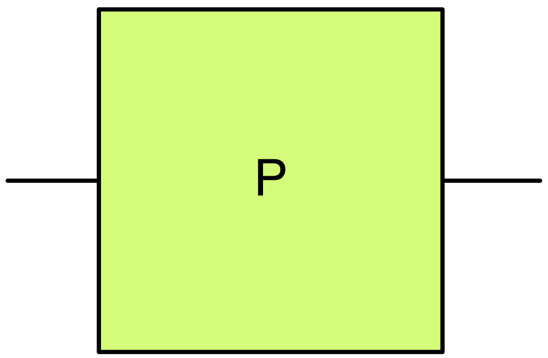
$$\begin{array}{ccc} A \times_O A & \longrightarrow & A \\ \downarrow & & \downarrow \delta_1 \\ A & \underset{\delta_0}{\longrightarrow} & O \end{array}$$

$$A \times_O A \xrightarrow{\mu} A$$

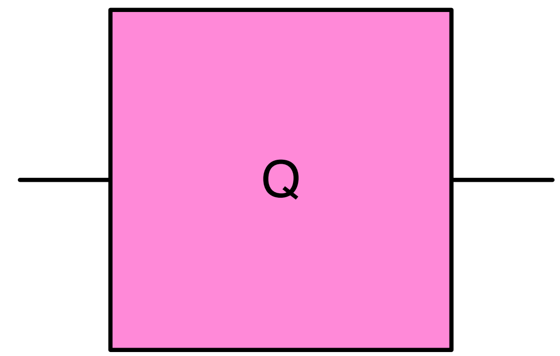
let's call it "composition"

  - unit $O \xrightarrow{\eta} A$ let's call it "identity"
  - satisfying associativity & unit laws

Distributive laws of PROPs

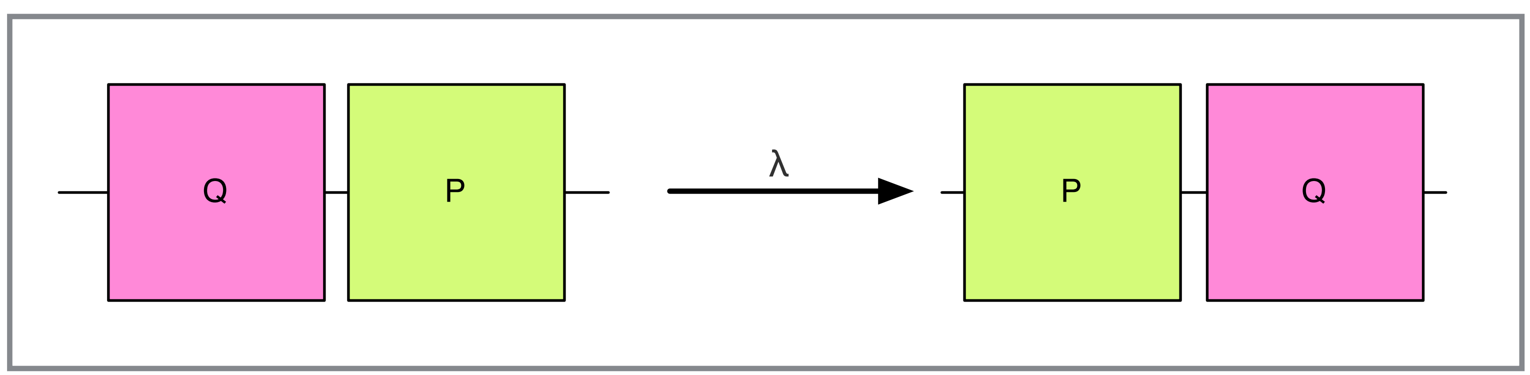


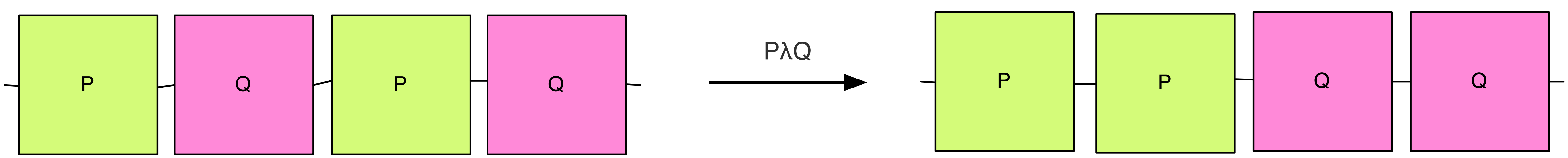
Green PROP P



Purple PROP Q

When can we understand P;Q as a PROP?





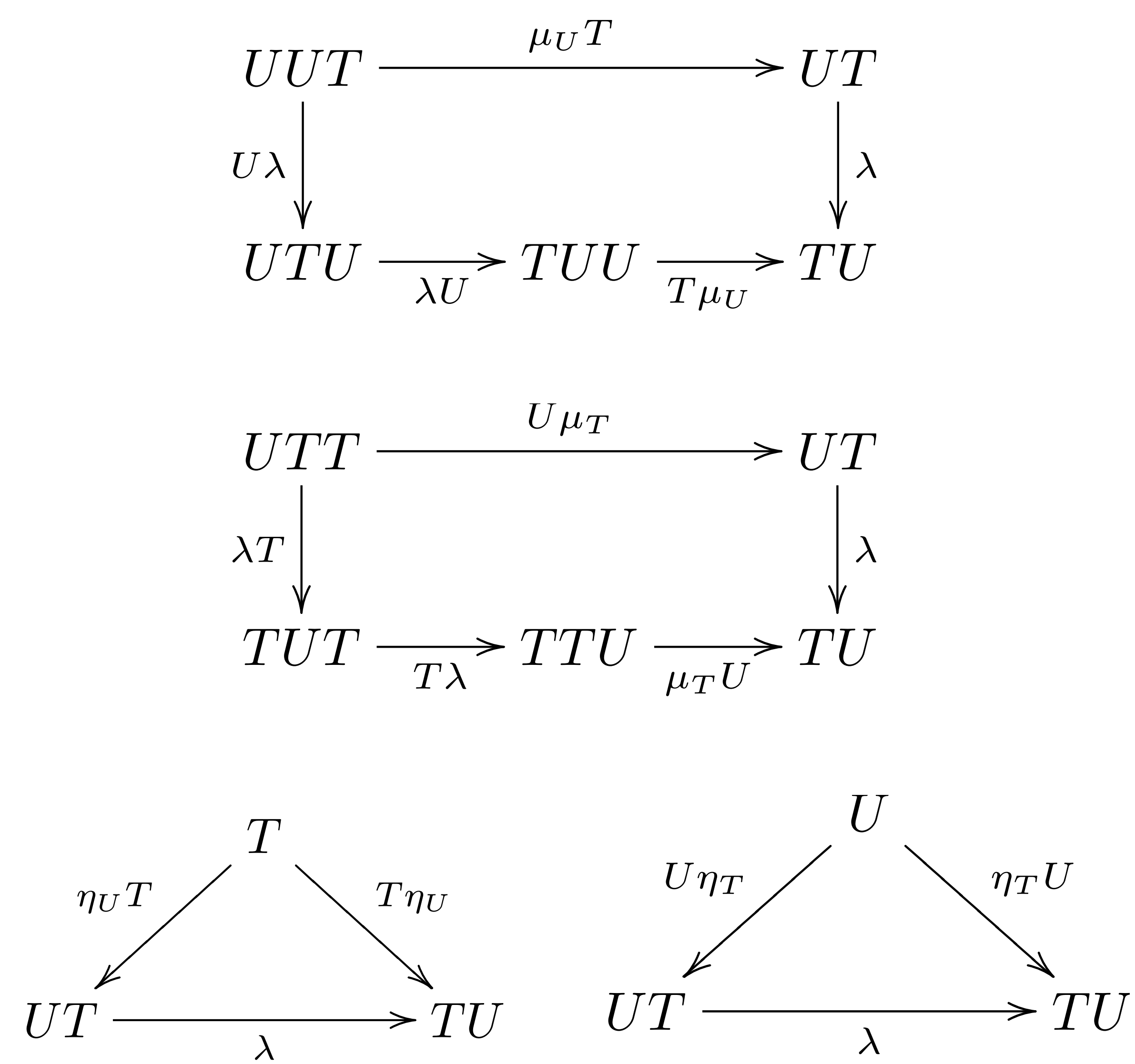
Distributive law of Monads

- Given monads T, U, a distributive law is a 2-cell

$$\lambda : UT \Rightarrow TU$$

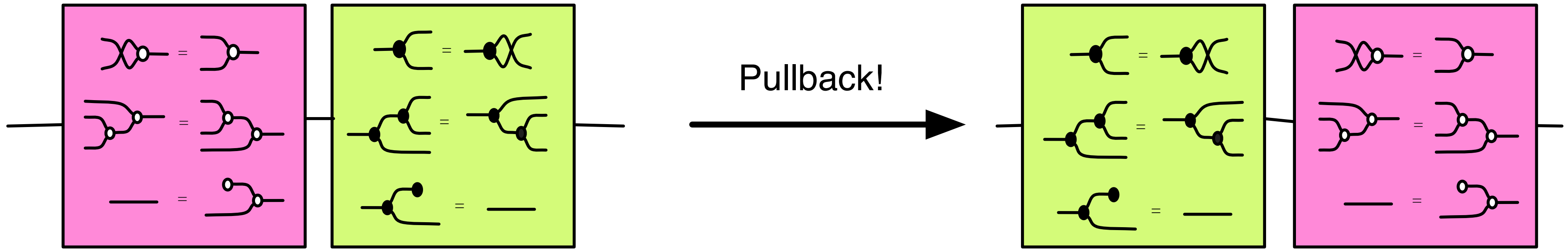
- that is compatible with multiplication and units in T and U in the obvious way (see diags)

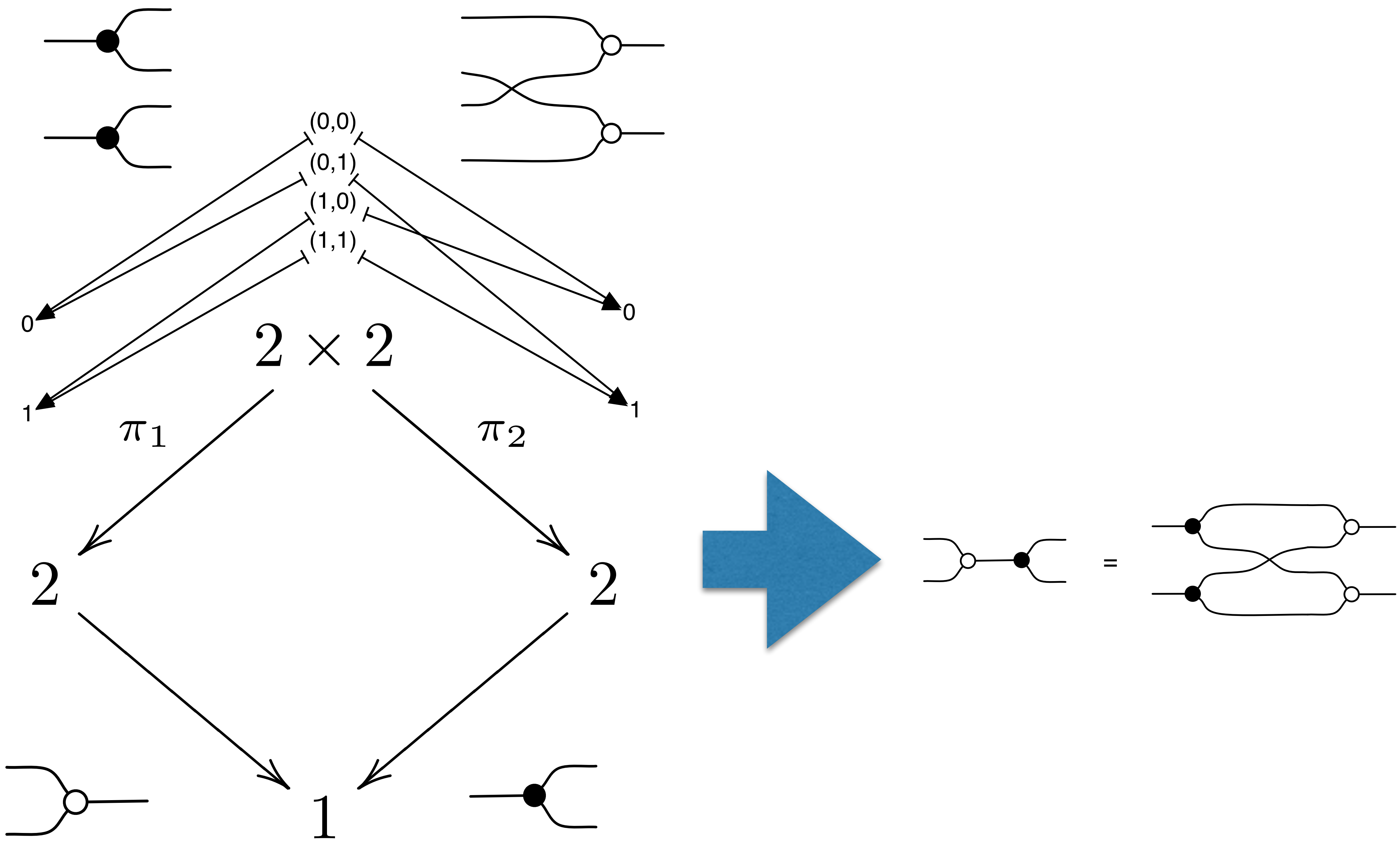
- **gives a monad structure on TU**



SMT of Spans

- The bicategory Span(**Set**) has spans of functions as 1-cells and span morphisms as 2-cells
    - composition is by **pullback**
    - we obtain the **category** of spans by identifying isomorphic spans
- We already have the SMT of functions (commutative monoids) and “backwards functions” (commutative comonoids)
- Pullback defines a distributive law of PROPs - implied by the universal property

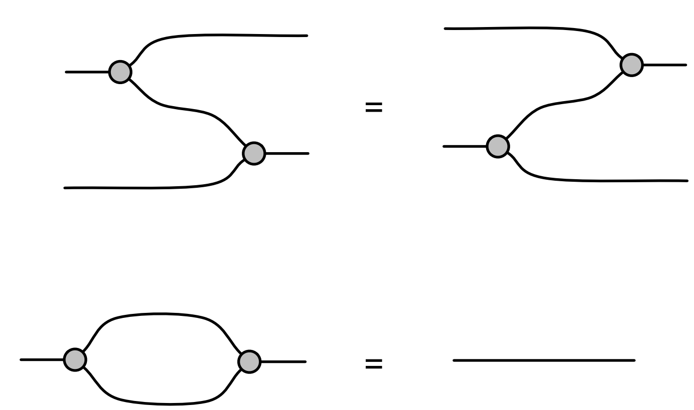




- the theory of bimonoids is a presentation of this distributive law
- so **B** ≅ **Mat** ≅ Span(**F**)
- for details see Steve Lack's paper

SMT of Cospans

- The bicategory Cospan(Set) has cospans of functions as 1-cells and cospan morphisms as 2-cells
    - composition is by **pushout**
    - pushout defines a distributive law
- obtain theory strongly separable Frobenius monoids — the theory of cospans!

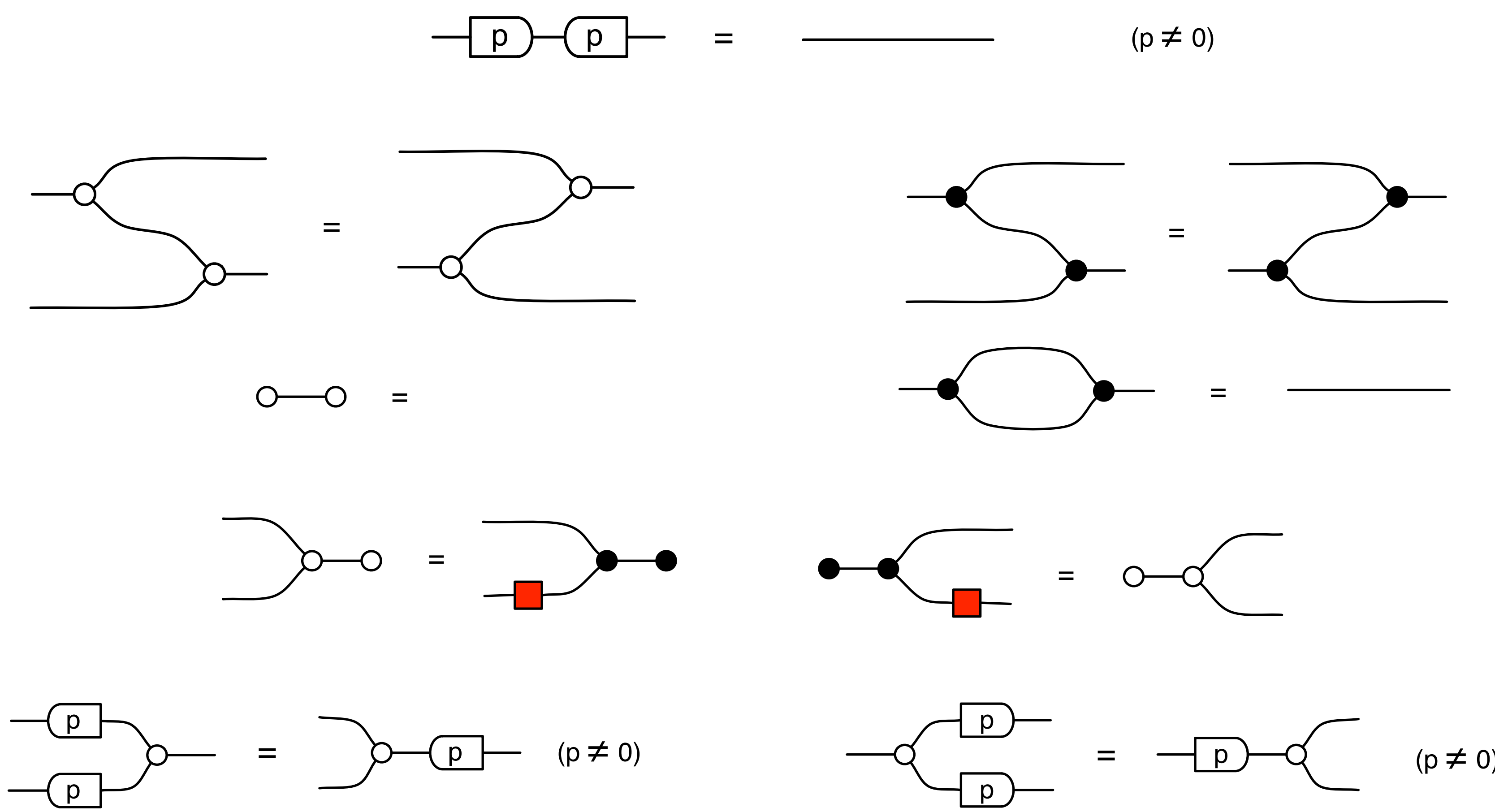


Proof of **IH** $\cong$ **LinRel** (outline)

- Two distributive laws
  - slight generalisation of Lack's notion
- **Mat$_\mathbf{Z}$** has both pullbacks and pushouts
  - it is equivalent to the category of free f.d. **Z**-modules
  - since **Z** is a PID, this category has pullbacks
  - because of transpose, **Mat$_\mathbf{Z}$** also has pushouts
- We thus obtain two distributive laws:
  - one from pullbacks, giving **spans of matrices**
  - one from pushouts, giving **cospans of matrices**

Spans of matrices

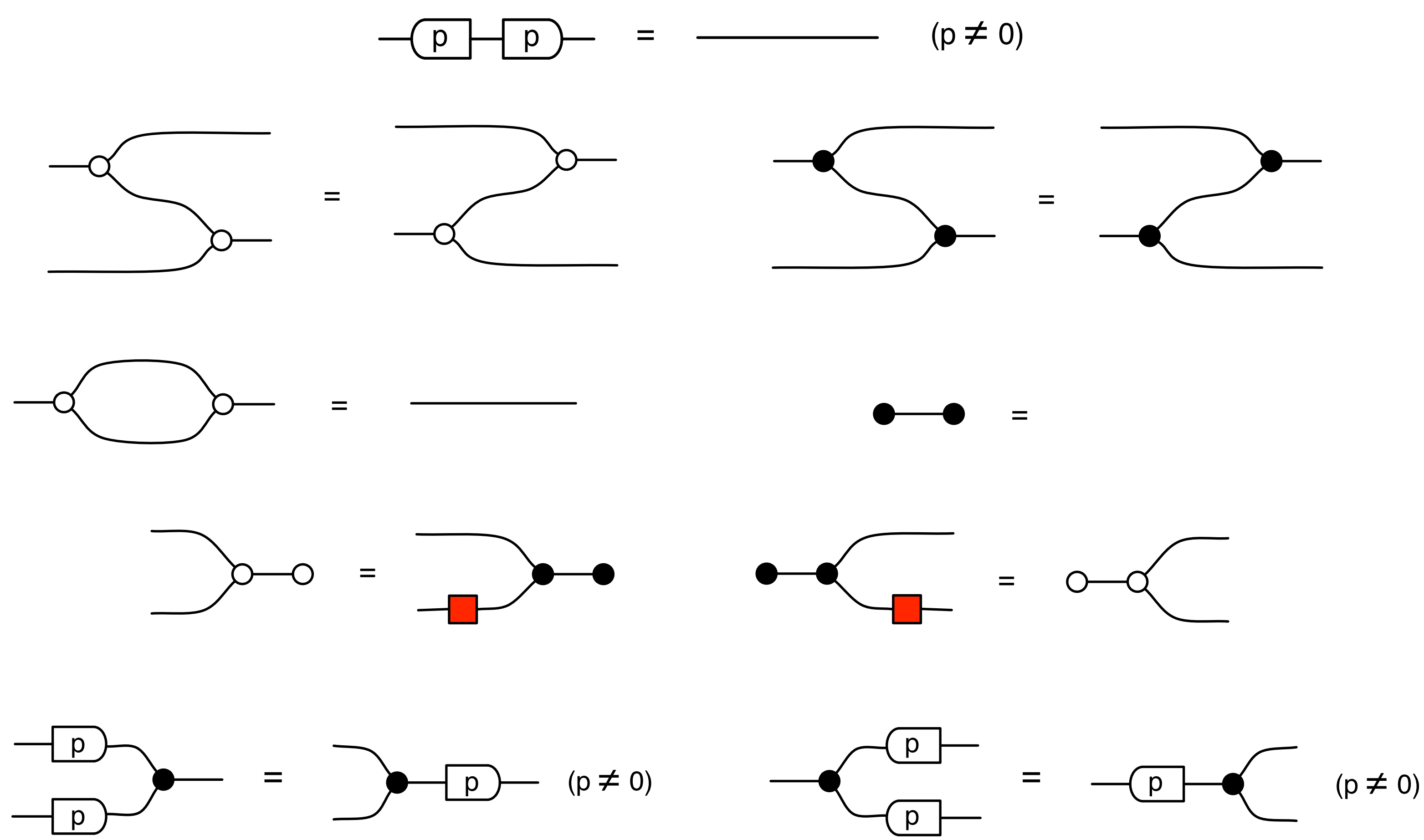
IR$^{\text{Span}}$



IR$^{\text{Span}}$ ≅ Span(**Mat**$_{\mathbf{Z}}$)

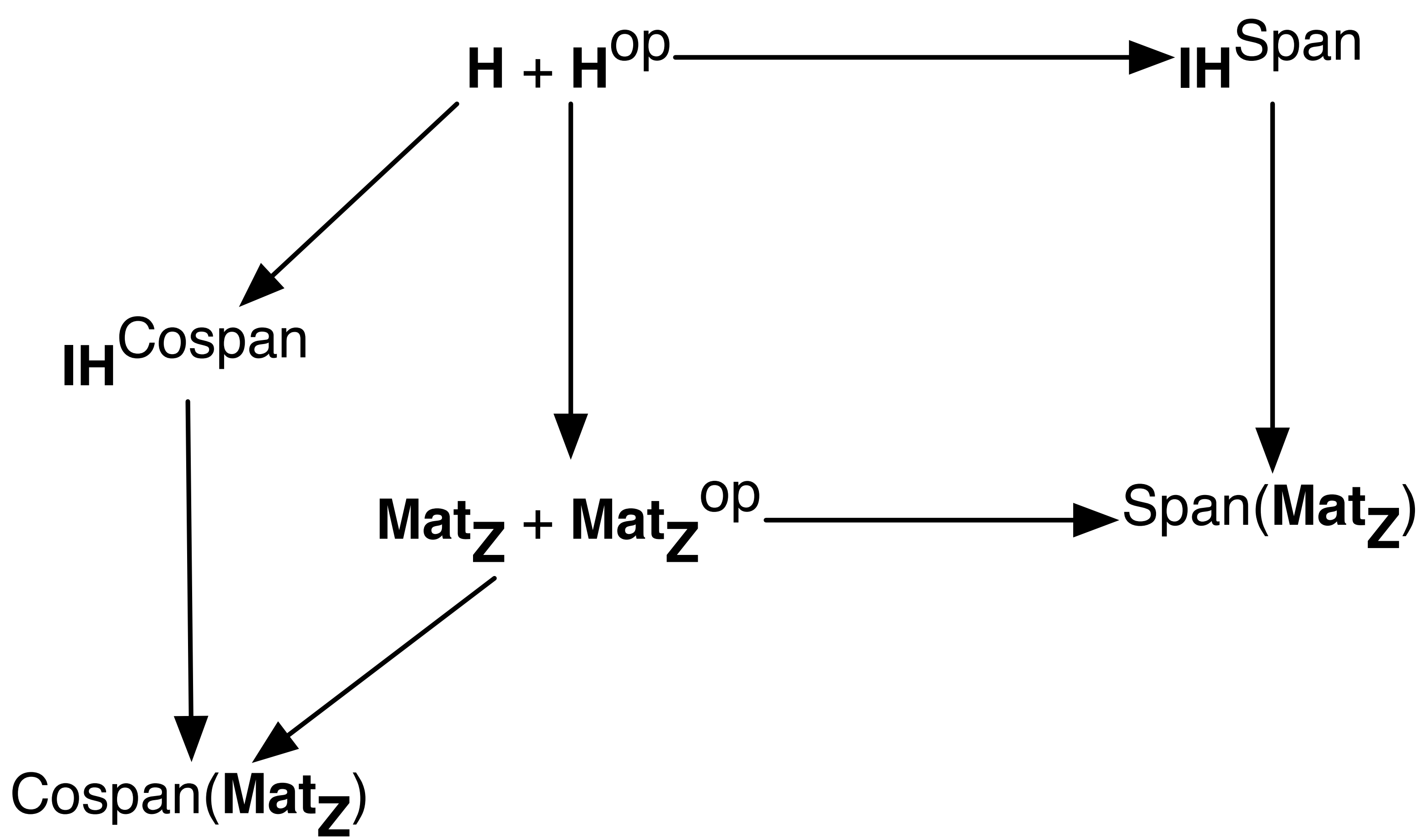
Cospans of matrices

IR$^{\text{Cospan}}$

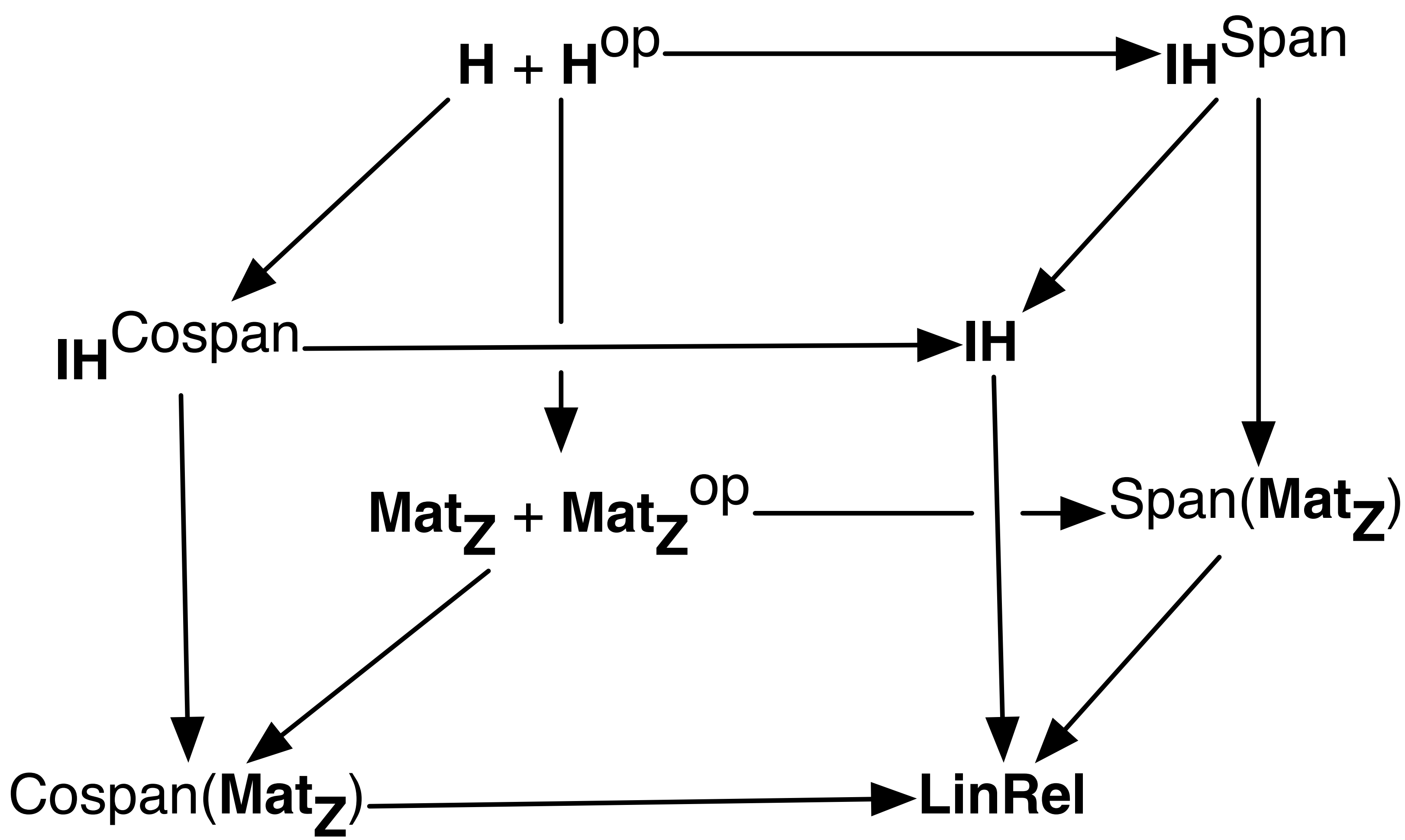


IR$^{\text{Cospan}}$ ≅ Cospan(**Mat**$_{\mathbf{Z}}$)

The cube - back faces

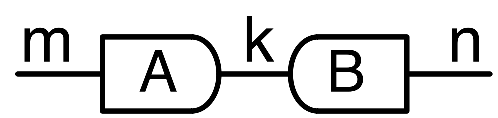


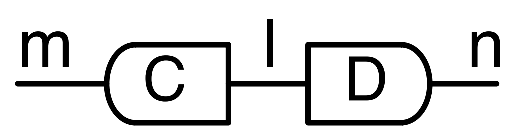
The cube



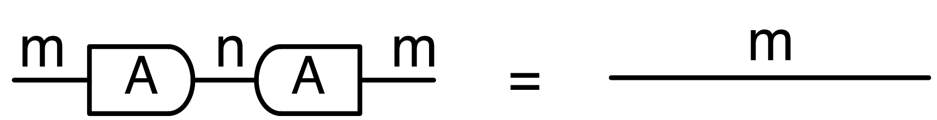
H + H^op
IH^Span
IH^Cospan
IH
Mat_Z + Mat_Z^op
Span(Mat_Z)
Cospan(Mat_Z)
LinRel

Corollary

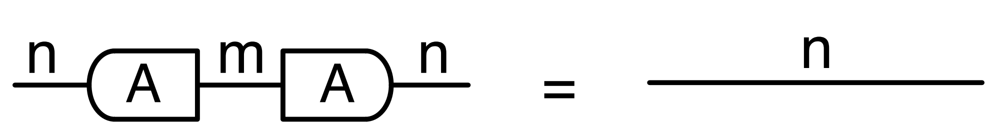
- The proof gives us some useful facts
  - every diagram in **IH** can be factorised in two ways
    - as a span 

    - as a cospan 

  - every mono in $\mathbf{Mat_Z}$ satisfies



  - every epi in $\mathbf{Mat_Z}$ satisfies



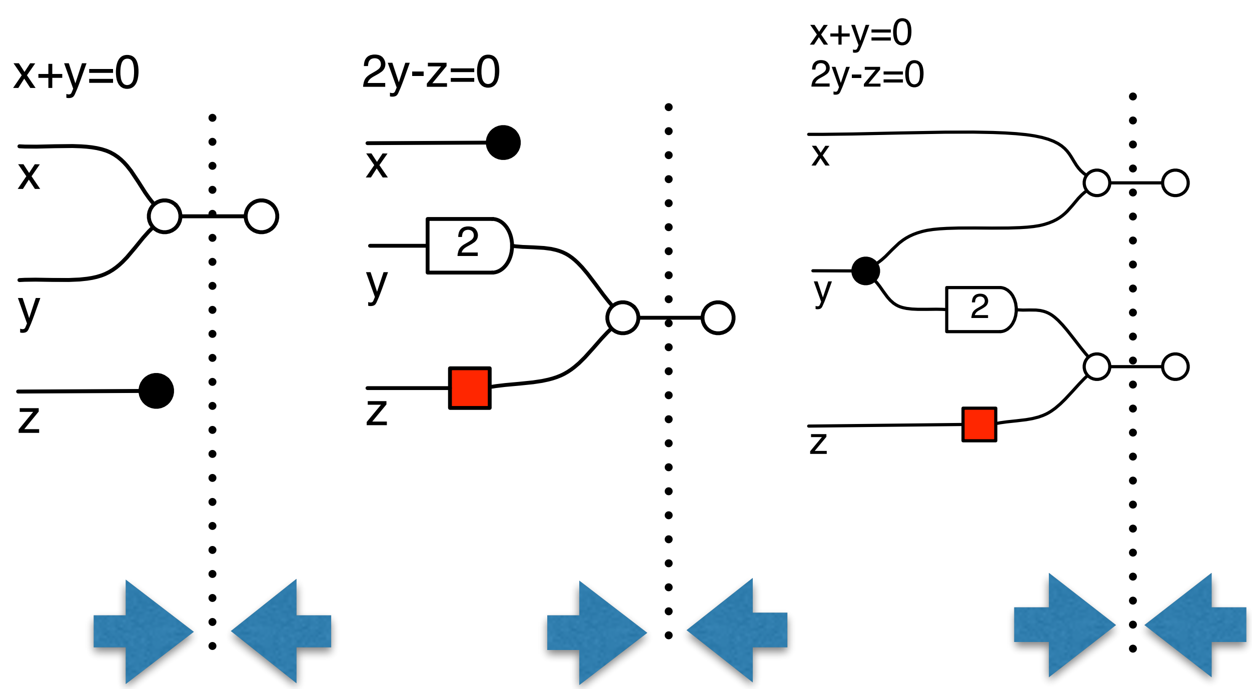
Plan

- basic theory of string diagrams
- theory of natural number matrices (bimonoids) and integer matrices (Hopf monoids)
- theory of linear relations (interacting Hopf monoids)
- distributive laws
- **linear algebra, diagrammatically**
- an application: signal flow graphs

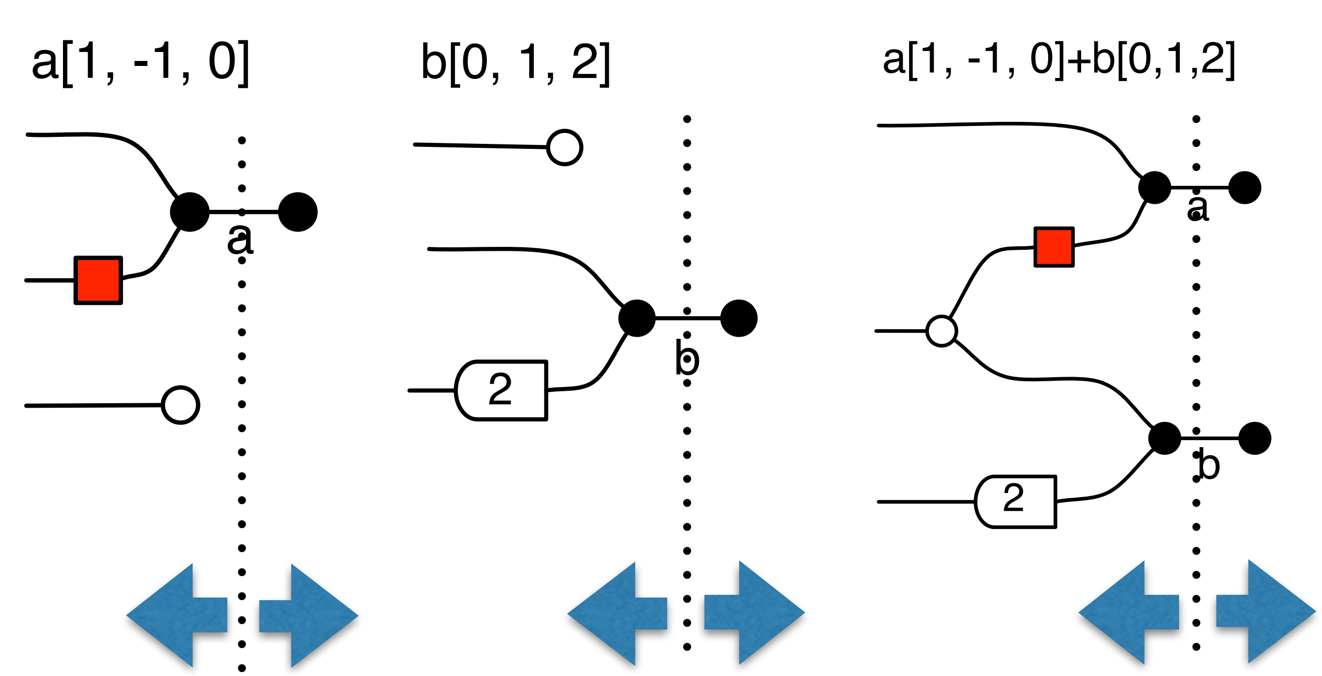
Factorisations

- Every diagram can be factorised as a span or a cospan of matrices
- This gives us the two different ways one can think of spaces

solutions of a list of homogeneous equations	linear combinations of basis vectors

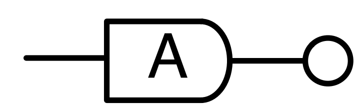


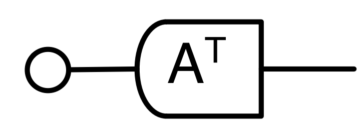
Cospans

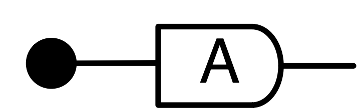


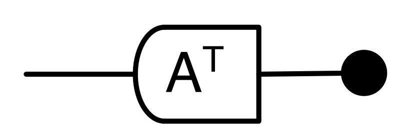
Spans

Image and kernel

- **Definition**
  - The **kernel** of A is 

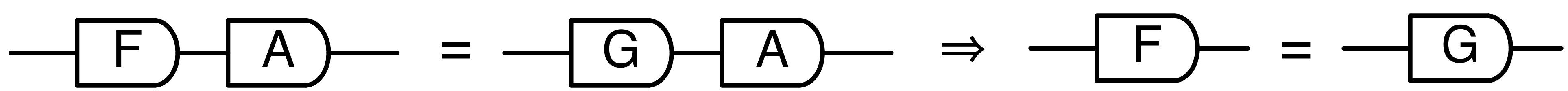
  - The **cokernel** of A is 

  - The **image** of A is 

  - The **coimage** of A is 

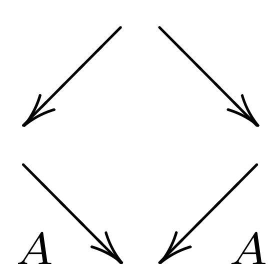
Injectivity

Injective matrices are the monos in **Mat$_\mathbb{Z}$**



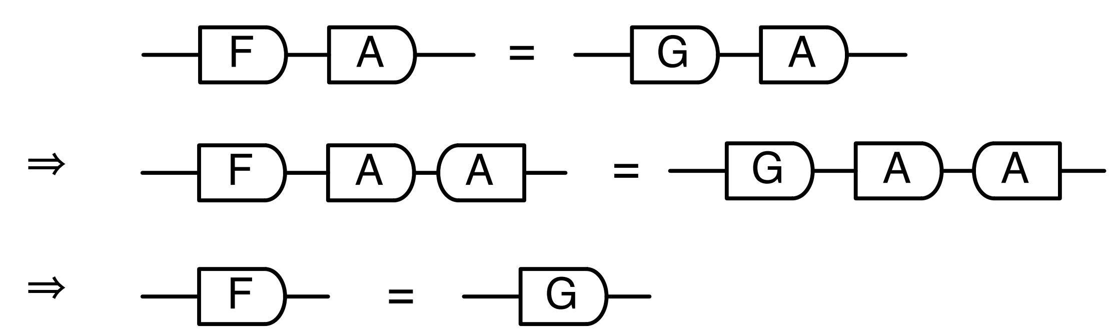
Theorem. A is injective iff

⇒



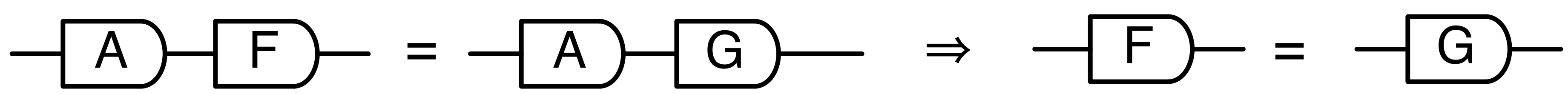
is pullback in **Mat$_\mathbb{Z}$**

⇐



Surjectivity

- Surjective matrices are the epis in $\mathbf{Mat}_{\mathbf{Z}}$, i.e.



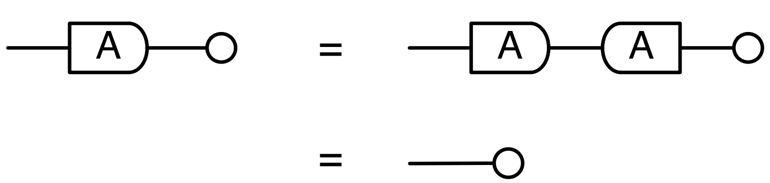
- **Theorem**. A is surjective iff

Proof: Bizarro of last slide

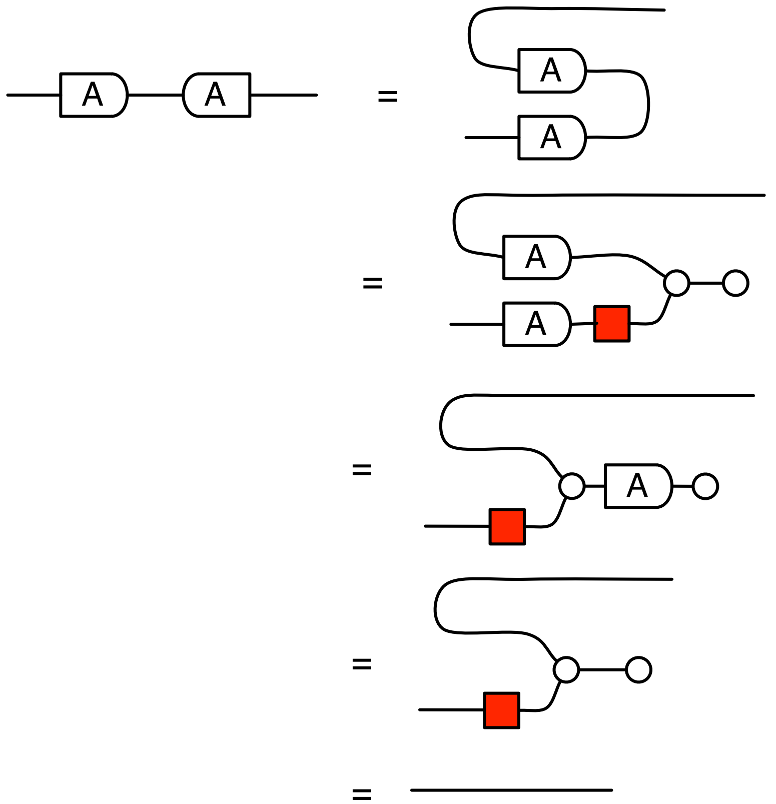
Injectivity and kernel

- **Theorem**. A is injective iff ker A = 0

$\Rightarrow$

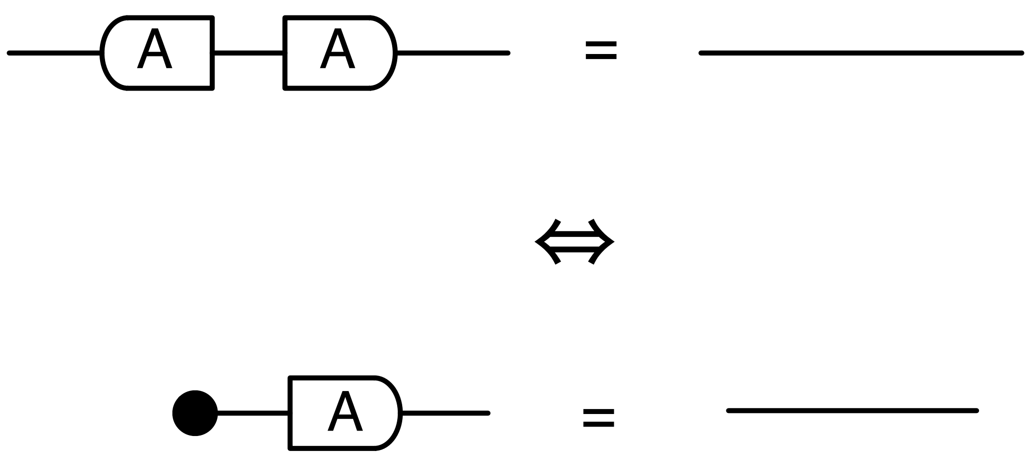


$\Leftarrow$



Surjectivity and image

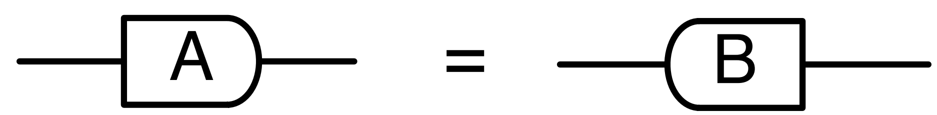
- **Theorem**. A is surjective iff im(A)=codomain



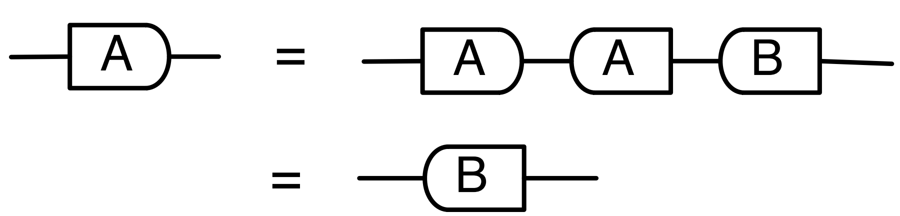
Proof: bizarro of last slide

Invertible matrices

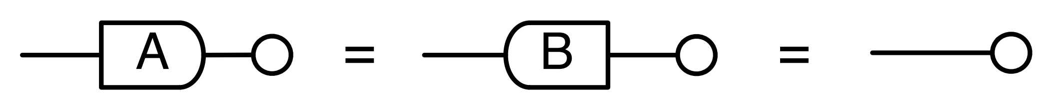
- **Theorem**: A is invertible with inverse B iff



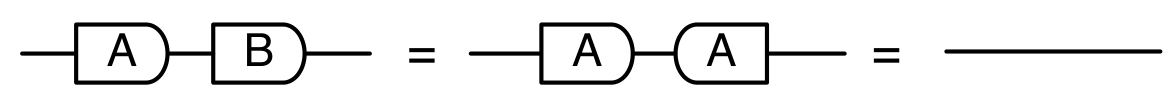
$\Rightarrow$



$\Leftarrow$



so A is injective



bizarro argument yields other half

Summary

- We have done a bit of linear algebra without mentioning
  - vectors, vector spaces and bases
  - linear dependence/independence, spans of a vector list
  - dimensions
- Similar stories can be told for other parts of linear algebra: decompositions, eigenvalues/eigenspaces, determinants
  - much of this is work in progress: check out the blog! :)

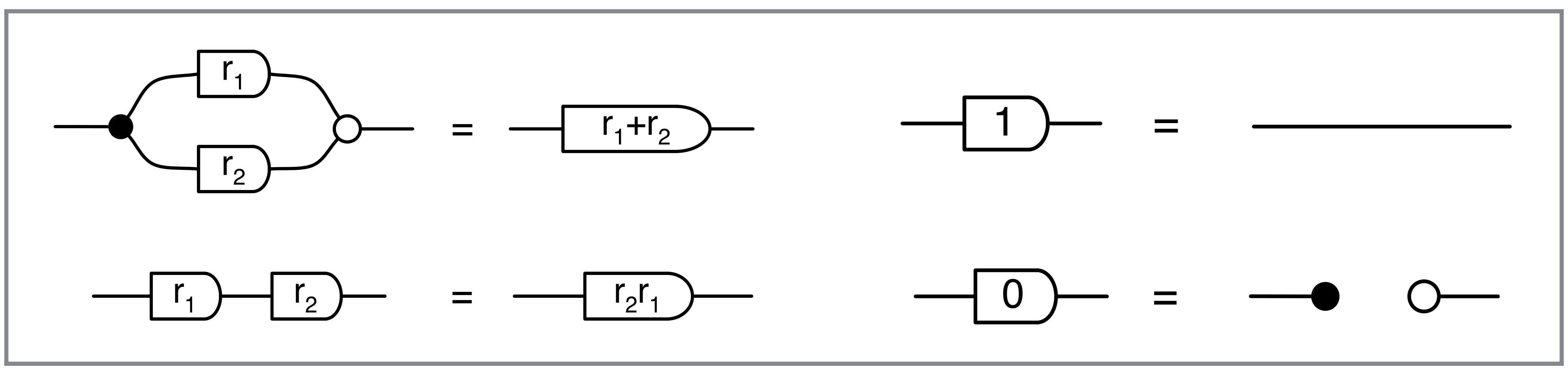
Plan

- basic theory of string diagrams
- theory of natural number matrices (bimonoids) and integer matrices (Hopf monoids)
- theory of linear relations (interacting Hopf monoids)
- distributive laws
- linear algebra, diagrammatically
- **an application: signal flow graphs**

Generalising (slightly)

- It is straightforward to generalise from **Z** to arbitrary PID **R**

- We can build the theory $\mathbf{H_R}$ by adding enough scalars to the graphical syntax together with equations

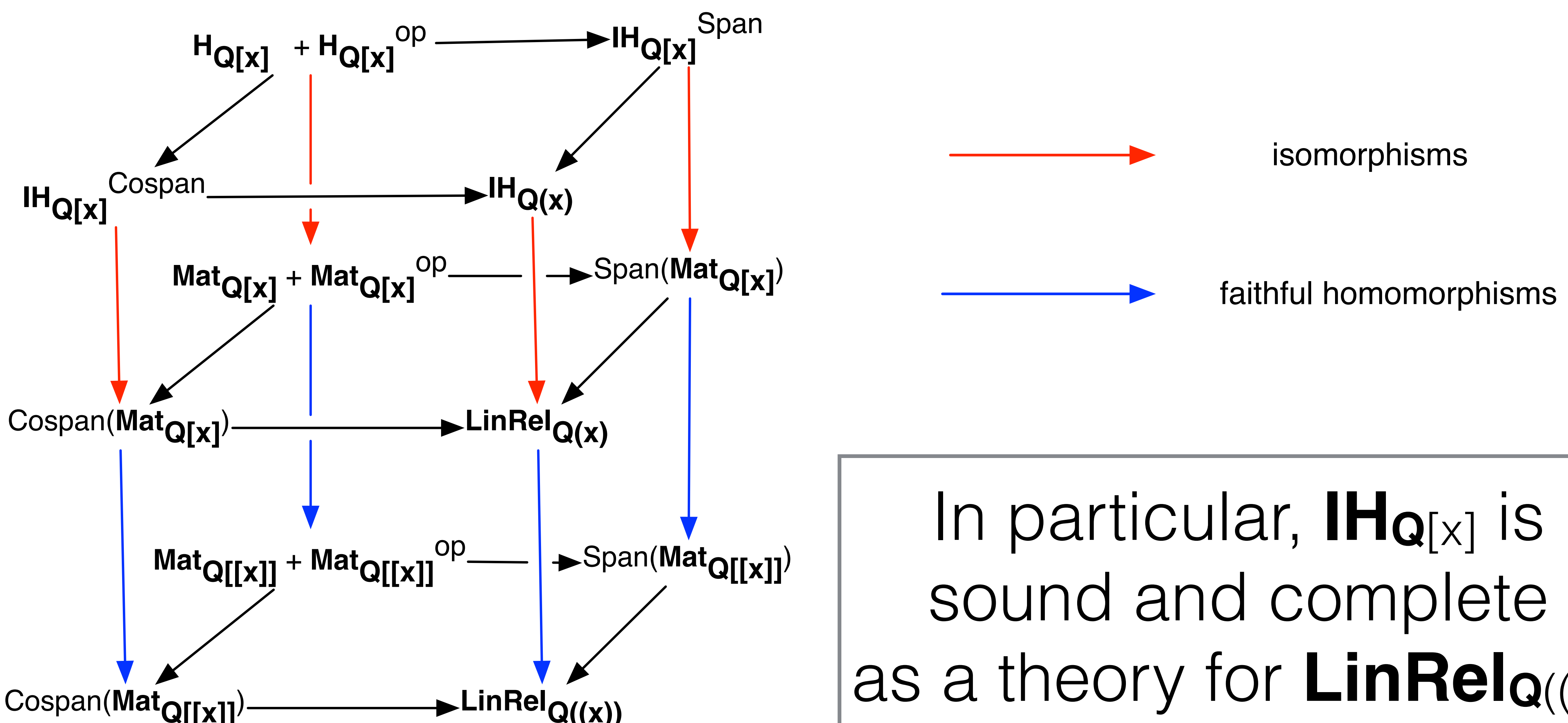


- The additional equations of $\mathbf{IH_R}$ are the same as before

Application: infinite series

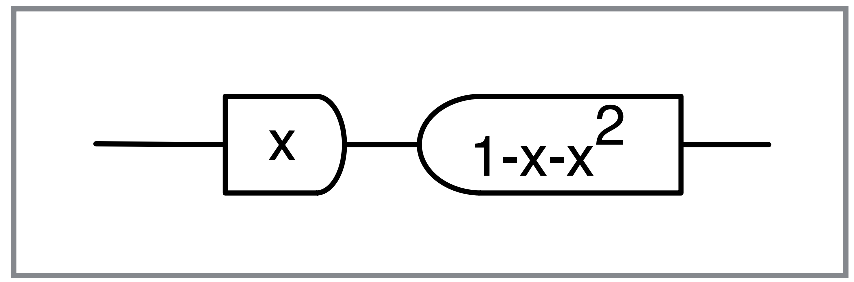
- Diagrammatic calculus for spaces over the field of fractions of **Q**[x] (polynomials with one variable, a PID) is especially interesting
  - polynomial fractions = nice syntax for many infinite series (generatingfunctionology!)
  - formally: there is an embedding of fields from poly fractions (syntax) to Laurent series (semantics)
- Moreover: diagrams are very closely related to **signal flow graphs**
  - invented by Shannon in the 40s, reinvented by Mason in the 50s, foundational structure in control and signal processing
  - useful circuit-like syntax for **linear time-invariant dynamical systems**

The cube (with extra level!)



In particular, $\mathbf{IH}_{\mathbf{Q}[x]}$ is sound and complete as a theory for $\mathbf{LinRel}_{\mathbf{Q}((x))}$

Example



As linear relation over **Q**(x) is the space generated by

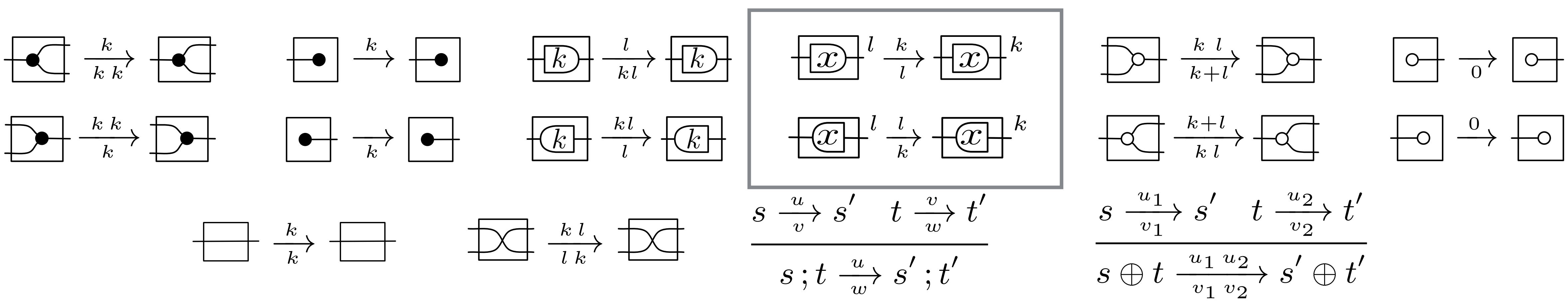
$$(1 \quad , \quad x/(1-x-x^2))$$

As linear relation over **Q**((x)) is the space generated by

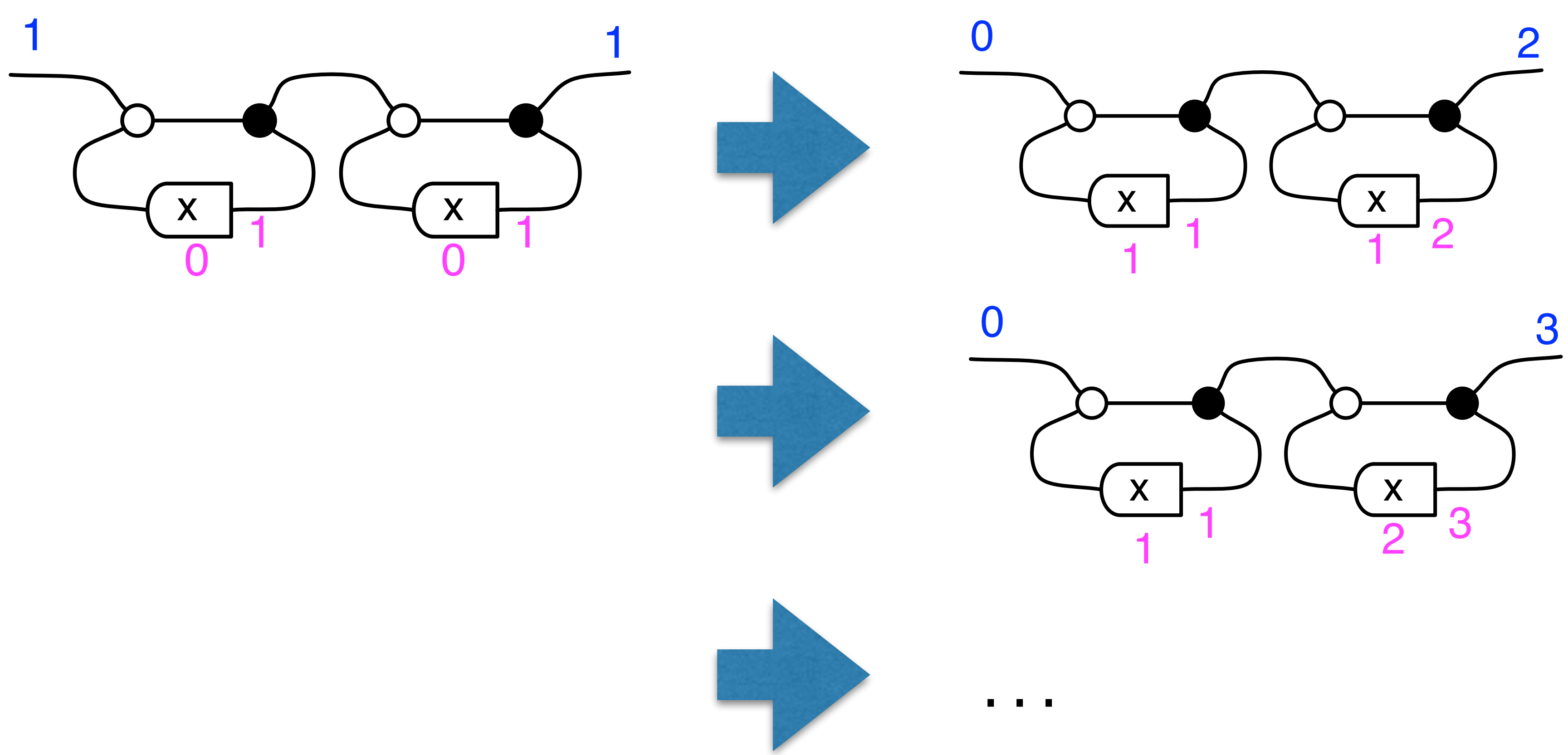
$$(\underline{1},0,0,\dots \quad , \quad \underline{0},1,1,2,3,5,8,\dots)$$

Operational semantics

Bonchi, S., Zanasi, *Full abstraction for signal flow graphs*, POPL ‘15

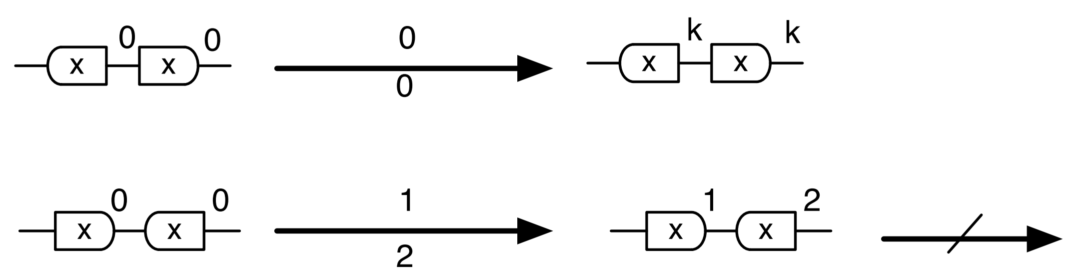


Example



Operational Semantics vs Denotational Semantics

Operational semantics closely related to denotational semantics [linear relations over **Q**((x))]
with some "implementation issues" in diagrams where signal flow is inconsistent e.g.

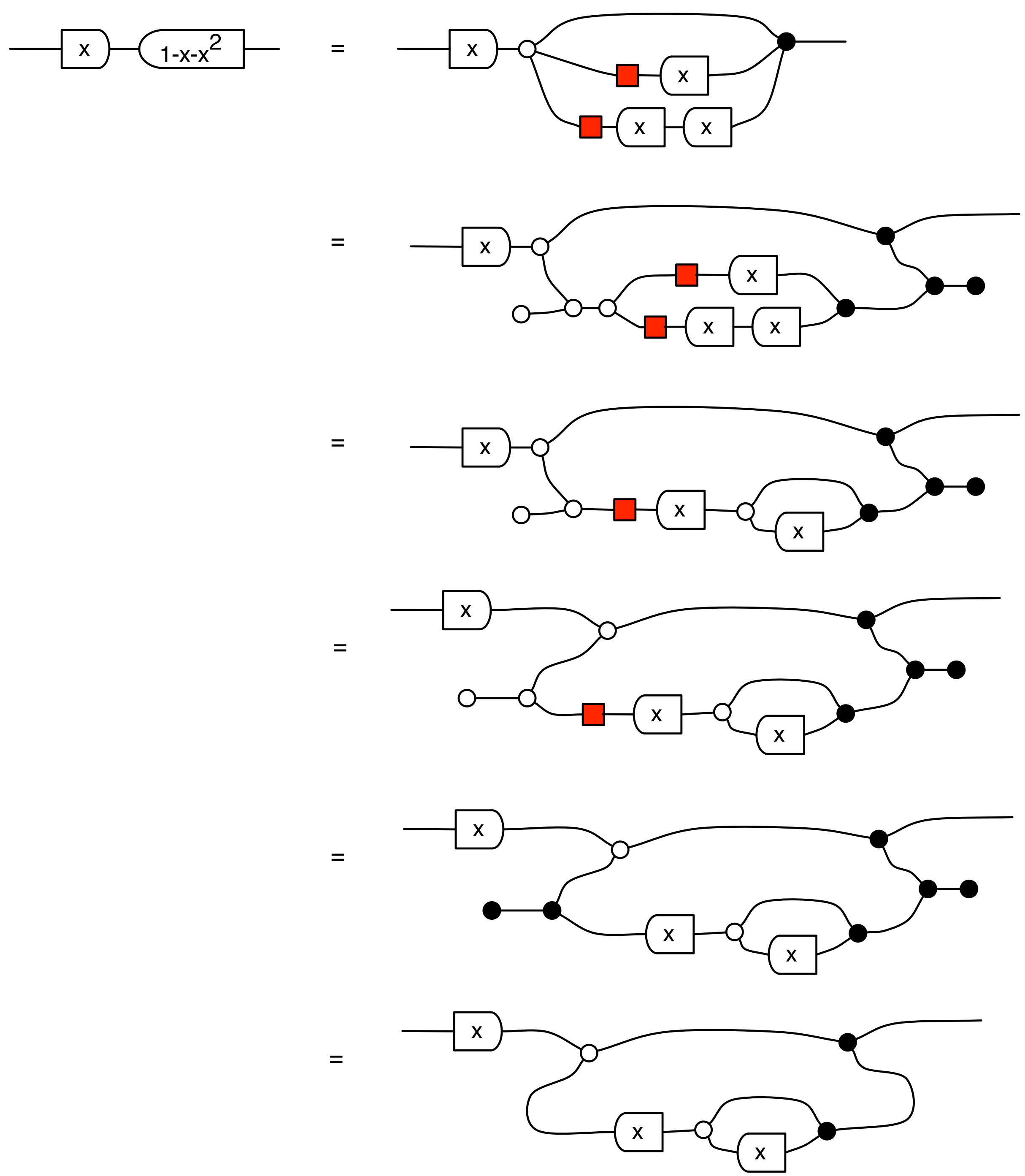


Realisability and Full Abstraction

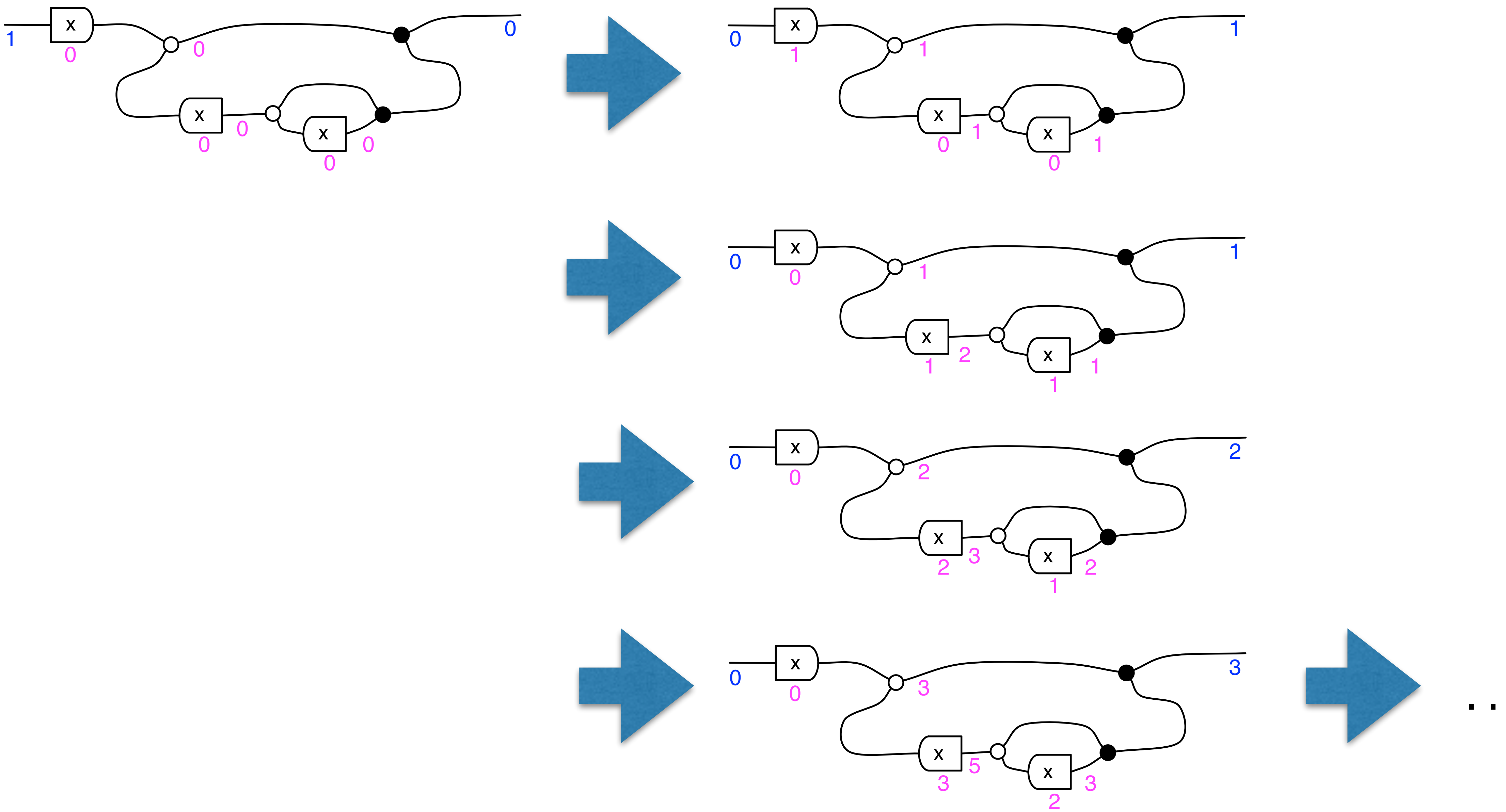
- **Realisability** Every diagram can be put in a form where the direction of signal flow is consistent

- **Full abstraction** *Operational equality* (in terms of behaviour, given by operational semantics) coincides with *denotational equality* (the denoted linear relation) on diagrams with consistent signal flow

Implementing Fibonacci



Running Fibonacci



Signal flow graphs

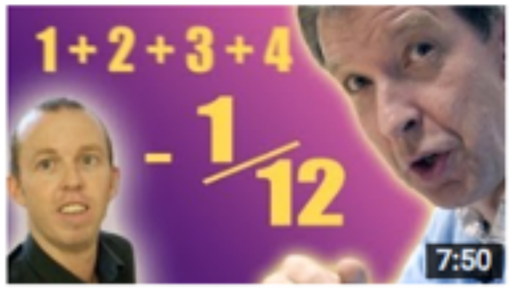
Signal flow graphs differ from electrical network graphs in that their branches are directed. In accounting for branch directions it is necessary to take an entirely different line of approach from that adopted in electrical network topology."

S.J. Mason, *Feedback Theory: I. Some Properties of Signal Flow Graphs,* 1953

Adding a signal flow direction is often a figment of one's imagination, and when something is not real, it will turn out to be cumbersome sooner or later.

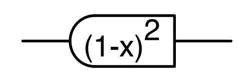
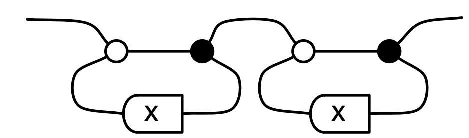
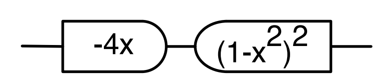
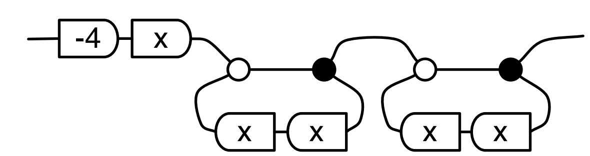
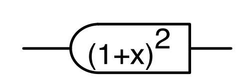
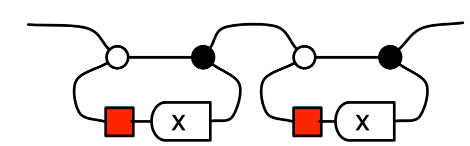
J.C. Willems, *Linear systems in discrete time,* 2009

"Summing up 1,2,3,4,…"



ASTOUNDING: 1 + 2 + 3 + 4 + 5 + ... = -1/12
3,275,126 views • 1 year ago
CC

https://www.youtube.com/watch?v=w-I6XTVZXww

	Generating function	Diagram	Signal flow graph
1,2,3,4,…	$\frac{1}{(1-x)^2}$		
0,-4,0,-8,..	$\frac{-4x}{(1-x^2)^2}$		
1,-2,3,-4,…	$\frac{1}{(1+x)^2}$		

$$s - 4s = \frac{1}{4} \quad \Rightarrow \quad s = -\frac{1}{12}$$

Bibliography

- Bonchi, S., Zanasi - Interacting Bialgebras are Frobenius, FoSSaCS '14
- Bonchi, S., Zanasi - Interacting Hopf Algebras, arXiv, '14
- Bonchi, S., Zanasi - A categorical semantics of signal flow graphs, CONCUR '14
- Bonchi, S., Zanasi - Full abstraction for signal flow graphs, PoPL '15

graphicallinearalgebra.net

Future work

- **Control** - with Paolo Rapisarda, Brendan Fong, …
- **Continuous semantics of flow** - inspiration from “Calculus in Coinductive Form” by Dusko Pavlovic & Martín Escardo (LiCS `99)
- **Graph theory** - string diagrams as compositional language of graphs (Apiwat Chantawibul and S., MFPS `15)
- **Operational semantics, distributive laws** - Fabio Zanasi and Filippo Bonchi
- **Petri nets**, **model checking** - Julian Rathke and Owen Stephens
- **Concurrent programming** - in the works, with Kostadin Stoilov