Diagrammatic relational algebra and applications

CATMI, Bergen, June 26-30 2023

Pawel Sobocinski, Tallinn University of Technology

Roadmap

- **Lecture 1 - Functorial semantics 1 - algebraic theories**
- Lecture 2 - Functorial semantics 2 - partial, relational and first-order theories
- Lecture 3 - Graphical linear algebra and applications

Compositional modelling

What is compositionality?

- **Modularity** - a system described as a composition of its parts
- Compositionality - a combination of:
  - a language (**syntax**) for composing systems
  - with operations that are compatible with the intended meaning (**semantics**)
  - such that the translation syntax ⇒ semantics is homomorphic

Goal: no "emergent" behaviour

Modelling status quo

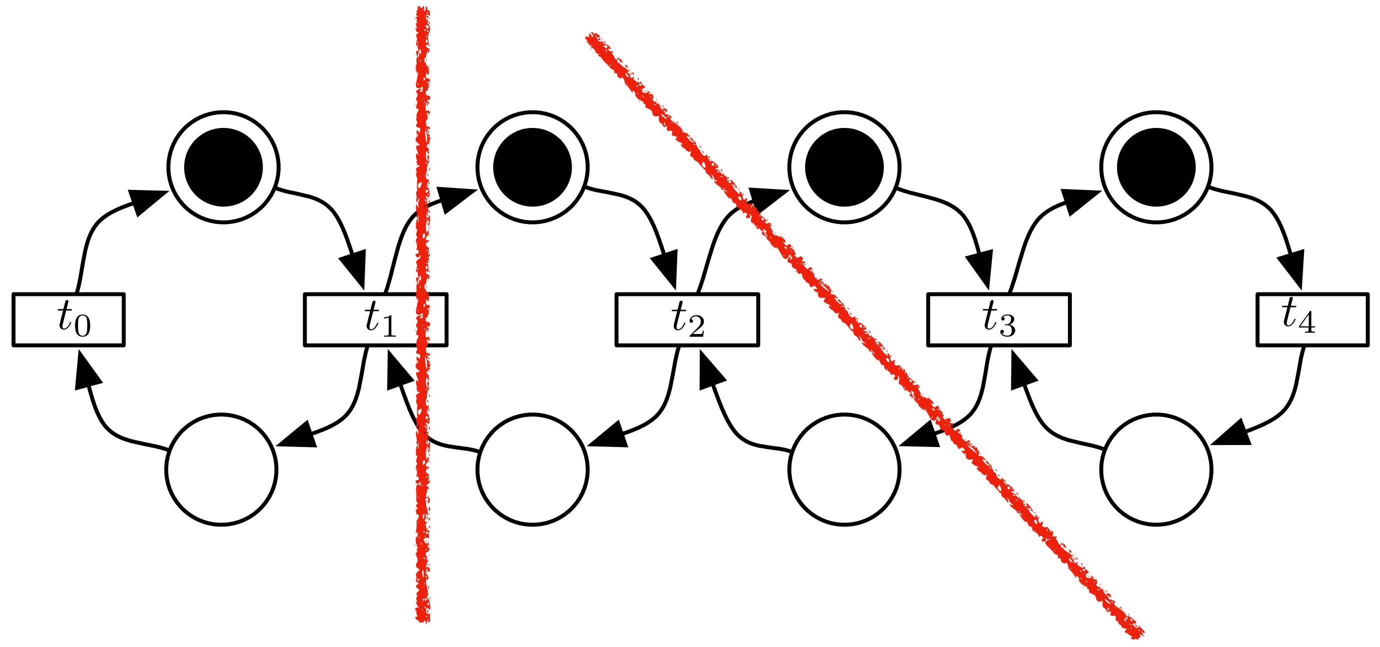
- models are global, monolithic and closed systems
- dynamics is obtained “a la physics” - analysing combinatorics of local interactions to obtain global behaviour via a set of differential equaitons
- not modular: often constructed fresh for each application
- interaction with environment is usually oversimplified or abstracted away
- analysis in functional terms, inputs driving outputs
- but we have more data than ever before – we need good models

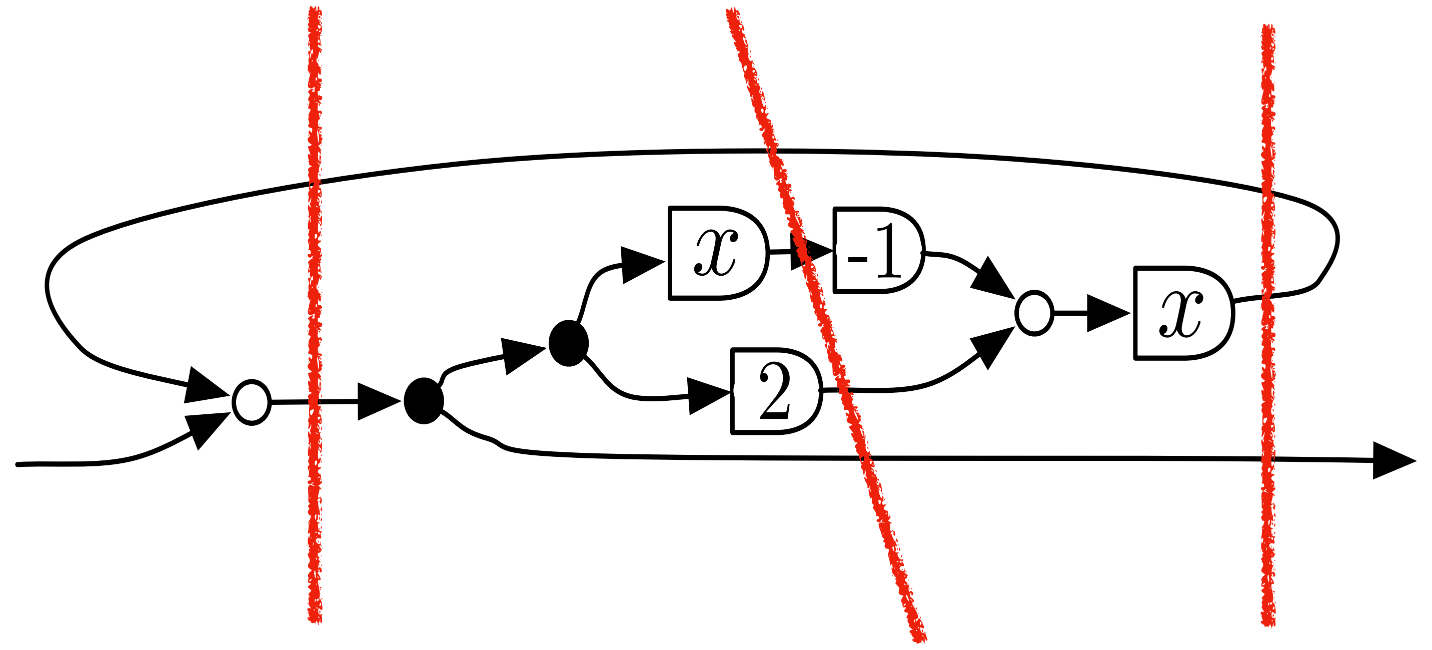
A problem with traditional modelling

The real world is not functional!

Although input/output thinking is useful in certain situations, … as a general methodology, input/output descriptions are ill-founded and clash with system interconnection. Interconnection, as we shall see, results in *variable sharing*, not in output- to-input assignment.

Jan C Willems, The Behavioral Approach to Open and Interconnected Systems, IEEE Control Systems Magazine, 2007





In such systems composition is often **relational**. There are many examples.

Towards a solution

New relational algebras?

- traditional syntax has functionality built in
  - all operations are functional
  - the main operation of term-building (substitution) is just fancy function composition
- 20th century extensions (essentially algebraic theories, first order theories) suffer from some of the same defects of term-building fundamentals
- some important insights have been obtained from the study of relational algebras: Peirce, Kleene, Tarski, Freyd and Scedrov, …
- Lawvere's insight: "functionality" is deeply associated with cartesian structure (i.e. categorical products)
  - traditional syntax is thus built to operate on "classical data": one that can be copied and discarded
- This, and other algebraic structure, can often be studied as additional structure on a **symmetric monoidal category**
- The plan for today and tomorrow: **Set**, **Par**, **Rel** as symmetric monoidal categories with structure

Traditional syntax

Theory of commutative monoids

$$(\underbrace{\{m,\ e\}}_{}, \ \{ \underbrace{m(m(x,y),z) = m(x,m(y,z)),\ m(x,y) = m(y,x),\ m(e,x) = x}_{} \})$$

signature consisting of operation symbols

equations

$$\{m,\ e\}$$

arity 2

arity 0

pairs of **terms** over some set of variables
implicit universal quantification

Traditional syntax

Universal algebra basics I

- A signature is a pair $\Sigma = (S,\alpha)$ where S is a set of **operation symbols** together with an **arity function** $\alpha : S \to \mathbf{N}$
- A **Σ-algebra** is a pair $(A,[-])$ where A is a set (semantic domain) and $[-]$ is a function that sends operation symbols to functions $[\sigma] : A^{\alpha(\sigma)} \to A$
- A **Σ-algebra homomorphism** is the obvious thing: a map between semantic domains that's homomorphic wrt operations:

$$\begin{tikzcd} A^n \arrow[r, "f^n"] \arrow[d, "\llbracket tn \rrbracket_A"'] & B^n \arrow[d, "\llbracket tn \rrbracket_B"'] \\ A \arrow[r, "f"'] & B. \end{tikzcd}$$

- Given a set of variables V, the **term** Σ-algebra T_V is
  - $T_V ::= V \mid t_0 \mid t_1(T_V) \mid t_2(T_V,T_V) \mid \ldots \mid t_n(T_V,\ldots,T_V) \mid \ldots$
- The term Σ-algebra satisfies a universal property, any $v : V \to A$ extends to a unique Σ-algebra homomorphism $v^* : T_V \to A$
  - compositionality!

Traditional syntax

Universal algebra basics II

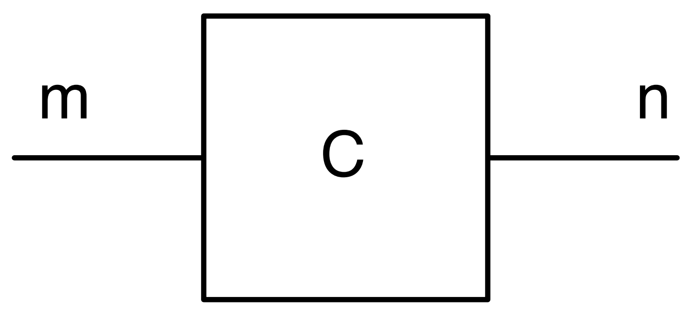
- An **equation** is a pair $(s,t) \in T_V \times T_V$
- An **algebraic theory** is a pair (Σ, E) where Σ is a signature and E is a set of equations.
  - Example: the theory of commutative monoids
- A **model** is a Σ-algebra where every equation $e \in E$ holds (for any valuation $v : V \to A$ *)
- A model homomorphism is a Σ-algebra homomorphism
- The class of models of a theory is called a **variety**
- **Theorem** (Birkhoff 1935) A class of Σ-algebras is a variety iff it is closed under homomorphic images, subalgebras and products.
- * Note: given that equations are required to hold under any evaluations, they are implicitly **universally quantified**
- For more expressivity,
  - **essentially algebraic theories**, **quasi-varieties**: operations are allowed to be partial, equations involve domains of definition
  - **first order theories**: syntax contains relation symbols and formulas are more involved
    - logical operations including negation, quantifiers

Symmetric monoidal categories

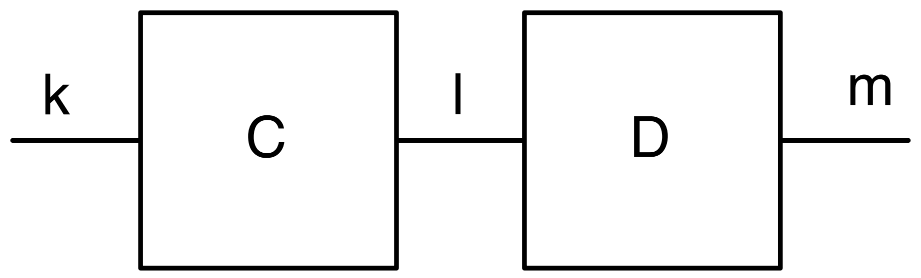
- A **monoidal** category **C** is a category equipped with monoidal product $\otimes$
  - $\otimes : \mathbf{C} \times \mathbf{C} \to \mathbf{C}$
  - an object $I \in C$ called the monoidal unit
  - together with coherent **natural** isomorphisms
    - $\alpha_{a,b,c} : (a \otimes b) \otimes c \to a \otimes (b \otimes c)$
    - $\rho_a : a \otimes I \to a$
    - $\lambda_a : I \otimes a \to a$
- A **symmetric monoidal** category additionally has a natural isomorphism $\sigma_{X,Y} : X \otimes Y \to Y \otimes X$ that satisfies $\sigma_{X,Y} ; \sigma_{Y,X} = id_{X,Y}$
- Relevant examples, in all cases the **cartesian product** of sets gives a symmetric monoidal structure
    - **Set**, **Par**, **Rel**
- For any set X, there are **strict** versions, $\mathbf{Set}_X$, $\mathbf{Par}_X$, $\mathbf{Rel}_X$.
  - In each case the objects are natural numbers, and arrows from m to n are arrows $X^m \to X^n$ in the relevant category
  - strict symmetric monoidal categories with objects natural numbers and $\otimes$ on objects acting as $+$ are called **props**

String diagrams - a quick tutorial

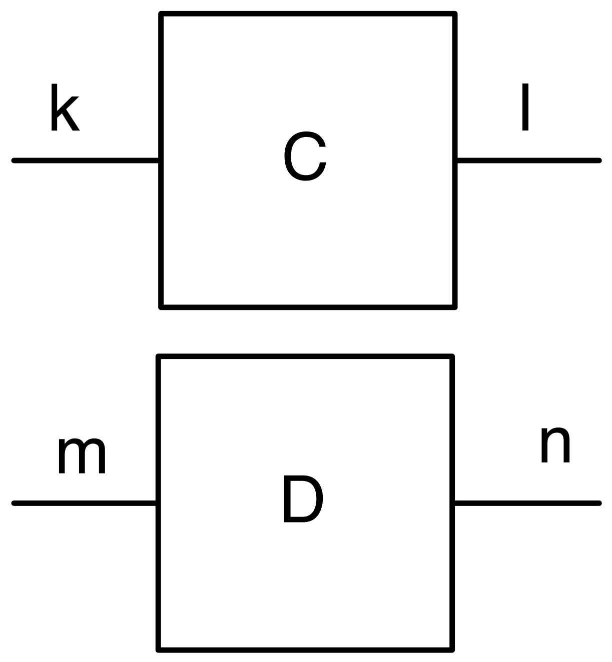
- Instead of writing C : m → n, we draw



- composition is plugging wires

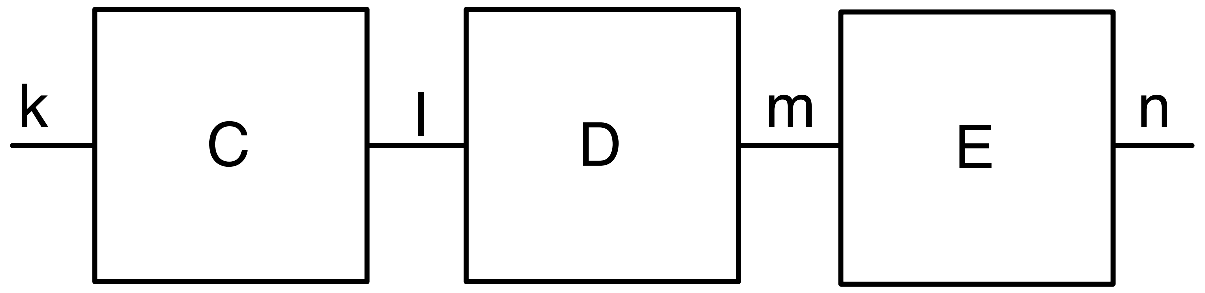


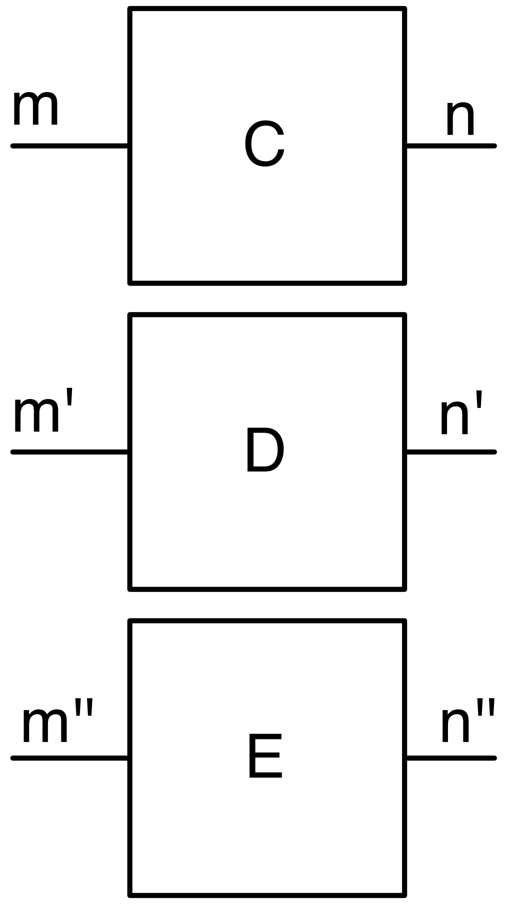
- monoidal product is “stacking” boxes



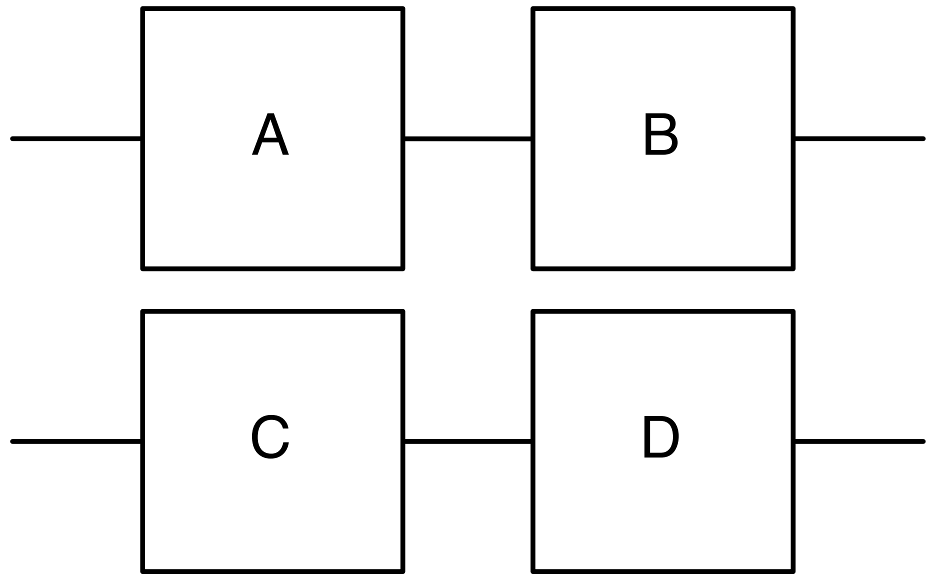
Perks of the notation

- associativity is built in:



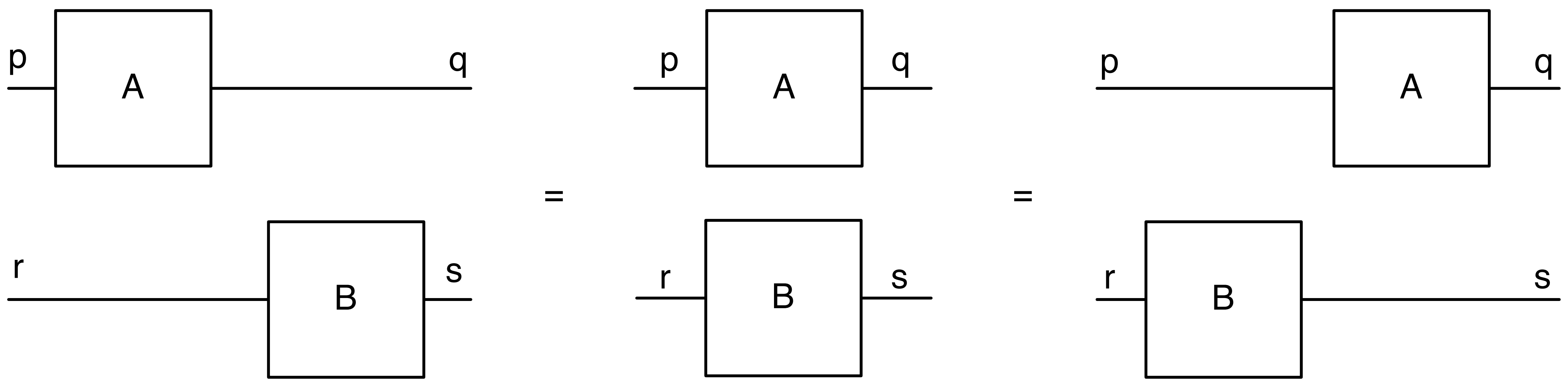


- functoriality of ⊗ is built in:

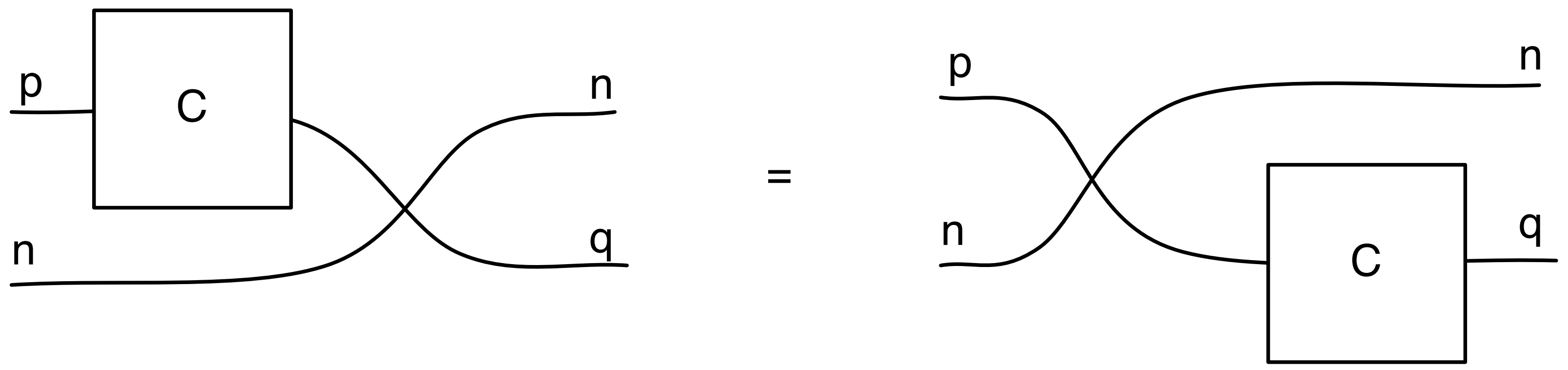


Identities and symmetries

- Identity arrows are drawn as wires. The monoidal identity is not drawn.



- symmetries and “only connectivity matters”

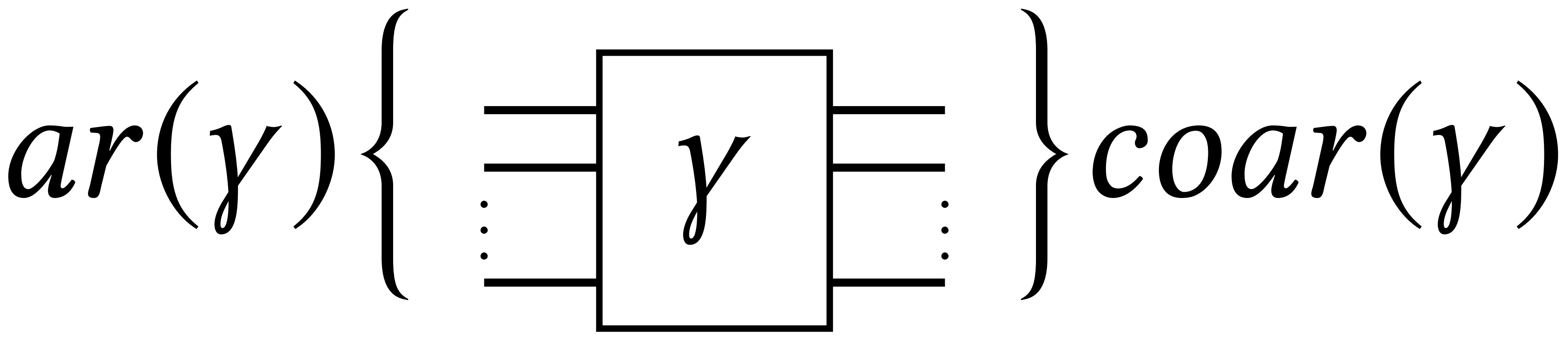


- What are string diagrams exactly? Are they topological objects? Are they combinatorial objects? Are they syntactic objects?
    - Yes

Equpping symmetric monoidal categories with structure

Monoidal theories

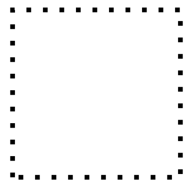
- A **monoidal signature** $\Gamma = (G, ar, coar)$ where G is a set of operations
  - $ar : G \to \mathbf{N}$ is gives arities
  - $coar : G \to \mathbf{N}$ gives **coarities**



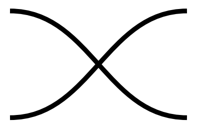
String diagrams as syntax
The free prop on a monoidal signature

- A inductive term language is useful, e.g. we can use structural induction

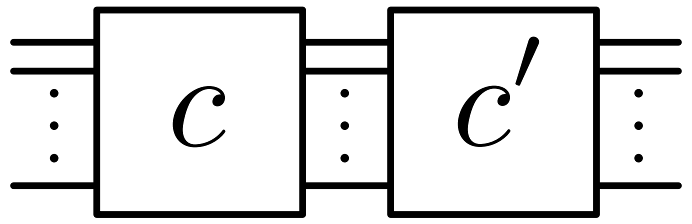
$$\frac{}{\gamma : (ar(\gamma), coar(\gamma))}$$

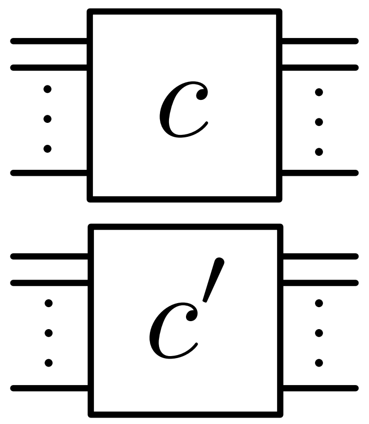
$\frac{}{}$  $:(0, 0)$

$\frac{}{}$ $:(1, 1)$

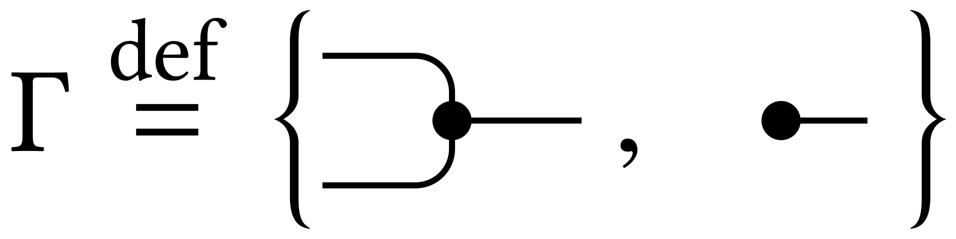
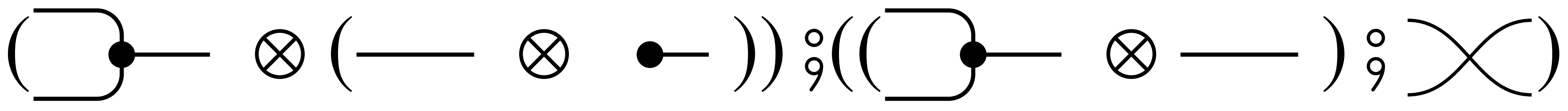
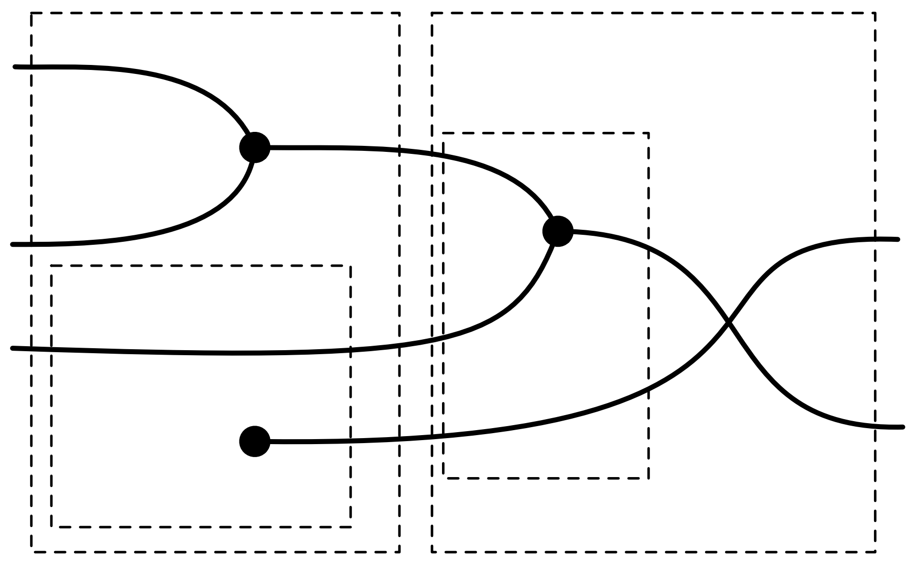
$\frac{}{}$  $:(2, 2)$

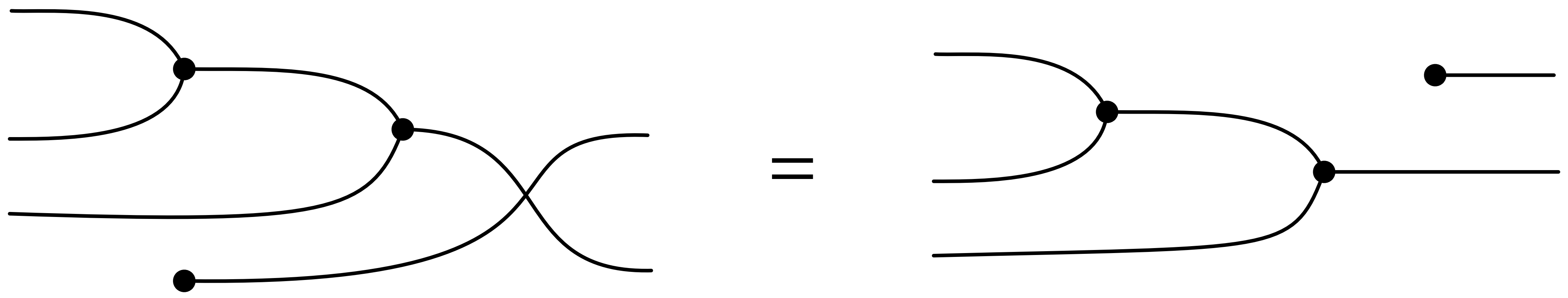
$$\frac{c : (n, z) \qquad d : (z, m)}{c \,;\, d : (n, m)} \qquad \frac{c : (n, m) \qquad d : (r, z)}{c \otimes d : (n{+}r, m{+}z)}$$

$c \,;\, c'$ is drawn 

$c \otimes c'$ is drawn 

From terms to string diagrams

- Consider 
    - then  is drawn 
- to go to string diagrams we need to quotient wrt the laws of symmetric strict monoidal cats. This means that:
    - erasing the dotted lines
    - “only connectivity matters”



- This is a nice description of the free prop on a signature: in particular it is easy to see that given a symmetric monoidal category **X**, an object X∈**X**, and a valuation of each γ ∈ Γ extends uniquely (structural induction) to a symmetric monoidal functor from string diagrams to **X**

A recipe for functorial semantics

- We have notion of syntax, but what should be the semantics?
- Mere symmetric monoidal categories do not have enough structure for a meaningful general solution
- This additional structure (usually a universal property) is the magic potion that makes everything work
- Lawvere discovered this in the 60s for universal algebra, in that case it is the notion of **categorical product**.
  - the “free thing” on the signature is the syntax
  - functorial semantics are functors that preserve the the thing
  - as we will see symmetric monoidal categories are often convenient hosts to study “the thing” from an algebraic perspective

Aside: Lawvere and cartesian categories

- Lawvere wasn't happy with the idea of algebraic theory as we have introduced it in the style of universal algebra (i.e. a pair (Σ, E))
- Equating the notion of theory with a particular presentation is not ideal since different presentations can yield the same notion of algebraic structure
- The syntactic account has an ad hoc underlying meta-theory: e.g. inductively defined terms over a fixed countable set of variables, meta theory of substituions, etc.

Abstract universal algebra

- Equate a theory with a category **L** with finite products (single sorted: with one generating object)
- doesn't suffer from reliance on particular presentations
- e.g. for commutative monoids, take the free category generated by {m,e}, quotient by least congruence generated by eqs
- A (classical) model is a product preserving functor **L** → **Set**
- Model homomorphisms are natural transformations
- Simple, beautiful, easily generalisable

Finite products

- The **category with free finite products on one object** is FinSet^{op}
- FinSet^{op} has (up to equivalence) an alternative "operational" description
  - objects: natural numbers, we think of $m = \{x_1,x_2,\ldots,x_m\}$
  - arrows $m \to n$: n-tuples of variables in $\{x_1,x_2,\ldots,x_m\}$, e.g.
    - there is exactly one arrow $1 \to 2$: (x_1,x_1)
    - there are two arrows $2 \to 1$: (x_1) and (x_2)
  - composition is **substitution**: e.g. $(x_1,x_1);(x_2) = x_1$

Finite products ctd

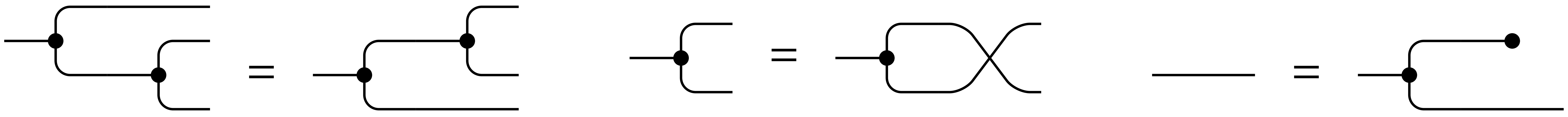
- The category with **free finite products on a signature Σ** has a similar operational description
  - objects: natural numbers, we think of $m = \{x_1, x_2, \ldots, x_m\}$
  - arrows $m \to n$: n-tuples of terms in $T_{\{x1,x2,\ldots,xm\}}$, e.g. for the sig of monoids
    - there is an arrow $1 \to 2$: (x_1, e)
    - there is an arrows $2 \to 1$: (m)
    - composition is **substitution**: e.g. $(x_1, e); (m) = m(x_1, e)$

Terms demystified!
The algebra of terms and substitution is simply a convenient description of a category with free products

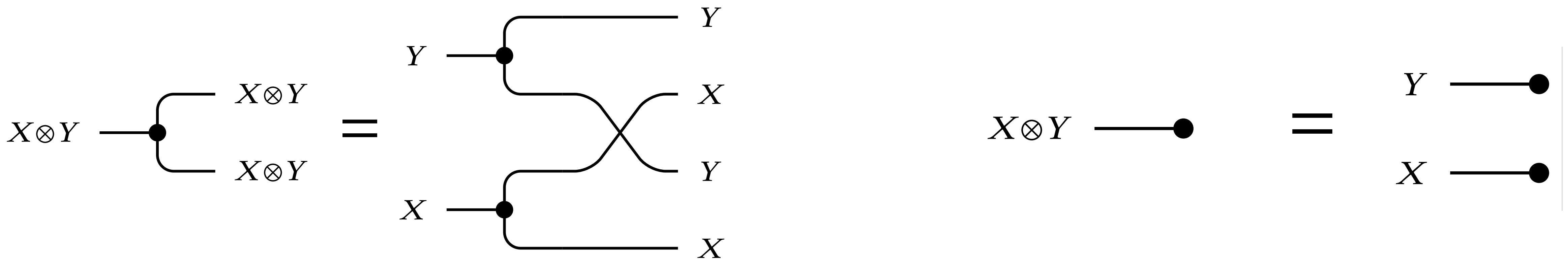
Algebraic structure in Set

Cartesian categories

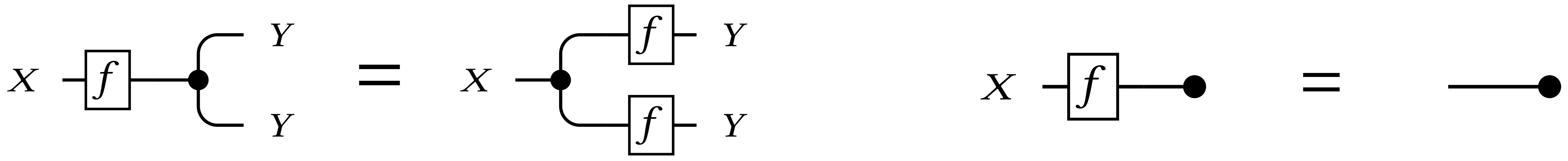
- A symmetric monoidal category is cartesian when the monoidal product satisfies the universal property of categorical product
- The symmetric monoidal category **Set** is (by definition) such an animal
- **Theorem (Fox 1976).** A symmetric monoidal category is cartesian iff every object can be equipped with a commutative comonoid structure which is coherent and natural.



coherent:



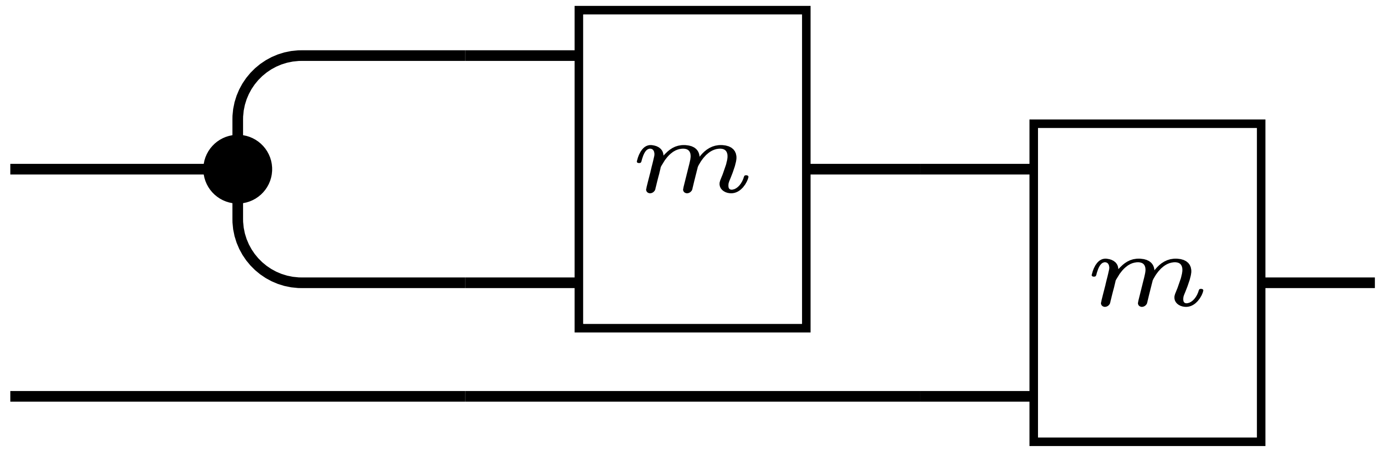
natural:



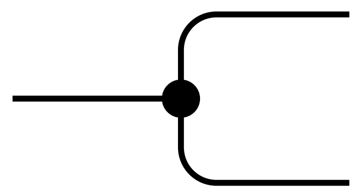
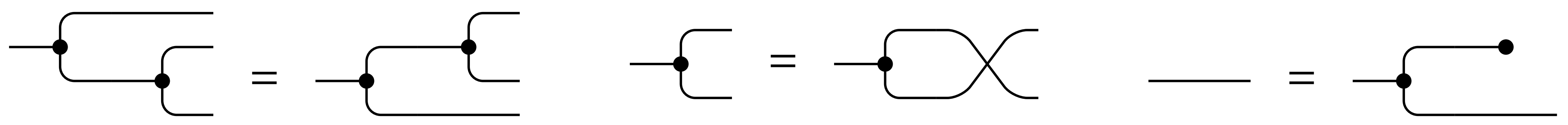
Lawvere with string diagrams

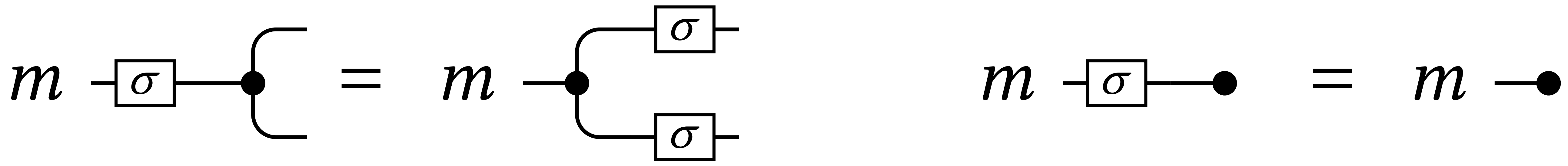
- **A single sorted Lawvere theory is a cartesian prop**
  - i.e. a prop where the monoidal product is the categorical product
- We already have one concrete description of the free cartesian category on a signature - arrows: classical terms, composition: substitution
- We now have a second: string diagrams!

$$m(m(x,x),y) \quad \Leftrightarrow$$



A recipe

- Turn a theory into a monoidal theory in two easy steps
  - Generators: $\Gamma \overset{\text{def}}{=} \Sigma +$  
  - Equations: E (as string diagrams) + 

+ 

e.g. as props, the Lawvere theory of commutative monoids is isomorphic to the monoidal theory of commutative bialgebras!

Diagrammatic relational algebra and applications

CATMI, Bergen, June 26-30 2023

Pawel Sobocinski, Tallinn University of Technology

Roadmap

- Lecture 1 - Functorial semantics 1
- **Lecture 2 - Functorial semantics 2**
- Lecture 3 - Graphical linear algebra and applications

Recap from yesterday, plan for today

- Yesterday
  - traditional syntax and universal algebra
  - cartesian products and Lawvere theories
  - functorial semantics, models as functors to **Set**
  - symmetric monoidal categories as carriers of algebraic structure
  - Fox's Theorem: characterising cartesianity with algebraic structure — the presence of commutative comonoid structure that is coherent and natural
- Today
  - Replacing **Set** with **Par** and **Rel**
  - partial theories (joint work with Di Liberti, Loregian and Nester)
  - relational theories (joint work with Bonchi and Pavlovic, continued by Nester)
  - first-order theories (work in progress with Bonchi, Di Giorgio and Haydon)

The recipe for functorial semantics

- find out the universal property at play
  - for traditional algebraic theories, this is (binary) categorical products
- find an algebraic characterisation in symmetric monoidal categories a la Fox
  - for the categorical product, this is the commutative comonoid structure that’s coherent and natural
- Then:
  - **syntax** = string diagrams with the structure (the free thing!)
  - **semantics**, any category with the universal property
    - for traditional algebraic theories, this is usually **Set**, but not always
  - **models** = functors that preserve the structure
  - **homomorphisms** = the canonical notion of natural transformation

The symmetric monoidal category **Par**

- objects are sets
- arrows are partial functions
- monoidal product is cartesian product
- symmetries are inherited from **Set**
- there is a natural poset enrichment

- if **C** has finite limits, there is a symmetric monoidal category Par(**C**)
  - objects are those of C
  - arrows from C to D are spans C ← C’ → D (up to iso) where the left leg is mono
  - composition is by pullback
  - monoidal product is pointwise product

- NB: The monoidal product in **Par** is **not** the categorical product!

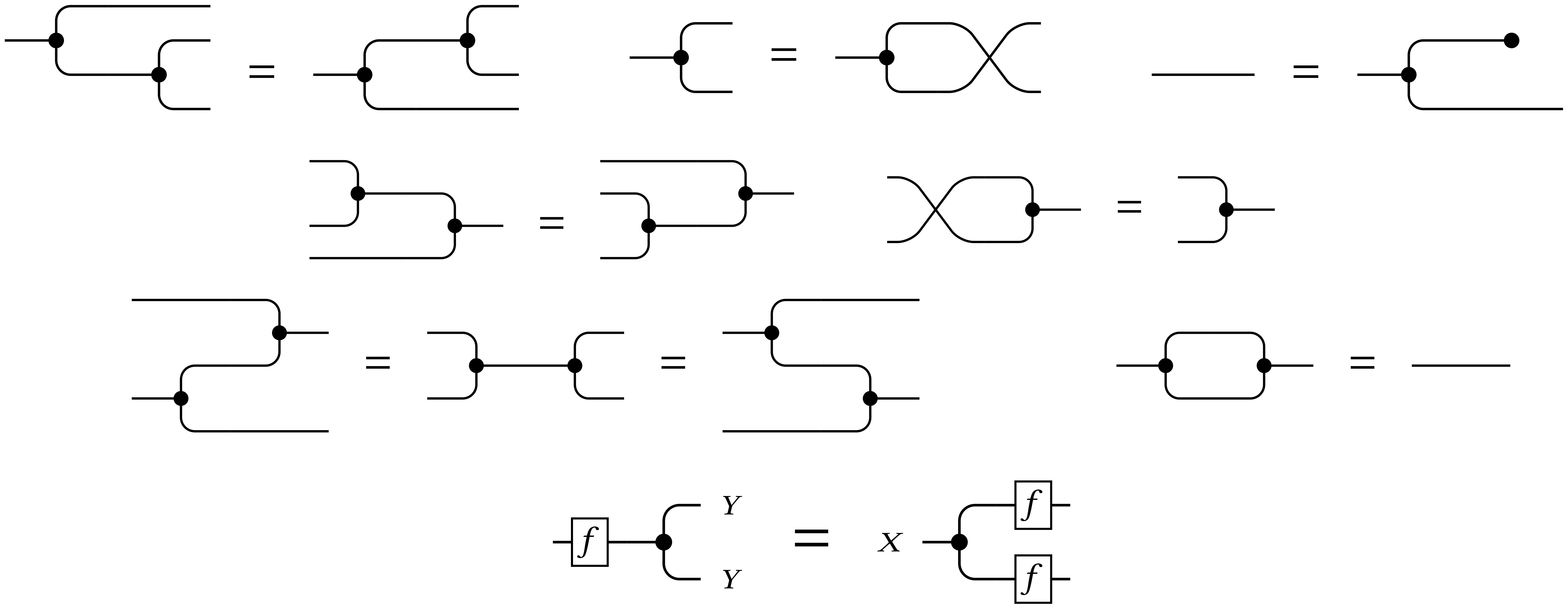
Algebraic structure in Par

Discrete cartesian restriction categories

- Partial theories: we want to replace **Set** with **Par** as the universe of models
- Lawvere identified cartesian categories as the categorical structure of interest for algebraic theories
- For partial theories, the corresponding categorical structure is given by **discrete cartesian restriction categories** (**dcr categories**)
  - **Par** is a DCR category. If **C** has finite limits, Par(**C**) is a DCR category.
- Instead of delving into the details, we can characterise them using a result similar to Fox's theorem

"Fox's theorem" for DCR categories

- **Theorem.** A **DCR category** is a symmetric monoidal category where every object is equipped with a coherent **partial Frobenius algebra** structure, such that the comultiplication is natural.



Consequences

- The free DCR category on an object is Par($\mathbf{F}^{op}$)

Given a signature Σ, we obtain a syntax for equations!

- Syntax = concrete description of the free DCR category on Σ in terms of string diagrams with partial Frobenius structure
- A presentation is then, as usual, the pair of a signature and equations
- its partial **Lawvere theory is the induced DCR prop**
- This is now a Lawvere-style functorial semantics for partial theories

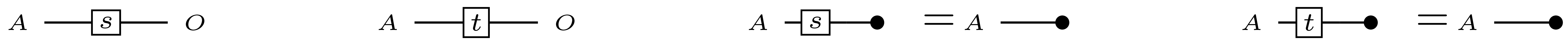
Functorial semantics for partial theories

- We have
  - a notion of syntax - string diagrams with the additional algebraic structure
  - a notion of semantics, any DCRC, but **Par** is a canonical choice
  - a notion of model, a functor syntax→semantics that preserves the DCRC structure
  - a notion of model homomorphism given by the canonical notion natural transformations of such functors

Examples

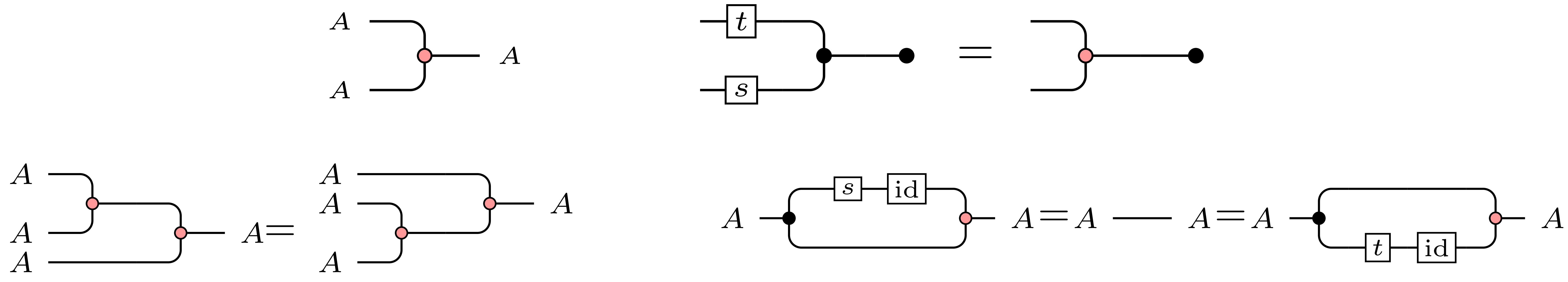
2-sorted

directed graphs



reflexive graphs

categories



+ monoidal categories, cartesian restriction categories, DCR categories, cartesian categories, cartesian closed categories, …

The symmetric monoidal category **Rel**

- objects are sets
- arrows from X to Y are relations $R \subseteq X \times Y$
    - composition is relational composition: given $R \subseteq X \times Y$ and $S \subseteq Y \times Z$, the composition $R;S = \{ (x,z) \mid \exists\, y.\ (x,y) \in R \wedge (y,z) \in S \}$
- poset enrichment: 2-cells are inclusions of relations
- monoidal product is cartesian product on objects. On arrows, given $R \subseteq X \times Y$ and $R' \subseteq X' \times Y'$, $R \otimes R' = \{ (xx', yy') \mid xRy \wedge x'R'y' \}$

- given a regular category **C**, there is a monoidal category Rel(**C**) with
    - objects are those of **C**
    - arrows are jointly mono spans $X \leftarrow R \rightarrow Y$, composition is pullback followed by factorisation
    - monoidal product is given by pointwise product

- NB. the monoidal product in **Rel** is **not** the categorical product

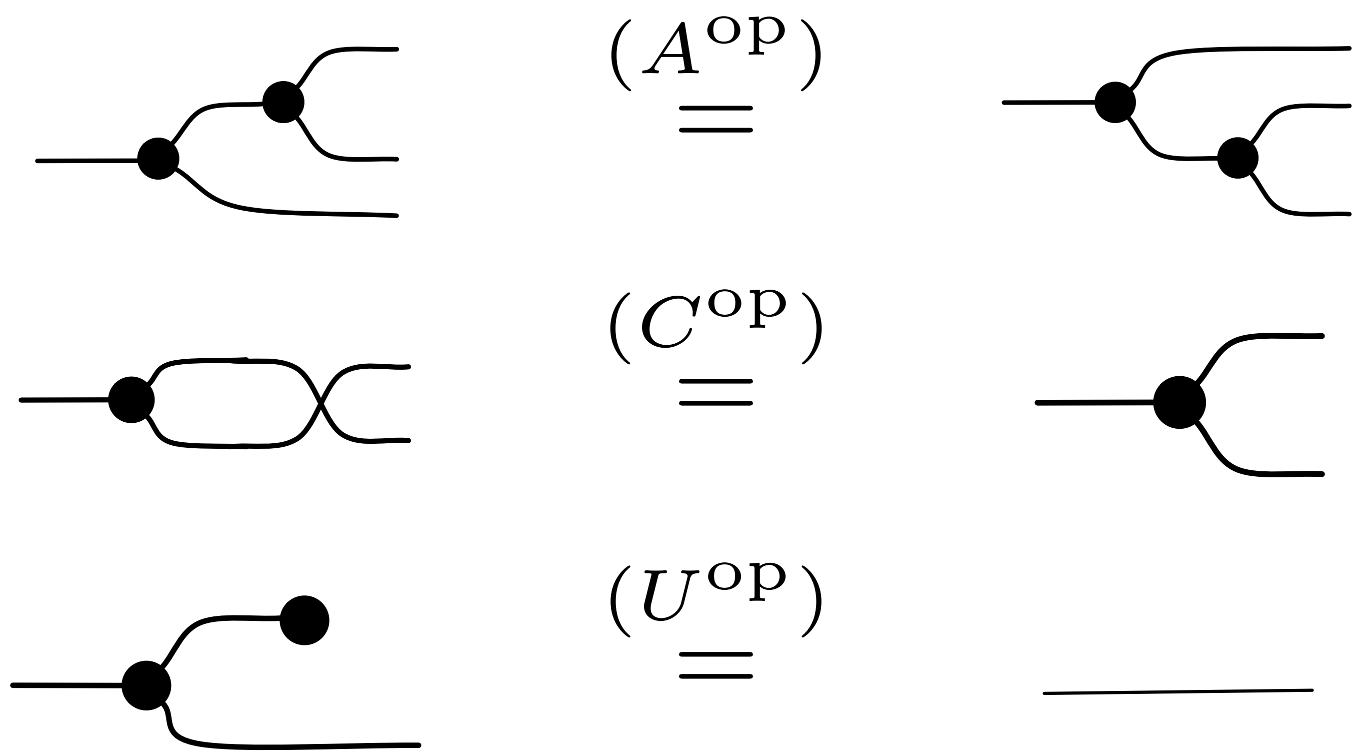
Algebraic structure in Rel I

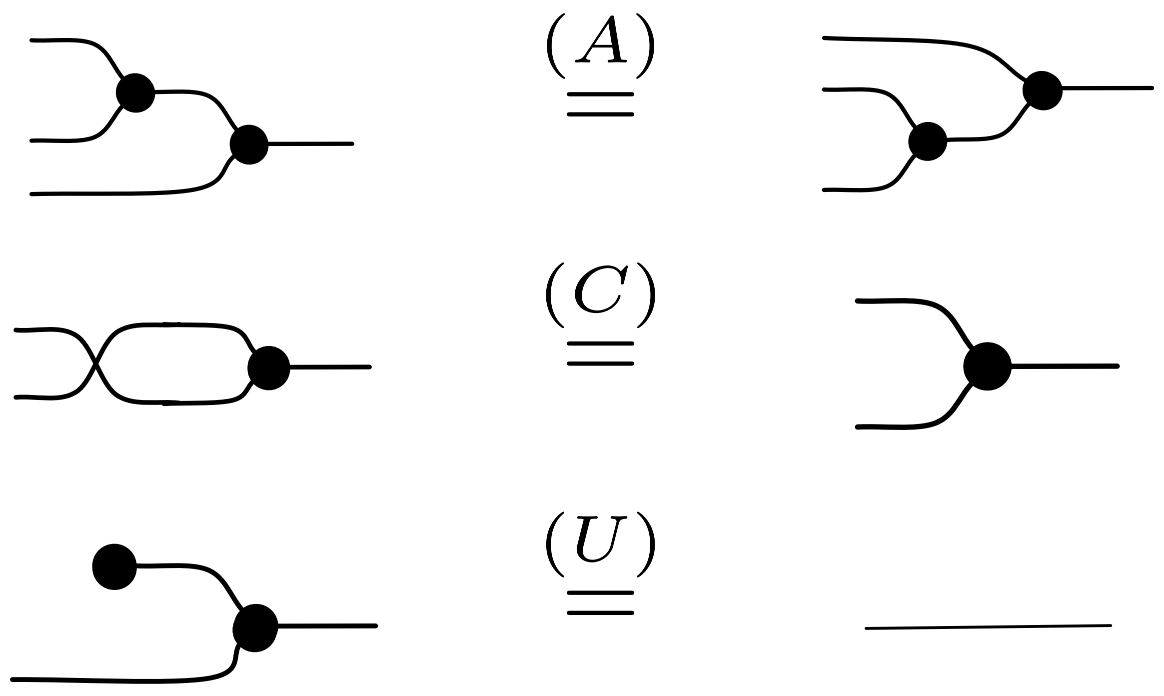
Cartesian bicategories (of relations)

- every object has a commutative comonoid structure
- with right adjoints
- s.t. every morphism is a weak comonoid homomorphism
- and the comonoid and monoid structures together form a special Frobenius monoid

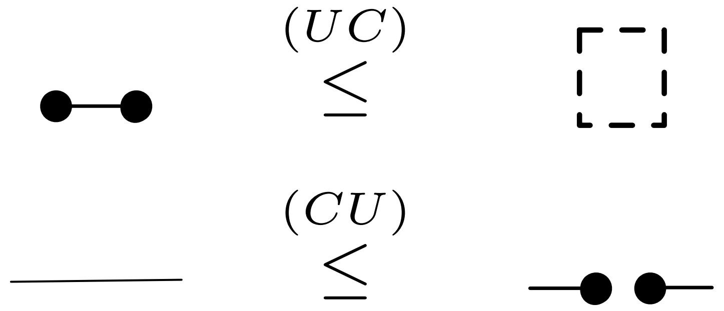
- **cartesian bicategories** are a general, category theoretic algebraic approach to relations (cf. allegories)

Unpacking this data, algebraically

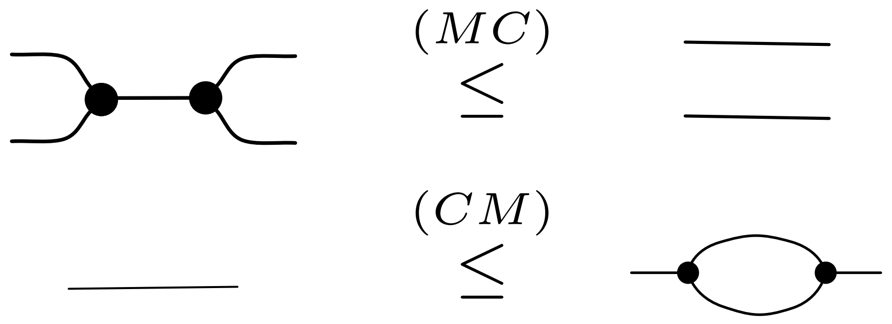




(unit is right adjoint to counit)



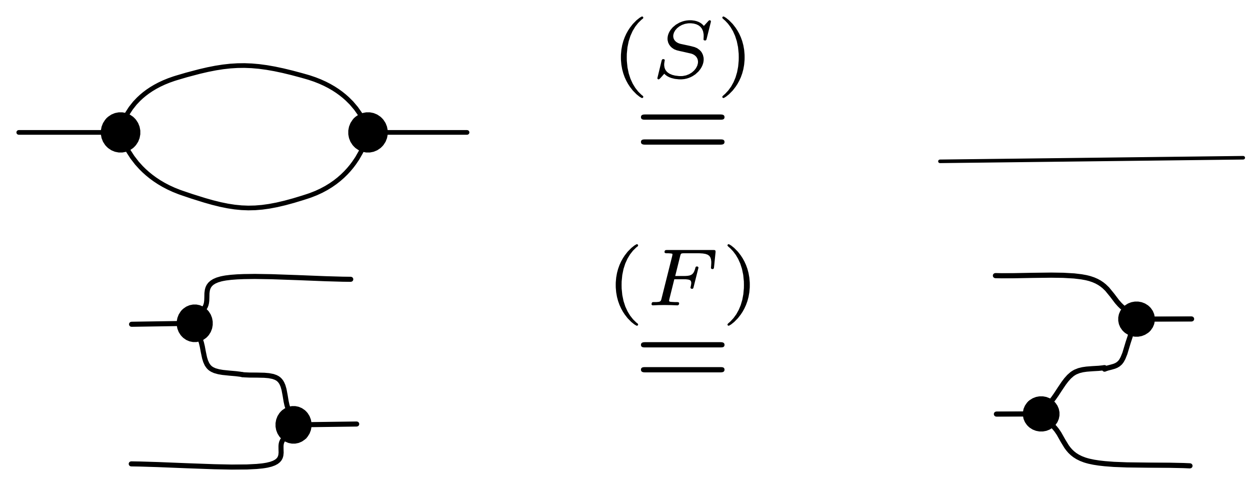
(multiplication is right adjoint to comultiplication)



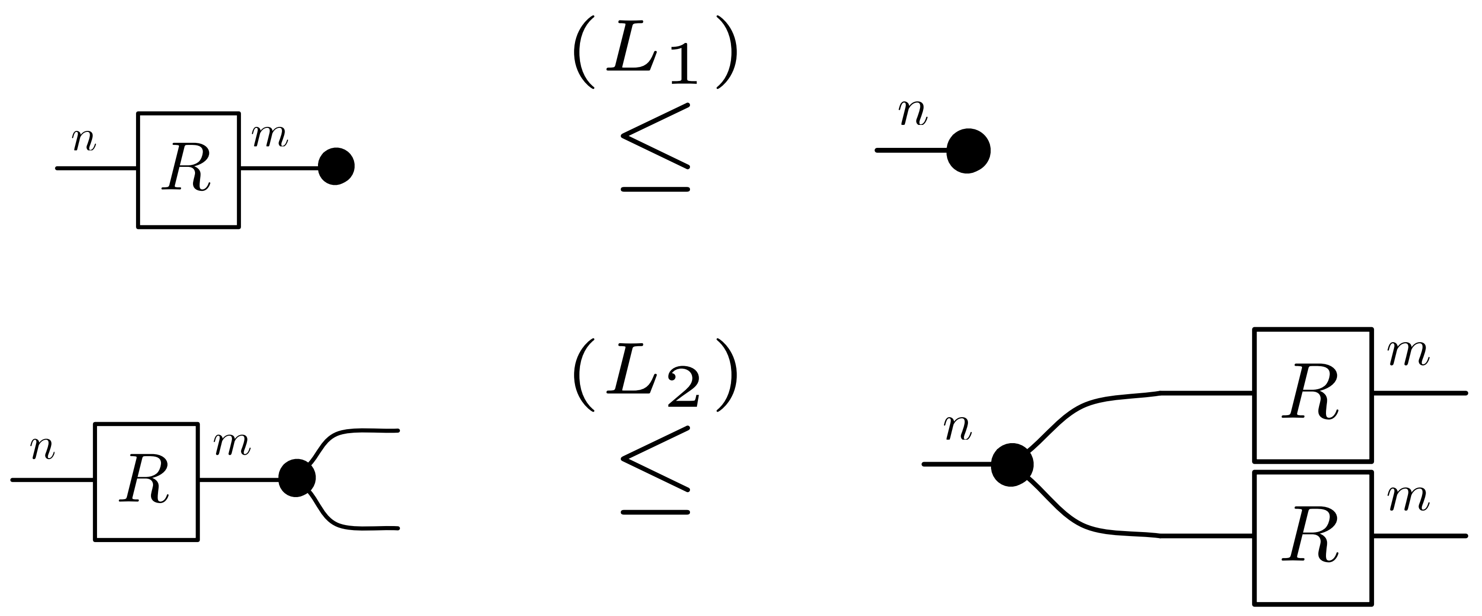
Convolution is a meet semilattice

The Frobenius law and lax naturality

(special Frobenius)



(all relations are weak comonoid homomorphisms)



Functorial semantics for relation theories

- We have
  - a notion of syntax - string diagrams with the additional algebraic structure
  - a notion of semantics - any cartesian bicategory of relations, but **Rel** is the canonical choice
  - a notion of model - functors syntax→semantics that preserve the cartesian bicategory structure
  - a notion of homomorphism, given by the canonical notion of natural transformations of such functors

A curious property of Rel

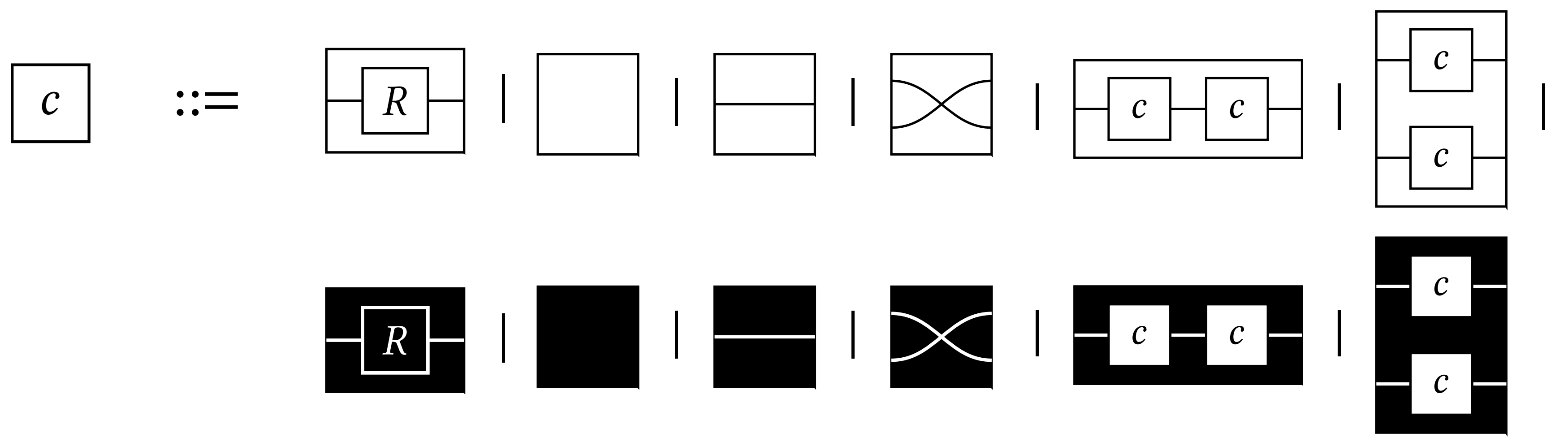
- There is a “De Morgan” version or **Rel** as a monoidal category.
- From now, let us call the usual one **Rel**$^+$. The other we will call **Rel**$^-$.

- Both **Rel**$^+$ and **Rel**$^-$ have the same objects, and monoidal product on objects is cartesian product. But:
  - **Rel**$^-$ composition works as follows: given $R \subseteq X \times Y$ and $S \subseteq Y \times Z$,
    - $R;S = \{ (x,z) \mid \forall\, y.\ xRy \vee ySz \}$
    - what is the identity?
  - On arrows, given $R \subseteq X \times Y$ and $R' \subseteq X' \times Y'$,
    - $R \otimes R' = \{ (xx', yy') \mid xRy \vee x'R'y' \}$
    - what are the symmetries?

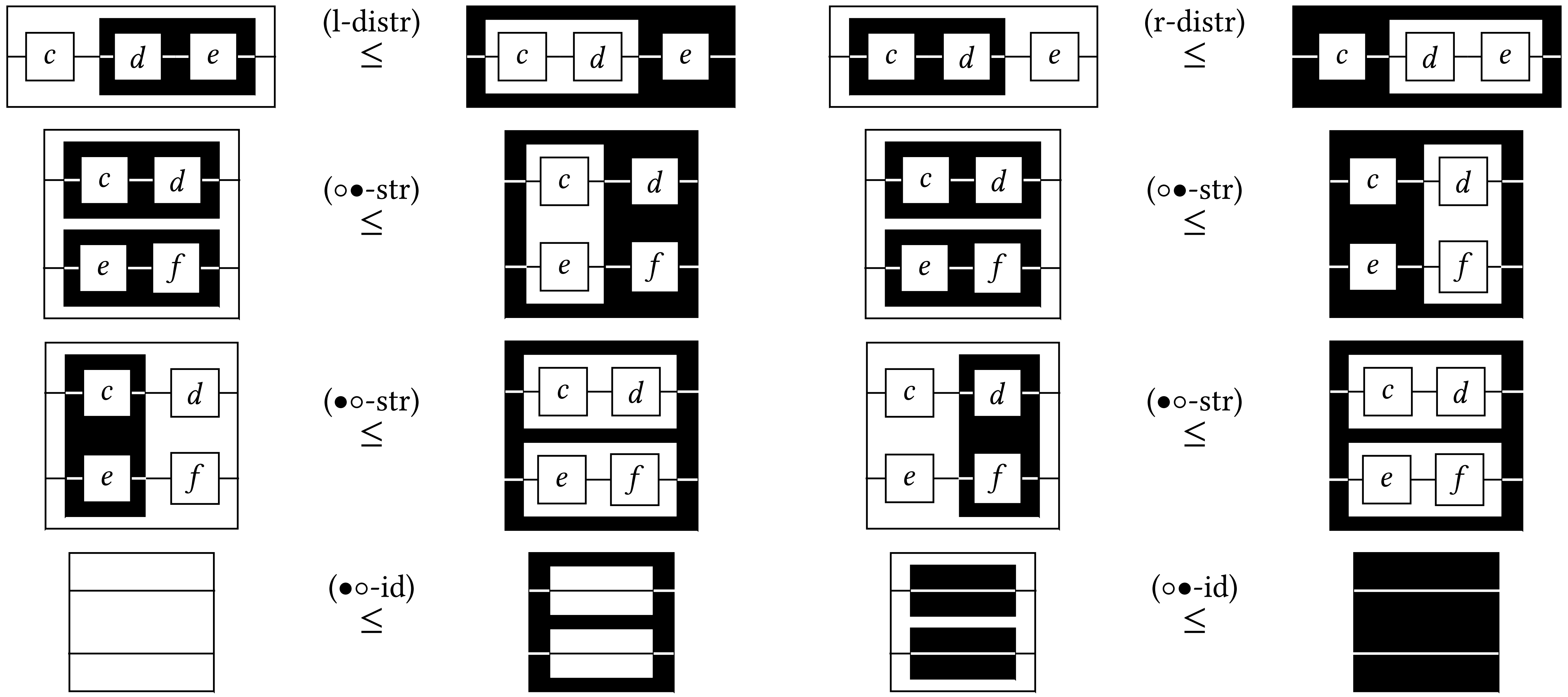
- **Rel**$^+$ and **Rel**$^-$ are isomorphic as symmetric monoidal categories

The linear bicategory Rel

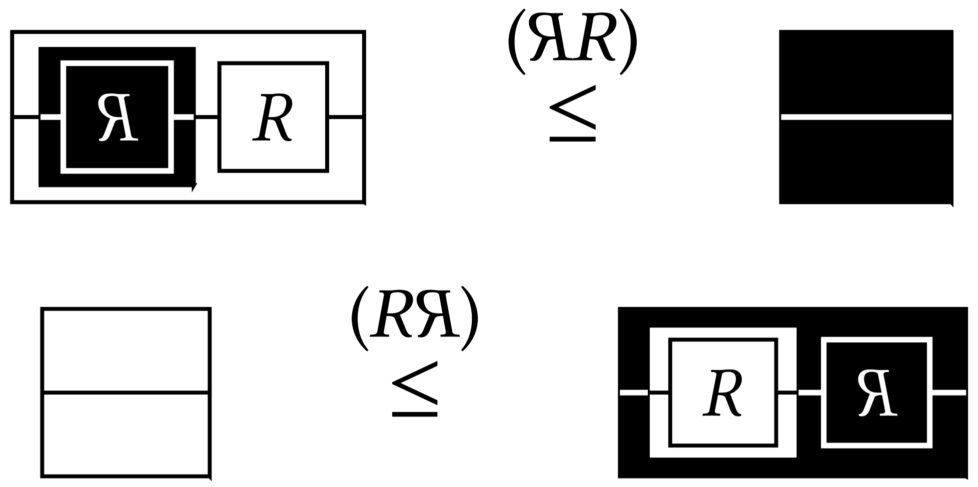
- there are two compositions and two tensors



- satisfying linear distributivity

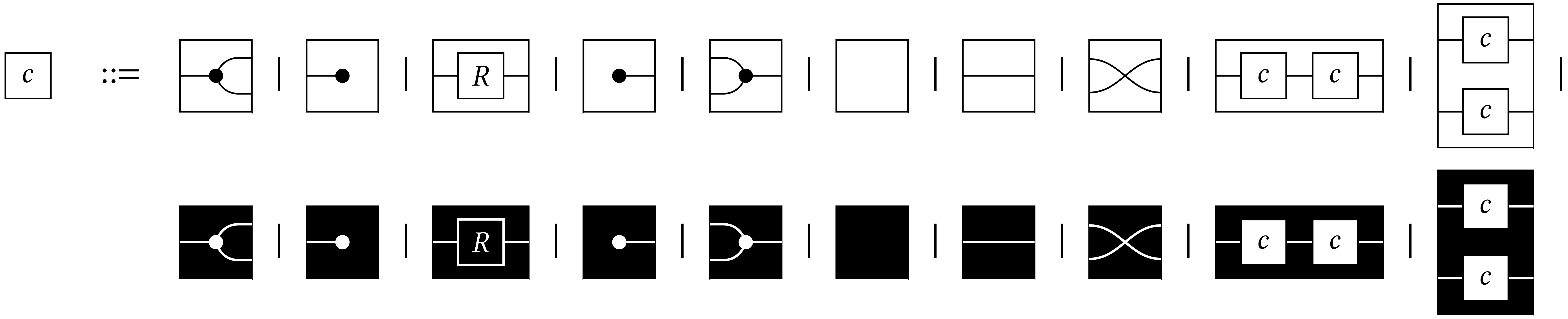


- and linear adjunctions



Cartesian bicategories + linear bicategories = first order bicategories

… a totally algebraic approach to first-order logic and first order theories



One can prove completeness (in Gödel’s sense, in the style of Henkin) in this theory

First order theories, algebraically

sans variables, quantifiers…

- natural encodings of various flavours of relational algebras
- diagrammatic syntax closely related to 19th century string diagrams: Peirce’s existential graphs
- a variable free treatment of first order logic, with a sound and complete axiomatisation
- easy encoding of Quine’s predicate functor logic
- a functorial semantics story, in the style we have seen so far

Summary

- Lawvere identified a universal property - cartesian products - which via Fox’s theorem gives you an algebraic structure
- Such algebraic structures can be studied as additional structure on symmetric monoidal categories
- Once you know the universal property <-> algebraic structure, the entire functorial semantics story falls out, we have:
  - a notion of **syntax** - string diagrams with the additional algebraic data built in
  - a notion of semantic universe - any category with the right structure, but typically there is a “canonical” one - **Set**, **Par**, or **Rel** in the examples
  - a notion of model - a functor syntax→semantics that preserves the structure
  - a notion of homomorphism - natural transformations

LIE-STØRMER CENTER

TAL TECH

Diagrammatic relational algebra and applications

CATMI, Bergen, June 26-30 2023

Pawel Sobocinski, Tallinn University of Technology

Roadmap

- Lecture 1 - Functorial semantics 1
- Lecture 2 - Functorial semantics 2
- **Lecture 3 - Graphical linear algebra and applications**

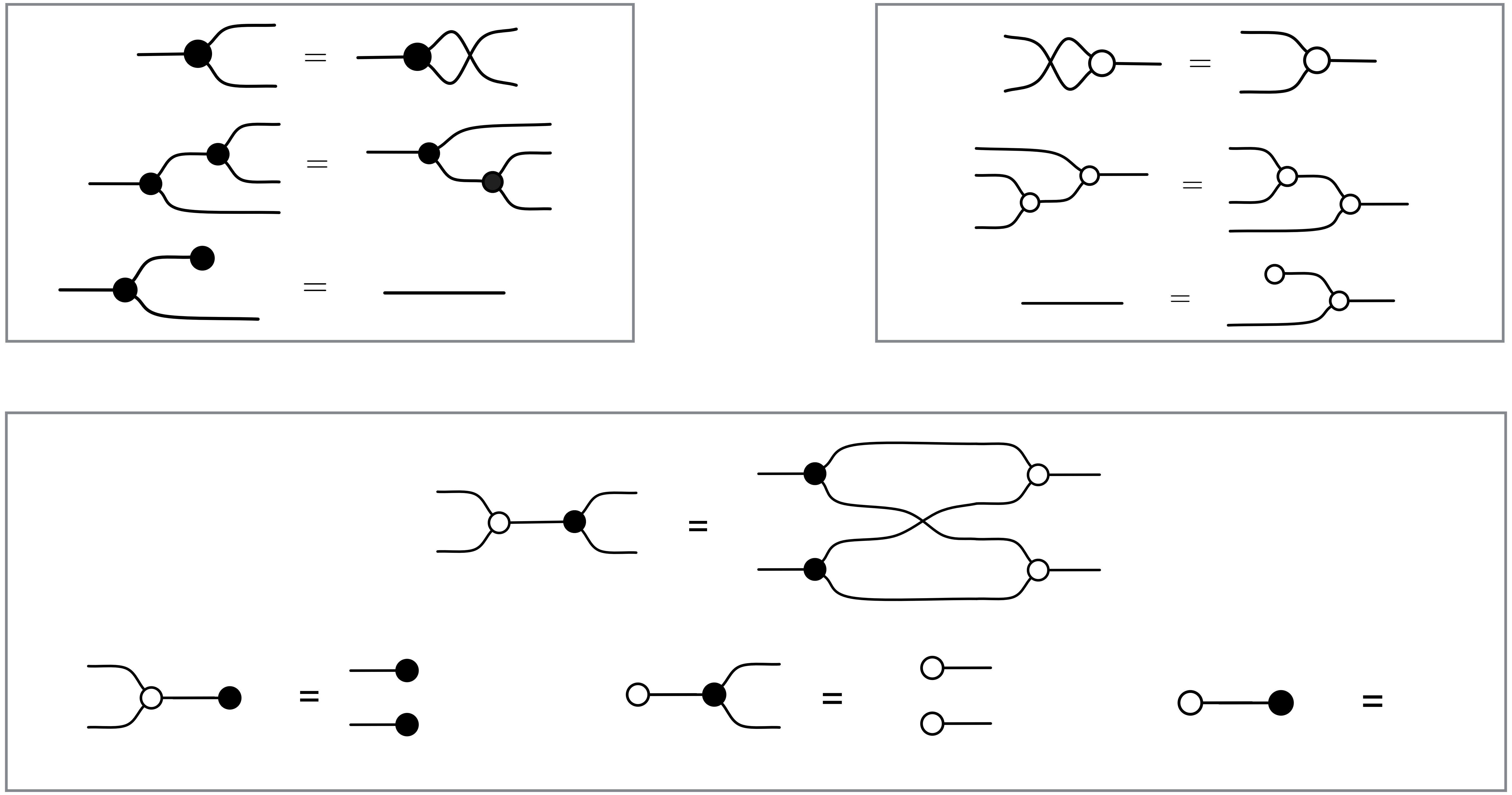
Recap and roadmap

- In the first two lectures we generalised Lawvere's functorial semantics to
  - partial algebraic theories
  - relational theories
  - first-order theories
- in each case string diagrams in symmetric monoidal categories are useful as carriers of the relevant categorical structure, seen algebraically
- in particular, they give us a nice syntactic calculus
- **Today.** Two relational theories: graphical linear algebra and graphical affine algebra

Two symmetric monoidal categories

- Our task is to axiomatise the following:
  - Given a field k, **LinRel$_k$** is the smc where
    - objects are natural numbers
    - arrows m to n are **linear relations** $m \to n$
      - i.e. those relations $R \subseteq k^{m+n}$ that are also k-linear subspaces
    - (ordinary) relational composition of linear relations is a linear relation
  - Similarly, **AffRel$_k$** is the smc where arrows are **affine relations**
    - Given a field k, an **affine relation** $m \to n$ is a relation $R \subseteq k^m \times k^n$ which is either empty, or s.t. there is a linear relation C and a vector (a,b) s.t. $R = (a,b) + C$
    - relational composition of affine relations is an affine relation

Starting point: the theory of bialgebras



Let **B** be the free prop on this data - we know that it is isomorphic to the Lawvere theory of commutative monoids

First glimpse of linear algebra

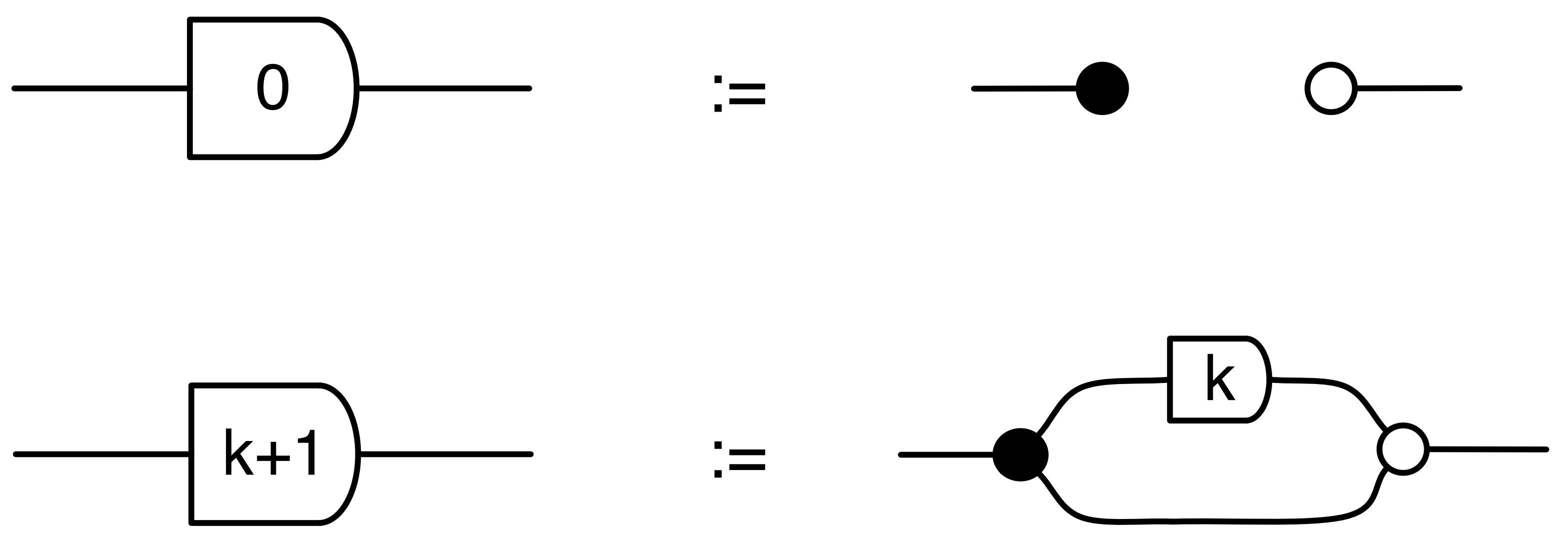
- let Mat be the prop where arrows m→n are n×m matrices of natural numbers
  - e.g. $\begin{pmatrix} 0 & 5 \end{pmatrix} : 2 \to 1$ $\begin{pmatrix} 3 \\ 15 \end{pmatrix} : 1 \to 2$ $\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} : 2 \to 2$
- composition is matrix multiplication
- monoidal product is direct sum

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

- symmetries are permutation matrices
- it's also true that **B** ≅ **Mat**

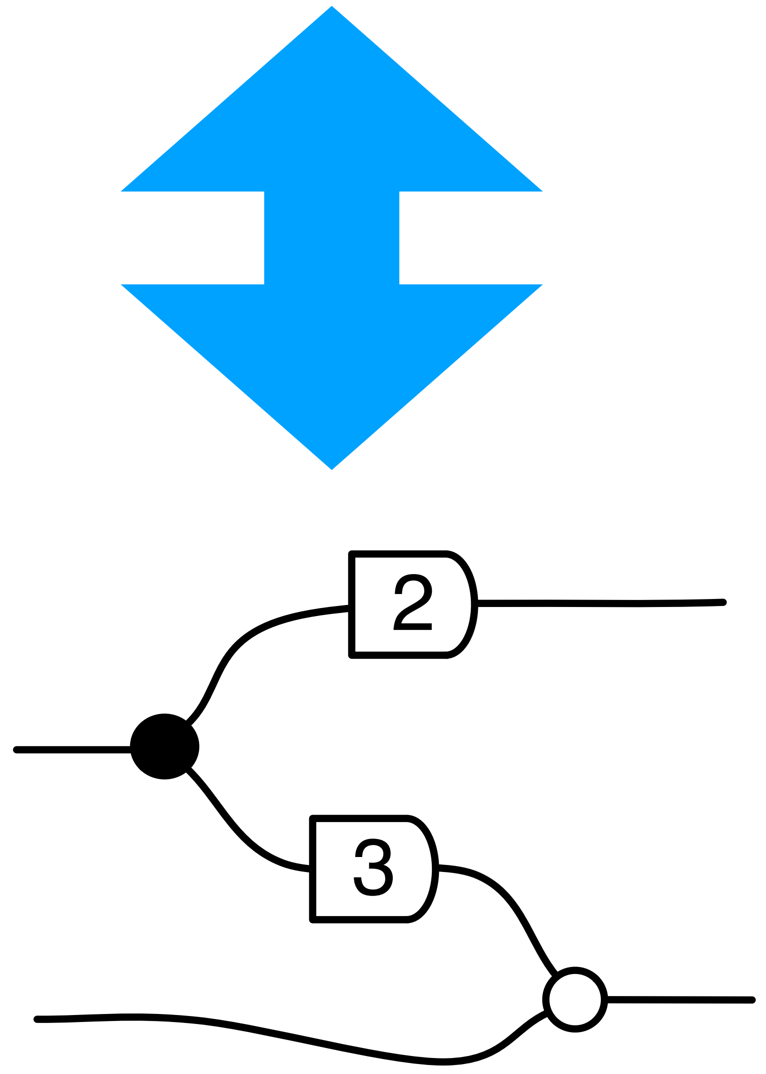
Where do the naturals come from?

- A syntactic sugar:



+1 is “add one path”

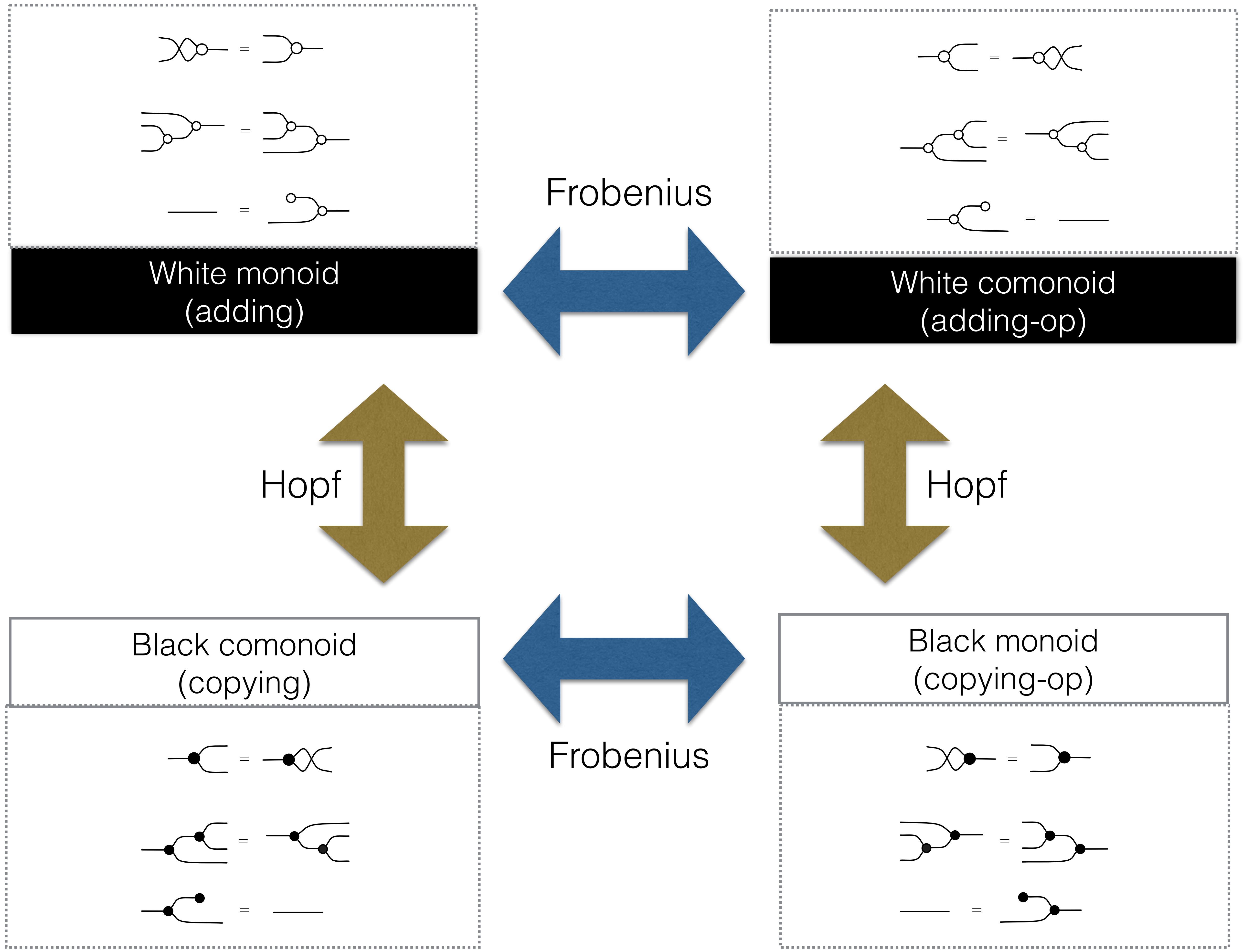
$$\begin{pmatrix} 2 & 0 \\ 3 & 1 \end{pmatrix}$$

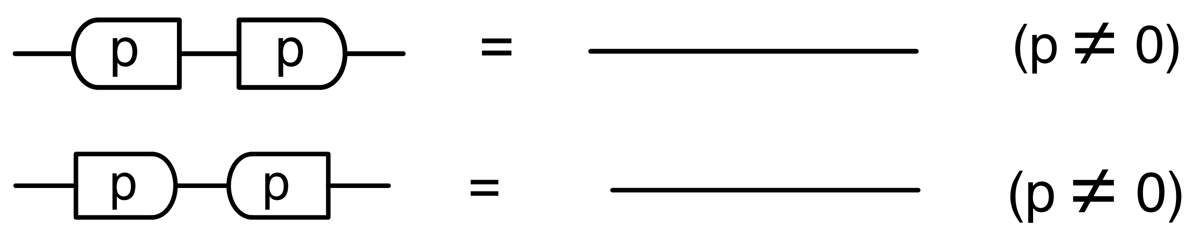


- For similar reasons, the following are isomorphic
  - monoidal theory of Hopf algebra **H**
  - Lawvere theory of abelian groups
  - The prop of matrices over the integers

The relational theory of linear relations

Interacting Hopf algebras aka graphical linear algebra



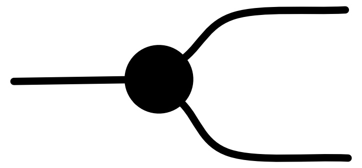


This is the relational theory of linear relations. Moreover:

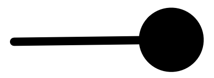
$\mathbf{IH} \cong \mathbf{LinRel}_{\mathbf{Q}}$

IH $\cong$ LinRel$_{\mathbb{Q}}$

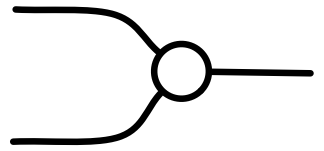
Where do the generators go?



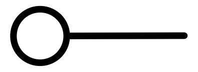
$$\left\{ \left(x, \left(\begin{smallmatrix} x \\ x \end{smallmatrix}\right)\right) \right\}$$



$$\{ (x, \bullet) \}$$



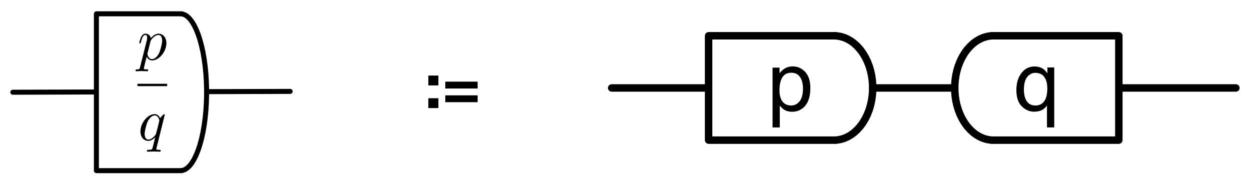
$$\left\{ \left(\left(\begin{smallmatrix} x \\ y \end{smallmatrix}\right), x+y\right) \right\}$$



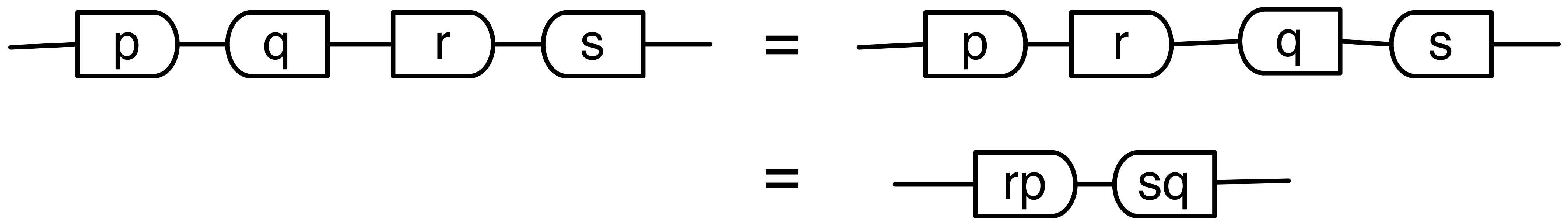
$$\{ (\bullet, 0) \}$$

Linear algebra = how these four relations and their opposites interact

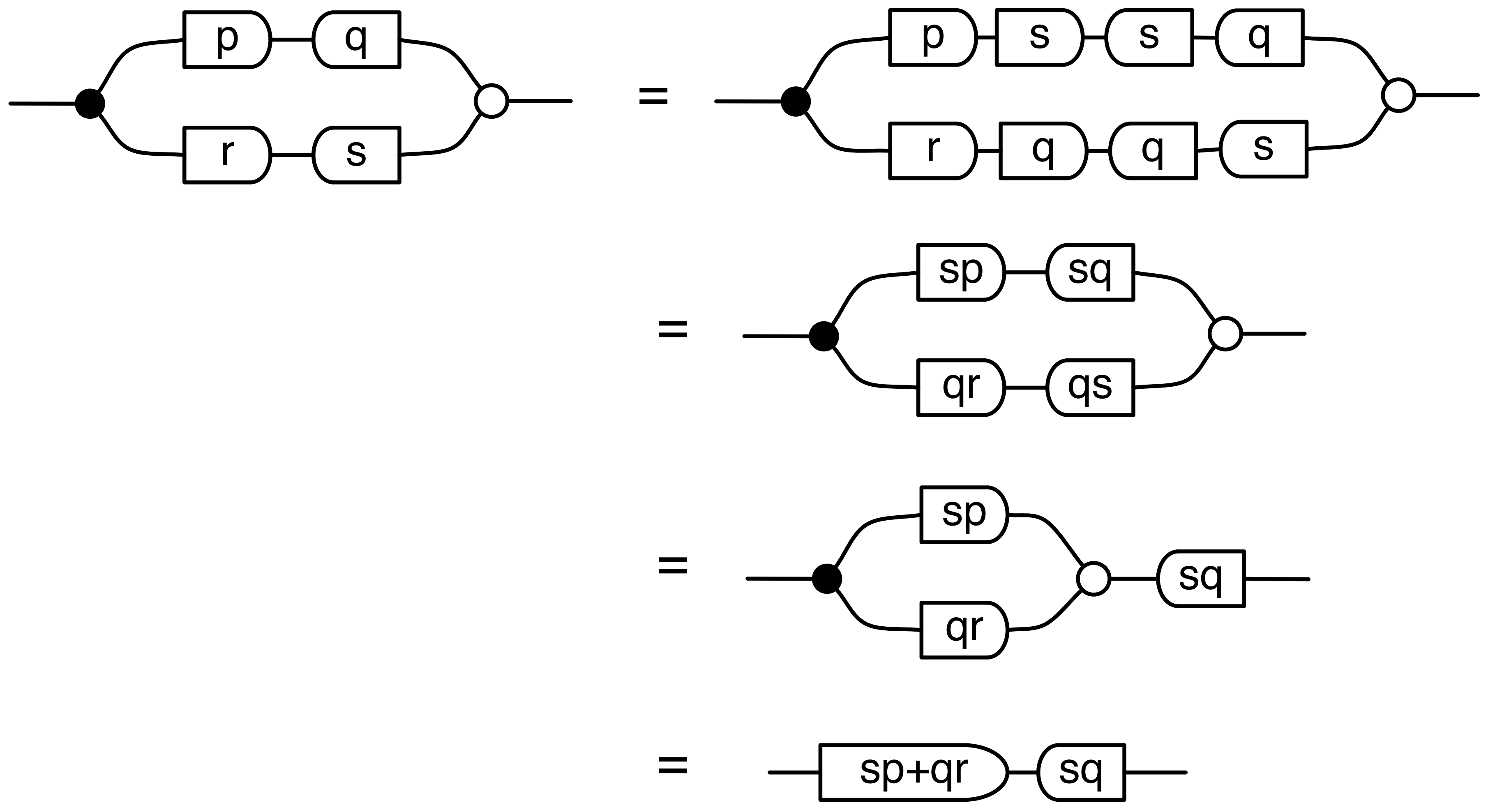
Where do the rationals come from?



multiplication:

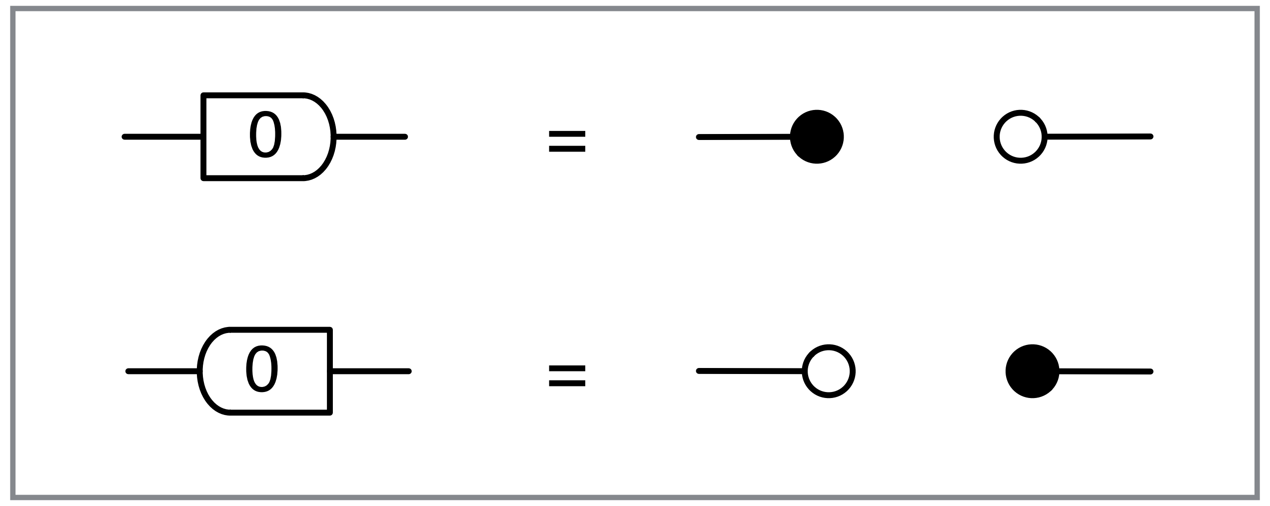


addition:

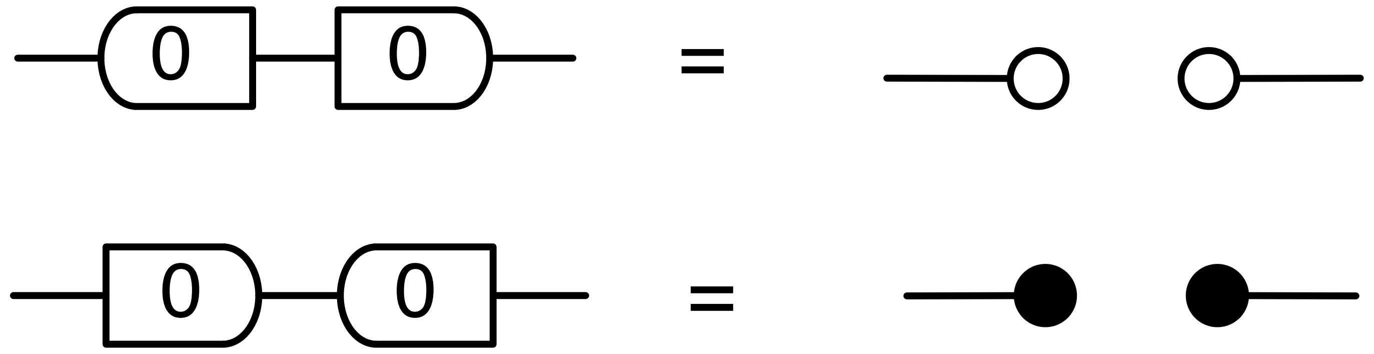


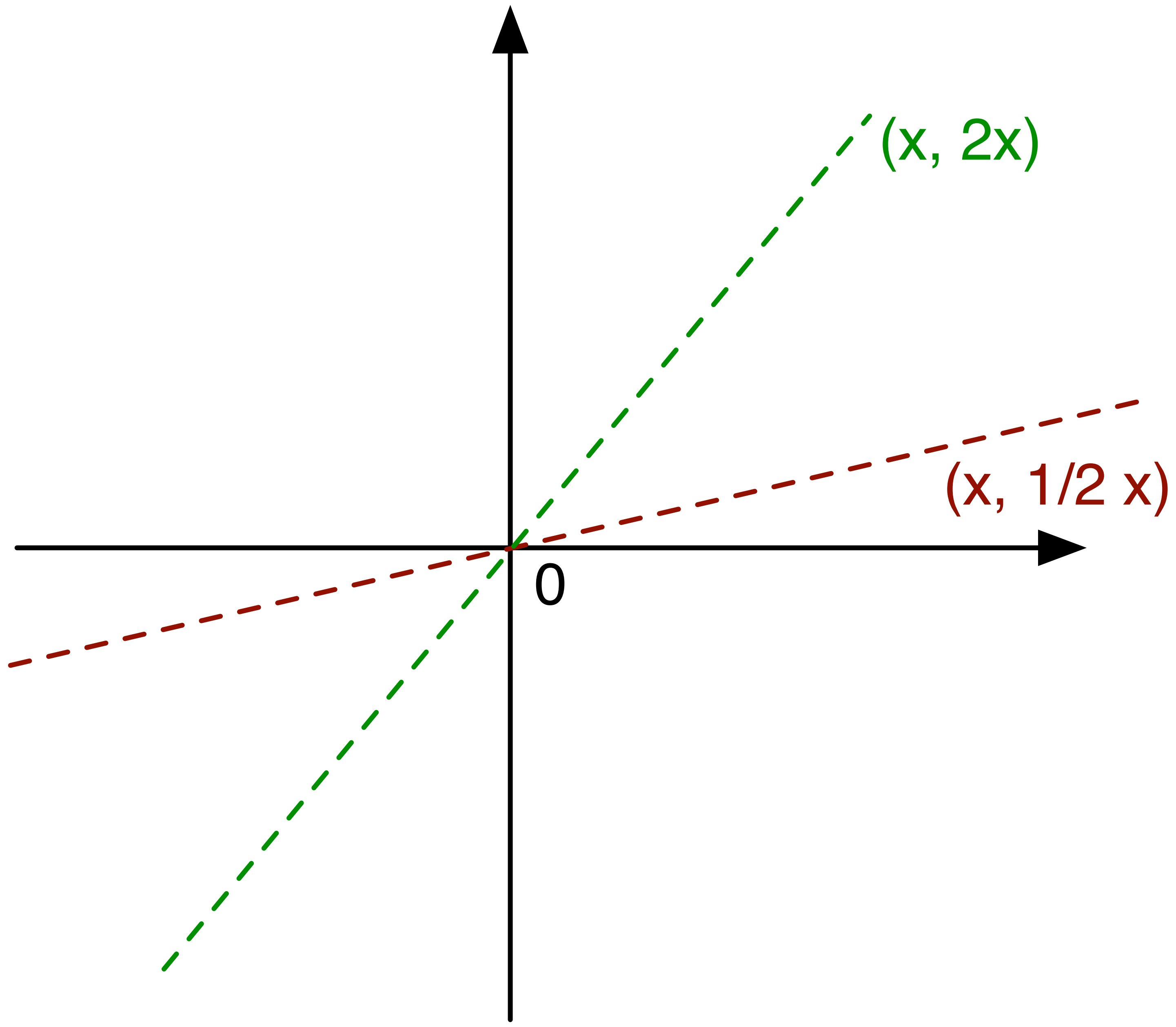
But what about division by 0?

- it’s ok, nothing blows up



- two ways of interpreting 0/0



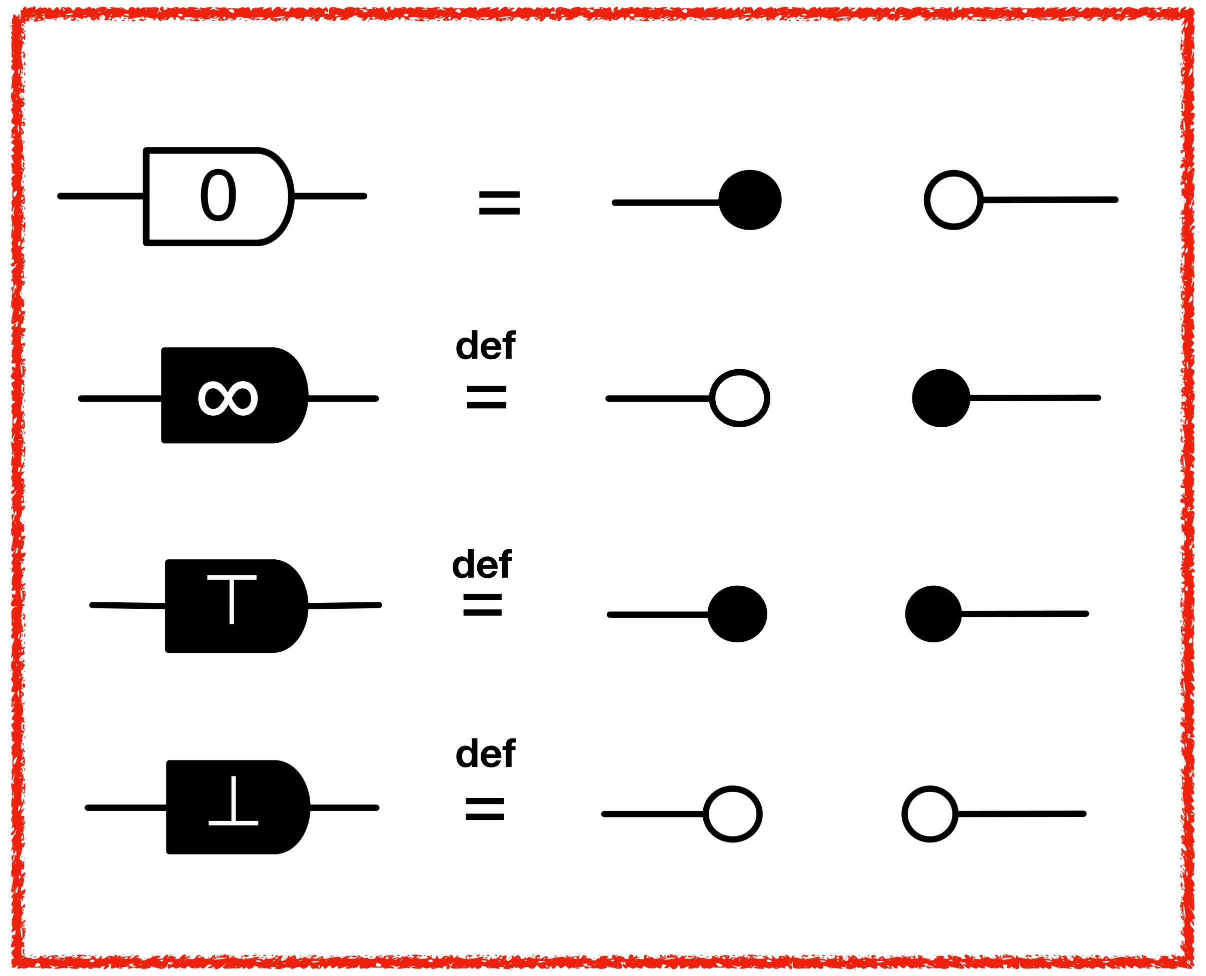


An extended number system

- $\mathrm{LinRel}_{\mathbf{Q}}[1,1]$
- projective arithmetic with two additional elements
  - the unique 0-dimensional subspace $\perp = \{ (0,0) \}$
  - The unique 2-dimensional subspace $\top = \{ (x,y) \mid x,y \in \mathbf{Q} \}$

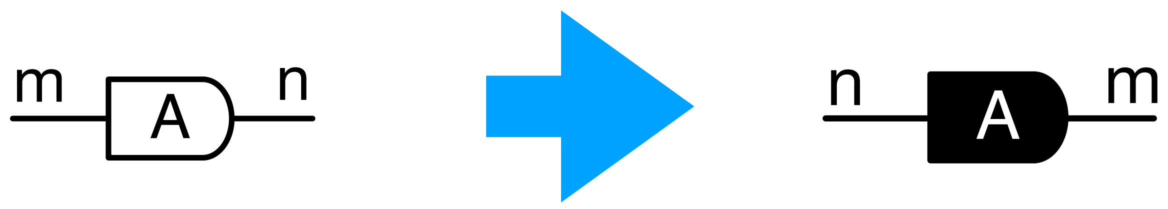
$+$	0	r/s	∞	$\top$	$\perp$
0	0	r/s	∞	$\top$	$\perp$
p/q	–	$(sp+qr)/qs$	∞	$\top$	$\perp$
∞	–	–	∞	∞	∞
$\top$	–	–	–	$\top$	∞
$\perp$	–	–	–	–	$\perp$

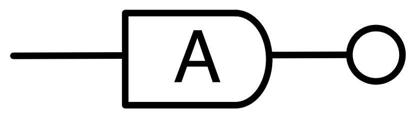
$\times$	0	r/s	∞	$\top$	$\perp$
0	0	0	$\perp$	0	$\perp$
p/q	0	pr/qs	∞	$\top$	$\perp$
∞	$\top$	∞	∞	$\top$	∞
$\top$	$\top$	$\top$	∞	$\top$	∞
$\perp$	0	$\perp$	$\perp$	0	$\perp$

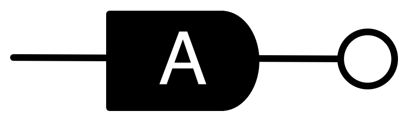


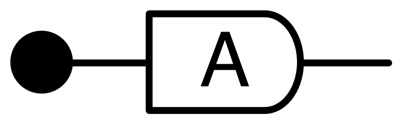
Some linear algebraic concepts in the graphical syntax

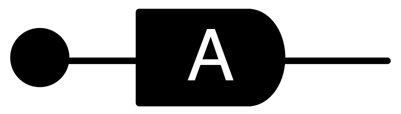
- transpose
  - combine colour and mirror image symmetries



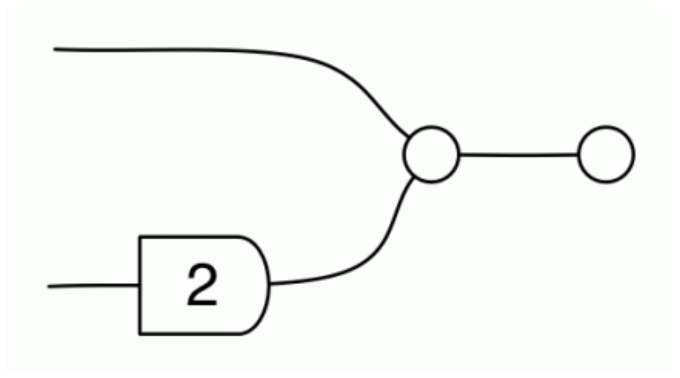
- kernel 

- cokernel 

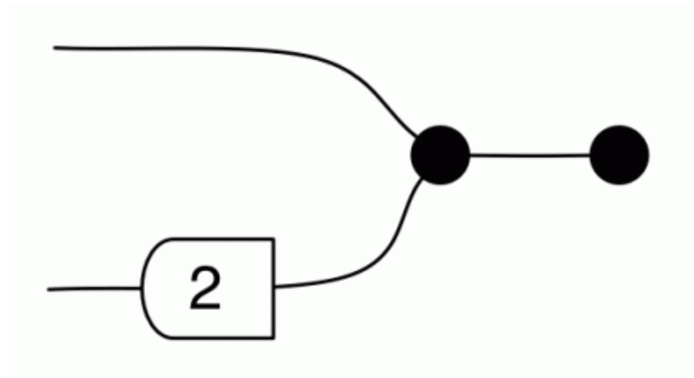
- image 

- coimage 

Fact. Given a linear subspace R:0->k in **LinRel**, its orthogonal complement R⊥ is its colour inverted diagram



$\begin{pmatrix} x \\ y \end{pmatrix} \mid x + 2y = 0$



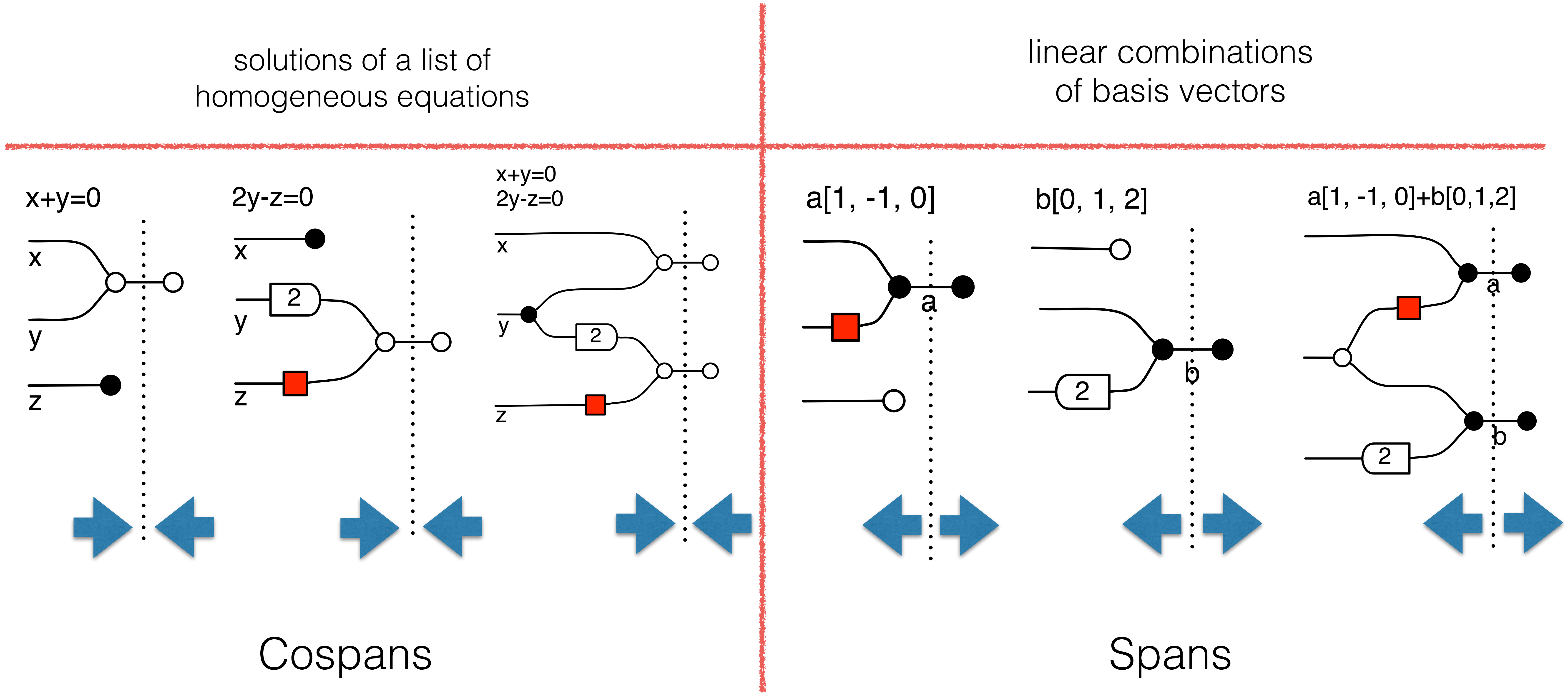
$\begin{pmatrix} x \\ 2x \end{pmatrix}$

*Corollary***.** The “fundamental theorem of linear algebra” has no mathematical content

$$\ker A = \mathrm{im}(A^T)^\perp$$
$$\ker A^T = \mathrm{im}(A)^\perp$$

Factorisations

- every diagram can be factorised as a span or cospan of matrices
- two different ways to think of linear spaces



Linear algebra with string diagrams

- the syntax exhibits the beautiful symmetries of linear algebra
- given that the theory is sound and complete, all standards results can be proved with diagrammatic reasoning
- linear algebra done righter?

- next, affine relations

Diagrammatic syntax for affine relations

$$\{ (x, \tbinom{x}{x}) \}$$

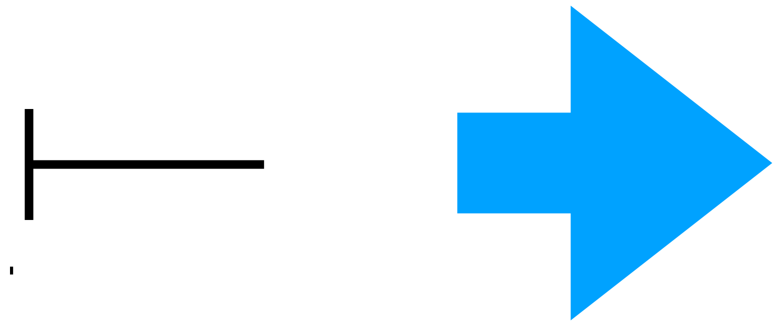
$$\{ (x, \bullet) \}$$

$$\{ (\tbinom{x}{y}, x+y) \}$$

$$\{ (\bullet, 0) \}$$

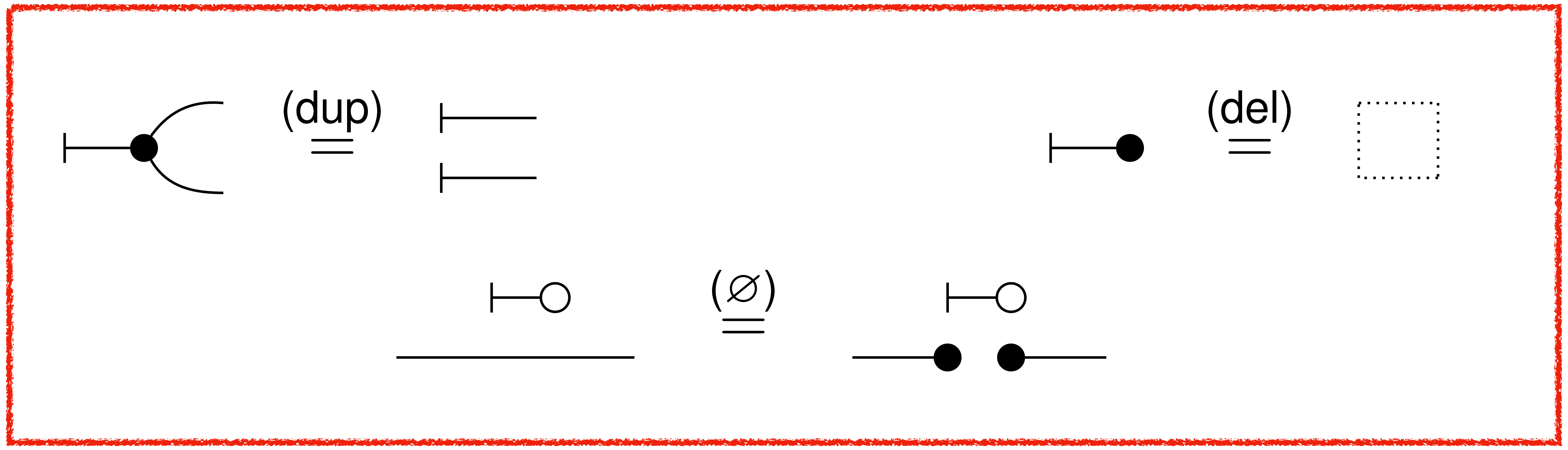
$$\{ (x, kx) \}$$

For affine relations, we only need one new generator!



$$\{ (\bullet, 1) \}$$

Equational characterisation



Together with the equations of IH, this is the relational theory of affine relations. Moreover:

IHA $\cong$ AffRel$_{Q}$

Case study

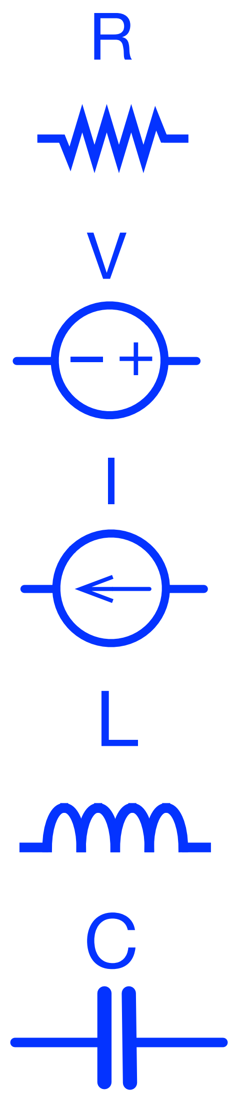
Non-passive electrical circuits

- work with the diagrammatic language for $\mathsf{AffRel}_{\mathbf{R}[x]}$
- introduce a syntactic prop of electrical circuits
- develop diagrammatic reasoning techniques
  - the impedance calculus
- prove classical “theorems” of electrical circuit theory

The prop of electrical circuits

- **ECirc**, free on the following signature

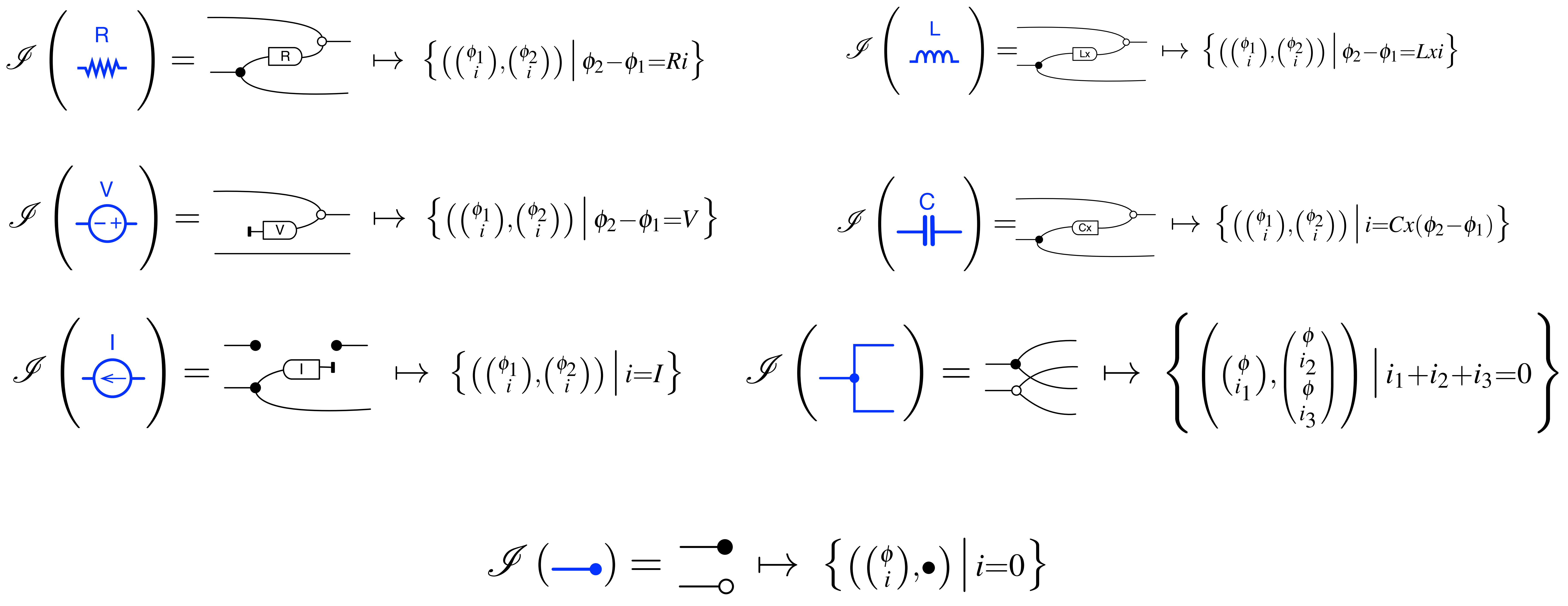
- resistor
- voltage source
- current source
- inductor
- capacitor



Compositional semantics

$$\mathscr{I} : \mathsf{ECirc} \to \mathsf{GAA}$$

$$\mathscr{I}(1) = 2$$



$$\mapsto \left\{\left(\begin{pmatrix}\phi_1\\ i\end{pmatrix},\begin{pmatrix}\phi_2\\ i\end{pmatrix}\right) \,\middle|\, \phi_2-\phi_1=Ri\right\}$$

$$\mapsto \left\{\left(\begin{pmatrix}\phi_1\\ i\end{pmatrix},\begin{pmatrix}\phi_2\\ i\end{pmatrix}\right) \,\middle|\, \phi_2-\phi_1=Lxi\right\}$$

$$\mapsto \left\{\left(\begin{pmatrix}\phi_1\\ i\end{pmatrix},\begin{pmatrix}\phi_2\\ i\end{pmatrix}\right) \,\middle|\, \phi_2-\phi_1=V\right\}$$

$$\mapsto \left\{\left(\begin{pmatrix}\phi_1\\ i\end{pmatrix},\begin{pmatrix}\phi_2\\ i\end{pmatrix}\right) \,\middle|\, i=Cx(\phi_2-\phi_1)\right\}$$

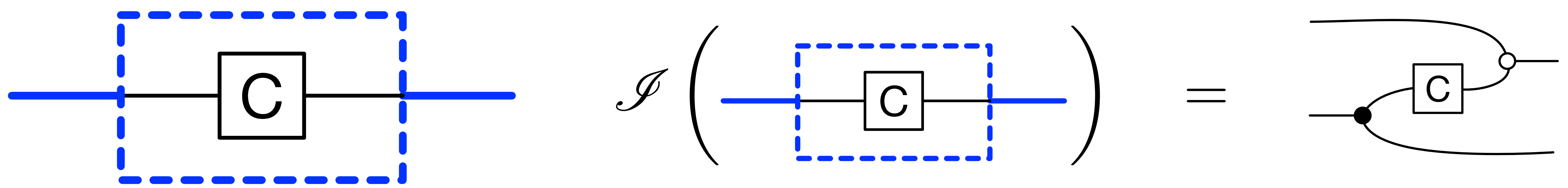
$$\mapsto \left\{\left(\begin{pmatrix}\phi_1\\ i\end{pmatrix},\begin{pmatrix}\phi_2\\ i\end{pmatrix}\right) \,\middle|\, i=I\right\}$$

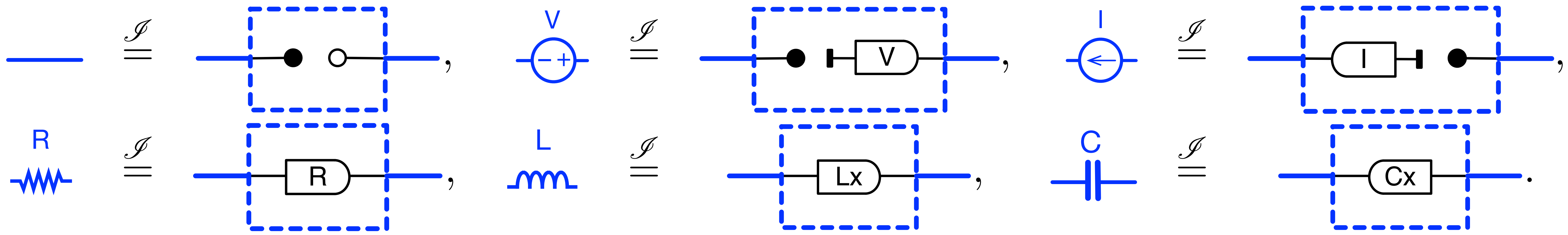
$$\mapsto \left\{\left(\begin{pmatrix}\phi\\ i_1\end{pmatrix},\begin{pmatrix}\phi\\ i_2\\ \phi\\ i_3\end{pmatrix}\right) \,\middle|\, i_1+i_2+i_3=0\right\}$$

$$\mapsto \left\{\left(\begin{pmatrix}\phi\\ i\end{pmatrix},\bullet\right) \,\middle|\, i=0\right\}$$

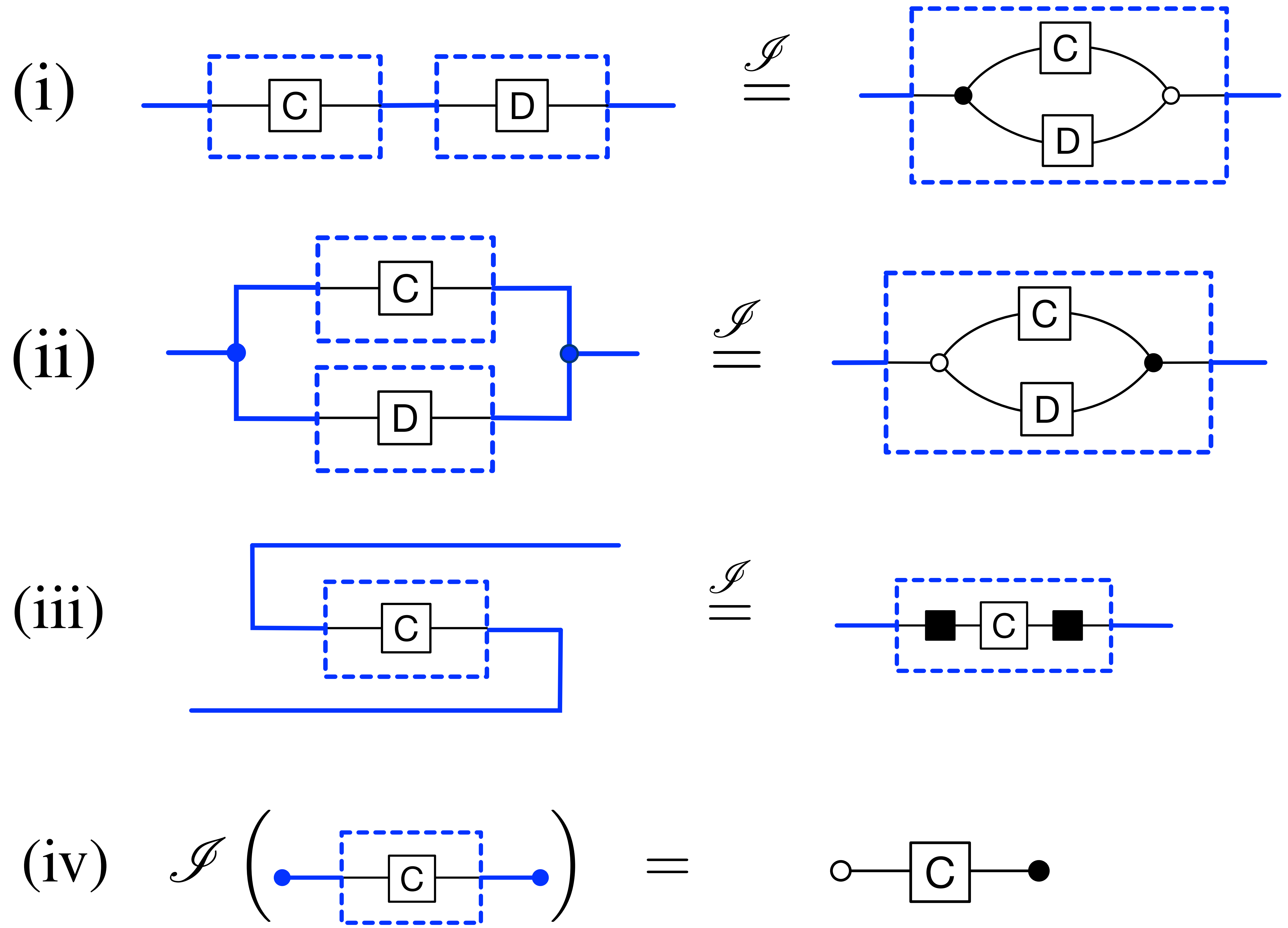
Impedance calculus

- Extend the signature of **ECirc** with impedance boxes

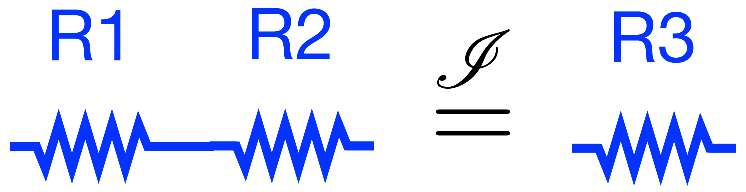


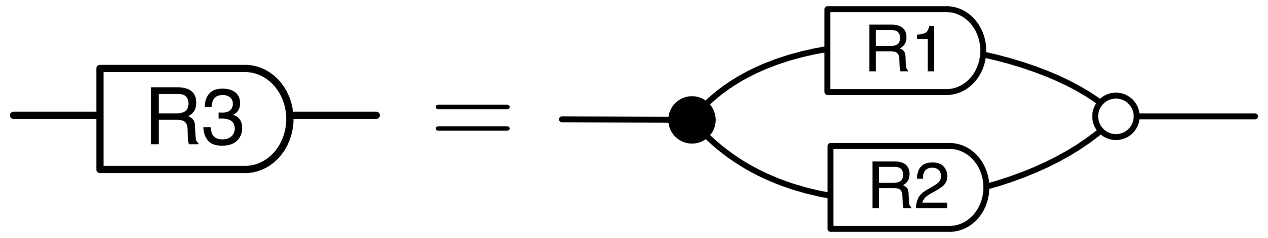


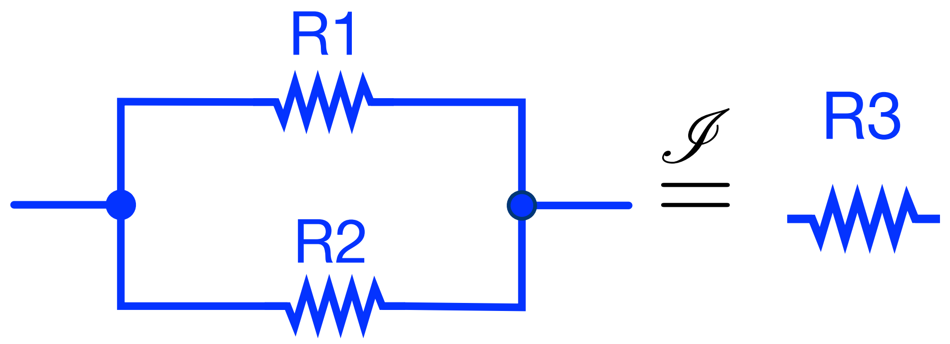
Impedance box lemma

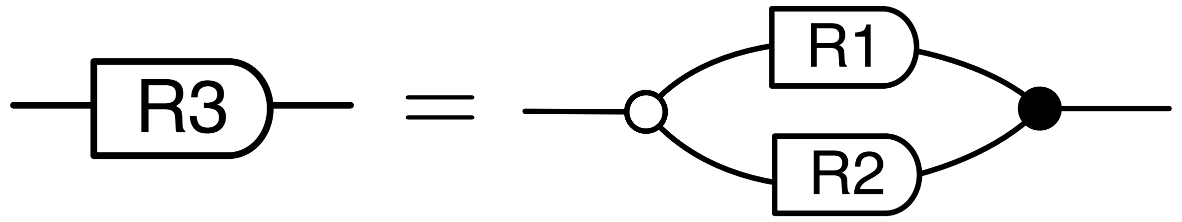


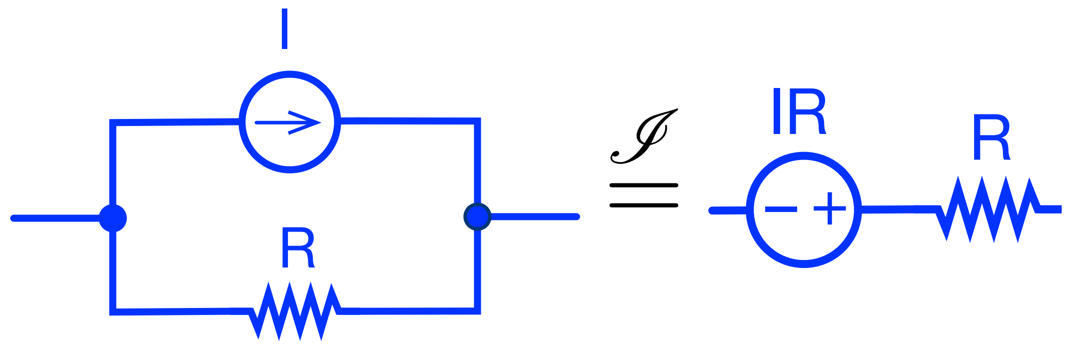
Proposition

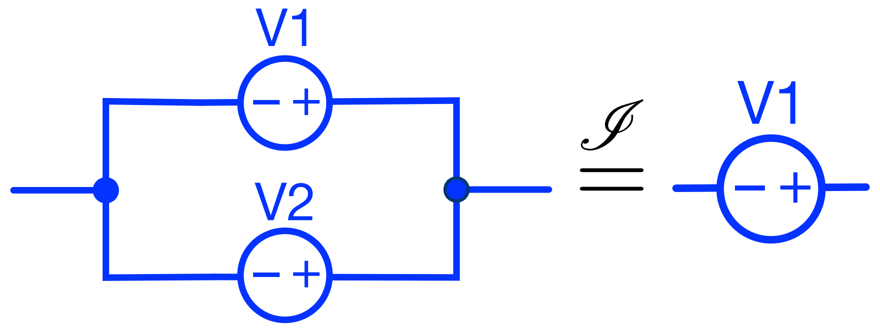
(i) 

where 

(ii) 

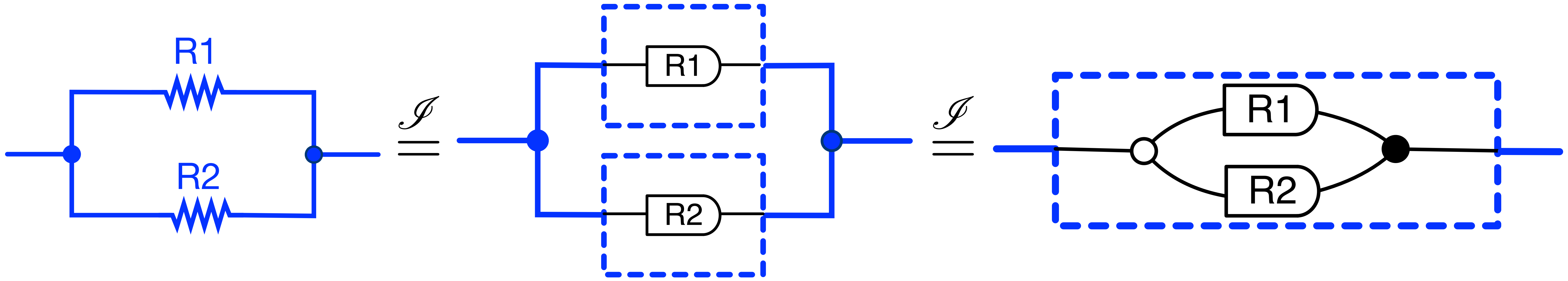
where 

(iii) 

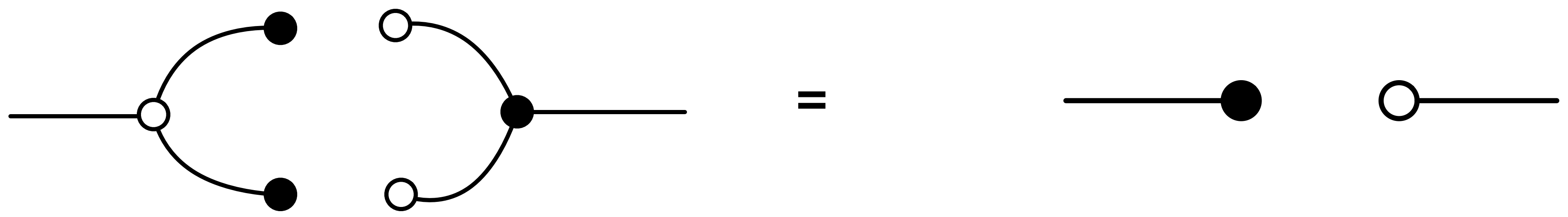
(iv) 

if $V1 = V2$, otherwise its semantics is $\emptyset$ (the empty relation)

Proof of (ii)



What if R1=R2=0?



Textbook formulas fail here because of “division by zero”

Some classical theorems

- Relativity of potentials
- Conservation of current
- Independent measurement theorem
- Superposition theorem
- Thévenin’s theorem
  - see Guillaume Boisseau’s thesis!

Conclusions

- String diagrams can carry algebraic data that characterises applications that are relevant in the 21st century
    - partial functions
    - non-classical (e.g. Quantum data)
    - relational structures
- The functorial semantics methodology scales (partial theories, relational theories, first order theories)
- Compositional reasoning with string diagrams and functorial semantics is a powerful tool
    - other examples: Petri nets, signal flow graphs (with different semantics), Bayesian networks, automata, …
- Reasoning with string diagrams fixes the deficiencies of traditional syntax and exposes errors, implicit assumptions, and conceptual inadequacies