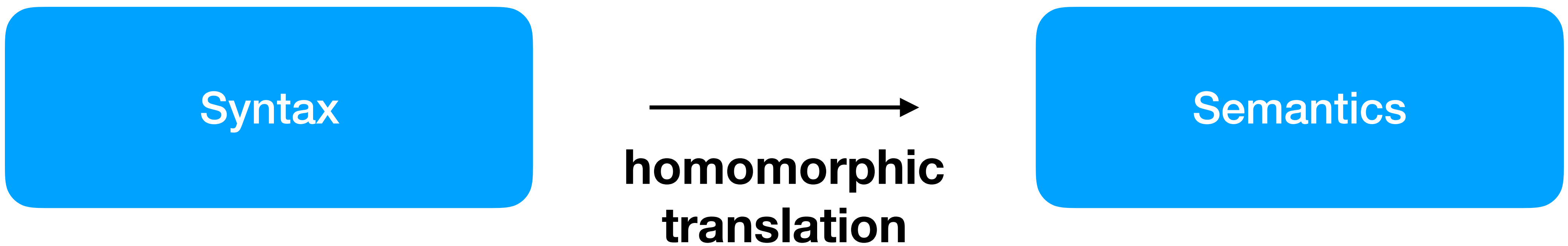
String diagrams

from control to concurrency and beyond

Pawel Sobocinski
Tallinn University of Technology
IFIP WG 2.2 Vienna 24/09/19

Joint work with Filippo Bonchi, Fabio Zanasi and Robin Piedeleu

Compositionality

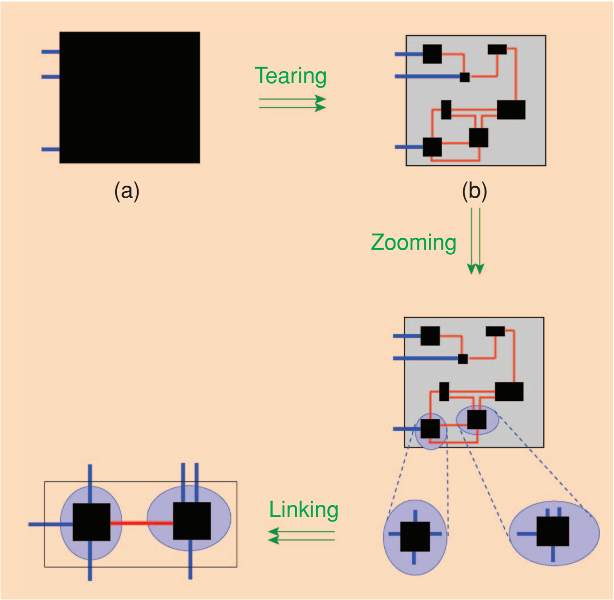


- for "nice" homomorphic translation
  - syntactic operations correspond to natural operations on the semantic domain
  - syntax expressive enough to capture enough of the semantic domain
  - natural notions of semantic equivalence find an axiomatisation in the syntax

Our approach

- in computer science, the tradition is to start with some syntax and study formal semantics as a separate subject
- we think that it is useful to **reverse** the process
    - start with the the algebra of the semantic domain (in CS, control, engineering, science, mathematics, …)
    - engineer an appropriate syntax to support that algebra

Behavioural control theory

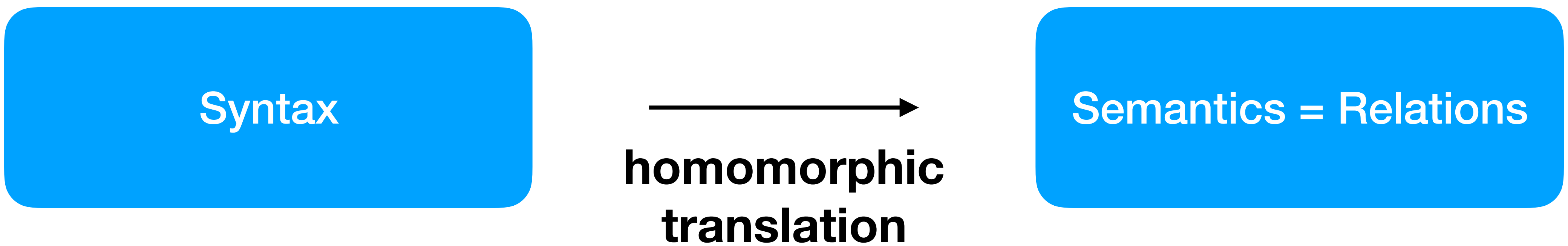


“Thinking of a dynamical system as a behavior, and of **inter-connection as variable sharing**, gets the physics right.”

- Willems’ thesis: abandon causality and functionality (paraphrasing mine)
    - causal thinking is a disease of the brain (Russell, 1912)
    - laws of physics are seldom functional
    - functional modelling is seldom compositional
    - Willems’ tearing procedure produces relational, not functional, behaviours

J. C. Willems, *The behavioural approach to open and interconnected systems: modeling by tearing, zooming, and linking*, IEEE Control Systems Magazine, 2007.

Compositionality



- What kind of algebra?
  - first order logic, regular logic, relational algebra, datalog, allegories, …
- What kind of relations?
  - vanilla, additive, linear, affine, …

$\mathbf{Rel}_\times$

- For Willems' intuitions, an appropriate universe seems to be the categorical algebra of the symmetric monoidal category $\mathbf{Rel}_\times$
  - objects: sets X, Y, Z, ….
  - arrows: (typed) relations, $R: X \to Y$, $S: Y \to Z$
  - composition: relational composition

    $$R \,;\, S = \{\, (x,z) \mid \exists y.\ xRy \wedge ySz \}$$

  - monoidal product: $R{\times}R'$: $X{\times}X' \to Y{\times}Y'$

    $$R \times R' = \{\, ((x,x'),(y,y')) \mid xRy \wedge x'R'y' \,\}$$

String diagrams

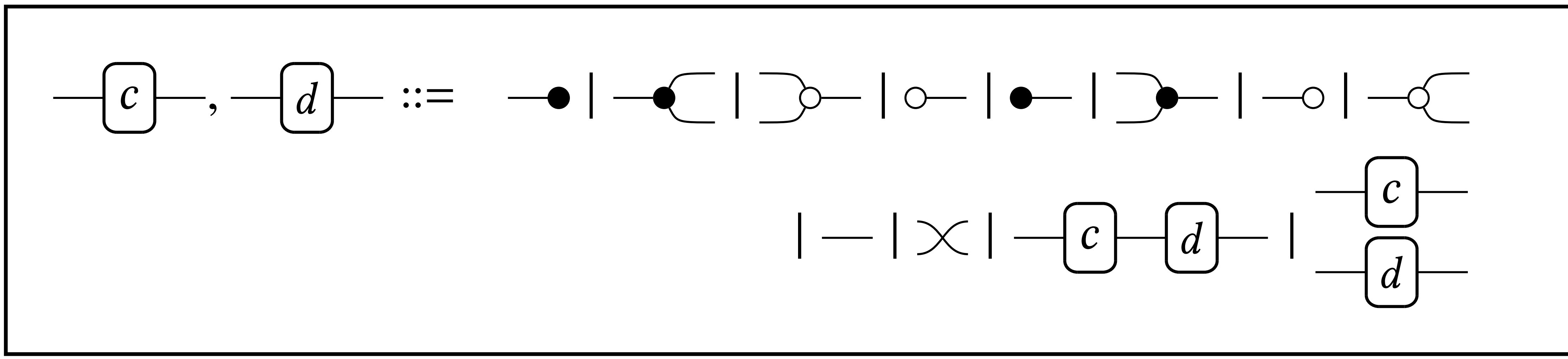
- diagrammatic syntax for symmetric monoidal categories
- diagrammatic reasoning: the laws of symmetric monoidal categories are baked in to the diagrams

Compositionality

- syntax expressive enough?
- axiomatisations?

Graphical Linear Algebra

- String diagrams generated by the following syntax



String diagrams $\longrightarrow$ **LinRel$_{\mathbf{Q}}$**

monoidal functor

Sound and fully complete axiomatisation - the theory of **IH** (Interacting Hopf algebras)

(Bonchi, S., Zanasi, Interacting Hopf Algebras, 2014)

Signal flow graphs

- The **IH** construction is parametric wrt any PID
- Starting with **R**[x] we get linear relations over its field of fractions **R**(x)
- This is yields a sound and complete equational system for reasoning about signal flow graphs: models of computation that compute solutions of rational functions

F. Bonchi, P. Sobociński and F. Zanasi, "Full Abstraction for Signal Flow Graphs", *In Principles of Programming Languages, POPL`15*

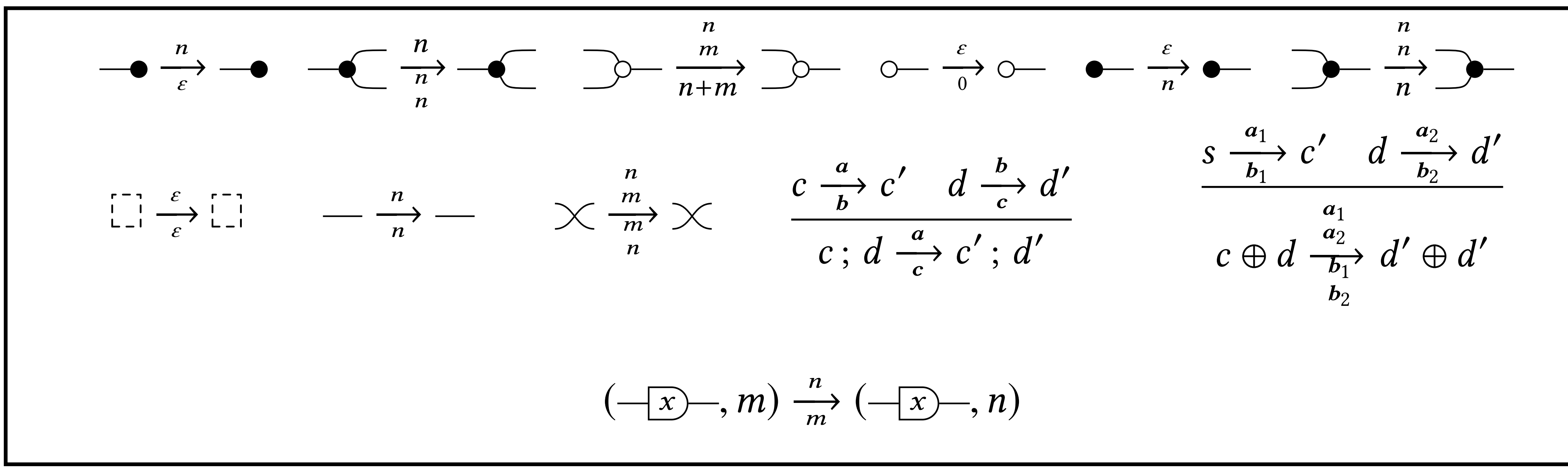
F. Bonchi, P. Sobociński and F. Zanasi, "The Calculus of Signal Flow Diagrams I: Linear Relations on Streams", Inf Comput

B. Fong, P. Rapisarda and P. Sobociński, "A categorical approach to open and interconnected dynamical systems", *LICS `16*

F. Bonchi, J. Holland, D. Pavlovic and P. Sobociński, "Refinement for signal flow graphs", C*ONCUR `17*

The operational view

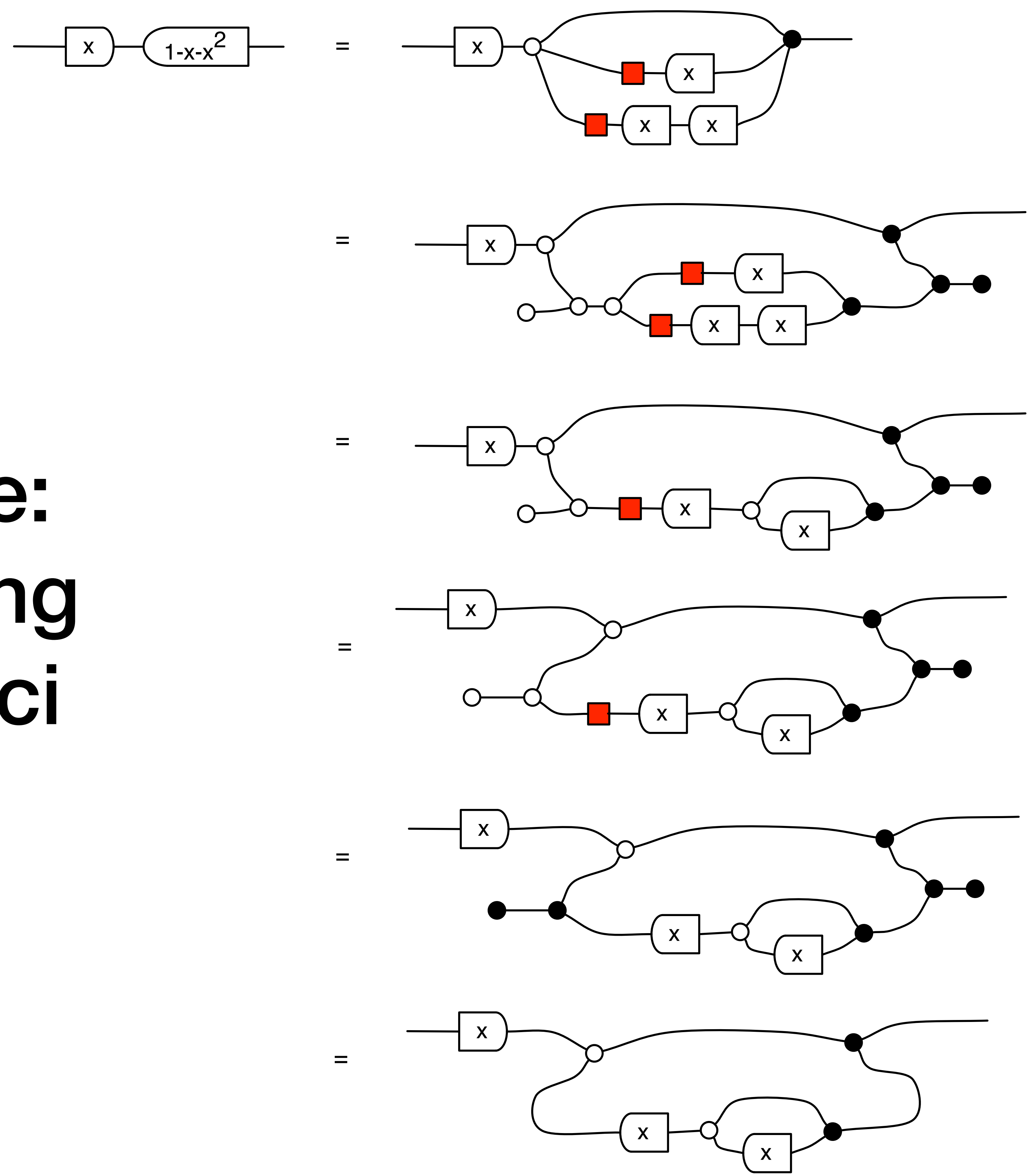
- The work on signal flow graphs emphasises the importance of the **operational view**



- For signal flow graphs, the signals come from a field, typically **R** or **Q**

Bonchi, Piedeleu, Sobocinski and Zanasi. Bialgebraic Semantics for String Diagrams. CONCUR 2019

Example: computing Fibonacci



Graphical Diophantine Algebra

- **Definition.** An additive relation of type k->l is a subset $R \subseteq \mathbf{N}^k \times \mathbf{N}^l$ s.t. $(0,0) \in \mathbf{R}$ and, if (a,b), $(a',b') \in \mathbf{R}$ then $(a+a',b+b') \in \mathbf{R}$
- An additive relation is f.g. if we can find a finite basis: i.e. every element can be expressed as a sum of basis elements
- These form a prop **AddRel** as a subprop of $\mathbf{Rel}_\times$
  - proving f.g. additive relations are closed under composition is a cute application of Dickson's Lemma

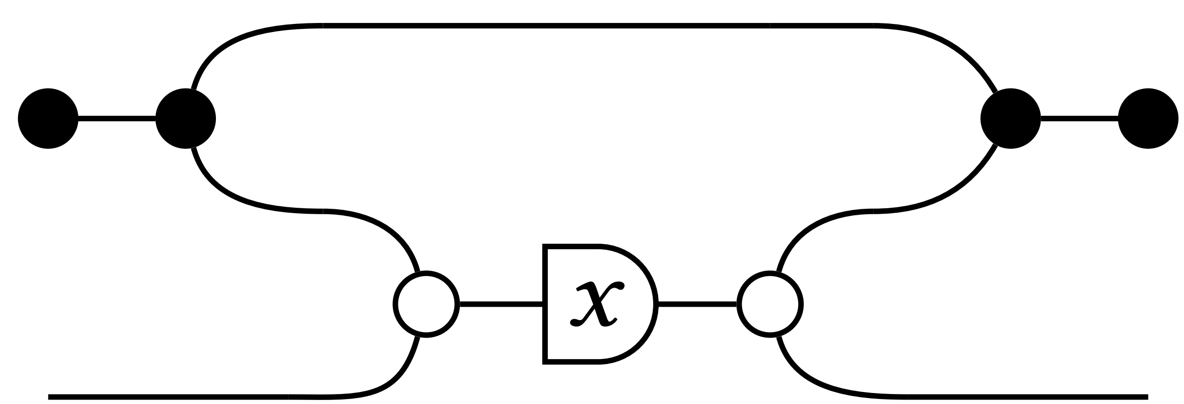
String diagrams $\xrightarrow{\textbf{monoidal functor}}$ f.g. additive relations

Same syntax as before, and…. sound and fully complete axiomatisation

Bonchi, Holland, Piedeleu, S, Zanasi. Diagrammatic algebra: from Linear to Concurrent Systems. PoPL 2019

From control to concurrency

- For linear relations, adding state yielded a compositional account of signal flow graphs
- For additive relations, adding state yields a compositional account of Petri nets



Graphical Affine Algebra

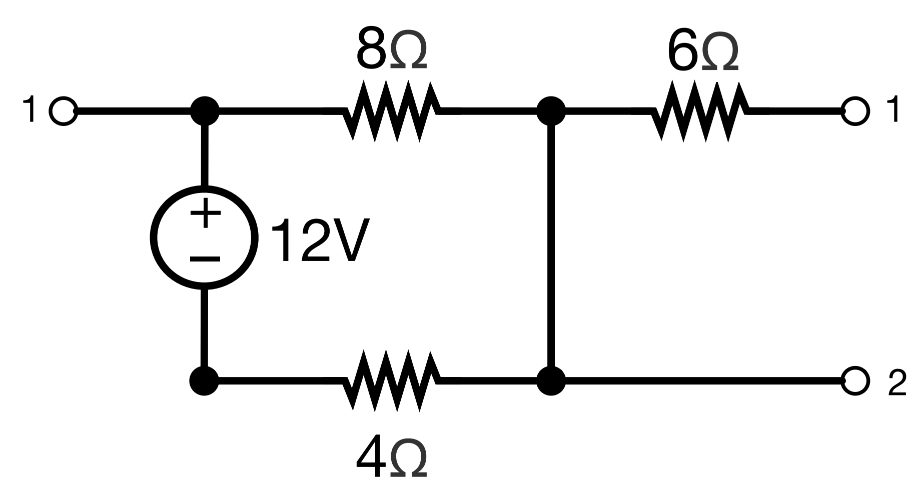
- The usual syntax extended with $\vdash\!\!-\!\!-$ that "outputs 1"

Two sound and complete axiomatisations.

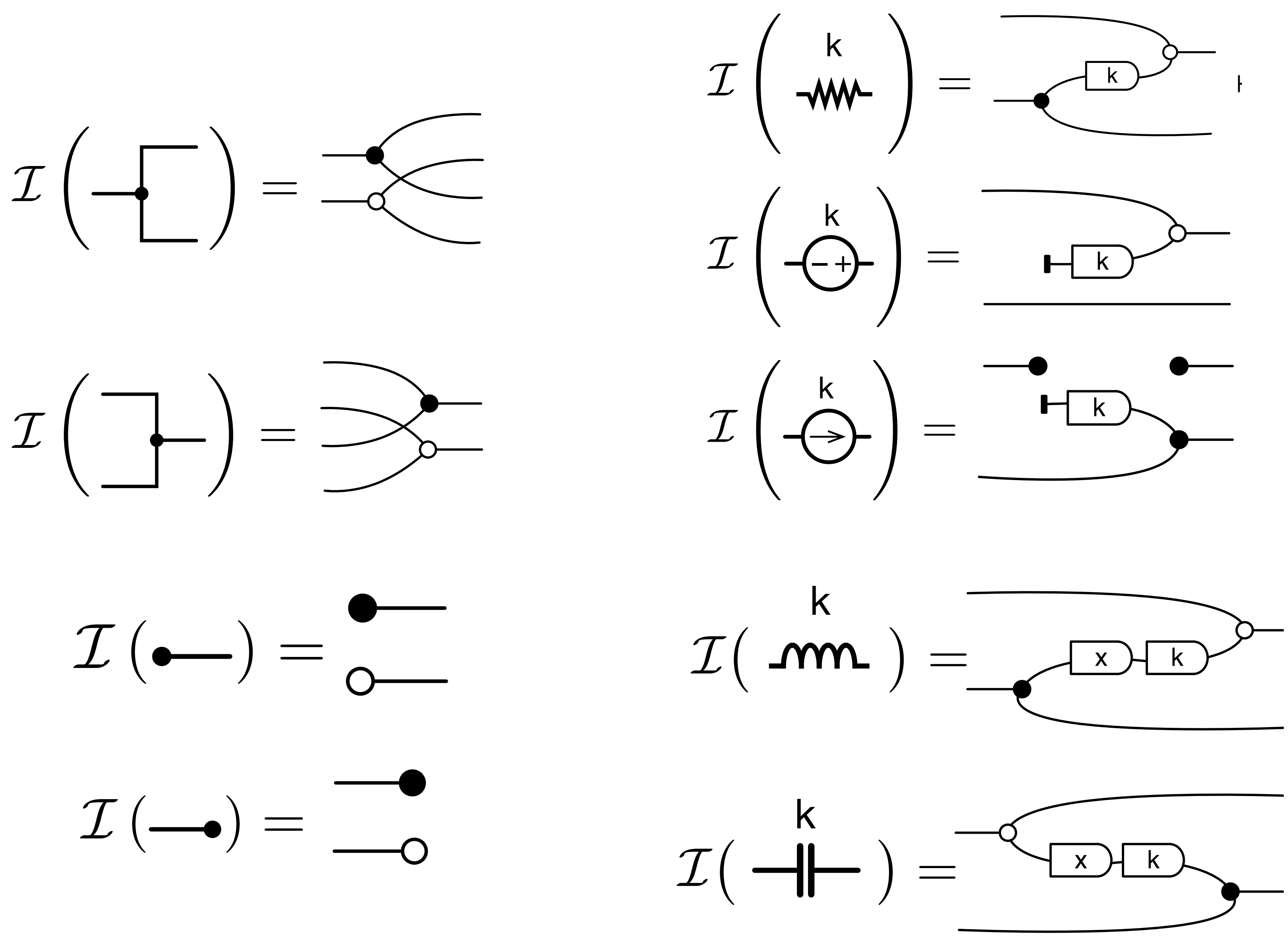
Bonchi, Piedeleu, Sobocinski, Zanasi. Graphical Affine Algebra. LiCS 2019

Fun application: electrical circuits

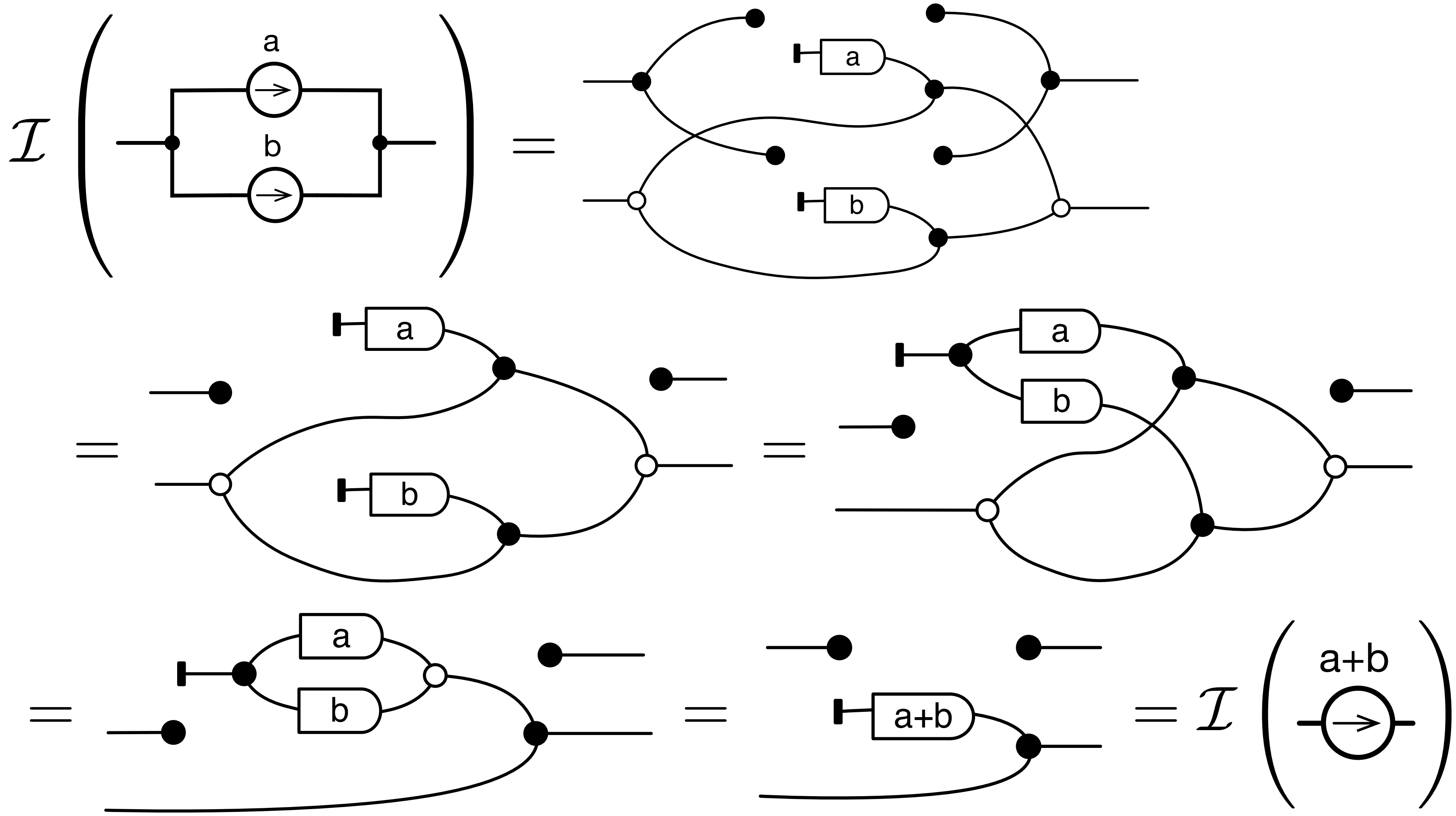
- Let's go back to the **R** world. We will use Graphical Affine Algebra as a sound and complete diagrammatic proof system for open circuits like:



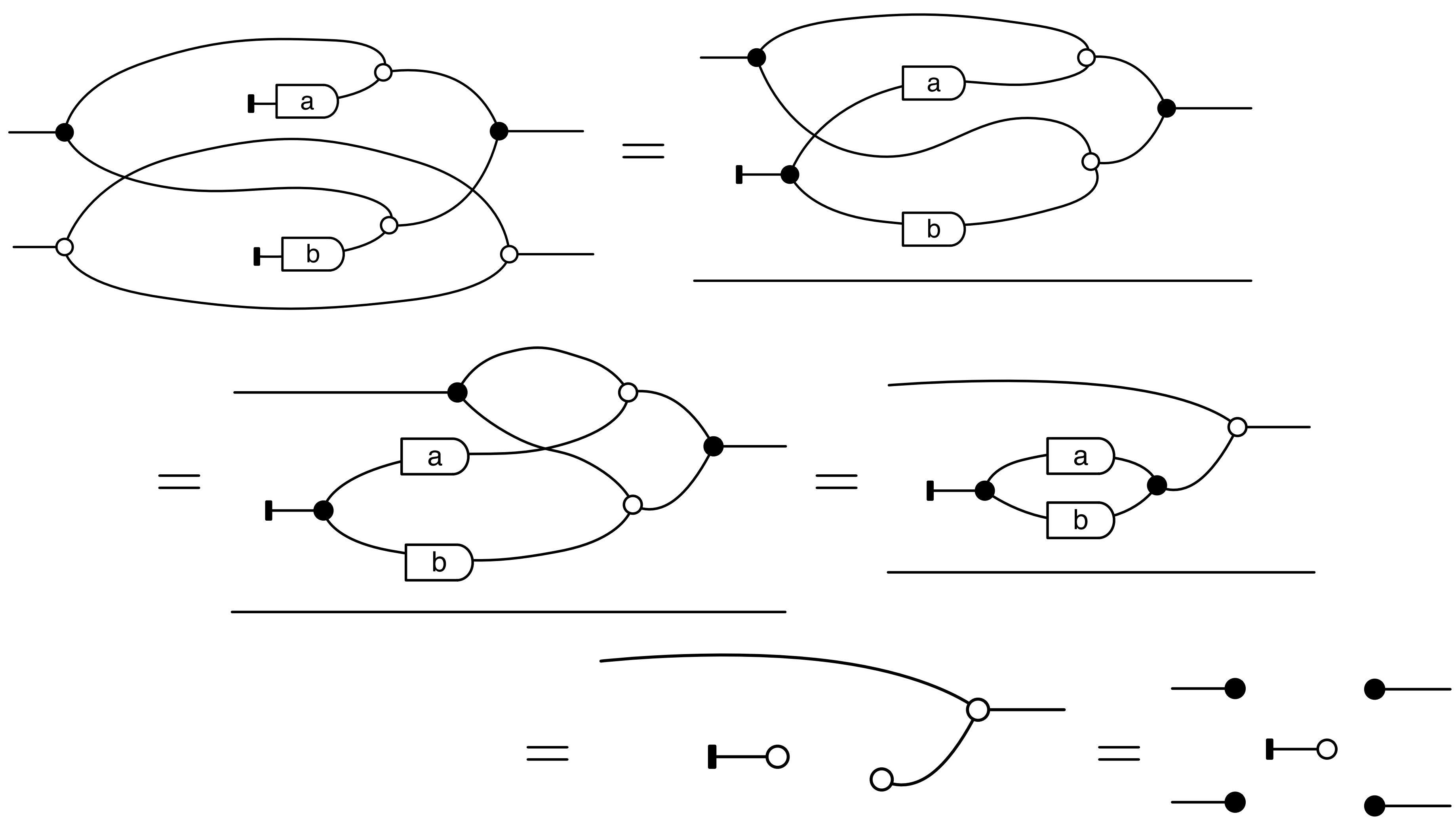
Compiling to string diagrams



Current sources in parallel are additive



Voltage sources in parallel are “illegal”



- inspired by process algebra - operational playing around leads to equations leads to denotations
- unlike process algebra, we are not reinventing the algebraic wheel: the basic operations for composing process are those of monoidal categories
- what most surprises me is **robustness**.
  - on the semantic side, the mathematics changes drastically
  - equationally, in terms of the string diagrams, we change some basic interaction of GLA primitives

Fong, Rapisarda and Sobocinski, "A categorical approach to open and interconnected dynamical systems", *LICS `16*

Compositional systems and methods

- new compositionality group at Taltech: applications of category theory to concurrency, control, game theory, engineering, machine learning, …
- come and visit!!
- SYCO 7 in Tallinn - March 30-31, 2020 - save the date!