The Algebra of Signal Flow Graphs

Filippo Bonchi, Paweł Sobociński, Fabio Zanasi

ENS DE LYON

Journées Structures Discrètes 2015

Interacting Hopf Algebras

We show

- an algebraic theory of matrices over a PID k (Hopf Algebras);
- an algebraic theory of subspaces over the field of fractions of k (Interacting Hopf Algebras).

Interacting Hopf Algebras

We show

- an algebraic theory of matrices over a PID k (Hopf Algebras);
- an algebraic theory of subspaces over the field of fractions of k (Interacting Hopf Algebras).

Interacting Hopf Algebras provide a (graphical) syntax and a sound and complete axiomatization for subspaces.

For instance, we can express both *systems of equations* and *bases* as term of our syntax; we can check that they denote the same subspace via the axiomatization.

Interacting Hopf Algebras

We show

- an algebraic theory of matrices over a PID k (Hopf Algebras);
- an algebraic theory of subspaces over the field of fractions of k (Interacting Hopf Algebras).

In this talk, we fix the PID to be the ring of polynomials $\mathsf{k}[x]$.

The terms of the corresponding syntax are well-known structures called *signal flow graphs*.

Interacting Hopf Algebras

If you are interested in, you can

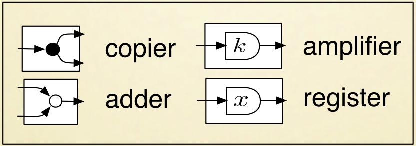
- have a look to the Ph.D thesis of Fabio Zanasi (ENS-Lyon),
- follow Pawel's blog http://graphicallinearalgebra.net,
- knock to my door.

In this talk, we fix the PID to be the ring of polynomials $\mathsf{k}[x]$.

The terms of the corresponding syntax are well-known structures called *signal flow graphs*.

Signal Flow Graphs

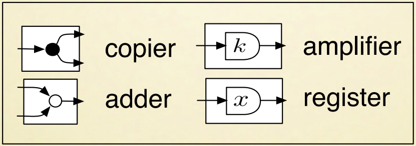
- Signal Flow Graphs (SFGs) are **stream processing circuits** widely adopted in Control Theory since at least the 1950s.
- Constructed combining four kinds of gate



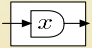
$k \in \mathsf{k}$

Signal Flow Graphs

- Signal Flow Graphs (SFGs) are **stream processing circuits** widely adopted in Control Theory since at least the 1950s.
- Constructed combining four kinds of gate

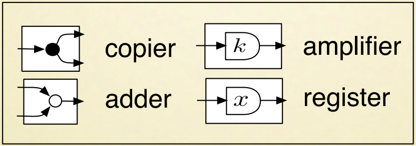


$k \in \mathsf{k}$

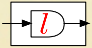


Signal Flow Graphs

- Signal Flow Graphs (SFGs) are **stream processing circuits** widely adopted in Control Theory since at least the 1950s.
- Constructed combining four kinds of gate

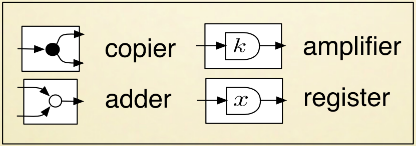


$k \in \mathsf{k}$



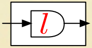
Signal Flow Graphs

- Signal Flow Graphs (SFGs) are **stream processing circuits** widely adopted in Control Theory since at least the 1950s.
- Constructed combining four kinds of gate



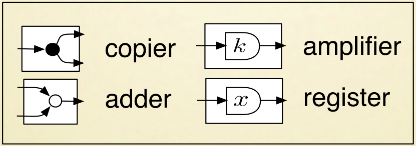
$k \in \mathsf{k}$

k

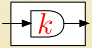


Signal Flow Graphs

- Signal Flow Graphs (SFGs) are **stream processing circuits** widely adopted in Control Theory since at least the 1950s.
- Constructed combining four kinds of gate



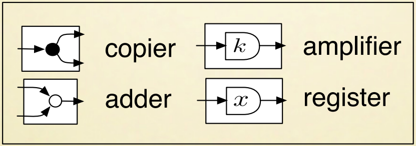
$k \in \mathsf{k}$



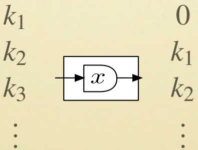
l

Signal Flow Graphs

- Signal Flow Graphs (SFGs) are **stream processing circuits** widely adopted in Control Theory since at least the 1950s.
- Constructed combining four kinds of gate

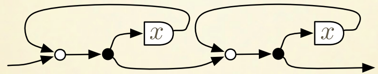


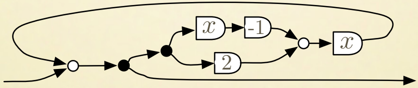
$k \in \mathsf{k}$



Signal Flow Graphs

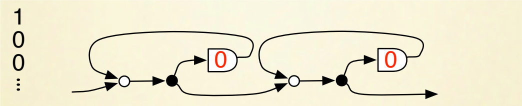
Two examples:

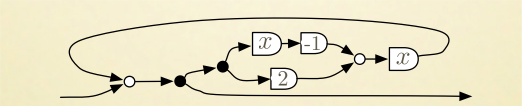




Signal Flow Graphs

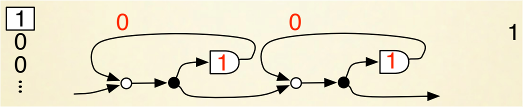
Two examples:

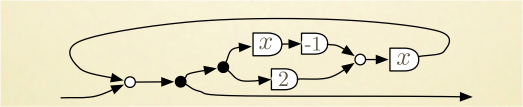




Signal Flow Graphs

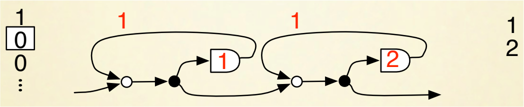
Two examples:

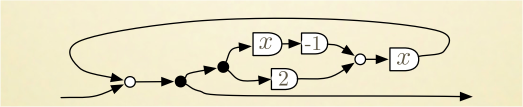




Signal Flow Graphs

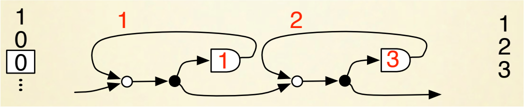
Two examples:

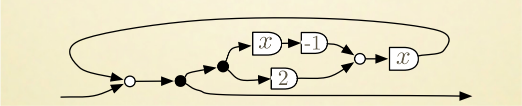




Signal Flow Graphs

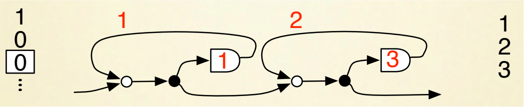
Two examples:

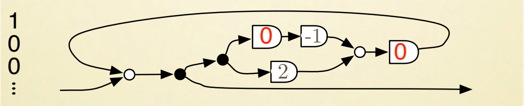




Signal Flow Graphs

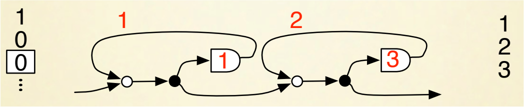
Two examples:

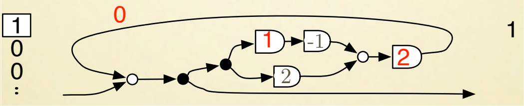




Signal Flow Graphs

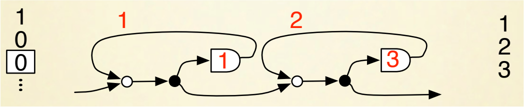
Two examples:

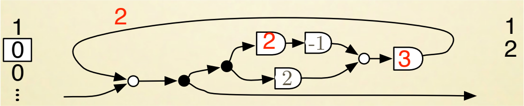




Signal Flow Graphs

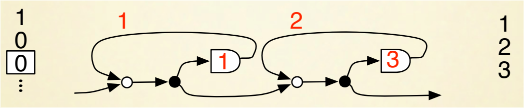
Two examples:

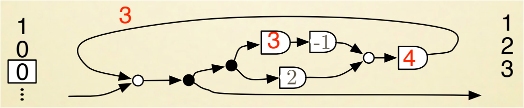




Signal Flow Graphs

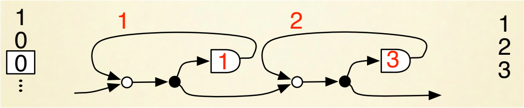
Two examples:

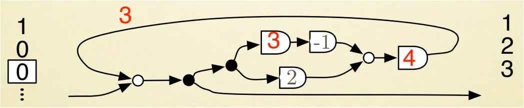




Signal Flow Graphs

Two examples:



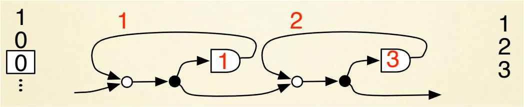


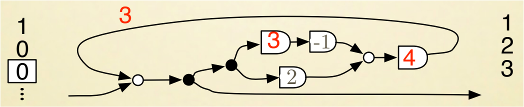
Both circuits implement the generating function

$$\frac{1}{(1-x)^2} = 1x + 2x^2 + 3x^3 + \ldots$$

Signal Flow Graphs

Two examples:





Both circuits implement the generating function

$$\frac{1}{(1-x)^2} = 1x + 2x^2 + 3x^3 + \ldots$$

Can we check this *statically*?

Signal Flow Graphs

- In traditional approaches, SFGs are not treated as interesting mathematical structures per se.
  - $\Rightarrow$ formal analysis typically mean translation into systems of linear equations.
- We study SFGs *directly* as graphical structures.

In this work

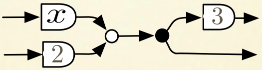
A graphical theory of Signal Flow Graphs

- String diagrammatic syntax for circuits.
- **Compositional** semantics.
- **Sound and complete axiomatisation** for semantic equivalence.
  - $\Rightarrow$ Two circuits implement the same specification if they can be transformed one into the other using the equational theory.

Outline

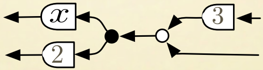
- Functional circuits

  ⇒ the signal flows from left to right



- Reverse functional circuits

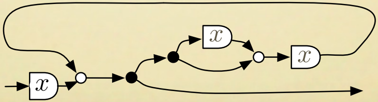
  ⇒ the signal flows from right to left



- Generalised circuits

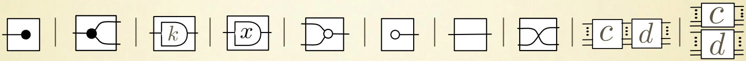
  ⇒ the signal can flow in both directions

  ⇒ environment for modeling signal flow graphs

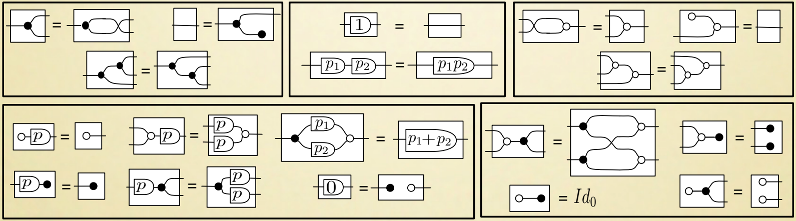


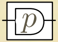
The theory $\mathbb{HA}$ of functional circuits

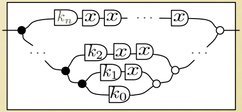
Functional circuits are the string diagrams generated by the grammar

$c, d ::=$ 

subject to the following equations:



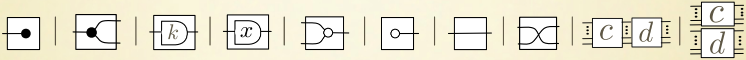
where, for a polynomial $p = k_0 + k_1 x + \cdots + k_n x^n$, 

is 

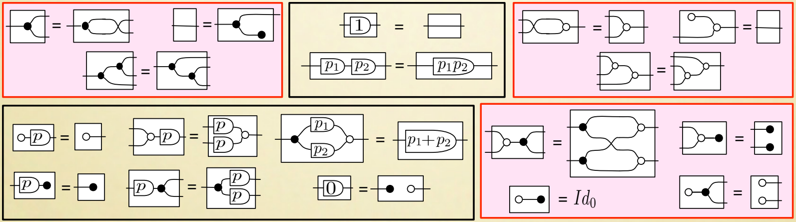
.

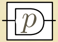
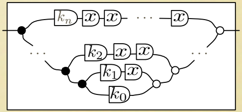
The theory $\mathbb{HA}$ of functional circuits

Functional circuits are the string diagrams generated by the grammar

$c, d ::=$ 

subject to the following equations:

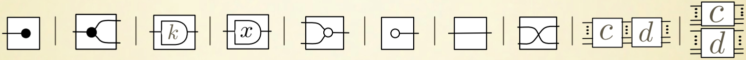


where, for a polynomial $p = k_0 + k_1 x + \cdots + k_n x^n$,  is 

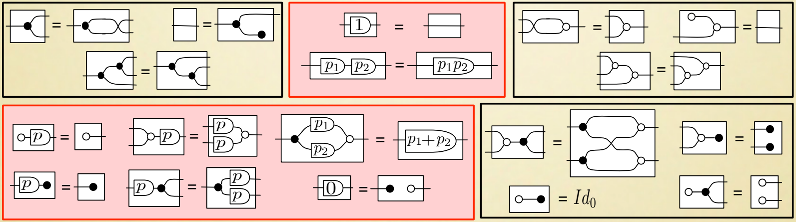
.

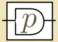
The theory $\mathbb{HA}$ of functional circuits

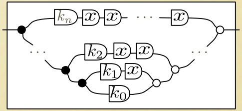
Functional circuits are the string diagrams generated by the grammar

$c, d ::=$ 

subject to the following equations:



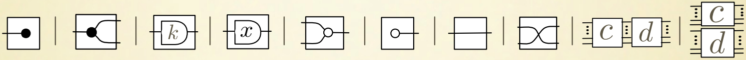
where, for a polynomial $p=k_0+k_1x+\cdots+k_nx^n$, 

is 

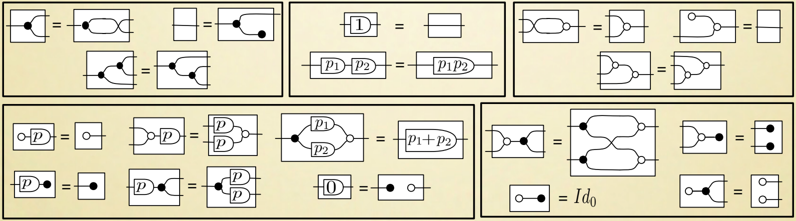
.

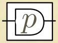
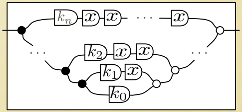
The theory $\mathbb{HA}$ of functional circuits

Functional circuits are the string diagrams generated by the grammar

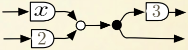
$c, d ::=$ 

subject to the following equations:

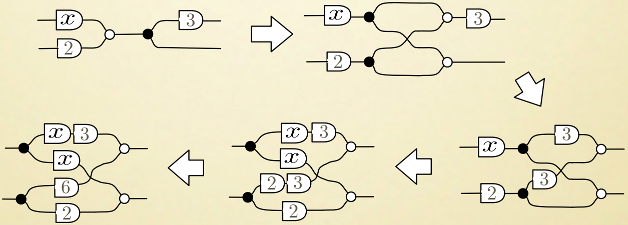


where, for a polynomial $p = k_0 + k_1 x + \cdots + k_n x^n$,  is  .

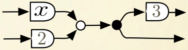
Semantics of functional circuits

- Functional circuits modulo the equations are in 1-1 correspondence with matrices over the polynomial ring $\mathsf{k}[x]$.
- Example: check the semantics of 

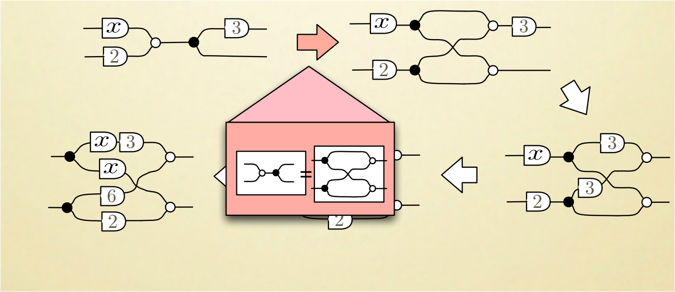
using the equational theory $\mathbb{HA}$.



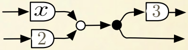
Semantics of functional circuits

- Functional circuits modulo the equations are in 1-1 correspondence with matrices over the polynomial ring $\mathsf{k}[x]$.
- Example: check the semantics of 

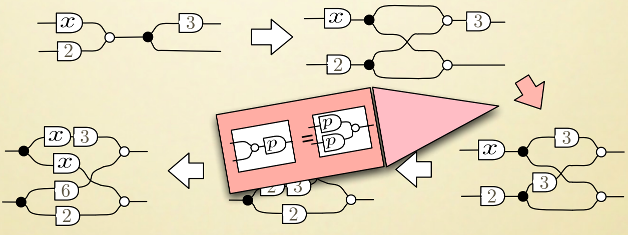
using the equational theory $\mathbb{HA}$.



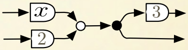
Semantics of functional circuits

- Functional circuits modulo the equations are in 1-1 correspondence with matrices over the polynomial ring $\mathsf{k}[x]$.
- Example: check the semantics of 

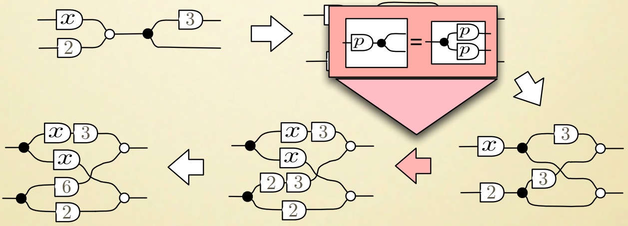
using the equational theory $\mathbb{HA}$.



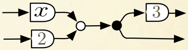
Semantics of functional circuits

- Functional circuits modulo the equations are in 1-1 correspondence with matrices over the polynomial ring $\mathsf{k}[x]$.
- Example: check the semantics of 

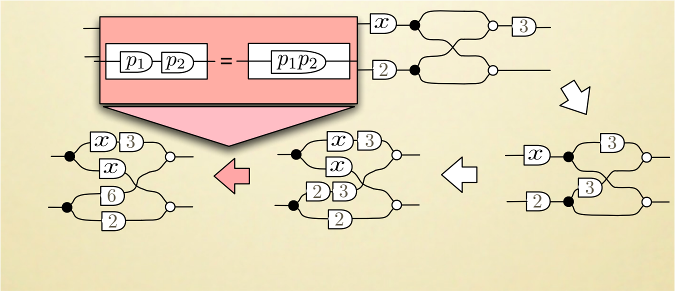
using the equational theory $\mathbb{HA}$.



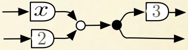
Semantics of functional circuits

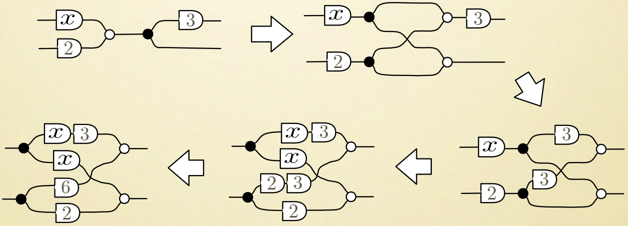
- Functional circuits modulo the equations are in 1-1 correspondence with matrices over the polynomial ring $\mathsf{k}[x]$.
- Example: check the semantics of 

using the equational theory $\mathbb{HA}$.

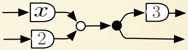


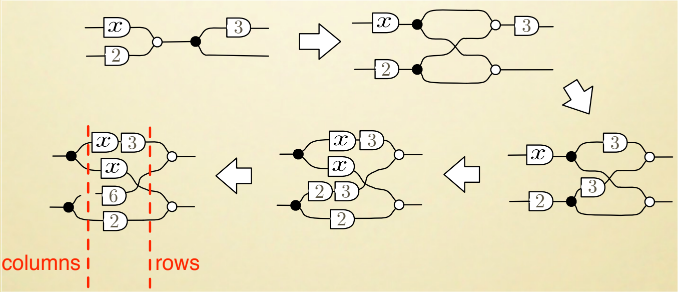
Semantics of functional circuits

- Functional circuits modulo the equations are in 1-1 correspondence with matrices over the polynomial ring $\mathsf{k}[x]$.
- Example: check the semantics of  using the equational theory $\mathbb{HA}$.



Semantics of functional circuits

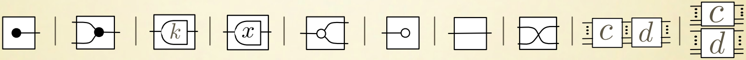
- Functional circuits modulo the equations are in 1-1 correspondence with matrices over the polynomial ring $\mathsf{k}[x]$.
- Example: check the semantics of  using the equational theory $\mathbb{HA}$.



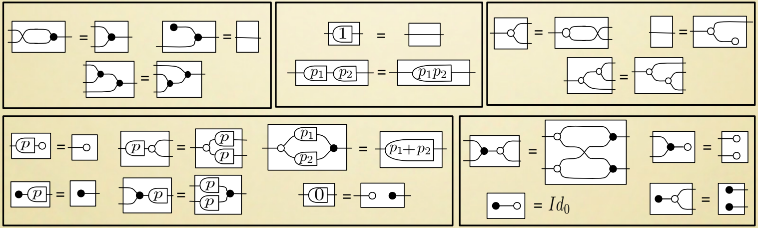
Its semantics is the matrix $\begin{pmatrix} 3x & 6 \\ x & 2 \end{pmatrix}$.

Reverse functional circuits

Reverse functional circuits are functional circuits “reflected about the *y*-axis”. They are the diagrams generated by the grammar

$c, d ::=$ 

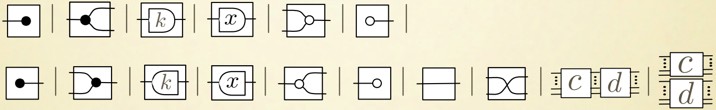
subject to equations dual to those of $\mathbb{HA}$:



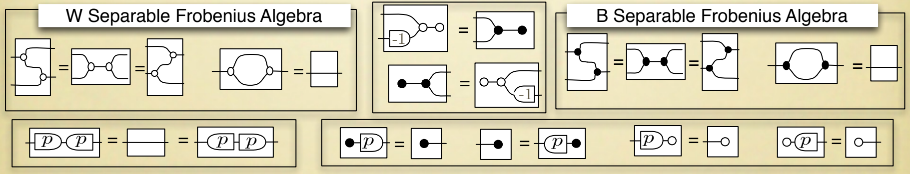
The theory $\mathbb{IH}$ of generalised circuits

Generalised circuits are string diagrams generated by the grammar

$c, d ::=$

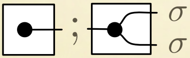


subject to the equations of the theories of functional and reverse functional circuits, plus the following:



Semantics of Generalised Circuits

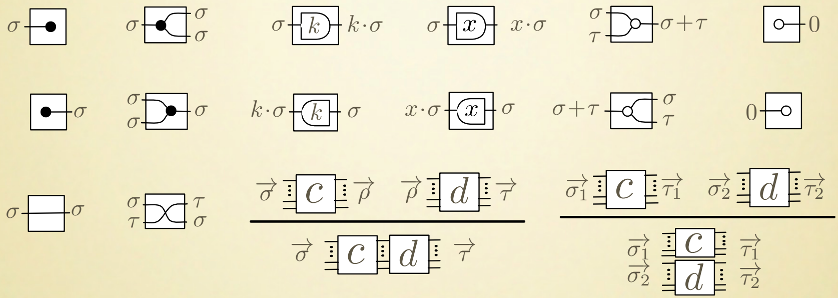
Circuits do not generally have a univocal flow direction — a *relational* model is required.

For instance, 

expresses the diagonal relation.

Semantics of Generalised Circuits

The semantics $[\![\cdot]\!]$ maps a circuit into a linear relation (subspace):

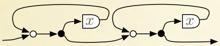
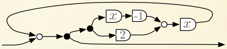


The axiomatisation of $\mathbb{IH}$ is sound and complete

$$[\![c]\!] = [\![d]\!] \quad \Leftrightarrow \quad c \overset{\mathbb{IH}}{=} d$$

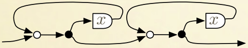
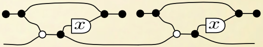
The key technical step in the proof consists in reducing a circuit in its Hermite Normal Form

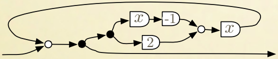
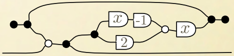
Graphical reasoning in $\mathbb{IH}$

Check:  and  implement $\frac{1}{(1-x)^2}$.

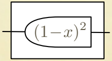
Proof strategy:

- Represent the two SFGs as generalised circuits

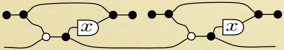
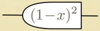
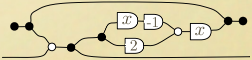
 $\rightsquigarrow$ 

 $\rightsquigarrow$ 

- Represent the specification as a generalised circuit:

σ  $\sigma \cdot \dfrac{1}{(1-x)^2}$

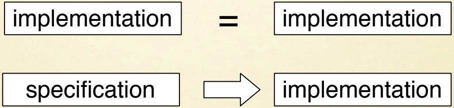
- Prove the three of them equal using the axioms of $\mathbb{IH}$:

 =  = 

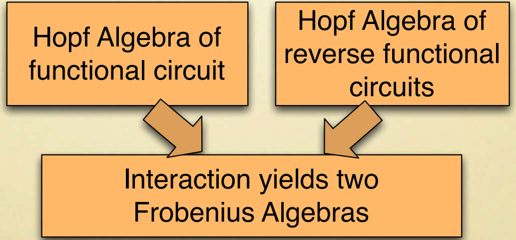
Conclusions

We proposed an algebraic environment for signal flow graphs

- compositional semantics in terms of linear relations
- sound and complete axiomatisation
  - graphical proof system



  - rich mathematical playground



Future Work

What are the fundamental structures of concurrency?
We still don't know! - Samson Abramsky 2014

- functional computations have a paradigmatic model: λ-calculus;
- concurrent computations do not: there are many different models like Petri nets, Process Calculi, Event Structures ...

A path toward an answer...

$$\frac{\text{Systems of linear difference equations} \qquad \mathbb{IH}}{\text{Diophantine Systems of linear difference equations} \qquad ?}$$