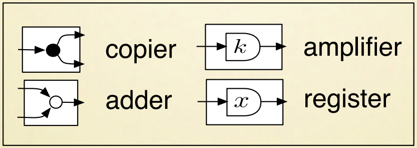
Full Abstraction for Signal Flow Graphs

Filippo Bonchi, Paweł Sobociński, Fabio Zanasi

ENS DE LYON

Signal Flow Graphs

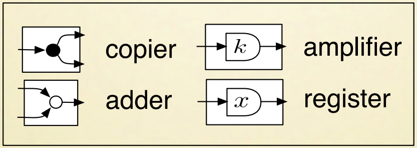
- Signal Flow Graphs are **stream processing circuits** studied in Control Theory since the 1950s.
- Constructed combining four kinds of gate



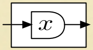
$k \in \mathsf{k}$

Signal Flow Graphs

- Signal Flow Graphs are **stream processing circuits** studied in Control Theory since the 1950s.
- Constructed combining four kinds of gate

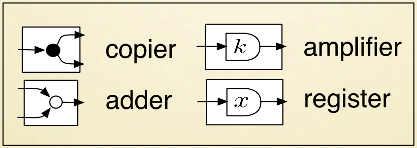


$k \in \mathsf{k}$

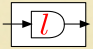


Signal Flow Graphs

- Signal Flow Graphs are **stream processing circuits** studied in Control Theory since the 1950s.
- Constructed combining four kinds of gate

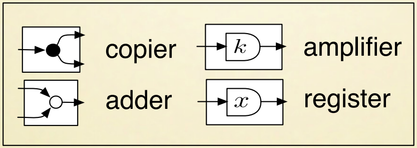


$k \in \mathsf{k}$

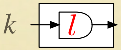


Signal Flow Graphs

- Signal Flow Graphs are **stream processing circuits** studied in Control Theory since the 1950s.
- Constructed combining four kinds of gate

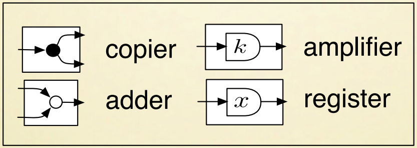


$k \in \mathsf{k}$

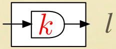


Signal Flow Graphs

- Signal Flow Graphs are **stream processing circuits** studied in Control Theory since the 1950s.
- Constructed combining four kinds of gate

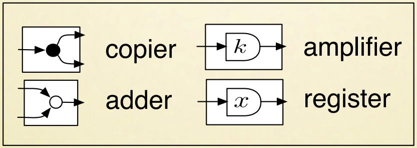


$k \in \mathsf{k}$

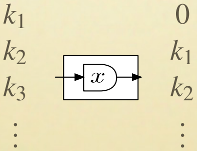


Signal Flow Graphs

- Signal Flow Graphs are **stream processing circuits** studied in Control Theory since the 1950s.
- Constructed combining four kinds of gate

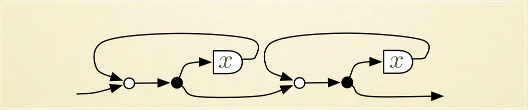


$k \in \mathsf{k}$



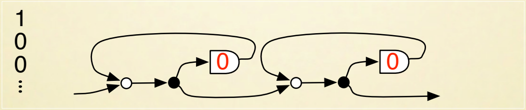
Signal Flow Graphs

An example:



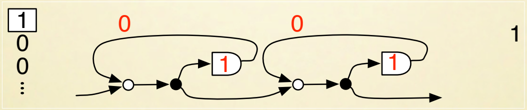
Signal Flow Graphs

An example:



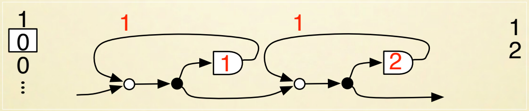
Signal Flow Graphs

An example:



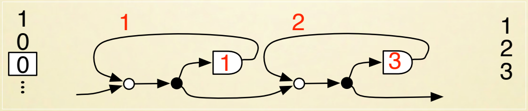
Signal Flow Graphs

An example:



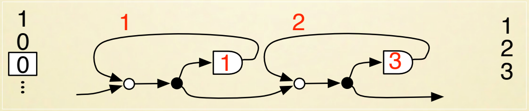
Signal Flow Graphs

An example:



Signal Flow Graphs

An example:



Input 1000... produces 1234....

Signal Flow Graphs

The orthodoxy

- SFGs are not treated as interesting mathematical objects per se.
- Formal analysis typically mean translation into a “lower-level” formalism like systems of linear equations.

Signal Flow Graphs

The orthodoxy

- SFGs are not treated as interesting mathematical objects per se.
- Formal analysis typically mean translation into a “lower-level” formalism like systems of linear equations.

In this work

- An high-level formalism where SFGs are first-class objects:

the calculus of signal flow diagrams

Signal Flow Graphs

The orthodoxy

- SFGs are not treated as interesting mathematical objects per se.
- Formal analysis typically mean translation into a “lower-level” formalism like systems of linear equations.

In this work

- An high-level formalism where SFGs are first-class objects:

  the calculus of signal flow diagrams

  - String diagrammatic (=graphical) syntax
  - Structural Operational Semantics
  - Denotational semantics
  - Sound and complete axiomatisation
  - Full Abstraction
  - Realisability

Flow direction is FUNDAMENTAL

“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

Flow direction is EVIL

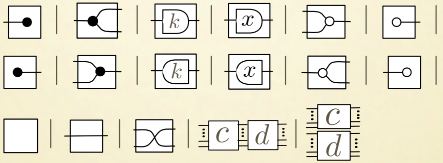
"Adding a signal flow direction is often a figment of one's imagination, and when something is not real, it will turn out to be cumbersome sooner or later. [...] The input/output framework is totally inappropriate for dealing with all but the most special system interconnections. [The input/output representation] often needlessly complicates matters, mathematically and conceptually. A good theory of systems takes the *behavior* as the basic notion."

J. Willems. Linear systems in discrete time. 2009

The Calculus of SF Diagrams

Circuit diagrams of Circ are generated by the grammar

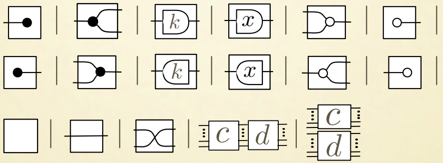
$c, d ::=$



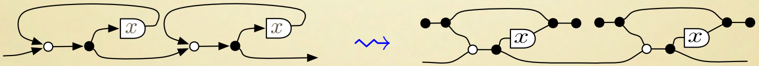
The Calculus of SF Diagrams

Circuit diagrams of Circ are generated by the grammar

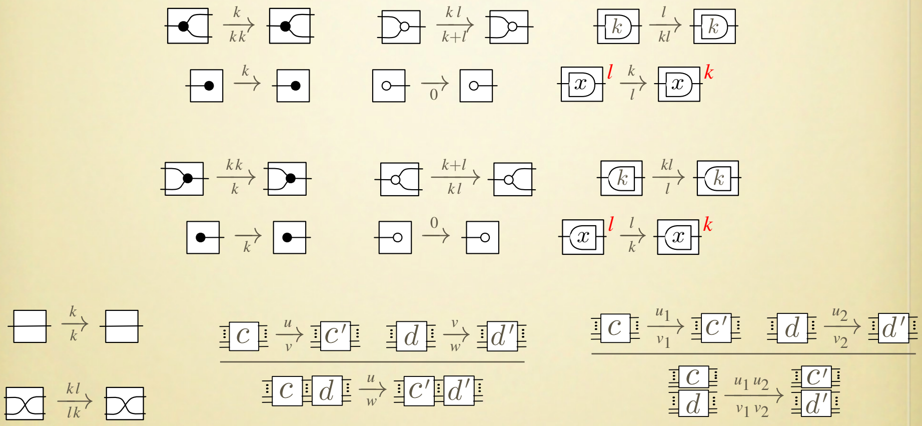
$c, d ::=$



We can represent (orthodox) signal flow graphs as circuit diagrams:



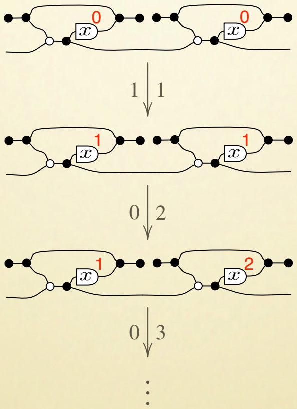
Structural Operational Semantics



$\xrightarrow[kk]{k}$
$\xrightarrow[k+l]{kl}$
k
$\xrightarrow[kl]{l}$
$\xrightarrow{k}$
$\xrightarrow[0]{}$
x l $\xrightarrow[l]{k}$ x k
$\xrightarrow[k]{kk}$
$\xrightarrow[kl]{k+l}$
$\xrightarrow[l]{kl}$
$\xrightarrow[k]{}$
$\xrightarrow{0}$
x l $\xrightarrow[k]{l}$ x k
$\xrightarrow[k]{k}$
$\xrightarrow[lk]{kl}$
$c \xrightarrow[v]{u} c'$ $d \xrightarrow[w]{v} d'$
$c\,d \xrightarrow[w]{u} c'\,d'$
$c \xrightarrow[v_1]{u_1} c'$ $d \xrightarrow[v_2]{u_2} d'$
$\frac{c}{d} \xrightarrow[v_1 v_2]{u_1 u_2} \frac{c'}{d'}$

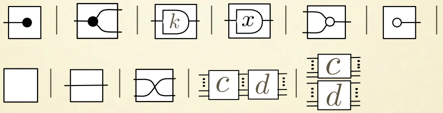
The operational semantics $\langle c \rangle$ is the set of all traces starting from an initial state for c (i.e. one where all the registers are labeled with 0).

Example



Functional Circuits

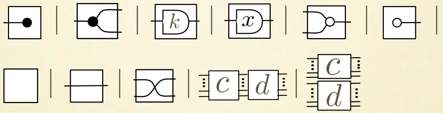
Circuit diagrams of $\overrightarrow{\mathsf{Circ}}$ are generated by the grammar

$c, d ::=$ 

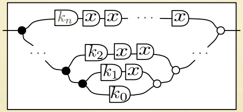
Functional Circuits

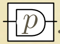
Circuit diagrams of $\overrightarrow{\mathsf{Circ}}$ are generated by the grammar

$c, d ::=$



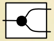
We can represent a polynomial $p = k_0 + k_1 x + \cdots + k_n x^n$ as



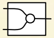
hereafter denoted by .

Polynomial matrices

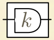
The map $\overrightarrow{[\![\cdot]\!]} \colon \overrightarrow{\mathsf{Circ}} \to \mathsf{Mat}\,\mathsf{k}[x]$ is inductively defined as follows:

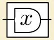
 $\longmapsto \begin{pmatrix} 1 \\ 1 \end{pmatrix}$

$\longmapsto\ !$

 $\longmapsto \begin{pmatrix} 1 & 1 \end{pmatrix}$

$\longmapsto$ ¡

 $\longmapsto \begin{pmatrix} k \end{pmatrix}$

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

$\longmapsto id_0$

$\longmapsto id_1$

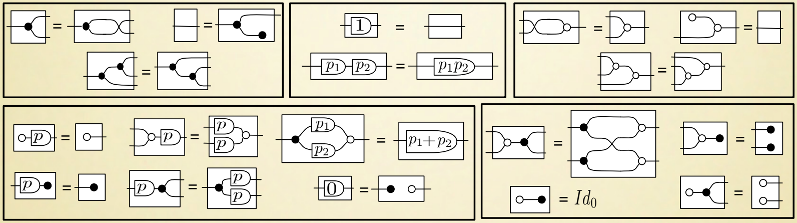
$\longmapsto \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$

$$c_1 \oplus c_2 \longmapsto \overrightarrow{[\![c_1]\!]} \oplus \overrightarrow{[\![c_2]\!]} \qquad c_1 \,;c_2 \longmapsto \overrightarrow{[\![c_1]\!]} \,;\overrightarrow{[\![c_2]\!]}$$

where $! \colon 0 \to 1$ and ¡ $\colon 1 \to 0$ are given by initiality and finality of 0 in $\mathsf{Mat}\,\mathsf{k}[x]$.

Axiomatization

The theory $\mathbb{HA}$ is $\overrightarrow{\mathsf{Circ}}$ quotiented by the following equations:



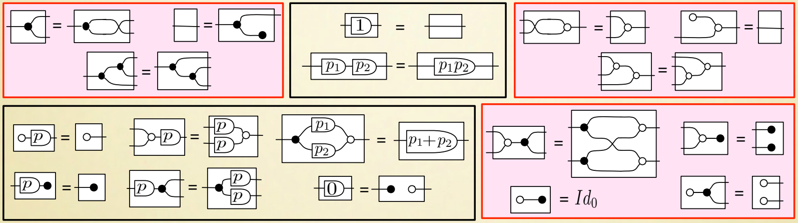
Soundness and Completeness

$$\overrightarrow{[\![c]\!]} = \overrightarrow{[\![d]\!]} \iff c \overset{\mathbb{HA}}{=} d$$

$\mathbb{HA}$ is isomorphic to $\mathsf{Mat}\,\mathsf{k}[x]$

Axiomatization

The theory $\mathbb{HA}$ is $\overrightarrow{\mathsf{Circ}}$ quotiented by the following equations:



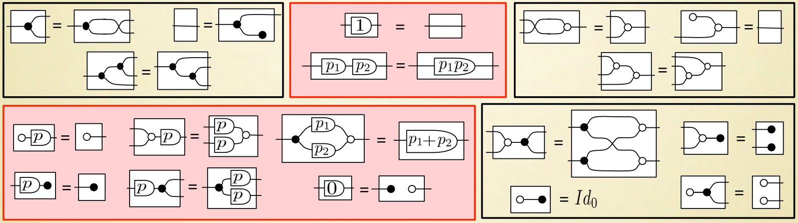
Soundness and Completeness

$$\overrightarrow{[\![c]\!]} = \overrightarrow{[\![d]\!]} \iff c \overset{\mathbb{HA}}{=} d$$

$\mathbb{HA}$ is isomorphic to $\mathsf{Mat}\,\mathsf{k}[x]$

Axiomatization

The theory $\mathbb{HA}$ is $\overrightarrow{\mathsf{Circ}}$ quotiented by the following equations:



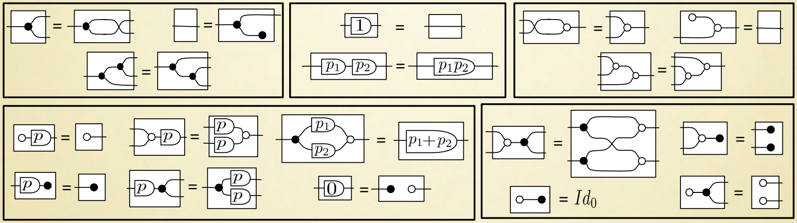
Soundness and Completeness

$$\overrightarrow{[\![c]\!]} = \overrightarrow{[\![d]\!]} \iff c \overset{\mathbb{HA}}{=} d$$

$\mathbb{HA}$ is isomorphic to $\mathsf{Mat}\,\mathsf{k}[x]$

Axiomatization

The theory $\mathbb{HA}$ is $\overrightarrow{\mathsf{Circ}}$ quotiented by the following equations:



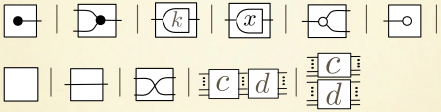
Soundness and Completeness

$$\overrightarrow{[\![c]\!]} = \overrightarrow{[\![d]\!]} \iff c \overset{\mathbb{HA}}{=} d$$

$\mathbb{HA}$ is isomorphic to $\mathsf{Mat}\,\mathsf{k}[x]$

Reverse Functional Circuits

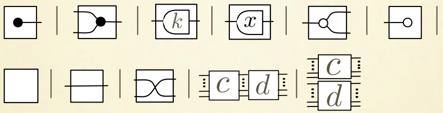
Circuit diagrams of $\overleftarrow{\mathsf{Circ}}$ are generated by the grammar



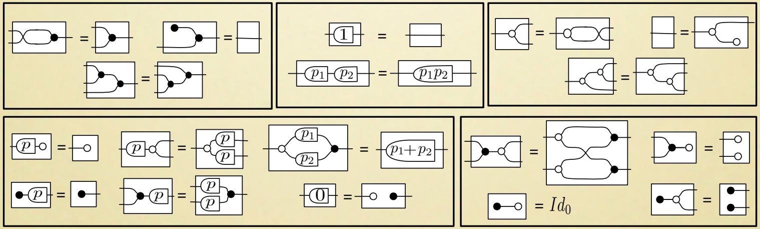
The map $\overleftarrow{[\![\cdot]\!]} \colon \overleftarrow{\mathsf{Circ}} \to \mathsf{Mat}\,\mathsf{k}[x]^{op}$ is defined dually to $\overrightarrow{[\![\cdot]\!]}$.

Reverse Functional Circuits

Circuit diagrams of $\overleftarrow{\mathsf{Circ}}$ are generated by the grammar

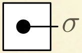


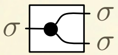
The map $\overleftarrow{[\![\cdot]\!]} \colon \overleftarrow{\mathsf{Circ}} \to \mathsf{Mat}\,\mathsf{k}[x]^{op}$ is defined dually to $\overrightarrow{[\![\cdot]\!]}$.
The theory $\mathbb{HA}^{op}$ is $\overleftarrow{\mathsf{Circ}}$ quotiented by the following equations:



Semantics of Generalised Circuits

When we allow combinations of functional and co-functional circuits (like in feedbacks) we may get *relational* behaviors.

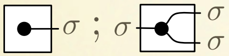
For instance, 

; 

expresses the diagonal relation.

Semantics of Generalised Circuits

When we allow combinations of functional and co-functional circuits (like in feedbacks) we may get *relational* behaviors.

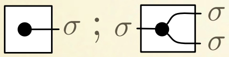
For instance, 

expresses the diagonal relation.

Moreover, polynomials are not enough: we need *fractions* of polynomials.

Semantics of Generalised Circuits

When we allow combinations of functional and co-functional circuits (like in feedbacks) we may get *relational* behaviors.

For instance, 

expresses the diagonal relation.

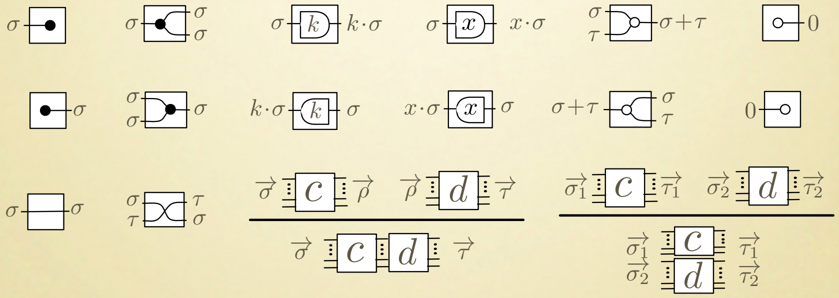
Moreover, polynomials are not enough: we need *fractions* of polynomials.

$$\begin{tikzcd}
\mathsf{k}[[x]] \arrow[rr, hook] & & \mathsf{k}((x)) \\
& \mathsf{k}\langle x\rangle \arrow[ul, hook] \arrow[dr, hook] & \\
\mathsf{k}[x] \arrow[uu, hook, "\hat{\cdot}"] \arrow[ur, hook] \arrow[rr, hook] & & \mathsf{k}(x) \arrow[uu, hook, "\tilde{\cdot}"']
\end{tikzcd}$$

$\mathsf{k}[x]$	the ring of polynomials	$\sum_0^n k_i x^i$ for some natural n
$\mathsf{k}(x)$	the field of fractions of polynomials	$\frac{p}{q}$ for $p, q \in \mathsf{k}[x]$ with $q \neq 0$
$\mathsf{k}\langle x\rangle$	the ring of rationals	$\frac{\sum_0^n k_i x^i}{\sum_0^m l_j x^j}$ with $l_0 \neq 0$
$\mathsf{k}[[x]]$	the ring of formal power series	$\sum_0^\infty k_i x^i$
$\mathsf{k}((x))$	the field of Laurent series	$\sum_d^\infty k_i x^i$ for some interger d

Denotational Semantics

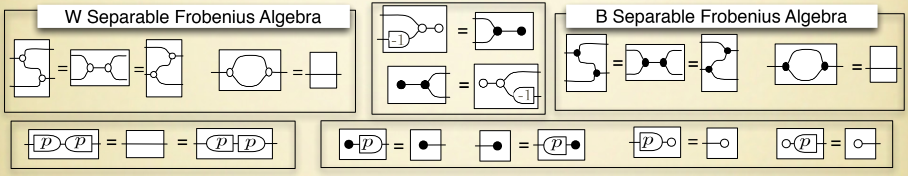
The semantics $[\![\cdot]\!]$ maps a circuit to a linear relation between stream vectors



σ
σ σ σ
σ k $k\cdot\sigma$
σ x $x\cdot\sigma$
σ τ $\sigma+\tau$
0
σ
σ σ σ
$k\cdot\sigma$ k σ
$x\cdot\sigma$ x σ
$\sigma+\tau$ σ τ
0
σ σ
σ τ τ σ
$\vec{\sigma}$ c $\vec{\rho}$ $\vec{\rho}$ d $\vec{\tau}$
$\vec{\sigma}$ c d $\vec{\tau}$
$\vec{\sigma_1}$ c $\vec{\tau_1}$ $\vec{\sigma_2}$ d $\vec{\tau_2}$
$\vec{\sigma_1}$ $\vec{\sigma_2}$ c d $\vec{\tau_1}$ $\vec{\tau_2}$

Axiomatisation of $[\![\cdot]\!]$

The equational theory of *interacting Hopf algebras* ($\mathbb{IH}$) is Circ quotiented by the axioms of $\mathbb{HA}$, $\mathbb{HA}^{op}$, plus the following:



Soundness and Completeness

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

Kleene's Theorem

$\mathbb{IH}$ is the category of subspaces over the field $\mathsf{k}(x)$.

Full Abstraction

Theorem (?)

For any c and d in Circ

$$[\![c]\!] = [\![d]\!] \iff \langle c \rangle = \langle d \rangle$$

Full Abstraction

Theorem (?)

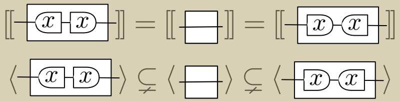
For any c and d in Circ

$$[\![c]\!] = [\![d]\!] \iff \langle c \rangle = \langle d \rangle$$

Not true in general.
The denotational semantics is *coarser* than the operational semantics.

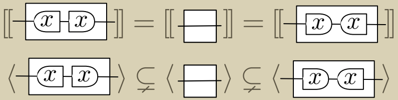
Full Abstraction

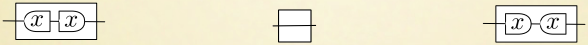
A counterexample



Full Abstraction

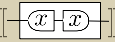
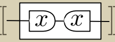
A counterexample

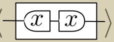
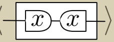


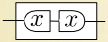


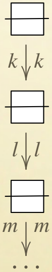
Full Abstraction

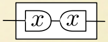
A counterexample

$[\![$  $]\!] = [\![$ $]\!] = [\![$  $]\!]$

$\langle$  $\rangle \subsetneq \langle$ $\rangle \subsetneq \langle$  $\rangle$

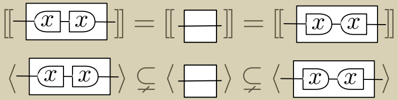


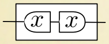


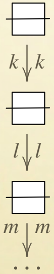


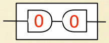
Full Abstraction

A counterexample



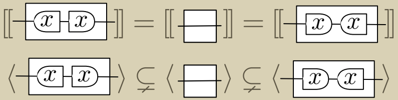


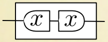


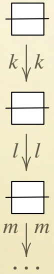


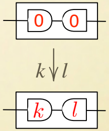
Full Abstraction

A counterexample



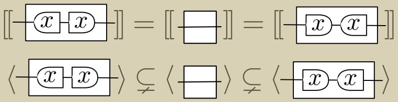


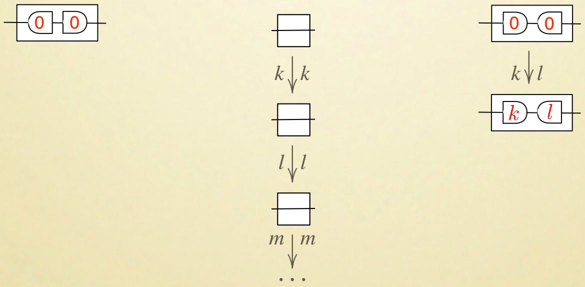




Full Abstraction

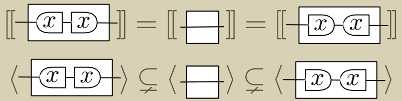
A counterexample

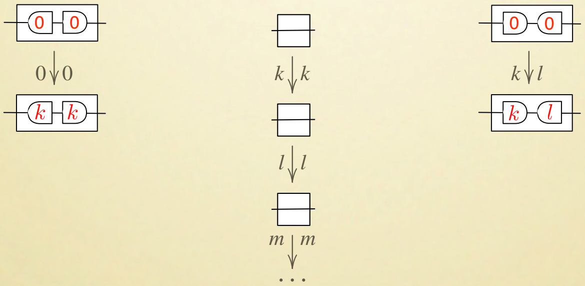




Full Abstraction

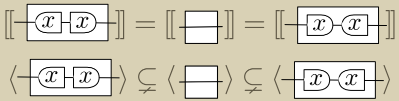
A counterexample

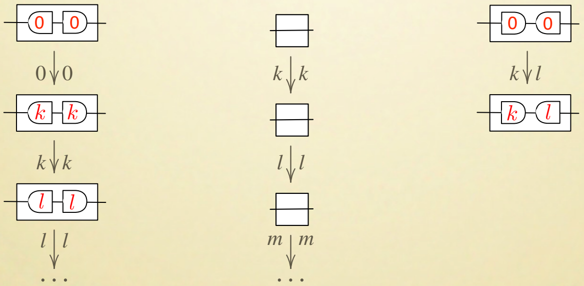




Full Abstraction

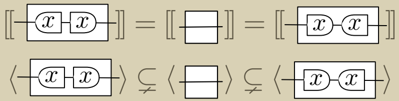
A counterexample

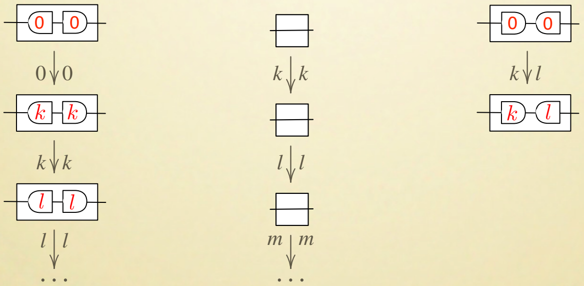


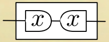
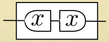


Full Abstraction

A counterexample





We say that  has *deadlocks* and  needs *initialisation*.

Full Abstraction

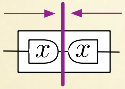
Theorem

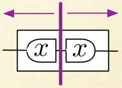
For any c and d in Circ **deadlock and initialisation free**

$$[\![c]\!] = [\![d]\!] \iff \langle c \rangle = \langle d \rangle$$

Realisability

In presence of deadlocks or initialisation, we cannot determine directionality of the flow.

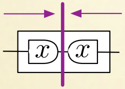


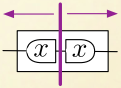


A trace for these circuits cannot be thought as the execution of a state-machine.

Realisability

In presence of deadlocks or initialisation, we cannot determine directionality of the flow.





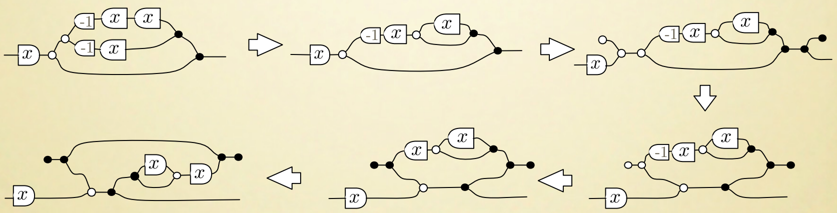
A trace for these circuits cannot be thought as the execution of a state-machine.
However, all the circuit diagrams can be put into an executable form using the equational theory $\overset{\mathbb{IH}}{=}$.

Realisability Theorem

For any circuit c of Circ there exists
d deadlock and initialisation free such that $c \overset{\mathbb{IH}}{=} d$.

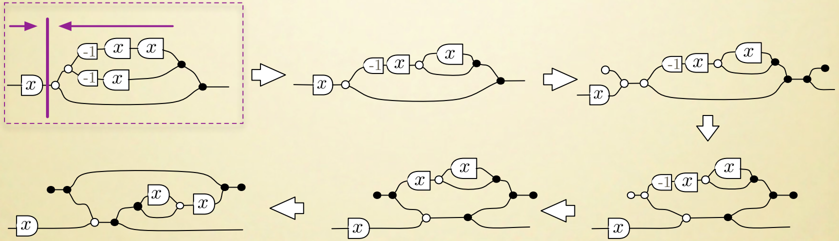
Realisation via $\mathbb{IH}$-rewriting

Implementing the Fibonacci circuit



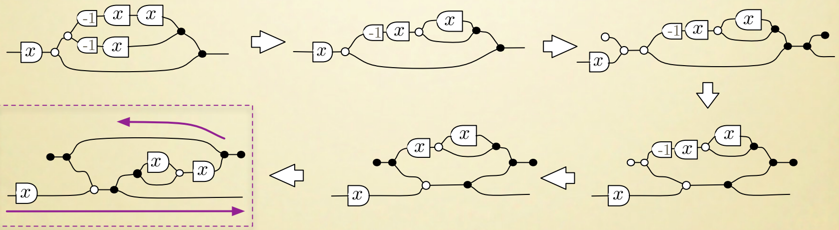
Realisation via $\mathbb{IH}$-rewriting

Implementing the Fibonacci circuit



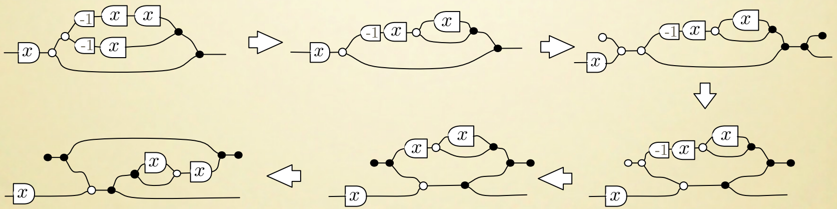
Realisation via $\mathbb{IH}$-rewriting

Implementing the Fibonacci circuit



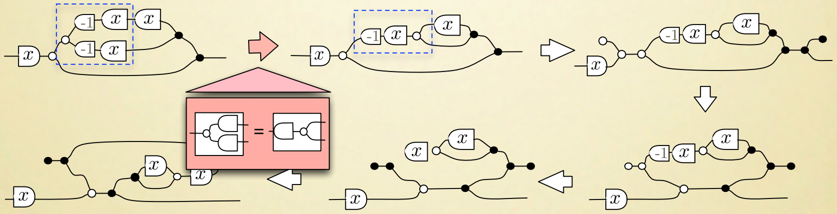
Realisation via $\mathbb{IH}$-rewriting

Implementing the Fibonacci circuit



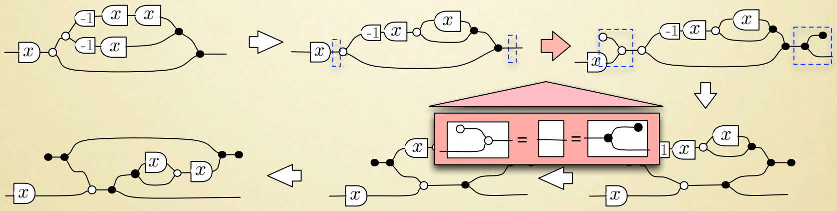
Realisation via IH-rewriting

Implementing the Fibonacci circuit



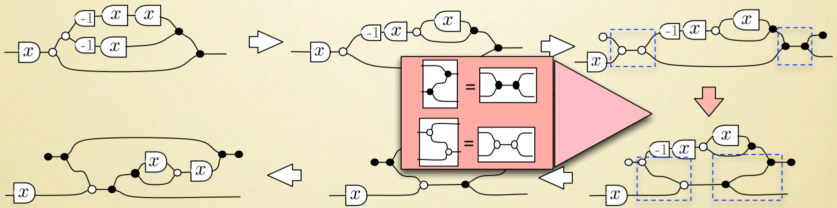
Realisation via $\mathbb{IH}$-rewriting

Implementing the Fibonacci circuit



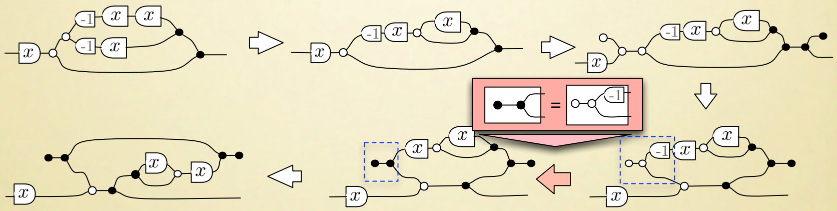
Realisation via IH-rewriting

Implementing the Fibonacci circuit



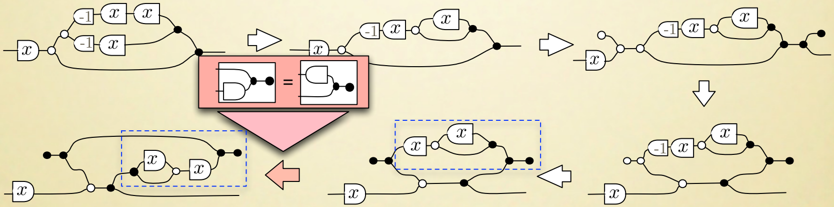
Realisation via IH-rewriting

Implementing the Fibonacci circuit



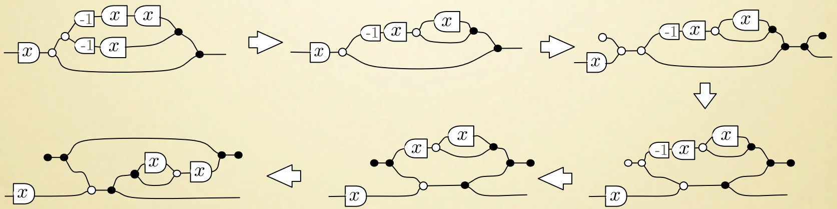
Realisation via IH-rewriting

Implementing the Fibonacci circuit



Realisation via $\mathbb{IH}$-rewriting

Implementing the Fibonacci circuit



Conclusions

- The calculus of signal flow diagrams does not rely on flow directionality as a primitive.

  The reason why physics has ceased to look for causes is that in fact there are no such things. The law of causality, I believe, like much that passes muster among philosophers, is a relic of a bygone age, surviving, like the monarchy, only because it is erroneously supposed to do no harm.

  *(*Bertrand Russell -1913*)*

- This allows for a more flexible syntax, disclosing a rich and elegant mathematical playground: $\mathbb{IH}$.
- Whenever flow directionality matters, the realisability theorem allows us rewrite any circuit diagram into an executable form.

Beyond this talk

- Full Abstraction for Signal Flow Graphs - PoPL'15.
- A Categorical Semantics for Signal Flow Graphs - CONCUR'14.
- Interacting Bialgebras are Frobenius - FoSSaCS'14.
- Interacting Hopf Algebras - `http://arxiv.org/abs/1403.7048`
- Paweł's blog - `http://graphicallinearalgebra.net/`
- Fabio's Ph.D. Thesis - Interacting Hopf Algebras: the theory of linear systems `http://zanasi.com/fabio/IHthesis_FZ.pdf`