The Algebra of Signal Flow Graphs

Filippo Bonchi, Paweł Sobociński, Fabio Zanasi

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 $k[x]$.

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 $k[x]$.

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

\[ k \in \mathbb{k} \]
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

![Diagram of Signal Flow Graphs]

\[ k \in k \]
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

![Signal Flow Graph Diagram]
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

\[ k \in k \]

\[ k \]

\[ l \]
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

\[ k \in k \]

\[ l \]
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

\[ k_1 \quad 0 \quad k_1 \]
\[ k_2 \quad k_1 \quad k_2 \]
\[ k_3 \quad k_2 \quad \vdots \]
Signal Flow Graphs

Two examples:

Both circuits implement the generating function

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

Can we check this statically?
Signal Flow Graphs

Two examples:

Both circuits implement the generating function:

\[ (1 - x^2) = 1 + 2x^2 + 3x^3 + \ldots \]

Can we check this statically?
Signal Flow Graphs

Two examples:

Both circuits implement the generating function

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

Can we check this statically?
Signal Flow Graphs

Two examples:

Both circuits implement the generating function

\[
(1 - x^2) = x + 2x^2 + 3x^3 + \ldots
\]

Can we check this statically?
Signal Flow Graphs

Two examples:

Both circuits implement the generating function

\[
(1 - x) = x + 2x^2 + 3x^3 + \ldots
\]

Can we check this statically?
Signal Flow Graphs

Two examples:

Both circuits implement the generating function

\[ \frac{1}{(1 - x)^2} = 1 + 2x + 3x^2 + \cdots \]

Can we check this statically?
Signal Flow Graphs

Two examples:

Both circuits implement the generating function:

\[ (1 - x) x = x + 2x^2 + 3x^3 + \ldots \]

Can we check this **statically**?
Signal Flow Graphs

Two examples:

Both circuits implement the generating function $1 - x^2 = x + 2x^2 + 3x^3 + \ldots$. Can we check this statically?
Signal Flow Graphs

Two examples:

Both circuits implement the generating function

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

Can we check this staticaly?
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.
  - 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.
  - 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 $\mathcal{HA}$ of functional circuits

Functional circuits are the string diagrams generated by the grammar

$c, d ::= \bullet | \circlearrowright | k | x | \square | \begin{array}{c} p \end{array} | \begin{array}{c} \circlearrowright \end{array}$

subject to the following equations:

where, for a polynomial $p = k_0 + k_1 x + \cdots + k_n x^n$, $\begin{array}{c} p \end{array}$ is
The theory HA of functional circuits

Functional circuits are the string diagrams generated by the grammar

\[ c, d ::= \quad \quad \quad \quad \quad \quad \]

subject to the following equations:

where, for a polynomial \( p=k_0+k_1x+\cdots+k_nx^n \), \( \square \) is
The theory $\mathbb{HA}$ of functional circuits

Functional circuits are the string diagrams generated by the grammar

$$c, d ::= \begin{array}{c}
\bullet \\
\bullet \\
k
x \\
\bullet \\
\bullet \\
cd \\
cd
\end{array}$$

subject to the following equations:

where, for a polynomial $p = k_0 + k_1 x + \cdots + k_n x^n$, $\mathcal{P}$ is
The theory $\mathcal{HA}$ of functional circuits

Functional circuits are the string diagrams generated by the grammar

$$c, d ::= \begin{array}{c}
\bullet \\
\mid \\
k \\
\mid x \\
\mid \bigcirc \\
\mid \bigcirc c d \\
\end{array}$$

subject to the following equations:

$$\begin{array}{c}
\begin{array}{ccc}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\bullet \\
\mid \\
k \\
\mid x \\
\end{array}
\end{array}
\end{array} & = & \begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\bullet \\
\mid \\
k \\
\end{array}
\end{array}
\end{array}
\end{array} \\
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\bullet \\
\mid \\
k \\
\end{array}
\end{array}
\end{array} & = & \begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\bullet \\
\mid \\
k \\
\end{array}
\end{array}
\end{array}
\end{array} \\
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\bullet \\
\mid \\
x \\
\end{array}
\end{array}
\end{array} & = & \begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\bullet \\
\mid \\
x \\
\end{array}
\end{array}
\end{array}
\end{array}
\end{array}$$

$$\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\bullet \\
\mid \\
x \\
\end{array}
\end{array}
\end{array} & = & \begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\bullet \\
\mid \\
x \\
\end{array}
\end{array}
\end{array}
\end{array} \\
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\bullet \\
\mid \\
x \\
\end{array}
\end{array}
\end{array} & = & \begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\bullet \\
\mid \\
x \\
\end{array}
\end{array}
\end{array}
\end{array} \\
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\bullet \\
\mid \\
x \\
\end{array}
\end{array}
\end{array} & = & \begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\bullet \\
\mid \\
x \\
\end{array}
\end{array}
\end{array}
\end{array}
\end{array}$$

$$\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\bullet \\
\mid \\
x \\
\end{array}
\end{array}
\end{array} & = & \begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\bullet \\
\mid \\
x \\
\end{array}
\end{array}
\end{array}
\end{array} \\
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\bullet \\
\mid \\
x \\
\end{array}
\end{array}
\end{array} & = & \begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\bullet \\
\mid \\
x \\
\end{array}
\end{array}
\end{array}
\end{array} \\
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\bullet \\
\mid \\
x \\
\end{array}
\end{array}
\end{array} & = & \begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\bullet \\
\mid \\
x \\
\end{array}
\end{array}
\end{array}
\end{array}
\end{array}$$

$$\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\bullet \\
\mid \\
x \\
\end{array}
\end{array}
\end{array} & = & \begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\bullet \\
\mid \\
x \\
\end{array}
\end{array}
\end{array}
\end{array} \\
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\bullet \\
\mid \\
x \\
\end{array}
\end{array}
\end{array} & = & \begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\bullet \\
\mid \\
x \\
\end{array}
\end{array}
\end{array}
\end{array} \\
\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\bullet \\
\mid \\
x \\
\end{array}
\end{array}
\end{array} & = & \begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\bullet \\
\mid \\
x \\
\end{array}
\end{array}
\end{array}
\end{array}
\end{array}$$

where, for a polynomial $p = k_0 + k_1 x + \cdots + k_n x^n$, $\begin{array}{c}
\begin{array}{c}
\begin{array}{c}
\bullet \\
\mid \\
x \\
\end{array}
\end{array}
\end{array}$ is
Semantics of functional circuits

- Functional circuits modulo the equations are in 1-1 correspondence with matrices over the polynomial ring \( k[x] \).
- Example: check the semantics of \( x^2x^3 \) using the equational theory \( \mathbb{HA} \).

![Diagram of functional circuits with arrows and nodes labeled with variables and integers.](image-url)
Semantics of functional circuits

- Functional circuits modulo the equations are in 1-1 correspondence with matrices over the polynomial ring $k[x]$.

- Example: check the semantics of $x^2 \cdot 3x$ using the equational theory $\mathbb{HA}$.

\[
x^2 \cdot 3x = 3x^2 \cdot 3x = (3x^2)(3x) = 9x^3
\]
Semantics of functional circuits

- Functional circuits modulo the equations are in 1-1 correspondence with matrices over the polynomial ring $k[x]$.
- Example: check the semantics of $x^2$ using the equational theory $HA$. 

$$x^2 = x^2$$

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

11 / 18
Semantics of functional circuits

- Functional circuits modulo the equations are in 1-1 correspondence with matrices over the polynomial ring \( k[x] \).
- Example: check the semantics of \[ x^2 + 3x^3 + x \]
  using the equational theory \( \mathbb{HA} \).

\[
\begin{align*}
\begin{bmatrix}
1 & 0 \\
0 & 1
\end{bmatrix}
\end{align*}
\]
Semantics of functional circuits

- Functional circuits modulo the equations are in 1-1 correspondence with matrices over the polynomial ring $k[x]$.
- Example: check the semantics of $\begin{pmatrix} x & 3 \\ 2 & x \end{pmatrix}$ using the equational theory $\mathbb{HA}$. 

\[
\begin{pmatrix} 3 & x & 6 \\ 2 & x & 3 \end{pmatrix}
\]

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

- Functional circuits modulo the equations are in 1-1 correspondence with matrices over the polynomial ring $k[x]$.

- Example: check the semantics of using the equational theory $\mathbb{H}A$. 

\[
\begin{align*}
\begin{bmatrix}
3 & x \\
6 & 2
\end{bmatrix}
\Rightarrow
\begin{bmatrix}
x & 3 \\
2 & 6
\end{bmatrix}
\Rightarrow
\begin{bmatrix}
x & 3 \\
x & 2
\end{bmatrix}
\Rightarrow
\begin{bmatrix}
x & 3 \\
2 & 3
\end{bmatrix}
\Rightarrow
\begin{bmatrix}
x & 3 \\
2 & 3
\end{bmatrix}
\end{align*}
\]
Semantics of functional circuits

- Functional circuits modulo the equations are in 1-1 correspondence with matrices over the polynomial ring $k[x]$.

- 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

$$c, d ::= \bullet | \begin{array}{c}
\bullet
\end{array} | \begin{array}{c}
k
\end{array} | \begin{array}{c}
x
\end{array} | \begin{array}{c}
\bullet
\end{array} | \begin{array}{c}
\bullet
\end{array} | \begin{array}{c}
\bullet
\end{array} | \begin{array}{c}
\bullet
\end{array} | \begin{array}{c}
\bullet
\end{array} | \begin{array}{c}
\bullet
\end{array}$$

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

<table>
<thead>
<tr>
<th>Equation</th>
<th>Diagram</th>
</tr>
</thead>
<tbody>
<tr>
<td>$c \cdot c = d \cdot d$</td>
<td><img src="Diagram1.png" alt="Diagram" /></td>
</tr>
<tr>
<td>$c \cdot d = d \cdot c$</td>
<td><img src="Diagram2.png" alt="Diagram" /></td>
</tr>
<tr>
<td>$c \cdot 0 = 0$</td>
<td><img src="Diagram3.png" alt="Diagram" /></td>
</tr>
<tr>
<td>$c \cdot 1 = c$</td>
<td><img src="Diagram4.png" alt="Diagram" /></td>
</tr>
<tr>
<td>$c \cdot c = d \cdot d$</td>
<td><img src="Diagram5.png" alt="Diagram" /></td>
</tr>
<tr>
<td>$c \cdot d = d \cdot c$</td>
<td><img src="Diagram6.png" alt="Diagram" /></td>
</tr>
<tr>
<td>$c \cdot 0 = 0$</td>
<td><img src="Diagram7.png" alt="Diagram" /></td>
</tr>
<tr>
<td>$c \cdot 1 = c$</td>
<td><img src="Diagram8.png" alt="Diagram" /></td>
</tr>
</tbody>
</table>

$\bullet = Id_0$
The theory $\mathcal{IH}$ of generalised circuits

Generalised circuits are string diagrams generated by the grammar

$$c, d ::= \bullet | \begin{array}{c} k \end{array} | \begin{array}{c} x \end{array} | \begin{array}{c} \phantom{k} \end{array} | \begin{array}{c} \phantom{x} \end{array}$$

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

**W Separable Frobenius Algebra**

$$\begin{array}{c} \phantom{k} \end{array} = \begin{array}{c} \phantom{x} \end{array} = \begin{array}{c} \phantom{k} \end{array} = \begin{array}{c} \phantom{x} \end{array} = \begin{array}{c} \phantom{k} \end{array} = \begin{array}{c} \phantom{x} \end{array}$$

**B Separable Frobenius Algebra**

$$\begin{array}{c} \phantom{k} \end{array} = \begin{array}{c} \phantom{x} \end{array} = \begin{array}{c} \phantom{k} \end{array} = \begin{array}{c} \phantom{x} \end{array} = \begin{array}{c} \phantom{k} \end{array} = \begin{array}{c} \phantom{x} \end{array}$$
Circuits do not generally have a univocal flow direction — a relational model is required.

For instance, \( \square ; \quad \square \uparrow \sigma \) expresses the diagonal relation.
Semantics of Generalised Circuits

The semantics $\llbracket \cdot \rrbracket$ maps a circuit into a linear relation (subspace):

$$\llbracket k \cdot \sigma \rrbracket$$

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

$$\llbracket c \rrbracket = \llbracket d \rrbracket \iff c \equiv d$$

The key technical step in the proof consists in reducing a circuit in its Hermite Normal Form.
Graphical reasoning in $\mathbb{IH}$

Check: \[ x x \quad \text{and} \quad x 2 x - 1 x 2 \]

Implement $\frac{1}{(1-x)^2}$.

Proof strategy:

- Represent the two SFGs as generalised circuits

- Represent the specification as a generalised circuit:

\[ \sigma \cdot \frac{1}{(1-x)^2} \]

- 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

\[
\begin{array}{c}
\text{implementation} = \text{implementation} \\
\text{specification} \rightarrow \text{implementation}
\end{array}
\]

- rich mathematical playground

Hopf Algebra of functional circuit

Hopf Algebra of reverse functional circuits

Interaction yields two Frobenius Algebras
Future Work

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

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

A path toward an answer...

Systems of linear difference equations
Diophantine Systems of linear difference equations