Layered monoidal theories

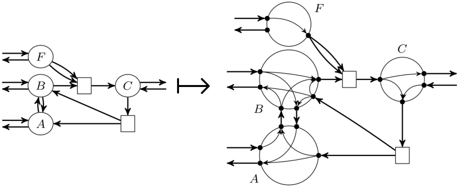
Leo Lobski

University College London

27 February 2026
Cambridge Logic and Semantics Seminar

Outline

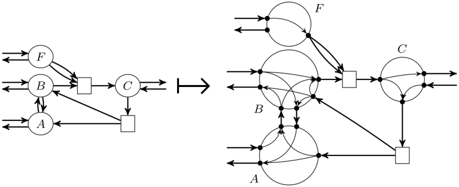
Motivation: layers of abstraction



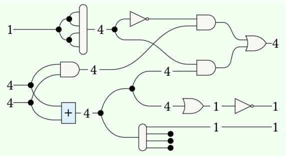
[1]

1 Andersen, Flamm, Merkle, Stadler. *Chemical Transformation Motifs*. 2017

Motivation: layers of abstraction



[1]

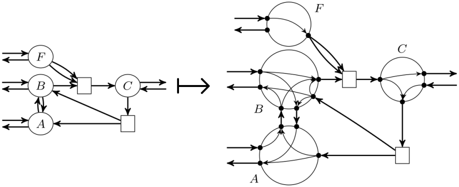


[2]

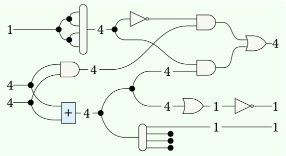
[1] Andersen, Flamm, Merkle, Stadler. *Chemical Transformation Motifs*. 2017

[2] Kaye. *Foundations of Digital Circuits*. 2024

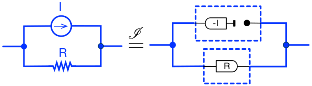
Motivation: layers of abstraction



1



2



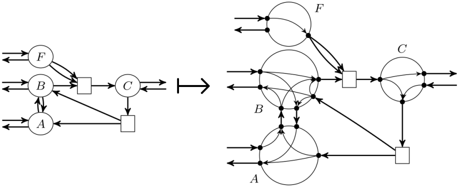
3

[1] Andersen, Flamm, Merkle, Stadler. *Chemical Transformation Motifs*. 2017

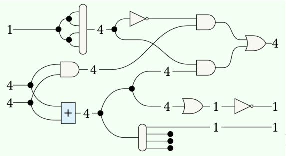
[2] Kaye. *Foundations of Digital Circuits*. 2024

[3] Boisseau, Sobociński. *String Diagrammatic Electrical Circuit Theory*. 2021

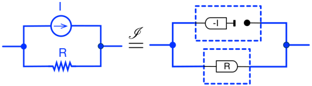
Motivation: layers of abstraction



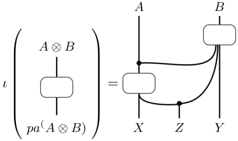
1



2



3



4

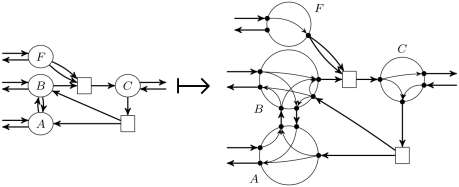
[1] Andersen, Flamm, Merkle, Stadler. *Chemical Transformation Motifs*. 2017

[2] Kaye. *Foundations of Digital Circuits*. 2024

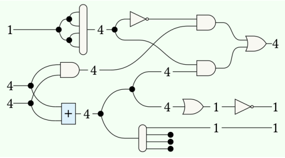
[3] Boisseau, Sobociński. *String Diagrammatic Electrical Circuit Theory*. 2021

[4] Englberger, Singh. *Causal Abstractions, Categorically Unified*. 2025

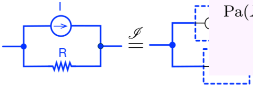
Motivation: layers of abstraction

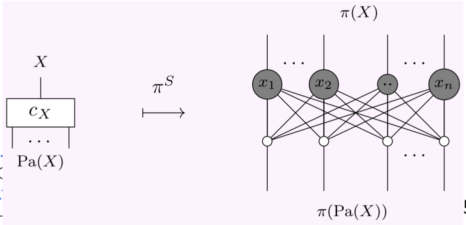


1

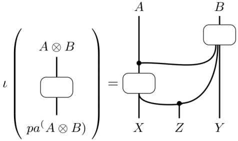


2





5



4

---

[1] Andersen, Flamm, Merkle, Stadler. *Chemical Transformation Motifs*. 2017

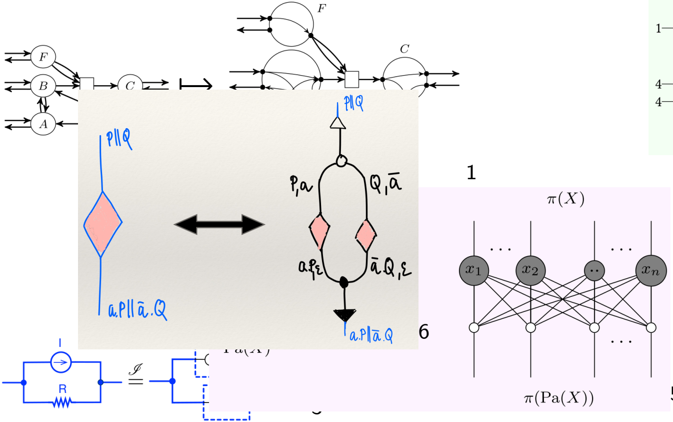
[2] Kaye. *Foundations of Digital Circuits*. 2024

[3] Boisseau, Sobociński. *String Diagrammatic Electrical Circuit Theory*. 2021

[4] Englberger, Singh. *Causal Abstractions, Categorically Unified*. 2025

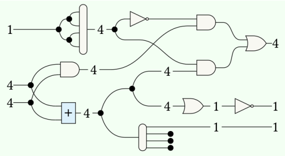
[5] Lorenz, Tull. *Causal and Compositional Abstraction*. 2026

Motivation: layers of abstraction



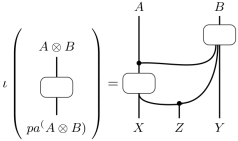
1

6



2

5



4

[1] Andersen, Flamm, Merkle, Stadler. *Chemical Transformation Motifs*. 2017

[2] Kaye. *Foundations of Digital Circuits*. 2024

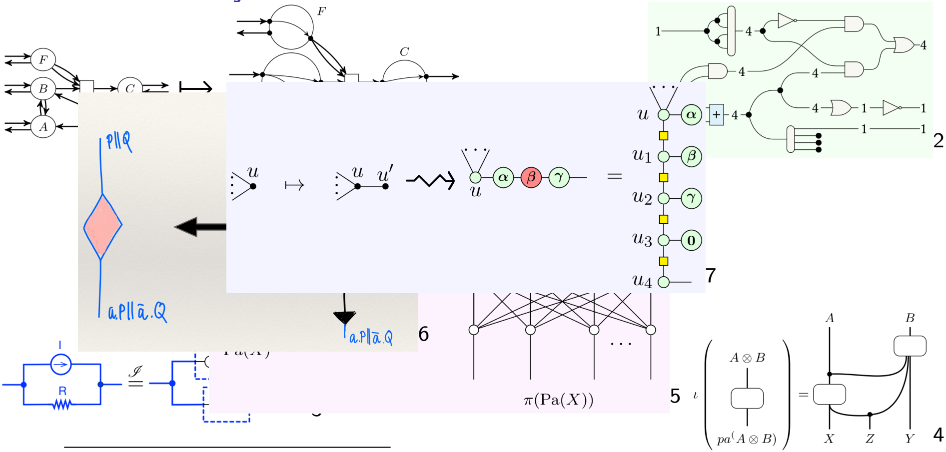
[3] Boisseau, Sobociński. *String Diagrammatic Electrical Circuit Theory*. 2021

[4] Englberger, Singh. *Causal Abstractions, Categorically Unified*. 2025

[5] Lorenz, Tull. *Causal and Compositional Abstraction*. 2026

[6] Krivine. *Diagrammatic reasoning for physical systems and composite explanations*. 2019

Motivation: layers of abstraction



[1] Andersen, Flamm, Merkle, Stadler. *Chemical Transformation Motifs*. 2017

[2] Kaye. *Foundations of Digital Circuits*. 2024

[3] Boisseau, Sobociński. *String Diagrammatic Electrical Circuit Theory*. 2021

[4] Englberger, Singh. *Causal Abstractions, Categorically Unified*. 2025

[5] Lorenz, Tull. *Causal and Compositional Abstraction*. 2026

[6] Krivine. *Diagrammatic reasoning for physical systems and composite explanations*. 2019

[7] Backens, Miller-Bakewell, de Felice, L., van de Wetering. *There and back again*. 2021

Signatures and types

Definition

A *layered signature* is a tuple $(\Omega, \mathcal{F}, \mathcal{M}_\omega)$, where

Signatures and types

Definition

A *layered signature* is a tuple $(\Omega, \mathcal{F}, \mathcal{M}_\omega)$, where

- Ω is a set,

Signatures and types

Definition

A *layered signature* is a tuple $(\Omega, \mathcal{F}, \mathcal{M}_\omega)$, where

- Ω is a set,
- $\mathcal{F} : \Omega \times \Omega \to \mathbf{Set}$ is a function,

Signatures and types

Definition

A *layered signature* is a tuple $(\Omega, \mathcal{F}, \mathcal{M}_\omega)$, where

- Ω is a set,
- $\mathcal{F} : \Omega \times \Omega \to \mathbf{Set}$ is a function,
- $\mathcal{M}_\omega = (C_\omega, \Sigma_\omega)$ is a monoidal signature for each $\omega \in \Omega$.

Signatures and types

Definition

A *layered signature* is a tuple $(\Omega, \mathcal{F}, \mathcal{M}_\omega)$, where

- Ω is a set,
- $\mathcal{F} : \Omega \times \Omega \to \textbf{Set}$ is a function,
- $\mathcal{M}_\omega = (C_\omega, \Sigma_\omega)$ is a monoidal signature for each $\omega \in \Omega$.

Definition

The *types* of a layered signature $(\Omega, \mathcal{F}, \mathcal{M}_\omega)$ are defined as:

Signatures and types

Definition

A *layered signature* is a tuple $(\Omega, \mathcal{F}, \mathcal{M}_\omega)$, where

- Ω is a set,
- $\mathcal{F} : \Omega \times \Omega \to \mathbf{Set}$ is a function,
- $\mathcal{M}_\omega = (C_\omega, \Sigma_\omega)$ is a monoidal signature for each $\omega \in \Omega$.

Definition

The *types* of a layered signature $(\Omega, \mathcal{F}, \mathcal{M}_\omega)$ are defined as:

$$\frac{}{\varepsilon : \varepsilon} \quad \frac{\omega \in \Omega}{\varepsilon : \omega} \quad \frac{\omega \in \Omega \qquad a \in C_\omega}{a : \omega} \quad \frac{A : \omega \qquad f \in \mathcal{F}(\omega, \tau)}{f(A) : \tau}$$

$$\frac{A : \omega \qquad B : \omega}{AB : \omega} \quad \frac{T \qquad S}{T, S},$$

Signatures and types

Definition

A *layered signature* is a tuple $(\Omega, \mathcal{F}, \mathcal{M}_\omega)$, where

- Ω is a set,
- $\mathcal{F} : \Omega \times \Omega \to \mathbf{Set}$ is a function,
- $\mathcal{M}_\omega = (C_\omega, \Sigma_\omega)$ is a monoidal signature for each $\omega \in \Omega$.

Definition

The *types* of a layered signature $(\Omega, \mathcal{F}, \mathcal{M}_\omega)$ are defined as:

$$\frac{}{\varepsilon : \varepsilon} \quad \frac{\omega \in \Omega}{\varepsilon : \omega} \quad \frac{\omega \in \Omega \qquad a \in C_\omega}{a : \omega} \quad \frac{A : \omega \qquad f \in \mathcal{F}(\omega, \tau)}{f(A) : \tau}$$

$$\frac{A : \omega \qquad B : \omega}{AB : \omega} \quad \frac{T \qquad S}{T, S},$$

such that , is a monoid with unit $\varepsilon : \varepsilon$.

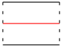
Basic terms

Basic terms

$$\frac{\omega \in \Omega}{\quad}\ \text{(int-unit)}$$

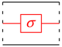
$: (\varepsilon : \omega \mid \varepsilon : \omega)$

$$\frac{A : \omega \qquad A \neq \varepsilon}{\quad}\ \text{(int-id)}$$



$: (A : \omega \mid A : \omega)$

$$\frac{\sigma \in \Sigma_\omega(a, b)}{\quad}\ \text{(int-gen)}$$



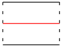
$: (a : \omega \mid b : \omega)$

Basic terms

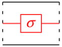
$\omega \in \Omega$ — (int-unit)

 $: (\varepsilon : \omega \mid \varepsilon : \omega)$

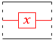
$A : \omega \qquad A \neq \varepsilon$ — (int-id)

 $: (A : \omega \mid A : \omega)$

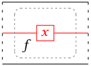
$\sigma \in \Sigma_\omega(a, b)$ — (int-gen)



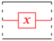
$: (a : \omega \mid b : \omega)$

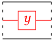


$: (A : \omega \mid B : \omega) \qquad f \in \mathcal{F}(\omega, \tau)$ — (int-box)

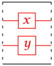


$: (f(A) : \tau \mid f(B) : \tau)$



$: (A : \omega \mid B : \omega)$ 

$: (C : \omega \mid D : \omega)$ — (int-tensor)



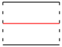
$: (AC : \omega \mid BD : \omega)$

Basic terms

(int-unit): premise $\omega \in \Omega$; conclusion:

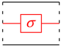
$: (\varepsilon : \omega \mid \varepsilon : \omega)$

(int-id): premises $A : \omega$ and $A \neq \varepsilon$; conclusion:



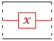
$: (A : \omega \mid A : \omega)$

(int-gen): premise $\sigma \in \Sigma_\omega(a, b)$; conclusion:

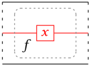


$: (a : \omega \mid b : \omega)$

(int-box): premises

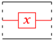


$: (A : \omega \mid B : \omega)$ and $f \in \mathcal{F}(\omega, \tau)$; conclusion:

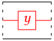


$: (f(A) : \tau \mid f(B) : \tau)$

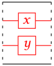
(int-tensor): premises



$: (A : \omega \mid B : \omega)$ and



$: (C : \omega \mid D : \omega)$; conclusion:



$: (AC : \omega \mid BD : \omega)$

(ext-unit): no premises; conclusion:

$: (\varepsilon : \varepsilon \mid \varepsilon : \varepsilon)$

$$\frac{x : (T \mid S) \qquad y : (S \mid U)}{x; y : (T \mid U)}\ \text{(comp)}$$

$$\frac{x : (T \mid S) \qquad y : (U \mid W)}{x \otimes y : (T, U \mid S, W)}\ \text{(ext-tensor)}$$

Theories

A *recursive sorting procedure* extends basic terms such that the set of internal terms is unchanged.

Theories

A *recursive sorting procedure* extends basic terms such that the set of internal terms is unchanged. Let us fix a recursive sorting procedure.

Theories

A *recursive sorting procedure* extends basic terms such that the set of internal terms is unchanged. Let us fix a recursive sorting procedure.

Definition

A *layered theory* $(\mathcal{L}, E^0, E^1, \eta, E^2)$ consists of:

Theories

A *recursive sorting procedure* extends basic terms such that the set of internal terms is unchanged. Let us fix a recursive sorting procedure.

Definition

A *layered theory* $(\mathcal{L}, E^0, E^1, \eta, E^2)$ consists of:

- a layered signature $\mathcal{L}$,

Theories

A *recursive sorting procedure* extends basic terms such that the set of internal terms is unchanged. Let us fix a recursive sorting procedure.

Definition

A *layered theory* $(\mathcal{L}, E^0, E^1, \eta, E^2)$ consists of:

- a layered signature $\mathcal{L}$,
- a set of 0-equations E^0,

Theories

A *recursive sorting procedure* extends basic terms such that the set of internal terms is unchanged. Let us fix a recursive sorting procedure.

Definition

A *layered theory* $(\mathcal{L}, E^0, E^1, \eta, E^2)$ consists of:

- a layered signature $\mathcal{L}$,
- a set of 0-equations E^0,
- a set of 1-equations E^1 with respect to E^0,

Theories

A *recursive sorting procedure* extends basic terms such that the set of internal terms is unchanged. Let us fix a recursive sorting procedure.

Definition

A *layered theory* $(\mathcal{L}, E^0, E^1, \eta, E^2)$ consists of:

- a layered signature $\mathcal{L}$,
- a set of 0-equations E^0,
- a set of 1-equations E^1 with respect to E^0,
- a choice of 2-cells η with respect to E^0,

Theories

A *recursive sorting procedure* extends basic terms such that the set of internal terms is unchanged. Let us fix a recursive sorting procedure.

Definition

A *layered theory* $(\mathcal{L}, E^0, E^1, \eta, E^2)$ consists of:

- a layered signature $\mathcal{L}$,
- a set of 0-equations E^0,
- a set of 1-equations E^1 with respect to E^0,
- a choice of 2-cells η with respect to E^0,
- a set of 2-equations E^2 with respect to E^1 and η.

Structural equations

The *structural 0-equations* S^0 are given by:

Structural equations

The *structural 0-equations* S^0 are given by:

$$(AB)C : \omega = A(BC) : \omega, \qquad \varepsilon A : \omega = A : \omega = A\varepsilon : \omega,$$
$$f(AB) : \tau = f(A)f(B) : \tau, \qquad f(\varepsilon) : \tau = \varepsilon : \tau.$$

Structural equations

The *structural 0-equations* S^0 are given by:

$$(AB)C : \omega = A(BC) : \omega, \qquad \varepsilon A : \omega = A : \omega = A\varepsilon : \omega,$$
$$f(AB) : \tau = f(A)f(B) : \tau, \qquad f(\varepsilon) : \tau = \varepsilon : \tau.$$

The *structural 1-equations* S^1 are given by:

Structural equations

The *structural 0-equations* S^0 are given by:

$$(AB)C : \omega = A(BC) : \omega, \qquad \varepsilon A : \omega = A : \omega = A\varepsilon : \omega,$$
$$f(AB) : \tau = f(A)f(B) : \tau, \qquad f(\varepsilon) : \tau = \varepsilon : \tau.$$

The *structural 1-equations* S^1 are given by:

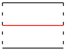
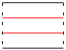
- the structural identities of monoidal theories, both internally and externally,

Structural equations

The *structural 0-equations* S^0 are given by:

$$(AB)C : \omega = A(BC) : \omega, \qquad \varepsilon A : \omega = A : \omega = A\varepsilon : \omega,$$
$$f(AB) : \tau = f(A)f(B) : \tau, \qquad f(\varepsilon) : \tau = \varepsilon : \tau.$$

The *structural 1-equations* S^1 are given by:

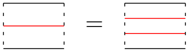
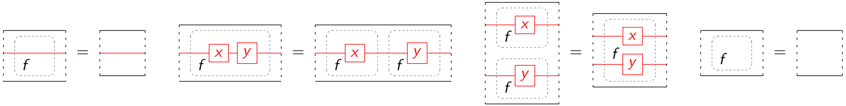
- the structural identities of monoidal theories, both internally and externally,
-  =  $: (AB : \omega \mid AB : \omega)$,

Structural equations

The *structural 0-equations* S^0 are given by:

$$(AB)C : \omega = A(BC) : \omega, \qquad \varepsilon A : \omega = A : \omega = A\varepsilon : \omega,$$
$$f(AB) : \tau = f(A)f(B) : \tau, \qquad f(\varepsilon) : \tau = \varepsilon : \tau.$$

The *structural 1-equations* S^1 are given by:

- the structural identities of monoidal theories, both internally and externally,
-  $: (AB : \omega \mid AB : \omega)$,
- 

.

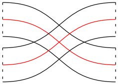
Additional terms

Symmetry terms:

Additional terms

Symmetry terms:

$$A : \omega \qquad B : \tau$$ (swap)

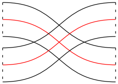


$$: (A : \omega, B : \tau \mid B : \tau, A : \omega)$$

Additional terms

Symmetry terms:

$$\frac{A : \omega \qquad B : \tau}{\quad} \text{(swap)}$$



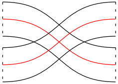
$: (A : \omega, B : \tau \mid B : \tau, A : \omega)$

Opfibrational terms:

Additional terms

Symmetry terms:

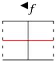
$$A : \omega \qquad B : \tau \quad \text{(swap)}$$



$: (A : \omega, B : \tau \mid B : \tau, A : \omega)$

Opfibrational terms:

$$A : \omega \qquad f \in \mathcal{F}(\omega, \tau) \quad \text{(ext-gen)}$$

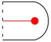


$: (A : \omega \mid f(A) : \tau)$

$$\omega \in \Omega \quad \text{(monoid-unit)}$$

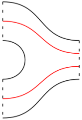
$: (\varepsilon : \varepsilon \mid \varepsilon : \omega)$

$$A : \omega \quad \text{(diag-counit)}$$



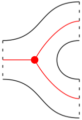
$: (A : \omega \mid \varepsilon : \varepsilon)$

$$A : \omega \qquad B : \omega \quad \text{(monoid)}$$



$: (A : \omega, B : \omega \mid AB : \omega)$

$$A : \omega \quad \text{(diag)}$$



$: (A : \omega \mid A : \omega, A : \omega)$

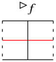
Additional terms

Fibrational terms:

Additional terms

Fibrational terms:

$A : \omega \qquad f \in \mathcal{F}(\omega, \tau)$ (ext-gen-op)

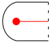


$: (f(A) : \tau \mid A : \omega)$

$\omega \in \Omega$ (counit)

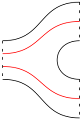
$: (\varepsilon : \omega \mid \varepsilon : \varepsilon)$

$A : \omega$ (codiag-unit)



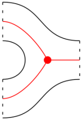
$: (\varepsilon : \varepsilon \mid A : \omega)$

$A : \omega \qquad B : \omega$ (comonoid)



$: (AB : \omega \mid A : \omega, B : \omega)$

$A : \omega$ (codiag)

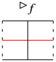


$: (A : \omega, A : \omega \mid A : \omega)$

Additional terms

Fibrational terms:

$$\frac{A : \omega \qquad f \in \mathcal{F}(\omega, \tau)}{\quad}\ \text{(ext-gen-op)}$$

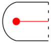


$: (f(A) : \tau \mid A : \omega)$

$$\frac{\omega \in \Omega}{\quad}\ \text{(counit)}$$

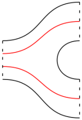
$: (\varepsilon : \omega \mid \varepsilon : \varepsilon)$

$$\frac{A : \omega}{\quad}\ \text{(codiag-unit)}$$



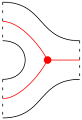
$: (\varepsilon : \varepsilon \mid A : \omega)$

$$\frac{A : \omega \qquad B : \omega}{\quad}\ \text{(comonoid)}$$



$: (AB : \omega \mid A : \omega, B : \omega)$

$$\frac{A : \omega}{\quad}\ \text{(codiag)}$$



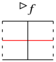
$: (A : \omega, A : \omega \mid A : \omega)$

Structural 2-cells:

Additional terms

Fibrational terms:

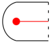
$$\frac{A : \omega \qquad f \in \mathcal{F}(\omega, \tau)}{\quad}\ \text{(ext-gen-op)}$$

 $: (f(A) : \tau \mid A : \omega)$

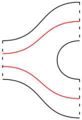
$$\frac{\omega \in \Omega}{\quad}\ \text{(counit)}$$

 $: (\varepsilon : \omega \mid \varepsilon : \varepsilon)$

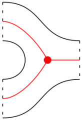
$$\frac{A : \omega}{\quad}\ \text{(codiag-unit)}$$

 $: (\varepsilon : \varepsilon \mid A : \omega)$

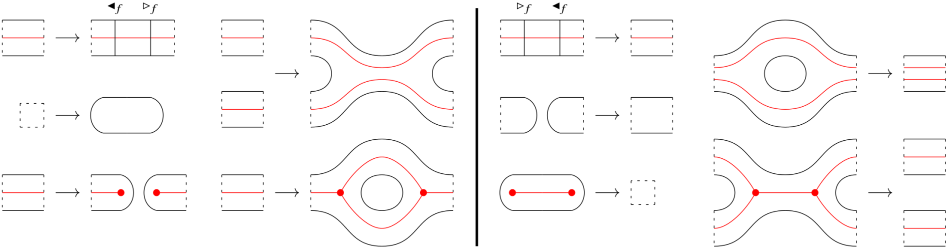
$$\frac{A : \omega \qquad B : \omega}{\quad}\ \text{(comonoid)}$$

 $: (AB : \omega \mid A : \omega, B : \omega)$

$$\frac{A : \omega}{\quad}\ \text{(codiag)}$$

 $: (A : \omega, A : \omega \mid A : \omega)$

Structural 2-cells:



Additional equations

- For symmetry terms:

Additional equations

- For symmetry terms: symmetry and naturality.

Additional equations

- For symmetry terms: symmetry and naturality.
- For opfibrational terms (**opfibrational theory**):

Additional equations

- For symmetry terms: symmetry and naturality.
- For opfibrational terms (**opfibrational theory**):
  - the equations of uniform comonoids for the terms (`diag`) and (`diag-counit`),

Additional equations

- For symmetry terms: symmetry and naturality.
- For opfibrational terms (**opfibrational theory**):
  - the equations of uniform comonoids for the terms `(diag)` and `(diag-counit)`,
  - the equations on the next slide.

Additional equations

- For symmetry terms: symmetry and naturality.
- For opfibrational terms (**opfibrational theory**):
  - the equations of uniform comonoids for the terms `(diag)` and `(diag-counit)`,
  - the equations on the next slide.
- For fibrational terms (**fibrational theory**):

Additional equations

- For symmetry terms: symmetry and naturality.
- For opfibrational terms (**opfibrational theory**):
  - the equations of uniform comonoids for the terms `(diag)` and `(diag-counit)`,
  - the equations on the next slide.
- For fibrational terms (**fibrational theory**): all the equations for opfibrational terms, reversed.

Additional equations

- For symmetry terms: symmetry and naturality.
- For opfibrational terms (**opfibrational theory**):
  - the equations of uniform comonoids for the terms (`diag`) and (`diag-counit`),
  - the equations on the next slide.
- For fibrational terms (**fibrational theory**): all the equations for opfibrational terms, reversed.
- For all the terms we've seen (**deflational theory**):

Additional equations

- For symmetry terms: symmetry and naturality.
- For opfibrational terms (**opfibrational theory**):
  - the equations of uniform comonoids for the terms `(diag)` and `(diag-counit)`,
  - the equations on the next slide.
- For fibrational terms (**fibrational theory**): all the equations for opfibrational terms, reversed.
- For all the terms we've seen (**deflational theory**):
  - The opfibrational and fibrational theories for respective terms,

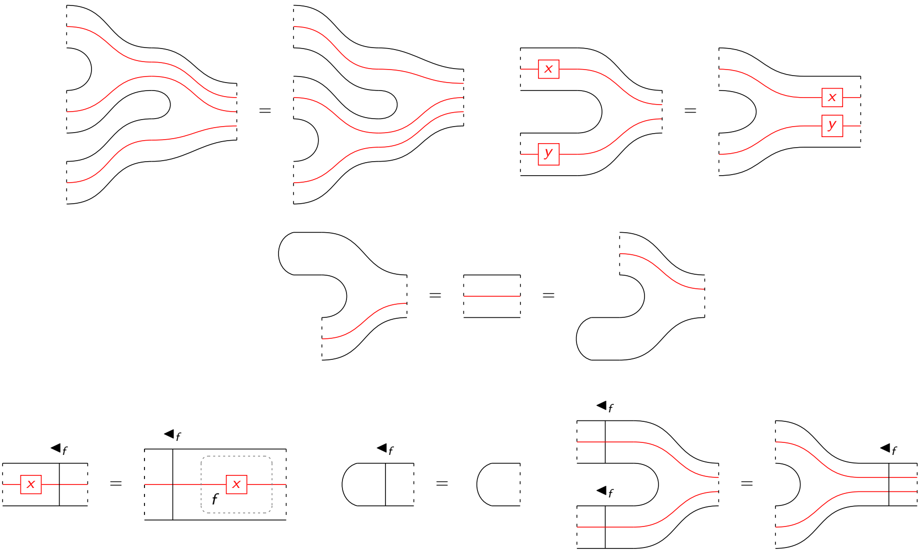
Additional equations

- For symmetry terms: symmetry and naturality.
- For opfibrational terms (**opfibrational theory**):
  - the equations of uniform comonoids for the terms `(diag)` and `(diag-counit)`,
  - the equations on the next slide.
- For fibrational terms (**fibrational theory**): all the equations for opfibrational terms, reversed.
- For all the terms we've seen (**deflational theory**):
  - The opfibrational and fibrational theories for respective terms,
  - The equations expressing the structural 2-cells as unit-counit pairs, with sections for counits

Additional equations

- For symmetry terms: symmetry and naturality.
- For opfibrational terms (**opfibrational theory**):
  - the equations of uniform comonoids for the terms (`diag`) and (`diag-counit`),
  - the equations on the next slide.
- For fibrational terms (**fibrational theory**): all the equations for opfibrational terms, reversed.
- For all the terms we've seen (**deflational theory**):
  - The opfibrational and fibrational theories for respective terms,
  - The equations expressing the structural 2-cells as unit-counit pairs, with sections for counits
  - Compatibility of 2-cells with sliding.

Additional equations



Semantics

theory	semantics
monoidal theory	(strict) monoidal category
opfibrational theory	split opfibration with indexed monoids
fibrational theory	split fibration with indexed comonoids
deflational theory	split monoidal deflation

Functor boxes

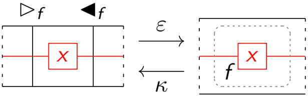
The usual notation for functor boxes decomposes as follows:

Functor boxes

The usual notation for functor boxes decomposes as follows:

Proposition

In a deflational theory, there are bidirectional 2-cells



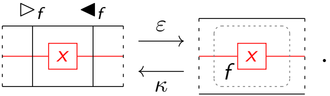
.

Functor boxes

The usual notation for functor boxes decomposes as follows:

Proposition

In a deflational theory, there are bidirectional 2-cells



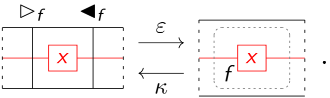
The term on the left-hand side is called a *cowindow*.

Functor boxes

The usual notation for functor boxes decomposes as follows:

Proposition

In a deflational theory, there are bidirectional 2-cells



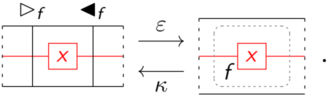
The term on the left-hand side is called a *cowindow*. The dual situation is called a *window*:

Functor boxes

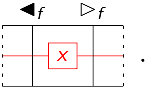
The usual notation for functor boxes decomposes as follows:

Proposition

In a deflational theory, there are bidirectional 2-cells



The term on the left-hand side is called a *cowindow*. The dual situation is called a *window*:

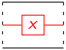


Functor coboxes

A *cobox* is a term generated by the following rule:

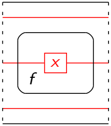
Functor coboxes

A *cobox* is a term generated by the following rule:

$f \in \mathcal{F}(\omega, \tau)$ $\qquad$ $A, B, C, D : \omega$ $\qquad$ 

$: (f(A) : \tau \mid f(B) : \tau)$

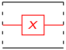
(cobox)



$: (CAD : \omega \mid CBD : \omega)$

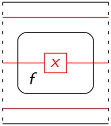
Functor coboxes

A *cobox* is a term generated by the following rule:

$f \in \mathcal{F}(\omega, \tau)$ $\quad$ $A, B, C, D : \omega$ $\quad$ 

$: (f(A) : \tau \mid f(B) : \tau)$

(cobox)

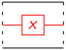


$: (CAD : \omega \mid CBD : \omega)$

Subject to the following equations:

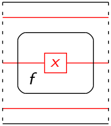
Functor coboxes

A *cobox* is a term generated by the following rule:

$f \in \mathcal{F}(\omega, \tau) \qquad A, B, C, D : \omega$ 

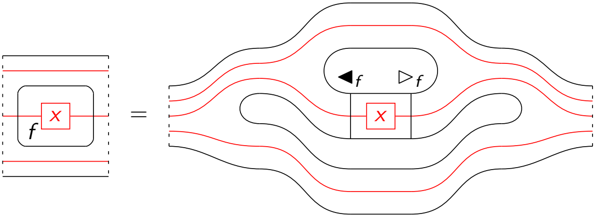
$: (f(A) : \tau \mid f(B) : \tau)$

(cobox)



$: (CAD : \omega \mid CBD : \omega)$

Subject to the following equations:



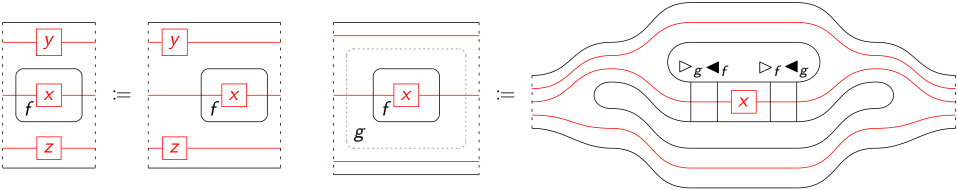
Note: NOT an internal term! However...

Functor coboxes

We make the following identifications:

Functor coboxes

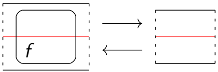
We make the following identifications:



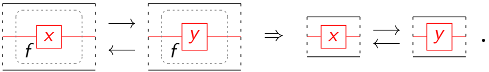
Functor coboxes

Proposition

In a deflational theory, there are bidirectional 2-cells (3) *if and only if for all terms x and y with the sort $(A : \omega \mid B : \omega)$, the existence of 2-cells on the right of* (4) *implies the existence of 2-cells on the left of* (4)*:*



(3)

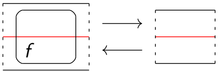


(4)

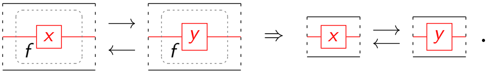
Functor coboxes

Proposition

In a deflational theory, there are bidirectional 2-cells (3) *if and only if for all terms x and y with the sort $(A : \omega \mid B : \omega)$, the existence of 2-cells on the right of* (4) *implies the existence of 2-cells on the left of* (4)*:*



(3)



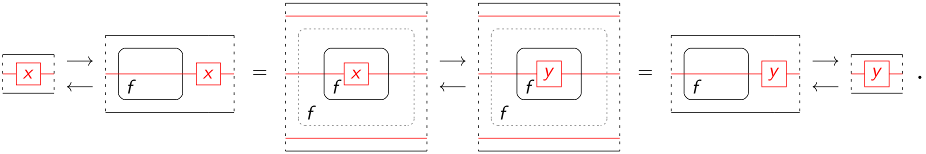
(4)

Moreover, if (3) *is an equality, then the implication* (4) *holds also when the bidirectional 2-cells on both sides are replaced with equalities.*

Functor coboxes

Proof.

Suppose the 2-cells (3) exist, and let x and y be terms such that $fx \rightleftarrows fy$. Then:

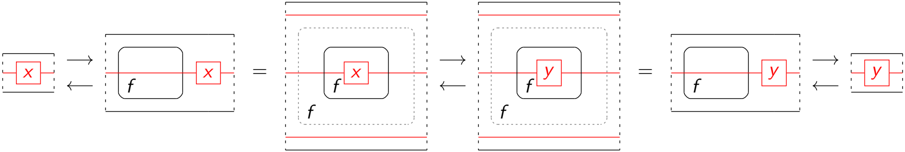


The same argument shows the case with equalities.

Functor coboxes

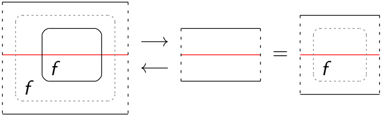
Proof.

Suppose the 2-cells (3) exist, and let x and y be terms such that $fx \rightleftarrows fy$. Then:



.

The same argument shows the case with equalities.
Conversely, suppose that the implication (4) holds. We use the following to obtain the required 2-cells:



. □

Digital circuits: Signature

Define the layered signature $(\mathbb{N}_+, \mathcal{F}, \Sigma_n)$ as follows:

Digital circuits: Signature

Define the layered signature $(\mathbb{N}_+, \mathcal{F}, \Sigma_n)$ as follows:

- for each $n \in \mathbb{N}_+$, the set $\mathcal{F}(n, 1)$ has exactly one element,

Digital circuits: Signature

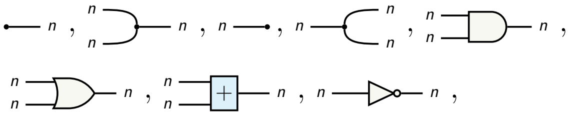
Define the layered signature $(\mathbb{N}_+, \mathcal{F}, \Sigma_n)$ as follows:

- for each $n \in \mathbb{N}_+$, the set $\mathcal{F}(n, 1)$ has exactly one element,
- for $n \neq 1$, the monoidal signature Σ_n has a single colour n, and the monoidal generators:

Digital circuits: Signature

Define the layered signature $(\mathbb{N}_+, \mathcal{F}, \Sigma_n)$ as follows:

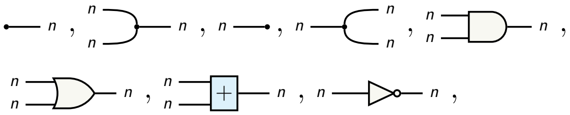
- for each $n \in \mathbb{N}_+$, the set $\mathcal{F}(n, 1)$ has exactly one element,
- for $n \neq 1$, the monoidal signature Σ_n has a single colour n, and the monoidal generators:

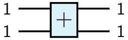


Digital circuits: Signature

Define the layered signature $(\mathbb{N}_+, \mathcal{F}, \Sigma_n)$ as follows:

- for each $n \in \mathbb{N}_+$, the set $\mathcal{F}(n, 1)$ has exactly one element,
- for $n \neq 1$, the monoidal signature Σ_n has a single colour n, and the monoidal generators:

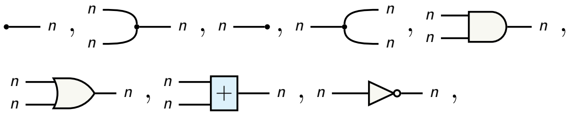


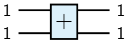
- for $n = 1$, the monoidal signature Σ_1 is as above, except that we replace the adder with  .

Digital circuits: Signature

Define the layered signature $(\mathbb{N}_+, \mathcal{F}, \Sigma_n)$ as follows:

- for each $n \in \mathbb{N}_+$, the set $\mathcal{F}(n, 1)$ has exactly one element,
- for $n \neq 1$, the monoidal signature Σ_n has a single colour n, and the monoidal generators:

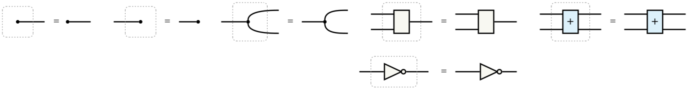


- for $n = 1$, the monoidal signature Σ_1 is as above, except that we replace the adder with  .

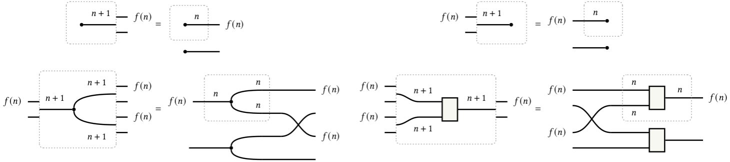
This signature is called *simple arithmetic circuits*.

Digital circuits: Theory

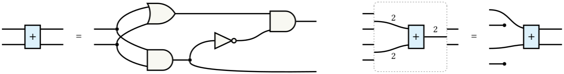
$f \in \mathcal{F}(1,1): \quad f(1) = 1$



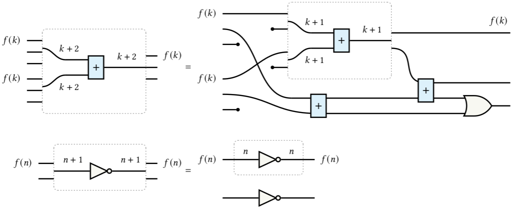
$f \in \mathcal{F}(n+1,1): \quad f(n+1) = f(n)1$



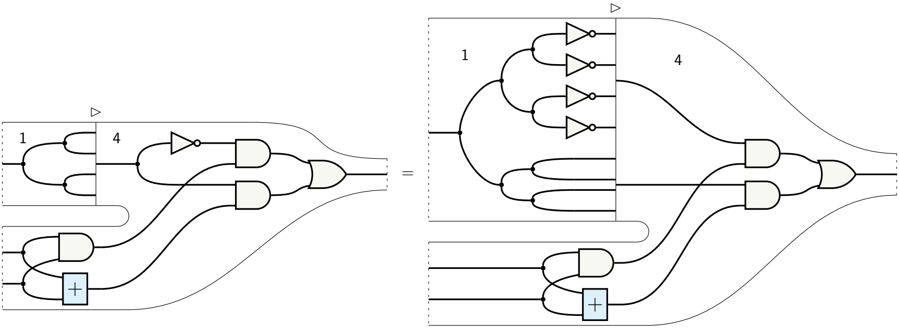
$n = 1:$



$n = k + 1$ for $k \geq 1:$



Digital circuits: A simple arithmetic logic unit



Electrical circuits: Signature

Define the layered signature with the following shape:

Electrical circuits: Signature

Define the layered signature with the following shape:

$$\begin{array}{ccc} \mathbf{Bip} & \hookrightarrow & \mathbf{ECirc} \\ B \downarrow & & \downarrow \mathcal{I} \\ \mathbf{Imp} & \xrightarrow{W} & \mathbf{GAA}_{\mathbb{R}(x)} \end{array} \quad .$$

Electrical circuits: Signature

Define the layered signature with the following shape:

$$\begin{array}{ccc} \mathbf{Bip} & \hookrightarrow & \mathbf{ECirc} \\ B \downarrow & & \downarrow \mathcal{I} \\ \mathbf{Imp} & \xrightarrow{W} & \mathbf{GAA}_{\mathbb{R}(x)} \end{array} .$$

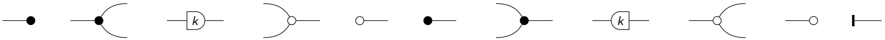
$\mathbf{GAA}_{\mathbb{R}(x)}$ is *graphical affine algebra*:

Electrical circuits: Signature

Define the layered signature with the following shape:

$$\begin{array}{ccc} \mathbf{Bip} & \hookrightarrow & \mathbf{ECirc} \\ B \downarrow & & \downarrow \mathcal{I} \\ \mathbf{Imp} & \xrightarrow{W} & \mathbf{GAA}_{\mathbb{R}(x)} \end{array} \quad .$$

$\mathbf{GAA}_{\mathbb{R}(x)}$ is *graphical affine algebra*:



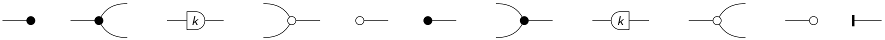
.

Electrical circuits: Signature

Define the layered signature with the following shape:

$$\begin{array}{ccc} \mathbf{Bip} & \hookrightarrow & \mathbf{ECirc} \\ B \downarrow & & \downarrow \mathcal{I} \\ \mathbf{Imp} & \xrightarrow{W} & \mathbf{GAA}_{\mathbb{R}(x)} \end{array} \quad .$$

$\mathbf{GAA}_{\mathbb{R}(x)}$ is *graphical affine algebra*:



.

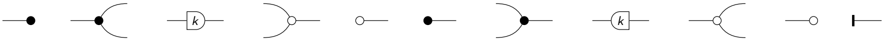
The generators in **Imp** are given by all the terms in **GAA** with sort $(1, 1)$.

Electrical circuits: Signature

Define the layered signature with the following shape:

$$\begin{array}{ccc} \mathbf{Bip} & \hookrightarrow & \mathbf{ECirc} \\ {\scriptstyle B}\downarrow & & \downarrow{\scriptstyle \mathcal{I}} \\ \mathbf{Imp} & \xrightarrow{W} & \mathbf{GAA}_{\mathbb{R}(x)} \end{array} \quad .$$

$\mathbf{GAA}_{\mathbb{R}(x)}$ is *graphical affine algebra*:



.

The generators in **Imp** are given by all the terms in **GAA** with sort $(1,1)$. The generators in **Bip** are given by the *bipoles* (5). The generators in **ECirc** are given by the bipoles (5) and (6).

(5)

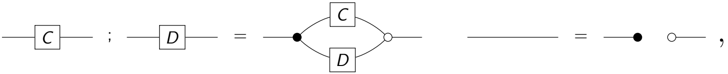

(6)

Electrical circuits: Theory

- in **GAA**, the equations of the graphical affine algebra hold: they axiomatise affine relations,

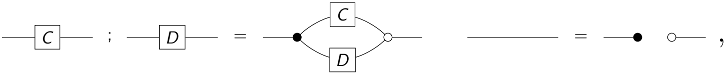
Electrical circuits: Theory

- in **GAA**, the equations of the graphical affine algebra hold: they axiomatise affine relations,
- in **Imp**, the equations of **GAA** hold inside the generators, together with



Electrical circuits: Theory

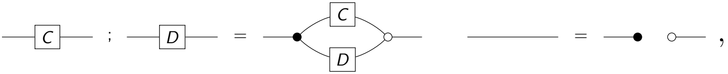
- in **GAA**, the equations of the graphical affine algebra hold: they axiomatise affine relations,
- in **Imp**, the equations of **GAA** hold inside the generators, together with



- on the colours, the functors $\mathcal{I}$: **ECirc** $\to$ **GAA** and W : **Imp** $\to$ **GAA** are defined by $\bullet \mapsto \bullet\bullet$, so that the resulting map on objects is $n \mapsto 2n$,

Electrical circuits: Theory

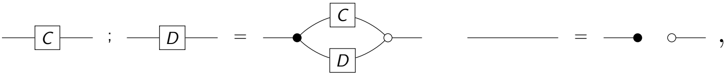
- in **GAA**, the equations of the graphical affine algebra hold: they axiomatise affine relations,
- in **Imp**, the equations of **GAA** hold inside the generators, together with



- on the colours, the functors $\mathcal{I}$: **ECirc** $\to$ **GAA** and W : **Imp** $\to$ **GAA** are defined by $\bullet \mapsto \bullet\bullet$, so that the resulting map on objects is $n \mapsto 2n$,
- the functor B : **Bip** $\to$ **Imp** is identity on colours,

Electrical circuits: Theory

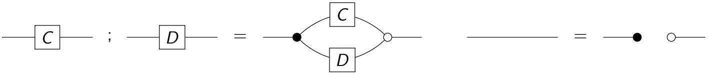
- in **GAA**, the equations of the graphical affine algebra hold: they axiomatise affine relations,
- in **Imp**, the equations of **GAA** hold inside the generators, together with



- on the colours, the functors $\mathcal{I}$: **ECirc** $\to$ **GAA** and W : **Imp** $\to$ **GAA** are defined by $\bullet \mapsto \bullet\bullet$, so that the resulting map on objects is $n \mapsto 2n$,
- the functor B : **Bip** $\to$ **Imp** is identity on colours,
- the functor **Bip** $\hookrightarrow$ **ECirc** is the embedding,

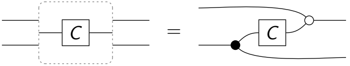
Electrical circuits: Theory

- in **GAA**, the equations of the graphical affine algebra hold: they axiomatise affine relations,
- in **Imp**, the equations of **GAA** hold inside the generators, together with



,

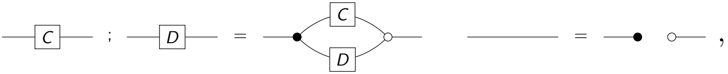
- on the colours, the functors $\mathcal{I}$: **ECirc** $\to$ **GAA** and W : **Imp** $\to$ **GAA** are defined by $\bullet \mapsto \bullet\bullet$, so that the resulting map on objects is $n \mapsto 2n$,
- the functor B : **Bip** $\to$ **Imp** is identity on colours,
- the functor **Bip** $\hookrightarrow$ **ECirc** is the embedding,
- the functor W : **Imp** $\to$ **GAA** is defined by the equation



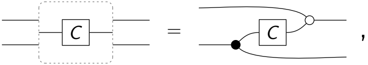
,

Electrical circuits: Theory

- in **GAA**, the equations of the graphical affine algebra hold: they axiomatise affine relations,
- in **Imp**, the equations of **GAA** hold inside the generators, together with

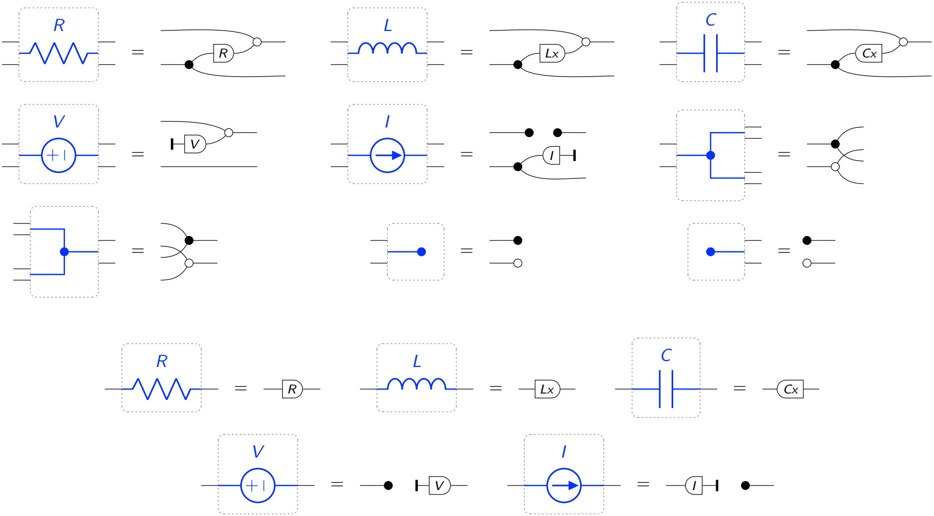


- on the colours, the functors $\mathcal{I}$: **ECirc** $\to$ **GAA** and W : **Imp** $\to$ **GAA** are defined by $\bullet \mapsto \bullet\bullet$, so that the resulting map on objects is $n \mapsto 2n$,
- the functor B : **Bip** $\to$ **Imp** is identity on colours,
- the functor **Bip** $\hookrightarrow$ **ECirc** is the embedding,
- the functor W : **Imp** $\to$ **GAA** is defined by the equation



- the functors $\mathcal{I}$: **ECirc** $\to$ **GAA** and B : **Bip** $\to$ **Imp** are defined on the next slide.

Electrical circuits: Theory



Electrical circuits: Impedance box

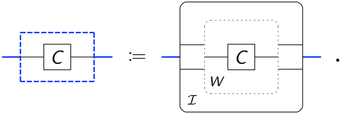
Within this layered theory, we recover the *impedance box* of Boisseau & Sobociński as a box-cobox combination, rather than an additional notational device.

Electrical circuits: Impedance box

Within this layered theory, we recover the *impedance box* of Boisseau & Sobociński as a box-cobox combination, rather than an additional notational device.

Definition

Let C be a generator of **Imp**. Define the *impedance box* as:



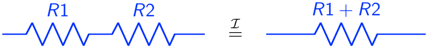
Electrical circuits: Impedance box

This can be used e.g. to derive the rule for composing two resistors in **ECirc**: 

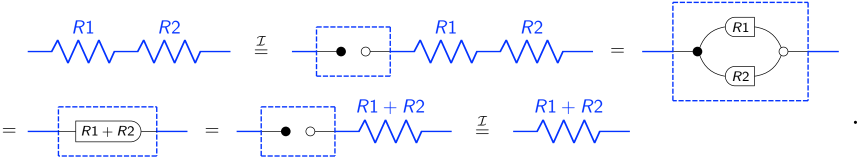
$\stackrel{\mathcal{I}}{=}$ 

(Proposition 3 of Boisseau & Sobociński):

Electrical circuits: Impedance box

This can be used e.g. to derive the rule for composing two resistors in **ECirc**: 

(Proposition 3 of Boisseau & Sobociński):



Chemical reactions: Signature

Define the layered signature with the shape $\textbf{Name} \longrightarrow \textbf{Disc} \longleftarrow \textbf{Scheme}$:

Chemical reactions: Signature

Define the layered signature with the shape $\mathbf{Name} \longrightarrow \mathbf{Disc} \longleftarrow \mathbf{Scheme}$:

- there are no generators in **Name**, and the colours are $\{glucose, ATP, glucose\text{-}6\text{-}phosphate, ADP, hydrogen\ ion\}$

Chemical reactions: Signature

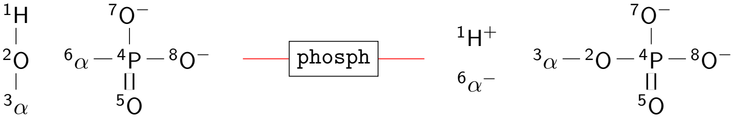
Define the layered signature with the shape $\mathbf{Name} \longrightarrow \mathbf{Disc} \longleftarrow \mathbf{Scheme}$:

- there are no generators in **Name**, and the colours are $\{$*glucose*, *ATP*, *glucose-6-phosphate*, *ADP*, *hydrogen ion*$\}$
- the colours of **Disc** and **Scheme** are the chemical graphs,

Chemical reactions: Signature

Define the layered signature with the shape **Name** $\longrightarrow$ **Disc** $\longleftarrow$ **Scheme** :

- there are no generators in **Name**, and the colours are $\{$*glucose*, *ATP*, *glucose-6-phosphate*, *ADP*, *hydrogen ion*$\}$
- the colours of **Disc** and **Scheme** are the chemical graphs,
- the generators of **Scheme** are given by the family

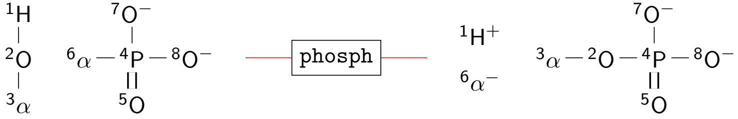


,

Chemical reactions: Signature

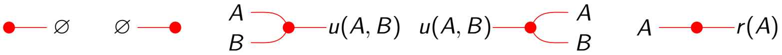
Define the layered signature with the shape $\textbf{Name} \longrightarrow \textbf{Disc} \longleftarrow \textbf{Scheme}$:

- there are no generators in **Name**, and the colours are $\{glucose, ATP, glucose\text{-}6\text{-}phosphate, ADP, hydrogen\ ion\}$
- the colours of **Disc** and **Scheme** are the chemical graphs,
- the generators of **Scheme** are given by the family



,

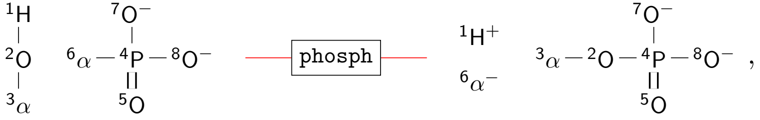
- the *structural generators* of **Disc** are the symmetry and:



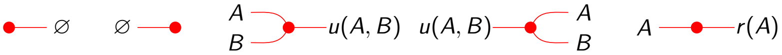
Chemical reactions: Signature

Define the layered signature with the shape **Name** $\longrightarrow$ **Disc** $\longleftarrow$ **Scheme** :

- there are no generators in **Name**, and the colours are $\{$*glucose*, *ATP*, *glucose-6-phosphate*, *ADP*, *hydrogen ion*$\}$
- the colours of **Disc** and **Scheme** are the chemical graphs,
- the generators of **Scheme** are given by the family



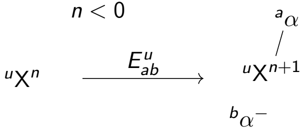
- the *structural generators* of **Disc** are the symmetry and:



- the remaining generators in **Disc** are given by the *disconnection rules*

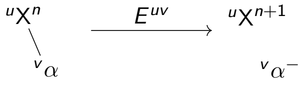
Chemical reactions: Disconnection rules

$X \in \mathbf{At} \setminus \{\alpha\}$
$n < 0$



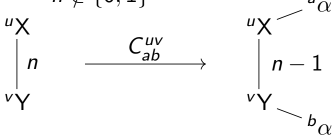
Electron detachment (negative charge)

$X \in \mathbf{At} \setminus \{\alpha\}$
$n \geq 0$



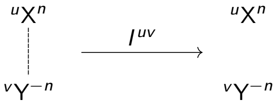
Electron detachment (nonnegative charge)

$X, Y \in \mathbf{At} \setminus \{\alpha\}$
$n \notin \{0, \mathtt{i}\}$



Covalent bond breaking

$n \neq 0$



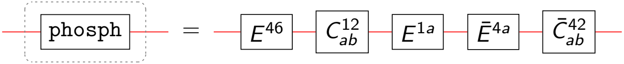
Ionic bond breaking

Chemical reactions: Theory

- the structural generators are isomorphisms in **Disc**,

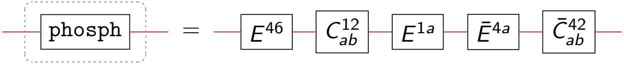
Chemical reactions: Theory

- the structural generators are isomorphisms in **Disc**,
- the functor **Scheme** $\rightarrow$ **Disc** is given by:



Chemical reactions: Theory

- the structural generators are isomorphisms in **Disc**,
- the functor **Scheme** $\to$ **Disc** is given by:



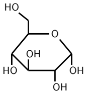
- the functor **Name** $\to$ **Disc** is defined below:

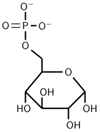
Chemical reactions: Theory

- the structural generators are isomorphisms in **Disc**,
- the functor **Scheme** $\to$ **Disc** is given by:

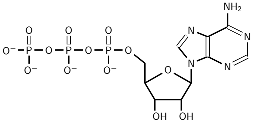
$$\boxed{\texttt{phosph}} = \boxed{E^{46}}\;\boxed{C^{12}_{ab}}\;\boxed{E^{1a}}\;\boxed{\bar{E}^{4a}}\;\boxed{\bar{C}^{42}_{ab}}$$

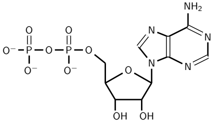
- the functor **Name** $\to$ **Disc** is defined below:

glucose $\mapsto$ 

glucose-6-phosphate $\mapsto$ 

hydrogen ion $\mapsto$ H^+

ATP $\mapsto$ 

ADP $\mapsto$ 

Chemical reactions: Reaction

The reaction below is now derivable as a term in the layered theory (but not in any of the constituting monoidal theories):

Chemical reactions: Reaction

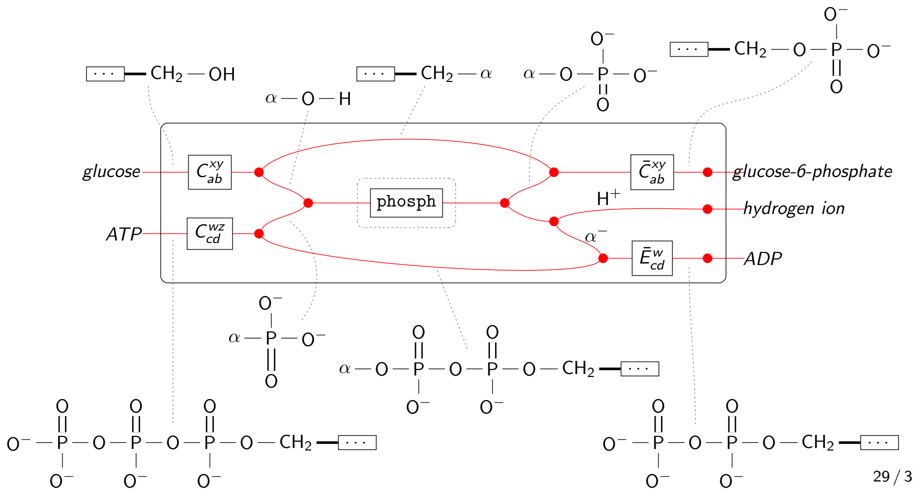
The reaction below is now derivable as a term in the layered theory (but not in any of the constituting monoidal theories):

$$\textit{glucose} + \textit{ATP} \longrightarrow \textit{glucose-6-phosphate} + \textit{ADP} + \textit{hydrogen ion}.$$

Chemical reactions: Reaction

The reaction below is now derivable as a term in the layered theory (but not in any of the constituting monoidal theories):

$$\textit{glucose} + \textit{ATP} \longrightarrow \textit{glucose-6-phosphate} + \textit{ADP} + \textit{hydrogen ion}.$$



Related work

- Internal string diagrams for TQFTs and profunctors:

Related work

- Internal string diagrams for TQFTs and profunctors:
  - Bruce Bartlett, Christopher L. Douglas, Christopher J. Schommer-Pries, and Jamie Vicary. 2015. *Modular categories as representations of the 3-dimensional bordism 2-category*.
  - Nick Hu. 2019. *External traced monoidal categories*. Master's thesis.
  - Nick Hu and Jamie Vicary. 2021. *Traced Monoidal Categories as Algebraic Structures in Prof*.

Related work

- Internal string diagrams for TQFTs and profunctors:
  - Bruce Bartlett, Christopher L. Douglas, Christopher J. Schommer-Pries, and Jamie Vicary. 2015. *Modular categories as representations of the 3-dimensional bordism 2-category*.
  - Nick Hu. 2019. *External traced monoidal categories*. Master's thesis.
  - Nick Hu and Jamie Vicary. 2021. *Traced Monoidal Categories as Algebraic Structures in Prof*.
- Open diagrams for coend calculus and string diagram collages:

Related work

- Internal string diagrams for TQFTs and profunctors:
  - Bruce Bartlett, Christopher L. Douglas, Christopher J. Schommer-Pries, and Jamie Vicary. 2015. *Modular categories as representations of the 3-dimensional bordism 2-category*.
  - Nick Hu. 2019. *External traced monoidal categories*. Master's thesis.
  - Nick Hu and Jamie Vicary. 2021. *Traced Monoidal Categories as Algebraic Structures in Prof*.
- Open diagrams for coend calculus and string diagram collages:
  - Mario Román. 2021. *Open Diagrams via Coend Calculus*.
  - Dylan Braithwaite and Mario Román. 2023. *Collages of String Diagrams*.

Related work

- Internal string diagrams for TQFTs and profunctors:
    - Bruce Bartlett, Christopher L. Douglas, Christopher J. Schommer-Pries, and Jamie Vicary. 2015. *Modular categories as representations of the 3-dimensional bordism 2-category*.
    - Nick Hu. 2019. *External traced monoidal categories*. Master's thesis.
    - Nick Hu and Jamie Vicary. 2021. *Traced Monoidal Categories as Algebraic Structures in Prof*.
- Open diagrams for coend calculus and string diagram collages:
    - Mario Román. 2021. *Open Diagrams via Coend Calculus*.
    - Dylan Braithwaite and Mario Román. 2023. *Collages of String Diagrams*.
- Holey string diagrams:

Related work

- Internal string diagrams for TQFTs and profunctors:
  - Bruce Bartlett, Christopher L. Douglas, Christopher J. Schommer-Pries, and Jamie Vicary. 2015. *Modular categories as representations of the 3-dimensional bordism 2-category*.
  - Nick Hu. 2019. *External traced monoidal categories*. Master's thesis.
  - Nick Hu and Jamie Vicary. 2021. *Traced Monoidal Categories as Algebraic Structures in Prof*.
- Open diagrams for coend calculus and string diagram collages:
  - Mario Román. 2021. *Open Diagrams via Coend Calculus*.
  - Dylan Braithwaite and Mario Román. 2023. *Collages of String Diagrams*.
- Holey string diagrams:
  - Matt Earnshaw and Mario Román. 2024. *Context-Free Languages of String Diagrams*.
  - James Hefford and Cole Comfort. 2023. *Coend Optics for Quantum Combs*.

References

- L. *Layered monoidal theories*. 2025. PhD thesis.
- L. and Fabio Zanasi. 2026. *Layered monoidal theories I: Diagrammatic algebra and applications*
- L. and Fabio Zanasi. 2026. *Layered monoidal theories II: Fibrational semantics*

Thank you for your attention!