Computer Science Logic 2018

Birmingham, United Kingdom
4–7 September

Rule Algebras for Adhesive Categories

Nicolas Behr (IRIF, Université Paris 7, France)

Joint work with **Paweł Sobociński** (ECS, University of Southampton, UK)

MARIE CURIE ACTIONS

INSTITUT
DE RECHERCHE
EN INFORMATIQUE
FONDAMENTALE

université PARIS DIDEROT PARIS 7

Motivation

- **formal power series:**

$$f(x) \in \mathbb{R}[[x]] \quad :\Leftrightarrow \quad f(x) = \sum_{n \geqslant 0} f_n\, x^n \quad (\text{with } f_n \in \mathbb{R} \text{ for all } n \in \mathbb{Z}_{\geqslant 0})$$

- **two natural linear operators:** $\hat{x}$ and ∂_x

$$\hat{x} : \mathbb{R}[[x]] \to \mathbb{R}[[x]] : x^n \mapsto x^{n+1}\,, \quad \partial_x : \mathbb{R}[[x]] \to \mathbb{R}[[x]] : x^n \mapsto \begin{cases} 0 & \text{if } n = 0 \\ n\, x^{n-1} & \text{if } n > 0 \end{cases}$$

Motivation

- **formal power series:**

$$f(x) \in \mathbb{R}[[x]] \quad :\Leftrightarrow \quad f(x) = \sum_{n\geqslant 0} f_n\, x^n \quad \text{(with } f_n \in \mathbb{R} \text{ for all } n \in \mathbb{Z}_{\geqslant 0}\text{)}$$

- **two natural linear operators:** $\hat{x}$ and ∂_x

$$\hat{x} : \mathbb{R}[[x]] \to \mathbb{R}[[x]] : x^n \mapsto x^{n+1}\,, \quad \partial_x : \mathbb{R}[[x]] \to \mathbb{R}[[x]] : x^n \mapsto \begin{cases} 0 & \text{if } n = 0 \\ n\, x^{n-1} & \text{if } n > 0 \end{cases}$$

- for all $p \in \mathbb{Z}_{>0}$ and $f(x), g(x) \in \mathbb{R}[[x]]$, “of course. . . ”

$$\partial_x^p\,(f(x)g(x)) = \sum_{k=0}^{p} \binom{p}{k} \left(\partial_x^k f(x)\right) \left(\partial_x^{p-k} g(x)\right)$$

Motivation

- **formal power series:**

$$f(x) \in \mathbb{R}[[x]] \quad :\Leftrightarrow \quad f(x) = \sum_{n \geqslant 0} f_n\, x^n \quad (\text{with } f_n \in \mathbb{R} \text{ for all } n \in \mathbb{Z}_{\geqslant 0})$$

- **two natural linear operators:** $\hat{x}$ and ∂_x

$$\hat{x} : \mathbb{R}[[x]] \to \mathbb{R}[[x]] : x^n \mapsto x^{n+1}\,, \quad \partial_x : \mathbb{R}[[x]] \to \mathbb{R}[[x]] : x^n \mapsto \begin{cases} 0 & \text{if } n = 0 \\ n\, x^{n-1} & \text{if } n > 0 \end{cases}$$

- for all $p \in \mathbb{Z}_{>0}$ and $f(x), g(x) \in \mathbb{R}[[x]]$, "of course…"

$$\partial_x^p\, (f(x)g(x)) = \sum_{k=0}^{p} \binom{p}{k} \left(\partial_x^k f(x)\right) \left(\partial_x^{p-k} g(x)\right)$$

$\Rightarrow$ **non-trivial "normal-ordering" type operator relation:** (for $p, q \in \mathbb{Z}_{\geqslant 0}$)

$$\partial_x^p \hat{x}^q = \sum_{k=0}^{min(p,q)} k! \binom{p}{k} \binom{q}{k} \hat{x}^{q-k} \partial_x^{p-k}$$

Motivation

- **non-trivial "normal-ordering" type operator relation:** (for $p, q \in \mathbb{Z}_{\geqslant 0}$)

$$\partial_x^p \hat{x}^q = \sum_{k=0}^{min(p,q)} k! \binom{p}{k} \binom{q}{k} \hat{x}^{q-k} \partial_x^{p-k} = \sum_{k=0}^{min(p,q)} \boxed{\frac{1}{k!} \left(\frac{p!}{(p-k)!}\right) \left(\frac{q!}{(q-k)!}\right)} \hat{x}^{q-k} \partial_x^{p-k}$$

# of ways to choose k objects from pools of p and q particles, disregarding order

$\Rightarrow$ **WHY?**

somewhat surprising answer:

Because $\hat{x}$ and ∂_x are the **canonical representations** of certain **rule algebra** elements associated to (discrete) **graph rewriting rules!**

Plan

The main construction:

- **Double-Pushout (DPO) rewriting** in **adhesive categories**
- From **DPO rewriting** to **DPO rule algebras**
- The framework: algebraic compositions, associativity, canonical representations. . .

Application examples:

- formal power series and the **Heisenberg-Weyl algebra**
- **combinatorics**
- **stochastic mechanics** of **continuous-time Markov chains**

Plan

The main construction:

- **Double-Pushout (DPO) rewriting** in **adhesive categories**
- From **DPO rewriting** to **DPO rule algebras**
- The framework: algebraic compositions, associativity, canonical representations...

Application examples:

- formal power series and the **Heisenberg-Weyl algebra**
- **combinatorics**
- **stochastic mechanics** of **continuous-time Markov chains**

Rule-algebra project "history" the first version of the rule algebra concept [1] (presented at **LiCS'16**, joint work with **V. Danos** and **I. Garnier**) had been based on **relation-algebraic** structures and covered the cases of rule algebras for rewriting of graphs; the new version extends this to rewriting of adhesive categories

[1] Nicolas Behr, Vincent Danos, and Ilias Garnier. "Stochastic mechanics of graph rewriting". In: *Proceedings of the 31st Annual ACM-IEEE Symposium on Logic in Computer Science (LICS 2016)* (2016), pp. 46–55

DPO rewriting and rule algebras

Background: adhesive categories

Adhesive categories (cf. [2], Def. 3.1)

A category $\mathbf{C}$ is said to be **adhesive** if

(i) $\mathbf{C}$ has pushouts along **monomorphisms**,

(ii) $\mathbf{C}$ has pullbacks, and if

(iii) pushouts along monomorphisms are **van Kampen (VK) squares**.

- **Examples [2]:**
  - $\mathbf{Set}$, the category of **(finite) sets and set functions**
  - $\mathbf{Graph}$, the category of **(finite) directed multigraphs and graph homomorphisms** (and also **colored/typed graphs, attributed graphs, hypergraphs,...**)
  - any **presheaf topos** and any **elementary topos** [3]
- **Note:** One might further generalize by considering **quasi-adhesive categories** (see [2], [4]).

[2] Stephen Lack and Paweł Sobociński. "Adhesive and quasiadhesive categories". In: *RAIRO-Theoretical Informatics and Applications* 39.3 (2005), pp. 511–545

[3] Stephen Lack and Paweł Sobociński. "Toposes are adhesive". In: *Graph Transformations, Third International Conference, (ICGT 2006)*. Vol. 4178. LNCS. Springer, 2006, pp. 184–198

[4] Richard Garner and Stephen Lack. "On the axioms for adhesive and quasiadhesive categories". In: *Theor. App. Categories* 27.3 (2012), pp. 27–46

Brief comments on abstract category-theoretical structures:

- **pushouts along monomorphisms** in the category $\mathbf{Set}$:

$$
\begin{tikzcd}
 & A \arrow[dl, hook'] \arrow[dr, hook] \arrow[dd, phantom, "\wedge" near end] & \\
B \arrow[dr, hook] & & C \arrow[dl, hook'] \\
 & D &
\end{tikzcd}
$$

Interpretation:
A – **intersection** of B and C in D
D – **union** of B and C along A

- **pullbacks along monomorphisms** in the category $\mathbf{Set}$:

$$
\begin{tikzcd}
 & A \arrow[dl, hook'] \arrow[dr, hook] \arrow[dd, phantom, "\vee" near start] & \\
B \arrow[dr, hook] & & C \arrow[dl, hook'] \\
 & D &
\end{tikzcd}
$$

Interpretation: A – **intersection** of B and C in D

Brief comments on abstract category-theoretical structures:

- **from** [5]**:**

Definition 1. A van Kampen square is a pushout diagram as in Fig 1 which satisfies the following condition:

– for any commutative cube, as illustrated, of which Fig 1 forms the bottom face and the back faces are pullbacks: the front faces are pullbacks iff the top face is a pushout.

$$
\begin{tikzcd}
 & & C' \arrow[dll, "m'"'] \arrow[dr, "f'"] \arrow[ddd, "c"'] & \\
A' \arrow[dr, "g'"'] \arrow[ddd, "a"'] & & & B' \arrow[dll, "n'"] \arrow[ddd, "b"] \\
 & D' \arrow[ddd, "d"] & & \\
 & & C \arrow[dll, "m"'] \arrow[dr, "f"] & \\
A \arrow[dr, "g"'] & & & B \arrow[dll, "n"] \\
 & D & &
\end{tikzcd}
$$

The following lemma shows that, in categories with pushouts and pullbacks, van Kampen squares paste together to give van Kampen squares.

[5] Stephen Lack and Paweł Sobociński. "Toposes are adhesive". In: *Graph Transformations, Third International Conference, (ICGT 2006)*. Vol. 4178. LNCS. Springer, 2006, pp. 184–198

Brief comments on abstract category-theoretical structures:

from [6]**:** in an adhesive category $\mathbf{C}$, for every object $Z \in ob(\mathbf{C})$ one may define the **subobject lattice** $\mathbf{Sub}(Z)$ via defining a preorder on the monomorphisms $x : X \hookrightarrow Z$ (with $x \prec y$ if there exists some monomorphism $i : X \hookrightarrow Y$ such that $y = i \circ x$)

Corollary 5.2 of [6]

The lattice $\mathbf{Sub}(Z)$ is **distributive**.

Proof: It is easy to verify that the front and back faces of the cube below are pullbacks. Because the bottom face is a pushout, we use adhesivity in order to conclude that the top face is a pushout, which in turn implies that $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$. □

$$\begin{tikzcd}
& & A\cap B\cap C \arrow[dll] \arrow[dr] \arrow[dd] & \\
A\cap B \arrow[dr] \arrow[ddd] & & & A\cap C \arrow[dll] \arrow[ddd] \\
& A\cap(B\cup C) \arrow[ddd] & & \\
& & B\cap C \arrow[dll] \arrow[dr] & \\
B \arrow[dr] & & & C \arrow[dll] \\
& B\cup C & &
\end{tikzcd}$$

[6] Stephen Lack and Paweł Sobociński. “Adhesive and quasiadhesive categories”. In: *RAIRO-Theoretical Informatics and Applications* 39.3 (2005), pp. 511–545

DPO rewriting

Double-Pushout (DPO) rewriting in an adhesive category (cf. [7], Def. 7.1)

A span p of morphisms

$$L \xleftarrow{l} K \xrightarrow{r} R$$

is called a **production**. p is said to be **left linear** if l is a monomorphism, and **linear** if both l and r are monomorphisms. We denote the **set of linear productions** by $Lin(\mathbf{C})$. We will also frequently make use of the alternative notation $L \xrightarrow{p} R$ where $p = (L \xleftarrow{l} K \xrightarrow{r} R) \in Lin(\mathbf{C})$.

Notes:

- A **homomorphism of productions** $p \to p'$ consists of arrows, $L \to L'$, $K \to K'$ and $R \to R'$, such that the obvious diagram commutes.
- A homomorphism is an **isomorphism** when all of its components are isomorphisms. **We do not distinguish between isomorphic productions.**

[7] Stephen Lack and Paweł Sobociński. "Adhesive and quasiadhesive categories". In: *RAIRO-Theoretical Informatics and Applications* 39.3 (2005), pp. 511–545

DPO rewriting

Admissible matches (cf. [8], Def. 7.2)

Given a production p, a **match** of p in an object $C \in ob(\mathbf{C})$ is a morphism $m : L \to C$. A match is said to satisfy the **gluing condition** if there exists an object E and morphisms $g : K \to E$ and $v : E \to C$ such that the square below is a pushout:

$$\begin{tikzcd} L \arrow[d, "m"'] \arrow[dr, phantom, "\lrcorner"] & K \arrow[l, "l"'] \arrow[d, dashed, "g"] \\ C & E \arrow[l, dashed, "v"] \end{tikzcd}$$

More concisely, the **gluing condition** holds if there is a **pushout complement** of $C \xleftarrow{m} L \xleftarrow{l} K$.

[8] Stephen Lack and Paweł Sobociński. “Adhesive and quasiadhesive categories”. In: *RAIRO-Theoretical Informatics and Applications* 39.3 (2005), pp. 511–545

Set of admissible matches and linear productions (cf. [9], Def. 7.3)

Let $\mathbf{C}$ be an adhesive category, and denote by $Lin(\mathbf{C})$ the **set of linear productions** on $\mathbf{C}$. Given an object $C \in \mathbf{C}$ and a linear production $p \in Lin(\mathbf{C})$, we denote the **set of admissible matches** $\mathcal{M}_p(C)$ as the set of monomorphisms $m : L \hookrightarrow C$ for which m satisfies the **gluing condition**. As a consequence, there exist objects and morphisms such that in the diagram below both squares are pushouts:

$$\begin{tikzcd} L \arrow[d, hook, "m"'] & K \arrow[l, hook', "l"'] \arrow[r, hook, "r"] \arrow[d, hook, dashed, "k"] & R \arrow[d, hook, dotted, "m'"] \\ C & K' \arrow[l, hook', dashed, "l'"'] \arrow[r, hook, dotted, "r'"] & D \end{tikzcd}$$

We write $p_m(C) := D$ for the object "produced" by the above diagram. The process is called **derivation** of C along production p and admissible match m, and denoted $C \underset{p,m}{\Longrightarrow} p_m(C)$.

[9] Stephen Lack and Paweł Sobociński. "Adhesive and quasiadhesive categories". In: *RAIRO-Theoretical Informatics and Applications* 39.3 (2005), pp. 511–545

Concurrent compositions of (linear) rules

Convention: unless mentioned otherwise, all arrows are assumed to be **monomorphisms**.

Dependency relations

For rules $p_1, p_2 \in Lin(\mathbf{C})$, a **dependency relation** consists of an object X_{12} and a span of monomorphisms $\mathbf{m} : R_1 \xleftarrow{x_1} X_{12} \xrightarrow{x_2} L_2$, s.t. K_{12}, K_{21} and morphisms illustrated below exist, where the cospan $R_1 \rightarrow Y_{12} \leftarrow L_2$ is the pushout of $\mathbf{m}$, and the two indicated regions are also pushouts; i.e. there exist pushouts complements of $K_1 \xrightarrow{r_1} R_1 \rightarrow Y_{12}$ and $K_2 \xrightarrow{l_2} L_2 \rightarrow Y_{12}$.

$$
\begin{tikzcd}
K_{21} \arrow[rr, dotted, "r_1'"] & & Y_{12} & & K_{12} \arrow[ll, dotted, "l_2'"'] \\
K_1 \arrow[u, dotted] \arrow[r, "r_1"'] & R_1 \arrow[ur, bend left] & X_{12} \arrow[l, "x_1"] \arrow[r, "x_2"'] & L_2 \arrow[ul, bend right] & K_2 \arrow[u, dotted] \arrow[l, "l_2"]
\end{tikzcd}
$$

Intuitively, the existence of the left and right pushout diagrams amounts to the two rules agreeing on the overlap specified by X_{12}, and amenable to being executed concurrently. We refer to such $\mathbf{m}$ as an **admissible match** of p_2 in p_1 and denote the set of these by $p_2 \Vdash p_1$.

Concurrent compositions of (linear) rules

- Algebraically speaking, given p_1, p_2 and $\mathbf{m} \in p_2 \Vdash p_1$, we can consider **"concurrent execution"** to be an operation that composes p_1 and p_2 "along" $\mathbf{m}$ to obtain a rule $p_2 \overset{\mathbf{m}}{\blacktriangleleft} p_1$. To obtain $p_2 \overset{\mathbf{m}}{\blacktriangleleft} p_1$, we extend the dependency relation by taking two further pushouts (marked with dotted arrows) and take a pullback (marked with dashed arrows):

$$
\begin{tikzcd}
 & & & Z_{12} \arrow[dll, dashed, bend right, "y_1"'] \arrow[drr, dashed, bend left, "y_2"] & & & \\
L_{12} & K_{21} \arrow[l, dotted, "l_1'"'] \arrow[rr, "r_1'"] & & Y_{12} & & K_{12} \arrow[ll, "l_2'"'] \arrow[r, dotted, "r_2'"] & R_{12} \\
L_1 \arrow[u, dotted] & K_1 \arrow[l, "l_1"] \arrow[r, "r_1"'] \arrow[u] & R_1 \arrow[ur, bend left] & X_{12} \arrow[l, "x_1"] \arrow[r, "x_2"'] & L_2 \arrow[ul, bend right] & K_2 \arrow[l, "l_2"] \arrow[r, "r_2"'] \arrow[u] & R_2 \arrow[u, dotted]
\end{tikzcd}
$$

- Now we define the **composite of p_1 with p_2 along m** as

$$
p_2 \overset{\mathbf{m}}{\blacktriangleleft} p_1 := (L_{12} \xleftarrow{z_1} Z_{12} \xrightarrow{z_2} R_{12}) \,, \; z_1 := l_1' \circ y_1 \,, \; z_2 := r_2' \circ y_2 \,.
$$

Concurrent compositions of (linear) rules

The following well-known result shows that **composition is compatible with application**:

Concurrency Theorem (cf. [10], Thm. 7.11)

Let $p, q \in Lin(\mathbf{C})$ be two linear rules and $C \in ob(\mathbf{C})$ an object.

- Given a **two-step sequence of derivations**

$$C \underset{p,m}{\Longrightarrow} p_m(C) \underset{q,n}{\Longrightarrow} q_n(p_m(C))\,,$$

  there exists a **composite rule** $r = p_2 \overset{\mathbf{d}}{\blacktriangleleft} p_1$ for **unique** $\mathbf{d} \in q \Vdash p$, and a **unique admissible match** $e \in \mathcal{M}_r(C)$, such that $C \underset{r,e}{\Longrightarrow} r_e(C)$ and $r_e(C) \cong q_n(p_m(C))$.

- Given a **dependency relation** $\mathbf{d} \in q \Vdash p$, $r = p_2 \overset{\mathbf{d}}{\blacktriangleleft} p_1$ and an **admissible match** $e \in \mathcal{M}_r(C)$, there exists a **unique pair of admissible matches** $m \in \mathcal{M}_p(C)$ and $n \in \mathcal{M}_q(p_m(C))$ such that $C \underset{p,m}{\Longrightarrow} p_m(C) \underset{q,n}{\Longrightarrow} q_n(p_m(C))$ with $q_n(p_m(C)) \cong r_e(C)$.

[10] Stephen Lack and Paweł Sobociński. “Adhesive and quasiadhesive categories”. In: *RAIRO-Theoretical Informatics and Applications* 39.3 (2005), pp. 511–545

Technical obstacle: proving associativity (NEW!)

We now show that **concurrent composition of linear rules** is, in a natural sense, **associative**:

Associativity Theorem (key result of our paper)

The **composition operation** $.\overset{\cdot}{\blacktriangleleft}.$ is **associative** in the following sense: given **linear rules**

$$p_1, p_2, p_3 \in Lin(\mathbf{C})$$

there exists a **bijective correspondence** between **pairs of admissible matches**

$$m_{21} \in p_2 \Vdash p_1 \quad \text{and} \quad m_{3(21)} \in p_3 \Vdash \left(p_2 \overset{m_{12}}{\blacktriangleleft} p_1\right),$$

and **pairs of admissible matches**

$$m_{32} \in p_3 \Vdash p_2 \quad \text{and} \quad m_{(32)1} \in \left(p_3 \overset{m_{23}}{\blacktriangleleft} p_2\right) \Vdash p_1$$

such that

$$p_3 \overset{m_{3(21)}}{\blacktriangleleft} \left(p_2 \overset{m_{21}}{\blacktriangleleft} p_1\right) = \left(p_3 \overset{m_{32}}{\blacktriangleleft} p_2\right) \overset{m_{(32)1}}{\blacktriangleleft} p_1\,.$$

On our proof of associativity

- Since DPO derivations are symmetric, it suffices to show one side of the correspondence. Our proof is **constructive**, demonstrating how, **given a pair of admissible matches**

$$(m_{21} \in p_2 \Vdash p_1 \text{ and } m_{3(21)} \in p_3 \Vdash \left(p_2 \overset{m_{12}}{\blacktriangleleft} p_1\right)) ,$$

  one obtains $m_{32} \in p_3 \Vdash p_2$ and $m_{(32)1} \in (p_3 \overset{m_{32}}{\blacktriangleleft} p_2) \Vdash p_1$ leading to the same two-step concurrent composition.

On our proof of associativity

- Since DPO derivations are symmetric, it suffices to show one side of the correspondence. Our proof is **constructive**, demonstrating how, **given a pair of admissible matches**

$$\left(m_{21} \in p_2 \Vdash p_1 \text{ and } m_{3(21)} \in p_3 \Vdash \left(p_2 \overset{m_{12}}{\blacktriangleleft} p_1\right)\right),$$

  one obtains $m_{32} \in p_3 \Vdash p_2$ and $m_{(32)1} \in (p_3 \overset{m_{32}}{\blacktriangleleft} p_2) \Vdash p_1$ leading to the same two-step concurrent composition.
- We begin with $p_2 \overset{m_{21}}{\blacktriangleleft} p_1$, p_3 and the dependency relation $m_{3(21)}$, illustrated below.

$$\begin{tikzcd}
& & & & & & & & & & & & & & & Y_{(12)3} & & & & K_{(12)3} \arrow[llll] \arrow[rr] & & R_{(12)3} \\
& L_{12} & & K_{12} \arrow[ll] \arrow[rrrr] & & & & Y_{12} & & & & K_{21} \arrow[llll] \arrow[rr] & & R_{12} \arrow[u, rr, bend left] & & X_{(12)3} \arrow[ll] \arrow[rr] & & L_3 \arrow[u, ll, bend right] & & K_3 \arrow[ll] \arrow[rr] \arrow[u] & & R_3 \arrow[u] \\
L_1 \arrow[ur] & & K_1 \arrow[ll] \arrow[rr] \arrow[ur] & & R_1 \arrow[u, rrr, bend left] & & X_{12} \arrow[ll] \arrow[rr] & & L_2 \arrow[ul] & & K_2 \arrow[ll] \arrow[rr] \arrow[ur] & & R_2 \arrow[ur] & & & & & & & & &
\end{tikzcd}$$

On our proof of associativity

- By Lemma 3.2, since the match $m_{3(21)}$ is by assumption admissible, we can find a pushout complement and pushout to extend the above diagram as follows,

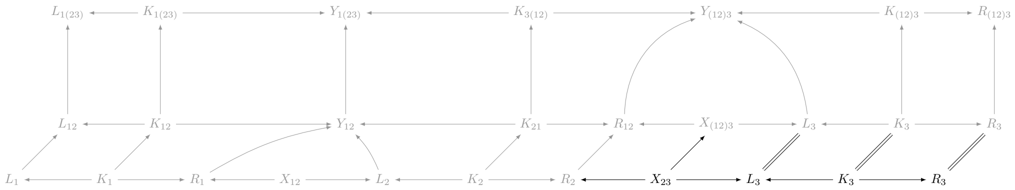
$$
\begin{tikzcd}
& & & & & & Y_{1(23)} & & & K_{3(12)} \arrow[lll] \arrow[rrr] & & & Y_{(12)3} & & K_{(12)3} \arrow[ll] \arrow[r] & R_{(12)3} \\
& L_{12} & & K_{12} \arrow[ll] \arrow[rrr] & & & Y_{12} \arrow[u] & & & K_{21} \arrow[lll] \arrow[rr] \arrow[u] & & R_{12} \arrow[ur, bend left] & X_{(12)3} \arrow[l] \arrow[r] & L_3 \arrow[ul, bend right] & K_3 \arrow[l] \arrow[r] \arrow[u] & R_3 \arrow[u] \\
L_1 \arrow[ur] & & K_1 \arrow[ll] \arrow[rr] \arrow[ur] & & R_1 \arrow[urr, bend left] & X_{12} \arrow[l] \arrow[rr] & & L_2 \arrow[ul] & K_2 \arrow[l] \arrow[rr] \arrow[ur] & & R_2 \arrow[ur] & & & & &
\end{tikzcd}
$$

  and again as below.

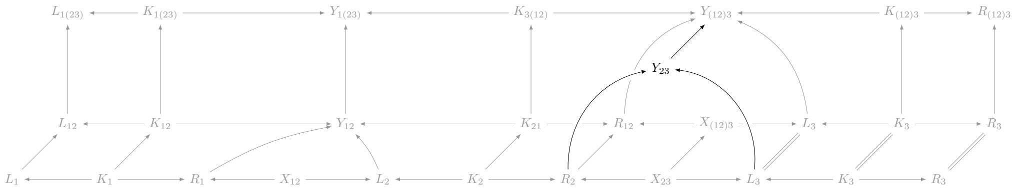
$$
\begin{tikzcd}
& L_{1(23)} & & K_{1(23)} \arrow[ll] \arrow[rrr] & & & Y_{1(23)} & & & K_{3(12)} \arrow[lll] \arrow[rrr] & & & Y_{(12)3} & & K_{(12)3} \arrow[ll] \arrow[r] & R_{(12)3} \\
& L_{12} \arrow[u] & & K_{12} \arrow[ll] \arrow[rrr] \arrow[u] & & & Y_{12} \arrow[u] & & & K_{21} \arrow[lll] \arrow[rr] \arrow[u] & & R_{12} \arrow[ur, bend left] & X_{(12)3} \arrow[l] \arrow[r] & L_3 \arrow[ul, bend right] & K_3 \arrow[l] \arrow[r] \arrow[u] & R_3 \arrow[u] \\
L_1 \arrow[ur] & & K_1 \arrow[ll] \arrow[rr] \arrow[ur] & & R_1 \arrow[urr, bend left] & X_{12} \arrow[l] \arrow[rr] & & L_2 \arrow[ul] & K_2 \arrow[l] \arrow[rr] \arrow[ur] & & R_2 \arrow[ur] & & & & &
\end{tikzcd}
$$

On our proof of associativity

- In the next step, we compute X_{23} as the evident pullback. Then we further extend the diagram via repeating the components of rule p_3.

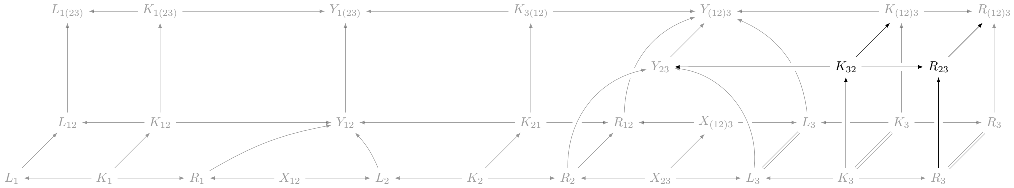


Now we push out R_2 and L_3 along X_{23}, obtaining $Y_{23} \to Y_{(12)3}$ from the universal property.

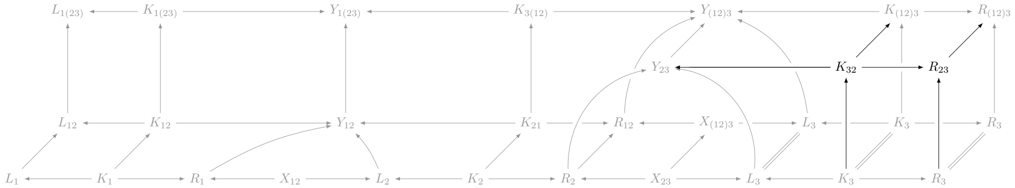


On our proof of associativity

- Next, we compute K_{32} by pulling back Y_{23} and $K_{1(23)}$ along $Y_{(12)3}$. We obtain $K_3 \to K_{32}$ from the universal property. To obtain the other morphisms, push out K_{32} and R_3 along K_3.



On our proof of associativity



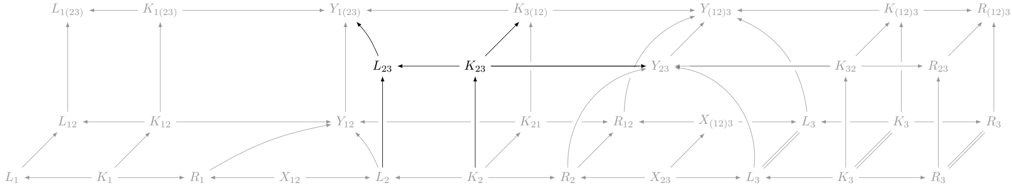
- We need to establish that the newly constructed front face on the left is a pushout. To do so, let us consider the cube on the left in isolation.

$$\begin{tikzcd}
& Y_{(12)3} & & K_{(12)3} \arrow[ll] \\
Y_{23} \arrow[ur] & & K_{32} \arrow[ll] \arrow[ur] & \\
& L_3 \arrow[uu] & & K_3 \arrow[ll] \arrow[uu] \\
L_3 \arrow[uu] \arrow[ur, equal] & & K_3 \arrow[ll] \arrow[uu] \arrow[ur, equal] &
\end{tikzcd}$$

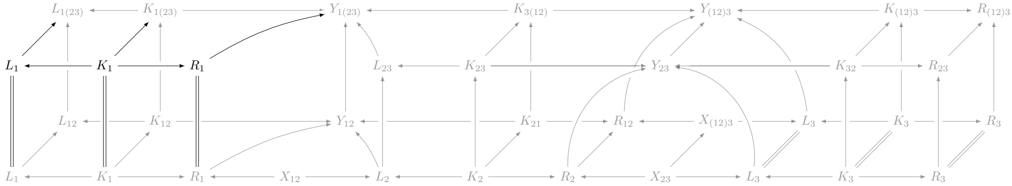
The rear face is a pushout, and therefore also a pullback. The bottom face is trivially both a pushout and a pullback. Pasting these two pushouts together yields a pushout, and since the top face—by construction—is a pullback, the front face is a pushout by Lemma 2.4: hence all faces of the cube, apart from the left and the right, are both pushouts and pullbacks.

On our proof of associativity

- We take advantage of the symmetry involved, and obtain two further pushouts as front faces in the following. Moreover, the two new upper faces are pushouts also.

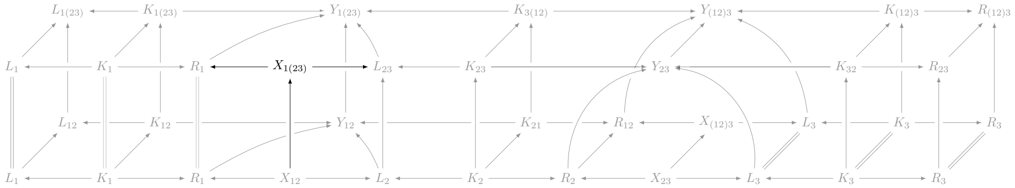


The next step is a trivial repetition of rule p_1: the new upper faces are both pushouts since they both arise as two pushouts pasted together.



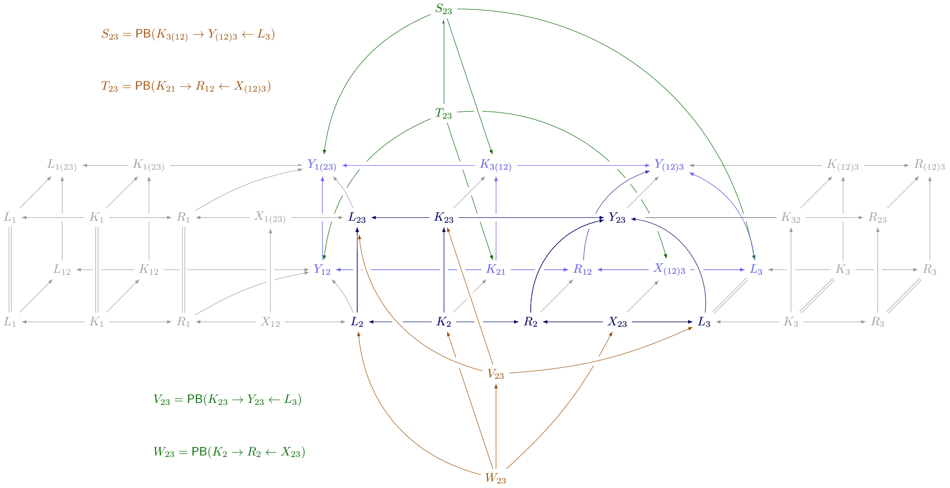
On our proof of associativity

- We now obtain $X_{(12)3}$ by pulling back R_1 and L_{23} along $Y_{1(23)}$, the remaining monomorphism $X_{12} \to X_{(12)3}$ follows from the universal property.



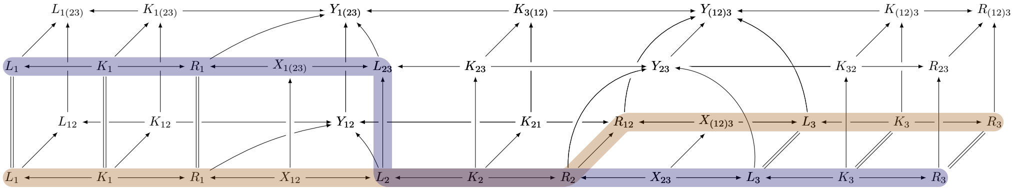
On our proof of associativity

- The final step consists in proving that the cospan $R_1 \to Y_{1(23)} \leftarrow L_{23}$ is the pushout of the span $R_1 \leftarrow X_{1(23)} \to L_{23}$. This proof requires a somewhat lengthy diagram chase presented in the appendix of our paper...



On our proof of associativity

- To conclude, the associativity property manifests itself in the following form, whereby the data provided along the path highlighted in orange below permits to uniquely compute the data provided along the path highlighted in blue (with both sets of overlaps computing the same “triple composite” production):



From associativity of concurrent derivations to rule algebras

- **non-determinacy in DPO rewriting:** each linear rule generically possesses **more than one** admissible match into a given object

$\Rightarrow$ need a structure to carry this non-determinism!

One interesting possibility (motivated by the physics literature)

- Each **linear rule** is lifted to an element of an abstract **vector space**.
- **Concurrent composition** of linear rules is lifted to a **bilinear multiplication operation** on this abstract vector space, endowing it with the structure of an **algebra**.
- The **action of rules on objects** is implemented by mapping each linear rule (seen as an element of the abstract algebra) to an endomorphism on an abstract vector space whose basis vectors are in bijection with the objects of the adhesive category.

Definition: rule algebra elements

Let $\delta : Lin(\mathbf{C}) \to \mathcal{R}_{\mathbf{C}}$ be defined as a morphism which maps each **linear rule** $p = (I \xrightharpoonup{r} O) \in Lin(\mathbf{C})$ to a **basis vector** $\delta(p)$ of a **free** $\mathbb{R}$**-vector space** $\mathcal{R}_{\mathbf{C}} \equiv (\mathcal{R}_{\mathbf{C}}, +, \cdot)$. In order to distinguish between elements of $Lin(\mathbf{C})$ and $\mathcal{R}_{\mathbf{C}}$, we introduce the **notation**

$$(O \stackrel{r}{\Leftarrow} I) := \delta\left(I \xrightharpoonup{r} O\right) .$$

We will later refer to $\mathcal{R}_{\mathbf{C}}$ as the $\mathbb{R}$-vector space of **rule algebra elements**.

The DPO rule algebra framework

Definition: rule algebra composition operation

Define the **rule algebra product** $*_{\mathcal{R}_{\mathbf{C}}}$ as the **binary operation**

$$*_{\mathcal{R}_{\mathbf{C}}} : \mathcal{R}_{\mathbf{C}} \times \mathcal{R}_{\mathbf{C}} \to \mathcal{R}_{\mathbf{C}} : (R_1, R_2) \mapsto R_1 *_{\mathcal{R}_{\mathbf{C}}} R_2 \,,$$

where for two basis vectors $R_i = \delta(p_i)$ encoding the linear rules $p_i \in Lin(\mathbf{C})$ $(i = 1, 2)$,

$$R_1 *_{\mathcal{R}_{\mathbf{C}}} R_2 := \sum_{\mathbf{m}_{12} \in p_1 \Vdash p_2} \delta\left(p_1 \overset{\mathbf{m}_{12}}{\blacktriangleleft} p_2\right) \,.$$

The definition is **extended** to arbitrary **(finite) linear combinations** of basis vectors by bilinearity, whence for $p_i, p_j \in Lin(\mathbf{C})$ and $\alpha_i, \beta_j \in \mathbb{R}$,

$$\left(\sum_i \alpha_i \cdot \delta(p_i)\right) *_{\mathcal{R}_{\mathbf{C}}} \left(\sum_j \beta_j \cdot \delta(p_j)\right) := \sum_{i,j} (\alpha_i \cdot \beta_j) \cdot (\delta(p_i) *_{\mathcal{R}_{\mathbf{C}}} \delta(p_j)) \,.$$

We refer to $\mathcal{R}_{\mathbf{C}} \equiv (\mathcal{R}_{\mathbf{C}}, *_{\mathcal{R}_{\mathbf{C}}})$ as the **rule algebra** (of linear DPO-type rewriting rules over the adhesive category $\mathbf{C}$).

Key theorem of the DPO rule algebra framework

Theorem

For every adhesive category $\mathbf{C}$, the associated rule algebra $\mathcal{R}_{\mathbf{C}} \equiv (\mathcal{R}_{\mathbf{C}}, *_{\mathcal{R}_{\mathbf{C}}})$ is an **associative algebra**. If $\mathbf{C}$ in addition possesses a **strict initial object** $c_{\varnothing} \in ob(\mathbf{C})$, $\mathcal{R}_{\mathbf{C}}$ is in addition a **unital algebra**, with **unit element** $R_{\varnothing} := (c_{\varnothing} \stackrel{\varnothing}{\Leftarrow} c_{\varnothing})$.

Proof.

Associativity follows immediately from the associativity of the operation $.\,\dot{\blacktriangleleft}\,.$ proved earlier. The claim that $R_{\varnothing}$ is the **unit element of the rule algebra** $\mathcal{R}_{\mathbf{C}}$ of an adhesive category $\mathbf{C}$ with strict initial object follows directly from the definition of the rule algebra product for $R_{\varnothing} *_{\mathcal{R}_{\mathbf{C}}} R$ and $R *_{\mathcal{R}_{\mathbf{C}}} R_{\varnothing}$ for $R \in \mathcal{R}_{\mathbf{C}}$. For clarity, we present below the category-theoretic composition calculation that underlies the equation $R_{\varnothing} *_{\mathcal{R}_{\mathbf{C}}} R = R$:

$$
\begin{tikzcd}
 & & & K \arrow[dll, dashed, bend right, "l"'] \arrow[drr, dashed, equal, bend left] & & & \\
L \arrow[r, dotted, equal] & L \arrow[rr, equal] & & L \arrow[dr, equal, bend left] & & K \arrow[ll, "l"'] \arrow[r, dotted, "r"] \arrow[d, equal] & R \arrow[d, dotted, equal] \\
\varnothing \arrow[u, dotted] & \varnothing \arrow[l] \arrow[r] \arrow[u] & \varnothing \arrow[ur, bend left] & \varnothing \arrow[l] \arrow[r] & L & K \arrow[l, "l"] \arrow[r, "r"'] & R
\end{tikzcd}
$$

$\square$

Canonical representations of DPO rule algebras

Definition

Let $\mathbf{C}$ be an **adhesive category with a strict initial object** $c_{\varnothing} \in ob(\mathbf{C})$, and let $\mathcal{R}_{\mathbf{C}}$ be its associated **rule algebra of DPO type**. Denote by $\hat{\mathbf{C}}$ the $\mathbb{R}$**-vector space of objects of** $\mathbf{C}$, whence (with $|C\rangle$ denoting the basis vector of $\hat{\mathbf{C}}$ associated to an element $C \in ob(\mathbf{C})$)

$$\hat{\mathbf{C}} := span_{\mathbb{R}}\left(\{\,|C\rangle|\, C \in ob(\mathbf{C})\}\right) \equiv (\hat{C}, +, \cdot)\,.$$

Then the **canonical representation** $\rho_{\mathbf{C}}$ of $\mathcal{R}_{\mathbf{C}}$ is defined as the algebra homomorphism $\rho_{\mathbf{C}} : \mathcal{R}_{\mathbf{C}} \to End(\hat{\mathbf{C}})$, defined to act on each rule algebra element $R = \delta(p)$ (for $p \in Lin(\mathbf{C})$) as

$$\rho_{\mathbf{C}}(\delta(p))\,|C\rangle := \begin{cases} \sum_{m \in \mathcal{M}_p(C)} |p_m(C)\rangle & \text{if } \mathcal{M}_p(C) \neq \varnothing \\ 0_{\hat{\mathbf{C}}} & \text{otherwise,} \end{cases}$$

extended to arbitrary elements of $\mathcal{R}_{\mathbf{C}}$ and of $\hat{\mathbf{C}}$ by linearity.

Note: The fact that ρ_C as given in this definition is indeed a **homomorphism of unital associative algebras** is shown in **Theorem 4.5** of our paper.

Summary: the DPO rule algebra framework

Slogan: DPO rule algebras arise as the associative unital algebras of concurrent compositions of DPO-type linear rewriting rules

Summary: the DPO rule algebra framework

Slogan: DPO rule algebras arise as the associative unital algebras of concurrent compositions of DPO-type linear rewriting rules

- each **linear rule** $p \equiv (I \xrightarrow{r} O)$ of a given **adhesive category** $\mathbf{C}$ is mapped to a **(basis) element** $\delta(p) \equiv (O \stackrel{r}{\Leftarrow} I)$ of a free $\mathbb{K}$-vector space (e.g. for $\mathbb{K} = \mathbb{R}$ or $\mathbb{K} = \mathbb{C}$)

Summary: the DPO rule algebra framework

Slogan: DPO rule algebras arise as the associative unital algebras of concurrent compositions of DPO-type linear rewriting rules

- each **linear rule** $p \equiv (I \stackrel{r}{\rightharpoonup} O)$ of a given **adhesive category** $\mathbf{C}$ is mapped to a **(basis) element** $\delta(p) \equiv (O \stackrel{r}{\Leftarrow} I)$ of a free $\mathbb{K}$-vector space (e.g. for $\mathbb{K} = \mathbb{R}$ or $\mathbb{K} = \mathbb{C}$)

- an **associative composition operation** is defined as

$$*_{\mathcal{R}_{\mathbf{C}}} : \mathcal{R}_{\mathbf{C}} \times \mathcal{R}_{\mathbf{C}} \to \mathcal{R}_{\mathbf{C}} : (R_1, R_2) \mapsto R_1 *_{\mathcal{R}_{\mathbf{C}}} R_2 \,,$$

where for two basis vectors $R_i = \delta(p_i)$ encoding the linear rules $p_i \in Lin(\mathbf{C})$ ($i = 1, 2$),

$$R_1 *_{\mathcal{R}_{\mathbf{C}}} R_2 := \sum_{\mathbf{m}_{12} \in p_1 \Vdash p_2} \delta\left(p_1 \overset{\mathbf{m}_{12}}{\blacktriangleleft} p_2\right) \,.$$

The definition is **extended** to arbitrary **(finite) linear combinations** of basis vectors by bilinearity.

Summary: the DPO rule algebra framework

Slogan: DPO rule algebras arise as the associative unital algebras of concurrent compositions of DPO-type linear rewriting rules

- If the adhesive category $\mathbf{C}$ in addition possesses a **strict initial object**, one may define a **canonical representation** $\rho_{\mathbf{C}} : \mathcal{R}_{\mathbf{C}} \to End_{\mathbb{K}}(\hat{\mathbf{C}})$ as

$$\hat{\mathbf{C}} := span_{\mathbb{R}}\left(\{\,|C\rangle |\, C \in ob(\mathbf{C})\}\right)\,, \quad \rho_{\mathbf{C}}(\delta(p))\,|C\rangle := \begin{cases} \sum_{m\in\mathcal{M}_p(C)} |p_m(C)\rangle & \text{if } \mathcal{M}_p(C) \neq \varnothing \\ 0_{\hat{\mathbf{C}}} & \text{otherwise,} \end{cases}$$

Application examples

The Heisenberg-Weyl algebra as the DPO discrete graph rewriting rule algebra

A first **consistency check** and **interesting special (and arguably simplest) case of rule algebras**:

The Heisenberg-Weyl algebra

Let $\mathcal{R}_0$ denote the rule algebra of DPO type rewriting for **discrete graphs**. Then the subalgebra $\mathcal{H}$ of $\mathcal{R}_0$ is defined as the algebra whose elementary **generators** are

$$x^{\dagger} := (\bullet \overset{\varnothing}{\Leftarrow} \varnothing)\,, \quad x := (\varnothing \overset{\varnothing}{\Leftarrow} \bullet)\,,$$

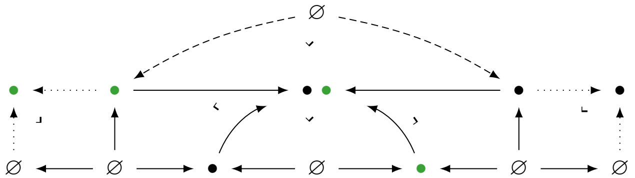
and whose elements are (finite) linear combinations of **words in $x^{\dagger}$ and x** (with concatenation given by the rule algebra multiplication $*_{\mathcal{R}_0}$) and of the **unit element** $R_{\varnothing} = (\varnothing \overset{\varnothing}{\Leftarrow} \varnothing)$. The canonical representation of $\mathcal{H}$ is the restriction of the canonical representation of $\mathcal{R}_0$ to $\mathcal{H}$.

The Heisenberg-Weyl algebra as the DPO discrete graph rewriting rule algebra

- famous property of the Heisenberg-Weyl algebra: with $a^\dagger := \rho(x^\dagger)$, $a := \rho(x)$, $\mathbb{1} := \rho(R_\varnothing)$,

$$[a, a^\dagger] := aa^\dagger - a^\dagger a = \mathbb{1}$$

- **realization/interpretation via the DPO rule algebra $\mathcal{R}_0$:** consider the following three **DPO-type compositions**

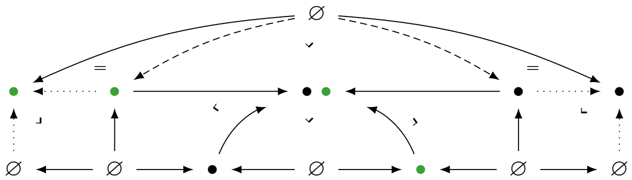


The Heisenberg-Weyl algebra as the DPO discrete graph rewriting rule algebra

- famous property of the Heisenberg-Weyl algebra: with $a^\dagger := \rho(x^\dagger)$, $a := \rho(x)$, $\mathbb{1} := \rho(R_\varnothing)$,

$$[a, a^\dagger] := aa^\dagger - a^\dagger a = \mathbb{1}$$

- **realization/interpretation via the DPO rule algebra $\mathcal{R}_0$:** consider the following three **DPO-type compositions**



$$\hat{=} \qquad (\varnothing \xleftarrow{\varnothing} \bullet) \overset{\varnothing \leftarrow \varnothing}{\blacktriangleleft} (\bullet \xleftarrow{\varnothing} \varnothing) = (\bullet \xleftarrow{\varnothing} \bullet)$$

The Heisenberg-Weyl algebra as the DPO discrete graph rewriting rule algebra

- famous property of the Heisenberg-Weyl algebra: with $a^\dagger := \rho(x^\dagger)$, $a := \rho(x)$, $\mathbb{1} := \rho(R_\varnothing)$,

$$[a, a^\dagger] := aa^\dagger - a^\dagger a = \mathbb{1}$$

- **realization/interpretation via the DPO rule algebra $\mathcal{R}_0$:** consider the following three **DPO-type compositions**

$$\begin{tikzcd}
 & & & \varnothing \arrow[dll, dashed] \arrow[drr, dashed] & & & \\
\varnothing & \varnothing \arrow[l, dotted] \arrow[rr] & & \bullet & & \varnothing \arrow[ll] \arrow[r, dotted] & \varnothing \\
\varnothing \arrow[u, dotted] & \varnothing \arrow[l] \arrow[r] \arrow[u] & \bullet \arrow[ur, bend left] & \bullet \arrow[l] \arrow[r] & \bullet \arrow[ul, bend right] & \varnothing \arrow[l] \arrow[r] \arrow[u] & \varnothing \arrow[u, dotted]
\end{tikzcd}$$

The Heisenberg-Weyl algebra as the DPO discrete graph rewriting rule algebra

- famous property of the Heisenberg-Weyl algebra: with $a^\dagger := \rho(x^\dagger)$, $a := \rho(x)$, $\mathbb{1} := \rho(R_\varnothing)$,

$$[a, a^\dagger] := aa^\dagger - a^\dagger a = \mathbb{1}$$

- **realization/interpretation via the DPO rule algebra $\mathcal{R}_0$:** consider the following three **DPO-type compositions**

$$\begin{tikzcd}
 & & & \varnothing \arrow[dlll, bend right] \arrow[dll, dashed] \arrow[drr, dashed] \arrow[drrr, bend left] & & & \\
\varnothing & \varnothing \arrow[l, dotted] \arrow[rr] \arrow[l, phantom, shift left=6, "="] & & \bullet & & \varnothing \arrow[ll] \arrow[r, dotted] \arrow[r, phantom, shift left=6, "="] & \varnothing \\
\varnothing \arrow[u, dotted] & \varnothing \arrow[l] \arrow[r] \arrow[u] & \bullet \arrow[ur, bend left] & \bullet \arrow[l] \arrow[r] & \bullet \arrow[ul, bend right] & \varnothing \arrow[l] \arrow[r] \arrow[u] & \varnothing \arrow[u, dotted]
\end{tikzcd}$$

$$\widehat{=} \qquad (\varnothing \xleftharpoondown{\varnothing} \bullet) \overset{\bullet \leftarrow \bullet}{\blacktriangleleft} (\bullet \xleftharpoondown{\varnothing} \varnothing) = (\varnothing \xleftharpoondown{\varnothing} \varnothing)$$

The Heisenberg-Weyl algebra as the DPO discrete graph rewriting rule algebra

- famous property of the Heisenberg-Weyl algebra: with $a^{\dagger} := \rho(x^{\dagger})$, $a := \rho(x)$, $\mathbb{1} := \rho(R_{\varnothing})$,

$$[a, a^{\dagger}] := aa^{\dagger} - a^{\dagger}a = \mathbb{1}$$

- **realization/interpretation via the DPO rule algebra $\mathcal{R}_0$:** consider the following three **DPO-type compositions**

$$
\begin{tikzcd}
 & & & \varnothing \arrow[dll, dashed] \arrow[drr, dashed] & & & \\
\bullet & \varnothing \arrow[l, dotted] \arrow[rr] & & \varnothing & & \varnothing \arrow[ll] \arrow[r, dotted] & \bullet \\
\bullet \arrow[u, dotted] & \varnothing \arrow[l] \arrow[r] \arrow[u] & \varnothing \arrow[ur, bend left] & \varnothing \arrow[l] \arrow[r] & \varnothing \arrow[ul, bend right] & \varnothing \arrow[l] \arrow[r] \arrow[u] & \bullet \arrow[u, dotted]
\end{tikzcd}
$$

The Heisenberg-Weyl algebra as the DPO discrete graph rewriting rule algebra

- famous property of the Heisenberg-Weyl algebra: with $a^\dagger := \rho(x^\dagger)$, $a := \rho(x)$, $\mathbb{1} := \rho(R_\varnothing)$,

$$[a, a^\dagger] := aa^\dagger - a^\dagger a = \mathbb{1}$$

- **realization/interpretation via the DPO rule algebra $\mathcal{R}_0$:** consider the following three **DPO-type compositions**

$$\begin{tikzcd}
 & & & \varnothing \arrow[dlll, bend right] \arrow[dll, dashed] \arrow[drr, dashed] \arrow[drrr, bend left] & & & \\
\bullet & \varnothing \arrow[l, dotted] \arrow[rr] & & \varnothing & & \varnothing \arrow[ll] \arrow[r, dotted] & \bullet \\
\bullet \arrow[u, dotted] & \varnothing \arrow[l] \arrow[r] \arrow[u] & \varnothing \arrow[ur, bend left] & \varnothing \arrow[l] \arrow[r] & \varnothing \arrow[ul, bend right] & \varnothing \arrow[l] \arrow[r] \arrow[u] & \bullet \arrow[u, dotted]
\end{tikzcd}$$

$$\hat{=} \qquad (\bullet \xleftarrow{\varnothing} \varnothing) \overset{\varnothing \leftarrow \varnothing}{\blacktriangleleft} (\varnothing \xleftarrow{\varnothing} \bullet) = (\bullet \xleftarrow{\varnothing} \bullet)$$

The Heisenberg-Weyl algebra as the DPO discrete graph rewriting rule algebra

- it is straightforward to verify that

$$\underbrace{x^{\dagger} *_{\mathcal{R}_0} \ldots *_{\mathcal{R}_0} x^{\dagger}}_{m \text{ times}} = \left(\bullet^{\uplus\, m} \stackrel{\varnothing}{\Leftarrow} \varnothing\right), \quad \underbrace{x *_{\mathcal{R}_0} \ldots *_{\mathcal{R}_0} x}_{n \text{ times}} = \left(\varnothing \stackrel{\varnothing}{\Leftarrow} \bullet^{\uplus\, n}\right)$$

The Heisenberg-Weyl algebra as the DPO discrete graph rewriting rule algebra

- it is straightforward to verify that

$$\underbrace{x^\dagger *_{\mathcal{R}_0} \ldots *_{\mathcal{R}_0} x^\dagger}_{m \text{ times}} = (\bullet^{\uplus\, m} \xleftarrow{\varnothing} \varnothing)\,, \quad \underbrace{x *_{\mathcal{R}_0} \ldots *_{\mathcal{R}_0} x}_{n \text{ times}} = (\varnothing \xleftarrow{\varnothing} \bullet^{\uplus\, n})$$

- analogously, we find the following:

$$\begin{aligned}
\Big(\underbrace{x *_{\mathcal{R}_0} \ldots *_{\mathcal{R}_0} x}_{m \text{ times}}\Big) \;*_{\mathcal{R}_0}\; \Big(\underbrace{x^\dagger *_{\mathcal{R}_0} \ldots *_{\mathcal{R}_0} x^\dagger}_{n \text{ times}}\Big) &= (\varnothing \xleftarrow{\varnothing} \bullet^{\uplus\, m}) \;*_{\mathcal{R}_0}\; (\bullet^{\uplus\, n} \xleftarrow{\varnothing} \varnothing)\\
&\equiv \delta(A_m) \;*_{\mathcal{R}_0}\; \delta(B_n)\\
&= \sum_{\mathbf{m}\in A_m \Vdash B_n} \delta\Big((\varnothing \xleftharpoonup{\varnothing} \bullet^{\uplus\, m}) \overset{\mathbf{m}}{\blacktriangleleft} (\bullet^{\uplus\, n} \xleftharpoonup{\varnothing} \varnothing)\Big)\\
&= \sum_{k=0}^{min(m,n)} \underbrace{\left(\frac{1}{k!}\frac{m!}{(m-k)!}\frac{n!}{(n-k)!}\right)}_{\#\text{ of ways to pick } k \text{ vertices from } m \text{ and from } n \text{ vertices disregarding order}} \cdot \left(\bullet^{\uplus\, n-k} \xleftarrow{\varnothing} \bullet^{\uplus\, m-k}\right)
\end{aligned}$$

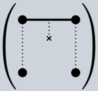
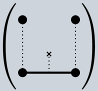
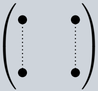
Combinatorics of graph ensembles and generators

Notational convention:

For the examples involving **rule algebras over adhesive categories of graphs**, we will employ a **graphical notation** where a **span of graph monomorphisms** $L \hookleftarrow K \hookrightarrow R$ is presented as a so-called **rule diagram**, with the graph L drawn **at the bottom**, the graph R drawn **at the top**, and where **dotted lines** indicate the **structure of the injective partial morphism** encoded in the span.

Example

We define the algebra $\mathcal{A}$ as the one generated $[11]$ by the rule algebra elements

$e_+ := \frac{1}{2} \cdot$  , $e_- := \frac{1}{2} \cdot$  , $d := \frac{1}{2} \cdot$ 

of the DPO rule algebra $\mathcal{R}_{\mathbf{uGraph}}$ constructed over the adhesive category $\mathbf{uGraph}$ of **(finite) undirected multigraphs and homomorphisms thereof**, whose **strict initial object** is the empty graph $\varnothing$.

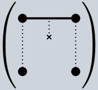
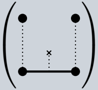
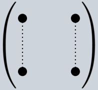
---

[11] As in the case of the Heisenberg-Weyl algebra, by "generated" we understand that a generic element of $\mathcal{A}$ is a finite linear combination of (finite) words in the generators and of the identity element $R_{\varnothing}$, with concatenation given by the rule algebra composition.

Combinatorics of graph ensembles and generators

Example

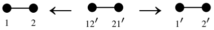
We define the algebra $\mathcal{A}$ as the one generated by the rule algebra elements

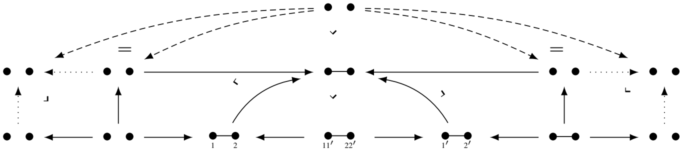
$e_+ := \frac{1}{2} \cdot$  , $\quad e_- := \frac{1}{2} \cdot$  , $\quad d := \frac{1}{2} \cdot$ 

of the DPO rule algebra $\mathcal{R}_{\mathbf{uGraph}}$.

- **commutation relations** (with $[x, y] := x *_{\mathcal{R}} y - y *_{\mathcal{R}} x$ for $\mathcal{R} \equiv \mathcal{R}_{\mathbf{uGraph}}$)

$$[e_-, e_+] = d\,, \quad [e_+, d] = [e_-, d] = 0\,.$$

- Here, the only nontrivial contribution (i.e. the one that renders the first commutator non-zero) may be computed from the DPO-type composition diagram [12] below and its variant for the admissible match  :



Combinatorics of graph ensembles and generators

- Let

$$E_+ := \rho(e_+)\,, \quad D := \rho(d)\,, \quad E_- := \rho(e_-)\,.$$

- Since the rules underlying the operators $E_\pm$ and D do **not modify the number of vertices** $n_V(G)$ when applied to a **graph state** $|G\rangle$, one may consider separately the action of these operators on the spaces $\hat{G}_N$ of **N-vertex graph states**

Combinatorics of graph ensembles and generators

- Already the action on the next more complicated case, i.e. **3-vertex graph states**, has a very interesting **combinatorial structure**:

$$
\begin{aligned}
E_+ \,|\bullet\ \ \bullet\ \ \bullet\rangle &= 3\,|\bullet\!-\!\bullet\ \ \bullet\rangle \equiv 3\,|\{1,0,0\}\rangle\\
E_+^2 \,|\bullet\ \ \bullet\ \ \bullet\rangle &= 3\left(|\bullet\!=\!\bullet\ \ \bullet\rangle + 2\,|\bullet\!-\!\bullet\!-\!\bullet\rangle\right) \equiv 3\left(|\{2,0,0\}\rangle + 2\,|\{1,1,0\}\rangle\right)\\
E_+^3 \,|\bullet\ \ \bullet\ \ \bullet\rangle &= 3\left(|\bullet\!\equiv\!\bullet\ \ \bullet\rangle + 6\,|\bullet\!=\!\bullet\!-\!\bullet\rangle + 2\,|\overset{\frown}{\bullet\!-\!\bullet\!-\!\bullet}\rangle\right)\\
&\equiv 3\left(|\{3,0,0\}\rangle + 6\,|\{2,1,0\}\rangle + 2\,|\{1,1,1\}\rangle\right)\\
&\ \ \vdots\\
E_+^n \,|\bullet\ \ \bullet\ \ \bullet\rangle &\equiv E_+^n\,|\{0,0,0\}\rangle = 3\sum_{k=0}^{n} T(n,k)\,|S(n,k)\rangle
\end{aligned}
$$

- the **state** $|\{f,g,h\}\rangle$ with $f \geqslant g \geqslant h \geqslant 0$ and $f+g+h=n$ is the **graph state on three vertices** with (in one of the possible presentations of the isomorphism class) f edges between the first two, g edges between the second two and h edges between the third and the first vertex
- $T(n,k)$ and $S(n,k)$ are given by the entry *A286030* of the OEIS database [13]

[13] *OEIS Foundation Inc. (2018), The On-Line Encyclopedia of Integer Sequences, https://oeis.org/A286030.*

Combinatorics of graph ensembles and generators

- Already the action on the next more complicated case, i.e. **3-vertex graph states**, has a very interesting **combinatorial structure**:

$$E_+^n \left|\bullet \quad \bullet \quad \bullet\right\rangle \equiv E_+^n \left|\{0,0,0\}\right\rangle = 3\sum_{k=0}^{n} T(n,k)\left|S(n,k)\right\rangle$$

- the **state** $|\{f,g,h\}\rangle$ with $f \geqslant g \geqslant h \geqslant 0$ and $f+g+h=n$ is the **graph state on three vertices** with (in one of the possible presentations of the isomorphism class) f edges between the first two, g edges between the second two and h edges between the third and the first vertex
- $T(n,k)$ and $S(n,k)$ are given by the entry *A286030* of the OEIS database [13]
- **Interpretation:** each triple $S(n,k)$ **encodes the outcome of a game of three players**, counting (without regarding the order of players) the number of wins per player for a total of n games. Then $T(n,k)/3^{(n-1)}$ gives the **probability that a particular pattern $S(n,k)$ occurs in a random sample**.

[13] *OEIS Foundation Inc. (2018), The On-Line Encyclopedia of Integer Sequences, https://oeis.org/A286030.*

Sketch – this would be an entire second talk...

Continuous-time Markov chains (CTMCs) from extensive categories

Let $\mathbf{C}$ be an **extensive** category (adhesive with strict initial object). For some (**finite!**) index set $\mathcal{I}$, let $\{r_i\}_{i\in\mathcal{I}}$ be a set of **base rates** (with $r_i \in \mathbb{R}_{>0}$), and

$$\left\{I_i \xleftarrow{i_i} K_i \xrightarrow{o_i} O_i\right\}_{i\in\mathcal{I}}$$

be a set of **linear rules**. Let $\hat{\mathbf{C}}$ be the $\mathbb{R}$-vector space of **states** (with basis vectors $|o\rangle$ for $o \in ob(\mathbf{C})$), and let $Prob(\mathbf{C})$ be the space of **sub-probability distributions over** $\hat{\mathbf{C}}$.

Then together with an **initial state** $|\Psi_0\rangle \in Prob(\mathbf{C})$, this date defines a **continuous-time Markov chain (CTMC)** with **time-dependent state** $|\Psi(t)\rangle \in Prob(\mathbf{C})$ for all $t \in \mathbb{R}_{\geq 0}$, **evolution equation**

$$\tfrac{d}{dt}\,|\Psi(t)\rangle = H\,|\Psi(t)\rangle$$

and **evolution operator**

$$H = \sum_{i\in\mathcal{I}} r_i \left(\rho(I_i \xleftarrow{i_i} K_i \xrightarrow{o_i} O_i) - \rho(I_i \xleftarrow{i_i} K_i \xrightarrow{i_i} I_i)\right)\;.$$

Stochastic mechanics of continuous-time Markov chains

Two more important “ingredients” (of the so-called **stochastic mechanics framework**):

- **“dual projection vector”:**

$$\langle| : \hat{\mathbf{C}} \to \mathbb{R} : |o\rangle \mapsto \langle\rangle o := 1_{\mathbb{R}}$$

- **observables:** for all **monomorphisms** $K \xhookrightarrow{m} M$ in $\mathbf{C}$, the **diagonal linear operators** $\mathcal{O}^m_M$ defined as

$$\mathcal{O}^m_M := \rho(M \xhookleftarrow{m} K \xhookrightarrow{m} M)$$

are **observables**, in the sense that

$$\mathcal{O}^m_M\,|o\rangle = N^m_M(o)\cdot|o\rangle\,,\quad N^m_M(o) = \text{\# ways to apply the the linear rule } M \xhookleftarrow{m} K \xhookrightarrow{m} M \text{ to } o$$

Stochastic mechanics of continuous-time Markov chains

Fact:

The **evolution operator** H of a CTMC satisfies

$$\langle | H \equiv 0\,,$$

whence the **expectation value**

$$\langle \mathcal{O}_M^m\rangle(t) := \langle | \,\mathcal{O}_M^m\, |\Psi(t)\rangle$$

satisfies the **evolution equation**

$$\tfrac{d}{dt}\langle \mathcal{O}_M^m\rangle(t) = \langle [\mathcal{O}_M^m, H]\rangle(t)\,, \quad [\mathcal{O}_M^m, H] := \mathcal{O}_M^m H - H\mathcal{O}_M^m\,.$$

$\Rightarrow$ together with a property called **jump-closure**, this permits to calculate **stochastic evolutions** from **(combinatorial) commutation relations** [14], [15], [16]

[14] Nicolas Behr, Vincent Danos, and Ilias Garnier. "Stochastic mechanics of graph rewriting". In: *Proceedings of the 31st Annual ACM-IEEE Symposium on Logic in Computer Science (LICS 2016)* (2016), pp. 46–55

[15] Nicolas Behr and Pawel Sobocinski. "Rule Algebras for Adhesive Categories". In: *arXiv preprint arXiv:1807.00785 (accepted for CSL'18)* (2018)

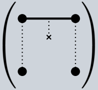
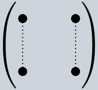
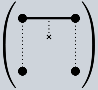
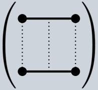
[16] Nicolas Behr, Vincent Danos, and Ilias Garnier. "Combinatorial Conversion and Moment Bisimulation for Stochastic Rewriting Systems (in preparation)". In: ()

An illustrative (toy) example

Example: edge “birth-death” systems

Let H be the evolution operator of a CTMC based on the two linear rules of **edge creation** and **edge annihilation** (with **base rates** $\kappa_+, \kappa_- \in \mathbb{R}_{\geqslant 0}$), whence

$$H = \kappa_+ \left(E_+ - O_{\bullet\,\bullet}\right) + \kappa_- \left(E_- - O_E\right)$$

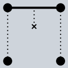
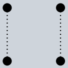
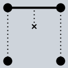
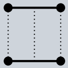
$E_+ := \frac{1}{2}\rho$  , $O_{\bullet\,\bullet} := \frac{1}{2}\rho$  , $E_- := \frac{1}{2}\rho$  , $O_E := \frac{1}{2}\rho$  .

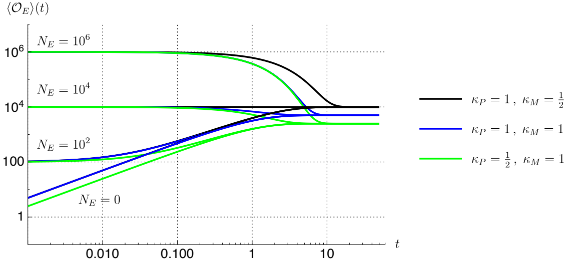
An illustrative (toy) example

Example: edge "birth-death" systems

Let H be the evolution operator of a CTMC based on the two linear rules of **edge creation** and **edge annihilation** (with **base rates** $\kappa_+, \kappa_- \in \mathbb{R}_{\geqslant 0}$), whence

$$H = \kappa_+ \left(E_+ - O_{\bullet\,\bullet}\right) + \kappa_- \left(E_- - O_E\right)$$

$E_+ := \frac{1}{2}\rho\Big($  $\Big)$, $O_{\bullet\,\bullet} := \frac{1}{2}\rho\Big($  $\Big)$, $E_- := \frac{1}{2}\rho\Big($  $\Big)$, $O_E := \frac{1}{2}\rho\Big($  $\Big)$.



Time-evolution of $\langle O_E \rangle(t)$ (the **average number of edges** at time t) for **initial state** $|\Psi(0)\rangle = |G_0\rangle$ with $N_V = 100$ vertices.

Conclusion and outlook

Conclusion and outlook

- we have successfully introduced a fully self-consistent framework of **DPO rule algebras** for **arbitrary adhesive categories**
- amongst the main technical results is a **novel proof of associativity** of DPO rule compositions
- the framework appears to have a vast variety of possible applications in computer science, combinatorics and beyond

- overview of **work in progress:**
  - extension to **restricted rewriting theories** (**MSCA project** with **J. Krivine (IRIF)**)
  - stochastic mechanics and bisimulations for complex systems based on rewriting (with **V. Danos and I. Garnier (ENS Paris)**)
  - **analytical and enumerative combinatorics** of graphical structures and their generators (with **N. Zeilberger (U Birmingham)**)

Thank you!

- Nicolas Behr, Vincent Danos, and Ilias Garnier. “Combinatorial Conversion and Moment Bisimulation for Stochastic Rewriting Systems (in preparation)”. In: ().
- Nicolas Behr, Vincent Danos, and Ilias Garnier. “Stochastic mechanics of graph rewriting”. In: *Proceedings of the 31st Annual ACM-IEEE Symposium on Logic in Computer Science (LICS 2016)* (2016), pp. 46–55.
- Nicolas Behr and Pawel Sobocinski. “Rule Algebras for Adhesive Categories”. In: *arXiv preprint arXiv:1807.00785 (accepted for CSL’18)* (2018).
- Richard Garner and Stephen Lack. “On the axioms for adhesive and quasiadhesive categories”. In: *Theor. App. Categories* 27.3 (2012), pp. 27–46.
- Stephen Lack and Paweł Sobociński. “Adhesive and quasiadhesive categories”. In: *RAIRO-Theoretical Informatics and Applications* 39.3 (2005), pp. 511–545.

- Stephen Lack and Paweł Sobociński. “Toposes are adhesive”. In: *Graph Transformations, Third International Conference, (ICGT 2006)*. Vol. 4178. LNCS. Springer, 2006, pp. 184–198.
- *OEIS Foundation Inc. (2018), The On-Line Encyclopedia of Integer Sequences, https://oeis.org/A286030*.