Diagrammatic Polyhedral Algebra

Filippo Bonchi
University of Pisa, Italy

Alessandro Di Giorgio
University of Pisa, Italy

Paweł Sobociński
Tallinn University of Technology, Estonia

Abstract

We extend the theory of Interacting Hopf algebras with an order primitive, and give a sound and complete axiomatisation of the prop of polyhedral cones. Next, we axiomatise an affine extension and prove soundness and completeness for the prop of polyhedra.

2012 ACM Subject Classification Theory of computation → Categorical semantics; Theory of computation → Concurrency

Keywords and phrases String diagrams, Polyhedral cones, Polyhedra

Digital Object Identifier 10.4230/LIPIcs.FSTTCS.2021.40

Related Version *Full Version*: `https://arxiv.org/abs/2105.10946` [5]

Funding *Filippo Bonchi*: Supported by the Ministero dell'Università e della Ricerca of Italy under Grant No. 201784YSZ5, PRIN2017 – ASPRA (Analysis of Program Analyses).
Alessandro Di Giorgio: Supported by the Ministero dell'Università e della Ricerca of Italy under Grant No. 201784YSZ5, PRIN2017 – ASPRA (Analysis of Program Analyses).
Paweł Sobociński: Supported by the ESF funded Estonian IT Academy research measure (project 2014-2020.4.05.19-0001) and the Estonian Research Council grant PRG1210.

Acknowledgements The authors have benefited, at the early stage of this work, of enlightening comments and exciting discussions with Guillaume Boisseau. In particular, Guillaume proposed several simplifications to the axiomatisations, showed us the first proof of the generalised spider and the one for its polar.

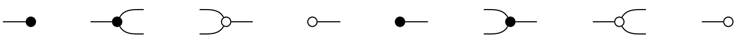
1 Introduction

Engineers and scientists of different fields often rely on diagrammatic notations to model systems of various sorts but, to perform a rigorous analysis, diagrams usually need to be translated to more traditional mathematical language. Indeed diagrams have the advantage to be quite intuitive, highlight connectivity, distribution and communication topology of systems but they usually have an informal meaning and, even when equipped with a formal semantics, diagrams cannot be easily manipulated like standard mathematical expressions. *Compositional network theory* [3, 24] is a multidisciplinary research program studying diagrams as first class citizens: diagrammatic languages come equipped with a formal semantics, which has the key feature to be compositional; moreover diagrams can be manipulated like ordinary symbolic expressions if an appropriate equational theory–ideally characterising semantic equality–can be identified. This approach has been shown effective in various settings like for instance, digital [20] and electrical circuits [4, 24], quantum protocols [14, 15], linear dynamical systems [2, 32], Petri nets [7], Bayesian networks [23] and query languages [19, 10].

The common technical infrastructure is provided by *string diagrams* [31]: arrows of a symmetric monoidal category freely generated by a monoidal signature. Intuitively, the signature is a set of generators and diagrams are simply obtained by composing in series

41st IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2021).
Editors: Mikołaj Bojańczyk and Chandra Chekuri; Article No. 40; pp. 40:1–40:18
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

(horizontally) and in parallel (vertically) generators plus some basic wires (— and ⨯). The following set of generators is common to (most of) the aforementioned systems and, surprisingly enough, (almost) the same algebraic laws hold in the various settings.

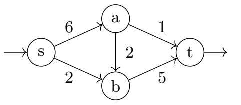


It is convenient to give an intuition of the intended meaning of such generators by relying on the semantics from [32] that, amongst the aforementioned works, is the most relevant for the present paper: the copier —●< receives one value on the left and emits two copies on the right; the discharger —● receives one value on the left and nothing on the right; the adder >○— receives two values on the left and emits their sum on the right; the zero ○— receives nothing on the left and constantly emits 0 on the right. The behaviour of the remaining four generators is the same but left and right are swapped. Here values are meant to be rational numbers. To deal with values from an arbitrary fields k, one has to add a generator —$\boxed{k}$— for each $k \in$ k; its intended meaning is the one of an amplifier: the value received on the left is multiplied by k and emitted on the right.

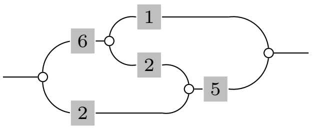
This semantics has two crucial properties: first, it enjoys a sound and complete axiomatisation called the theory of Interacting Hopf Algebras ($\mathbb{IH}$); second, it can express exactly *linear relations*, namely relations forming vector spaces over k. In other words, diagrams modulo the laws of $\mathbb{IH}$ are in one to one correspondence with linear relations.

In this paper we extend $\mathbb{IH}$ in order to express exactly relations that are *polyhedra*, rather than mere vector spaces. Indeed, polyhedra allow the modeling of bounded spaces which are ubiquitous in computer science. For instance, in abstract interpretation [17] polyhedra represent bounded sets of possible values of variables; in concurrency theory and linear optimisation one always deals with systems having a bounded amounts of resources.

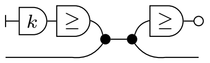
To catch a glimpse of our result, consider the flow network [1] in (1): edges are labeled with a positive real number representing their maximum capacity; the flow enters in the source (the node s) and exits from the sink (the node t).



(1)



(2)



(3)

The network in (1) is represented within our diagrammatic language as in (2) where —$\boxed{k}$— is syntactic sugar for the diagram in (3). Here —$\boxed{\geq}$— and ⊢— are the two novel generators that we need to add to Interacting Hopf Algebras to express exactly polyhedra: —$\boxed{\geq}$— constrains the observation on the left to be greater or equal to the one on the right; ⊢— constantly emits 1 on the right. Observe that (3) forces the values on the left and on the right to be equal and to be in the interval $[0, k]$; the use of >○— and —○< for the nodes forces the sum of the flows entering on the left to be equal to the sum of the flows leaving from the right.

An important property of flow networks is the maximum flow that can enter in the source and arrive to the the sink. The sound and complete axiomatisation that we introduce allows to compute their maximum flow by mean of intuitive graphical manipulations: for instance, the diagram in (2) can be transformed in —$\boxed{5}$—, meaning that its maximum flow is exactly 5. We will come back to flow networks at the end of §5 (Example 31).

The remainder of the paper is organised as follows. We recall the basic categorical tools for string diagrams in §2 and the theory of Interacting Hopf Algebras in §3. In §4, we extend the syntax of Interacting Hopf Algebras with the generator —$\boxed{\geq}$—. On the semantic side, this allows to move from linear relations to *polyhedral cones*, for which we give a sound and fully complete axiomatisation in terms of the diagrammatic syntax. The proof of completeness involves a diagrammatic account of Fourier-Motzkin elimination, two normal forms leading to the Weyl-Minkowski theorem, and a simple, inductive account of the notion of polar cone.

The results in §4 represent our main technical effort. Indeed, to pass from polyhedral cones to polyhedra in §5, it is enough to add the generator ⊢—, originally introduced in [9] to move from linear to affine relations, and one extra axiom. The proof substantially exploits the homogenization technique to reduce completeness for polyhedra to the just proved completeness for polyhedral cones.

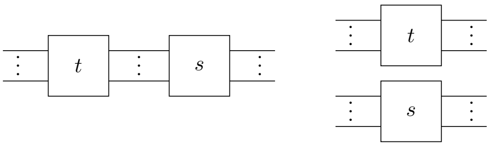
Finally, in §6, we conclude by showing a stateful extension of our diagrammatic calculus. By simply adding a register $-\boxed{x}-$ we obtain a complete axiomatisation for stateful polyhedral processes: these are exactly all transition systems where both states and labels are vectors from some vector spaces and the underlying transition relation forms a polyhedron. Stateful polyhedral processes seem to be a sweet spot in terms of expressivity: on the one hand, they properly generalise signal flow graphs [29], on the other, as illustrated in §6, they allow us to give a compositional account of continuous Petri nets [18].

2 Props and Symmetric Monoidal Theories

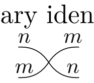
The diagrammatic languages studied in network theory, e.g., [16, 30, 21, 7], can be treated formally using the category theoretic notion of prop [28, 26] (product and permutation category). A *prop* is a symmetric strict monoidal (ssm) category with objects natural numbers, where the monoidal product $\oplus$ on objects is addition. Morphisms between props are ssm functors that act as identity on objects. The usual methodology is to use two props: Syn, the arrows of which are the diagrammatic terms of the language, and Sem, the arrows of which are the intended semantics. A morphism $[\![\cdot]\!] : \mathsf{Syn} \to \mathsf{Sem}$ assigns semantics to terms, with the functoriality of $[\![\cdot]\!]$ guaranteeing compositionality.

The syntactic prop Syn is usually freely generated from a *monoidal signature* Σ, namely a set of generators $o\colon n \to m$ with arity $n \in \mathbb{N}$ and coarity $m \in \mathbb{N}$. Intuitively, the arrows of Syn are diagrams wired up from the generators. A way of giving a concrete description is via Σ-terms. The set of Σ-*terms* is obtained by composing generators in Σ, the identities $id_0\colon 0 \to 0$, $id_1\colon 1 \to 1$ and the symmetry $\sigma_{1,1}\colon 2 \to 2$ with ; and $\oplus$. This is a purely formal process: given Σ-terms $t\colon k \to l$, $u\colon l \to m$, $v\colon m \to n$, one constructs Σ-terms $t;u\colon k \to m$ and $t \oplus v\colon k+n \to l+n$. Now, the *prop freely generated by a signature* Σ, hereafter denoted by P_Σ, has as its arrows $n \to m$ the set of Σ-terms $n \to m$ modulo the laws of ssm categories.

There is a well-known, natural graphical representation for arrows of a freely generated prop as string diagrams, which we now sketch. A Σ-term $n \to m$ is pictured as a box with n ordered wires on the left and m on the right. Composition via ; and $\oplus$ are rendered graphically by horizontal and vertical juxtaposition of boxes, respectively.



(4)

Moreover $id_1\colon 1 \to 1$ is pictured as ——, the symmetry $\sigma_{1,1}\colon 1+1 \to 1+1$ as , and the unit object for $\oplus$, that is, $id_0\colon 0 \to 0$ as the empty diagram . Arbitrary identities id_n and symmetries $\sigma_{n,m}$ are generated according to (4) and drawn as $\overset{n}{——}$ and 

, respectively.

Given a diagrammatic language Syn and a morphism $[\![\cdot]\!] : \mathsf{Syn} \to \mathsf{Sem}$, a useful task is to identify a sound and (ideally) complete set of characterising equations E: $[\![c]\!] = [\![d]\!]$ iff c and d are equal in $\overset{E}{=}$, the smallest congruence (w.r.t. ; and $\oplus$) containing E. Formally, the set E consists of pairs $(t, t' \colon n \to m)$ of Σ-terms with the same arity and coarity. Then Σ together with E form a *symmetric monoidal theory* (smt), providing a calculus of diagrammatic reasoning. Any smt (Σ, E) yields a prop $\mathsf{P}_{\Sigma,E}$, obtained by quotienting the P_Σ by $\overset{E}{=}$.

Another issue is expressivity: one would like to characterise the image of Syn through $[\![\cdot]\!]$, namely a subprop IM of Sem consisting of exactly those arrows d of Sem for which there exists some c in Syn such that $[\![c]\!] = d$. When this is possible and a sound and complete axiomatization is available, the semantics map $[\![\cdot]\!]$ factors as follows:

$$\begin{tikzcd} \mathsf{Syn} = \mathsf{P}_\Sigma \arrow[r, two heads, "q"] \arrow[rrr, bend left, "{[\![\cdot]\!]}"] & \mathsf{P}_{\Sigma,E} \arrow[r, "\cong"] & \mathsf{IM} \arrow[r, tail, "\iota"] & \mathsf{Sem} \ . \end{tikzcd}$$

The morphism q quotients P_Σ by $\overset{E}{=}$, ι is the inclusion of IM in Sem and $\cong$ is an iso between $\mathsf{P}_{\Sigma,E}$ and IM. In this case we say that (Σ, E) is the (symmetric monoidal) theory of IM.

Let k be an ordered field. In this paper Sem is fixed to be the following prop.

▶ **Definition 1.** Rel_k *is the prop where arrows $n \to m$ are relations $R \subseteq \mathsf{k}^n \times \mathsf{k}^m$.*

- *Composition is relational: given $R \colon n \to m$ and $S \colon m \to o$,*

$$R \,;\, S = \{\, (u, v) \in \mathsf{k}^n \times \mathsf{k}^o \ \mid\ \exists w \in \mathsf{k}^m.\ (u, w) \in S \wedge (w, v) \in R \,\}$$

- *The monoidal product is cartesian product: given $R \colon n \to m$ and $S \colon o \to p$,*

$$R \oplus S = \{\, (\begin{pmatrix} u_1 \\ u_2 \end{pmatrix}, \begin{pmatrix} v_1 \\ v_2 \end{pmatrix}) \in \mathsf{k}^{n+o} \times \mathsf{k}^{m+p} \ \mid\ (u_1, v_1) \in R \wedge (v_1, v_2) \in S \,\}$$

- *The symmetries $\sigma_{n,m} \colon n+m \to m+n$ are the relations* $\{\, (\begin{pmatrix} u \\ v \end{pmatrix}, \begin{pmatrix} v \\ u \end{pmatrix}) \ \mid\ u \in \mathsf{k}^n, v \in \mathsf{k}^m \,\}$

For IM, we will consider the following three props.

▶ **Definition 2.** *We define three sub-props of* Rel_k*. Arrows $n \to m$*

- *in* $\mathsf{LinRel}_\mathsf{k}$ *are vector spaces* $\{(x, y) \in \mathsf{k}^n \times \mathsf{k}^m \mid A \begin{pmatrix} x \\ y \end{pmatrix} = 0\}$ *for some matrix A;*
- *in* PC_k *are polyhedral cones* $\{(x, y) \in \mathsf{k}^n \times \mathsf{k}^m \mid A \begin{pmatrix} x \\ y \end{pmatrix} \geq 0\}$ *for some matrix A;*
- *in* P_k *are polyhedra* $\{(x, y) \in \mathsf{k}^n \times \mathsf{k}^m \mid A \begin{pmatrix} x \\ y \end{pmatrix} + b \geq 0\}$ *for some matrix A and $b \in \mathsf{k}^p$.*

Identities, permutations, composition and monoidal product are defined as in Rel_k*.*

▶ Remark 3. In Definition 2, A is a matrix with $n + m$ columns and p rows, for some $p \in \mathbb{N}$. Observe that the matrix A gives rise also to arrows $n' \to m'$ with n', m' different from n, m but, such that $n' + m' = n + m$. This is justified by the isomorphism of $\mathsf{k}^n \times \mathsf{k}^m$ and $\mathsf{k}^{n'} \times \mathsf{k}^{m'}$. Note therefore that the left and the right boundaries should not be confused with inputs and outputs. This is a common feature in diagrammatic approaches relying on a notion of relational composition which is unbiased.

Showing that the above are well-defined – e.g. that the composition of polyhedral cones is a polyhedral cone – requires some well-known results, which are given in [5, Appendix A]. In §3, we recall the theory of $\mathsf{LinRel}_\mathsf{k}$, in §4 we identify the theory of PC_k and, in §5, that of P_k.

2.1 Ordered Props and Symmetric Monoidal Inequality Theories

As relations $R, S \colon n \to m$ in Rel_k carry the partial order of inclusion $\subseteq$, it is useful to be able to state when $[\![c]\!] \subseteq [\![d]\!]$ for some c, d in Syn (see e.g. [6] for motivating examples). In order to consider such inclusions, it is convenient to look at Rel_k as an ordered prop.

▶ **Definition 4.** *An* ordered prop *is a prop enriched over the category of posets: a symmetric strict monoidal 2-category with objects the natural numbers, monoidal product on objects given by addition, where each set of arrows $n \to m$ is a poset, with composition and monoidal product monotonic. Similarly, a* pre-ordered prop *is a prop enriched over the category of pre-orders. A morphism of (pre-)ordered props is an identity-on-objects symmetric strict 2-functor.*

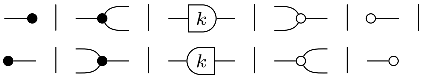
Just as SMTs yield props, *Symmetric Monoidal Inequalities Theories* [6] (SMITs) give rise to ordered props. A SMIT is a pair (Σ, I) where Σ is a signature and I is a set of inequations: as for equations, the underlying data is a pair $(t, t' \colon n \to m)$ of Σ-terms with the same arity and coarity. Unlike equations, however, we understand this data as directed: $t \leq t'$.

To obtain the free ordered prop from an SMIT, first, we construct the free pre-ordered prop: arrows are Σ-terms. The homset orders, hereafter denoted by $\overset{I}{\subseteq}$, are determined by closing I by reflexivity, transitivity, ; and $\oplus$: this is the smallest precongruence (w.r.t. ; and $\oplus$) containing I. Then, we obtain the free ordered prop by quotienting the free pre-ordered prop by the equivalence induced by $\overset{I}{\subseteq}$, i.e. quotienting wrt anti-symmetry.

Any prop can be regarded as an ordered prop with the discrete ordering. Moreover any SMT (Σ, E) gives rise to a canonical SMIT (Σ, I) where each equation is replaced by two inequalities $I = E \cup E^{op}$ in the obvious way. In the remainder of this paper, we always consider SMITs but, for the sake of readability, we generically refer to them just as *theories*. Such theories consist of both inequalities and equations, that are generically called *axioms*. Similarly, all props considered in the paper, their morphisms and isomorphisms are ordered.

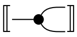
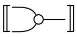
3 The theory of Linear relations

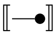
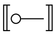
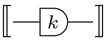
In this section, we recall from [11, 2, 32] the theory of Interacting Hopf algebras. The signature consists of the following set of generators, where k ranges over a fixed field k.

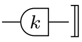


(5)
(6)

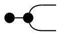
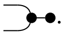
For each generator, its arity and coarity are given by the number of dangling wires on the left and, respectively, on the right. For instance has arity 1 and coarity 0. We call Circ the prop freely generated by this signature and we refer to its arrows as circuits. We use Circ as the *syntax* of our starting diagrammatic language. The semantics is given as the prop morphism $[\![\cdot]\!] : \mathsf{Circ} \to \mathsf{Rel}_\mathsf{k}$ defined for the generators in (5) as

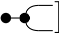
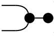
$[\![$  $]\!] = \{(x, \begin{pmatrix} x \\ x \end{pmatrix}) \mid x \in \mathsf{k}\}$ $\quad$ $[\![$  $]\!] = \{(\begin{pmatrix} x \\ y \end{pmatrix}, x + y) \mid x, y \in \mathsf{k}\}$

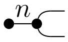
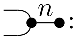
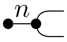
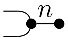
$[\![$  $]\!] = \{(x, \bullet) \mid x \in \mathsf{k}\}$ $\quad$ $[\![$  $]\!] = \{(\bullet, 0)\}$ $\quad$ $[\![$  $]\!] = \{(x, k \cdot x) \mid x \in \mathsf{k}\}$ $\quad$ (7)

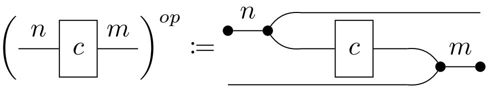
and, symmetrically, for the generators in (6). For instance, $[\![$  $]\!] = \{(k \cdot x, x) \mid x \in \mathsf{k}\}$. The semantics of the identities, symmetries and compositions is given by the *functoriality* of $[\![\cdot]\!]$, e.g., $[\![c\,;d]\!] = [\![c]\!]\ ;\ [\![d]\!]$. Above we used $\bullet$ for the unique element of the vector space k^0.

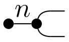
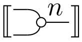
We call $\overrightarrow{\mathsf{Circ}}$ the prop freely generated from the generators in (5) and $\overleftarrow{\mathsf{Circ}}$ the one freely generated from (6). The semantics of circuits in $\overrightarrow{\mathsf{Circ}}$ can be thought of as functions taking inputs on left ports and giving output on the right ports, with the intuition for the generators as given in the Introduction. Symmetrically, the semantics of circuits in $\overleftarrow{\mathsf{Circ}}$ are functions with inputs on the right ports and outputs on the left. The semantics of an arbitrary circuit in Circ is, in general, a relation.

▶ **Example 5.** Two circuits will play a special role in our exposition:  and . Using the definition of $[\![\cdot]\!]$, it is immediate to see that their semantics forces the two ports on the right (resp. left) to carry the same value.

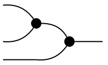
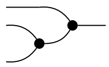
$[\![$$]\!] = \{(\bullet, \begin{pmatrix} x \\ x \end{pmatrix}) \mid x \in \mathsf{k}\}$ $\qquad$ $[\![$$]\!] = \{(\begin{pmatrix} x \\ x \end{pmatrix}, \bullet) \mid x \in \mathsf{k}\}$

Using these diagrams (along with and) one defines for each $n \in \mathbb{N}$, : $0 \to n+n$ and : $n+n \to 0$ with semantics $\{(\begin{pmatrix} x \\ x \end{pmatrix}, \bullet) \mid x \in \mathsf{k}^n\}$ and $\{(\bullet, \begin{pmatrix} x \\ x \end{pmatrix}) \mid x \in \mathsf{k}^n\}$. These circuits give rise, modulo the axioms that we will illustrate later, to a self-dual compact closed structure. See [13, Sec.5.1] for full details. As for identities and symmetries, also for  and  we will sometimes omit n for readability. Given an arbitrary circuit $c\colon n \to m$, its *opposite* circuit $c^{op}\colon m \to n$ is defined as illustrated below. It is easy to see that c^{op} denotes the opposite relation of $[\![c]\!]$, i.e., $[\![c^{op}]\!] = \{(y,x) \in \mathsf{k}^m \times \mathsf{k}^n \mid (x,y) \in [\![c]\!]\}$.



As for  above, one can define the n-version of each of the generators in (5) and (6) (as well as generators (8) and (10) that we shall introduce later). For instance $[\![$$]\!] = \{(\begin{pmatrix} x \\ y \end{pmatrix}, x+y) \mid x,y \in \mathsf{k}^n\}$. When clear from the context, we will omit the n.

A sound and complete axiomatisation for semantic equality was developed in [11, 2, 32], and in [6] for inclusion. The above signature together with the axioms, recalled in Figure 1, form the theory of Interacting Hopf Algebras. The resulting prop is denoted by $\mathbb{IH}_{\mathsf{k}}$.

▶ Remark 6. Thanks to the compact closed structure, each of the axioms and laws that we prove in the text can be read both as $c \overset{\mathbb{IH}}{=} d$ and $c^{op} \overset{\mathbb{IH}}{=} d^{op}$. For example, by $\bullet$*–coas* we also know that  $\overset{\mathbb{IH}}{=}$ .

▶ **Theorem 7.** *For all circuits c, d in* Circ*, $[\![c]\!] \subseteq [\![d]\!]$ if and only if $c \overset{\mathbb{IH}}{\subseteq} d$.*

We now come to expressivity: which relations in $\mathsf{Rel}_{\mathsf{k}}$ are expressed by Circ? The answer is that Circ captures exactly $\mathsf{LinRel}_{\mathsf{k}}$ (see Definition 2).

▶ **Theorem 8.** $\mathbb{IH}_{\mathsf{k}} \cong \mathsf{LinRel}_{\mathsf{k}}$.

The above result means that $\mathbb{IH}_{\mathsf{k}}$ is the theory of linear relations. It is convenient to recall from [27] a useful fact: circuits in $\overrightarrow{\mathsf{Circ}}$ express exactly k-matrices, as illustrated below:

▶ **Example 9.** Consider the circuit $c\colon 3 \to 4$ below and its representation as a 4×3 matrix. Note that $A_{ij} = k$ whenever k is the scalar encountered on the path from the ith port to the jth port. If there is no path, then $A_{ij} = 0$. It is easy to check that $[\![c]\!] = \{(x,y) \in \mathsf{k}^3 \times \mathsf{k}^4 \mid y = Ax\}$.

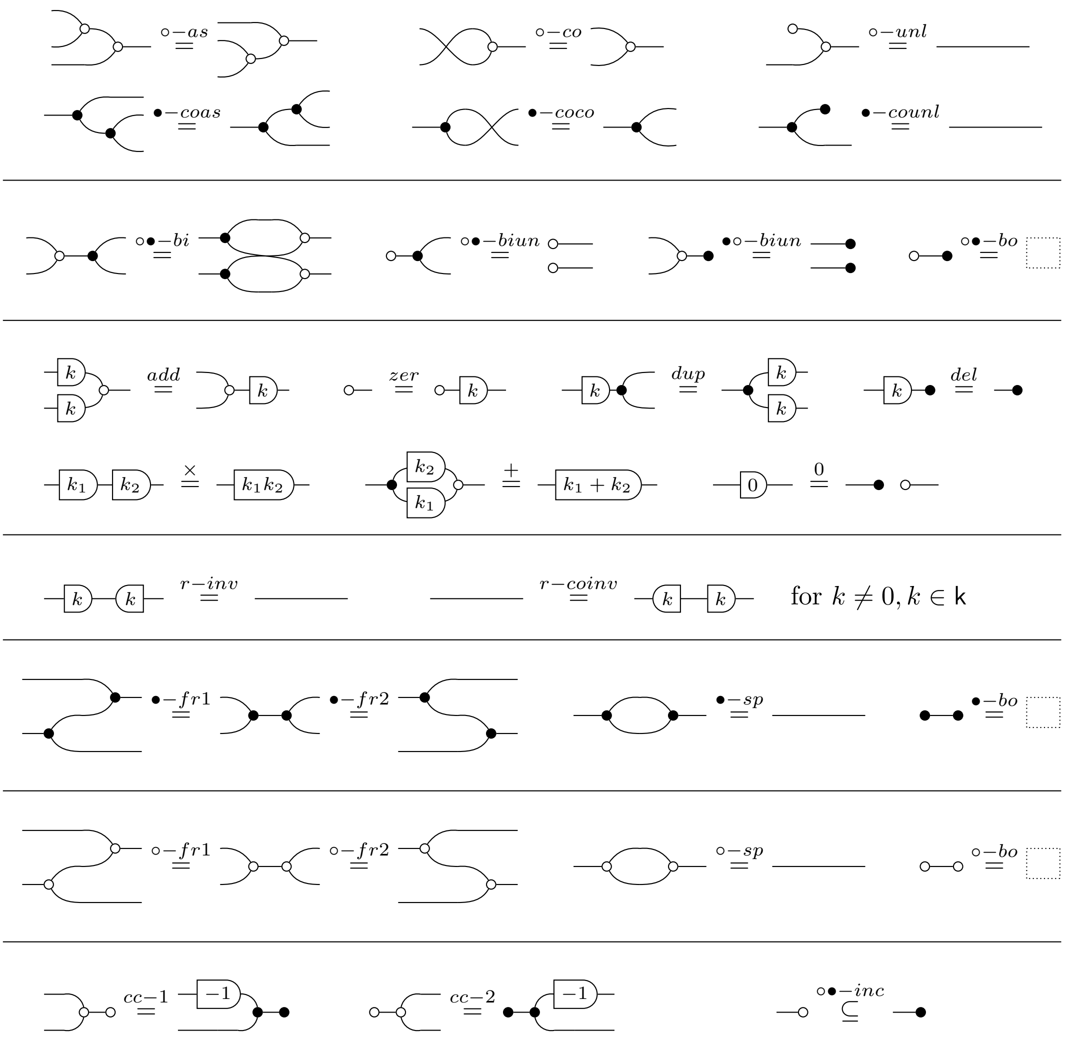
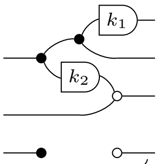
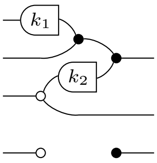


Figure 1 Axioms of Interacting Hopf Algebras ($\mathbb{IH}_{\mathsf{k}}$).

$c =$ 

$$A = \begin{pmatrix} k_1 & 0 & 0 \\ 1 & 0 & 0 \\ k_2 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

$d =$ 

Dually, circuits in $\overleftarrow{\mathsf{Circ}}$ are "reversed" matrices: inputs on the right and outputs on the left. For instance $d\colon 4 \to 3$ again encodes A, but its semantics is $[\![d]\!] = \{(y, x) \in \mathsf{k}^4 \times \mathsf{k}^3 \mid y = Ax\}$.

4 The Theory of Polyhedral cones

Hereafter, we assume k to be an *ordered field*, namely a field equipped with a total order $\leq$ such that for all $i, j, k \in \mathsf{k}$: (a) if $i \leq j$, then $i + k \leq j + k$; (b) if $0 \leq i$ and $0 \leq j$, then $0 \leq ij$.

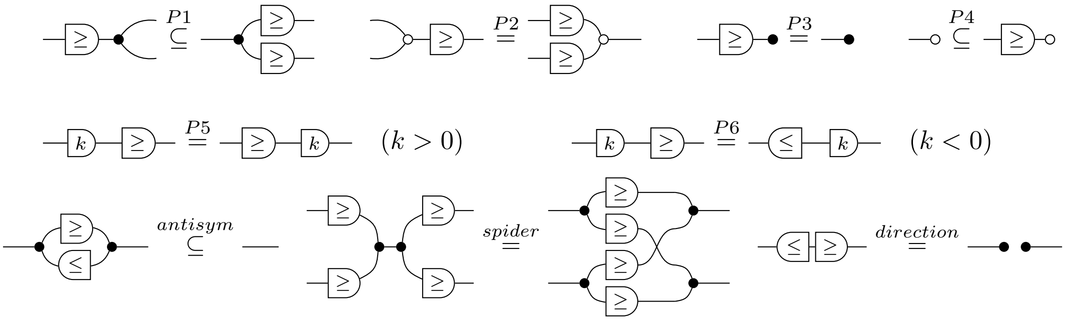
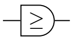


Figure 2 Axioms of $\mathbb{IH}_{\mathsf{k}}^{\geq}$.

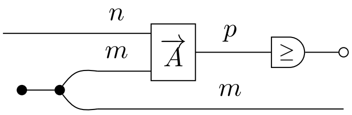
We extend the signature in (5) and (6) with the following generator



(8)

and denote the resulting free prop $\mathsf{Circ}^{\geq}$. The morphism $[\![\cdot]\!] : \mathsf{Circ}^{\geq} \to \mathsf{Rel}_{\mathsf{k}}$ behaves as (7) for the generators in (5) and (6), whereas for $-\!\boxed{\geq}\!-$, it is defined as hinted by our syntax: $[\![-\!\boxed{\geq}\!-]\!] = \{\, (x, y) \mid x, y \in \mathsf{k}, x \geq y \,\}$.

▶ **Example 10.** Let $\overset{n+m}{\text{———}}\boxed{\vec{A}}\overset{p}{\text{——}}$ be a diagram in $\overrightarrow{\mathsf{Circ}}$ denoting some matrix A (see Example 9). Consider the following circuit in $\mathsf{Circ}^{\geq}$



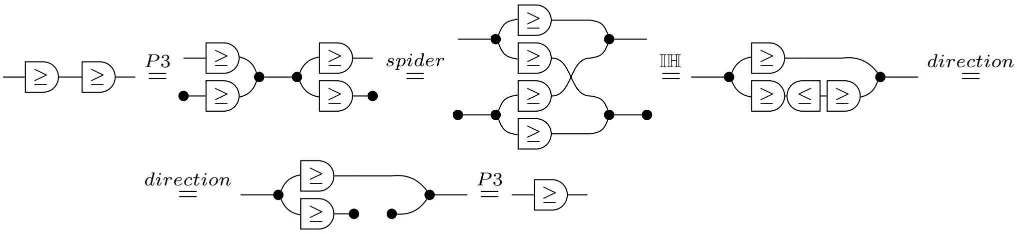
(9)

It easy to check that its semantics is the relation $C = \{(x, y) \in \mathsf{k}^n \times \mathsf{k}^m \mid A \begin{pmatrix} x \\ y \end{pmatrix} \geq 0\}$. Thus (9) denotes a *polyhedral cone*, i.e., the set of solutions of some system of linear inequations.

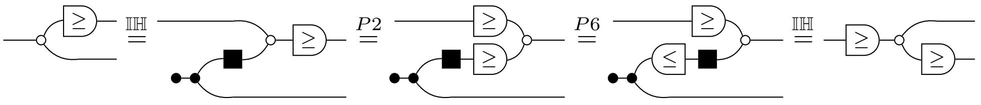
We denote by $\mathbb{IH}_{\mathsf{k}}^{\geq}$ the prop generated by the theory consisting of this signature (namely, (5), (6) and (8)), the axioms of $\mathbb{IH}_{\mathsf{k}}$ and the axioms in Figure 2, where $-\!\boxed{\leq}\!-$ is just $-\!\boxed{\geq}\!-^{op}$ (see Example 5). The first two rows of axioms describe the interactions of $-\!\boxed{\geq}\!-$ with the generators in (5). The third row asserts that $\geq$ is antisymmetric and satisfies an appropriate spider condition. In the last axiom, the right-to-left inclusion states that for all $k, l \in \mathsf{k}$, there exists an upper bound u, i.e. $u \geq k$, $u \geq l$. The left-to-right inclusion is redundant.

The axioms are perhaps surprising: e.g. reflexivity and transitivity are not included. As a taster for working with the diagrammatic calculus, we prove these properties below.

▶ Remark 11. We will often use the alternative antipode notation $-\!\blacksquare\!-$ for the scalar $-\!\boxed{-1}\!-$.

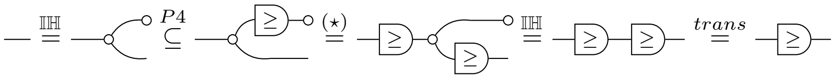


The derivation above proves transitivity. The following derivation



(⋆)

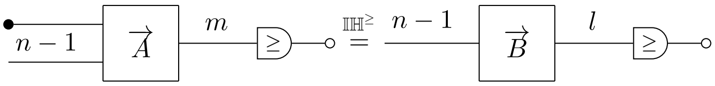
is used below to show that $\geq$ is reflexive:



▶ Remark 12. When we annotate equalities with $\mathbb{IH}$, we are making use of multiple unmentioned derived laws presented in related works. These can be seen to hold also by appealing to Theorem 7.

Routine computations confirm that all the axioms are sound. To prove completeness, we give diagrammatic proofs of several well-known results.

▶ **Proposition 13** (Fourier-Motzkin elimination). *For each arrow $A\colon n \to m$ of $\overrightarrow{\mathsf{Circ}}$, there exists an arrow $B\colon n-1 \to l$ of $\overrightarrow{\mathsf{Circ}}$ such that*



The proof, in [5, Appendix B.1], mimics the Fourier-Motzkin elimination, an algorithm for eliminating variables from a system of linear inequations (i.e. projecting a polyhedral cone).

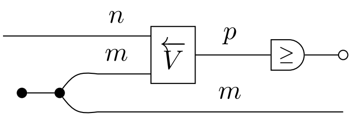
The next step is a normal form theorem. A circuit $c\colon n \to m$ of $\mathsf{Circ}^{\geq}$ is said to be in *polyhedral normal form* if there is an arrow $\xrightarrow{n+m}\boxed{\overrightarrow{A}}\xrightarrow{p}$ of $\overrightarrow{\mathsf{Circ}}$, such that $c = (9)$.

▶ **Theorem 14** (First Normal Form). *For each arrow $c\colon n \to m$ of $\mathsf{Circ}^{\geq}$, there is another arrow $d\colon n \to m$ of $\mathsf{Circ}^{\geq}$ in polyhedral normal form such that $c \overset{\mathbb{IH}^{\geq}}{=} d$.*

The proof is by induction on the structure of $\mathsf{Circ}^{\geq}$. The only challenging case is sequential composition, which uses the Fourier-Moztkin elimination. Details are in [5, Appendix B.2].

Diagrams in $\mathsf{Circ}^{\geq}$ enjoy a second normal form: an arrow $c\colon n \to m$ of $\mathsf{Circ}^{\geq}$ is said to be in *finitely generated normal form* if there is an arrow $\xrightarrow{n+m}\boxed{\overleftarrow{V}}\xrightarrow{p}$ of $\overleftarrow{\mathsf{Circ}}$, such that

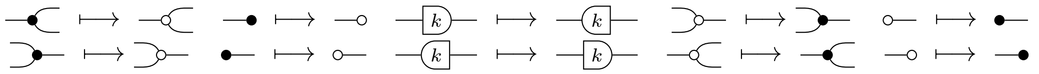
$c =$

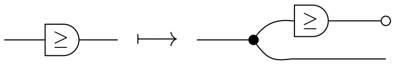


As recalled in Example 9, the semantics of the circuit in $\overleftarrow{\mathsf{Circ}}$ is $\left[\!\left[\xrightarrow{n+m}\boxed{\overleftarrow{V}}\xrightarrow{p} \right]\!\right] =$ $\{(u,z) \in \mathsf{k}^{n+m} \times \mathsf{k}^{p} \mid u = Vz\}$ for some $(n+m) \times p$ matrix V. Then, it is easy to check that $[\![c]\!] = \{(x,y) \in \mathsf{k}^{n} \times \mathsf{k}^{m} \mid \exists z \in \mathsf{k}^{p} \text{ s.t. } \begin{pmatrix} x \\ y \end{pmatrix} = Vz, z \geq 0\}$. The matrix V can be regarded as a set of column vectors $\{v_1, \ldots, v_p\}$ and $[\![c]\!]$ as the *conic combination* of those vectors, defined as $\mathsf{cone}(V) = \{z_1 v_1 + \ldots + z_p v_p \mid z_i \in \mathsf{k},\ z_i \geq 0\}$. Sets of vectors generated in this way are known as *finitely generated cones*, which justifies the name of the normal form.

To prove the existence of this normal form, we introduce the polar operator, an important construction in convex analysis which, in our approach, has a simple inductive definition.

▶ **Definition 15.** *The polar operator* $\cdot^\circ\colon \mathsf{Circ}^{\geq} \to \mathsf{Circ}^{\geq}$ *is the functor inductively defined as:*





$(c\,;d)^\circ = c^\circ\,;d^\circ \quad (c \oplus d)^\circ = c^\circ \oplus d^\circ$

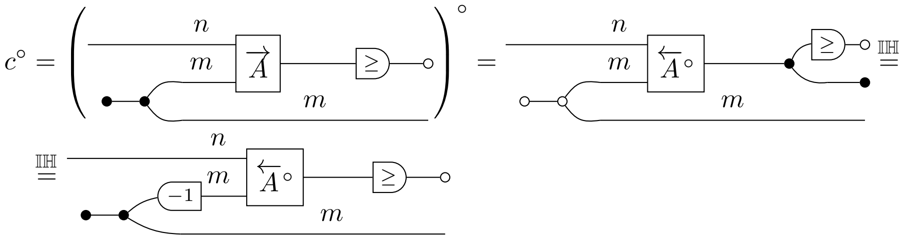
The polar operator enjoys the following useful properties.

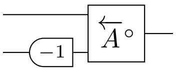
▶ **Proposition 16.** *For all arrows* $c, d\colon n \to m$ *in* $\mathsf{Circ}^{\geq}$*, it holds that*

1. *if* $c \overset{\mathbb{IH}^{\geq}}{\subseteq} d$ *then* $(d)^\circ \overset{\mathbb{IH}^{\geq}}{\subseteq} (c)^\circ$*;*
2. $(c^\circ)^\circ \overset{\mathbb{IH}^{\geq}}{=} c$*;*
3. *if* c *is an arrow of* $\overrightarrow{\mathsf{Circ}}$*, then* c° *is an arrow of* $\overleftarrow{\mathsf{Circ}}$*.*

▶ **Proposition 17.** *For each arrow* $c\colon n \to m$ *of* $\mathsf{Circ}^{\geq}$ *in polyhedral normal form there is an arrow* $d\colon n \to m$ *of* $\mathsf{Circ}^{\geq}$ *in finitely generated normal form, such that* $(c)^\circ \overset{\mathbb{IH}^{\geq}}{=} d$*.*

Proof. Let c as in (9). Then, by applying the definition of $\cdot^\circ$



which is in finitely generated normal form, since  is by Proposition 16 in $\overleftarrow{\mathsf{Circ}}$. ◀

▶ **Theorem 18** (Second Normal Form)**.** *For each arrow* $c\colon n \to m$ *of* $\mathsf{Circ}^{\geq}$*, there is an arrow* $d\colon n \to m$ *of* $\mathsf{Circ}^{\geq}$ *in finitely generated normal form such that* $c \overset{\mathbb{IH}^{\geq}}{=} d$*.*

Proof. By Theorem 14, there exists an arrow $p\colon n \to m$ of $\mathsf{Circ}^{\geq}$ in polyhedral normal form, such that $c^\circ \overset{\mathbb{IH}^{\geq}}{=} p$. By Proposition 16.1, $p^\circ \overset{\mathbb{IH}^{\geq}}{=} (c^\circ)^\circ$ and, by Proposition 16.2 , $(c^\circ)^\circ \overset{\mathbb{IH}^{\geq}}{=} c$, thus $p^\circ \overset{\mathbb{IH}^{\geq}}{=} c$. Since p is in polyhedral normal form, by Proposition 17, there exists a circuit d in finitely generated normal form such that $d \overset{\mathbb{IH}^{\geq}}{=} p^\circ \overset{\mathbb{IH}^{\geq}}{=} c$. ◀

An immediate consequence of the two normal form theorems is the well-known Weyl-Minkowski theorem, which states that every polyhedral cone is finitely generated and, vice-versa, every finitely generated cone is polyhedral. It is worth emphasising that neither the polyhedral nor the finitely generated normal form are unique: different matrices may give rise to the same cone. However, with the finitely generated normal form, proving completeness requires only a few more lemmas, which are given in [5, Appendix B.4].

▶ **Theorem 19** (Completeness)**.** *For all circuits* $c, d \in \mathsf{Circ}^{\geq}$*, if* $[\![c]\!] \subseteq [\![d]\!]$ *then* $c \overset{\mathbb{IH}^{\geq}}{\subseteq} d$*.*

We now come to the problem of expressivity: what is the image of $\mathsf{Circ}^{\geq}$ through $[\![\cdot]\!]$? It turns out that $\mathsf{Circ}^{\geq}$ denotes exactly the arrows of $\mathsf{PC_k}$ (see Definition 2).

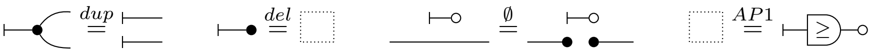


Figure 3 Axioms of $\mathsf{a}\mathbb{IH}^{\geq}_{\mathsf{k}}$.

▶ **Proposition 20** (Expressivity). *For each arrow $C\colon n \to m$ in PC_{k} there exists a circuit $c\colon n \to m$ of $\mathsf{Circ}^{\geq}$, such that $C = [\![c]\!]$. Vice-versa, for each circuit $c\colon n \to m$ of $\mathsf{Circ}^{\geq}$ there exists an arrow $C\colon n \to m$ of PC_{k}, such that $[\![c]\!] = C$.*

By Theorem 19 and Proposition 20 it follows that

▶ **Corollary 21.** $\mathbb{IH}^{\geq}_{\mathsf{k}} \cong \mathsf{PC}_{\mathsf{k}}$

The above allows us to conclude that $\mathbb{IH}^{\geq}_{\mathsf{k}}$ is the theory of polyhedral cones.

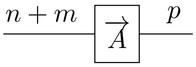
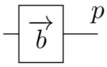
5 The theory of Polyhedra

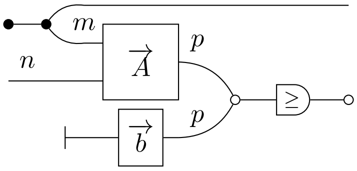
In [9], the signature of Circ was extended with an additional generator

$$\vdash\!\!\!- \qquad (10)$$

with semantics $[\![\vdash\!\!\!-]\!] = \{(\bullet, 1)\}$. The three leftmost equations in Figure 3 provide a complete axiomatisation for semantic equality. In terms of expressivity, the resulting calculus expresses exactly *affine spaces*, namely sets of solutions of $Ax + b = 0$ for some matrix A and vector b.

Here we extend $\mathsf{Circ}^{\geq}$ with (10). Let $\mathsf{ACirc}^{\geq}$ be the prop freely generated by (5), (6), (8) and (10). The circuits of $\mathsf{ACirc}^{\geq}$ can denote *polyhedra*, namely sets $P = \{\, x \in \mathsf{k}^n \mid Ax + b \geq 0 \,\}$. Observe that the empty set $\emptyset$ is a polyhedron, but not a polyhedral cone.

▶ **Example 22.** Let  and  be circuits in $\overrightarrow{\mathsf{Circ}}$ denoting, respectively some matrix A and some vector b. Consider the following circuit in $\mathsf{ACirc}^{\geq}$.



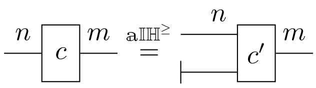
(11)

It is easy to check that its semantics is the relation $P = \{\, (x,y) \in \mathsf{k}^n \times \mathsf{k}^m \mid A \begin{pmatrix} y \\ x \end{pmatrix} + b \geq 0 \,\}$. Another useful circuit is $\vdash\!\!-\!\!\circ$: $[\![\vdash\!\!-\!\!\circ]\!] = \{(\bullet, 1)\};\{(0, \bullet)\} = \emptyset$. Intuitively, it behaves as a logical false, since for any relation R in $\mathsf{Rel}_{\mathsf{k}}$, $R \oplus \emptyset = \emptyset = \emptyset \oplus R$.

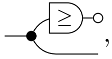
In order to obtain a complete axiomatisation, it is enough to add to the three axioms in [9], only one axiom: $AP1$ in Figure 3. Intuitively, $AP1$ states that $1 \geq 0$. The prop freely generated by (5), (6), (8), (10) and the axioms in Figures 1, 2 and 3 is denoted by $\mathsf{a}\mathbb{IH}^{\geq}_{\mathsf{k}}$.

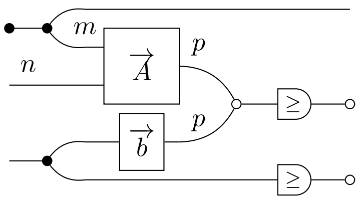
With these axioms, any circuit in $\mathsf{ACirc}^{\geq}$ can be shown equivalent to one of the form (11). This is shown using the first normal form (Theorem 14) for $\mathsf{Circ}^{\geq}$ and the following lemma.

▶ **Lemma 23.** *For any $c\colon n \to m$ of $\mathsf{ACirc}^{\geq}$, there exists $c'\colon n + 1 \to m$ of $\mathsf{Circ}^{\geq}$ such that*



▶ **Theorem 24.** *For all c of $\mathsf{ACirc}^{\geq}$, there exist d in the form of* (11) *such that $c \overset{\mathrm{a}\mathbb{IH}^{\geq}}{=} d$.*

To prove completeness, the notion of homogenization is pivotal. The *homogenization* of a polyhedron $P = \{x \in \mathsf{k}^n \mid Ax + b \geq 0\}$ is the cone $P^H = \{(x, y) \in \mathsf{k}^{n+1} \mid Ax + by \geq 0, y \geq 0\}$. Diagrammatically, this amounts to replace the $\vdash\!\!-$ in (11) with , obtaining the following diagram



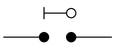
(12)

▶ **Lemma 25.** *Let $P_1, P_2 \subseteq \mathsf{k}^n$ be two non-empty polyhedra. Then, $P_1 \subseteq P_2$ iff $P_1^H \subseteq P_2^H$.*

Using Theorem 24 and Lemma 25, we can reduce completeness for *non-empty* polyhedra to completeness of polyhedral cones.

▶ **Theorem 26.** *Let $c, d \colon n \to m$ in $\mathsf{ACirc}^{\geq}$ denote non-empty polyhedra. If $[\![c]\!] \subseteq [\![d]\!]$, then $c \overset{\mathrm{a}\mathbb{IH}^{\geq}}{\subseteq} d$.*

Completeness for empty polyhedra requires a few additional lemmas, given in [5, Appendix D].

▶ **Theorem 27.** *For all circuits c in $\mathsf{ACirc}^{\geq}$, if $[\![c]\!] = \emptyset$ then $c \overset{\mathrm{a}\mathbb{IH}^{\geq}}{=}$* 

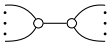
▶ **Corollary 28** (Completeness)**.** *For all circuits c, d in $\mathsf{ACirc}^{\geq}$, if $[\![c]\!] \subseteq [\![d]\!]$ then $c \overset{\mathrm{a}\mathbb{IH}^{\geq}}{\subseteq} d$.*

Finally, we characterise the semantic image of circuits in $\mathsf{ACirc}^{\geq}$.

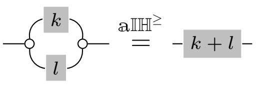
▶ **Proposition 29** (Expressivity)**.** *For each arrow $P \colon n \to m$ in $\mathsf{P_k}$ there exist a circuit $c \colon n \to m$ in $\mathsf{ACirc}^{\geq}$, such that $P = [\![c]\!]$. Vice-versa, for each circuit $c \colon n \to m$ of $\mathsf{ACirc}^{\geq}$ there exists an arrow $P \colon n \to m$ of $\mathsf{P_k}$, such that $[\![c]\!] = P$.*

Indeed, $\mathrm{a}\mathbb{IH}^{\geq}_{\mathsf{k}}$ is the theory of polyhedra:

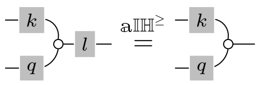
▶ **Corollary 30.** $\mathrm{a}\mathbb{IH}^{\geq}_{\mathsf{k}} \cong \mathsf{P_k}$

▶ **Example 31** (Flow networks)**.** Consider again flow networks, previously mentioned in the Introduction: edges with capacity k can be expressed in $\mathsf{ACirc}^{\geq}$ by the diagram in (3) hereafter referred as $-\boxed{k}-$. Observe that $[\![-\boxed{k}-]\!] = \{(x, x) \mid 0 \leq x \leq k\}$ is exactly the expected meaning of an edge in a flow network. Nodes with n incoming edges and m outgoing edges can be encoded by the diagram n  m. Again the semantics is the expected one: the total incoming flow must be equal to the total outgoing flow. For an example of the encoding, check the flow network in (1) and the corresponding diagram in (2).

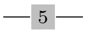
The axioms in $\mathrm{a}\mathbb{IH}^{\geq}_{\mathsf{k}}$ can be exploited to compute the maximum flow of a network. By using the following two derived laws (proved in [5, Appendix E])

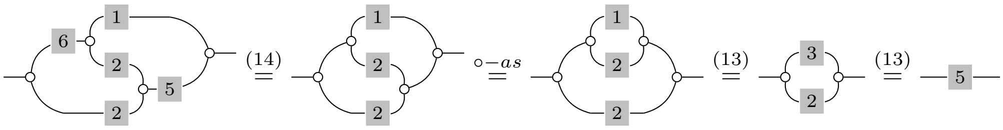


(13)



$(k + q \leq l)$ (14)

one can transform the diagram in (2) into 



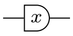
meaning that $[\![(2)]\!] = \{(x, x) \mid 0 \leq x \leq 5\}$, i.e., the maximum flow of (2) is exactly 5.

6 Adding states to polyhedra

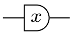
We have shown that $\mathsf{ACirc}^{\geq}$ with its associated equational theory $\mathsf{a}\mathbb{IH}^{\geq}_{\mathsf{k}}$ provides a sound and complete calculus for polyhedra. In this section, we extend the calculus with a canonical notion of *state*. Our development follows step-by-step the general recipe illustrated in [8, §4].

6.1 The Calculus of Stateful Polyhedral Processes

We call SPP the prop freely generated by (5), (6), (8), (10) and the following.

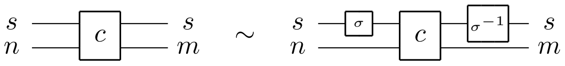


(15)

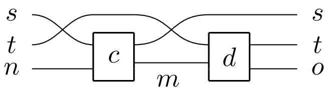
Intuitively, the register  is a synchronous buffer holding a value $k \in \mathsf{k}$: when it receives $l \in \mathsf{k}$ on the left port, it emits k on the right one and stores l. To give a formal semantics to such behaviour we exploit a "state bootstrapping" technique that appears in several places in the literature, e.g. in the setting of cartesian bicategories [25] and geometry of interaction [22].

▶ **Definition 32** (Stateful processes [25]). *Let T be a prop. Define $\mathsf{St}(\mathsf{T})$ as the prop where:*

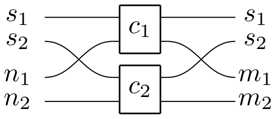
- *morphisms $n \to n$ are pairs (s, c) where $s \in \mathbb{N}$ and $c \colon s + n \to s + m$ is a morphism of T, quotiented by the smallest equivalence relation including every instance of*

  for a permutation $\sigma{:}s \to s$; the order is defined as $(s, c) \overset{\mathsf{St}(\mathsf{T})}{\subseteq} (s, d)$ if and only $c \overset{\mathsf{T}}{\subseteq} d$.
- *the composition of $(s, c) \colon n \to m$ and $(t, d) \colon m \to o$ is $(s + t, e)$ where e is the arrow of T given by*

- *the monoidal product of $(s_1, c_1) \colon n_1 \to m_1$ and $(s_2, c_2) \colon n_2 \to m_2$ is $(s_1 + s_2, e)$ where e is given by*

- *the identity on n is $(0, id_n)$ and the symmetry of n, m is $(0, \sigma_{n,m})$.*

We use $\mathsf{St}(\mathsf{Rel_k})$ as our semantic domain: in an arrow $(s, R) \colon n \to m$, s records the number of registers while $R \colon s + n \to s + m$ is a relation $R \subseteq \mathsf{k}^s \times \mathsf{k}^n \times \mathsf{k}^s \times \mathsf{k}^m$ containing quadruples (u, l, v, r) representing transitions: u and v are the starting and arrival state (namely vectors in k^s, holding a value in k to each of the s registers), while l and r are vectors of values occurring on the left and the right ports. The equivalence relation $\sim$ ensures that registers remains anonymous: it equates arrows that only differ by a bijective relabelling of their lists of registers. This is, therefore, a syntactic form of equivalence similar in flavour to α-equivalence, since it discards intentional details not relevant for the dynamics of processes.

We can now give the semantics of SPP as the morphism $\langle\!\langle\cdot\rangle\!\rangle\colon \mathsf{SPP} \to \mathsf{St}(\mathsf{Rel}_\mathsf{k})$ defined:

$$\langle\!\langle -\!\boxed{x}\!- \rangle\!\rangle = (1, \{(k,l,l,k) \mid l,k \in \mathsf{k}\}) \qquad \text{and} \qquad \langle\!\langle o \rangle\!\rangle = (0, [\![o]\!])$$

for all generators o in (5), (6), (8), (10). For instance $\langle\!\langle -\!\!\bullet \rangle\!\rangle = (0, \{(\bullet, k) \mid k \in \mathsf{k}\})$. The semantics of $-\!\boxed{x}\!-$ is the expected behaviour: from any state k (the stored value), it makes a transition to state l when l is on the left port and k is on the right. This can be restated as a structural operational semantics (sos) axiom $(-\!\boxed{x}\!-, k) \xrightarrow[k]{l} (-\!\boxed{x}\!-, l)$ where the labels above and under the arrow stand, respectively, for the values on the left and right ports.

Theorem 30 in [8] ensures that no other data is needed for an axiomatisation: let $\mathbb{S}\mathrm{a}\mathbb{IH}^{\geq}$ be the prop generated by (5), (6), (8), (10), (15) and the axioms in Figures 1, 2 and 3.

▶ **Theorem 33.** *For all c, d in* SPP*, if $\langle\!\langle c\rangle\!\rangle \subseteq \langle\!\langle d\rangle\!\rangle$ then $c \overset{\mathbb{S}\mathrm{a}\mathbb{IH}^{\geq}}{\subseteq} d$. Moreover $\mathbb{S}\mathrm{a}\mathbb{IH}^{\geq} \cong \mathsf{St}(\mathsf{P}_\mathsf{k})$.*

Proof of Theorem 33. We make more clear the correspondence with [8]. Considering the following diagram.

$$\begin{tikzcd}
\mathsf{SPP} \arrow[r, two heads, "q"] \arrow[rrrr, bend left, "\langle\!\langle\cdot\rangle\!\rangle"] & \mathbb{S}\text{Ә}\mathbb{IH}^{\geq} \arrow[r, "F"] & \mathsf{St}(\mathrm{a}\mathbb{IH}^{\geq}_{\mathsf{k}}) \arrow[r, "\mathsf{St}(\cong)"] & \mathsf{St}(\mathsf{P}_\mathsf{k}) \arrow[r, tail, "\mathsf{St}(\iota)"] & \mathsf{St}(\mathsf{Rel}_\mathsf{k})
\end{tikzcd}$$

The morphism $\mathsf{St}(\iota)$ is just the obvious extension of the inclusion $\iota\colon \mathsf{P}_\mathsf{k} \to \mathsf{Rel}_\mathsf{k}$. Similarly, $\mathsf{St}(\cong)$ is the extension of the isomorphism shown in Corollary 30. The morphism q is just the obvious quotient from SPP to $\mathbb{S}\mathrm{a}\mathbb{IH}^{\geq}$. The interesting part is provided by the morphism $F\colon \mathbb{S}\mathrm{a}\mathbb{IH}^{\geq} \to \mathsf{St}(\mathrm{a}\mathbb{IH}^{\geq}_{\mathsf{k}})$ defined in [8, §4.1]: take T as $\mathrm{a}\mathbb{IH}^{\geq}_{\mathsf{k}}$ and $\mathsf{T} + \mathsf{X}$ as $\mathbb{S}\mathrm{a}\mathbb{IH}^{\geq}$. By Theorem 30 in [8], since $\mathrm{a}\mathbb{IH}^{\geq}_{\mathsf{k}}$ is compact closed, then F is an isomorphism of props. To see that it is an isomorphism of ordered props, it is immediate to check that both F and its inverse G defined in [8, §4.1] preserves the order. ◀

We conclude by observing the semantics can be presented with intuitive sos rules. Indeed, the same rules as in [8, §2] – interpreted over a field rather than the naturals – and:

$$\frac{x \geq y}{(-\!\boxed{\geq}\!-, \bullet) \xrightarrow[y]{x} (-\!\boxed{\geq}\!-, \bullet)} \qquad (\vdash\!\!-, \bullet) \xrightarrow[1]{\bullet} (\vdash\!\!-, \bullet)$$

This diagrammatic language is, therefore, similar in flavour to traditional process calculi, and we call it the *calculus of stateful polyhedral processes.* Theorem 33 affirms that it expresses exactly the stateful polyhedral processes.

6.2 Bounded Continuous Petri Nets

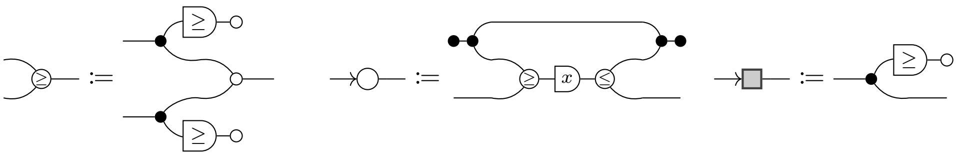
Hereafter k is fixed to be the real numbers $\mathbb{R}$ and the set of non-negative reals is denoted by $\mathbb{R}_+ = \{\, r \in \mathbb{R} \mid r \geq 0 \,\}$. A *continuous* Petri net [18] differs from a (discrete) Petri net in that:

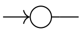
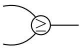
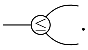
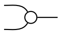
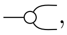
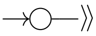
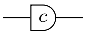
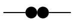
- markings are real valued – that is, places hold a non-negative real number of tokens,
- transitions can consume and produce non-negative real numbers of tokens,
- transitions can be fired a non-negative real number amount of times – for example a transition can be fired 0.5 times, producing and consuming half the tokens.

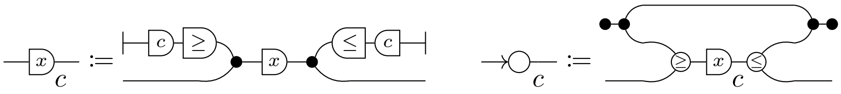
▶ **Definition 34** (Continuous Petri nets and their semantics). *A Petri net $\mathcal{P} = (P, T, {}^\circ-, -^\circ)$ consists of a finite set of places P, a finite set of transitions T, and functions ${}^\circ-, -^\circ : T \to {\mathbb{R}_+}^P$. Given $\mathbf{y}, \mathbf{z} \in {\mathbb{R}_+}^P$, we write $\mathbf{y} \to \mathbf{z}$ if there exists $\mathbf{t} \in {\mathbb{R}_+}^T$ such that ${}^\circ\mathbf{t} \leq \mathbf{y}$ and $\mathbf{z} = \mathbf{y} - {}^\circ\mathbf{t} + \mathbf{t}^\circ$, where ${}^\circ\mathbf{t}$ and $\mathbf{t}^\circ$ are the evident liftings of ${}^\circ()$ and $()^\circ$, e.g. ${}^\circ\mathbf{t}(p) = \sum_{s\in T} \mathbf{t}(s) \cdot {}^\circ s(p)$. The (step) operational semantics of $\mathcal{P}$ is the relation $\langle \mathcal{P} \rangle = \{(\mathbf{y}, \mathbf{z}) \mid \mathbf{y} \to \mathbf{z}\} \subseteq {\mathbb{R}_+}^P \times {\mathbb{R}_+}^P$.*

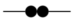
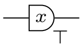
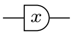
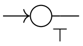
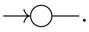
As for ordinary Petri nets, one can consider *bounded nets*: each place has a maximum capacity $c \in \mathbb{R}_+ \cup \{\top\}$: a place with capacity $\top$ is unbounded. The above definition is therefore extended with a boundary function $\mathbf{b} \in (\mathbb{R}_+ \cup \{\top\})^P$ and the transition relation $\mathbf{y} \to \mathbf{z}$ is modified by additionally requiring that $\mathbf{y}, \mathbf{z} \leq \mathbf{b}$. Since $r \leq \top$ for all $r \in \mathbb{R}_+$, continuous Petri nets are instances of bounded continuous nets where every place is unbounded.

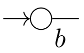
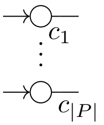
To encode continuous Petri nets and their bounded variant as stateful polyhedral processes, it is convenient to introduce syntactic sugar: the circuit below left is an adder that takes only positive values as inputs, the central circuit models a place, and the last one a transition.

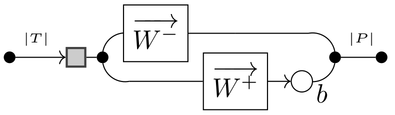


Observe that for , it is essential the use of  and and its opposite . Indeed, replacing them by ordinary adders  and , would give as semantics the whole space $\mathbb{R}^2 \times \mathbb{R}^2$, while as defined above $\langle\!\langle$  $\rangle\!\rangle = (1, \{(m, i, m - o + i, o) \mid i, o, m \in \mathbb{R}_+, o \geq m\})$, modelling exactly the expected behaviour of a place. In the diagrams below  is either a scalar $r \in \mathbb{R}_+$ or $\top =$ .



The leftmost diagram models a buffer with capacity c, while the rightmost a place with capacity c. Since $\top =$ , it holds that  $\overset{\mathbb{S}\mathsf{a}\mathbb{IH}^{\geq}}{=}$  and  $\overset{\mathbb{S}\mathsf{a}\mathbb{IH}^{\geq}}{=}$ .

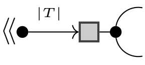
By choosing an ordering on places and transitions, the functions ${}^\circ-, -^\circ : T \to {\mathbb{R}_+}^P$ can be regarded as $\mathbb{R}_+$-matrices of type $|T| \to |P|$ and thus can be encoded as $\overrightarrow{\mathsf{Circ}}$ circuits, hereafter denoted by respectively W^- and W^+. The ordering on P also makes the boundary function b a vector $\begin{pmatrix} c_1 \\ \vdots \\ c_{|P|} \end{pmatrix}$ in $(\mathbb{R}_+ \cup \{\top\})^{|P|}$: we write  for . Any bounded Continuous Petri net $\mathcal{P}$ can be encoded as the following circuit $d_{\mathcal{P}} : 0 \to 0$ in SPP.



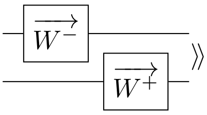
It is easy to show that $\mathcal{P}$ and $d_{\mathcal{P}}$ have the same semantics.

▶ **Proposition 35.** *For all bounded continuous Petri net $\mathcal{P}$, $\langle \mathcal{P} \rangle \sim \langle\!\langle d_{\mathcal{P}} \rangle\!\rangle$.*

Proof of Proposition 35. In order to compute $\langle\!\langle d_{\mathcal{P}} \rangle\!\rangle$, it is convenient to cut $d_{\mathcal{P}}$ in three parts. The leftmost part of $d_{\mathcal{P}}$ has the following semantics

$\langle\!\langle$  $\rangle\!\rangle = \quad (0, \{(\bullet, \bullet, \bullet, \begin{pmatrix} t \\ t \end{pmatrix}) \mid t \in {\mathbb{R}_+}^{|T|}\})$

The central part

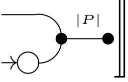
$\langle\!\langle$ 

$$\rangle\!\rangle = \quad (0\{(\bullet, \begin{pmatrix}x_1\\x_2\end{pmatrix}, \bullet, \begin{pmatrix}y_1\\y_2\end{pmatrix}) \mid W^-x_1 = y_1,\, W^+x_2 = y_2\})$$

By definition of $\langle\!\langle\cdot\rangle\!\rangle$, the composition of the two semantics above is the pair

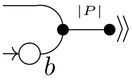
$$(0, \{(\bullet,\, \bullet,\, \bullet,\, \begin{pmatrix}y_1\\y_2\end{pmatrix}) \mid \exists t \in {\mathbb{R}_+}^{|T|} \text{ s.t. } W^-t = y_1,\, W^+t = y_2\})$$

The right-most part

$\llbracket$ 

$$\rrbracket = \quad \{(y, \begin{pmatrix}i\\o\end{pmatrix}, y - o + i, \bullet) \mid i, o, y \in {\mathbb{R}_+}^{|T|},\, o \geq y\}$$

The semantics of rightmost part is the pair

$\langle\!\langle$ 

$$\rangle\!\rangle = \quad (|P|, \{(y, \begin{pmatrix}o\\i\end{pmatrix}, y-o+i, \bullet) \mid i, o, y \in {\mathbb{R}_+}^{|P|},\, o \geq y,\, y-o+i \leq b,\, y \leq b\})$$

By composing everything we obtain $(|P|, \{(y, \bullet, y - i + o, \bullet) \mid \exists t \in {\mathbb{R}_+}^{|T|}$ s.t. $i, o, y \in {\mathbb{R}_+}^{|P|},\, o \geq y,\, y - o + i \leq b,\, y \leq b,\, W^-t = o,\, W^+t = i\})$ that is $\langle\!\langle d_{\mathcal{P}}\rangle\!\rangle = (|P|, \{(y, \bullet, z, \bullet) \mid \exists t \in {\mathbb{R}_+}^{|T|}$ s.t. $W^-t \geq y,\, y - W^-t + W^+t = z \leq b,\, y \leq b\})$.

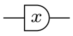
Since the equivalence is stated modulo $\sim$, then it is safe to fix an ordering on P and T. Thus, rather than considering $(y, z) \in \langle P\rangle$ as functions in ${\mathbb{R}_+}^P$ they can be regarded as vectors in ${\mathbb{R}_+}^{|P|}$. One can thus conclude by observing that $y \to z$ if and only if there exists $t \in {\mathbb{R}_+}^{|T|}$ such that $W^-t \geq y$, $y - W^-t + W^+t = z$ and $y, z \leq b$. ◀

7 Conclusions and Future Work

We have introduced the theories of polyhedral cones and the one of polyhedra. In other words, we have identified suitable sets of generators and axioms for which we proved completeness and expressivity. As side results, we get an inductive definition of the notion of polar cone, as well as an understanding of Weyl-Minkowski theorem as a normal form result.

As shown by Example 31, the theory of polyhedra allows us to represent networks with bounded resources, not expressible in $\mathbb{IH}$, and to manipulate them as symbolic expressions.

Indeed the passage from linear relations to polyhedra is a reflection of the fact that, operationally, we are able to consider several patterns of computations important in computer science, as opposed to purely linear patterns, traditionally studied in system/control theory.

For instance, as shown in §6, the addition to $\mathsf{a}\mathbb{IH}^{\geq}$ of a single generator, , directly gives us a concurrent extension of the signal flow calculus [2, 12], introduced as a compositional account for linear dynamical systems, that is expressive enough to encode continuous Petri nets [18].

References

1 Ravindra K. Ahuja, Thomas L. Magnanti, and James B. Orlin. *Network Flows: Theory, Algorithms, and Applications.* Prentice-Hall, Inc., USA, 1993.

2 John Baez and Jason Erbele. Categories in control. *Theory and Applications of Categories*, 30:836–881, 2015. arXiv:1405.6881.

3 John C. Baez. Network theory, 2014. URL: `http://math.ucr.edu/home/baez/networks/`.

4 John C Baez and Brendan Fong. A compositional framework for passive linear networks. *arXiv preprint*, 2015. `arXiv:1504.05625`.

5 Filippo Bonchi, Alessandro Di Giorgio, and Pawel Sobocinski. Diagrammatic polyhedral algebra, 2021. `arXiv:2105.10946`.

6 Filippo Bonchi, Joshua Holland, Dusko Pavlovic, and Paweł Sobociński. Refinement for signal flow graphs. In *28th International Conference on Concurrency Theory, CONCUR 2017, September 5-8, 2017, Berlin, Germany*, pages 24:1–24:16, 2017. `doi:10.4230/LIPIcs.CONCUR.2017.24`.

7 Filippo Bonchi, Joshua Holland, Robin Piedeleu, Pawel Sobocinski, and Fabio Zanasi. Diagrammatic algebra: from linear to concurrent systems. *Proc. ACM Program. Lang.*, 3(POPL):25:1–25:28, 2019. `doi:10.1145/3290338`.

8 Filippo Bonchi, Joshua Holland, Robin Piedeleu, Paweł Sobociński, and Fabio Zanasi. Diagrammatic algebra: from linear to concurrent systems. *Proceedings of the 46th ACM SIGPLAN Symposium on Principles of Programming Languages (POPL)*, 3:1–28, 2019.

9 Filippo Bonchi, Robin Piedeleu, Paweł Sobociński, and Fabio Zanasi. Graphical affine algebra. In *Proceedings of the 34th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 1–12, 2019.

10 Filippo Bonchi, Jens Seeber, and Pawel Sobocinski. Graphical Conjunctive Queries. In Dan Ghica and Achim Jung, editors, *27th EACSL Annual Conference on Computer Science Logic (CSL 2018)*, volume 119 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 13:1–13:23, Dagstuhl, Germany, 2018. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. `doi:10.4230/LIPIcs.CSL.2018.13`.

11 Filippo Bonchi, Paweł Sobociński, and Fabio Zanasi. A categorical semantics of signal flow graphs. In *Proceedings of the 25th International Conference on Concurrency Theory (CONCUR)*, pages 435–450. Springer, 2014.

12 Filippo Bonchi, Pawel Sobocinski, and Fabio Zanasi. Full abstraction for signal flow graphs. In *Proceedings of the 42nd Annual ACM SIGPLAN Symposium on Principles of Programming Languages (POPL)*, pages 515–526, 2015.

13 Filippo Bonchi, Pawel Sobocinski, and Fabio Zanasi. The calculus of signal flow diagrams I: linear relations on streams. *Information and Computation*, 252:2–29, 2017.

14 Bob Coecke and Ross Duncan. Interacting quantum observables: categorical algebra and diagrammatics. *New Journal of Physics*, 13(4):043016, 2011.

15 Bob Coecke, Ross Duncan, Aleks Kissinger, and Quanlong Wang. Strong complementarity and non-locality in categorical quantum mechanics. In *LiCS 2012*, pages 245–254, 2012.

16 Bob Coecke and Aleks Kissinger. *Picturing Quantum Processes - A first course in Quantum Theory and Diagrammatic Reasoning*. Cambridge University Press, 2017.

17 Patrick Cousot and Radhia Cousot. Abstract interpretation: A unified lattice model for static analysis of programs by construction or approximation of fixpoints. In Robert M. Graham, Michael A. Harrison, and Ravi Sethi, editors, *Conference Record of the Fourth ACM Symposium on Principles of Programming Languages, Los Angeles, California, USA, January 1977*, pages 238–252. ACM, 1977. `doi:10.1145/512950.512973`.

18 René David and Hassane Alla. *Discrete, Continuous, and Hybrid Petri Nets*. Springer, Berlin, 2 edition, 2010. `doi:10.1007/978-3-642-10669-9`.

19 Brendan Fong and David Spivak. String diagrams for regular logic (extended abstract). In John Baez and Bob Coecke, editors, *Applied Category Theory 2019*, volume 323 of *Electronic Proceedings in Theoretical Computer Science*, pages 196–229. Open Publishing Association, September 2020. `doi:10.4204/eptcs.323.14`.

20 Dan R Ghica and Achim Jung. Categorical semantics of digital circuits. In *Proceedings of the 16th Conference on Formal Methods in Computer-Aided Design (FMCAD)*, pages 41–48, 2016.

21 Nathan Haydon and Paweł Sobociński. Compositional diagrammatic first-order logic. In *11th International Conference on the Theory and Application of Diagrams (DIAGRAMS 2020)*, 2020.

22 Naohiko Hoshino, Koko Muroya, and Ichiro Hasuo. Memoryful geometry of interaction: from coalgebraic components to algebraic effects. In *Proceedings of the Joint Meeting of the Twenty-Third EACSL Annual Conference on Computer Science Logic (CSL) and the Twenty-Ninth Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, page 52. ACM, 2014.

23 Bart Jacobs and Fabio Zanasi. The logical essentials of bayesian reasoning. *CoRR*, abs/1804.01193, 2018. arXiv:1804.01193.

24 Franciscus Rebro John C. Baez, Brandon Coya. Props in network theory. *CoRR*, abs/1707.08321, 2017. arXiv:1707.08321.

25 P Katis, N Sabadini, and RFC Walters. Bicategories of processes. *Journal of Pure and Applied Algebra*, 115(2):141–178, 1997.

26 Stephen Lack. Composing PROPs. *Theory and Application of Categories*, 13(9):147–163, 2004.

27 Yves Lafont. Towards an algebraic theory of Boolean circuits. *Journal of Pure and Applied Algebra*, 184(2–3):257–310, 2003.

28 Saunders Mac Lane. Categorical algebra. *Bulletin of the American Mathematical Society*, 71:40–106, 1965.

29 Samuel J Mason. *Feedback Theory: I. Some Properties of Signal Flow Graphs*. MIT Research Laboratory of Electronics, 1953.

30 Robin Piedeleu and Fabio Zanasi. A string diagrammatic axiomatisation of finite-state automata. In *FoSSaCS 2021*, 2021.

31 Peter Selinger. A survey of graphical languages for monoidal categories. *Springer Lecture Notes in Physics*, 13(813):289–355, 2011.

32 Fabio Zanasi. *Interacting Hopf Algebras: the theory of linear systems*. PhD thesis, Ecole Normale Supérieure de Lyon, 2015.