Refinement for Signal Flow Graphs

Filippo Bonchi*[1], **Joshua Holland**[2], **Dusko Pavlovic**†[3], **and**
Paweł Sobociński‡[4]

1 Ecole Normale Supérieure, Lyon, France
2 University of Southampton, Southampton, UK
3 University of Hawaii at Manoa, Honolulu, Hawaii, US
4 University of Southampton, Southampton, UK

Abstract

The symmetric monoidal theory of Interacting Hopf Algebras provides a sound and complete axiomatisation for *linear relations* over a given field. As is the case for ordinary relations, linear relations have a natural order that coincides with *inclusion*. In this paper, we give a presentation for this ordering by extending the theory of Interacting Hopf Algebras with a single additional inequation. We show that the extended theory gives rise to an abelian bicategory—a concept due to Carboni and Walters—and highlight similarities with the algebra of relations. Most importantly, the ordering leads to a well-behaved notion of *refinement* for signal flow graphs.

1998 ACM Subject Classification F.3.1 Specifying and Verifying and Reasoning about Programs, F.3.2 Semantics of Programming Languages

Keywords and phrases Signal flow graphs, refinement, operational semantics, string diagrams, symmetric monoidal inequality theory

Digital Object Identifier 10.4230/LIPIcs.CONCUR.2017.24

1 Introduction

Signal Flow Graphs (SFGs) were introduced in the 1940s by Shannon [19] as a formal circuit model of a class of simple analog computing machines. They are a common abstraction in control theory and signal processing, used for modelling physical systems and their controllers. Nowadays, cyber-physical systems are modelled and simulated in graphical environments such as Simulink and Modelica that can be seen as great-grandchildren of SFGs.

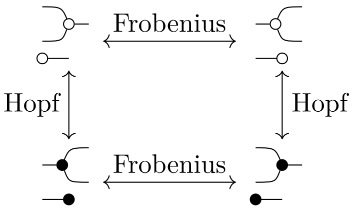
Their ubiquity is merited because SFGs serve *both* as processors of analogue signals (analytic functions) in continuous time, and as stream transducers in discrete time. The latter makes them amenable to techniques developed by computer scientists for programming language semantics. For instance Rutten [18] showed that coinduction, just as in process algebra, provides a useful proof principle for SFGs. Another example is the *signal flow calculus* [6] where SFGs are represented using string diagrammatic syntax equipped with both a structural operational semantics and a denotational semantics in terms of *linear relations*. Most importantly, denotational equality, which by full abstraction [6] coincides with observational equivalence (trace equivalence), enjoys a sound and complete axiomatization [4]. The same equational theory was independently proposed by Baez and Erbele [3].

---

[*] Partially supported by the project ANR-16-CE25-0011 REPAS and AFOSR.
[†] Partially supported by NSF and AFOSR.
[‡] Partially supported by AFOSR.

28th International Conference on Concurrency Theory (CONCUR 2017).
Editors: Roland Meyer and Uwe Nestmann; Article No. 24; pp. 24:1–24:16
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

The axiomatisation of [4, 3]—a *symmetric monoidal theory* (SMT) whose terms are typically rendered graphically as string diagrams—is the starting point of the present work. We adopt the terminology of [5]: the theory of *interacting Hopf algebras* over a ring R, denoted $\mathbb{IH}_{\mathsf{R}}$, consists of a pair of monoids (distinguished by black and white colouring) and a pair of comonoids (again, black and white). These black-white (co)monoids satisfy the equations of Frobenius and Hopf algebras, individually recalled in Examples 3 and 4, as illustrated in the schematic to the right.



(1)

A theorem in [4] states that $\mathbb{IH}_{\mathsf{R}}$ is a presentation for $\mathbf{LinRel}_{\mathsf{k}}$ the category with arrows *linear relations* (a.k.a. *additive relations*) over k, the field of fractions of R: relations that are also linear subspaces. This paved the way for an equational study of elementary linear algebra by means of string diagrams that, in [22], became *graphical linear algebra.*

Like relations, linear relations are equipped with an ordering that plays a pivotal role in many applications. It is therefore worth seeing $\mathbf{LinRel}_{\mathsf{k}}$ not as a mere category but rather as a *poset enriched* category. In this work, we provide a presentation for the underlying posetal structure of $\mathbf{LinRel}_{\mathsf{k}}$. Our main result states that it is enough to add a *single inequation*

$$-\!\circ \;\leq\; -\!\bullet \qquad (2)$$

to the equational theory of $\mathbb{IH}_{\mathsf{R}}$ in order to obtain a sound and complete axiomatization of the ordering between the arrows of $\mathbf{LinRel}_{\mathsf{k}}$. Viewed as linear relations, (2) says that the unique zero-dimensional subspace $\{0\}$ of k, considered as a vector space over itself, is a subset of the unique one-dimensional subspace. Of course, the reverse inequality does not hold.

The focus on the order sheds lights on some interesting properties of $\mathbb{IH}_{\mathsf{R}}$. We show that $\mathbb{IH}_{\mathsf{R}}$ forms an abelian bicategory [11, Def. 5.1] and that it supports operations akin to the algebra of relations [14]. Moreover, the order resolves a mystery surrounding the equational theory of interacting Hopf algebras. The system summarised in (1) is symmetric. There is no difference, equationally, between the white (co)monoid and the black (co)monoid, in spite their different meaning as linear relations: the white is the additive structure, while the black is "copying", e.g. —●< is typically the diagonal relation. Crucially, (2) breaks this symmetry.

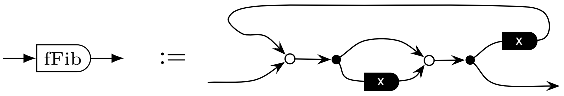
When $\mathsf{R} = \mathsf{k}[x]$ (the ring of polynomials with indeterminate x and coefficients from field k), $\mathbb{IH}_{\mathsf{R}}$ provides a sound and complete axiomatisation for SFGs [5]. The addition of (2) gives a sound and complete axiomatisation for what we call *refinement* of SFGs.

1.1 Structure of the paper

The problem of refinement of SFGs is informally explained with an example in Section 2. In Section 3 we recall the basic concepts of SMTs and, in Section 4, the theory of Interacting Hopf Algebras. In Section 5 we extend the concept of monoidal theory to handle inequations. Section 6 is devoted to proving our main result and, in Section 7, we shed light on the algebraic structure of the resulting theory, drawing parallels with relational algebra. Finally, in Section 8 we return to our motivating problem and discuss related work in Section 9.

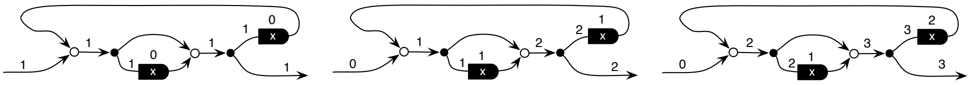
2 Fibonacci's rabbits and guinea pigs

A signal flow graph of sort (m, n), using the discrete semantics, is a stream transducer that takes m input streams and produces n output streams. For example, consider the $(1, 1)$ SFG below, which implements the well-known Fibonacci recurrence relation:



(3)

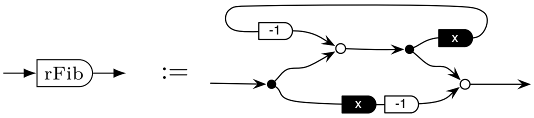
The white circles are adders (two inputs and one output), the black circles are duplicators (one input and two outputs). The 'x' gates are delays, or one-state buffers, which we assume to be initialised with zero. Given the sequence of inputs 1; 0; 0; 0; 0; 0; . . . , the output is the Fibonacci sequence 1; 2; 3; 5; 8; 13; . . . . We illustrate the first few steps below: the state of each delay is illustrated by the number above it, the remaining numbers keep track of the value on each wire at each iteration. A formal operational semantics is recalled in Section 8.



(4)

This output—according to the Fibonacci's rule [20] for rabbit reproduction—is the total number of rabbit pairs in each month, starting with a pair of rabbits (the first input is 1), and subsequently not adding nor taking away pairs (all further inputs are 0). Other inputs are possible, in this sense generalising Fibonacci; e.g. adding a pair for two months and taking away two every third month (input 1; 1; −2; 1; 1; −2; . . .) yields (1; 3; 3; 5; 10; 14; . . .).

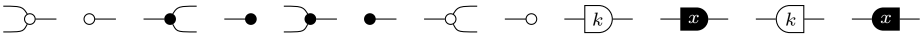
A *trace* of an (m, n) SFG c is a pair $(\boldsymbol{\alpha}, \boldsymbol{\beta})$ where $\boldsymbol{\alpha}$ is an m-tuple and $\boldsymbol{\beta}$ is the output n-tuple produced by c on $\boldsymbol{\alpha}$: e.g. (3) has (1; 0; 0; . . . , 1; 2; 3; . . .) and (1; 1; −2; . . . , 1; 3; 3; . . .) as traces. The *behaviour* of a signal flow graph is the set of all its traces. Note that behaviour is a functional *relation* on streams; in particular, if an SFG is invertible then its inverse has the opposite relation as behaviour. Here (3) *is* invertible and has the following inverse, where the '−1 gates', instances of amplifiers, multiply their input by −1:



(5)

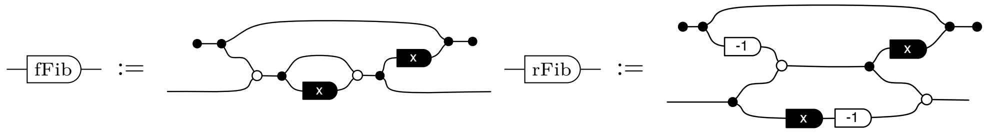
The SFG above thus solves the toy *sustainable rabbit farming problem*: how many rabbits must the farmer buy and sell in each month to maintain, say, four pairs in her rabbit pen? The answer is obtained by using 4; 4; 4; 4; 4; . . . as input to (5), resulting in 4; −4; 0; −4; 0; . . . : i.e. four pairs bought in the first month and, subsequently, four pairs sold every 2nd month.

The proof that →rFib→ is the inverse of →fFib→ consists of an algebraic manipulation of string diagrams: we shall demonstrate this below, after a brief discussion of the mathematics behind the approach. As explained in the Introduction, the theory $\mathbb{IH}_{\mathsf{R}}$ of Interacting Hopf Algebras characterises linear relations, and an example of such a relation is the behaviour of any signal flow graph when $\mathsf{R} = \mathsf{k}[x]$. This algebraic theory is not a classical (finite product) algebraic theory but a symmetric monoidal theory. This means replacing traditional tree-like syntax with string diagrams. Concretely, $\mathbb{IH}_{\mathsf{k}[x]}$ involves two commutative monoids and two commutative comonoids, meaning that string diagrams are built up from the following:



where k ranges over the coefficients in k. The different colouring given to the indeterminate x is inspired by its special semantic role in SFGs.

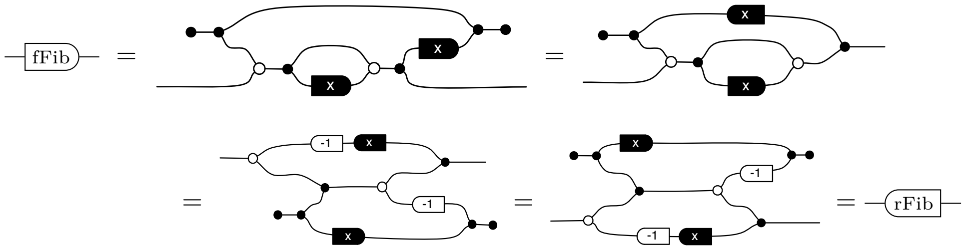
The Fibonacci SFG (3), and its inverse (5)—as string diagrams—look as follows:



(6)

Note the similarity with SFGs (3) and (5) – the string diagram, roughly speaking, is obtained by forgetting the direction of flow—i.e. erasing the arrowheads—and by replacing U-turn wires with so-called ‘cups’ and ‘caps’, formed by composing multiplication with counit, and unit with comultiplication. We use notation $-\boxed{\text{fFib}}-$, $-\boxed{\text{rFib}}-$ to emphasise that (undirected) string diagrams have been obtained by translating SFGs that have a left-to-right signal flow.

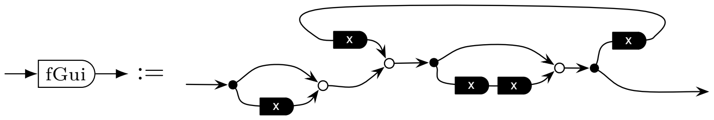
Using the equational theory of [4, 3] we show—by diagrammatic reasoning, i.e. algebraic manipulation of string diagrams—that the inverse of the Fibonacci SFG is as claimed:



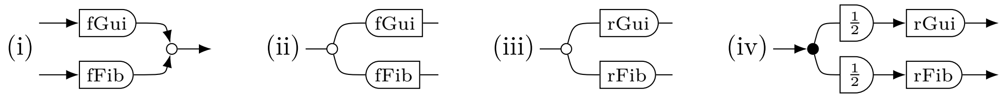
(7)

Note that $-\boxed{\text{fFib}}-$ is *equal* to the mirror image of its inverse $-\boxed{\text{rFib}}-$: this, operationally speaking, means simply that the behaviours of $\to\boxed{\text{fFib}}\to$ and $\to\boxed{\text{rFib}}\to$ are opposites as relations. This worked example demonstrates the power of equational reasoning in $\mathbb{IH}_{\mathsf{k}[x]}$.

Let us now consider a more interesting variant of the sustainable farming problem. Suppose that the rabbit farmer also keeps guinea pigs, which, for the sake of this paper, gestate twice as long as rabbits. The SFG for guinea pigs is the following:



For instance, on input 1; 0; 0; 0; 0; 0; 0; . . . the output is 1; 1; 2; 2; 3; 3; 5; . . . . The combined rabbit and guinea pig pen has SFG (8(i)), with two inputs and one output. The inputs mean buying or selling a species; the output is the total number of animals in the pen.



(8)

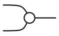
We now pose the *sustainable farming problem*: how many rabbits and guinea pigs must the farmer buy and sell in order to keep the total population fixed? To solve the problem, we draw the string diagram that serves as the *specification* of the inverse of (8(i)), as in (8(ii)). Now $-\boxed{\text{fFib}}-$ and $-\boxed{\text{fGui}}-$ are invertible so we can replace (8(ii)) with the equal diagram (8(iii)).

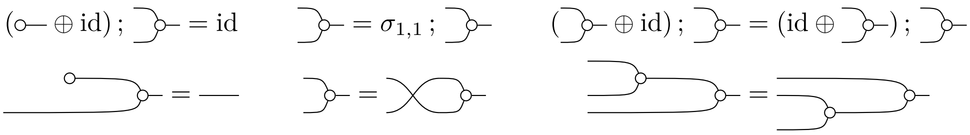
This is as far as we can go in our quest for a SFG, since unfortunately, the specification is not functional: there is no unique solution to the sustainable farming problem. One simple solution is to divide the pen in half and limit the species separately. This, as a SFG, is (8(iv)). As string diagrams (8(iii)) and (8(iv)) are *not* equal. Yet every trace of the second is a trace of the first; this is an example of *refinement*. In this paper, we extend $\mathbb{IH}$ and characterise this refinement relation. Indeed, we will see that, as string diagrams, (8(iv)) $\leq$ (8(iii)).

3 Symmetric Monoidal Theories and props

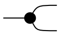
A *symmetric monoidal theory* (SMT) (Σ, E) consists of a set Σ of *generators* $o : m \to n$, each with an *arity* m and *coarity* n ($m, n \in \mathbb{N}$), along with a set E of equations, which are pairs $(t_1, t_2 : m \to n)$ of Σ-terms; t_1 and t_2 must have the same arity and coarity. A Σ-*term* is constructed inductively from generators in Σ, together with the identity $\mathrm{id} : 1 \to 1$ and the symmetry $\sigma_{1,1} : 2 \to 2$, using composition ; and monoidal product $\oplus$. Given Σ-terms $t : k \to l$, $u : l \to m$ and $v : m \to n$, we construct Σ-terms $t \,;\, u : k \to m$ and $t \oplus v : k + m \to l + n$.

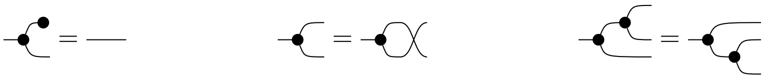
Σ-terms are rendered as diagrams and are considered up to the laws of symmetric strict monoidal categories. Analogously to SFGs, generators are drawn as "circuit components" with dangling wires. The identity is drawn — and the symmetry . Composition of terms is placing them side-by-side and joining the wires. The monoidal product $\oplus$ is stacking terms on top of each other, as in the following examples. Next we introduce some important SMTs, which are used as building blocks to construct the full SMT for reasoning about SFGs.

▶ **Example 1** (The SMT (Σ_M, E_M) of commutative monoids). Σ_M contains two generators: *multiplication*  $: 2 \to 1$ and *unit*  $: 0 \to 1$. E_M contains three equations; we show them both with composition and product explicitly and as diagrams.



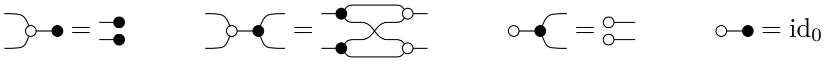
(9)

▶ **Example 2** (The SMT (Σ_C, E_C) of commutative comonoids). Again there are two generators, but this time mirrored: there is a *comultiplication*  $: 1 \to 2$ and a *counit*  $: 1 \to 0$. The equations are the following, this time given only in the diagrammatic form:

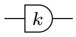

(10)

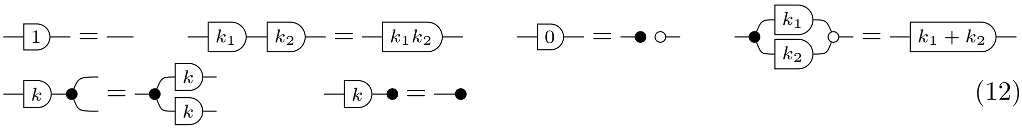
Hopf and Frobenius (bi)monoids are two important ways that monoids and comonoids interact; both are needed for the theories we define in this paper.

▶ **Example 3** (R Hopf monoids). The generators of the SMT are simply those in $\Sigma_M \cup \Sigma_C$, and to the equations in E_M and E_C we add the following *bimonoid laws*:



(11)

For a commutative ring R, we need to add generators  for every $k \in \mathsf{R}$ and stipulate



(12)

▶ **Example 4** ((extra special) Frobenius monoids). The generators are $\Sigma_M \cup \Sigma_C$. Keeping (1) in mind, we colour all the generators in grey, which will later be instantiated as either black or white. Our equations are now the Frobenius law, together with the special and extra equations. We often call the latter the "bone" equation, due to its appearance when drawn.

(13)

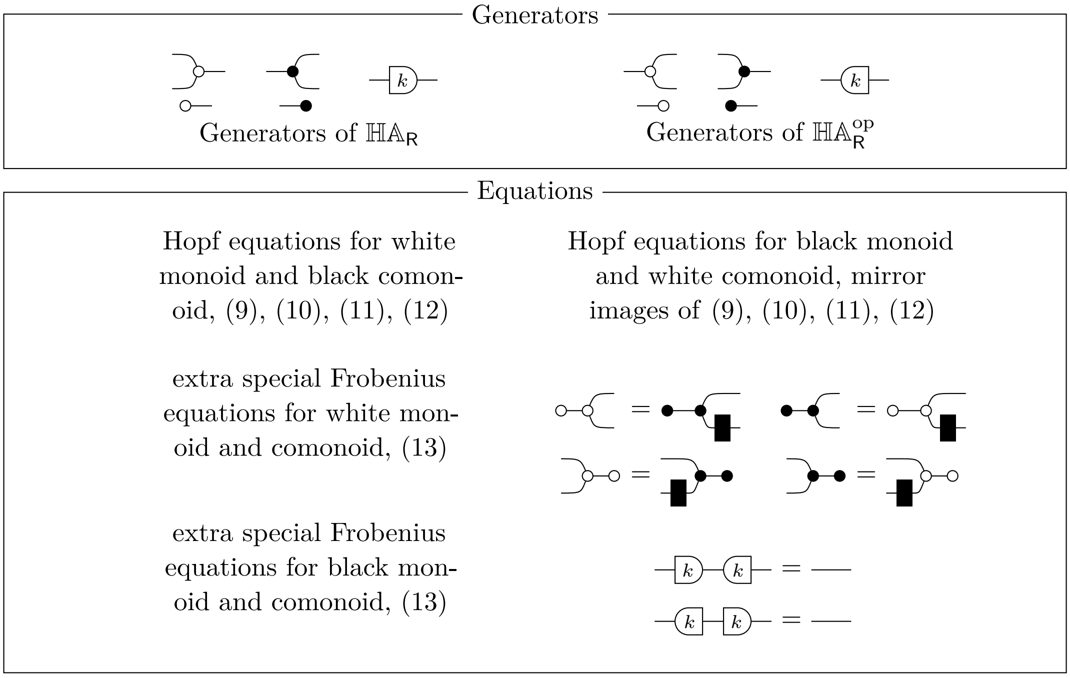


Figure 1 The presentation of $\mathbb{IH}_{\mathsf{R}}$. k takes all values in $\mathsf{R} \setminus \{0\}$. ─▮─ is the *antipode*, which is defined to be either of $\boxed{-1}$ or $\boxed{-1}$; they can be shown to be equal. See [7] for more details.

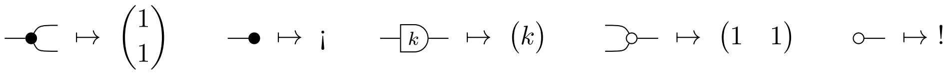
We can obtain a symmetric monoidal category from an SMT (Σ, E) as follows:

- objects are natural numbers
- arrows $m \to n$ are Σ-terms $m \to n$ modulo the laws of symmetric monoidal categories and the (smallest congruence containing) the equations $t_1 = t_2$ for each pair $(t_1, t_2) \in E$

Such a category is a special type of symmetric monoidal category called a prop.

▶ **Definition 5.** A *prop* (product and permutation category) is a strict symmetric monoidal category with objects $\mathbb{N}$, where $m \oplus n := m + n$. A homomorphism is an identity-on-objects symmetric monoidal functor, giving a category $\mathbf{PROP}$.

▶ **Example 6.** Given a commutative ring R, the prop $\mathbf{Mat}_{\mathsf{R}}$ of matrices over R has as arrows $m \to n$ the $n \times m$ matrices, composition ; is matrix multiplication and $A \oplus B$ is the matrix $\begin{pmatrix} A & 0 \\ 0 & B \end{pmatrix}$. In [16, 7] it is shown that $\mathbf{Mat}_{\mathsf{R}}$ is isomorphic to the prop $\mathbb{HA}_{\mathsf{R}}$ arising from the SMT of Hopf monoids over R (Example 3). The isomorphism $\mathcal{S}' : \mathbb{HA}_{\mathsf{R}} \to \mathbf{Mat}_{\mathsf{R}}$ maps



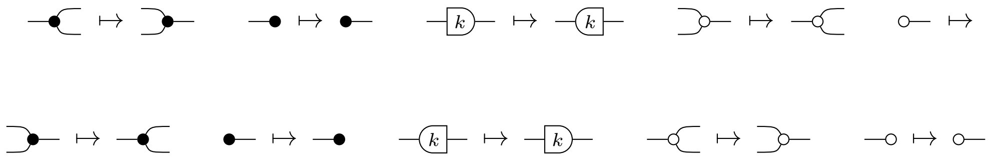
(14)

where $! : 0 \to 1$ and ¡ $: 1 \to 0$ are given by the universal property of 0 in $\mathbf{Mat}_{\mathsf{R}}$.

Observe (14) defines $\mathcal{S}'$ for all arrows A of $\mathbb{HA}_{\mathsf{R}}$. More generally, to specify a homomorphism from a prop obtained from an SMT (Σ, E), it is enough to define it on the generators in Σ, and check that the equations E hold in the image. We shall often use this argument.

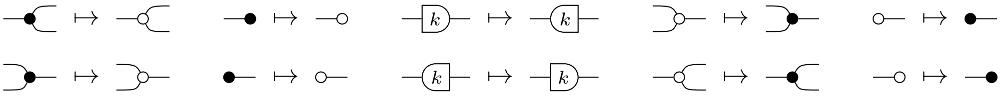
4 Interacting Hopf Algebras

Zanasi with the first and third authors introduced the SMT of Interacting Hopf Algebras [5, 7] as a foundation for SFGs [4, 6, 8]. We recall the equational theory in Fig. 1 where R is a fixed principal ideal domain, and denote the resulting prop by $\mathbb{IH}_{\mathsf{R}}$. As illustrated in (1), the theory features monoids and comonoids that interact either as extra special Frobenius monoids (Example 4) or as Hopf monoids (Example 3). One remarkable feature of this equational theory is that it contains *two* symmetries: (i) that the mirror image (†) of any equation is an equation, and (ii) is that the photographic negative (°, white ↔ black) of any equation is an equation. Formally, define prop morphisms $(-)^{\dagger}\colon \mathbb{IH}_{\mathsf{R}}^{\mathrm{op}} \to \mathbb{IH}_{\mathsf{R}}$ mapping

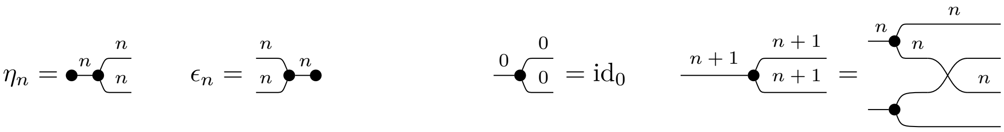


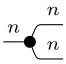
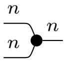
(15)

and $(-)^{\circ}\colon \mathbb{IH}_{\mathsf{R}} \to \mathbb{IH}_{\mathsf{R}}$ mapping

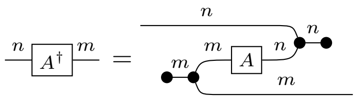


The morphism $(-)^{\dagger}$ is related to the self-dual compact closed structure [15] of $\mathbb{IH}_{\mathsf{R}}$ defined for each $n \in \mathbb{N}$ by assigning $\eta_n : 0 \to n+n$ (cap) and $\epsilon_n : n+n \to 0$ (cup) the string diagrams:

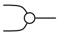
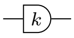


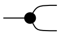
▶ **Remark 7.** *Above we used*  *for the n-fold monoidal product of* *,*  *for* id_n *and*  *is inductively defined above, and similarly for*  *and* *. The same convention will be used for the white structure. We shall omit the labels when there is no risk of ambiguity.*

The contravariant morphism induced by the compact closed structure coincides with $(-)^{\dagger}$, as defined in (15), that is for all $A : m \to n$:



Consider the prop $\mathbf{LinRel}_{\mathsf{k}}$ of linear relations over k, the field of fractions of R. An arrow $m \to n$ is a linear subspace of $\mathsf{k}^m \times \mathsf{k}^n$, ; is relational composition and $\oplus$ is direct sum. Also $\mathbf{LinRel}_{\mathsf{k}}$ has a self-dual compact closed structure: the induced contravariant morphism $(-)^{\dagger}\colon \mathbf{LinRel}_{\mathsf{k}}{}^{\mathrm{op}} \to \mathbf{LinRel}_{\mathsf{k}}$ maps a linear relation $R \subseteq \mathsf{k}^m \times \mathsf{k}^n$ to its opposite $R^{\dagger} \subseteq \mathsf{k}^n \times \mathsf{k}^m$. Let $*$ be the unique element of the 0 dimensional vector space k^0. Then

 $\mapsto \{(\begin{pmatrix} x \\ y \end{pmatrix}, x+y) \mid x, y \in \mathsf{k}\}$ ○— $\mapsto \{(*, 0)\}$  $\mapsto \{(x, kx) \mid x \in \mathsf{k}\}$

 $\mapsto \{(x, \begin{pmatrix} x \\ x \end{pmatrix}) \mid x \in \mathsf{k}\}$ —● $\mapsto \{(x, *) \mid x \in \mathsf{k}\}$

defines a unique prop morphism $\mathcal{S} : \mathbb{IH}_{\mathsf{R}} \to \mathbf{LinRel}_{\mathsf{k}}$ that preserves $(-)^{\dagger}$. For instance,

$\mathcal{S}($●—$) = \mathcal{S}($—●$^{\dagger}) = (\mathcal{S}($—●$))^{\dagger} = \{\, (*,\, x) \mid x \in \mathsf{k} \,\}.$

▶ **Theorem 8** ([7]). *$\mathcal{S} : \mathbb{IH}_{\mathsf{R}} \to \mathbf{LinRel}_{\mathsf{k}}$ is an isomorphism.*

Let us explain the relationship of $\mathcal{S}$ with $\mathcal{S}'$ from Example 6. Observe that any $A : m \to n$ in $\mathbb{IH}_{\mathsf{R}}$ built out of the leftmost five generators of Fig. 1 (drawn $\overset{m}{-}\boxed{A}\overset{n}{-}$) is also in $\mathbb{HA}_{\mathsf{R}}$ and, similarly, any term built of the five rightmost generators ($\overset{m}{-}\boxed{A}\overset{n}{-}$) is in $\mathbb{HA}_{\mathsf{R}}^{\mathrm{op}}$. Indeed, we have prop embeddings $\mathbb{HA}_{\mathsf{R}} \to \mathbb{IH}_{\mathsf{R}} \leftarrow \mathbb{HA}_{\mathsf{R}}^{\mathrm{op}}$. Similarly, there are embeddings $\mathbf{Mat}_{\mathsf{R}} \to \mathbf{LinRel}_{\mathsf{k}} \leftarrow \mathbf{Mat}_{\mathsf{R}}{}^{\mathrm{op}}$ mapping a matrix to its graph, and the following commutes [7]:

$$\begin{tikzcd}
\mathbb{HA}_{\mathsf{R}} \arrow[r] \arrow[d, "\mathcal{S}'"'] & \mathbb{IH}_{\mathsf{R}} \arrow[d, "\mathcal{S}"] & \mathbb{HA}_{\mathsf{R}}^{\mathrm{op}} \arrow[l] \arrow[d, "\mathcal{S}'^{\mathrm{op}}"] \\
\mathbf{Mat}_{\mathsf{R}} \arrow[r] & \mathbf{LinRel}_{\mathsf{k}} & \mathbf{Mat}_{\mathsf{R}}{}^{\mathrm{op}} \arrow[l]
\end{tikzcd}$$

The following result informs us that every arrow of $\mathbb{IH}_{\mathsf{R}}$ can be written in *span form.*

▶ **Lemma 9** ([7]). *For all $\overset{m}{-}\boxed{A}\overset{n}{-}$ in $\mathbb{IH}_{\mathsf{R}}$ there exist $k \in \mathbb{N}$, $\overset{m}{-}\boxed{A_1}\overset{k}{-}$ and $\overset{k}{-}\boxed{A_2}\overset{n}{-}$ such that $\overset{m}{-}\boxed{A}\overset{n}{-} = \overset{m}{-}\boxed{A_1}\overset{k}{-}\boxed{A_2}\overset{n}{-}$.*

Moreover, the following property of $\mathbb{HA}_{\mathsf{R}}$ also holds in $\mathbb{IH}_{\mathsf{R}}$.

▶ **Lemma 10** ([7]). *For all $\overset{m}{-}\boxed{A}\overset{n}{-}$, $\overset{m}{-}\boxed{A}\overset{n}{-}\bullet = \overset{m}{-}\bullet$ and $\circ\overset{m}{-}\boxed{A}\overset{n}{-} = \circ\overset{n}{-}$.*

5 Symmetric Monoidal Inequality Theories and Ordered Props

We reviewed the construction of props from SMTs in the previous section. In order to capture *inequalities* of terms, however, we need a new notion. We thus introduce the concept of a *Symmetric Monoidal Inequality Theory* (SMIT), which allows the specification of a partial order on terms built out of generators, analogously to how SMTs specify equivalence relations.

▶ **Definition 11** (Symmetric Monoidal Inequality Theory). A SMIT is a pair (Σ, I). As for SMTs, Σ is a collection of generators $o : m \to n$, and I is a set of pairs (t_1, t_2) of Σ-terms with the same (co)arity, but we now think of them as representing *inequalities*. That is, where before the interpretation of a pair (t_1, t_2) was that $t_1 = t_2$, we now stipulate that $t_1 \leq t_2$.

Set I leads to a preorder on terms by reflexive and transitive closure. A partial order arises through anti-symmetry: t_1 and t_2 are equated when $t_1 \leq t_2$ and $t_2 \leq t_1$. We will use $\leq_I$, or $\leq$ when I is clear from context, and write the corresponding equivalence as equality. The equivalence classes are the arrows of a 2-category $\mathbb{T}_{(\Sigma,I)}$ that we call an *ordered prop*.

▶ **Definition 12** (Ordered Prop). A *2-prop* is a strict symmetric monoidal 2-category whose objects are natural numbers and monoidal product on objects is addition. An *ordered prop* is a 2-prop which is locally posetal, that is, where every hom-category is a poset – i.e. there is at most one 2-cell ($\leq$) between any two arrows. Together with ordered prop morphisms (identity-on-objects strict monoidal 2-functors) we have a category **OrdPROP**.

Since ordered props are a kind of 2-category, in any ordered prop, for all f, f', g, g', we have:

$$\text{if } f \leq f' \text{ and } g \leq g' \text{then } f \,;\, g \leq f' \,;\, g' \tag{16}$$

$$\text{if } f \leq f' \text{ and } g \leq g' \text{then } f \oplus g \leq f' \oplus g'. \tag{17}$$

▶ **Example 13.** The prop $\mathbf{LinRel}_{\mathsf{k}}$ of linear relations has a partial order on arrows given by inclusion as subspaces. It is straightforward to check (16) and (17). The prop $\mathbf{Mat}_{\mathsf{R}}$ of matrices (Example 6) can be also regarded as an ordered prop with discrete order.

Returning to SMITs, the arrows of $\mathbb{T}_{(\Sigma,I)}$ are the equivalence classes of Σ-terms; since arities and coarities are respected, we may use the partial order $\leq_I$ to define the 2-cells in the hom-category $\mathbb{T}_{(\Sigma,I)}(m,n)$. It follows that $\mathbb{T}_{(\Sigma,I)}$ is an ordered prop. Note that SMITs are a generalisation of SMTs. First, any prop can be made into a (discrete) ordered prop by adding identity 2-cells. This gives an embedding (a faithful homomorphism of ordered pros: a strict identity-on-objects symmetric monoidal 2-functor) $\mathbf{PROP} \hookrightarrow \mathbf{OrdPROP}$.

▶ **Remark 14.** *Any SMT* (Σ, E) *can considered as a SMIT by taking the symmetric closure of* E*: i.e.* $I = E \cup E^{op}$*. Then the image of the prop generated by the SMT* (Σ, E) *under the embedding is isomorphic to the ordered prop given by SMIT* (Σ, I)*.*

6 Presenting the 2-dimensional structure of $\mathbf{LinRel}_{\mathsf{k}}$

In this section we prove our main theorem. We extend the SMT of Interacting Hopf Algebras to a SMIT and the isomorphism of Theorem 8 to a 2-isomorphism of ordered props. In other words, we characterise the subset order of linear relations (Example 13).

The symmetry discussed in Section 4 is broken in the ordered setting. Indeed, to get from the SMT to a SMIT we follow the procedure of Remark 14 and add *just one inequality* (2): —○ ≤ —●. Interpreted as linear relations (via $\mathcal{S} : \mathbb{IH}_{\mathsf{R}} \to \mathbf{LinRel}_{\mathsf{k}}$), (2) says that the unique 0-dimensional subspace $\{0\}$ of k considered as a k-vector space is included in the unique 1-dimensional subspace, i.e. k itself. This is, of course, a strict inclusion.

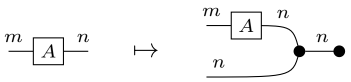
▶ **Theorem 15.** $\mathbb{IH}_{\mathsf{R}} \cong \mathbf{LinRel}_{\mathsf{k}}$ *as ordered props.*

For the proof we need to recall some elementary linear algebra. Regarding an $m \times n$ matrix A as a list of its column vectors $\mathbf{a}_1$, $\mathbf{a}_2$, …, $\mathbf{a}_n$, the *span* of A ($\mathrm{Sp}(A)$) is the linear subspace of k^m with elements linear combinations $\lambda_1\mathbf{a}_1 + \lambda_2\mathbf{a}_2 + \cdots + \lambda_n\mathbf{a}_n$, $\lambda_i \in \mathsf{k}$. The following is a well-known fact of linear algebra (see, e.g. [2, Proposition 2.13]).

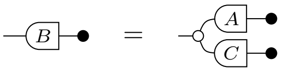
▶ **Lemma 16.** *Suppose that for some* $m \times n$ *matrix* A *we have* $\mathrm{Sp}(A) \subseteq V$*. Then there exists* $m \times n'$ *matrix* C *such that* $V = \mathrm{Sp}(A, C)$*.*

Proof of Theorem 15. Inequation (2) is clearly sound, we thus only have to show completeness; that is (2) suffices to account for any inclusion between *arbitrary* linear relations.

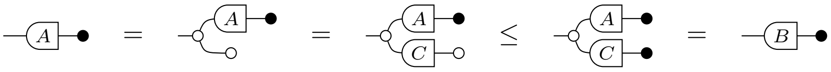
Let therefore $A, B : m \to n$ be arrows of $\mathbb{IH}_{\mathsf{R}}$ such that $\mathcal{S}(A) \subseteq \mathcal{S}(B) \subseteq \mathsf{k}^m \times \mathsf{k}^n$. Now $\mathsf{k}^m \times \mathsf{k}^n \cong \mathsf{k}^{m+n} \times \mathsf{k}^0 \cong \mathsf{k}^{m+n}$; this, diagrammatically, means the following manipulation:



Using this "rewiring" argument we may assume w.l.o.g. that $A, B : m \to 0$. Further, using Lemma 9, we may assume that A, B consist only of the rightmost five generators in Figure 1: indeed,  by Lemmas 9 and 10. It is thus harmless to consider A and B as matrices, and our initial assumption means that $\mathrm{Sp}(A) \subseteq \mathrm{Sp}(B)$. By the conclusion of Lemma 16, there exists C such that $\mathrm{Sp}(B) = \mathrm{Sp}(A, C)$. Diagrammatically (via $\mathcal{S}$), this gives the following, where for readability we omit decorating the wires:



But we have

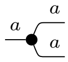
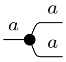
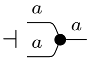
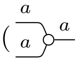
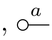
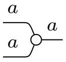
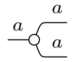


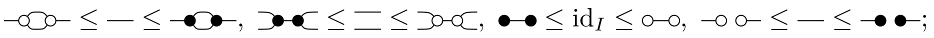
showing that $A \leq B$ is derivable from (2). ◀

While we have shown that (2) suffices to characterise inclusions between subspaces, it is convenient to identify some structural properties that our inequational theory satisfies. By doing so, we are building up a toolbox—useful for reasoning in applications—of principles for reasoning about the structure of the order between linear relations.

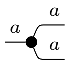
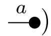
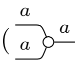
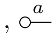
Below we use the notion of *adjunction* in an ordered prop: arrow $f : m \to n$ has a *right adjoint* if there exists $g : n \to m$ such that $\mathrm{id}_m \leq f\,;g$ and $g\,;f \leq \mathrm{id}_n$, in which case we write $f \dashv g$. Right adjoints, if they exist, are unique: if also $f \dashv g'$ then $g = g'$.

▶ **Definition 17.** An abelian bicategory [11] $\mathbf{A}$ is a (loc. posetal) monoidal bicategory where:

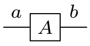
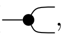
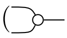
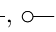
- **(i)** every object a is a commutative comonoid ( , ) with right adjoints  $\dashv$  ,  $\dashv$  , and a commutative monoid ( , ) with right adjoints  $\dashv$  ,  $\dashv$  . This translates to the following (labelling on the wires omitted for clarity):

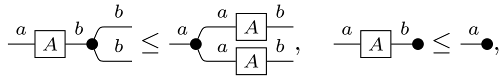


(18)

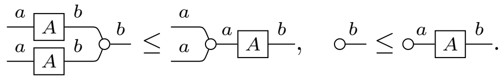
- **(ii)** ( , ) and ( , ) with their right adjoints satisfy the Frobenius equations:

(19)

- **(iii)** every arrow  is a lax ( , )-comonoid homomorphism and a lax ( , )-monoid homomorphism:

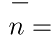


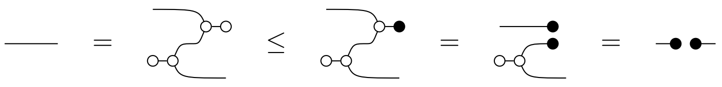
(20)



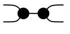
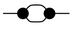
(21)

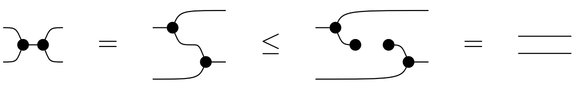
A more concise definition is: $\mathbf{A}$ and $\mathbf{A}^{\mathrm{op}}$ are both bicategories of relations in the sense of [11].

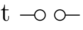
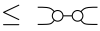
Below, we show that, as an ordered prop, $\mathbb{IH}_{\mathsf{R}}$ is an abelian bicategory. For each object $n \in \mathbb{N}$, comonoids and monoid structures are defined inductively as in Remark 7. A straightforward induction generalises the Frobenius equations for all n in (19), given that they are present for $n = 1$ in Fig. 1. Next we tackle the case of the black units (the rightmost two black inequations of (18)). The unit of the adjunction is a 2-cell witnessing $\mathrm{id}_0 \leq$  and these terms are equated in Fig. 1. It remains to show existence of the counit. For $n = 1$:



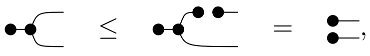
The above argument easily generalises to all n.

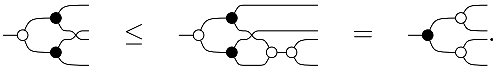
Showing adjointness for the black comultiplication (the leftmost two black inequations of (18)) amounts to demonstrating that  $\leq$  and — $\leq$  . The second is the black special equation in Fig. 1. The first follows from the adjointness of the unit and counit:



For the white case in (18), the inequations are opposite. The same proofs with colours and the sense of the inequality exchanged give the results that  $\leq$ — and  $\leq$  .

Now, to show that all arrows are lax comonoid homomorphisms, it is enough to check that each of the generators obeys the conditions of (20). Several of these are in fact equalities. The derivations for the two interesting cases are given below:





Again, the white case (21) is symmetric.

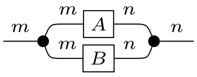
7 The 2-dimensional algebra of Linear Relations

The structure identified in the previous sections enables us to highlight some interesting properties of the ordering $\leq$. We start with a few elementary, but useful observations.

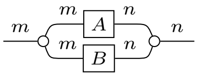
▶ **Lemma 18.** *For all $A, B \in \mathbb{IH}_{\mathsf{R}}(m,n)$, the following hold:*
(a) *If $A \leq B$, then $A^{\dagger} \leq B^{\dagger}$;*
(b) *$(A^{\dagger})^{\dagger} = A = (A^{\circ})^{\circ}$;*
(c) *If $A \leq B$, then $A^{\circ} \geq B^{\circ}$;*
(d) *$(A^{\dagger})^{\circ} = (A^{\circ})^{\dagger}$.*

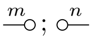
Proof. For all equations $A = B$ in $\mathbb{IH}_{\mathsf{R}}$ (Fig. 1), one has $A^{\dagger} = B^{\dagger}$ and $A^{\circ} = B^{\circ}$. For the only inequation of $\mathbb{IH}_{\mathsf{R}}$, —○ $\leq$ —●, we clearly have —○$^{\dagger} \leq$ —●† and —○$^{\circ} \geq$ —●$^{\circ}$: this implies (a) and (c). The proofs of (b) and (d) are inductions on the definitions of $(-)^{\dagger}$ and $(-)^{\circ}$. ◀

We proceed by showing that every homset $\mathbb{IH}_{\mathsf{R}}(m,n)$ carries a lattice structure. Given two arrows $A, B \colon m \to n$ we define $A \wedge B$, $\top$, $A \vee B$ and $\bot$ as follows.

$A \wedge B =$ 

$\top =$ 

$A \vee B =$ 

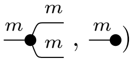
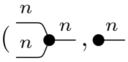
$\bot =$ 

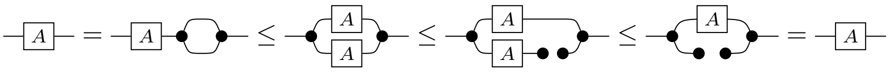
▶ **Lemma 19.** *For all $A, B \in \mathbb{IH}_{\mathsf{R}}(m,n)$, the following hold:*
(a) *$(A \vee B)^{\dagger} = A^{\dagger} \vee B^{\dagger}$ and $\bot^{\dagger} = \bot$;*
(b) *$(A \wedge B)^{\dagger} = A^{\dagger} \wedge B^{\dagger}$ and $\top^{\dagger} = \top$;*
(c) *$(A \vee B)^{\circ} = A^{\circ} \wedge B^{\circ}$ and $\bot^{\circ} = \top$;*
(d) *$(A \wedge B)^{\circ} = A^{\circ} \vee B^{\circ}$ and $\top^{\circ} = \bot$.*

Proof. Trivial by unfolding the definitions. ◀

▶ **Theorem 20.** *The operations $(\vee, \bot, \wedge, \top)$ define a lattice structure on every homset $\mathbb{IH}_{\mathsf{R}}(m,n)$ wrt the ordering $\leq$.*

Proof. First observe that for all $A \colon m \to n$, we have, by the rightmost inequations in (18) and (20), that $A \leq A \,;\, \top \leq \top$. By Lemmas 18(b) and 19(d), one deduces $A \geq \bot$.

Now, it is enough to check that both $(\mathbb{IH}_{\mathsf{R}}(m,n), \wedge, \top)$ and $(\mathbb{IH}_{\mathsf{R}}(m,n), \vee, \bot)$ are commutative and idempotent monoids. Observe that if this holds for $(\mathbb{IH}_{\mathsf{R}}(m,n), \wedge, \top)$ then, by Lemmas 18(b) and 19(d), it holds also for $(\mathbb{IH}_{\mathsf{R}}(m,n), \vee, \bot)$. The fact that $(\mathbb{IH}_{\mathsf{R}}(m,n), \wedge, \top)$ is a commutative monoid follows immediately from the fact that () and () are commutative (co)monoids. For idempotency we observe the following chain of inequalities.



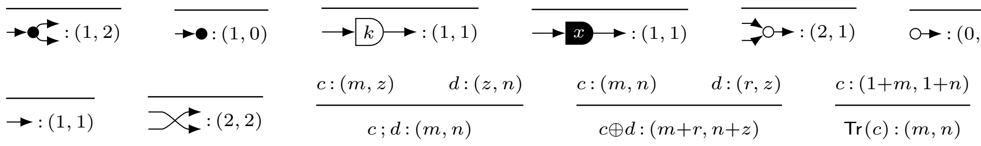


Figure 2 Sort inference rules.

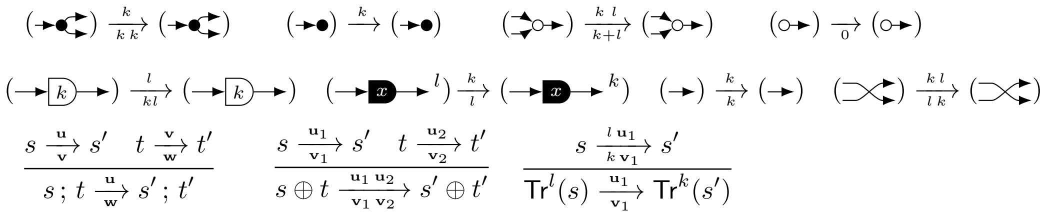
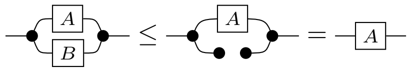
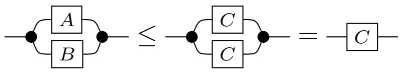


Figure 3 Structural rules for operational semantics, with k, l ranging over k and $\mathbf{u}$, $\mathbf{v}$, $\mathbf{w}$ vectors of elements of k of the appropriate size.

Now to see that $\wedge$ defines a meet, note that $A \wedge B \leq A$:  and, by a symmetric argument, $A \wedge B \leq B$. Assuming that $A \leq C$ and $B \leq C$ we have



The argument showing that $\vee$ defines a join is again symmetric. ◀

▶ **Lemma 21.** *For all $A, B, C \in \mathbb{IH}_{\mathsf{R}}(m, n)$, the following hold:*
(a) $C\,;(A \wedge B) \leq (C\,;A) \wedge (C\,;B)$ *and* $C\,;\top \leq \top$*;*
(b) $(A \wedge B)\,;C \leq (A\,;C) \wedge (B\,;C)$ *and* $\top\,;C \leq \top$*;*
(c) $(A \vee B)\,;C \geq (A\,;C) \vee (B\,;C)$ *and* $\bot\,;C \leq \bot$*;*
(d) $C\,;(A \vee B) \geq (C\,;A) \vee (C\,;B)$ *and* $C\,;\bot \leq \bot$.

Proof. Part (a) follows from (20). With (a), Lemmas 18(a) and 19(b) imply (b). With (b), Lemmas 18(b) and 19(d) imply (c). With (c), Lemmas 18(a) and 19(a) imply (d). ◀

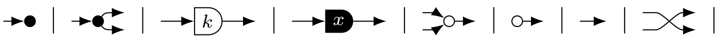
To summarise, we have a well-behaved set of operations

$$\top,\ \ \bot,\ \ \wedge,\ \ \vee,\ \ (-)^{\dagger},\ \ (-)^{\circ},\ \ ;\ ,\ \ \mathrm{id}$$

which, because of its similarity to the algebra of relations [14], we call *the algebra of linear relations*. Observe that the operations $\top$, $\wedge$, $(-)^{\dagger}$, $;$ and *id* have exactly the same meaning: full relation, intersection, inverse, composition and identity relation. Instead $\bot$, $\vee$, $(-)^{\circ}$ which in the algebra of relations denote, respectively, empty relation, union and complement, do not coincide. The reason is that, in general, these operations cannot be defined on linear relations: e.g., the union of two linear relations may not be linear.

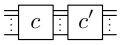
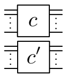
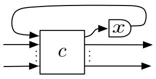
8 Back to Signal Flow Graphs

We recall from [6] the *Directed Signal Flow Calculus*. The syntax is given by the grammar below where k ranges over a fixed field k.

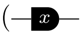
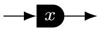
c ::=  $\mid c\oplus c \mid c\,;c \mid \mathsf{Tr}(c)$

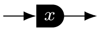
A *sort* is a pair $(m,\, n)$, with $m, n \in \mathbb{N}$. We shall consider only terms that are sortable, according to the rules of Fig. 2. An inductive argument confirms uniqueness of sorts: if $c : (m,\, n)$ and $c : (m',\, n')$ then $m = m'$ and $n = n'$. We will refer to sortable terms as *circuits* since, intuitively, a term $c : (m,\, n)$ represents a circuit with m inputs on the left and n outputs on the right.

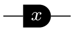
In the syntax specification we used a graphical rendering of the components: we will seldom write terms in the traditional way and instead represent them as diagrams:

$c\,;c'$ is drawn  $c\oplus c'$ is drawn  $\mathsf{Tr}(c)$ is drawn  .

The graphical notation identifies some syntactically different terms, e.g. $(c_1 \oplus c_2)\,;(d_1 \oplus d_2)$ and $(c_1\,;d_1) \oplus (c_2\,;d_2)$. This is harmless (see Remark 1 in [6]) since such circuits are observationally equivalent wrt the operational semantics that we introduce next.

The operational semantics interprets terms as stream transducers. The wires carry elements of a field k that enter and exit through input and output ports. Formally, it is a transition system that has *augmented* circuits as states: each *delay* component () and each *guarded feedback* (Tr) are assigned some value $k \in \mathsf{k}$. States are obtained by replacing delays and feedbacks in the syntax specification with  ^{k} and Tr^k for each $k \in \mathsf{k}$. We only consider sortable states; the discipline is obtained by adding the following to Fig. 2.

 $^k \;:\; (1,\, 1)$ and $\dfrac{c : (1{+}m,\, 1{+}n)}{\mathsf{Tr}^k(c) : (m,\, n)}$

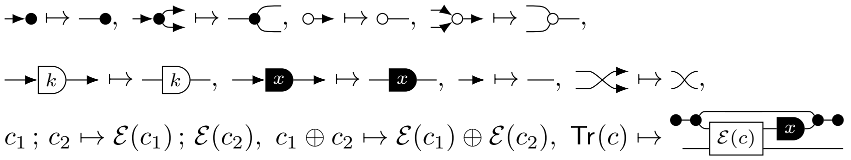
The structural rules for operational semantics are given in Fig. 3 where we use strings of length n to represents vectors in k^n. If state $s : (m,\, n)$ is the source of a transition $\xrightarrow[\mathbf{w}]{\mathbf{v}} t$ then t is also a state with sort $(m,\, n)$ and $\mathbf{v}$ and $\mathbf{w}$ are strings (vectors) in k^m and k^n, respectively. Intuitively, $s \xrightarrow[\mathbf{w}]{\mathbf{v}} t$ means that s can become t in one step whenever it inputs $\mathbf{v}$ on the m ports on the left and outputs $\mathbf{w}$ on the n ports on the right. Each circuit c then yields a transition system with a chosen *initial state* s_0 of c, obtained by replacing delays and feedbacks in c with  ^{0} and Tr^0: this means that we only consider executions where the registers are initialised with 0. A different semantics is considered in [13] where registers can be initialised with arbitrary values.

A *computation* of a circuit c, is an infinite path $s_0 \xrightarrow[\mathbf{v}_0]{\mathbf{u}_0} s_1 \xrightarrow[\mathbf{v}_1]{\mathbf{u}_1} \ldots$ in the transition system of c, starting from its initial state s_0. When c has sort $(m,\, n)$, each $\mathbf{u}_i$ and $\mathbf{v}_i$ are strings over k, say $k_{i1} \ldots k_{im}$ and $l_{i1} \ldots l_{in}$, respectively. The *trace*, also called *trajectory*, of this computation is then a pair of vectors $\boldsymbol{\alpha} = \begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_n \end{pmatrix}, \boldsymbol{\beta} = \begin{pmatrix} \beta_1 \\ \vdots \\ \beta_m \end{pmatrix}$ where $\alpha_j = k_{0j}k_{1j}\ldots$ and $\beta_j = l_{0j}l_{1j}\ldots$. For example, consider the (1,1) circuit in (3): the first three steps of an infinite computation are illustrated in (4). The trace for this computation is thus $(1;0;0;0;\ldots\,,\,1;2;3;5;\ldots)$.

We write $\mathit{it}(c)$ for the set of traces, and this is our notion of *observable behaviour*. Two circuits c and d are *observationally equivalent*, written $c \sim d$, iff $\mathit{it}(c) = \mathit{it}(d)$.

A few considerations are in order about the role of the *denotational semantics* given in [6]. The signal flow calculus is canonical in the sense that it enjoys a Kleene-like theorem [8, Thorem 7.4]: it denotes all and only the *rational functions* on streams (see [18] and [17]). Moreover the denotational semantics is fully abstract with respect to the observational equivalence (Corollary 2 and Proposition 4 in [6]) and, from this correspondence, a sound and complete axiomatization for $\sim$ follows. We focus on some technical details of this axiomatisation below.

The idea is to translate circuits of sort (m, n) into arrows $m \to n$ of $\mathbb{IH}_{\mathsf{k}[x]}$, where $\mathsf{k}[x]$ is the principal ideal domain of polynomials with indeterminate x and coefficients from k. Intuitively, the inductively defined translation $\mathcal{E}$ "erases directions" from the wires:



$$c_1 \,;\, c_2 \mapsto \mathcal{E}(c_1) \,;\, \mathcal{E}(c_2),\ \ c_1 \oplus c_2 \mapsto \mathcal{E}(c_1) \oplus \mathcal{E}(c_2),\ \ \mathsf{Tr}(c) \mapsto \qquad (22)$$

where $-\boxed{k}-$ and $-\boxed{x}-$, in $\mathbb{IH}_{\mathsf{k}[x]}$, correspond to polynomials k and x in $\mathsf{k}[x]$.

For an example consider the circuits $\rightarrow\boxed{\text{fFib}}\rightarrow$ and $\rightarrow\boxed{\text{rFib}}\rightarrow$ in (3) and (5): the corresponding string diagrams $\mathcal{E}(\rightarrow\boxed{\text{fFib}}\rightarrow)$ and $\mathcal{E}(\rightarrow\boxed{\text{fGui}}\rightarrow)$ are shown in (6).

The following ensures that the theory of Fig. 1 is sound and complete for trace equivalence.

▶ **Theorem 22** ([6])**.** *$c \sim d$ iff $\mathcal{E}(c) = \mathcal{E}(d)$, for all circuits c, d.*

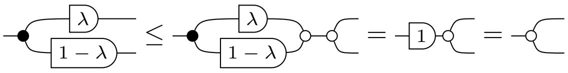
To prove equivalence of signal flow calculus terms it is thus enough to view them as string diagrams in $\mathbb{IH}_{\mathsf{k}[x]}$ by forgetting flow direction, and use the equational theory of Fig. 1.

The reader may wonder why we introduced the directed signal flow calculus rather then using string diagrams directly. The reason is that string diagrams of $\mathbb{IH}_{\mathsf{k}[x]}$ are undirected and flow directionality is essential to execute them (see Remark 2 in [6]). String diagrams, however, *do* provide a useful language to reason about signal flow graphs. For instance, using the algebra of linear relations from Section 7, the *opposite* of an arbitrary circuit c can be specified by the string diagram $\mathcal{E}(c)^\dagger$.

It is therefore natural to think of string diagrams in $\mathbb{IH}_{\mathsf{k}[x]}$ as *specifications* and of circuits in the directed signal flow calculus as *implementations.* More formally, we say that a specification A (an arrow of $\mathbb{IH}_{\mathsf{k}[x]}$) *refines* a specification B whenever $A \leq B$ and we say that a circuit c *implements* a specification A whenever $\mathcal{E}(c)$ refines A, i.e., $\mathcal{E}(c) \leq A$.

▶ **Remark 23.** *One could have defined $c \precsim d$ iff $it(c) \subseteq it(d)$ but this notion would collapse to $\sim$, since the observational behaviour of any circuit, which we have defined as a relation, is actually the graph of a function. To see this, note that the operational semantics of Fig. 3 is deterministic: given any state s and transitions $s \xrightarrow[\mathbf{v}]{\mathbf{u}}$, $s \xrightarrow[\mathbf{v}']{\mathbf{u}}$, it follows that $\mathbf{v} = \mathbf{v}'$. A similar, but non-deterministic, semantics subsuming that of Fig. 3 was given in [6, Fig. 2] for arbitrary string diagrams. In fact, losing direction of signal flow makes the definition simpler, since the feedback becomes expressible in terms of the more basic components and does not thus need a separate structural rule. It is the possibly non-deterministic nature of string-diagrams-as-SFG-specifications that makes the refinement relation interesting.*

We now return to the motivating example of Section 2. The fact that the circuit in (5) solves the sustainable rabbit farming problem is witnessed by the fact that it is an implementation of $\mathcal{E}(\rightarrow\boxed{\text{fFib}}\rightarrow)^\dagger$. Here, since the behaviour of $\rightarrow\boxed{\text{fFib}}\rightarrow$ is invertible, there is an equivalence: see the derivation in (7). Instead, the sustainable farming problem for rabbits and guinea pigs cannot be solved by equational reasoning since the combined SFG (8(ii)) (henceforth $\to\boxed{\text{comb}}\to$) is not invertible. To prove that SFG (8(iv)) (henceforth $\to\boxed{\text{sol}}\to$) is a solution, we should show that it implements $\mathcal{E}(\to\boxed{\text{comb}}\to)^{\dagger}$, namely we should check that $\mathcal{E}(\to\boxed{\text{comb}}\to) \leq \mathcal{E}(\to\boxed{\text{comb}}\to)^{\dagger}$. It follows from the general fact shown below; taking $\lambda = \frac{1}{2}$ gives the claimed solution.



9 Related work

Although we concentrated on the discrete semantics, signal flow graphs also have a continuous incarnation where delays act as integrators; for this reason they are a useful foundational model in signal processing and control theory: as a consequence, for computer scientists [1] they are also important as models of cyber-physical systems that can be analysed and verified in concert with discrete models. For example, in *loc. cit.* the authors study SFGs with the aid of block diagrams that are closely related to the Signal Flow Calculus of Section 8.

The operation (22) of passing from the directed calculus to string diagrams by "erasing arrowheads" is similar in spirit to the ideas of Willems [23], who argued that concepts of input and output are inherently non-compositional, complicate the mathematics, and—perhaps most importantly—do not actually exist in the underlying physical reality. It is this realisation that gives rise to the equational theory of Interacting Hopf Algebras. Moreover, Baez and Erbele [3] prove that the same equational theory is suitable for the continuous behaviour. Remarkably similar symmetric monoidal theories appear in concurrency [9, 10, 21] and quantum computing [12]. None of these works, however, investigates the underlying posetal structure. We believe that the structure of cartesian and abelian bicategories [11] may be successfully exploited in those fields.

References

1 Rob Arthan, Ursula Martin, and Paulo Oliva. A Hoare logic for linear systems. *Form Asp Comp*, 25:345–363, 2013.

2 Sheldon Axler. *Linear Algebra Done Right.* Springer, 2nd edition, 1997.

3 John C. Baez and Jason Erbele. Categories in control. Technical report, arXiv:1405.6881, 2014.

4 Filippo Bonchi, Paweł Sobociński, and Fabio Zanasi. A categorical semantics of signal flow graphs. In *Concurrency Theory - 25th International Conference, (CONCUR 2014)*, volume 8704 of *LNCS*, pages 435–450. Springer, 2014. `doi:10.1007/978-3-662-44584-6_30`.

5 Filippo Bonchi, Paweł Sobociński, and Fabio Zanasi. Interacting bialgebras are Frobenius. In *Foundations of Software Science and Computation Structures - 17th International Conference, (FOSSACS 2014)*, number 8412 in LNCS. Springer, 2014.

6 Filippo Bonchi, Paweł Sobociński, and Fabio Zanasi. Full Abstraction for Signal Flow Graphs. In *Proceedings of the 42nd Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages - POPL '15*, pages 515–526, New York, New York, USA, 2015. ACM Press. `doi:10.1145/2676726.2676993`.

7 Filippo Bonchi, Paweł Sobociński, and Fabio Zanasi. Interacting Hopf algebras. *Journal of Pure and Applied Algebra*, 221(1):144–184, mar 2017. `doi:10.1016/j.jpaa.2016.06.002`.

8 Filippo Bonchi, Paweł Sobociński, and Fabio Zanasi. The Calculus of Signal Flow Diagrams I: Linear relations on streams. *Information and Computation*, 252(v):2–29, feb 2017. `doi:10.1016/j.ic.2016.03.002`.

9 Roberto Bruni, Ivan Lanese, and Ugo Montanari. A basic algebra of stateless connectors. *Theor. Comput. Sci.*, 366:98–120, 2006.

10 Roberto Bruni, Hernán C. Melgratti, Ugo Montanari, and Paweł Sobociński. Connector algebras for C/E and P/T nets' interactions. *Log. Meth. Comput. Sci.*, 9(3), 2013. `doi:10.2168/LMCS-9(3:16)2013`.

11 Aurelio Carboni and Robert Frank Carslaw Walters. Cartesian bicategories I. *Journal of Pure and Applied Algebra*, 49(1–2):11–32, nov 1987. `doi:10.1016/0022-4049(87)90121-6`.

12 Bob Coecke and Ross Duncan. Interacting quantum observables. In *ICALP'08*, pages 298–310, 2008.

13 Brendan Fong. The Algebra of Open and Interconnected Systems. *arXiv.org*, page 230, 2016. URL: `http://arxiv.org/abs/1609.05382`.

14 Bjarni Jónsson and Alfred Tarski. Representation problems for relation algebras. *Bulletin of the American Mathematical Society*, 54(1):80–80, 1948.

15 Gregory M Kelly and Miguel L Laplaza. Coherence for compact closed categories. *Journal of Pure and Applied Algebra*, 19:193–213, 1980.

16 Yves Lafont. Towards an algebraic theory of boolean circuits. *J Pure Appl Alg*, 184:257–310, 2003.

17 Bhagwandas Pannalal Lathi. *Signal processing and linear systems.* Oxford university press New York, 1998.

18 Jan J M M Rutten. A tutorial on coinductive stream calculus and signal flow graphs. *Theoretical Computer Science*, 343(3):443–481, oct 2005. `doi:10.1016/j.tcs.2005.06.019`.

19 Claude E. Shannon. The theory and design of linear differential equation machines. Technical report, National Defence Research Council, 1942.

20 Laurence Sigler. *Fibonacci's Liber Abaci: A Translation into Modern English of Leonardo Pisano's Book of Calculation.* Springer, 2002.

21 Paweł Sobociński. Nets, relations and linking diagrams. In *Algebra and Coalgebra in Computer Science - 5th International Conference, (CALCO 2013)*, volume 8089 of *LNCS*, pages 282–298. Springer, 2013.

22 Paweł Sobociński. Graphical linear algebra. (blog series), 2015. URL: `https://GraphicalLinearAlgebra.net`.

23 Jan C. Willems. The behavioural approach to open and interconnected systems. *IEEE Contr. Syst. Mag.*, 27:46–99, 2007.