Science of Computer Programming 218 (2022) 102791

Contents lists available at ScienceDirect

Science of Computer Programming

www.elsevier.com/locate/scico

High-level axioms for graphical linear algebra ☆

João Paixão [a,*], Lucas Rufino [a], Paweł Sobociński [b]

[a] *Universidade Federal do Rio de Janeiro, Avenida Athos da Silveira Ramos, 274, Cidade Universitaria, Rio de Janeiro, 21941-916, RJ, Brazil*
[b] *Tallinn University of Technology, Ehitajate tee 5, Tallinn, 12616, Estonia*

ARTICLE INFO

Article history:
Received 4 May 2021
Received in revised form 24 January 2022
Accepted 13 February 2022
Available online 23 February 2022

Keywords:
Graphical linear algebra
Calculational proofs
Diagrammatic language
String diagrams
Recursive linear algebra

ABSTRACT

We focus on a modular, graphical language—graphical linear algebra—and use it as high-level language for calculational reasoning. We propose a minimal framework of axioms that highlight the dualities and symmetries of linear algebra, and showcase the resulting diagrammatic calculus. Our work develops a relational approach to linear algebra, closely connected to classical relational algebra. With the symmetrical high-level axioms we are able to provide a fully diagrammatic proof that a fragment of Graphical Linear Algebra is equivalent to matrix algebra.

1. Introduction

Our goal is to develop the language of Graphical Linear Algebra (GLA), a diagrammatic approach to linear algebra. Its equational theory is the theory of Interacting Hopf Algebras [1,2], and it has been applied (and adapted) in a series of papers [3–8] to the modelling of signal flow graphs, Petri nets and even (non-passive) electrical circuits. In this paper, however, we focus on elucidating its status as an alternative *language* for linear algebra, external to specific applications. Some of this has been the focus of the third author's blog [9].

The diagrammatic syntax is extremely simple, yet powerful. Its simplicity is witnessed by the small number of concepts involved. Indeed, diagrams are built from just two structures, which we identify with black-white colouring. The black structure, roughly speaking, is *copying*, while the white structure is *adding*. Crucially, the interpretation of the diagrams is *relational*. Instead of vector spaces and linear maps, the semantic universe is vector spaces and linear *relations* [10–13]. It is somewhat astonishing that *all* other concepts are derived from only the interaction between these two simple structures.

One of our central contributions is the crystallisation of these interactions into just a few high-level axioms. We show that they suffice to reconstruct the entire equational theory of Interacting Hopf Algebras. A divergence from that work is our focus on *inequations*. Indeed, our axioms are inspired by the notion of *abelian bicategory* of Carboni and Walters [14]. Inequational reasoning leads to shorter and more concise proofs, as known in the relational community. For examples see [15] and [16], the latter with results closely related to ours. Importantly, in spite of having roots in deep mathematical

☆ This research was supported by the ESF funded Estonian IT Academy research measure (project 2014-2020.4.05.19-0001) and the Estonian Research Council grant PRG1210. This work was also supported by the following Brazilian research agencies: CAPES and CNPq. The second author is funded by the grant PIBIC (UFRJ) by CNPQ. We thank Jason Erbele, Gabriel Ferreira, and Iago Leal for fruitful conversations during various stages of this project.

* Corresponding author.
E-mail address: jpaixao@dcc.ufrj.br (J. Paixão).

https://doi.org/10.1016/j.scico.2022.102791

frameworks (functorial semantics [17,16], cartesian bicategories, abelian bicategories [14]), GLA is in fact a very simple system to work with: the rules of soundly manipulating diagrams can be easily taught to a student.

The semantics of diagrams is *compositional* – the meaning of a compound diagram is calculated from the meanings of its sub-diagrams. Here there is a connection with *relational algebra*, the two operations of diagram composition mapping to standard ways of composing relations: relational composition and cartesian product. In fact, the semantics of diagrams is captured as a monoidal *functor* from the category of diagrams GLA to the category of k-linear relations (where k stands for a field) $\mathbf{LinRel}_{\mathsf{k}}$:

$$\mathsf{GLA} \to \mathbf{LinRel}_{\mathsf{k}}$$

One of our favourite features of GLA is how it makes the underlying symmetries of linear algebra apparent. Two symmetries are immediately built into the syntax of diagrams: *(i) mirror-image*: any diagram can be "flipped around the y-axis", and *(ii) colour-swap*: any diagram can be replaced with its "photographic negative", swapping the black and white colouring. Thus one proof can sometimes result in four theorems. We feel like these important symmetries are sometimes hidden in traditional approaches. Moreover, some classical "theorems" become trivial consequences of the diagrammatic syntax.

The compositional nature of the syntax means that we can express many classical concepts in a straightforward and intuitive fashion. Traditional linear algebra, in spite of its underlying simplicity and elegance, has a tendency for conceptual and notational proliferation: Kernel, NullSpace, Image, sum of matrices, sum and intersection of vector spaces, etc. In GLA, all these concepts are expressible within the diagrammatic language. Moreover, the properties and relationships between the concepts can be derived via calculational proofs within the diagrammatic formalism. In many cases, these 2D proofs are shorter, more concise and more informative than their traditional 1D counterparts.

It is useful to explain in what sense our approach is original, and why we deem it worthwhile:

- we focus on the 2-dimensional language as a bona-fide syntax, and on the associated equational theory as a calculus for calculational reasoning in a style that could, eventually, be implemented in a theorem prover, and perhaps even lead to novel, compositional algorithms for linear algebra. Thus we impose on ourselves the constraint to use *only* the calculus, and not rely on semantic arguments, nor high-level structures such as distributive laws, nor other high-level categorical arguments;
- through experimenting with various ways of proving results, we highlight the axiomatic framework and proof techniques that we find the most convenient and powerful. By doing so, this paper is a step towards the convergence at a reasoning framework and the development of a related toolbox of techniques.

As an example of the power of the approach, we culminate with Theorem 59, which states that the "left to right" fragment of GLA exactly corresponds to the class of matrices. Although not in itself original, its proof has previously required [1] the use of distributive laws of props [18]. Our proof uses only the calculus. It is therefore "fully syntactic" and evidence of progress with respect to our stated goals.

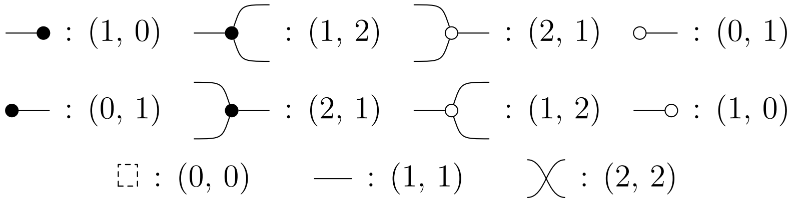
This article is a revised, extended version of the conference paper [19]. In that work we only sketched the main axioms and did not tackle the important subject of scalars. By developing the scalars, we are able to complete the axiomatisation (adding the important axiom that non-zero scalars are invertible). Moreover, the inductive techniques used for scalars lead to our treatment of matrices, and finally to the new proof of Theorem 59.

Structure of the paper In Section 2 we introduce the diagrammatic language, its semantics, and the six sound axioms (Axioms 1-6) for reasoning about linear relations. We then use those axioms in Section 3 to show that, together with Axiom 7, all of the equations of Interacting Hopf Algebras can be derived. In Section 4 we exhibit some familiar linear algebraic structure diagrammatically, and in Section 5 we show how the diagrammatic language deals with matrices. We conclude in Section 6 with some remarks about ongoing and future work.

2. Syntax and semantics of graphical linear algebra

In this section we introduce the diagrammatic language used throughout the paper. We explain how diagrams are constructed and outline basic principles of how to manipulate, and reason with them. Our diagrams are an instance of a particular class of *string diagrams*, which are well-known [20] to characterise the arrows of free strict monoidal categories. They are, therefore, rigorous mathematical objects.

We emphasise a syntactic approach: we generate string diagrams as terms and then quotient them with respect to the laws of strict symmetric monoidal categories. This quotienting process is particularly pleasant and corresponds to a topological understanding of diagrams where only the connectivity matters: intuitive deformations of diagrams do not change their meaning. Thus our work does not require extensive familiarity of (monoidal) category theory: the rules of constructing and manipulating the diagrams can be easily explained.

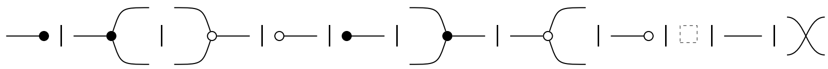


$$\frac{c:(k_1,k_2) \qquad d:(k_2,k_3)}{c\,;d:(k_1,k_3)} \qquad \frac{c:(k_1,l_1) \qquad d:(k_2,l_2)}{c\oplus d:(k_1+k_2,l_1+l_2)}$$

Fig. 1. The sorting discipline.

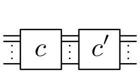
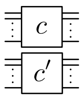
2.1. Syntax

Our starting point is the simple grammar for a language of diagrams, generated from the Backus-Naur form below.

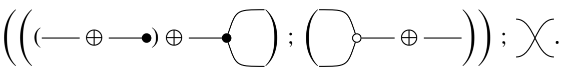
c, d ::=  | $c\,;d$ | $c \oplus d$ (1)

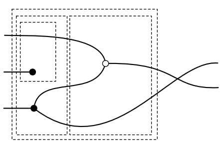
The first line of (1) consists of eight constants that we refer to as *generators*. Although they are given in diagrammatic form, for now we can consider them as mere symbols. Already here we can see two fundamental symmetries that are present throughout this paper: for every generator there is a "mirror image" generator (inverted in the "y-axis"), and for every generator there is a "colour inverse" generator (swapping black with white). Thus, it suffices to note that there are generators , and that the set of generators is closed with respect to the two symmetries. We shall discuss the formal semantics of the syntax in Section 2.2, but it is useful to already provide some intuition at this point. It is helpful to think of , as gates that, respectively, *discard* and *copy* the value on the left, and of , as *zero* and *add* gates. The mirror-image versions have the same intuitions, but from right-to-left.

The second line of (1) contains some structural terms and two binary operations for composing diagrams. The term is the empty diagram, is a wire and allows swapping the order of two wires. The two binary operations ; and $\oplus$ allow us to construct diagrams from smaller diagrams. Indeed, the diagrammatic convention is to draw $c\,;c'$ (or $c' \cdot c$) as series composition, connecting the "dangling" wires on the right of c with those on the left of c', and to draw $c \oplus c'$ as c stacked on top of c', (parallel composition), that is:

$c\,;c'$ is drawn  and $c \oplus c'$ is drawn .

The simple sorting discipline of Fig. 1 counts the "dangling" wires and thus ensures that the diagrammatic convention for ; makes sense. A *sort* is a pair of natural numbers (m, n): m counts the dangling wires on the left and n those on the right. We consider only those terms of (1) that have a sort. It is not difficult to show that if a term has a sort, it is unique.

Example 1. Consider the term . It has sort $(3, 2)$ and the diagrammatic representation is



where the dashed-line boxes play the role of disambiguating associativity of operations.

Symmetric strict monoidal categories Raw terms are quotiented with respect to the laws of symmetric strict monoidal (SSM) categories, summarised in Fig. 2. We omit the (well-known) details [20] here and mention only that this amounts to eschewing the need for "dotted line boxes" and ensuring that diagrams which have the same topological connectivity are equated.

We refer to terms-modulo-SSM-equations as *string diagrams*. The resulting string diagrams are then the arrows of a SSM category known as a PROP.

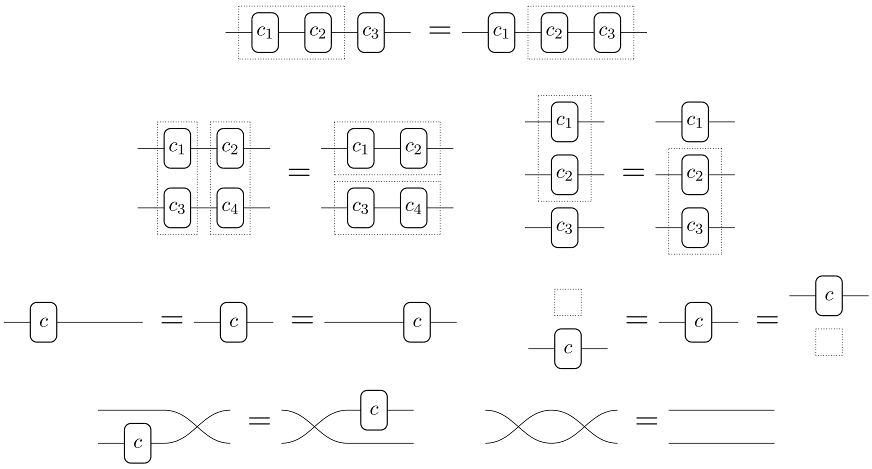
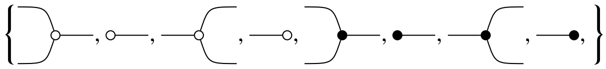


Fig. 2. Laws of Symmetric Monoidal Categories. Sort labels are omitted for readability.

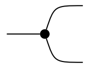
Definition 2. A PROP (products and permutations category) is a SSM category where the set of objects is the set of natural numbers, and, on objects $m \oplus n := m + n$. String diagrams generated from (1) are the arrows of a PROP GLA.

The idea is that the set of arrows from m to n is the set of string diagrams with m "dangling wires" on the left and n on the right.

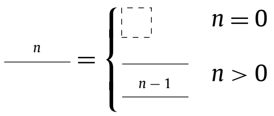
In fact, GLA is the *free* PROP on the set of generators


(2)

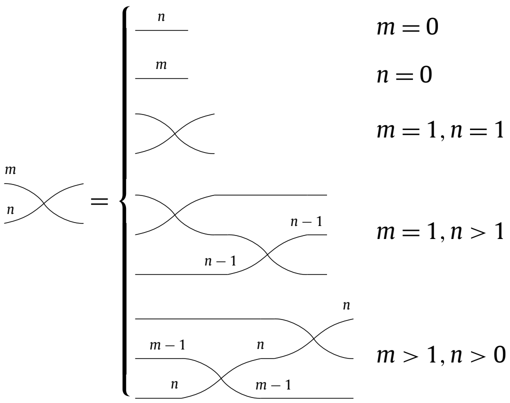
This perspective makes clear the status of GLA as a *syntax*: for example to define a morphism $\mathsf{GLA} \to \mathbb{X}$ to some general PROP $\mathbb{X}$, it suffices to define its action on the generators (2): the rest follows by induction.

We can also give recursive definitions. For example, the following generalisation of copying  will be useful in subsequent sections. Similar constructions can be given for the other generators in (2).

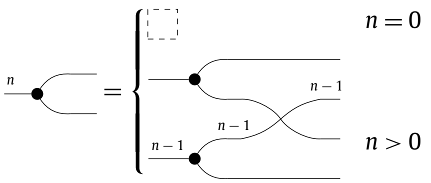
Definition 3 *(Generalized generators).*



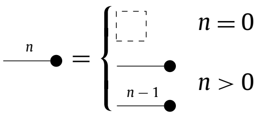
(3)



(4)



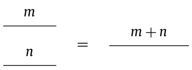
(5)



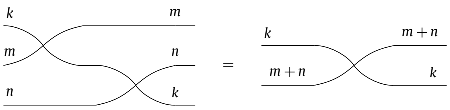
(6)

The next proposition follows via an easy induction argument.

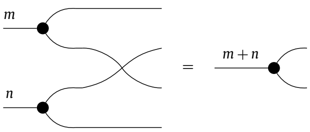
Proposition 4. *The types defined in Definition 3 are closed by the following operations.*



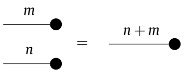
(7)



(8)



(9)



(10)

2.2. Semantics

We use GLA as a diagrammatic language for linear algebra. Differently from traditional developments, the graphical syntax has a relational meaning. Indeed, the central mathematical concept is that of *linear relation* [10–13]. Linear relations are also sometimes called *additive* relations [11] in the literature.

Definition 5. Fix a field k and k-vector spaces V, W. A linear relation from V to W is a set $R \subseteq V \times W$ that is a subspace of $V \times W$, considered as a k-vector space. Explicitly this means that:

1. $(0, 0) \in R$.
2. If $k \in \mathsf{k}$ and $(v, w) \in R$, $(kv, kw) \in R$.
3. If $(v, w), (v', w') \in R$, $(v + v', w + w') \in R$.

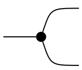
Given a field k, the PROP $\mathbf{LinRel}_{\mathsf{k}}$ is of particular interest.

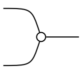
Definition 6. [1] The PROP $\mathbf{LinRel}_{\mathsf{k}}$ has as arrows $m \to n$ linear relations $R \subseteq \mathsf{k}^m \times \mathsf{k}^n$. Composition is relational composition, and monoidal product is given by direct sum.

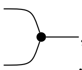
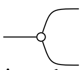
We now give the interpretation of the string diagrams of our language GLA as linear relations:

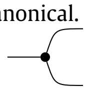
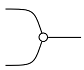
$$\mathsf{GLA} \to \mathbf{LinRel}_{\mathsf{k}} \tag{11}$$

Given that GLA is a free prop, it suffices to define the action on the generators.

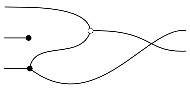
 $\longmapsto \left\{ \left(x, \begin{pmatrix} x \\ x \end{pmatrix} \right) \mid x \in \mathsf{k} \right\} \subseteq \mathsf{k} \times \mathsf{k}^2$ $\longmapsto \{(x, \star) \mid x \in \mathsf{k}\} \subseteq \mathsf{k} \times \mathsf{k}^0$

 $\longmapsto \left\{ \left(\begin{pmatrix} x \\ y \end{pmatrix}, x + y \right) \mid x, y \in \mathsf{k} \right\} \subseteq \mathsf{k}^2 \times \mathsf{k}$ $\longmapsto \{(\star, 0)\} \subseteq \mathsf{k}^0 \times \mathsf{k}$

The generators , ,  and map (respectively) to the opposite relations of the above, as hinted by the symmetric diagrammatic notation. As we shall see in the sequel, this semantic interpretation is canonical.

Intuitively, therefore, the *black structure* in diagrams refers to copying and discarding – indeed  is *copy* and is *discard*. Instead, the *white structure* refers to addition in k – indeed  is *add* and is *zero*.

Example 7. Consider the term of Example 1, reproduced below without the unnecessary dashed-line box annotations.

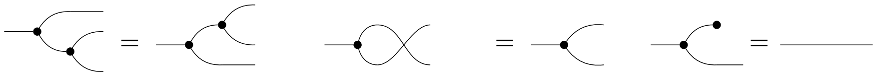


Its semantics is the (functional) relation $\left\{\left(\begin{pmatrix}x\\y\\z\end{pmatrix}, \begin{pmatrix}z\\x+z\end{pmatrix}\right) \mid x, y, z \in \mathsf{k}\right\}$.

2.3. Diagrammatic reasoning

In this section we identify the (in)equational theory of GLA. It is common to define algebraic structure in a PROP as a *monoidal theory* – roughly speaking, the PROP version of an equational algebraic theory. Our goal, instead, is to arrive at a powerful calculus for linear algebra, with few high level axioms. These are inspired by the notion of *abelian bicategory* as defined by Carboni and Walters [14].

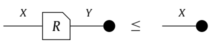
Axiom 1 *(Commutative comonoid).* The copy structure satisfies the equations of commutative comonoids, that is, *associativity*, *commutativity* and *unitality*, as given below:



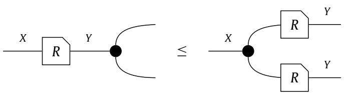
2.3.1. Inequalities

Instead of a purely equational axiomatisation, we find the use of inequalities very convenient in proofs. From a technical point of view this means the extension of the semantic mapping (11) to a (2-)functor between poset-enriched monoidal categories. Indeed, $\mathbf{LinRel}_{\mathsf{k}}$ has a natural poset-enrichment given by (set-theoretical) *inclusion* of relations.

We state three axioms below where R ranges over arbitrary GLA string diagrams. Given that R can have arbitrary sort, the axioms make use of Definition 3.

Axiom 2 *(Discard).* 

The axiom is sound with respect to (11). The left-hand-side denotes $LHS = \{(x, \star) \mid \exists y.\, (x, y) \in R\}$, while the right hand side denotes the relation $RHS = \{(x, \star) \mid \top\}$. Clearly $LHS \subseteq RHS$.

Axiom 3 *(Copy).* 

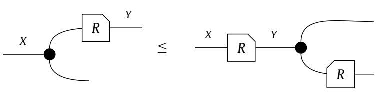
The left-hand-side is the relation $LHS = \{(x, \binom{y}{y}) \mid (x, y) \in R\}$, while the right-hand-side is $RHS = \{(x, \binom{y}{y'}) \mid (x, y), (x, y') \in R\}$ and soundness is again easy.

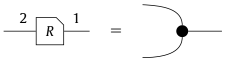
Proposition 8. $\{(x, \binom{y}{y}) \mid (x, y) \in R\} \subseteq \{(x, \binom{y}{y'}) \mid (x, y), (x, y') \in R\}$

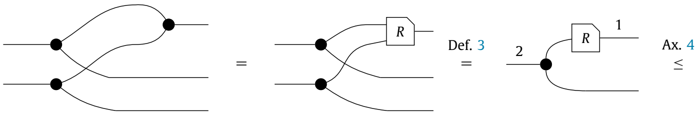
Proof.

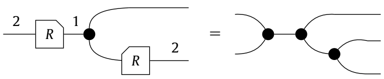
$$\begin{aligned}
& \{(x, \tbinom{y}{y}) \mid (x, y) \in R\} \\
= \quad & \{\ \text{Substitution}\ \} \\
& \{(x, \tbinom{y}{y'}) \mid (x, y), (x, y') \in R, y = y'\} \\
\subseteq \quad & \{\ \text{Trivial}\ \} \\
& \{(x, \tbinom{y}{y'}) \mid (x, y), (x, y') \in R\} \quad \square
\end{aligned}$$

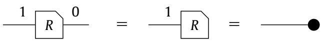
The diagrammatic notation for R allows us to represent the mirror image symmetry in an intuitive way. In the following diagram, the bottom right occurrence of R is the reflected diagram—which we write R^o in linear syntax—and which we have seen denotes the opposite relation under the mapping (11).

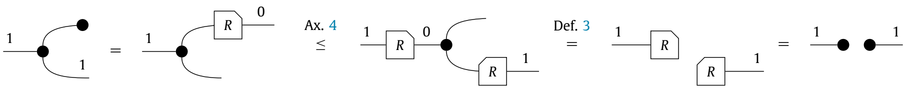
Axiom 4 *(Wrong way).* 

For example, let , then Axiom 4 says that





For another example, let , then Axiom 4 says that



Soundness of Axiom 4 is very similar to the soundness of Axiom 3.

Proposition 9. $\{(x, \binom{y}{x}) \mid (x, y) \in R\} \subseteq \{(x, \binom{y}{x'}) \mid (x, y), (x', y) \in R\}$.

Proof.

$$
\begin{aligned}
& \{(x, \tbinom{y}{x}) \mid (x, y) \in R\} \\
= \quad & \{\ \text{Substitution}\ \} \\
& \{(x, \tbinom{y}{x'}) \mid (x, y), (x', y) \in R, y' = x\} \\
\subseteq \quad & \{\ \text{Trivial}\ \} \\
& \{(x, \tbinom{y}{x'}) \mid (x, y), (x', y) \in R\} \quad \square
\end{aligned}
$$

The relationship between Axioms 3 and 4 is worth explaining. In tandem, they capture a topological intuition: a relation that is adjacent to a black node can “commute” with the node, resulting in a potentially larger relation. In Axiom 3 the relation approaches from the left. In Axiom 4 the relation comes from the “wrong side”, but it can still commute with the node to obtain a larger relation, but one must take care of the lower right wire that “curves around”. It is this “curving around” that results in the R^o.

2.3.2. Symmetries

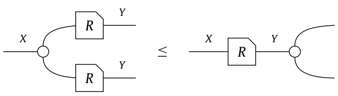
In Section 2.1, we saw that the generators of GLA are closed under two symmetries: the “mirror-image” $(-)^o$ and the “colour-swap”. Henceforth we denote the colour swap symmetry by $(-)^\dagger$. Clearly, for any R, we have $R^{oo} = R$ and $R^{\dagger\dagger} = R$. We are now ready to state two more axioms.

Axiom 5 *(Converse).* $R^o \leq S \iff R \leq S^o$, or in diagrams:

In particular, the mirror-image symmetry is *covariant*: if $R \leq S$ then $R^o \leq S^o$. Soundness is immediate.

Axiom 6 *(Colour inverse).* $R^\dagger \leq S \iff R \geq S^\dagger$, or in diagrams:

Thus the colour-swap symmetry is *contravariant*: if $R \leq S$ then $S^\dagger \leq R^\dagger$. Here soundness is not as obvious: one way is to show that the inequalities of Axioms 2, 3 and 4 reverse when it is the white structure that is under consideration. For example, “inverting the colours of Axiom 2”:

Proposition 10 *(“Axiom 3†”).* 

Proof. Axiom 6 applied to Axiom 3. □

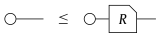
Soundness can be shown:

Proposition 11. $\{(x_1+x_2, \binom{y_1}{y_2}) \mid (x_1, y_1), (x_2, y_2) \in R\} \subseteq \{(x, \binom{y_1}{y_2}) \mid (x, y_1+y_2) \in R\}$.

Proof.

$$
\begin{aligned}
& \{(x_1+x_2, \tbinom{y_1}{y_2}) \mid (x_1, y_1), (x_2, y_2) \in R\} \\
\subseteq\ & \{\ R \text{ is a linear relation (Definition 5)}\} \\
& \{(x_1+x_2, \tbinom{y_1}{y_2}) \mid (x_1+x_2, y_1+y_2) \in R\} \\
\subseteq\ & \{\ x = x_1+x_2\ \} \\
& \{(x, \tbinom{y_1}{y_2}) \mid (x, y_1+y_2) \in R\} \quad \square
\end{aligned}
$$

Another example is “inverting the colours of Axiom 2”.

Proposition 12 *(“Axiom 2†”).* 

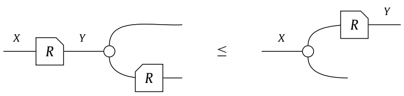
Proof. Axiom 6 applied to Axiom 2. □

Soundness can be shown:

Proposition 13. $\{(*, 0)\} \subseteq \{(*, x) \mid (0, x) \in R\}$.

Proof.

$$
\begin{aligned}
& \{(*, 0)\} \\
\subseteq\ & \{\ R \text{ is a linear relation (Definition 5)}\ \} \\
& \{(*, 0) \mid (0, 0) \in R\} \\
\subseteq\ & \{\ x = 0\ \} \\
& \{(*, x) \mid (0, x) \in R\} \quad \square
\end{aligned}
$$

Proposition 14 *(“Axiom 4†”).* 

Proof. Axiom 6 applied to Axiom 4. □

Soundness can be shown:

Proposition 15. $\{(x_1, \binom{y_1-y_2}{x_2}) \mid (x_1, y_1), (x_2, y_2) \in R\} \subseteq \{(x_1, \binom{y}{x_2}) \mid (x_1-x_2, y) \in R\}$.

Proof.

$$
\begin{aligned}
& \{(x_1, \tbinom{y_1-y_2}{x_2}) \mid (x_1, y_1), (x_2, y_2) \in R\} \\
\subseteq\ & \{\ R \text{ is a linear relation (Definition 5)}\ \} \\
& \{(x_1, \tbinom{y_1-y_2}{x_2}) \mid (x_1-x_2, y_1-y_2) \in R\} \\
\subseteq\ & \{\ y = y_1-y_2\ \} \\
& \{(x_1, \tbinom{y}{x_2}) \mid (x_1-x_2, y) \in R\} \quad \square
\end{aligned}
$$

2.4. Consequences of the symmetries

First of all, it is easy to see that the two operations commute.

Proposition 16. $(R^o)^\dagger = (R^\dagger)^o$.

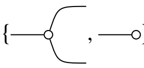
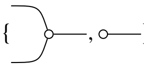
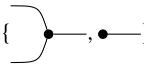
By the above proposition we can define the transpose operation.

Definition 17. $R^t := (R^o)^\dagger = (R^\dagger)^o$.

Prove one, get three theorems for free principle We will explore the symmetries of Axioms 5-6 throughout the paper. When we prove a result, we will usually assume we can use any of the other three theorems (its converse theorem, its colour inverse theorem, and its transpose theorem) afterwards just by referencing the original result. For example when we use, in a proof, the colour inverse of Axiom 3 (Proposition 10) we will simply denote Ax. 3).

3. Algebraic structure

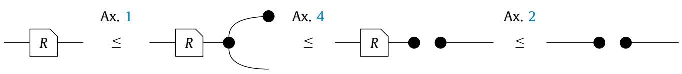
In this section we identify some of the algebraic structure that is a consequence of our the six axioms so far. The two structures that play an important role are *bialgebras* and *special Frobenius algebras*. As Lack explains in [18], these are two canonical ways in which monoids and comonoids can interact. See [18] for additional information on the provenance and importance of these two algebraic structures.

First note that it is an easy consequence of Axioms 1, 5 and 6 that the group of generators {  } is a commutative comonoid, and that {  }, {  } are both commutative monoids.

3.1. Top and bottom elements

Every hom-set, i.e. the set of (m, n) string diagrams for $m, n \in \mathbb{N}$, has a top and bottom element. Indeed:

Proposition 18 *(Top element).* 

Proof. 

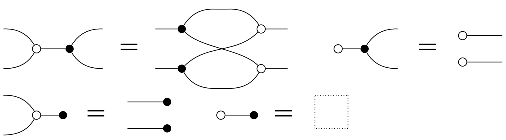
In the second step, we are using the second example described right after Axiom 4. □

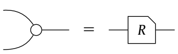
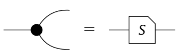
Dually, from Axiom 6 we obtain that is the bottom element.

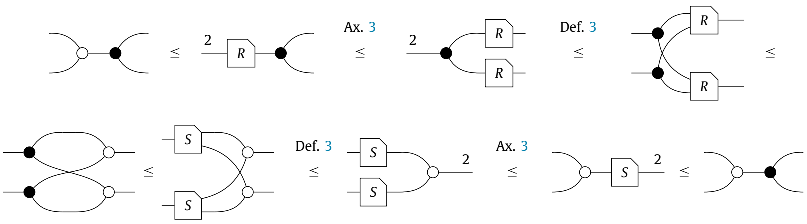
3.2. Bialgebra structure

White and black structures act as bialgebras when they interact. This can be summarised by the following equations. The following proof was inspired by [21].

Proposition 19 *(Bialgebra).*



Proof. Let  and . We have

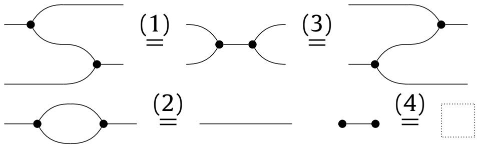


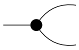
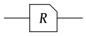
We use Axiom 3 in the first inequality and its colour inverse in the second inequality. Note that while using Axiom 3 in the proof, we must use Definition 3. The other derivations are similar. □

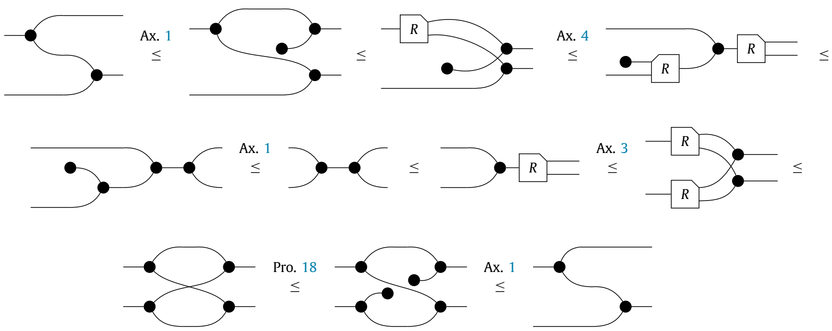
3.3. Frobenius structure

We have seen that the white and black structure interact according to the rules of bialgebras. On the other hand, individually the white and black structures interact as (extraspecial) Frobenius algebras.

Proposition 20 *(Frobenius). The following equations hold.*

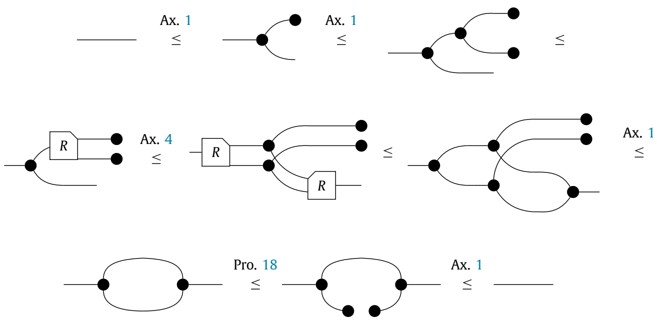


Proof. Let  =  . Then, we have the proof of Equation (1):



In the second step of the above proof, we are using the first example described right after Axiom 4.

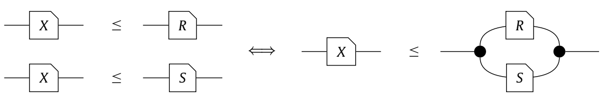
Proof of Equation (2):

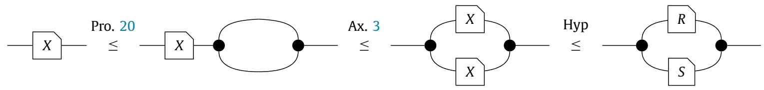


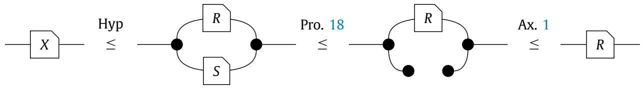
Equations (3) and (4) of the proposition can be proven similarly. □

With the previous proposition we can show now that the set of (m, n) string diagrams for $m, n \in \mathbb{N}$, has also a meet and join operation, expressible within the GLA structure, as we show below.

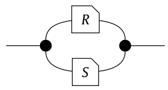
Proposition 21 *(Universal property of meet).*



Proof. $\Rightarrow$ 

$\Leftarrow$ 

The other inequality follows similarly. □

Corollary 22. *The partial order $\leq$ is a meet semilattice with top, where meet of R and S, written here as $R \cap S$, is*  *and the top element is given by Proposition 18.*

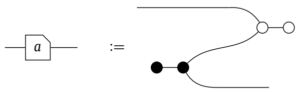
By duality of Axiom 6, we obtain the join and the bottom element just switching the colours. Therefore we have:

Corollary 23. *The partial order $\leq$ is a lattice with top, bottom, meet and join.*

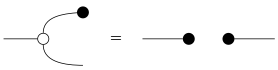
In the semantics, meet is intersection $R \cap S$, while join is the smallest subspace containing R and S: i.e. the subspace closure of the union $R \cup S$, usually denoted $R + S$.

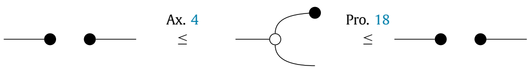
3.4. Antipode

One of the most surprising facts about this presentation of linear algebra is the plethora of concepts derivable from the basic components of copying and adding (black and white) structures. This includes the notion of *antipode* a, i.e. -1. Informally if the left wire of the antipode carries a value, then the right wire carries it opposite. The following propositions and definitions were inspired by [22].

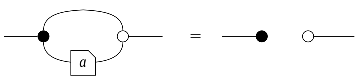
Definition 24 *(Antipode).* 

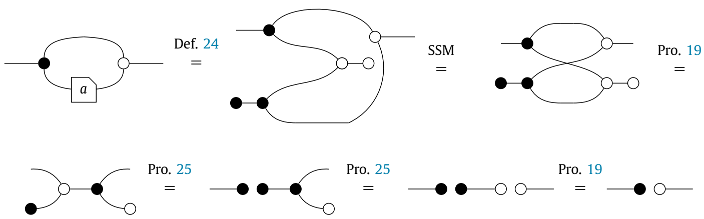
Before proving properties of the antipode, we prove a useful proposition.

Proposition 25. 

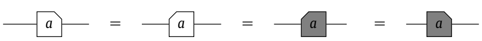
Proof. 

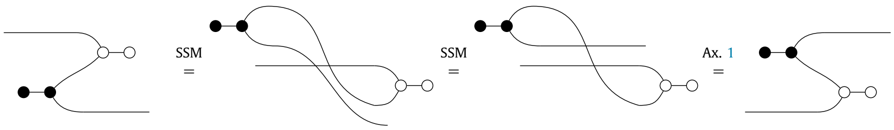
In the first step, we are using essentially the same idea from the second example of Axiom 4. □

Proposition 26. *(Hopf)* 

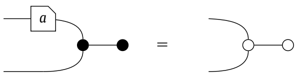
Proof. 

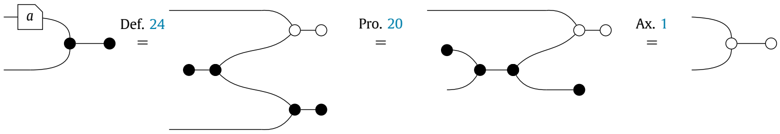
□

Proposition 27.  $(a = a^o = a^\dagger = a^t)$

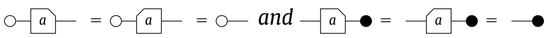
Proof. 

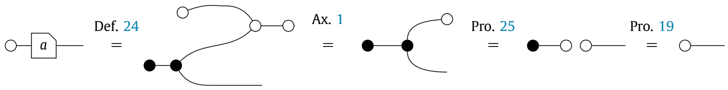
In the last step we are using the commutative property twice. All other equations are proven similarly. □

Proposition 28. 

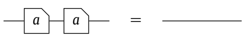
Proof. 

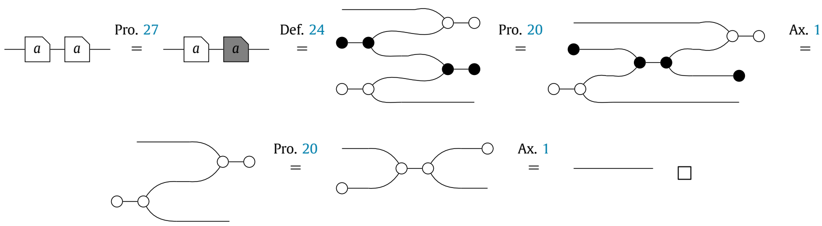
□

Proposition 29. 

Proof. 

By Proposition 27 the other three symmetric properties have symmetrical proofs. □

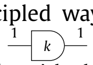
Proposition 30. 

Proof. 

□

3.5. Scalars

In the conference version [19] of this article scalars were not addressed in detail. Nevertheless, they are an important part of the algebra of IH, and deserve some attention.

Linear algebra is often understood as parametric over a fixed field k, say the real numbers or complex numbers. Given that string diagrams are finite, we have no hope of expressing arbitrary real numbers for cardinality reasons. In fact, there is a principled way of introducing the elements of any field into the diagrammatic language. In summary, an additional generator  is added for each $k \in \mathsf{k}$, and the axioms in Fig. 3, recalled from [1], ensure that the field operations are compatible with the diagrammatic algebra.

In [19] we implicitly assumed parametricity over k, in the above sense, giving a "scalar-free" account. In this article, instead, we develop the pure diagrammatic-view, addressing what results when considering merely adding and copying. In fact, the base language (of adding and copying) is powerful enough to express any *rational* number. However, an additional axiom, Axiom 7, has to be added to ensure that $1/k$ is the multiplicative inverse of k whenever $k \neq 0$. When parametrised over a fixed field, this axiom is redundant because invertibility is *already* a part of the field structure.[1] Therefore, Axiom 7, which will be presented after the introduction of natural numbers, can be replaced with the contents of Fig. 3.

To develop the diagrammatic theory of rational numbers, we first focus on the *natural numbers* as particular GLA diagrams of sort $(1, 1)$.

[1] Unfortunately, Theorem 7 as stated in [19] is wrong since it explicitly refers to the rationals.

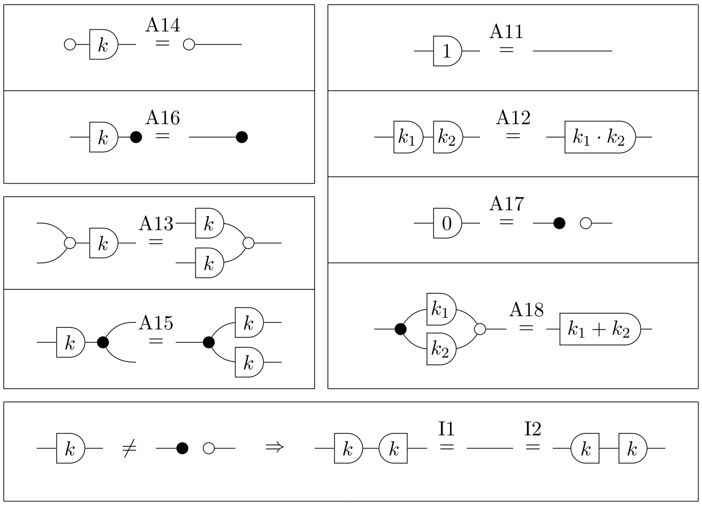
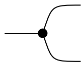
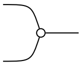
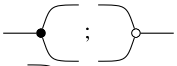
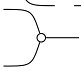
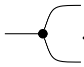
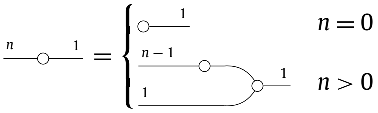


Fig. 3. Parametrising over a field k.

A natural number m in GLA is a (1, 1) diagram which, roughly speaking, begins with iterated  on the left, and ends with iterated  on the right. The idea is that m corresponds to a diagram with m paths from the left boundary to the right: 1 is identified with the identity, 2 with  and so on.

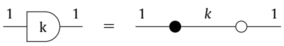
We first define recursively-defined types of iterated  and .

Definition 31.

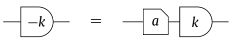


(12)

Diagrams that correspond to natural numbers can now be defined as follows.

Definition 32. 

Using this, one can easily construct a new definition for the whole type of integers by using the antipode.

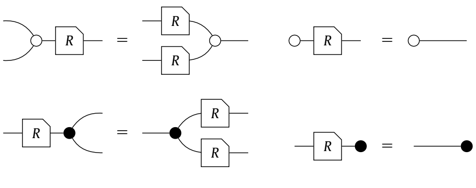


(13)

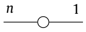
Since the antipode is invertible, its presence on the numerical structure has no relevant effect on proofs. Thus we shall be working with natural numbers for clarity purposes.

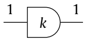
To elaborate on the properties of diagrammatic integers, we need to introduce some concepts. The next proposition states that a class of diagrams called Maps (Definition 46) enjoys a distributivity property wrt adding and copying.

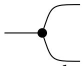
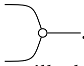
Proposition 33. *If R is a Map, the following equations hold:*

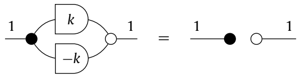


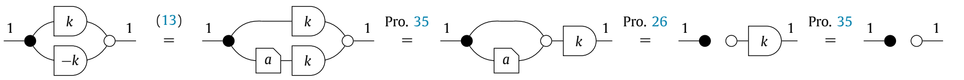
The following propositions can be easily proved by induction.

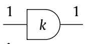
Proposition 34.  *is a Map.*

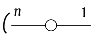
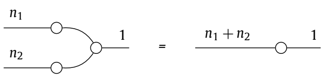
Proposition 35.  *is a Map.*

With this in place we can now make use of Proposition 33 in proofs concerning natural numbers. The fact that numbers are maps makes them particularly easy to handle, since we can just "slide" them through  and . The following proof will be an example of how earlier proofs can be extended quite easily to include numbers. It will also be the only proof concerning non-natural numbers in this paper.

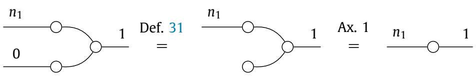
Proposition 36. 

Proof.  □

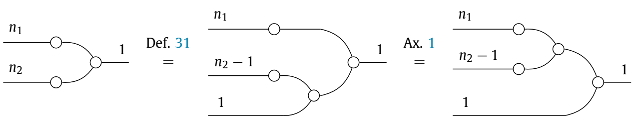
The type  is closed by addition and multiplication, which have a natural interpretation in the graphical language. Instead of using semantic information about the PROP of linear relations, or the existence of distributive laws (see e.g. [2]), here we use only the syntactic structure of GLA, focusing on recursive definitions and inductive proofs.

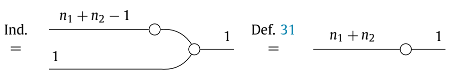
Proposition 37 *(* *closed by Sum).* 

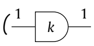
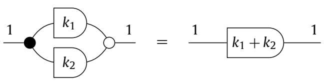
Proof. First, we do the base case where $n_2 = 0$.

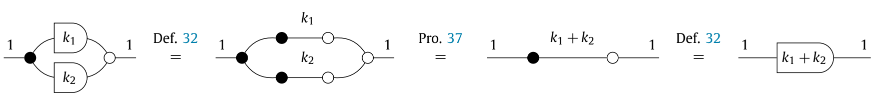


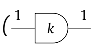
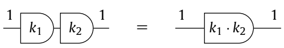
Now we do the inductive step, taking $n_2 > 0$.



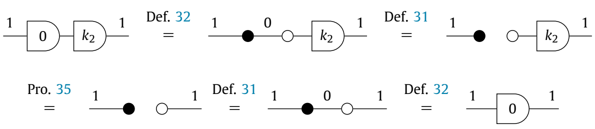
 □

Proposition 38 *(* *closed by addition).* 

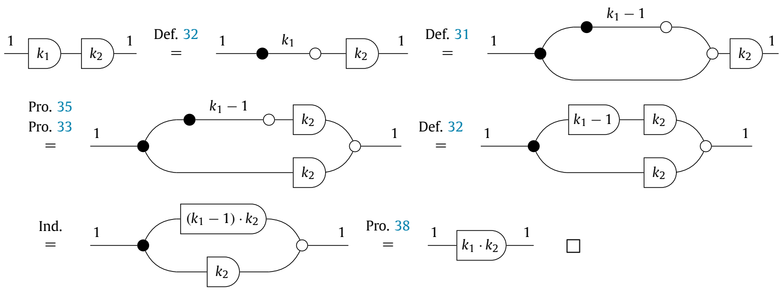
Proof.  □

Proposition 39 *(* *closed by composition).* 

Proof. This time, we are doing induction on k_1. So, first, in the base case, we take $k_1 = 0$.



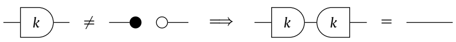
In the inductive step, we take $k_1 > 0$.



□

By using Propositions 26 and 30, we could also show that integers (as in equation (13)) are also closed by addition and composition.

Having identified the type of natural numbers within the graphical language we can state our last axiom.

Axiom 7. 

This axiom and its transpose ensure that all non-zero natural numbers are invertible.

The complete set of equations is known as the theory of Interacting Hopf (IH) Algebras [1]. In fact, given the previous results of this paper we have the main result of this section.

Theorem 40. *Axioms 1-7 suffice to derive all of the algebraic structure of Interacting Hopf (IH) algebras [1,2], reproduced here in Fig. 4. In fact, they are also sound in that theory. In particular, the PROP obtained from* GLA*, quotiented by the axioms, is* isomorphic *to* **LinRel**$_{\mathbb{Q}}$*, the PROP of linear relations over the rationals.*

Proof.
- (Black Monoid) Axiom 1 $\implies$ $A4$, $A5$, and $A6$
- (White Monoid) Axiom 1 and Axiom 6 $\implies$ $A1$, $A2$, and $A3$
- (Bialgebra) Proposition 19 $\implies$ $A7$, $A8$, $A9$, and $A10$.
- (Numbers) Definition 32 $\implies$ $A11$ and $A17$.
- (Numbers closed by composition) Proposition 39 $\implies$ $A12$.
- (Numbers are Maps) Proposition 33 and Proposition 35 $\implies$ $A13$, $A14$, $A15$ and $A16$.
- (Numbers closed by addition) Proposition 38 $\implies$ $A18$.
- (Invertible numbers) Axiom 7 $\implies$ $I1$.
- (Invertible numbers) Axiom 7^t $\implies$ $I2$.
- (Black Frobenius) Proposition 20 $\implies$ $I4$, $I7$ and $I9$.
- (White Frobenius) Proposition 20 and Axiom 6 $\implies$ $I3$, $I6$ and $I8$.
- (Antipode properties) Proposition 28 $\implies$ $I5$

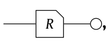
All the converse axioms are proven by Axiom 5. The fact that our axioms 1-6 are sound in that theory (i.e. that they can be derived from the axioms of Fig. 4) follows from the fact that they imply the structure of an abelian bicategory of relations [14], shown in [1, Section 5]. It is worthwhile to expand on a slight nuance here: in the purely equational theory, the black and the white structure satisfy the *same* axioms. Starting with 4, to recover the inequalities that play such an important role in our axioms, one uses the meet operation of Section 3: $R \leq S$ is defined to be $R = R \cap S$. Thus the choice of the black structure as the one that provides the meet is one way to break the black-white symmetry. The fact that this theory is isomorphic to the PROP of linear relations over the rationals is the main result of [1, Theorem 6.4]. □

The above result justifies our claims about the canonicity of the semantics (11).

4. A diagrammatic dictionary

The goal of this section is to provide a dictionary, showing how familiar linear algebraic concepts manifest in the graphical language. Given a diagram R, we first define some derived diagrams from R directly in GLA but which are inspired by classical relations and subspaces in linear algebra and relational algebra. Most of the results on this subsection are from [16], but we include here for completeness.

Definition 41 *(Derived diagrams).*

1. the nullspace of R: $N(R) :=$ ,

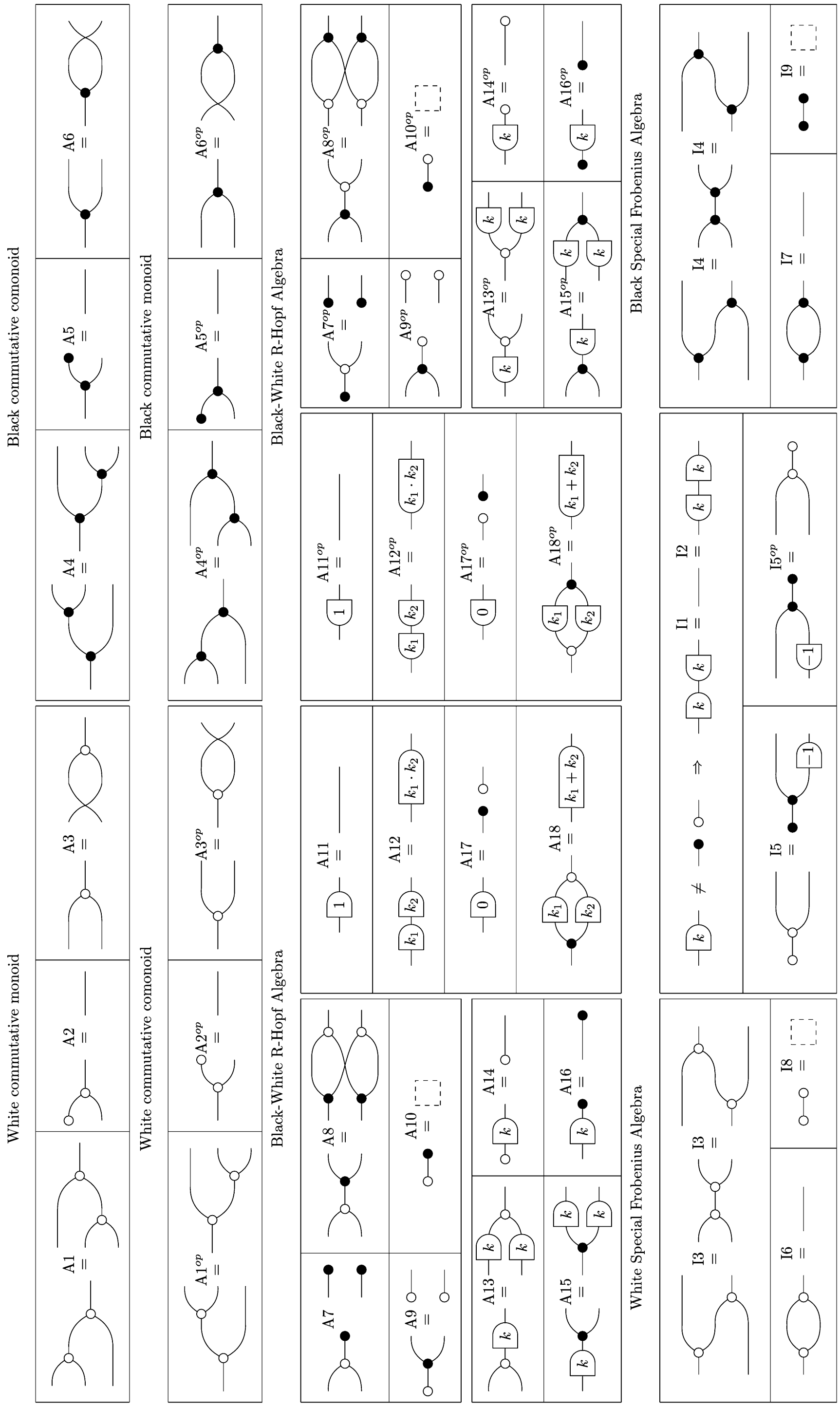
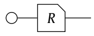
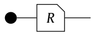
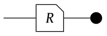
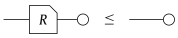
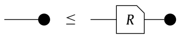
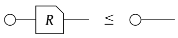
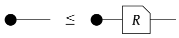


Fig. 4. Axiom of Interacting Hopf (IH) algebras (image adapted from [2]).

2. the multivalued part of R: $Mul(R) :=$  ,
3. the range of R: $Ran(R) :=$  ,
4. and the domain of R: $Dom(R) :=$  .

Definition 42 *(Dictionary).* Let R be a diagram. We call R

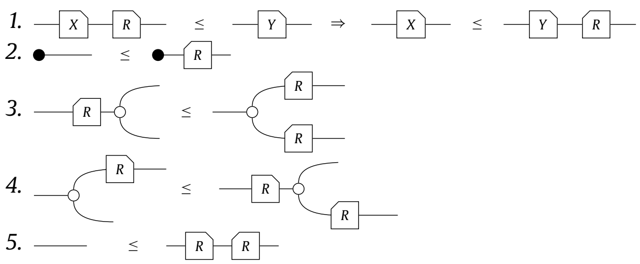
injective (INJ) if  ; total (TOT) if  ;
single-valued (SV) if  ; surjective (SUR) if  .

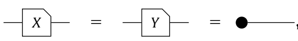
Moreover, since the converses of these inequalities hold in GLA, the four inequalities are actually equalities. With the definition above we clearly obtain a nice symmetry.

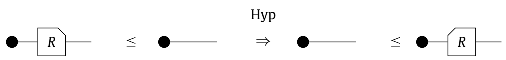
Proposition 43 *(Dictionary symmetry). R is total $\iff$ R^o is surjective $\iff$ $R^\dagger$ is injective $\iff$ $(R^\dagger)^o$ is single-valued.*

The next proposition shows that the notions in Definition 42 that were described through the converse inequalities of Axiom 2, can also be characterized by universal properties (item 1 from Proposition 44), the converse inequalities of Axioms 3 (item 3 from Proposition 44) and 4 (item 4 from Proposition 44), or by comparing the relation composed with its converse and the identity (item 5 from Proposition 44).

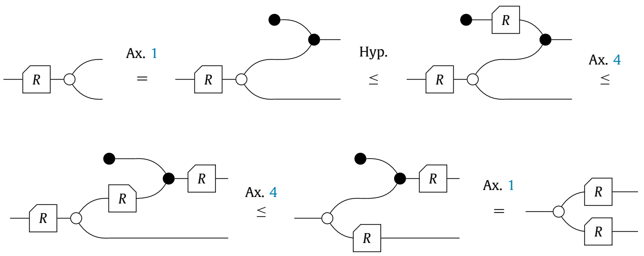
Proposition 44 *(Total). All the following statements are equivalent.*

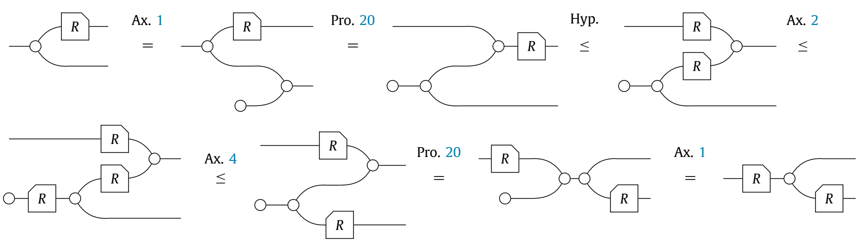


Proof. $(1 \Rightarrow 2)$
We can instantiate the hypothesis by taking , obtaining following implication:



The first term is Axiom 2, and the second term is what we were trying to prove.

$(2 \Rightarrow 3)$ 

$(3 \Rightarrow 4)$ 

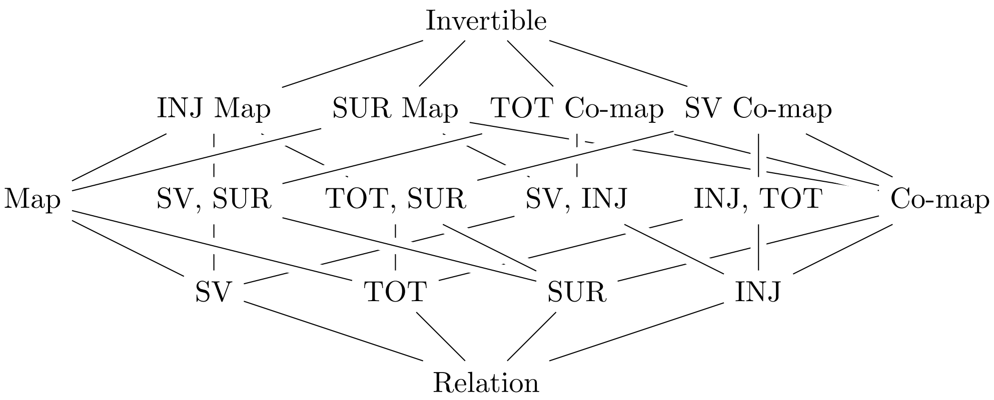
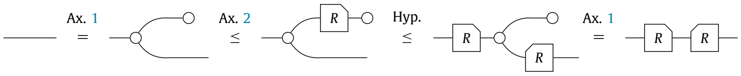
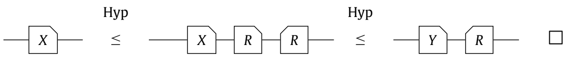


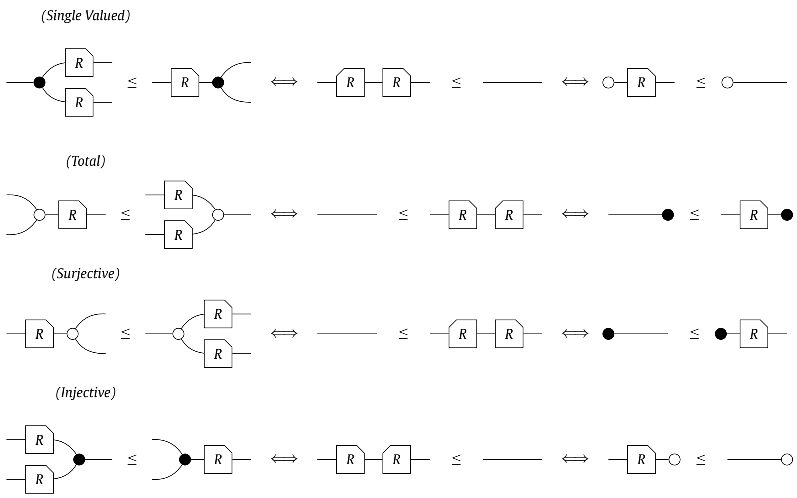
Fig. 5. Special relations.

(4 ⇒ 5) 

(5 ⇒ 1) 

With the symmetries described in Proposition 43, we are able to obtain the same characterizations for injective, single-valued and surjective.

Proposition 45.



From Proposition 45, it is easy to see the following proposition which will be useful in the next section.

Definition 46. A diagram is a Map if it is TOT and SV. Also, a diagram is a Co-map if it is INJ and SUR. A diagram is *invertible* if it is a Map and a Co-map (see Fig. 5).

For example, by Proposition 29, we have that the antipode is invertible.
The following proposition is very useful for the remainder of the paper.

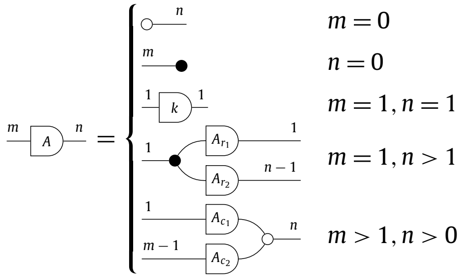
Proposition 47. *TOT, SV, INJ, SUR are properties closed by series and parallel composition (i.e., if A and B are TOT, then A; B is TOT and $A \oplus B$ is TOT).*

If we take the converse of this Proposition 33, we obtain a characterization for Co-maps.

5. Matrices

In this section we continue the recursive/inductive style of definitions and proofs to study the type of matrices in GLA. A matrix, diagrammatically, is intuitively a “generalised” scalar of Section 3.5: like scalars these are diagrams built up of black comonoid structure followed by white monoid structure. However, we do not consider merely (1, 1) diagrams, and allow arbitrary boundaries.

Definition 48 *(Mat).*



(14)

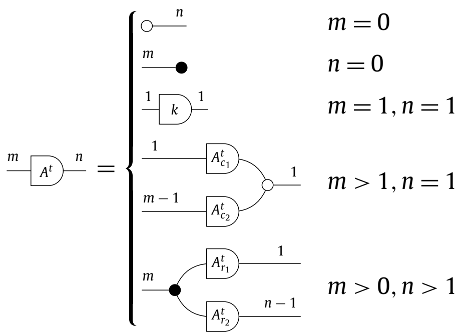
The definition above is inspired by the recursive definition of “classical” matrices:

Definition 49 *(Recursive definition of classical matrices).*

$$A(n,m) = \begin{cases} k \in \mathsf{k} & m = 1, n = 1 \\ \begin{bmatrix} A_{r_1}(1,1) \\ A_{r_2}(n-1,1) \end{bmatrix} & m = 1, n > 1 \\ \begin{bmatrix} A_{c_1}(n,1) & A_{c_2}(n,m-1) \end{bmatrix} & m > 1, n > 0 \end{cases}$$

This definition is similar to defining a matrix as a list of column vectors. We can also represent a matrix as a list of row vectors.

Proposition 50 *(Mat^t).*



(15)

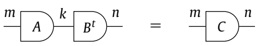
We prove the equivalence of these two definitions in Pro. 55. As in the case of scalars (Proposition 35), the next propositions follow via straightforward induction arguments.

Proposition 51. *Mat is a Map (TOT and SV).*

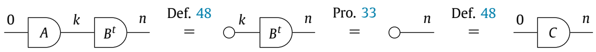
Proposition 52. *Mat^t is a Map (TOT and SV).*

Again, similar to natural numbers, with these propositions we can use Proposition 33 for regular and transposed matrices.

Our goal is to give an inductive proof that the type Mat is closed by the algebraic operations of sum and multiplication, which correspond to the usual operations on matrices. We start by showing that Mat is closed by composition with its transpose.

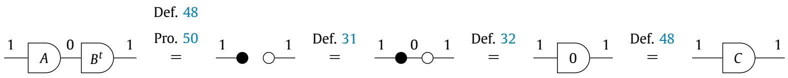
Proposition 53 *(Mat ; Mat^t = Mat).* 

Proof. Here, since there are multiple parameters, we will need to do more than one induction. First, we will consider the trivial cases where $m = 0$ and $n = 0$, respectively. The proof for $m = 0$ is the following:

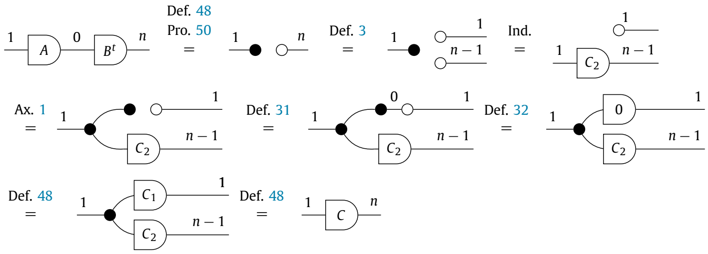


The case where $n = 0$ is completely symmetric, so there is no need to draw its proof.

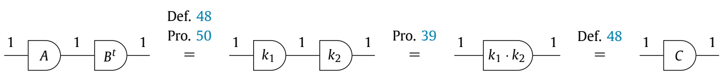
Now we shall prove the case where $k = 0$. In this one, we will require an induction on m and n. First, we do the base case where $m = n = 1$ and $k = 0$.



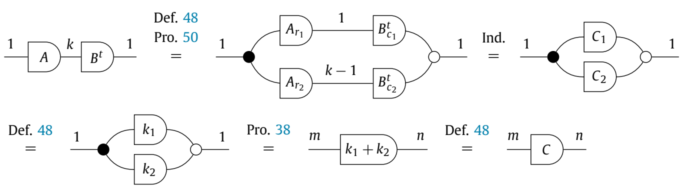
Now we do the induction on n:



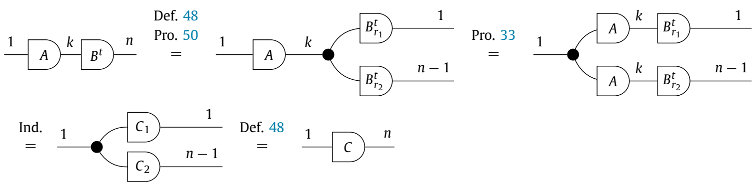
The induction on m is symmetric. After considering all the cases where a parameter equals 0, we will do one more base case where $m = n = k = 1$ and then do an induction on each one of the parameters. So, first, the base case:



Now, we do the induction on k.



Now, the induction on n.



Algorithm 1 Matrix multiplication.

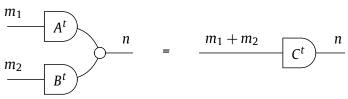
procedure MULT($B(n,k)$, $A(k,m)$)

  if $m > 1$ **then** ▷ Split A "vertically"
    $\begin{bmatrix} A_1 & A_2 \end{bmatrix} \leftarrow A$
    $C \leftarrow \begin{bmatrix} \text{MULT}(B, A_1) & \text{MULT}(B, A_2) \end{bmatrix}$
  else if $n > 1$ **then** ▷ Split B "horizontally"
    $\begin{bmatrix} B_1 \\ B_2 \end{bmatrix} \leftarrow B$
    $C \leftarrow \begin{bmatrix} \text{MULT}(B_1, A) \\ \text{MULT}(B_2, A) \end{bmatrix}$
  else if $k > 1$ **then** ▷ Split A "horizontally" and B "vertically"
    $\begin{bmatrix} B_1 & B_2 \end{bmatrix} \leftarrow B$
    $\begin{bmatrix} A_1 \\ A_2 \end{bmatrix} \leftarrow A$
    $C \leftarrow \text{MULT}(B_1, A_1) + \text{MULT}(B_2, A_2)$
  else ▷ "Regular" multiplication of numbers($m = 1, k = 1, n = 1$)
    $C \leftarrow B \cdot A$
  return C

Lastly, the induction on m is symmetric to the previous one. And this completes the proof. □

This proof actually mimics a recursive matrix multiplication Algorithm 1. This can be seen as a demonstration that working with the diagrammatic language can yield compositional algorithms for linear algebra.

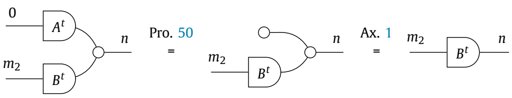
Proposition 54 *(Mat^t closed by sum).*



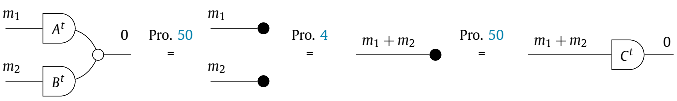
In the classical language, this operation can be viewed as stacking the two matrices side by side:

$$\begin{bmatrix} A^t & B^t \end{bmatrix} = C^t \tag{16}$$

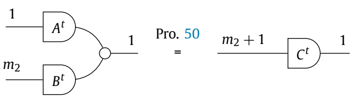
Proof. Here, we do induction on m_1 and n, while leaving m_2 free. First, we do the trivial cases where $m = 0$ and $n = 0$, respectively.



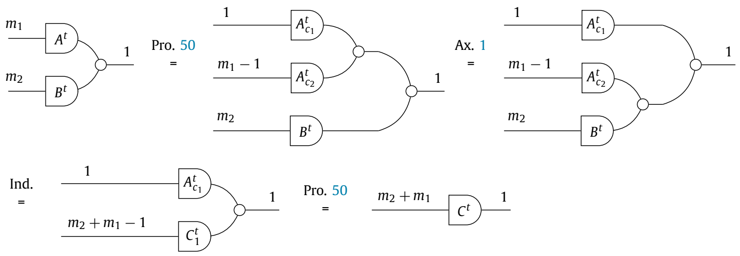
Now, the case where $n = 0$.



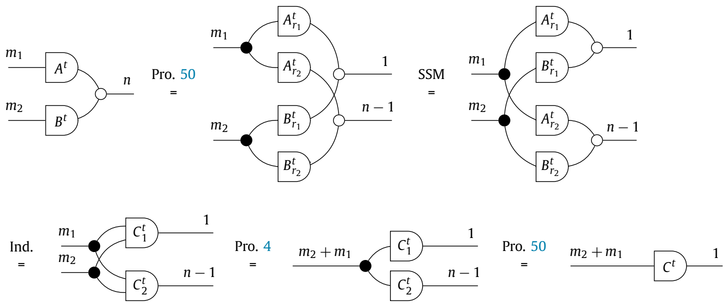
After doing the trivial cases, we prove the base case for the induction. Specifically, the case where $m_1 = n = 1$.



Now we do the induction on m_1.

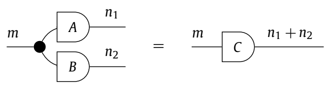


Finally, the induction on n.



□

Notice that, by proving that the type Mat^t is closed by sum we also get for free its transposed version, namely that the type Mat is closed by copy.

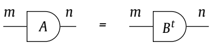


(17)

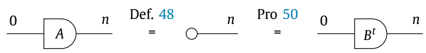
As one would expect, in the classical language it is written as:

$$\begin{bmatrix} A \\ B \end{bmatrix} = C \tag{18}$$

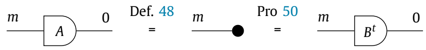
We will be referencing Pro. 54 for both results (sum and copy). The same will hold true for any proposition that has a relevant transposed version.

Proposition 55 *(Mat = Mat^t).* 

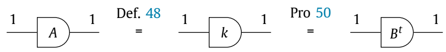
Proof. First, as always, we do the cases where $m = 0$ and $n = 0$, respectively.



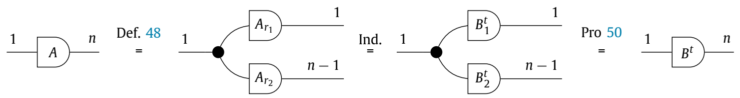
Now, the case where $n = 0$.



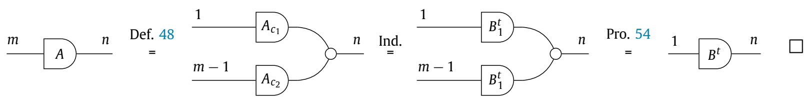
Now, the base case for the induction, where $n = m = 1$.

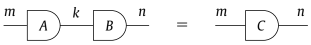


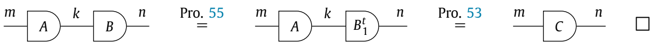
Induction on n.



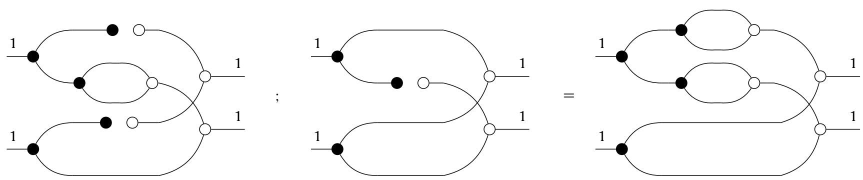
Lastly, induction on m.



Corollary 56 *(Mat is closed by composition).* 

Proof. 

One can view this proof as an algorithm for matrix multiplication, where A and B are the inputs and C is the output. The following image is an example of this operation.

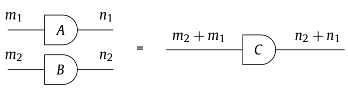


The equivalent way of writing it in the classical language is:

$$\begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \times \begin{bmatrix} 0 & 0 \\ 2 & 1 \end{bmatrix} = \begin{bmatrix} 2 & 1 \\ 2 & 1 \end{bmatrix}$$

Notice the relation between the operations ; and $\times$. Classically, matrices are composed from right to left, but in the graphical language things are done in the opposite direction. So the first equation is written as $A; B = C$, but the second is $B \times A = C$.

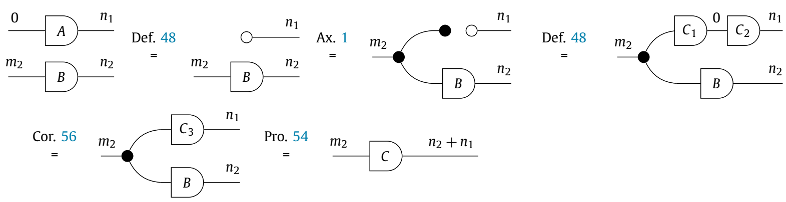
After seeing the example above, one may notice that there is a nice way of comparing classical matrices to their graphical counterparts. The quantity A_{ij} can be determined simply by counting the number of paths from the j-th input wire to the i-th output wire. Similarly, a scalar k in the graphical language is a diagram with 1 input wire and 1 output wire, where the number of paths from left to right is exactly k.

Proposition 57 *(Mat closed by parallel composition).* 

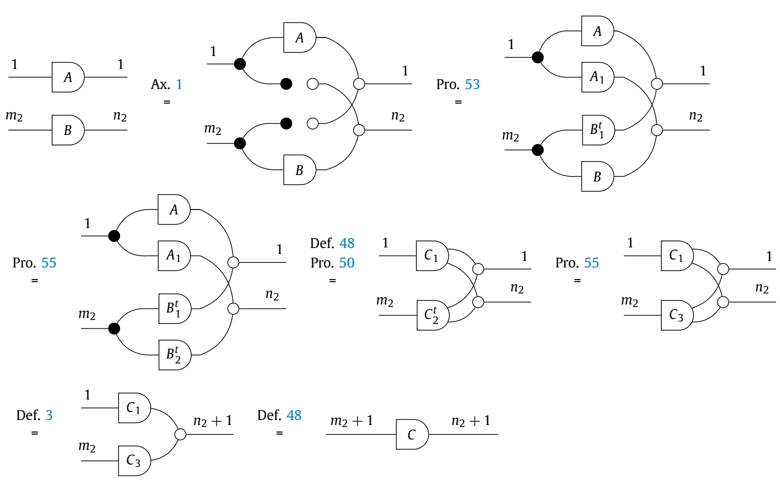
Parallel composition is the same as direct sum in the classical language. The equivalent statement would be:

$$\begin{bmatrix} A & 0 \\ 0 & B \end{bmatrix} = C \tag{19}$$

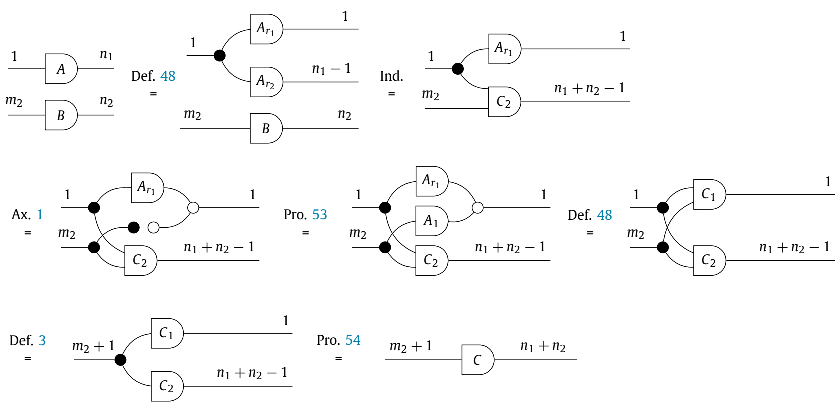
Proof. Here, we will do induction on the parameters m_1 and n_1. First, starting with the cases where $m_1 = 0$ or $n_1 = 0$, respectively.



The case where $n_1 = 0$ is symmetric. Now, the base case for the induction, where $m_1 = n_1 = 1$.



Now, induction on n_1.

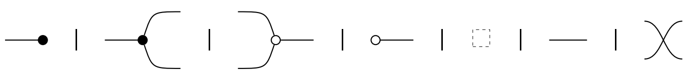


Lastly, induction on m_1 is symmetric. This completes the proof. $\square$

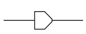
Notice that, in the last step, we use Pro. 54 to reference its transposed version shown in (17).

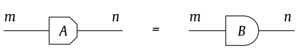
We can now showcase the power of the toolbox of results we have developed. Consider the following PROP of (bicommutative) Hopf algebras (HA), which intuitively uses those elements of (1) where the causal flow is left-to-right, i.e. the inputs are on the left and the outputs are on the right.

Definition 58 *(HA).*

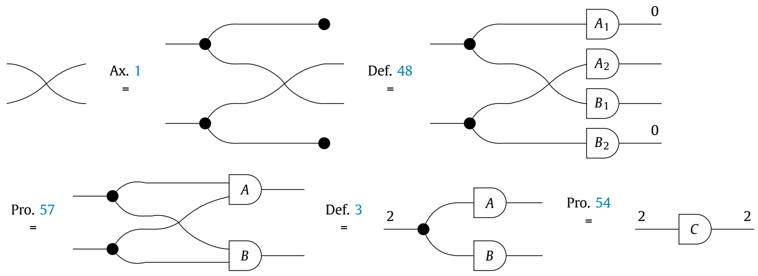
$c, d \quad ::=$  $\mid \; c\,;d \; \mid \; c \oplus d$ (20)

The toolbox of results proved above can now be used to give a purely “syntactic” proof of a known result, namely that every arrow of HA is equal to a matrix.

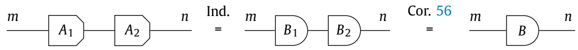
In order to avoid confusion, we will use the symbol  to denote a diagram of type HA. Since it is a freely generated type, we can use structural induction to prove the following.

Theorem 59 *(HA = Mat).* 

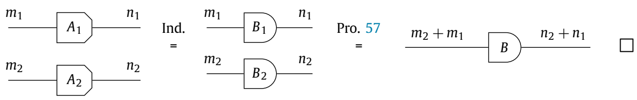
Proof. The base cases will be on the generators. We will be showing only one case, since the other ones are quite easy to do.



Now, in the inductive step, first we consider the composition case.

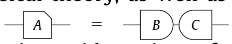


Finally, the parallel composition case.



6. Conclusion and future work

We showcased *Graphical Linear Algebra*, a simple bichromatic diagrammatic language for linear algebra. We introduced a high level axiomatisation—which we argue is especially convenient for use in calculational proofs—and showed that it suffices to derive all of the algebraic structure of the theory of Interacting Hopf Algebras. We also focused on the modular nature of the language and how it captures many of the classical concepts of linear algebra.

We have shown, in a constructive way, a characterization of HA under the inductive types of Mat which correspond to matrices defined recursively. The next step is going to be constructing a characterization for general relations in GLA. A direction where future work will be especially fruitful is a completely diagrammatic descriptions of various other normal form and matrix factorisations (LU Decomposition, Cholesky Decomposition, Smith Normal Form, etc), both in elucidating the classical theory, as well as obtaining useful relational generalisations of Theorem 59 again in a purely "syntactic" proof such as  [2,1].

There is a wide variety of works that facilitate reasoning with arrays, matrices, and linear algebraic applications in the context of principled approaches to programming [23–25]. We believe that diagrammatic methods can be influential in the design of new principled programming approaches, both by suggesting novel programming constructs as well as providing efficient data structures. Future work will focus on integrating our techniques with the aforementioned traditions.

CRediT authorship contribution statement

All the authors participated in every aspect of the paper.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] F. Bonchi, P. Sobociński, F. Zanasi, Interacting Hopf algebras, J. Pure Appl. Algebra 221 (1) (2017) 144–184.
[2] F. Zanasi, Interacting Hopf algebras: the theory of linear systems, Ph.D. thesis, Ecole Normale Supérieure de Lyon, 2015.
[3] J. Baez, J. Erbele, Categories in control, Theory Appl. Categ. 30 (2015) 836–881.
[4] F. Bonchi, P. Sobociński, F. Zanasi, A categorical semantics of signal flow graphs, in: International Conference on Concurrency Theory, Springer, 2014, pp. 435–450.
[5] F. Bonchi, P. Sobociński, F. Zanasi, Full abstraction for signal flow graphs, in: POPL 2015, ACM, 2015, pp. 515–526.
[6] F. Bonchi, J. Holland, D. Pavlovic, P. Sobociński, Refinement for signal flow graphs, in: 28th International Conference on Concurrency Theory (CONCUR 2017), Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2017.
[7] F. Bonchi, R. Piedeleu, P. Sobociński, F. Zanasi, Graphical affine algebra, in: ACM/IEEE Symposium on Logic and Computer Science (LiCS '19), 2019.
[8] F. Bonchi, J. Holland, R. Piedeleu, P. Sobociński, F. Zanasi, Diagrammatic algebra: from linear to concurrent systems, Proc. ACM Program. Lang. 3 (POPL) (2019) 1–28.

[9] P. Sobociński, Graphical linear algebra, mathematical blog [online], available: https://graphicallinearalgebra.net.
[10] R. Arens, et al., Operational calculus of linear relations, Pac. J. Math. 11 (1) (1961) 9–23.
[11] S. Mac Lane, An algebra of additive relations, Proc. Natl. Acad. Sci. USA 47 (7) (1961) 1043.
[12] E.A. Coddington, Extension Theory of Formally Normal and Symmetric Subspaces, vol. 134, American Mathematical Soc., 1973.
[13] R. Cross, Multivalued Linear Operators, vol. 213, CRC Press, 1998.
[14] A. Carboni, R.F. Walters, Cartesian bicategories I, J. Pure Appl. Algebra 49 (1–2) (1987) 11–32.
[15] P.J. Freyd, A. Scedrov, Categories, Allegories, vol. 39, Elsevier, 1990.
[16] F. Bonchi, D. Pavlovic, P. Sobociński, Functorial semantics for relational theories, arXiv preprint, arXiv:1711.08699, 2017.
[17] F.W. Lawvere, Functorial semantics of algebraic theories, Proc. Natl. Acad. Sci. 50 (5) (1963) 869–872.
[18] S. Lack, Composing props, Theory Appl. Categ. 13 (9) (2004) 147–163.
[19] J. Paixão, P. Sobociński, Calculational proofs in relational graphical linear algebra, in: Brazilian Symposium on Formal Methods, Springer, 2020, pp. 83–100.
[20] P. Selinger, A survey of graphical languages for monoidal categories, in: New Structures for Physics, Springer, 2010, pp. 289–355.
[21] J. Baez, Bimonoids from biproducts, https://golem.ph.utexas.edu/category/2010/09/bimonoids_from_biproducts.html, Sep 2010.
[22] J. Erbele, Redundancy and zebra snakes, https://graphicallinearalgebra.net/2017/10/23/episode-r1-redundancy-and-zebra-snakes/.
[23] K. Berkling, Arrays and the lambda calculus, in: Arrays, Functional Languages, and Parallel Systems, Springer, 1991, pp. 1–17.
[24] P.W. O'Hearn, J.C. Reynolds, From Algol to polymorphic linear lambda-calculus, J. ACM (JACM) 47 (1) (2000) 167–223.
[25] J.C. Reynolds, Reasoning about arrays, Commun. ACM 22 (5) (1979) 290–299.