University of Southampton

Faculty of Engineering and Physical Science
School of Electronics and Computer Science

Defining Partial Orders on Graphical Models of Concurrent Systems

by
Joshua Holland

A thesis for the degree of
Master of Philosophy

November 2021

University of Southampton

Abstract

Faculty of Engineering and Physical Sciences
School of Electronics and Computer Science

Master of Philosophy

Defining Partial Orders on Graphical Models of Concurrent Systems

by Joshua Holland

Our interest is in models of concurrency, and their theoretical axiomatisation and analysis. We build on a rich thread of research [BSZ14, FSR16, BSZ17a] interpreting models such as Petri nets as so-called *string diagrams*, a notation for morphisms of symmetric monoidal categories. From there, we can use structure-preserving mappings between the model and a semantic domain. The main contribution of the thesis is the definition of a *symmetric monoidal inequality theory*, which extends the standard tool used in this field to handle inequalities. Armed with this, we answer more questions about systems than just whether they have the same behaviours, such as describing specifications which leave open ambiguity or choices for implementors, proofs that systems satisfy such a specification (or not), and demonstrations that one system exhibits some (but not necessarily all) behaviours of another.

Contents

Declaration of Authorship

I declare that this thesis and the work presented in it is my own and has been generated by me as the result of my own original research.

I confirm that:

1. This work was done wholly or mainly while in candidature for a research degree at this University;
2. Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated;
3. Where I have consulted the published work of others, this is always clearly attributed;

4. Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work;
5. I have acknowledged all main sources of help;
6. Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself;
7. Parts of this work have been published as:
    - Filippo Bonchi, Joshua Holland, Robin Piedeleu, Paweł Sobociński, and Fabio Zanasi. Diagrammatic algebra: From linear to concurrent systems. In *46th ACM SIGPLAN Symposium on Principles of Programming Languages (POPL 2019)*, 2019
    - Filippo Bonchi, Joshua Holland, Dusko Pavlovic, and Paweł Sobociński. Refinement for signal flow graphs. In Roland Meyer and Uwe Nestmann, editors, *28th International Conference on Concurrency Theory (CONCUR 2017)*, volume 85 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 24:1–24:16, Dagstuhl, Germany, 2017. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik

Signed:... Date:..................

Chapter 1

Introduction

In the world of concurrent systems, various graphical languages are used to design and specify required behaviour. One formalism that has seen some success is the use of categories and relations as semantics [BSZ14, FSR16, BSZ17a]. Central to this theory is viewing systems *compositionally*: understanding them as being built up (or *composed*) from simple parts. To that end, the starting point for these theories is a collection of extremely simple *generators*, together with some basic *equations* which impose relationships among them. However, this paradigm can only talk about equality, and answer questions about whether two systems have the same behaviour or not.

One of the key features of relations is *inclusion*, but so far in the treatment of graphical languages, there is no accounting for this. It is testament to the importance of inclusion to working with relations that it is perhaps more common to prove that relations are equal by mutual inclusion than directly. However, until now, there was no way to replicate that proof technique for the graphical languages. This thesis addresses that deficiency, and applies our new tools to signal flow graphs and Petri nets.

We start by going over basics of general category theory, then focus on the specific categories which we will use: symmetric monoidal categories and props. A vanilla category has just one (associative and unital) operation, *composition*. It is common to use this to represent the passing of data from the output of one process as the input to the next, but we are interested in systems where we can also compose systems in parallel. To model this, we move to *symmetric monoidal* categories, which have an additional operation which represents the parallel composition we want. In fact, we restrict ourselves to an especially narrow form of symmetric monoidal category, called the *prop*. A prop is a particularly simple kind of symmetric monoidal category, which contains the structure we need to represent our concurrent systems and no extra baggage. We demonstrate that props are sufficient for our purposes by defining all of the props we will use for semantics: various different kinds of *relations*.

The use of relations (subsets of the Cartesian product) rather than the functions is an important point. Functions are everywhere in mathematics and computer science. Why do we choose to depart from such a successful model of computation as 'every possible input is mapped to a unique output'? For one, this is not so much a departure as a generalisation: every function can be viewed as a relation[1]. But also, more importantly, this is a viewpoint that often cannot be sustained in the world of concurrent systems. Perhaps a system can become deadlocked and unable to accept any input; it may enter a state where not all the possible inputs are allowed; one input may have multiple potential outputs (non-determinism); or indeed any combination of these is possible. These are real-world situations which the functional perspective cannot capture but relational models can.

The other hallmark of this vein of research is the usage of *string diagrams*. This is the graphical part of the thesis, and we will use them at every opportunity. Formally, they are a notation for morphisms of

[1] by considering its *graph*: the collection of pairs $(x, f(x))$.

props[2]; for our purposes, they are a uniform mathematical language for specifying, defining and proving properties of concurrent systems.

String diagrams are well-suited to the compositional approach for several reasons: they are already graphically similar to the languages of concurrency we are examining; they naturally encode the laws of symmetric monoidal categories; and they are intuitive to work with. We hope to justify the last of these throughout the thesis, and we will give a brief overview of the former two points now.

To do this, obviously it is necessary to show what a string diagram actually looks like: indeed, we are already on Page of a thesis entitled *Defining Partial Orders on Graphical Models of Concurrent Systems* and we have not yet shown anything graphical! We'll begin with Figure 1.1, a reproduction of Figure 1 from Peterson's survey article on Petri nets [Pet77], which predates even the definition of string diagrams, let alone their application to this field.

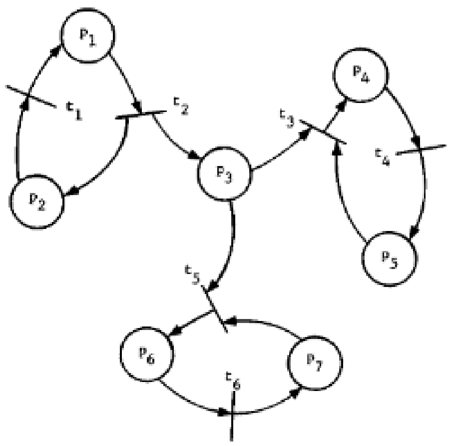


Figure 1.1: A Petri net, as construed in the late 1970s

In the notation defined in Chapter 5, this takes the form of Figure 1.2, which really is just a morphism in a particular prop, and is therefore amenable to the full suite of compositional analysis.

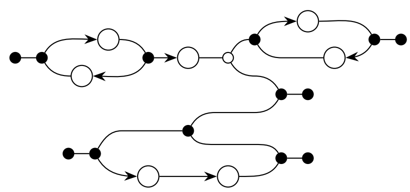


Figure 1.2: The string diagram corresponding to the Petri net in Figure 1.1

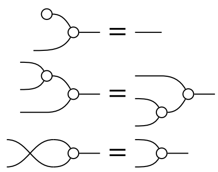
We will fully describe the notation in the sequel, but the round places are clearly visible, and the transitions t_1 to t_6 are discernible as connected clusters of (zero or more) black nodes separated by the large white places and one smaller white 'choice' node. Some of the arrows have been erased: this is an important philosophical distinction, related to the move from function to relations, where we step away from enforcing causality and instead consider *behaviours*.

Defining algebraic structures by means of a *presentation* consisting of *generators* and *equations* is common in, say, group theory: $\langle x \mid x^n = 1\rangle$ and $\langle x, y \mid x^2 = y^2 = (xy)^2 = 1\rangle$ are well-known as the presentations of

[2]though in fact, they are rather more generally applicable

the cyclic group of order n and the Klein four-group respectively. The equivalent tool for string diagrams is called a *symmetric monoidal theory* (SMT), and fits the graphical approach particularly well.

For an example, we will introduce a SMT which will be a sub-theory running throughout the thesis, and which will hopefully draw on some familiar concepts. It is called the theory of commutative monoids, and the prop it generates is isomorphic to the prop of functions between finite sets. We will sketch out the nature of the isomorphism, but leave full details to Chapter 2. This theory has two generators, written as (the multiplication) and (the unit). Each generator has associated a *type*, the domain and codomain of the morphism it will stand for. Types are always clear from the number of dangling wires on the left and right side of the diagram: has type $2 \to 1$ and type $0 \to 1$. The three equations encode the three laws of communtative monoids: the unit law, associativity, and commutativity.



The correspondence to the prop of functions is harder to describe in words than to understand visually. Numbering the dangling wires on each side starting from 0 at the top, the diagram corresponding to the function $f : m \to n$ connects the xth wire on the left to the yth on the right if and only if $f(x) = y$. It's intuitively obvious that the shapes of our generators ensure that every member of the domain has a unique image in the codomain, and that none of the axioms change the meaning of a diagram. With a proof of this fact, this gives an isomorphism between the prop of diagrams and the prop of functions. We use exactly the same procedure in all the other examples: give a collection of generators and equations, and then show an isomorphism between the generated theory and some prop representing semantics.

The centrepiece of the collection of SMTs that we investigate is $\mathbb{IH}_{\mathsf{R}}$, the prop of *interacting Hopf algebras*. It contains the white monoid structure above, interpreted as addition in some principal ideal domain R, along with a parallel black structure to copy ring elements. The generated prop is isomorphic to the *linear relations* over k, the field of fractions of R: a morphism $m \to n$ is a linear subspace of $\mathsf{k}^m \times \mathsf{k}^n$. $\mathbb{IH}_{\mathsf{R}}$ gives us a graphical notation for all of linear algebra, and thus a language to talk about any linear system we wish to analyse.

But, like any prop of relations, the linear relations have an intrinsic notion of containment which the setting of props and SMTs is unable to handle. To resolve this, we must upgrade our props to *ordered props*, and to do that we must introduce the last piece of background theory: the bicategory. We define what an ordered prop is, and then handle the other half of the extension to orderings, by augmenting the definition of SMT to give the main contribution: the symmetric monoidal *inequality* theory, or SMIT.

A SMIT, just like a SMT, has some generators, along with equations relating them, but it also contains some *inequalities*, to enforce further relations between the diagrams, allowing us to capture the inclusions which were previously unaccounted for. For example, we will define a number of theories depending on a field k, and define generators ○— representing the trivial linear subspace $\{0\}$ and ●— the whole 1-dimensional linear space k. Then, our SMITs will contain the self-evident inequality ○— $\leq$ ●—, i.e. $\{0\} \subseteq \mathsf{k}$.

The first task, after defining the process to get an ordered prop from a SMIT, is to verify that the examples of Chapter 2 may be translated into the new world (partial) order while maintaining an appropriate notion of isomorphism with the semantic categories.

Having thereby set up the theoretical background, we move on to our first application: the signal flow graphs (SFGs) of Shannon and Mason, ubiquitous in the world of signal processing and understanding feedback. We build on previous results interpreting SFGs as string diagrams, explaining what a signal flow graph is and giving a variety of different semantics to them: a denotational semantics, and two different operational semantics, corresponding to initialising the registers with either zero or non-zero values. We extend the existing SMTs to SMITs, and verify that the isomorphisms and theory are all preserved, in both the zero and non-zero cases.

An important new technique enabled by the use of SMITs and ordered props stems from the fact that the collection of diagrams representing SFGs is not all possible diagrams—there are diagrams which do not correspond to any 'real' SFG, but which nonetheless *do* correspond to a linear relation, which we consider as a collection of permissible behaviours, i.e. a specification. We can then prove diagrammatically that some diagram which *does* correspond to a SFG is a valid implementation of the specification.

The final chapter parallels the previous one, applying the theory of SMITs to a model of concurrency, this time considering Petri nets. This requires some more theoretical machinery: because the number of tokens in a Petri net place cannot be negative, we must move away from linear theories and switch to *additive relations*. This amounts to a surprisingly small amount of change to our theories, and we can then show results to before: that props representing Petri nets as string diagrams are isomorphic to props representing their behaviours, and that we can use diagrammatic reasoning to demonstrate equality and inclusion of behaviours.

Chapter 2

Background and previous work

This chapter reviews previous work and theoretical background which is the foundation for the remainder of the thesis. We begin by covering basic notions of general category theory, such as functors, epi-, mono- and isomorphisms, products and coproducts, and full and faithful functors. We also give details of props, the particular class of categories that we focus on. We provide numerous examples of props, many of which will be of use as we further develop our theory in later chapters. Most importantly, string diagrams and symmetric monoidal theories (SMTs) are introduced. These are the main notation we use for props and the chief means of specification, but they are more significant than mere notation and nuts-and-bolts infrastructure. They are a way of thinking that inform all the results and proofs in this thesis. After many examples of and results about SMTs, the chapter concludes by giving the definitions of bicategories and in particular locally posetal bicategories, which are the framework for axiomatising the notions of inclusion and refinement.

2.1 Categories

We work in the language of category theory, dating back to Eilenberg and Mac Lane [EML45]. Categories can be used as a language for processes with a *domain* and *codomain*, along with *composition*. Classically, categories were thought of as abstracting algebraic structures with structure-preserving mappings; so-called *concrete categories* like **Set**, **Grp** and **Top** were among the first to be studied. However, the definition of a category has proved to be extremely versatile, and many other examples have been investigated, including purely abstract finite categories and the relational theories underpinning this work.

In this paradigm, we distinguish two primitive kinds: *objects*, which we will use to keep track of inputs and outputs, typically relegated to the background, and *morphisms* which are the main focus. The first piece of evidence for the primacy of morphisms is that the key operation of *composition* combines two morphisms to produce another one. In summary, a category has the following definition:

Definition 2.1. A *category* is a collection of *objects* (typically written $X, Y, Z, \ldots$) and a collection of *morphisms* $(f, g, h, \ldots)$ such that

(i) every morphism is associated with a (unique) pair of objects called its *domain* and *codomain*. If a morphism f has domain X and codomain Y, we succinctly write this as $f : X \to Y$ and speak of f as being 'from X to Y'

(ii) for all objects X, Y, Z and every pair of morphisms $f : X \to Y$ and $g : Y \to Z$ there is a unique morphism $f ; g : X \to Z$, their *composite*[1]

[1] Some write composition from right to left: $g \circ f$, or even gf. We will only ever write composition from left to right as $f; g$, and

(iii) if $f : W \to X, g : X \to Y$ and $h : Y \to Z$ then $(f;g);h = f;(g;h)$ (composition is associative)

(iv) for every object X there is an *identity* morphism $\mathrm{id}_X : X \to X$ such that, if $f : X \to Y$ and $g : W \to X$, then $\mathrm{id}_X; f = f$ and $g; \mathrm{id}_X = g$

In a category C, the class of all morphisms with domain X and codomain Y is called an *hom-set*[2] and is written $C(X,Y)$.

For arbitrary categories, we write C, D. The prototypical category is **Set**, where the objects are sets and the morphisms are functions between them. Composition and identities are the normal function composition and identity functions. Similarly, restricting objects to groups and morphisms to group homomorphisms gives the category **Grp**, and taking for objects the topological spaces and for morphisms continuous maps gives the category **Top**. We could try to form a category **Cat** whose objects are categories[3], but then what should morphisms be? In **Grp** and **Top**, the morphisms are mappings which preserve the relevant structure. It is the same in **Cat**; such morphisms are known as *functors*.

Definition 2.2. Given categories C and D, a *functor* $F : C \to D$ is a mapping of objects of C to those of D together with a mapping $C(X,Y) \to D(FX,FY)$ for each pair of objects X, Y such that, for all objects X, Y, Z of C and all morphisms $f : X \to Y, g : Y \to Z$,

(i) $F(\mathrm{id}_X) = \mathrm{id}_{FX}$

(ii) $F(f;g) = Ff;Fg$

The identity functor acts by the identity mapping on both objects and morphisms, and functors are composed by composing the comprised maps.

One important way to construct new categories from old is simply by formally 'turning around' all the morphisms. Despite its seeming triviality, we will frequently use it in our development.

Definition 2.3. If C is a category, then the *opposite category* C^{op} has the same objects and morphisms as C, but the directions are reversed. If $f : X \to Y$ in C, then $f : Y \to X$ in C^{op} (we will sometimes write this corresponding morphism as f^{op} to avoid confusion), and if $f : X \to Y$ and $g : Y \to Z$ are morphisms in C^{op} (that is, morphisms $f : Y \to X$ and $g : Z \to Y$ in C) then their composite in C^{op} is given by

$$f ;_{C^{\mathrm{op}}} g = g ;_C f$$

Many notions of category theory which we will use are inspired by set theory, simply rephrased to avoid talking about specific members of sets since in general morphisms in a category need not bear any resemblance to functions. Three important examples of this are the equivalents of injective, surjective and bijective functions.

Definition 2.4. A morphism $f : X \to Y$ in some category is

(i) *monic* (or a *monomorphism*) iff for all morphisms $g, g' : W \to X$

$$g;f = g';f \implies g = g'$$

(ii) *epic* (or an *epimorphism*) iff for all morphisms $h, h' : Y \to Z$

$$f;h = f;h' \implies h = h'$$

never drop the semicolon.

[2]We will only consider *locally small* categories where all hom-sets are indeed sets.

[3]with some restriction on size, if one is concerned about well-foundedness, though for our purposes we may ignore such issues.

(iii) an *isomorphism* (or *invertible*) iff it has a two-sided *inverse*, some $f^{-1} : Y \to X$ such that $f; f^{-1} = \mathrm{id}_X$ and $f^{-1}; f = \mathrm{id}_Y$.

We say that two objects are *isomorphic* if there is an isomorphism between them.

It's an easy lemma to justify the notation f^{-1} by showing that inverses are unique.

Lemma 2.5. *If g and h are both two-sided inverses of f, then $g = h$.*

Proof.
$$g = g; \mathrm{id}_X = g; (f; h) = (g; f); h = \mathrm{id}_Y; h = h$$
□

Note that in general, unlike in **Set**, being both monic and epic does not guarantee being an isomorphism. Another important idea in category theory inspired by a set-theoretic construction is the *categorical product*, generalising the Cartesian product, and its cousin the coproduct, which corresponds to taking disjoint unions.

To make these definitions, we use an ubiquitous tool of category theory, the *commutative diagram*. A commutative diagram shows some objects and morphisms in a category and asserts that any two composites formed by following paths through the diagram which begin and end at the same place are equal[4]. Commutative diagrams may be 'pasted' together along shared borders, as we will see in the next lemma.

Definition 2.6. Given two objects X, Y in a category, their *(binary) product* consists of an object $X \times Y$ and morphisms $\pi_X : X \times Y \to X$ and $\pi_Y : X \times Y \to Y$ such that for any morphisms $f : Z \to X, g : Z \to Y$ there is a unique $\langle f, g\rangle : Z \to X \times Y$ with

$$\begin{tikzcd}
& Z \arrow[dl, "f"'] \arrow[d, dashed, "\langle f, g\rangle"'] \arrow[dr, "g"] & \\
X & X \times Y \arrow[l, "\pi_X"] \arrow[r, "\pi_Y"'] & Y
\end{tikzcd}$$

Their *(binary) coproduct* on the other hand consists of an object $X + Y$ with morphisms $\iota_X : X \to X + Y$ and $\iota_Y : Y \to X + Y$ such that for any morphisms $f : X \to Z$ and $g : Y \to Z$ there is a unique $[f, g] : X + Y \to Z$ such that

$$\begin{tikzcd}
& Z & \\
X \arrow[ur, "f"] \arrow[r, "\iota_X"] & X + Y \arrow[u, dashed, "{[f, g]}"] & Y \arrow[ul, "g"'] \arrow[l, "\iota_Y"']
\end{tikzcd}$$

Note that products and coproducts need not exist. Similarly to our definitions of identities and inverses, our language and notation suggest that the product and coproduct of a pair of objects (if they exist at all) are unique, despite no such assertion being part of the definition. In fact they are typically *not* unique, but, as is common throughout category theory, they are unique up to a unique isomorphism, and this is almost always close enough. As an example, we give the proof that products are essentially unique.

[4]Some authors will state this condition separately each time as 'the diagram commutes', but we will simply never draw a non-commutative diagram.

Lemma 2.7. *If (P, π_X, π_Y) and (P', π'_X, π'_Y) are both products of some objects X, Y, then there is a unique isomorphism $\varphi : P \to P'$ with*

$$\begin{tikzcd}
& P \arrow[dl, "\pi_X"'] \arrow[dr, "\pi_Y"] \arrow[dd, "\varphi"] & \\
X & & Y \\
& P' \arrow[ul, "\pi'_X"] \arrow[ur, "\pi'_y"'] &
\end{tikzcd}$$

Proof. Since (P, π_X, π_Y) and (P', π'_X, π'_Y) are both products, there are unique morphisms $\langle \pi_x . \pi_Y \rangle'$ and $\langle \pi'_X, \pi'_Y \rangle$ such that

$$\begin{tikzcd}
& P \arrow[dl, "\pi_X"'] \arrow[dr, "\pi_Y"] \arrow[dd, dashed, "\langle \pi_X, \pi_Y \rangle'"] & \\
X & & Y \\
& P' \arrow[ul, "\pi'_X"] \arrow[ur, "\pi'_Y"'] &
\end{tikzcd}$$

and

$$\begin{tikzcd}
& P' \arrow[dl, "\pi'_X"'] \arrow[dr, "\pi'_Y"] \arrow[dd, dashed, "\langle \pi'_X, \pi'_Y \rangle"] & \\
X & & Y \\
& P \arrow[ul, "\pi_X"] \arrow[ur, "\pi_Y"'] &
\end{tikzcd}$$

Pasting these together we see that

$$\begin{tikzcd}
& P \arrow[dl, "\pi_X"'] \arrow[dr, "\pi_Y"] \arrow[d, dashed, "\langle \pi_X, \pi_Y \rangle'"] & \\
X & P' \arrow[l, "\pi'_X"'] \arrow[r, "\pi'_Y"] \arrow[d, dashed, "\langle \pi'_X, \pi'_Y \rangle"] & Y \\
& P \arrow[ul, "\pi_X"] \arrow[ur, "\pi_Y"'] &
\end{tikzcd}$$

But of course id_P also fits in this diagram, so by uniqueness $\mathrm{id}_P = \langle \pi_X, \pi_Y \rangle'; \langle \pi'_X, \pi'_Y \rangle$. Pasting the opposite way shows that $\mathrm{id}_{P'} = \langle \pi'_X, \pi'_Y \rangle; \langle \pi_X, \pi_Y \rangle'$, and so these are inverse to each other. □

Since **Cat** is itself a category, the definitions for monic and epic could be applied to functors. In practice, weaker notions are easier to prove and are often good enough to prove useful theorems.

Definition 2.8. A functor $F : \mathcal{C} \to \mathcal{D}$ is *faithful* iff for every pair of objects X, Y in $\mathcal{C}$, its action on the hom-set $\mathcal{C}(X, Y) \to \mathcal{D}(FX, FY)$ is injective. It is *full* if it is always surjective on hom-sets. If F is both full and faithful, we say that it is *fully faithful*.

The first example of fullness and faithfulness being enough to show a fairly strong result is the next lemma. We will use it constantly later on, when we focus on functors which are not just bijections on objects, but actually identities.

Lemma 2.9. *If $F : \mathcal{C} \to \mathcal{D}$ is a fully faithful functor which is bijective on objects, then F is an isomorphism of categories.*

Proof. We need to construct an inverse functor $F^{-1} : \mathcal{D} \to \mathcal{C}$. Its action on objects is obvious by hypothesis.

Take an arbitrary morphism $f : FX \to FY$ in $\mathcal{D}$. Since F is fully faithful, it is injective and surjective, hence a bijection between the hom-sets $\mathcal{C}(X, Y)$ and $\mathcal{D}(FX, FY)$, and therefore there is a unique $F^{-1}f : X \to Y$ such that $F(F^{-1}f) = f$. So by construction $F^{-1}; F = \mathrm{id}_{\mathcal{D}}$ (note the reversed order!), and we need only check $F^{-1}; F = \mathrm{id}_{\mathcal{C}}$. But this is immediate since everything is a bijection. □

Our final elementary concept for categories should also be a familiar concept generalised from set theory and abstract algebra: a subcategory.

Definition 2.10. A *subcategory* of a category $\mathcal{C}$ is a category whose objects and morphisms are subclasses of those of $\mathcal{C}$ and where composition and identities agree with those of $\mathcal{C}$. A *full subcategory* is a subcategory where the evident inclusion functor is full.

Categories have been so successful in so many different disciplines because their definition is very simple and easy to satisfy. However, from now on, we will add some more axioms and restrict our attention to a special kind of category called a prop [ML65]. In a prop, the objects are the natural numbers, and have no purpose other than keeping track of the 'type' of the morphisms. In addition to our fundamental categorical operation of composition, props are equipped with a *tensor product*, or *monoidal product*, which can be used to model processes running in parallel. Suppose $f : m \to n$ is a process with m inputs and n outputs, and $f' : m' \to n'$ is another, then running them in parallel is $f \oplus f' : m + m' \to n + n'$. Of course, we will require this $\oplus$ operation to be sufficiently well-behaved, and also have some 'plumbing' to rearrange inputs and outputs as necessary.

Definition 2.11 ([ML65])**.** A *prop* is a category whose objects are the natural numbers equipped with an operation $\oplus$ called the *tensor product* on pairs of morphisms and, for each m, n, a distinguished morphism (called the *symmetry*) $\sigma_{m,n} : m + n \to n + m$, such that, for all morphisms f, f', g, g', h and all naturals l, m, m', n, n',

(i) if $f : m \to n$ and $f' : m' \to n'$, then $f \oplus f' : m + m' \to n + n'$

(ii) $\mathrm{id}_m \oplus \mathrm{id}_n = \mathrm{id}_{m+n}$

(iii) whenever the composites are defined, $(f; g) \oplus (f'; g') = (f \oplus f'); (g \oplus g')$

(iv) $(f \oplus g) \oplus h = f \oplus (g \oplus h)$

(v) $f \oplus \mathrm{id}_0 = \mathrm{id}_0 \oplus f = f$

(vi) if $f : m \to n$ then $(f \oplus \mathrm{id}_l); \sigma_{n,l} = \sigma_{m,l}; (\mathrm{id}_l \oplus f)$

(vii) $\sigma_{m,n}; \sigma_{n,m} = \mathrm{id}_{m+n}$

Remark 2.12. These axioms are those for a strict symmetric monoidal category where the monoidal product on objects is addition. (i)–(iii) encode functoriality of $\oplus$, (iv) and (v) record its (strict) associativity and unit laws, and (vi) and (vii) enforce naturality and invertibility of $\sigma_{m,n}$.

Mac Lane's cited definition is phrased differently but equivalently. The requirement that every permutation is included follows from the standard result of group theory that every permutation is a composition of adjacent transpositions, which correspond to the morphisms $\mathrm{id}_k \oplus \sigma_{1,1} \oplus \mathrm{id}_l$ for some k, l.

One of the simplest examples of props that we will be working with is the prop version of the classical category **Set**: the finite sets FinSet. We will also work with the finite relations FinRel, which is analogous to **Rel**, and the equivalence relations EqRel. When defining a prop, we do not need to specify the objects, since they are always the natural numbers.

Example 2.13. In the prop of *finite sets* FinSet, we associate each natural number n with the finite ordinal $\overline{n} = \{0, 1, \dots, n-1\}$. The morphisms $m \to n$ in this prop are the functions $\overline{m} \to \overline{n}$. Composition and identities are as usual for functions, and the monoidal product of $f : m \to n$ and $g : m' \to n'$ is $f \oplus g : m + m' \to n + n'$ given by

$$(f \oplus g)(p) = \begin{cases} f(p) & \text{if } 0 \leq p < m \\ g(p-m) & \text{if } m \leq p < m + m' \end{cases}$$

By identifying a function with its graph, FinSet embeds into the prop of *finite relations* FinRel, where a morphism $m \to n$ is a subset of $\overline{m} \times \overline{n}$, composition of $R \subseteq \overline{l} \times \overline{m}$ and $S \subseteq \overline{m} \times \overline{n}$ is the standard relational notion:

$$R; S = \{(p, q) \in \overline{l} \times \overline{n} \mid \text{there is } q \in \overline{m} \text{ such that } (p, r) \in R \text{ and } (r, q) \in S\}$$

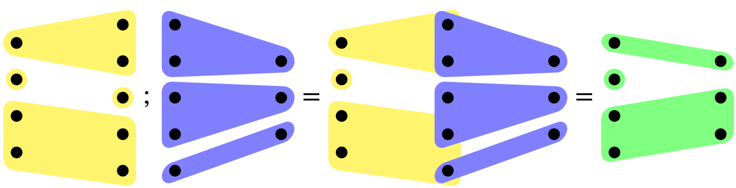
and the monoidal product of $R \subseteq \overline{m} \times \overline{n}$ and $R' \subseteq \overline{m'} \times \overline{n'}$ is the obvious relation putting R and R' 'next to each other' on $\overline{m + m'} \times \overline{n + n'}$:

$$(p, q) \in R \oplus R' \text{ iff } \begin{cases} 0 \leq p < m,\ 0 \leq q < n \text{ and } (p, q) \in R, \text{ or} \\ m \leq p < m + m',\ n \leq q < n + n' \text{ and } (p - m, q - n) \in R' \end{cases}$$

In the prop of *equivalence relations* EqRel, a morphism $m \to n$ is an equivalence relation (equivalently a partition) on the disjoint union $\overline{m} + \overline{n}$. The composite of $\alpha : l \to m$ and $\beta : m \to n$ is the restriction to $\overline{l} + \overline{n}$ of the finest partition of $\overline{l} + \overline{m} + \overline{n}$ which is coarser than α and β when restricted to $\overline{l} + \overline{m}$ and $\overline{m} + \overline{n}$ respectively. The identity at n is $\{(p, p) \mid p \in \overline{n}\}$, and the tensor product $\alpha \oplus \alpha' : m + m' \to n + n'$ is a similar placement 'side-by-side':

$$p \text{ is } \alpha \oplus \alpha'\text{-related to } q \text{ iff } \begin{cases} p, q \in \overline{m} + \overline{n} \text{ and } p \text{ is } \alpha\text{-related to } q, \text{ or} \\ p - m, q - n \in \overline{m'} + \overline{n'} \text{ and } p - m \text{ is } \alpha'\text{-related to } q - n \end{cases}$$

While the above definition of composition in EqRel seems complicated, it is in fact an intuitive operation of merging equivalence classes along the common interface, best understood by drawing diagrams. Below we illustrate the composition of an equivalence relation $4 \to 5$ with another $5 \to 3$ to obtain a composite $4 \to 3$. Coloured areas indicate equivalence classes.



We can start from an arbitrary set and build a prop of relations over that, rather than just the finite ones.

Example 2.14. For any set X, Rel_X is the prop where the morphisms $m \to n$ are the subsets of $X^m \times X^n$. Composition is the relational one defined as in FinRel, and the monoidal product of $R \subseteq X^m \times X^n$ and $R' \subseteq X^{m'} \times X^{n'}$ is

$$R \oplus S = \left\{ \left(\begin{pmatrix} \mathbf{x} \\ \mathbf{x}' \end{pmatrix}, \begin{pmatrix} \mathbf{y} \\ \mathbf{y}' \end{pmatrix} \right) \mid (\mathbf{x}, \mathbf{y}) \in R \text{ and } (\mathbf{x}', \mathbf{y}') \in R' \right\}$$

Two more examples of props that we will return to frequently are the *matrices over a ring* and the *linear relations over a field*. They are sub-props of some Rel_X, and are related to each other in a similar way to FinSet and FinRel.

Example 2.15. If R is a ring[5], then the $m \to n$ morphisms of the prop of matrices over R, denoted Mat_R, are the $n \times m$ matrices with entries in R. If $A : l \to m$ and $B : m \to n$, then the composite $A; B$ is the matrix product BA. The identities are the usual identity matrices, and the tensor product of A and B is the block matrix $\begin{pmatrix} A & 0 \\ 0 & B \end{pmatrix}$. $\sigma_{m,n}$ is defined by the block matrix $\begin{pmatrix} 0 & I_m \\ I_n & 0 \end{pmatrix}$ where as usual I_n is the $n \times n$ identity matrix.

If k is a field, then $\mathsf{LinRel}_\mathsf{k}$, the prop of linear relations over k, has as its morphisms $m \to n$ all linear subspaces of $\mathsf{k}^m \times \mathsf{k}^n$ (here $\times$ is direct product of vector spaces). The composite of $R \subseteq \mathsf{k}^l \times \mathsf{k}^m$ and $S \subseteq \mathsf{k}^m \times \mathsf{k}^n$ is the relation

$$R; S = \{(\mathbf{u}, \mathbf{w}) \in \mathsf{k}^l \times \mathsf{k}^n \mid \exists \mathbf{v} \in \mathsf{k}^m \text{ such that } (\mathbf{u}, \mathbf{v}) \in \mathsf{k}^l \times \mathsf{k}^m \text{ and } (\mathbf{v}, \mathbf{w}) \in \mathsf{k}^m \times \mathsf{k}^n\}$$

and the identity at n is $\{(\mathbf{v}, \mathbf{v}) \in \mathsf{k}^n \times \mathsf{k}^n \mid \mathbf{v} \in \mathsf{k}^n\}$; these are the usual definitions of composition and identity in a category of relations. The tensor product of $R \subseteq \mathsf{k}^m \times \mathsf{k}^n$ and $R' \subseteq \mathsf{k}^{m'} \times \mathsf{k}^{n'}$ is

$$R \oplus S = \left\{ \left(\begin{pmatrix} \mathbf{u} \\ \mathbf{u}' \end{pmatrix}, \begin{pmatrix} \mathbf{v} \\ \mathbf{v}' \end{pmatrix} \right) \mid (\mathbf{u}, \mathbf{v}) \in R \text{ and } (\mathbf{u}', \mathbf{v}') \in R' \right\}$$

and finally the symmetry is

$$\sigma_{m,n} = \left\{ \left(\begin{pmatrix} \mathbf{u} \\ \mathbf{v} \end{pmatrix}, \begin{pmatrix} \mathbf{v} \\ \mathbf{u} \end{pmatrix} \right) \mid \mathbf{u} \in \mathsf{k}^m, \mathbf{v} \in \mathsf{k}^n \right\}$$

When R is a subring of k, Mat_R embeds into $\mathsf{LinRel}_\mathsf{k}$ by sending an $n \times m$ matrix A to its graph $\{(\mathbf{v}, A\mathbf{v}) \mid \mathbf{v} \in \mathsf{k}^m\}$. The embeddings $\mathsf{Mat}_\mathsf{R} \hookrightarrow \mathsf{LinRel}_\mathsf{k}$ and $\mathsf{FinSet} \hookrightarrow \mathsf{FinRel}$ are not only functors, but satisfy the stronger requirements of being *prop morphisms*.

Definition 2.16. A *prop morphism* between props $\mathbb{S}$ and $\mathbb{T}$ is a functor $F : \mathbb{S} \to \mathbb{T}$ which is the identity on objects such that[6]

(i) for all morphisms f, g of $\mathbb{S}$, $Ff \oplus Fg = F(f \oplus g)$

(ii) for all natural numbers m, n, $F(\sigma^{\mathbb{S}}_{m,n}) = \sigma^{\mathbb{T}}_{m,n}$

We can easily show that composing prop morphisms (as functors) gives a prop morphism, the identity functor on a prop is a prop morphism, and if a functor is the inverse of a prop morphism, then it is a prop morphism itself. This means that the category of props **PROP** is a subcategory of **Cat** whose isomorphisms are exactly the prop morphisms which are invertible functors. Every prop morphism is by definition bijective on objects, so Lemma 2.9 always applies.

Lemma 2.17. *Let $\mathbb{S}, \mathbb{T}, \mathbb{U}$ be props, and $F : \mathbb{S} \to \mathbb{T}$, $G : \mathbb{T} \to \mathbb{U}$ prop morphisms. Then*

[5] We assume all rings are commutative and unital.

[6] In the language of monoidal categories, a prop morphism is a strict symmetric functor which is the identity on objects.

(i) $F; G$ (composing as functors) is a prop morphism

(ii) the identity functor is a prop morphism

(iii) if F has an inverse functor F^{-1} then F^{-1} is a prop morphism

Proof. (i) Composing two identities is still an identity, so $F; G$ is identity on objects. If f, g are morphisms of $\mathbb{S}$, then

$$(F; G)(f \oplus g) = G(F(f \oplus g)) = G(Ff \oplus Fg) = G(Ff) \oplus G(Fg) = (F; G)f \oplus (F; G)g$$

and similarly

$$F; G(\sigma^{\mathbb{S}}_{m,n}) = G(F(\sigma^{\mathbb{S}}_{m,n})) = G(\sigma^{\mathbb{T}}_{m,n}) = \sigma^{\mathbb{U}}_{m,n}$$

(ii) This is immediate.

(iii) Let f, g be morphisms in $\mathbb{T}$. Then there are f', g' morphisms of $\mathbb{S}$ with $f = Ff'$ and $g = Fg'$. So

$$F^{-1}(f \oplus g) = F^{-1}(Ff' \oplus Fg') = F^{-1}(F(f' \oplus g')) = f' \oplus g' = Ff \oplus Fg$$

and

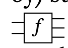
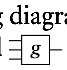
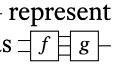
$$F^{-1}(\sigma^{\mathbb{T}}_{m,n}) = F^{-1}(F(\sigma^{\mathbb{S}}_{m,n})) = \sigma^{\mathbb{S}}_{m,n}$$

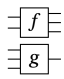
□

2.2 String diagrams and symmetric monoidal theories

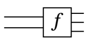
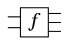
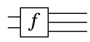
If we are to use props to model processes, we will need a way to take a basic set of primitives and produce the category containing all possible combinations. When we do this, we will also often need to impose equations representing relations between our primitives. For example, we might have a process which simply duplicates its input, and another process which discards it. We might then want to require that copying an input and then discarding one of the copies is the same as doing nothing at all. The tool which allows us to do both of these is the *symmetric monoidal theory* or *SMT*.

In any category, the objects are of secondary importance compared to the morphisms. A prop takes this to the extreme, and accordingly the morphisms take the spotlight even more. In parallel with SMTs, we will introduce *string diagrams* [JS91, Sel11], the graphical notation for morphisms of props that we will almost exclusively use in the remainder of this thesis.

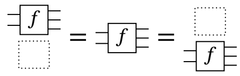
The domain and codomain of a morphism are encoded in a string diagram by 'dangling wires' on the left and right respectively. One can therefore immediately see when composition of (morphisms represented by) string diagrams is possible: when they have a 'compatible interface'. Taking this metaphor further, if  and  represent some $f : 2 \to 3$ and $g : 3 \to 1$ respectively, then the composite $f; g : 2 \to 1$ may be drawn as  simply connecting corresponding wires. If instead we wished to depict the tensor product $f \oplus g : 2 + 3 \to 3 + 1$ then we would stack the diagrams above each other, like so:



Note that when composing or tensoring multiple morphisms, there is no way to distinguish $(f; g); h$ and $f; (g; h)$ or $(f \oplus g) \oplus h$ and $f \oplus (g \oplus h)$. In other words, they implicitly encode the associativity of the ; and $\oplus$ operations. If, in accordance with prop law (ii), we draw id_n as a stack of n wires (so that id_2 is), then the unit laws for ; and $\oplus$ at f become

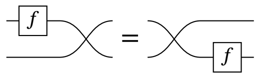
 =  = 

and

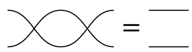


drawing an empty dotted box around the otherwise invisible stack of zero wires representing id_0. The second equation appears to be almost completely trivial, and the first could be summarised as saying that wires may stretch.

In equations involving string diagrams, we often use a shorthand to avoid having to draw an excessive number of wires, or needing ellipses when we talk about an arbitrary morphism. We simply draw a bunch of wires which start and end in the same place as if they are a single wire, sometimes labelling them by a number for additional clarity if necessary. For example, the last notation used for the string diagrams of every prop, representing the symmetry $\sigma_{m,n} : m + n \to n + m$. These are drawn as  and then the laws of Definition 2.11 (vi) and (vii) translate as



and



These also have pithy interpretations: 'morphisms slide over symmetries' and 'wires don't tangle'.

For our purposes, this simple intuitive definition of string diagrams will suffice. The more sceptical reader may find more formal definitions, as well as proofs of soundness and completeness of proofs by string diagrams, in the original paper by Joyal and Street [JS91] or in the survey by Selinger [Sel11].

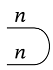
As an example of the expressive power of string diagrams, we will give the next definition using both traditional notation and string diagrams.

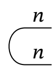
Definition 2.18 ([KL80]). (a) A prop is (self-dual) *compact closed* iff for every n there are two morphisms $\varepsilon_n : n + n \to 0$ and $\eta_n : 0 \to n + n$ such that for every m, n

(i) $(\eta_n \oplus \mathrm{id}_n); (\mathrm{id}_n \oplus \varepsilon_n) = \mathrm{id}_n = (\mathrm{id}_n \oplus \eta_n); (\varepsilon_n \oplus \mathrm{id}_n)$

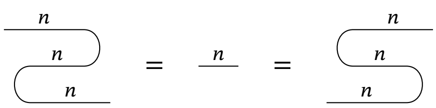
(ii) $(\mathrm{id}_m \oplus \sigma_{m,n} \oplus \mathrm{id}_n); (\varepsilon_m \oplus \varepsilon_n) = \varepsilon_{m+n}$

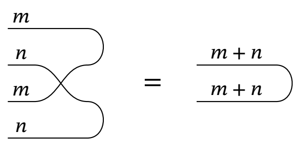
(iii) $(\eta_m \oplus \eta_n); (\mathrm{id}_m \oplus \sigma_{m,n} \oplus \mathrm{id}_n) = \eta_{m+n}$

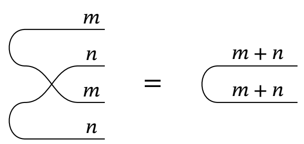
(b) A prop is (self-dual) *compact closed* iff for every n there are morphisms 

and 

such that for every m, n

(i)


(ii)


(iii)


It is a useful exercise to verify the translation between these two definitions. We can immediately see several advantages to definition (b) over definition (a). For a start, an intuitive sense of the intended meaning of the morphisms ε_n and η_n is clear just from the shape of the diagrams representing them. Also, it takes some time even to verify that the equations in part (a) are well-typed, while in (b) it's a simple at-a-glance matter of following the wires. Finally, finding proofs is often easier as parts of string diagrams can rapidly be recognised as instances of axioms.

So far we have thought of string diagrams as a notation for morphisms in some pre-existing prop, but we can go the other way and define props whose morphisms are themselves string diagrams. We specify some initial morphisms and equations, and then build the morphisms as all possible string diagrams freely generated by the *generators*, along with identities and symmetries, and then impose the laws by taking quotients. We use this process to find a *presentation* for some prop we wish to work in, by finding a prop isomorphism from our target prop to some prop arising in the way described. The data to generate a prop from string diagrams comprises a *symmetric monoidal theory* (SMT).

Definition 2.19. A *symmetric monoidal theory* (SMT) consists of a set Σ of *generators*, which are diagrams with dangling wires on the left and right, and a set E of *equations* (or laws), which are pairs of string diagrams freely generated as below with the same type.

A string diagram is freely generated from Σ if

(i) it is the empty diagram $\boxed{}$, the identity — or the symmetry $\times$

(ii) it is a member of Σ

(iii) it is of the form $({}^{m}\!-\boxed{A}-{}^{n} \oplus {}^{m'}\!-\boxed{B}-{}^{n'})$, henceforth depicted as $\begin{array}{c} {}^{m}-\boxed{A}-{}^{n} \\ {}^{m'}-\boxed{B}-{}^{n'} \end{array}$, where ${}^{m}\!-\boxed{A}-{}^{n}$ and ${}^{m'}\!-\boxed{B}-{}^{n'}$ freely generated from Σ

(iv) it is of the form $({}^{l}\!-\boxed{A}-{}^{m} ; {}^{m}\!-\boxed{B}-{}^{n})$, henceforth depicted as ${}^{l}\!-\boxed{A}-{}^{m}-\boxed{B}-{}^{n}$, where ${}^{l}\!-\boxed{A}-{}^{m}$ and ${}^{m}\!-\boxed{B}-{}^{n}$ freely generated from Σ

From E we define an equivalence relation $\sim_E$ on string diagrams generated from Σ inductively by the following (omitting labels on wires):

$$\begin{aligned} E_0 = E &\cup \{(-\boxed{A}-, -\boxed{A}-) \mid -\boxed{A}- \text{ is any string diagram over } \Sigma\} && (2.1)\\ &\cup \{(-\boxed{A}-, -\boxed{B}-) \mid -\boxed{A}- = -\boxed{B}- \text{ is an instance of a prop axiom}\} && (2.2) \end{aligned}$$

$$\begin{aligned} E_{n+1} = E_n &\cup \{(-\boxed{B}-, -\boxed{A}-) \mid (-\boxed{A}-, -\boxed{B}-) \in E_n\} && (2.3)\\ &\cup \{(-\boxed{A}-, -\boxed{C}-) \mid \exists -\boxed{B}- \text{ such that } (-\boxed{A}-, -\boxed{B}-), (-\boxed{B}-, -\boxed{C}-) \in E_n\} && (2.4)\\ &\cup \{(-\boxed{A}-\boxed{B}-, -\boxed{A'}-\boxed{B'}-) \mid (-\boxed{A}-, -\boxed{A'}-), (-\boxed{B}-, -\boxed{B'}-) \in E_n\} && (2.5)\\ &\cup \left\{\left(\begin{array}{c}-\boxed{A}-\\-\boxed{B}-\end{array}, \begin{array}{c}-\boxed{A'}-\\-\boxed{B'}-\end{array}\right) \mid (-\boxed{A}-, -\boxed{A'}-), (-\boxed{B}-, -\boxed{B}-) \in E_n\right\} && (2.6) \end{aligned}$$

and finally $\sim_E = \bigcup_{n=0}^{\infty} E_n$. Observe that if $-\boxed{A}- \sim_E -\boxed{B}-$ then they have the same domain and codomain.

Then the prop generated from (Σ, E), denoted $\mathbb{T}_{(\Sigma,E)}$, has as morphisms $m \to n$ string diagrams of that type freely generated from Σ modulo[7] the equivalence relation $\sim_E$. The composite of $-\boxed{A}-$ and

[7] As usual, we will typically abuse notation to write a representative of an equivalence class instead of the class itself.

$-\boxed{B}-$ is $-\boxed{A}\!-\!\boxed{B}-$, their tensor product is $\begin{smallmatrix}-\boxed{A}-\\-\boxed{B}-\end{smallmatrix}$ and the identities are $\overset{n}{\text{---}}$. These are all well-defined by the conditions we put on $\sim_E$.

We say that a SMT (Σ, E) is a *presentation* for a prop $\mathbb{T}$ iff $\mathbb{T}_{(\Sigma,E)}$ is isomorphic (as a prop) to $\mathbb{T}$.

Remark 2.20. In principle, one should be careful about associativity and stretching of wires. For example, $(-\boxed{}-;-) = -\boxed{}---$ looks very similar to simply $-\boxed{}-$, especially when imprecisely drawn by hand. Also, $\begin{smallmatrix}-\boxed{}\!-\!\boxed{}-\\-\boxed{}\!-\!\boxed{}-\end{smallmatrix}$ could be read as an instance of either (iii) or (iv), requiring dotted lines or brackets or some other way to disambiguate. In all such cases, however, a prop axiom included in E_0 via (2.2) ensures that, as morphisms (equivalence classes under E) they are equal; in the first case, the unit law for composition (Definition 2.1(iv)) and in the second interchange law for props (Definition 2.11(iii)).

In any case, throughout this thesis, we always deal with string diagrams as members of their equivalence classes at least up to the laws of props, and are justified in doing so by Joyal and Street [JS91].

Often we will be lazy with language and refer to the prop generated from a SMT simply as a SMT.

One reason that we focus so much on SMTs in this thesis is because they enable the powerful proof technique of structural induction. Closely related to structural induction is defining prop morphisms by recursion, similarly to how we defined the equivalence relation $\sim_E$ above. However, we need structural induction to show that recursive definitions are well-defined.

Proposition 2.21. *Suppose P is a property of morphisms in a prop $\mathbb{T}_{(\Sigma,E)}$ generated by a SMT (Σ, E). If*

(i) $P(\ )$, $P(-)$ and $P(\times)$ all hold

(ii) $P(-\boxed{\sigma}-)$ holds for every generator $-\boxed{\sigma}- \in \Sigma$

(iii) for all morphisms $-\boxed{A}-$ and $-\boxed{B}-$, $P(-\boxed{A}-)$ and $P(-\boxed{B}-)$ together imply $P(-\boxed{A}\!-\!\boxed{B}-)$

(iv) for all morphisms $-\boxed{A}-$ and $-\boxed{B}-$, $P(-\boxed{A}-)$ and $P(-\boxed{B}-)$ together imply $P\left(\begin{smallmatrix}-\boxed{A}-\\-\boxed{B}-\end{smallmatrix}\right)$

then P is true for every morphism of $\mathbb{T}_{(\Sigma,E)}$.

Proof. Define the *size* of a string diagram to be the total number of components in it, so the size of $\ $ is 0, $-$ or $\times$ have size 1, and the size of $-\boxed{A}\!-\!\boxed{B}-$ or $\begin{smallmatrix}-\boxed{A}-\\-\boxed{B}-\end{smallmatrix}$ is the sum of the sizes of $-\boxed{A}-$ and $-\boxed{B}-$. The size of a prop morphism (an equivalence class of string diagrams) is the minimal size of all string diagrams in the equivalence class. We will show that P is true for all morphisms by strong induction on the size of a morphism.

Let $-\boxed{A}-$ be a morphism of size n and consider a representative witnessing this size. Then there are three cases:

(i) the representative is one of $\ $, $-$, $\times$ or a member of Σ. Then by hypothesis (i) or (ii) $P(-\boxed{A}-)$ holds.

(ii) the representative is $-\boxed{B}\!-\!\boxed{C}-$ for string diagrams $-\boxed{B}-$ and $-\boxed{C}-$. Each of these must have size at most $n-1$ and so this is an upper bound on the sizes of the equivalence classes they represent. By the inductive hypothesis, $P(-\boxed{B}-)$ and $P(-\boxed{C}-)$. But then by assumption $P(-\boxed{B}\!-\!\boxed{C}-)$ holds, that is, $P(-\boxed{A}-)$ is true.

(iii) the representative is $\begin{smallmatrix}-\boxed{B}-\\ -\boxed{C}-\end{smallmatrix}$ for string diagrams $-\boxed{B}-$ and $-\boxed{C}-$. But similarly to case (ii) the sizes of $-\boxed{B}-$ and $-\boxed{C}-$ are strictly less, and the same combination of induction hypothesis and assumption gives us $P(-\boxed{A}-)$.

Therefore P holds for morphisms of every size, that is every morphism of $\mathbb{T}_{(\Sigma,E)}$. $\square$

When we want to show that a SMT generates a prop, we need to give an isomorphism between the prop generated by that SMT and the given prop. In general, if we wish to define a prop morphism out of a prop generated from a SMT, it is enough to specify where each generator goes, since the requirement that a prop morphism preserve composites, tensor products, identities and symmetries rigidly defines the image of all the non-generator morphisms.

That is, for any SMT (Σ, E) and any prop $\mathbb{T}$, if $F : \mathbb{T}_{(\Sigma,E)} \to \mathbb{T}$ is to be a prop morphism, we must have

$$F(\overset{n}{—}) = \mathrm{id}_n \qquad\qquad F(\times) = \sigma_{m,n} \tag{2.7}$$

for all m, n. Similarly, once the images of the generators are fixed, we have all the necessary data to define the action of F on any morphism of $T_{(\Sigma,E)}$. $F(-\boxed{A}-\boxed{B}-)$ can be defined recursively by composing in $\mathbb{T}$ as $F(-\boxed{A}-);F(-\boxed{B}-)$, and similarly $F\left(\begin{smallmatrix}-\boxed{A}-\\ -\boxed{B}-\end{smallmatrix}\right)$ as $F(-\boxed{A}-) \oplus F(-\boxed{B}-)$. Of course, since morphisms of $\mathbb{T}_{(\Sigma,E)}$ are really equivalence classes, we need to check that this gives a well-defined prop morphism; in fact, adherence to the laws of E is both necessary and sufficient.

Proposition 2.22. *Let (Σ, E) be a SMT and $\mathbb{T}$ any prop. Suppose F maps each generator $\overset{m}{—}\boxed{\sigma}\overset{n}{—}$ to an $m \to n$ morphism in $\mathbb{T}$. Then F extends uniquely to a (well-defined) prop morphism $\mathbb{T}_{(\Sigma,E)} \to \mathbb{T}$ iff for every pair $(-\boxed{A}-, -\boxed{B}-) \in E$, $F(-\boxed{A}-) = F(-\boxed{B}-)$ (using the above recursive definition for F).*

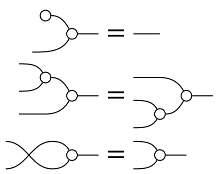
Proof. $\Longrightarrow$: This is immediate: if $-\boxed{A}- = -\boxed{B}-$ in $\mathbb{T}_{(\Sigma,E)}$ and F is well-defined as a prop morphism, then $F(-\boxed{A}-) = F(-\boxed{B}-)$.

$\Longleftarrow$: Uniqueness is automatic, as discussed in the preamble to the proposition, and F is explicitly defined to be a prop morphism. Therefore we only need to confirm that it is well-defined. We proceed by induction on n, with reference to the E_n from Definition 2.19, on the statement 'if $(-\boxed{A}-, -\boxed{B}-) \in E_n$ then $F(-\boxed{A}-) = F(-\boxed{B}-)$'.

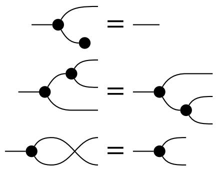
The base case follows from the assumption or from the fact that the prop laws hold in $\mathbb{T}$. For the inductive step, suppose that $(-\boxed{A}-, -\boxed{B}-) \in E_{n+1}$, for $n \geq 0$. Then there are five cases, corresponding to the five members of the union defining E_{n+1}. It is obvious that each case the induction hypothesis and the definition of F combine together just as needed to show the result. $\square$

We will now introduce a suite of SMTs which will come to underpin the developments of later chapters. They can be combined together as the building blocks of more complex theories.

Example 2.23. (i) (commutative monoids) Classically, a monoid is a set together with an associative binary operation which has a unit element. We can define a SMT (Σ_M, E_M) based on this structure. Σ_M contains representatives for the multiplication and unit $\{\rhd\!-, \circ\!-\}$ and E_M contains equations encoding the unit law, associativity and commutativity:



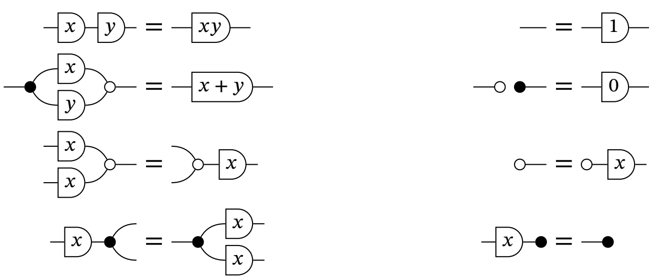
(ii) (commutative comonoids) There is no particular reason (other than convention) that we should write the monoid operations the way round we have in Σ_M. We could also flip them horizontally; to distinguish this from the previous monoid structure, we colour these in black rather than white. In line with decades of category theory tradition of prefixing 'co-' to the opposite version, this SMT is known as that of (commutative[8]) *comonoids*, consisting of comultiplication and counit $\Sigma_C = \{\ ,\ \}$ with equations E_C



(iii) (bimonoids) We said before that we intended to combine theories like building blocks, and now we have our first example. We simply take all the generators for monoids and comonoids, $\Sigma_B = \Sigma_M \cup \Sigma_C$, all the equations we have already seen, $E_M \cup E_C$, and add some compatibility between the generators with

(iv) (Hopf algebras over a PID R) In the above theory of bimonoids, each equation (including those inherited from the theories of monoids and bimonoids) preserves the number of paths from a given 'entrance' on the left to an 'exit' on the right. We can build on this intuition by writing $\boxed{1}$ for — and $\boxed{2}$ for , and more generally recursively defining $\boxed{n+1}$ as . With these definitions, addition can be defined analogously to the inductive step, and multiply by composition: this can be seen by counting paths from left to right.

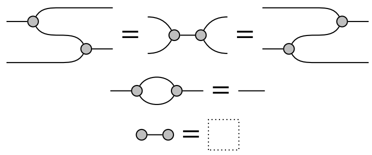
We can extend this analogy to any principal ideal domain by adding a generator for each element of the ring, and imposing equations so that the definitions in the prop agree with the ring operations. Thus $\Sigma_{\mathbb{HA}_\mathsf{R}} = \Sigma_B \cup \{\boxed{x} \mid x \in \mathsf{R}\}$ and $E_{\mathbb{HA}_\mathsf{R}}$ is E_B together with



(v) (Frobenius monoids) Bimonoids and Hopf algebras emphasise counting the number of connections between ports. If, on the other hand, we only wish to track a binary connected or not state, we can combine monoids and comonoids in a different way. To avoid confusion with $\mathbb{HA}_\mathsf{R}$, we will colour the generators in grey instead of black and white. $\Sigma_{\mathsf{Frob}} = \Sigma_M \cup \Sigma_C$ and E_{Frob} consists of the equations of

[8] sadly *not* 'mmutative'

$E_M \cup E_C$ plus

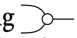
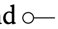


These theories generate some familiar props.

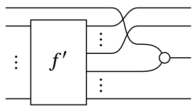
Proposition 2.24. *(i)* (Σ_M, E_M) *presents* FinSet*, the prop of functions between finite sets.*

(ii) $(\Sigma_{\mathbb{HA}_R}, E_{\mathbb{HA}_R})$ *presents* Mat_R*, the prop of matrices over the PID* R.

The proofs are prototypical for such presentation claims: we recursively define a prop morphism (via Proposition 2.22), then show that it is full and faithful. Then, by Lemma 2.9, we may conclude that it is an isomorphism. We give details for case (i), but only the definition of the isomorphism in part (ii); the full proof may be found in [Zan15].

Proof. (i) Define $\mathcal{S}_{\mathsf{FinSet}}$ recursively by sending  $: 2 \to 1$ and  $: 0 \to 1$ to the unique functions of those types in FinSet. For this to be well-defined, we need to check that it respects the laws of E_M. This is immediate however, as 1 is final in FinSet and each law is between morphisms with codomain 1. So $\mathcal{S}_{\mathsf{FinSet}}$ is a well-defined prop morphism, and we need to show that it is full and faithful. This is the same as being bijective on hom-sets, so it is enough to find an inverse function to the action of $\mathcal{S}_{\mathsf{FinSet}}$ on each hom-set and show that these inverses define a prop morphism.

For each n, we define this inverse assignment on hom-sets by recursion on m, the domain of the function. The base case is mapping the empty function $\overline{0} \to \overline{n}$ to the n-fold monoidal sum (○—)$^{\oplus n}$. If we have defined the mapping for all functions $\overline{m-1} \to \overline{n}$, and we wish to define the string diagram corresponding to some $f : \overline{m} \to \overline{n}$, first let $f' : \overline{m-1} \to \overline{n}$ be given by $f'(i) = f(i+1)$. Then define the diagram for f as

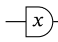


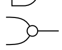
connecting the spare wire at the top to the $f(0)$th wire coming out of the diagram for f' (which exists by recursion).

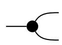
We need to check that it is functorial, that is, it preserves identities and compositions, and also that it respects tensor products and symmetries. We can do all of this simultaneously by demonstrating the intuitive property that, in the diagram corresponding to f, the ith port on the left is connected to the jth port on the right iff $j = f(i)$. This is a straightforward induction, reflecting the recursive definition. The base case of the empty function is immediate, and it is easy to see that the recursive step preserves this property, as do all the equations of E_M.

Then using this property, it is obvious that this proposed inverse is a prop morphism, and that it is inverse to $\mathcal{S}_{\mathsf{FinSet}}$.

(ii) $\mathcal{S}_{\mathbb{HA}_R}$ is defined by

 $\mapsto (x)$ $\mapsto !$ (2.8)

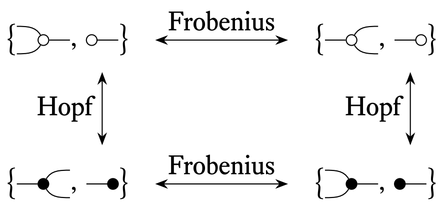
 $\mapsto \begin{pmatrix} 1 & 1 \end{pmatrix}$ $\mapsto ¡$ (2.9)

 $\mapsto \begin{pmatrix} 1 \\ 1 \end{pmatrix}$ (2.10)

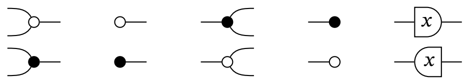
writing ! and ¡ for the unique 0×1 and 1×0 matrices.

□

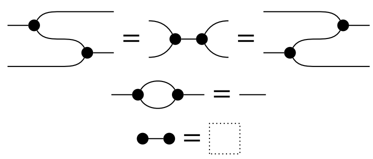
Note that the black comonoid structure of $\mathbb{HA}_{\mathsf{R}}$ ends up meaning 'copying' and the white monoid 'adding'. If one imagines members of R travelling along the wires, then each generator of $\mathbb{HA}_{\mathsf{R}}$ has an intuitive interpretiation in this way, which is compatible with the isomorphism $\mathcal{S}_{\mathbb{HA}_{\mathsf{R}}}$: inputting the components of a vector $\mathbf{v}$ onto the left of the string diagram representing a matrix A and composing the interpretations of the generators will eventually output the components of the vector $A\mathbf{v}$ on the right. ○— always outputs 0, —● discards its input, and $-\!\boxed{x}\!-$ multiplies its input by x. Following these rules one can always work from left to right in a string diagram of $\mathbb{HA}_{\mathsf{R}}$. If we give up this determinism, we can (perhaps surprisingly) arrive at another prop we introduced earlier: $\mathsf{LinRel}_{\mathsf{k}}$, where k is the field of fractions of R. We do this by combining $\mathbb{HA}_{\mathsf{R}}$ with its opposite, which we obtain by horizontally reflecting all its generators and equations, and then adding some compatibility conditions. Because we now have two Hopf algebras which are allowed to interact via these additional equations, the theory is known as the the prop of *interacting Hopf algebras* $\mathbb{IH}_{\mathsf{R}}$. The following schematic illustrates how the generators fit together:



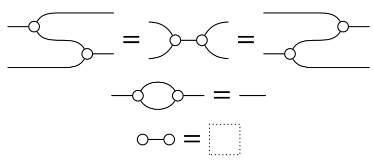
Definition 2.25. The prop $\mathbb{IH}_{\mathsf{R}}$ has as its generators, for all $x \in \mathsf{R}$,



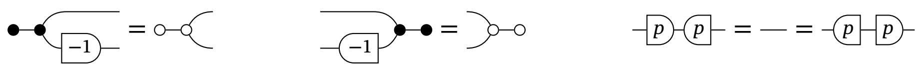
and as equations those of $E_{\mathbb{HA}_{\mathsf{R}}}$, their mirror images, the following laws for (black and white) Frobenius monoids:



and

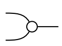
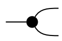


There are also three equations, two of which govern the interaction between the black and white structure, and the third provides multiplicative inverses:

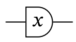


where p ranges over every non-zero member of R.

As for $\mathbb{HA}_{\mathsf{R}}$, we leave the details of the isomorphism $\mathcal{S}_{\mathbb{IH}_{\mathsf{R}}} : \mathbb{IH}_{\mathsf{R}} \to \mathsf{LinRel}_{\mathsf{k}}$ to Zanasi [Zan15], but we will give its definition (on generators):

 $\mapsto \left\{\left(\begin{pmatrix} x \\ y \end{pmatrix}, x+y\right) \mid x, y \in \mathsf{k}\right\}$  $\mapsto \{(x, (x \quad x)) \mid x \in \mathsf{k}\}$

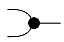
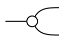
 $\mapsto \{(*, 0)\}$ $\mapsto \{(x, *) \mid x \in \mathsf{k}\}$

 $\mapsto \{(y, xy) \mid y \in \mathsf{k}\}$

Note that this definition is compatible with $\mathcal{S}_{\mathbb{HA}_{\mathsf{R}}}$ (and the self-evident $\mathcal{S}_{\mathbb{HA}^{\mathrm{op}}_{\mathsf{R}}} : \mathbb{HA}^{\mathrm{op}}_{\mathsf{R}} \to \mathsf{Mat}^{\mathrm{op}}_{\mathsf{R}}$). To explain what we mean by 'compatible', first observe that all the generators and equations for $\mathbb{HA}_{\mathsf{R}}$ and $\mathbb{HA}^{\mathrm{op}}_{\mathsf{R}}$ are in $\mathbb{IH}_{\mathsf{R}}$. This means that there is an obvious pair of embeddings $\mathbb{HA}_{\mathsf{R}} \hookrightarrow \mathbb{IH}_{\mathsf{R}} \hookleftarrow \mathbb{HA}^{\mathrm{op}}_{\mathsf{R}}$. Similarly $\mathsf{Mat}_{\mathsf{R}}$ and $\mathsf{Mat}_{\mathsf{R}}$ may be embedded in $\mathsf{LinRel}_{\mathsf{k}}$, by sending the $n \times m$ matrix A to its graph $\{(\mathbf{v}, A\mathbf{v}) \mid \mathbf{v} \in \mathsf{k}^m\}$ for $\mathsf{Mat}_{\mathsf{R}}$ and to the converse $\{(A\mathbf{v}, \mathbf{v}) \mid \mathbf{v} \in \mathsf{k}^m\}$ for $\mathsf{Mat}^{\mathrm{op}}_{\mathsf{R}}$. Then 'compatibiity' means that the following commutes in **PROP**:

$$\begin{tikzcd}
\mathbb{HA}_{\mathsf{R}} \arrow[r, hook] \arrow[d, dashed, "\mathcal{S}_{\mathbb{HA}_{\mathsf{R}}}"] & \mathbb{IH}_{\mathsf{R}} \arrow[d, dashed, "\mathcal{S}_{\mathbb{IH}_{\mathsf{R}}}"] & \mathbb{HA}^{\mathrm{op}}_{\mathsf{R}} \arrow[l, hook'] \arrow[d, dashed, "\mathcal{S}_{\mathbb{HA}^{\mathrm{op}}_{\mathsf{R}}}"] \\
\mathsf{Mat}_{\mathsf{R}} \arrow[r, hook] & \mathsf{LinRel}_{\mathsf{R}} & \mathsf{Mat}^{\mathrm{op}}_{\mathsf{R}} \arrow[l, hook']
\end{tikzcd} \tag{2.11}$$

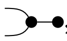
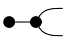
In $\mathbb{HA}_{\mathsf{R}}$ (and $\mathbb{HA}^{\mathrm{op}}_{\mathsf{R}}$) the black and white structure are easily distinguished. However, the combined theory $\mathbb{IH}_{\mathsf{R}}$ is completely symmetric with respect to colour—we could define $\mathcal{S}'_{\mathbb{IH}_{\mathsf{R}}}$ by

 $\mapsto \left\{\left(\begin{pmatrix} x \\ y \end{pmatrix}, x+y\right) \mid x, y \in \mathsf{k}\right\}$  $\mapsto \{(x, (x \quad x)) \mid x \in \mathsf{k}\}$ etc.

and this would also give an isomorphism to $\mathsf{LinRel}_{\mathsf{k}}$. With the same changes made to $\mathcal{S}_{\mathbb{HA}_{\mathsf{R}}}$ and $\mathcal{S}_{\mathbb{HA}^{\mathrm{op}}_{\mathsf{R}}}$, (2.11) will still commute! We will resolve this mystery in Chapter 3.

We conclude this section by observing that $\mathbb{IH}_{\mathsf{R}}$ is compact closed (Definition 2.18), via the black structure.

Lemma 2.26. $\mathbb{IH}_{\mathsf{R}}$ *is compact closed.*

Proof. Define $\varepsilon_1 : 1 + 1 \to 0$ as , η_1 as  and use requirements (ii) and (iii) of the definition of compact closure to inductively generalise them to all n. The Frobenius axioms ensure that condition (i) holds. □

2.3 Bicategories

The final theoretical ingredient needed for this thesis is a way to talk about relationships between morphisms. There are several ways to do this, including bicategories and enriched categories. Locally posetal bicategories are the same as categories enriched in Poset and locally posetal 2-categories. We use bicategories because they permit economical notation and axiomatisation, and because they are the standard in this area of research (see for example [BPS17]). In the case of posets and partially ordered sets, there is little practical difference between the various approaches.

Just as categories generalise sets and functions between them, bicategories generalise categories, functors and natural transformations. In full generality, bicategories can be rather awkward, but we will only need the (rather more tractable) *locally posetal* ones.

Informally speaking, a locally posetal bicategory is a category in which one may compare morphisms in the same hom-set, in a way compatible with the composition. That is, every hom-set should have the extra structure of a partial order $\leq$ (a reflexive and transitive binary relation) and for any $f, f' : X \to Y$ and $g, g' : Y \to Z$, if $f \leq f'$ and $g \leq g'$ then $f; g \leq f'; g'$. We will also consider examples where we only have a preorder, which omits this last antisymmetry requirement.

Any category is trivially a locally posetal bicategory, by means of the discrete ordering where $f \leq f'$ iff $f = f'$, but many of the examples we have met already admit a more interesting posetal structure. As we deal mostly with props, we will also define ordered and preordered props, which are locally (pre)ordered bicategories which are further props with $f \leq f'$ and $g \leq g'$ implying $f \oplus g \leq f' \oplus g'$. The prototypical examples of locally posetal bicategories are the relational categories we have already defined, where the partial order is subset inclusion.

Definition 2.27. A *locally posetal bicategory* is a category with a partial order on morphisms $\leq$ such that if $f \leq f'$ then f and f' have the same domain and the same codomain, and whenever $f, f' : X \to Y, g, g' : Y \to Z$ if $f \leq f'$ and $g \leq g'$ then $f; g \leq f'; g'$.

Often we will use the same name to refer both to a locally posetal bicategory and the category with the same objects and morphisms. For example, we might define the following:

Definition 2.28. The locally posetal bicategory of linear relations over a field k, denoted $\mathsf{LinRel}_{\mathsf{k}}$, has as objects the natural numbers, and as morphisms $m \to n$ the linear subspaces of $\mathsf{k}^m \times \mathsf{k}^n$ as before. $R \leq S$ if and only if $R \subseteq S$ as subspaces (i.e. as sets).

It is normally clear from context whether we are considering categories or locally posetal bicategories. If there is any chance of confusion, we will explicitly say which we are dealing with, so this conflation of notation will not cause any problems.

Chapter 3

Handling inequalities

This chapter provides the main theoretical definitions and results to allow our diagrammatic techniques to be used to reason about inclusion and inequalities. Our first step is to reinterpret the definition of locally posetal bicategory into a suitable version for props. Next, we define the *symmetric monoidal inequality theory* (SMIT), the version of SMT appropriate for generating ordered props. After establishing the relationship between SMTs and SMITs, we extend the presentation claims from Chapter 2 to the corresponding results for SMITs and ordered props, culminating in an equivalence $\mathbb{IH}_{\mathsf{R}} \cong \mathsf{LinRel}_{\mathsf{k}}$.

Often the interpretation of a prop of string diagrams is relational. In contrast to functions, relations offer a more general picture, allowing for non-totality and non-uniqueness. Additionally, they support a natural notion of *inclusion*, and it is the aim of this section to extend the notion of SMT to capture this concept. First, we need to define two different versions of ordered prop.

Definition 3.1. An *ordered prop* is a prop equipped with a partial order on each hom-set, and a *preordered prop* is a prop with a preorder on each hom-set. We abuse notation to write all of these orders as inequality $\leq$ disambiguated by the type of the morphisms on either side (since those types must be the same). Furthermore, these orderings must be compatible with each other and the other operations of the prop, so for all morphisms f, f', g, g'

$$\text{if } f \leq f' \text{ and } g \leq g' \text{ then } f;g \leq f';g' \tag{3.1}$$

$$\text{if } f \leq f' \text{ and } g \leq g' \text{ then } f \oplus g \leq f' \oplus g'. \tag{3.2}$$

whenever the composites exist. Together with (pre)ordered prop morphisms (prop morphisms which are monotone on every hom-set) there are categories of preordered props **POrdPROP** and ordered props **OrdPROP**.

Note that (pre)ordered props are locally posetal bicategories. Any prop can trivially be made into an ordered prop with the discrete order (where $f \leq f'$ iff $f = f'$; see Remark 3.3 below) but more interestingly Rel_X and $\mathsf{LinRel}_{\mathsf{k}}$ have the richer structure of subset inclusion, and this is what we wish to capture in our graphical theories.

3.1 Symmetric monoidal inequality theories

To bring the notion of ordering into the graphical notation, we need an analogue of SMT which allows the axiomatisation of order structure in these theories. Just as a preordered prop is a prop with the addition of a preorder on each hom-set, a symmetric monoidal *inequality* theory is a SMT with the addition of some axiomatic inequalities, which we will use to generate the full structure of a preordering on the hom-sets. In most cases of interest to us, the axioms for the ordering, together with the given equations, are enough to ensure that the preorder is a partial order[1]; for this reason, we do not have separate processes to convert a SMIT into a preordered prop and an ordered prop.

Definition 3.2. A *symmetric monoidal inequality theory* (SMIT) consists of a triple (Σ, E, I) where (Σ, E) is a SMT, and I is a set of *inequalities*, which, similarly to equations, are pairs (c, c') of Σ-terms with the same sort.

From (Σ, E, I) we can generate a preordered prop $\mathbb{T}_{(\Sigma,E,I)}$. The starting point is the prop $\mathbb{T}_{(\Sigma,E)}$, to each hom-set of which we must add a preorder based on I satisfying Equations (3.1) and (3.2). In fact, we will define a preorder $\leq_I$ on $\bigcup_{n,m} \mathbb{T}_{(\Sigma,E)}(n,m)$ such that $-\boxed{A}- \leq_I -\boxed{B}-$ implies that $-\boxed{A}-$ and $-\boxed{B}-$ are in the same hom-set. We use a similar approach as when defining $\sim_E$ in Definition 2.19, by letting

$$\leq_0 = I \cup \{(-\boxed{A}-, -\boxed{A}-) \mid -\boxed{A}- \text{ is any morphism of } \mathbb{T}_{(\Sigma,E)}\} \tag{3.3}$$

$$\leq_{n+1} = \leq_n \cup \{(-\boxed{A}-, -\boxed{C}-) \mid \exists -\boxed{B}- \text{ such that } (-\boxed{A}-, -\boxed{B}-), (-\boxed{B}-, -\boxed{C}-) \in \leq_n\} \tag{3.4}$$

$$\cup \{(-\boxed{A}-\boxed{B}-, -\boxed{A'}-\boxed{B'}-) \mid (-\boxed{A}-, -\boxed{A'}-), (-\boxed{B}-, -\boxed{B'}-) \in \leq_n\} \tag{3.5}$$

$$\cup \left\{ \left(\begin{matrix} -\boxed{A}- \\ -\boxed{B}- \end{matrix}, \begin{matrix} -\boxed{A'}- \\ -\boxed{B'}- \end{matrix} \right) \mid (-\boxed{A}-, -\boxed{A'}-), (-\boxed{B}-, -\boxed{B}-) \in \leq_n \right\} \tag{3.6}$$

and $\leq_I = \bigcup_{n=0}^{\infty} \leq_n$. Then, for each hom-set, the restriction of $\leq_I$ is clearly a preorder, contains I, satisfies Equations (3.1) and (3.2), and so defines a preordered prop structure on $\mathbb{T}_{(\Sigma,E)}$.

This definition allows us to reason inequationally in exactly the same way as we could use equations in a SMT—each step of reasoning is by applying an axiom to a subsection of a string diagram until the desired result is achieved.

Remark 3.3. SMITs generalise SMTs. Any prop can be made into a discrete preordered prop by adding identity 2-cells. At the level of presentations, this corresponds to mapping (Σ, E) to $(\Sigma, E, \varnothing)$: the union defining $\leq_0$ clearly corresponds to adding these identities, and since I itself is empty it's obvious that $\leq_{n+1} = \leq_n$ for all n. This gives an embedding (of categories) $D : \mathbf{PROP} \hookrightarrow \mathbf{POrdPROP}$ such that $D(\mathbb{T}_{(\Sigma,E)}) = \mathbb{T}_{(\Sigma,E,\varnothing)}$.

Moreover, D is left adjoint to the forgetful functor $U : \mathbf{POrdPROP} \to \mathbf{PROP}$. To see this, we must show that for any prop $\mathbb{T}$ and any preordered prop $\mathbb{U}$ there is a natural bijection between the hom-sets $\mathbf{POrdPROP}(D\mathbb{T}, \mathbb{U})$ and $\mathbf{PROP}(\mathbb{T}, U\mathbb{U})$. But that is obvious, because any preordered prop morphism whose domain is discrete has no restriction beyond simply being a prop morphism, since the requirement of being monotone is always vacuously true. Similarly, every prop morphism $\mathbb{T} \to U\mathbb{U}$ always lifts to a preordered prop morphism $D\mathbb{T} \to \mathbb{U}$.

Using SMITs, we can now capture the orderings of $\mathsf{LinRel}_\mathsf{k}$ and FinRel. Our proofs use a characterisation of relations as spans. By the universal property of the product, any relation $R \subseteq X \times Y$ can be split into two parts $X \leftarrow R \rightarrow Y$. If a category is sufficiently well-behaved, we can define relations over that category as a class of spans.

Definition 3.4 ([Bé67]). Let C be a category with finite products, pullbacks and an epi-mono factorisation system. A *span* from an object A to another object B is some object X and a pair of morphisms in the shape

[1] That is, $f \leq f'$ and $f' \leq f$ together imply that f and f' are equal, or, in more category-theoretic terminology, isomorphic morphisms are equal.

$A \leftarrow X \rightarrow B$. A *span morphism* from $A \leftarrow X \rightarrow B$ to $A \leftarrow X' \rightarrow B$ is a morphism $X \rightarrow X'$ in C making commute

$$\begin{tikzcd} & X \arrow[dl] \arrow[dr] \arrow[d] & \\ A & X' \arrow[l] \arrow[r] & B \end{tikzcd}$$

The *bicategory of relations* over C $\mathsf{Rel}(C)$ has the same objects as C. An morphism $A \rightarrow B$ is an isomorphism class (with respect to span morphisms) of jointly monic spans from A to B, i.e. some $A \leftarrow X \rightarrow B$ such that the induced $X \rightarrow A \times B$ is monic. The identity morphism at A is (the isomorphism class containing) the identity span $A \xleftarrow{\text{id}} A \xrightarrow{\text{id}} A$. Composition is by pullback followed by restriction to the monic part of the factorisation, and the 2-cells are span morphisms. By monicity, there is at most one 2-cell between two relations, so $\mathsf{Rel}(C)$ is always locally posetal.

Dually, if C has finite coproducts and pushouts, the *bicategory of corelations* consists of jointly epic cospans $A \rightarrow X \leftarrow B$, and composition is by pushout followed by restriction to the epic part. Note that the sense of the order is *not* reversed: $A \rightarrow X \leftarrow B \leq A \rightarrow X' \leftarrow B$ iff there is a suitable morphism $X \rightarrow X'$.

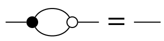
We begin with the relatively straightforward treatment of finite relations, first by noting that the above notion of relation agrees with the direct definition.

Proposition 3.5 (folklore). $\mathsf{Rel}(\mathsf{FinSet}) \cong \mathsf{FinRel}$ *as props.*

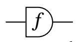
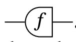
An immediate question is whether $\mathsf{Corel}(\mathsf{FinSet})$ similarly corresponds to some natural construction on finite sets. The answer is affirmative: corelations are the equivalence relations EqRel. The ordering is by coarseness: $\alpha \leq \beta$ iff $(i, j) \in \alpha$ implies $(i, j) \in \beta$ for all (i, j).

Proposition 3.6 (folklore). $\mathsf{Corel}(\mathsf{FinSet}) \cong \mathsf{EqRel}$ *as props.*

Coya and Fong [CF17] give axiomatisations of $\mathsf{Rel}(\mathsf{FinSet})$ and $\mathsf{Corel}(\mathsf{FinSet})$ as SMTs. $\mathsf{Rel}(\mathsf{FinSet})$ is presented by B from Example 2.23(iii) with one additional equation, the so-called *special equation*, seen already (although coloured differently) as part of the Frobenius SMT:

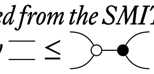


Writing B' for the result of augmenting B with this equation, the isomorphism $\mathcal{S} : \mathsf{Rel}(\mathsf{FinSet}) \rightarrow B'$ acts similarly to the one $\mathsf{FinSet} \rightarrow M$: the ith port on the left is connected to the jth port on the right iff (i, j) is in the relation. The special equation serves to remove redundant paths. $\mathsf{Corel}(\mathsf{FinSet})$ is presented by the Frobenius monoids from Example 2.23(v).

Both B and B' may be constructed as a composition of props [Lac04]; for this reason, both of them enjoy a useful factorisation property. Every morphism may be written so that all of the comonoid generators are to the left of the monoid generators. $\mathcal{S}$ is well-behaved with respect to this factorisation, in the sense that it restricts to isomorphisms $\mathsf{FinSet} \rightarrow M$ and $\mathsf{FinSet}^{\mathrm{op}} \rightarrow C$. We abuse the notation for scalars to write the image of a FinSet morphism f as  and of a $\mathsf{FinSet}^{\mathrm{op}}$ one as .

We extend Propositions 3.5 and 3.6 to the level of SMITs and ordered props.

Theorem 3.7. *(a) The ordered prop generated by the SMIT consisting of the SMT of bimonoids from Example 2.23(iii) together with the special equation and the inequality* $\leq$ — *is isomorphic (as an ordered prop) to* $\mathsf{Rel}(\mathsf{FinSet})$.

(b) The ordered prop generated from the SMIT which is the SMT for special Frobenius monoids of Example 2.23(v) along with the inequality  *is isomorphic (as an ordered prop) to* $\mathsf{Corel}(\mathsf{FinSet})$.

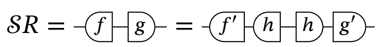
Proof. For part (a), we extend $\mathcal{S}$ to an isomorphism of ordered props. It is enough to check that the new inequality is sound (that is, the image of every inequality under $\mathcal{S}$ is true) and complete for the order in FinRel. It is clearly sound; the image of the left hand side is the empty relation $\varnothing \subseteq \overline{1} \times \overline{1}$, and the right hand side is the full relation.

For completeness, suppose we have relations $R \subseteq R' : m \to n$. Then they are jointly mono spans over FinSet $m \xleftarrow{f} z \xrightarrow{g} n$ and $m \xleftarrow{f'} z' \xrightarrow{g'} n$ with some $h : z \to z'$ making the following commute

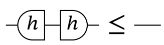
$$\begin{tikzcd} & z \arrow[dl, "f"'] \arrow[dr, "g"] \arrow[d, "h"] & \\ m & z' \arrow[l, "f'"] \arrow[r, "g'"'] & n \end{tikzcd}$$

Diagrammatically,

and by functoriality of $\mathcal{S}$ we may write

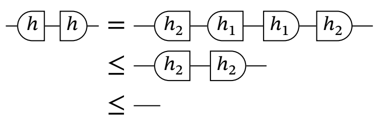


So to show completeness, it is enough to show that for any FinSet-morphism h



(3.7)

We use induction on the structure of a term in M. If $h = h_1; h_2$ where h_1 and h_2 each satisfy (3.7), then

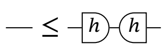


since $\mathcal{S}$ is a functor
by inductive hypothesis and (3.1)
by inductive hypothesis

The inductive step for $\oplus$ is similar, using (3.2) instead.

For the base case, there are two generators to check. For the multiplication $: 2 \to 1$, we in fact have $=$ — as an axiom. The other case are the units, where we have explicitly added the required inequality $\leq$ —.

For part (b), analogously to part (a), we need only show that for any FinSet-morphism h,



The induction step is the same as before, the case of units is the axiom $=$, and the case of multiplication is the exact inequality we added to the SMIT. $\square$

It is worth explaining the relationship of $\mathcal{S}_{\mathbb{IH}}$ with the isomorphism $\mathcal{S}_{\mathbb{HA}}$ from Proposition 2.24(ii). Observe that any string diagram of $\mathbb{IH}_{\mathsf{R}}$ built out of the generators $\{\ ,\ ,\ ,\ ,\ \}$ is also a string diagram in $\mathbb{HA}_{\mathsf{R}}$ (for such a diagram $A : k \to l$ we further abuse scalar notation to write $\overset{k}{—}\boxed{A}\overset{l}{—}$) and, similarly, any diagram built of the opposite generators ($\overset{k}{—}\boxed{A}\overset{l}{—}$) is a string diagram in $\mathbb{HA}_{\mathsf{R}}^{\mathrm{op}}$. This means that we have two embeddings of props $\mathbb{HA}_{\mathsf{R}} \to \mathbb{IH}_{\mathsf{R}} \leftarrow \mathbb{HA}_{\mathsf{R}}^{\mathrm{op}}$. Similarly, we have two embeddings $\mathsf{Mat}_{\mathsf{R}} \to \mathsf{LinRel}_{\mathsf{k}} \leftarrow \mathsf{Mat}_{\mathsf{R}}^{\mathrm{op}}$ mapping every matrix to its (opposite) graph. In [BSZ17b], it shown that the following diagram commutes.

$$\begin{tikzcd} \mathbb{HA}_{\mathsf{R}} \arrow[r] \arrow[d, "\mathcal{S}_{\mathbb{HA}}"'] & \mathbb{IH}_{\mathsf{R}} \arrow[d, "\mathcal{S}_{\mathbb{IH}}"] & \mathbb{HA}_{\mathsf{R}}^{\mathrm{op}} \arrow[l] \arrow[d, "\mathcal{S}_{\mathbb{HA}^{\mathrm{op}}}"] \\ \mathsf{Mat}_{\mathsf{R}} \arrow[r] & \mathsf{LinRel}_{\mathsf{k}} & \mathsf{Mat}_{\mathsf{R}}^{\mathrm{op}} \arrow[l] \end{tikzcd} \tag{3.8}$$

Intuitively, this means that any diagram $\overset{m}{-}\boxed{A\rangle}\overset{n}{-}$ corresponds (via $\mathcal{S}_{\mathbb{IH}}$) to a functional linear relation (matrix) and $\overset{m}{-}\boxed{\langle A}\overset{n}{-}$ to a cofunctional linear relation (reversed matrix). The following result informs us that every morphism of $\mathbb{IH}_{\mathsf{R}}$ can be written in both *span form* and *cospan form*.

Lemma 3.8 ([BSZ17b]). *For all* $\overset{m}{-}\boxed{A}\overset{n}{-}$ *in* $\mathbb{IH}_{\mathsf{R}}$ *there exist* $k, k' \in \mathbb{N}$, $\overset{m}{-}\boxed{\langle A_1}\overset{k}{-}$, $\overset{k}{-}\boxed{A_2\rangle}\overset{n}{-}$, $\overset{m}{-}\boxed{A_1'\rangle}\overset{k'}{-}$ *and* $\overset{k'}{-}\boxed{\langle A_2'}\overset{n}{-}$ *such that*

$$\overset{m}{-}\boxed{A_1'\rangle}\overset{k'}{-}\boxed{\langle A_2'}\overset{n}{-} \;=\; \overset{m}{-}\boxed{A}\overset{n}{-} \;=\; \overset{m}{-}\boxed{\langle A_1}\overset{k}{-}\boxed{A_2\rangle}\overset{n}{-}$$

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

Lemma 3.9 ([BSZ17b]). *For all* $\overset{m}{-}\boxed{A\rangle}\overset{n}{-}$, $\overset{m}{-}\boxed{A\rangle}\overset{n}{-}\bullet \;=\; \overset{m}{-}\bullet$ *and* $\circ\overset{m}{-}\boxed{A\rangle}\overset{n}{-} \;=\; \circ\overset{n}{-}$.

We show that it is possible to extend the SMT $\mathbb{IH}_{\mathsf{R}}$ to a SMIT, in a way compatible with the isomorphism $\mathcal{S}_{\mathbb{IH}} : \mathbb{IH}_{\mathsf{R}} \to \mathsf{LinRel}_{\mathsf{k}}$. In other words, we give an inequational characterisation of the subset order of linear relations. The symmetry under $(-)^{\bullet}$ discussed earlier is broken by moving to the ordered setting. Indeed, to get from the SMT to a SMIT we add *just one inequality*:

$$-\!\circ \;\leq\; -\!\bullet \tag{3.9}$$

Interpreted as linear relations (via $\mathcal{S}_{\mathbb{IH}}$), (3.9) means simply that the unique 0-dimensional subspace $\{0\}$ of k is included in the unique 1-dimensional subspace, k itself.

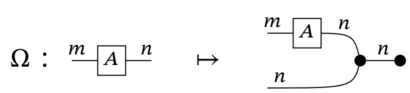
Theorem 3.10. $\mathbb{IH}_{\mathsf{R}} \cong \mathsf{LinRel}_{\mathsf{k}}$ *as ordered props.*

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 $\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n$, recall that $\mathrm{Sp}(A)$, the *span* of A, is the linear subspace of k^m with elements linear combinations $\lambda_1\mathbf{a}_1 + \lambda_2\mathbf{a}_2 + \cdots + \lambda_n\mathbf{a}_n$, for $\lambda_i \in \mathsf{k}$; this is easily generalised to multiple matrices with the same number of rows by considering linear combinations of all columns together. The following is a well-known fact of linear algebra.

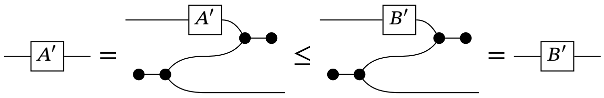
Lemma 3.11 (e.g. [Axl97, Proposition 2.13]). *Let V be a vector space over a field k with a linear subspace U. Suppose that for some $m \times n$ k-matrix A we have $\mathrm{Sp}(A) \subseteq U$. Then there exist some n' and an $m \times n'$ matrix C such that $U = \mathrm{Sp}(A, C)$.*

Proof of Theorem 3.10. The inequality (3.9) is clearly sound, we thus only have to show completeness: that (3.9) suffices to account for any inclusion between *arbitrary* linear relations.

Let therefore $A, B : m \to n$ be morphisms 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, corresponds to the manipulation

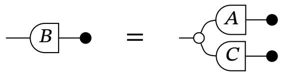


In fact, because of this 'rewiring' argument, we may assume, without loss of generality, that $A, B : m \to 0$. If the theorem holds for all such diagrams, then for arbitrary $A', B' : m \to n$ with $\mathcal{S}(A') \subseteq \mathcal{S}(B')$, we also have $\mathcal{S}(\Omega A') \subseteq \mathcal{S}(\Omega B')$. So by assumption $\Omega A' \leq \Omega B'$, but then we can reason

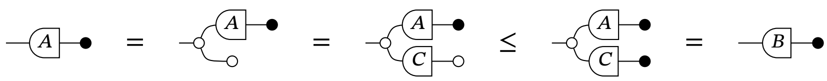


Further, using Lemma 3.8 and Lemma 3.9, we may show = = . Therefore we can further assume that A, B consist only of generators of $\mathbb{HA}_{\mathsf{R}}^{\mathrm{op}}$. It is thus harmless to consider A and B as (reflected) matrices, and our initial assumption means that $\mathrm{Sp}(A) \subseteq \mathrm{Sp}(B)$, since clearly $\mathcal{S}(\text{-}\boxed{A}\text{-}\bullet) = \mathrm{Sp}(A)$.

By the conclusion of Lemma 3.11, there exists C such that $\mathrm{Sp}(B) = \mathrm{Sp}(A, C)$. Diagrammatically (via $\mathcal{S}_{\mathbb{IH}}$), this joint span is the following, where for readability we omit decorating the wires:



But we have

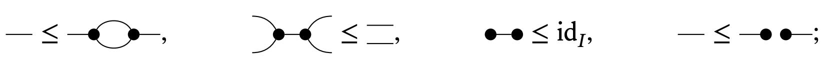


showing that the inequality $A \leq B$ is derivable from (3.9). $\square$

We also show that $\mathbb{IH}_{\mathsf{R}}$ enjoys a particularly nice form of order, namely that of a bicategory of relations, in the sense of Carboni and Walters [CW87].

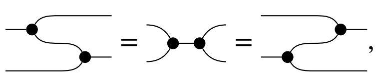
Definition 3.12. A bicategory of relations is a locally posetal monoidal bicategory where

1. for every object a there is a commutative comonoid structure, that is, a pair of morphisms (: $1 \to 2$, : $1 \to 0$), along with right adjoints $\dashv$, $\dashv$. Adjointness means to the following, here and onwards often omitting the labelling on the wires for clarity:



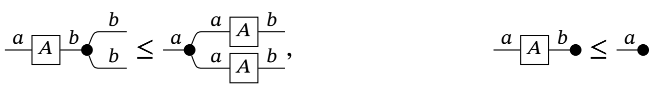
(3.10)

2. (,) and right adjoints satisfy the Frobenius equations:



(3.11)

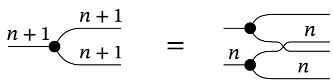
3. every morphism $\boxed{A}$ is a lax (,)-comonoid homomorphism, that is,



(3.12)

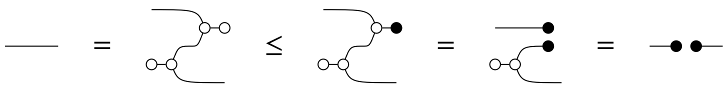
Theorem 3.13. *The ordered prop $\mathbb{IH}_{\mathsf{R}}$ is a bicategory of relations.*

Proof. For each object $n \in \mathbb{N}$, comonoid structures are defined inductively with the base case for $n = 0$ as the empty diagram and



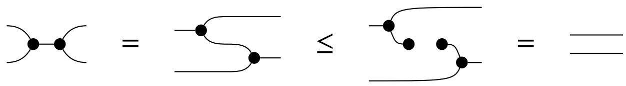
A straightforward induction generalises the comonoid equations for all n.

We first tackle the case of the black units (the second pair of inequalities in (3.10)). First we must check that $\leq$ and these terms are equated by the definition of $\mathbb{IH}_{\mathsf{R}}$. Conversely, for $n = 1$:

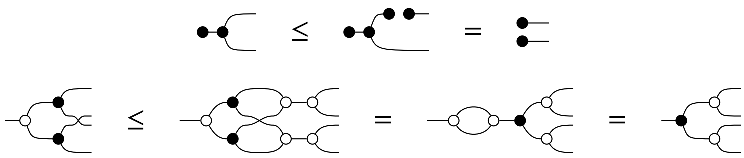


The above argument easily generalises to all n.

Showing adjointness for the black comultiplication (the first pair of inequalities of (3.10)) amounts to demonstrating that $\leq$ and $\leq$. The second is the black special equation in Definition 2.25. The first follows easily from the adjointness of the unit and counit:



Now, to show that all morphisms are lax comonoid homomorphisms, it is enough to check that each of the generators obeys the conditions of (3.12). Several of these are in fact equalities. The derivations for the two interesting cases are given below:

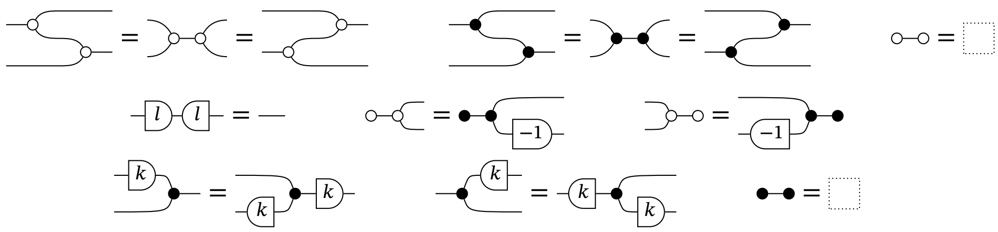


□

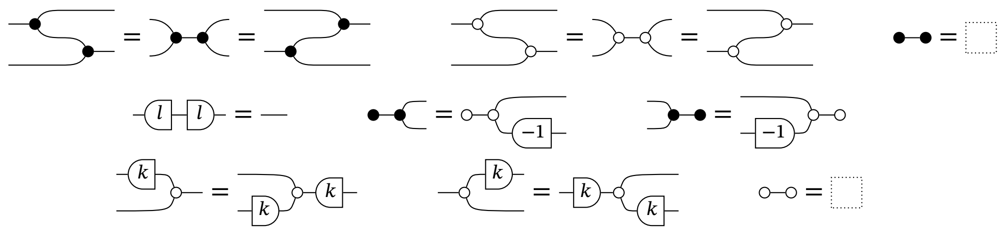
We now tackle the more complicated task of axiomatising the order in $\mathsf{Rel}(\mathsf{Mat}_{\mathsf{R}})$ and $\mathsf{Corel}(\mathsf{Mat}_{\mathsf{R}})$. The latter has a non-theoretical application we explore later, although similarly to the case for $\mathsf{Rel}(\mathsf{FinSet})$ and $\mathsf{Corel}(\mathsf{FinSet})$ we get both results for free by duality. While the situation is more involved, essentially the same proof technique as the one for Theorem 3.7 may be used.

We first give the definitions of the SMTs presenting $\mathsf{Rel}(\mathsf{Mat}_{\mathsf{R}})$ and $\mathsf{Corel}(\mathsf{Mat}_{\mathsf{R}})$ as one-dimensional props. They are constructed by quotienting a coproduct, as we have seen before.

Definition 3.14. $\mathbb{IH}^{\mathsf{Rel}}_{\mathsf{R}}$ is the quotient of $\mathbb{HA}_{\mathsf{R}} + \mathbb{HA}^{\mathrm{op}}_{\mathsf{R}}$ by the following equations, where k is any element of R and l ranges over non-zero elements of R.



$\mathbb{IH}^{\mathsf{Cor}}_{\mathsf{R}}$ is similarly a quotient of $\mathbb{HA}_{\mathsf{R}} + \mathbb{HA}^{\mathrm{op}}_{\mathsf{R}}$ by these "photographic negative" versions of the above equations, where k and l are as before.



Note that unlike the case of FinSet, we must be careful to keep track of the colours of our generators, as there are now two different pairs of monoid and comonoid structure.

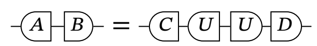
A fuller exploration of the origins of these laws can be found elsewhere [Zan15, FSR16, FZ17]. We simply note that the results of Fong, Rapisarda and Sobociński [FSR16] for the case where $\mathsf{R} = \mathsf{k}[s, s^{-1}]$ generalise to show that these are presentations of the props of interest.

Proposition 3.15. $\mathbb{IH}^{\mathsf{Rel}}_{\mathsf{R}} \cong \mathsf{Rel}(\mathsf{Mat}_{\mathsf{R}})$ *and* $\mathbb{IH}^{\mathsf{Cor}}_{\mathsf{R}} \cong \mathsf{Corel}(\mathsf{Mat}_{\mathsf{R}})$.

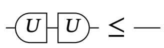
We now ask what extra inequalities must be added to the presentations of $\mathbb{IH}^{\mathsf{Rel}}_{\mathsf{R}}$ and $\mathbb{IH}^{\mathsf{Cor}}_{\mathsf{R}}$ to capture the 2-categorical structure present in the more concrete categories they axiomatise. It turns out that we must work slightly harder than for $\mathbb{IH}_{\mathsf{R}}$ in these weaker theories.

The argument from the previous section can be reused as follows. Suppose we have (jointly monic) spans $\mathsf{R}^m \xleftarrow{A} \mathsf{R}^r \xrightarrow{B} \mathsf{R}^n \leq \mathsf{R}^m \xleftarrow{C} \mathsf{R}^{r'} \xrightarrow{D} \mathsf{R}^n$ in $\mathsf{Rel}(\mathsf{Mat}_{\mathsf{R}})$. Then there is an $r' \times r$ matrix U such that $A = CU$ and $B = DU$. Diagrammatically, this means

in $\mathbb{HA}_{\mathsf{R}}$, carefully noting the reversed order of composition. Thus our span is



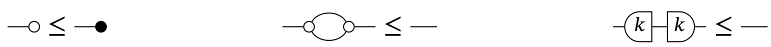
and so to show completeness of any proposed presentation, it is enough to show that for any U,



(3.13)

We will re-use the hopefully by now familiar inequality (3.9). However, without all the equations of $\mathbb{IH}_{\mathsf{R}}$, we also need to add inequalities between some of the diagrams which are equated there.

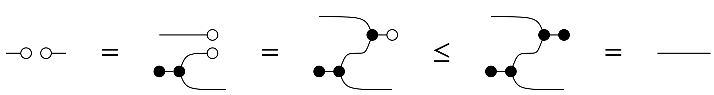
Proposition 3.16. *The order of* $\mathsf{Rel}(\mathsf{Mat}_{\mathsf{R}})$ *is presented by adding the following inequalities to the SMIT derived from the SMT of* $\mathbb{IH}^{\mathsf{Rel}}_{\mathsf{R}}$*:*

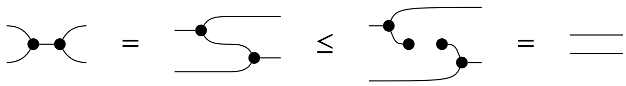


where k ranges over every (non-zero) element of R.

Proof. Soundness is straightforward to check.

For completeness, by the observation above, it is enough to show that (3.13) holds for every U. We proceed by induction on the structure of U, noting that the induction steps for composition and monoidal product are both trivial. We need therefore only consider the 5 cases in which U consists of a single generator of $\mathbb{HA}_{\mathsf{R}}$. An axiom of $\mathbb{IH}^{\mathsf{Rel}}_{\mathsf{R}}$ ensures that = , and two of the imposed inequalities take care of the cases when U is and . The first axiom covers the remaining two cases:

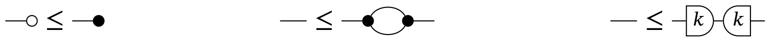




where in the second derivation we use the result (easily shown by reversing the colours and sense of the equality from the first) that — ≤ —● ●—. □

The situation in $\mathbb{IH}_{\mathsf{R}}^{\mathsf{Cor}}$ is dual, as we record in the below corollary.

Corollary 3.17. *The order of* $\mathsf{Corel}(\mathsf{Mat}_{\mathsf{R}})$ *is presented by adding the following inequalities to the SMIT derived from the SMT of* $\mathbb{IH}_{\mathsf{R}}^{\mathsf{Cor}}$*:*



where k ranges over every (non-zero) element of R.

These results are interesting in that they give slightly stronger relationships between the generators that are identified in $\mathbb{IH}_{\mathsf{R}}$ but not the weaker theories. Also, the proof method is rather different to that used for $\mathbb{IH}_{\mathsf{R}}$, since we do not have access to all the machinery of linear algebra that is central to that proof. Notice further that we do not need to include the $k = 0$ case, since we have both "bone" equations available to us, and the axiom of $\mathbb{HA}_{\mathsf{R}}$ that —(0)— = —● ○— allows us to infer that result.

Chapter 4

Signal flow graphs

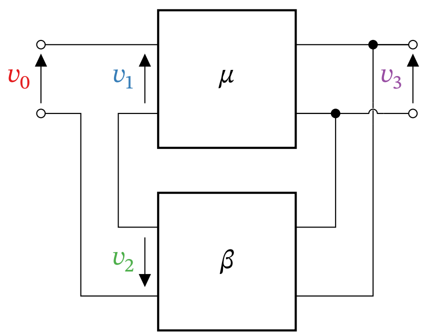
This chapter examines our first application of orderings, to signal flow graphs (SFGs). They date back to Shannon [Sha42], were popularised and studied in detail by Mason [Mas52], and have been used since then as an abstraction for many different systems, ranging from control theory and signal processing [Lat98] to modelling cardiac output [GLK55]: wherever feedback plays an important role.

Mason's interest was in creating a general theory for electronic systems involving feedback, although other than his motivating examples there is nothing specific to electronics:

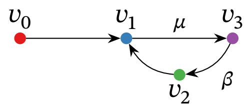
> There remains … a need for development at the working level so that engineers may find the feedback approach more useful … The flow graph provides a visual presentation of the relationships entering an analysis problem and facilitates certain manipulations leading to the solution. [Mas52]

These manipulations are, in fact, special cases of the general system of reasoning in SMTs and SMITs that we have been building throughout this thesis.

The first example analysed by Mason begins with translates the following electronic circuit to a signal flow graph and then to the equivalent system of equations.



Following the laws of circuit theory, we can establish relations between the (colour-coded) voltages v_i corresponding to the flow graph



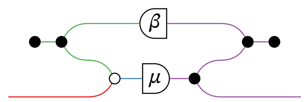
(4.1)

Each node has as its value the sum of the values of the nodes on the ends of incoming arrows, possibly multiplied by a factor (here μ or β); v_0 has no dependencies on other nodes and may therefore have any arbitrary value. Thus the graph translates into the system

$$\begin{aligned} v_0 &= \text{any given input} \\ v_1 &= v_0 + v_2 \\ v_2 &= \beta v_3 \\ v_3 &= \mu v_1 \end{aligned}$$

and that may be readily solved using conventional techniques to give $v_3 = \frac{\mu}{1-\mu\beta} v_0$. Our signal flow graph defines a transformation from an input voltage v_0 to an output v_3 via a feedback loop passing through v_2 and v_1. Moreover, it is straightforward to go the other way, and turn a linear system of equations back into a signal flow graph.

We deviate somewhat from the notation of signal flow graphs as they were first examined in the middle of the 20th century, and draw (and indeed define) them in a format more amenable to analysis using the ideas of SMTs and SMITs. We redraw the graph of (4.1) as



The individual nodes v_i have been turned into stretches of wire, where v_0 is the lower left open 'input' wire leading into the adder, v_1 is the short part from the output of the addition to the μ generator, v_2 is the upper left part from the output of the β generator round to the upper side of the adder, and v_3 is the whole right-hand half, including the trailing 'output' loose end.

We will also move away from the continuous world of electronics to a more discrete view. Our input is not some constant voltage that we assume has settled to an equilibrium state, but a (possibly infinite) stream of values that are fed in one by one.

Synopsis

We build on previous results [BSZ14, BSZ15, BSZ17a] axiomatising SFGs as string diagrams inside props, and show that the isomorphisms extend to the level of ordered props. To do this, we recall the definitions of formal Laurent and power series, as well as polynomial rings. We also give the definition of a prop of signal flow graphs as a sub-prop of $\mathbb{IH}_{\mathsf{k}[x]}$, where our base PID is now the ring of polynomials over a field.

We define specifications as string diagrams, and explain what it means for a SFG to implement a specification. We cover ways of defining semantics either denotationally or operationally, and extend the existing results about their compatibility to show that they still agree in the framework of ordered props. To do this, we explore notions around deadlock and initialisation.

After the above exposition, which is all focused on having delays always start containing a value of zero, we examine the situation where any field member is allowed, and show similar compatibility results.

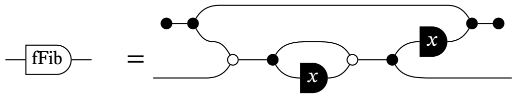
The key results in this section are

- Theorem 4.5, establishing that inclusion of behaviours according to the denotational semantics is the same as syntactic inclusion according to the SMIT;
- Theorem 4.8, showing that, for diagrams which do not exhibit certain degenerate behaviours, the operational semantics also corresponds to the inclusion in the SMIT; and
- Proposition 4.12, showing that, when we allow delays to be initialised with non-zero values, extending traces backwards in time does not harm the correspondence between semantics and syntax.

4.1 Zero-initialised SFGs

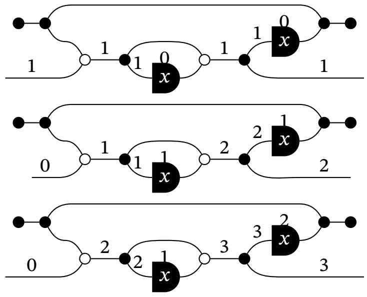
Throughout this chapter, we conflate a signal flow graph as historically defined with the version re-cast into a notation which fits within our string diagrams, and indeed we can use the theories of SMTs and SMITs to reason about SFGs.

An example of a signal flow graph is below, modelling rabbits which breed according to the well-known (though perhaps biologically implausible) rules of Fibonacci: each month, every mature pair of rabbits gives birth to an immature pair which will mature and start breeding the next month.



(4.2)

We interpret  gates as delays, or registers, which we assume to be initialised with zero, justifying their different appearance to members of k which simply act as amplifiers. An important element is the feedback loop passing through a delay. 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 steps of the computation below: the state of each delay is shown by the number above it, and the remaining numbers keep track of the value on each wire at each iteration.



For the remainder of this chapter, we fix a field k. Our first step is to define some k-algebras we will use as our domains of behaviours of SFGs, in the spirit of generating functions.

Definition 4.1. A *formal Laurent series* is a function $\sigma : \mathbb{Z} \to \mathsf{k}$ such that there is some $i \in \mathbb{Z}$ for which $\sigma(j) = 0$ for all $j < i$. The least $d \in \mathbb{Z}$ with $\sigma(d) \neq 0$ is the *order* of σ.

We will often write a formal Laurent series as a formal sum $\sum_{i=d}^{\infty} \sigma(i)x^i$. Sum and product of $\sigma = \sum_{i=d}^{\infty} \sigma(i)x^i$ and $\tau = \sum_{i=e}^{\infty} \tau(i)x^i$ are defined in the normal way:

$$\sigma + \tau = \sum_{i=\min(d,e)}^{\infty} (\sigma(i) + \tau(i))\, x^i$$

$$\sigma \cdot \tau = \sum_{i=\min(d,e)}^{\infty} \left(\sum_{k+j=i} \sigma(j) \cdot \tau(k) \right) x^i$$

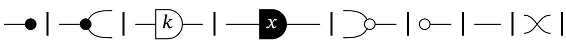
and if σ is non-zero, then it has inverse σ^{-1} given by

$$\sigma^{-1}(i) = \begin{cases} 0 & \text{if } i < -d \\ \sigma(d)^{-1} & \text{if } i = -d \\ \frac{\sum_{i=1}^{n} (\sigma(d+i) \cdot \sigma^{-1}(-d+n-i))}{-\sigma(d)} & \text{if } i = -d + n \text{ for } n \neq 0 \end{cases}$$

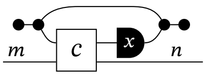
With these operations, formal Laurent series form a field denoted by $\mathsf{k}((x))$. Important sub-k-algebras are the *formal power series*, the formal Laurent series of order ≥ 0, written $\mathsf{k}[[x]]$, and the *polynomials* $\mathsf{k}[x]$.

We identify the prop SF consisting of signal flow graphs, which allow a well-defined, mechanistic, left-to-right execution, in contrast to general string diagrams, where the non-functional generators such as and mean that this is impossible. Note that signal flow graphs as defined here are only equated up to the laws of props, not all the equations of $\mathbb{IH}_{\mathsf{k}[x]}$. We will also consider the image of SF under what is effectively the quotient map $q : \mathsf{SF} \to \mathbb{IH}_{\mathsf{k}[x]}$, but for the purposes of cleanly defining the operational semantics, it is important not to introduce arbitrary polynomials as generators. Note that we use this roman font to emphasise the distinction, as opposed to the blackboard bold one for our theories with more equations imposed. Indeed, we will also use the notation IH for the prop with all the generators of $\mathbb{IH}_{\mathsf{k}}$ along with x and x but no equations.

Definition 4.2. The *prop of signal flow graphs* SF consists of diagrams matching the syntax

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

where k ranges over k, we allow only well-defined composition in the sense of $\mathbb{IH}_{\mathsf{k}[x]}$ and by $Tr(c)$ we have abbreviated the following "guarded feedback" operation mapping $1 + m \to 1 + n$ diagrams to the $m \to n$ ones

$Tr(c) =$ 

taken up to the laws of props (as in Definition 2.11). Since every generator of SF is also a generator of $\mathbb{IH}_{\mathsf{k}[x]}$ and we have imposed no additional equations beyond the prop axioms, we immediately get a prop morphism $q : \mathsf{SF} \to \mathbb{IH}_{\mathsf{k}[x]}$. Indeed, viewing SF as a discrete ordered prop, q is also a morphism of ordered props.

Note that a priori the image $q(\mathsf{SF})$ is very far from being all of $\mathbb{IH}_{\mathsf{k}[x]}$. General string diagrams can be thought of as *specifications* and SFGs in SF as *implementations*. We say that a specification A (a morphism of $\mathbb{IH}_{\mathsf{k}[x]}$) *refines* a specification B whenever $A \leq B$ and we say that a diagram $c \in \mathsf{SF}$ *implements* a specification A whenever $q(c)$ refines A, that is, $q(c) \leq A$.

We define operational semantics on *augmented diagrams* via the rules shown in Figure 4.1. An augmented diagram is obtained from a diagram of IH by replacing delays x with labelled versions x^{k}, for some $k \in \mathsf{k}$. 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. Operational semantics are defined for *every* diagram of IH, but those of SF are special in that they admit a deterministic execution associating a stream of inputs on the left with a stream of outputs on the right. Each diagram c then yields a transition system with the *initial state* s_0 of c obtained by replacing delays in c with x^{0}; this means that we only consider executions where the registers are initialised with 0. For a SFG in some state s, every choice $\mathbf{v}$ uniquely determines a $\mathbf{w}$ and t such that there is $s \xrightarrow[\mathbf{w}]{\mathbf{v}} t$, but this may not be true for an arbitrary string diagram. Different semantics are considered in [FSR16] and below in Section 4.2, where registers can be initialised with arbitrary values.

A *computation* of a diagram c is a (possibly 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 type $m \to n$, 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 $\boldsymbol{\alpha} = \left(\begin{smallmatrix}\alpha_1 \\ \vdots \\ \alpha_m\end{smallmatrix}\right)$, $\boldsymbol{\beta} = \left(\begin{smallmatrix}\beta_1 \\ \vdots \\ \beta_n\end{smallmatrix}\right)$ where $\alpha_j = k_{0j}k_{1j} \ldots$ and $\beta_j = l_{0j}l_{1j} \ldots$. In a finite computation, all α_j and β_j have the same length, and in an infinite computation they are all infinite. We denote by $ft(c)$ the set of all finite trajectories of a diagram c, and by $it(c)$ the set of all its infinite ones. The *observable behaviour* $\langle c \rangle$ of a diagram c is the pair $(ft(c), it(c))$. We can encode a finite trajectory as a pair of vectors of polynomials in $\mathsf{k}[x]$ and an infinite one as a pair of vectors of formal power series in $\mathsf{k}[[x]]$.

We have the following guarantee of compositionality for $\langle - \rangle$.

Proposition 4.3. $\langle - \rangle : \mathsf{SF} \to \mathsf{LinRel}_{\mathsf{k}[x]} \times \mathsf{LinRel}_{\mathsf{k}[[x]]}$ *is a morphism of ordered props.*

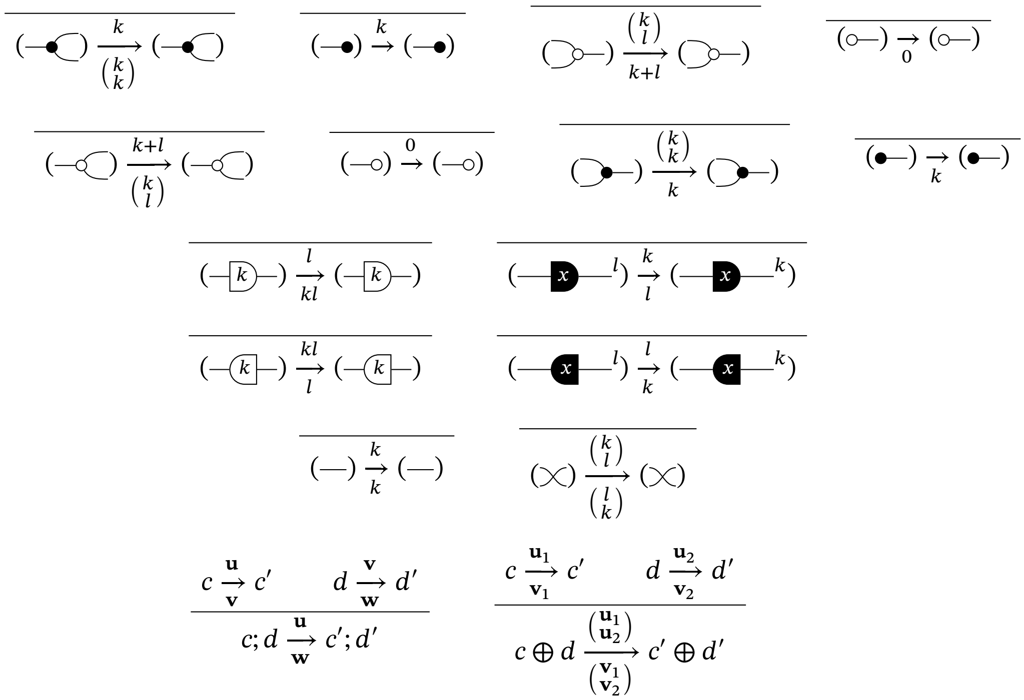


Figure 4.1: Structural rules for operational semantics, with k, l ranging over k and $\mathbf{u}$, $\mathbf{v}$, $\mathbf{w}$ vectors of elements of k of the appropriate size.

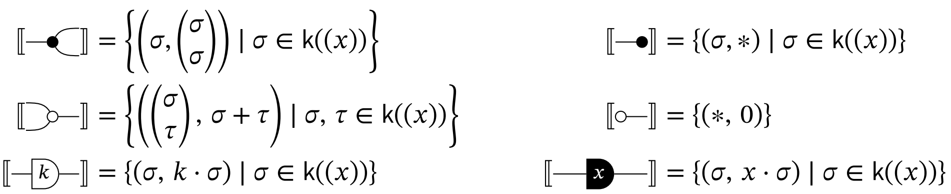
Proof. Since SF has no equations or inequalities, it is enough to note that the identities and composition are compatible with those in the codomain. □

We now show that the notion of inclusion of behaviours tightly corresponds to this syntactic notion of refinement. Before we can state the precise result, we must explore the ideas of deadlock and initialisation.

We begin by establishing denotational semantics and the various domains of interpretation we will be using throughout this section.

Our domain of denotation for diagrams is $\mathsf{LinRel}_{\mathsf{k}((x))}$.

Definition 4.4. The semantics take the form of a prop morphism $[\![-]\!] : \mathsf{IH} \to \mathsf{LinRel}_{\mathsf{k}((x))}$, inductively defined by:



where 0, k, and x refer to formal Laurent series, and the action on opposite generators is defined as the relational converse.

We extend from [BSZ14, BSZ15] the guarantee that these semantics are compatible with the equational theory to the level of inequalities.

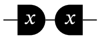
Theorem 4.5. *For any diagrams c, d in* SF*, $[\![c]\!] \subseteq [\![d]\!]$ iff $q(c) \leq q(d)$ in $\mathbb{IH}_{\mathsf{k}[x]}$.*

Proof. The $\Leftarrow$ direction is easy. For the converse, we know from Theorem 3.10 that $\mathbb{IH}_{\mathsf{k}[x]} \cong \mathsf{LinRel}_{\mathsf{k}(x)}$. $[\![-]\!]$ is the composition $\mathcal{S}_{\mathbb{IH}_{\mathsf{k}[x]}} ; [\tilde{-}]$, where $[\tilde{-}]$ is the faithful ordered prop morphism taking the subspace R in $\mathsf{LinRel}_{\mathsf{k}(x)}$ to the corresponding one in $\mathsf{LinRel}_{\mathsf{k}((x))}$ generated by every member of R viewed as a Laurent series. But the composite of faithful morphisms is still faithful. $\square$

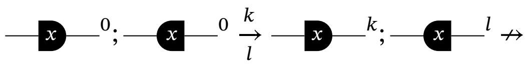
We now show that the notion of inclusion of behaviours tightly corresponds to this syntactic notion of refinement. Before we can state the precise result, we must explore the ideas of deadlock and initialisation.

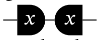
These are ways in which the operational and denotational semantics for a diagram may disagree. Deadlock leads to the operational semantics having too many behaviours compared to the denotation, and initialisation is a situation with too few. This suggests a duality between the two problems which is visible in a structural classification of diagrams displaying them.

An example of a diagram which displays deadlock behaviour is



It may reach a state from which no further transitions are possible, as in the following computation, where we make a transition labelled by $k \neq l$ at the first step.

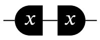


But from the denotational point of view, we cannot distinguish between this diagram and the identity —, as these semantics can be thought of as only considering infinite traces, which by definition do not deadlock. This should be expected seeing that, equationally,  = —.

Informally speaking, the problem with the above diagram and others which display deadlocking is that the signal is flowing from the outsides to the centre: that is, they are in *cospan form*. Accordingly, a diagram in *span form* is deadlock-free: no computation reaches a state from which no transition is possible.

However, putting every diagram into span form (as Lemma 3.8 guarantees we may) will not solve all our problems. Diagrams in span form, where the signal flows from the middle outwards towards the boundaries, do not suffer from deadlock, but are susceptible to *initialisation*.

Our example for this case is



Every computation of this diagram must begin with $(0, 0)$ before it continues with the same behaviour as —. Formally, we can say that a diagram is *initialisation-free* if $s_0 \xrightarrow[0\ldots0]{0\ldots0} s_1$ implies $s_0 = s_1$, where s_0 is its starting state with all the registers initialised to 0. Again, intuitively, the problem derives from the shape of the diagram.

The key result is that the denotational and operational semantics coincide for diagrams which are both deadlock- and initialisation-free; hereafter we refer to such diagrams as *well-behaved*. We can sum up this discussion with the following statement.

Proposition 4.6 ([BSZ15]). *Diagrams in span form are deadlock-free, while those in cospan form are initialisation-free. For well-behaved diagrams* c, d, $[\![c]\!] = [\![d]\!]$ *iff* $\langle c \rangle = \langle d \rangle$.

Proof. We sketch out only the proof of the second part, since we will make use of the definitions later on.

We define two dual mappings $\mathcal{F}$ and $\mathcal{U}$ which relate our denotational and observational domains: formal Laurent series, formal power series and polynomial traces.

First, we say that a trace $(\alpha, \beta) \in \mathsf{k}[[x]]^m \times \mathsf{k}[[x]]^n$ *generates* $(\sigma, \tau) \in \mathsf{k}((x))^m \times \mathsf{k}((x))^n$ if there is some $z \in \mathbb{Z}$ such that

1. $\alpha_j(i) = \sigma_j(i + z)$ and $\beta_k(i) = \tau_k(i + z)$ for all $i \in \mathbb{N}$, $1 \leq j \leq m$ and $1 \leq k \leq n$.

2. z is less than or equal to any order of $\sigma_1, \ldots, \sigma_n$ and $\tau_1, \ldots, \tau_m$.

The first condition means that the Laurent series should be the same as the power series, except possibly shifted, and the second ensures that the shift does not cause any non-zero terms to be shifted to a negative position and thereby ignored by the first clause only considering $i \in \mathbb{N}$.

A finite trace $(\alpha, \beta) \in \mathsf{k}[x]^m \times \mathsf{k}[x]^n$ of length z is a *prefix* of $(\sigma, \tau) \in \mathsf{k}[[x]]^m \times \mathsf{k}[[x]]^n$ iff $\alpha_j(i) = \sigma_j(i)$ and $\beta_h(i) = \tau_h(i)$ for all $0 \leq i \leq z$, $1 \leq j \leq m$ and $1 \leq h \leq n$.

Now, if (f, g) is a morphism of $\mathsf{LinRel}_{\mathsf{k}[x]} \times \mathsf{LinRel}_{\mathsf{k}[[x]]}$, we define $\mathcal{F}(f, g)$ as the following morphism of $\mathsf{LinRel}_{\mathsf{k}((x))}$:

$$\{(\sigma, \tau) \mid \text{there is a trace } (\alpha, \beta) \in g \text{ generating } (\sigma, \tau)\}$$

That is, the finite traces are ignored, and every possible pair of Laurent series generated by an infinite trace in g is included.

Conversely, for a morphism S of $\mathsf{LinRel}_{\mathsf{k}((x))}$, $\mathcal{U}(S)$ is (f, g) where g is given by

$$\{(\alpha, \beta) \mid \text{there exists } (\sigma, \tau) \in S \text{ generated by } (\alpha, \beta)\}$$

and f is the set of all prefixes of the traces in g.

It then follows that, for any (not necessarily well-behaved) diagram c:

1. $\mathcal{F}\langle c \rangle = [\![c]\!]$

2. If c is deadlock-free, then $\langle c \rangle \subseteq \mathcal{F}; \mathcal{U}\langle c \rangle$

3. If c is initialisation-free, then $\langle c \rangle \supseteq \mathcal{F}; \mathcal{U}\langle c \rangle$

The first of these means that whenever $\langle c \rangle = \langle d \rangle$, $[\![c]\!] = \mathcal{F}\langle c \rangle = \mathcal{F}\langle d \rangle = [\![d]\!]$. The second and third together give the equivalence when c and d are well-behaved: $[\![c]\!] = [\![d]\!]$ implies that $\langle c \rangle = \mathcal{U}(\mathcal{F}\langle c \rangle) = \mathcal{U}[\![c]\!] = \mathcal{U}[\![d]\!] = \mathcal{U}(\mathcal{F}\langle d \rangle) = \langle d \rangle$. □

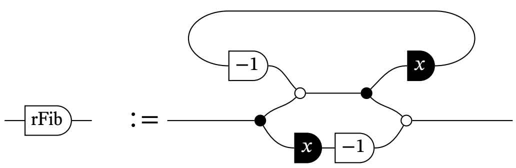
Now, by observing that $\mathcal{F}$ and $\mathcal{U}$ are monotone, we may infer that the equivalence of the semantics holds also in the case of *inclusion*, as in:

Corollary 4.7. *If c, d are well-behaved diagrams, then $[\![c]\!] \subseteq [\![d]\!]$ iff $\langle c \rangle \subseteq \langle d \rangle$.*

Then, by Theorem 4.5, we have the main result of this section: the correspondence between syntactic refinement and inclusion of behaviours.

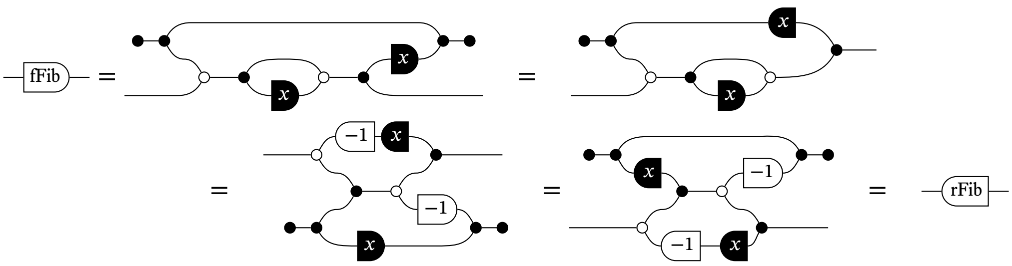
Theorem 4.8. *For any well-behaved diagrams c, d, $\langle c \rangle \subseteq \langle d \rangle$ iff $c \leq d$ in $\mathbb{IH}_{\mathsf{k}[x]}$.*

This result allows us to show that our rabbit breeding SFG —fFib— has an inverse —rFib—.



(4.3)

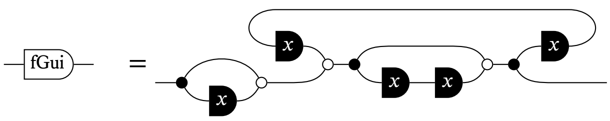
Reflection in a vertical axis corresponds to relational converse. If both relations are actually functions, then that corresponds to the functional inverse. So to show they are inverse, it is enough to show that —fFib— is equal to the ‘mirror image’ version of —rFib—, as every diagram of SF is guaranteed to be a function. The reasoning that they are inverse proceeds as follows:



(4.4)

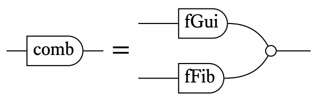
where we use the reflected notation to illustrate that the string diagram as a whole is reflected. This inverse gives a recipe for the number of animals our hypothetical rabbit breeder should buy or sell to maintain a stable rabbit population.

However, not all SFGs admit an inverse. To illustrate this point, we introduce the SFG for guinea pigs, which (for our purposes) breed similarly to rabbits except that they require two months to reach maturity. The SFG representing this is



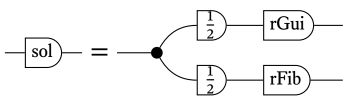
Individually we can invert this; we denote the inverse as —rGui—.

Now if we have a farm consisting of both rabbits and guinea pigs, the total number of pairs of animals is modelled by the following:



(4.5)

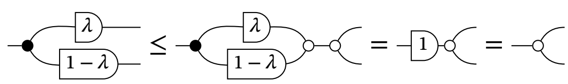
which is not invertible. However, there are still strategies our rabbit breeder can use to control the population, such as keeping the same number of rabbits and guinea pigs, using each inverse individually:



(4.6)

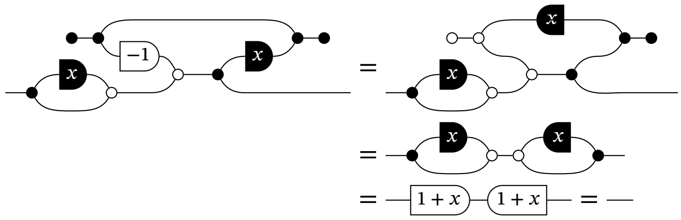
Having added an ordering to $\mathbb{IH}_{\mathsf{k}[x]}$, we can formalise the relationship between (4.5) and (4.6).

To prove that SFG —sol— from (4.6) is an inverse to —comb— of (4.5), we should show that it implements —comb—†, namely we should check that —sol— $\leq$ —comb—†. It follows from the general fact shown below; taking $\lambda = \frac{1}{2}$ gives the claimed solution.

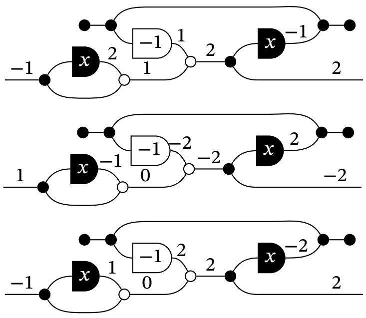


4.2 Arbitrary initialisation

We now consider the case where the registers are allowed to be initialised with arbitrary values. Perhaps surprisingly, $\mathbb{IH}_{\mathsf{k}[x]}$ is no longer sound in this generalisation. For example, the following is valid in $\mathbb{IH}_{\mathsf{k}[x]}$:



But if we view the original diagram as a SFG and initialise the left-hand register with, say, 2 and the right with −1, then with the operational semantics above, we can observe the following behaviour:



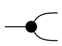
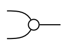
That is, an input of $-1;\ 1;\ -1;\ 1;\ -1\ \ldots$ gives an output of $2;\ -2;\ 2;\ -2;\ 2\ \ldots$ which is not a possible behaviour for —. For this reason, we have to change our equational system.

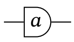
We build on the work of [FSR16], which shows that $\mathbb{IH}^{\mathsf{Cor}}_{\mathsf{k}[x,x^{-1}]}$, defined as in Definition 3.14, is a sound and complete proof system for linear time-invariant (LTI) systems, where $\mathsf{k}[x,x^{-1}]$ is the PID of polynomials in a variable and its formal inverse over a field k. In this section, we show that our notion of inequality as defined by the SMIT above is compatible with the idea of passing to a sub-behaviour.

First, we should clarify precisely what is meant by *linear time-invariant system.* In control theory, an LTI behaviour has a *domain* $(\mathsf{k}^{\mathbb{Z}})^m$, a *codomain* $(\mathsf{k}^{\mathbb{Z}})^n$ (where m, n are natural numbers) and is a linear subspace of *trajectories* $\mathcal{B} \subseteq (\mathsf{k}^{\mathbb{Z}})^m \times (\mathsf{k}^{\mathbb{Z}})^n$. $\mathcal{B}$ must satisfy two further conditions. The first is time-invariance: for every trajectory $w \in \mathcal{B}$ and fixed $i \in \mathbb{Z}$, the trajectory whose value at time t is $w(t+i)$ is also in $\mathcal{B}$. The other is known as completeness. If $t_0 \le t_1$ are integers, write $w|_{[t_0,t_1]}$ for the restriction of a behaviour $w \in \mathcal{B}$ to the set $[t_0,t_1] = \{t_0, t_0+1, \ldots, t_1\}$, and $\mathcal{B}|_{[t_0,t_1]}$ for the set of all restricted behaviours. Then $\mathcal{B}$ is complete iff whenever $w|_{[t_0,t_1]} \in \mathcal{B}|_{[t_0,t_1]}$ for all t_0, t_1, $w \in \mathcal{B}$.

Since we are now allowing trajectories to go arbitrarily far back in time, we can no longer use Laurent series as the denotational domain, but instead switch to $\mathsf{Rel}_{\mathsf{k}^{\mathbb{Z}}}$. We therefore inductively define $(\!(-)\!) : \mathsf{IH} \to \mathsf{LTI}$ by

 $\mapsto \{(\tau, *) \mid \tau \in \mathsf{k}^{\mathbb{Z}}\}$ $\mapsto \{(*, 0)\}$

 $\mapsto \left\{\left(\sigma, \begin{pmatrix}\sigma\\ \sigma\end{pmatrix}\right) \mid \sigma \in \mathsf{k}^{\mathbb{Z}}\right\}$  $\mapsto \left\{\left(\begin{pmatrix}\sigma\\ \tau\end{pmatrix}, \sigma + \tau\right) \mid \sigma, \tau \in \mathsf{k}^{\mathbb{Z}}\right\}$

 $\mapsto \{(\sigma, a\sigma) \mid \sigma \in \mathsf{k}^{\mathbb{Z}}\}$  $\mapsto \{(\sigma, x\sigma) \mid \sigma \in \mathsf{k}^{\mathbb{Z}}\}$

where the addition and scalar multiplication act pointwise, and x acts on a biinfinite stream by delaying it by one step, so that $x\sigma(t) = \sigma(t-1)$. As usual, the map on the other generators is the relational converse of the mirror images.

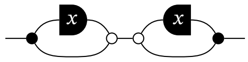
Definition 4.9. The prop LTI is the sub-prop of $\mathsf{Rel}_{\mathsf{k}^{\mathbb{Z}}}$ whose morphisms $m \to n$ are those behaviours with domain $(\mathsf{k}^{\mathbb{Z}})^m$ and codomain $(\mathsf{k}^{\mathbb{Z}})^n$.

There is a full and faithful functor $\Phi : \mathsf{Corel}(\mathsf{Mat}_{\mathsf{R}}) \to \mathsf{LTI}$, defined by $\mathsf{R}^m \xrightarrow{A} \mathsf{R}^z \xleftarrow{B} \mathsf{R}^n \mapsto \ker[A \;\; -B]$. A result of systems theory from Willems [Wil86] is key.

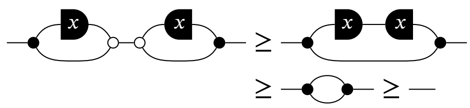
Theorem 4.10 (Willems). *If M, N are matrices over $\mathsf{k}[x, x^{-1}]$, then $\ker M \subseteq \ker N$ iff there is a matrix X such that $XM = N$.*

We use this to show that the isomorphism of [FSR16] also extends to an ordered prop isomorphism. Soundness is again trivial. Given corelations $\xrightarrow{A}\xleftarrow{B} \le \xrightarrow{C}\xleftarrow{D}$, there is some matrix U such that $C = UA$ and $D = UB$. By the theorem, $\ker A \subseteq \ker C$ and $\ker B \subseteq \ker D$, and then it is simple algebra to show that $\Phi(\xrightarrow{A}\xleftarrow{B}) = \ker[A \;\; -B] \subseteq \ker[C \;\; -D] = \Phi(\xrightarrow{C}\xleftarrow{D})$.

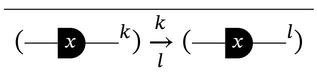
Let's now return to the earlier example, a signal flow graph equal to the following string diagram.



Unlike the simple case of $\mathbb{IH}_{\mathsf{k}[x]}$ this is not equal to the identity —. We can, however, prove that the identity behaviour is *contained* in its behaviours:



Similarly to before, we can define operational semantics on SFGs. In fact, the rules are again those from Fig. 4.1, but we introduce two new ideas: *register assignments* and *reverse computations*. A register assignment is simply what one would expect, associating to each string diagram c a *state* as above, where each delay component has some (arbitrary) value of k associated with it. A reverse computation is one where we use the same rules as before, but we swap the left and right sides of the rule for delay:



We can now define a *biinfinite trajectory* to complement our notions of finite and infinite trajectory.

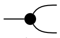
Definition 4.11. Given a SFG $c : m \to n$, a *biinfinite trajectory* consists of $w \in (\mathsf{k}^m)^{\mathbb{Z}} \times (\mathsf{k}^n)^{\mathbb{Z}}$, along with a register assignment σ, infinite forward and backward trajectories $(\boldsymbol{\alpha}, \boldsymbol{\beta})$ and $(\boldsymbol{\alpha}', \boldsymbol{\beta}')$ respectively, each initialised at σ, such that

$$w(t) = \begin{cases} (\boldsymbol{\alpha}(t), \boldsymbol{\beta}(t)) & t \geq 0 \\ (\boldsymbol{\alpha}'(-t), \boldsymbol{\beta}'(-t)) & t \leq 0 \end{cases}$$

The collection of all such trajectories of c is written as $bt(c)$.

A straightforward structural induction gives the following; in particular, SFGs represent LTI behaviours.

Proposition 4.12. *For any SFG c, $(\!|c|\!) = bt(c)$.*

Proof. The check is routine for each generator; for example, it's clear that any biinfinite stream is admissible as input into  and that the behaviour of addition matches up between the denotation defined by $(\!|-|\!)$ and the operational semantics of Figure 4.1 and Definition 4.11. It is also obvious that both notions of composition defined in the operational semantics are the same as the relational ones arising from the inductive definition of $(\!|-|\!)$. □

Just like before, string diagrams $\mathbb{IH}^{\mathsf{Cor}}_{\mathsf{k}[x,x^{-1}]}$ can be thought of as specifications, and SFGs as implementations. Once again, by Theorem 4.5, we can use refinement as a definition of correct implementation.

Chapter 5

Petri nets

This chapter parallels Chapter 4 in taking a model of concurrent systems and turning instances into string diagrams of a SMIT in a way which allows inequational reasoning which respects behaviour. That model is the Petri net[1], an invention of Carl Adam Petri[2], who wished to describe chemical processes. They are now used in diverse fields such as business process modelling, concurrent programming, manufacturing systems analysis and web services.

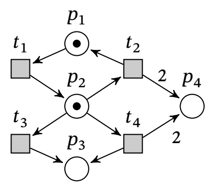
First, we give a formal definition of a Petri net, and explain the intuition it captures. We then have to take a theoretical diversion, since our semantic domain shifts from linear relations to *additive relations*, replacing the arbitrary PID with the semiring $\mathbb{N}$. This requires some changes to the SMITs we consider, and we show examples of diagrams which must be handled carefully to recover the distinctive properties of $\mathbb{N}$.

Having established the necessary facts about AddRel, we give the definition of the SMIT we will use to embed a representation of Petri nets. The main contribution from this chapter is extending the isomorphism between syntactic prop $\mathbb{RC}$ and AddRel for traditional (non-ordered) props to cover inequalities for ordered props.

Definition 5.1. A *Petri net* $\mathcal{P} = (P, T, {}^\circ{-}, {-}^\circ)$ consists of a finite set of places P, a finite set of transitions T, and functions ${}^\circ{-}, {-}^\circ : T \to \mathbb{N}^P$. Given *markings* $\mathbf{a}, \mathbf{b} \in \mathbb{N}^P$, we write $\mathbf{a} \to \mathbf{b}$ if there exists $\mathbf{t} \in T$ such that ${}^\circ\mathbf{t} \leq \mathbf{a}$ (pointwise) and $\mathbf{b} = \mathbf{a} - {}^\circ\mathbf{t} + \mathbf{t}^\circ$. The (step) operational semantics of $\mathcal{P}$ consist of the relation $[\![\mathcal{P}]\!] = \{(\mathbf{a}, \mathbf{b}) \mid \mathbf{a} \to \mathbf{b}\} \subseteq \mathbb{N}^P \times \mathbb{N}^P$.

Intuitively speaking, a Petri net consists of places, each of which may contain some number of *tokens* summarised by the marking from $\mathbb{N}^P$, and transitions, which have pre- and post-sets labelled by ${}^\circ{-}$ and ${-}^\circ$ respectively. If the relation $\mathbf{a} \to \mathbf{b}$ holds, then it is possible, starting with tokens as in $\mathbf{a}$, for the transition witnessing the relation to 'fire' and leave the net with tokens as in $\mathbf{b}$.

Example 5.2. Below is displayed a marked Petri net with the traditional graphical notation: circles represent places, squares transitions, and dots the number of tokens present at a place. We adopt for simplicity of drawing the convention that unlabelled arrows mean that the multiplicity of the transition is 1 at that place.

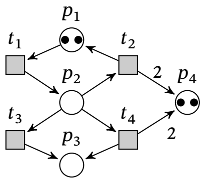


---

[1] We mean the notion also referred to in the literature as P/T (Place/Transition) net.

[2] allegedly in 1939, at the age of 13

Starting from the illustrated marking with one token at each of p_1 and p_2, and no tokens anywhere else, t_2 may fire, consuming the token at p_2 and creating new ones at p_1 and p_4:



With this new marking, only t_1 may fire, since no other transition has the tokens required by its pre-set.

Since negative and fractional tokens do not make sense, we need to shift our attention from abstract rings and fields to the semiring of natural numbers: from linear relations to *additive relations*.

5.1 Additive relations

Definition 5.3. An *additive relation* (over $\mathbb{N}$) of type $k \to l$ is a subset $R \subseteq \mathbb{N}^k \times \mathbb{N}^l$ such that

- $(\mathbf{0}, \mathbf{0}) \in R$
- if $(\mathbf{a}, \mathbf{b})$, $(\mathbf{a}', \mathbf{b}') \in R$ then $(\mathbf{a} + \mathbf{a}',\ \mathbf{b} + \mathbf{b}') \in R$

If $R, R' : k \to l$ are additive relations of the same type, then both the intersection $R \cap R'$ and the Minkowski sum $R + R' = \{(\mathbf{a} + \mathbf{a}', \mathbf{b} + \mathbf{b}') \mid (\mathbf{a}, \mathbf{b}) \in R \text{ and } (\mathbf{a}', \mathbf{b}') \in R'\}$ are additive relations. Every pair $(\mathbf{a}, \mathbf{b}) \in \mathbb{N}^k \times \mathbb{N}^l$ generates an additive relation $\langle(\mathbf{a}, \mathbf{b})\rangle = \{(p\mathbf{a}, p\mathbf{b}) \mid p \in \mathbb{N}\}$. More generally, for a finite set $G = \{(\mathbf{a}_1, \mathbf{b}_1), \dots, (\mathbf{a}_p, \mathbf{b}_p)\}$ of points in $\mathbb{N}^k \times \mathbb{N}^l$, we use a shorthand notation for the Minkowski sum

$$\langle G \rangle = \langle(\mathbf{a}_1, \mathbf{b}_1), \dots, (\mathbf{a}_p, \mathbf{b}_p)\rangle = \langle(\mathbf{a}_1, \mathbf{b}_1)\rangle + \cdots + \langle(\mathbf{a}_p, \mathbf{b}_p)\rangle$$

We are interested only in those additive relations which can be written in the above form.

Definition 5.4. An additive relation $R : k \to l$ is *finitely generated* finitely generated if there exists a finite set of vectors G such that $R = \langle G \rangle$. G is called a *generating set* for R.

Unlike linear relations, additive relations cannot all be expressed as sets of linear combinations of a finite number of vectors: not all are finitely generated, for example $\{(m, n) \in \mathbb{N}^2 \mid m > n\} \cup \{(0, 0)\}$. Henceforward, whenever we say 'additive relation', we always mean a *finitely generated* additive relation.

It is useful to think of additive relations as forming a sub-prop AddRel of $\mathsf{Rel}_{\mathbb{N}}$ (Example 2.14). For this to make sense, we need to verify that the property of being finitely generated is closed under composition and monoidal product. The monoidal product is straightforward: if $R : k \to l$ and $R' : k' \to l'$ are finitely generated additive relations with generating sets $\{(\mathbf{a}_1, \mathbf{b}_1), \cdots, (\mathbf{a}_p, \mathbf{b}_p)\}$ and $\{(\mathbf{a}'_1, \mathbf{b}'_1), \cdots, (\mathbf{a}'_q, \mathbf{b}'_q)\}$ respectively, $R \oplus R'$ has generating set $\left\{\left(\begin{pmatrix}\mathbf{a}_i \\ \mathbf{a}'_j\end{pmatrix}, \begin{pmatrix}\mathbf{b}_i \\ \mathbf{b}'_j\end{pmatrix}\right) \mid 1 \le i \le p \text{ and } 1 \le j \le q\right\}$. The case of composition also goes through, but it is non-trivial: other semirings, like the tropical semiring, do not have this closure property.

Proposition 5.5. *The composition of two finitely generated additive relations is finitely generated.*

Proof. Suppose that finitely generated additive relations $R : k \to l$ and $S : l \to m$ have respective generating sets $\{(\mathbf{a}_1, \mathbf{b}_1), \dots, (\mathbf{a}_p, \mathbf{b}_p)\}$ and $\{(\mathbf{c}_1, \mathbf{d}_1), \dots, (\mathbf{c}_q, \mathbf{d}_q)\}$. We find a generating set for $R ; S$. Let $U = \begin{pmatrix} U_k \\ U_l \end{pmatrix}$ and $V = \begin{pmatrix} V_l \\ V_m \end{pmatrix}$ be the $(k + l) \times p$ and $(l + m) \times q$ matrices whose columns are the generating vectors of R and S, respectively. By Dickson's lemma [Dic13], the set $\{(\mathbf{e},\mathbf{f}) \in \mathbb{N}^p \times \mathbb{N}^q \mid U_l\mathbf{e} = V_l\mathbf{f}\}$ has finitely many minimal elements $(\mathbf{e}_1,\mathbf{f}_1),\dots,(\mathbf{e}_d,\mathbf{f}_d)$. Then $\{(U_k\mathbf{e}_1, V_m\mathbf{f}_1),\dots,(U_k\mathbf{e}_d, V_m\mathbf{f}_d)\}$ generates $R;S$. □

We will also need some notion of *bases* of additive relations, appropriately adapted from the concept in linear algebra.

Definition 5.6. A collection of vectors in $\mathbb{N}^d$ is *dependent* if it contains an element which is expressible as a linear combination (with natural number coefficients) of the others, and *independent* otherwise. A *basis* of an additive relation is an independent generating set.

We have the following divergence from the situation in linear algebra.

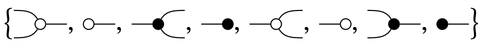
Proposition 5.7 ([Sch98]). *Every additive relation has a unique basis, called its* Hilbert basis.

Proof. Let $R \subseteq \mathbb{N}^d$ be an additive relation, and put $R^* = R \setminus \{\mathbf{0}\}$. We will show that $H = R^* \setminus (R^* + R^*)$, the set of *irreducible elements*, is a basis for R. By definition, it is independent since no irreducible vector is a sum of other vectors.

We prove that H is a generating set for R. If R is empty, then so is H and the proposition holds trivially. Suppose $\mathbf{a} \in R$. If $\mathbf{a}$ is zero or a member of H, then clearly $\mathbf{a} \in \langle H\rangle$, so assume $\mathbf{a} \in R^* + R^*$. Then there are $\mathbf{b}$, $\mathbf{c} \in R^*$ such that $\mathbf{a} = \mathbf{b} + \mathbf{c}$. Writing $|\mathbf{a}|$ for the L_1 norm of $\mathbf{a}$, we must have $|\mathbf{a}| > |\mathbf{b}|$ and $|\mathbf{a}| > |\mathbf{c}|$. Now, either $\mathbf{b}$ and $\mathbf{c}$ are both irreducible, or at least one of them may be further decomposed into other elements with decreased norm. There is no infinite decreasing sequence in $\mathbb{N}^d$, so this decomposition process must terminate to give the element $\mathbf{a}$ as a sum of irreducibles from H.

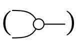
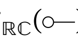
Finally for uniqueness, any generating set must contain H because there is no way to write an irreducible element as a sum. □

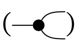
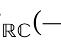
To present AddRel with a SMT (and indeed a SMIT), we will use the generators

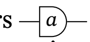
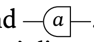


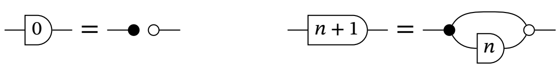
(5.1)

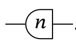
to generate a prop $\mathbb{RC}$ (for resource calculus) with the same interpretation of adding and copying as in $\mathbb{IH}$. To be formal, we will use the map $\mathcal{S}_{\mathbb{RC}} : \mathbb{RC} \to \mathsf{AddRel}$ defined inductively exactly as $\mathcal{S}_{\mathbb{IH}}$ was:

$\mathcal{S}_{\mathbb{RC}}($$) = \left\{\left(\begin{pmatrix}a\\b\end{pmatrix}, a+b\right) \mid a,b \in \mathbb{N}\right\}$ $\qquad$ $\mathcal{S}_{\mathbb{RC}}($$) = \{(*, 0)\}$

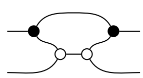
$\mathcal{S}_{\mathbb{RC}}($$) = \left\{\left(a, \begin{pmatrix}a\\a\end{pmatrix}\right) \mid a \in \mathbb{N}\right\}$ $\qquad$ $\mathcal{S}_{\mathbb{RC}}($$) = \{(a, *) \mid a \in \mathbb{N}\}$

along with the relational converse for the mirror image generators. One may wonder why we have not included the scalar generators  and . In fact, since we are working over the natural numbers, we can inductively define 'syntactic sugar' diagrams which serve the same purpose as follows:



and the mirror image versions for . Showing that these sugars satisfy the usual addition and multiplication rules for natural numbers amounts to showing that the corresponding equations in $\mathbb{IH}_{\mathsf{R}}$ are sound when applied to the image of the naturals in R.

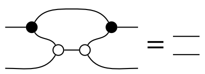
Example 5.8. There are two examples worth highlighting which play an important role in the axiomatisation given below. First,



which $\mathcal{S}_{\mathbb{RC}}$ maps to

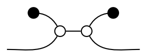
$$\left\{\left(\begin{pmatrix}a\\b\end{pmatrix},\begin{pmatrix}c\\d\end{pmatrix}\right) \mid a = c \text{ and } a + b = c + d\right\}.$$

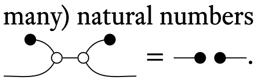
Substituting a for c and using the cancellativity of $\mathbb{N}$, $a + b = a + d$ implies $b = d$. So this relation is equal to the identity relation on $\mathbb{N}^2$. In other words, we should have



This is a form of 'controlled subtraction' analogous to the Peano axiom $Sm = Sn \implies m = n$. Unrestricted subtraction is impossible in the naturals, without negative numbers, but this equation allows subtraction when we can supply 'evidence' that the result is non-negative.

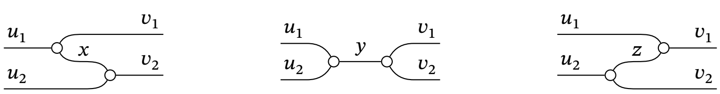
Next, we have



whose image is $\{(a, b) \mid$ there exist c, d such that $a+c = b+d\}$. Clearly, any pair of naturals has (infinitely many) natural numbers greater or equal to both. It follows that the relation is total; we therefore desire . This encodes the Archimedean property of the naturals: given any two natural numbers, there exists another one larger than both of them.

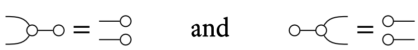
Definition 5.9. The prop $\mathbb{RC}$ (or *resource calculus*) is that corresponding to the SMIT with the generators of (5.1) and the equations and inequality of Figure 5.1.

$\mathbb{RC}$ deviates from the theory of linear relations: the white monoid-comonoid pair forms a special bimonoid, not a Frobenius monoid. Here, the Frobenius structure—if present—would play the role of assuming the presence of additive inverses [CPV12, BPS17]. Consider the diagrams appearing Frobenius equation itself, shown below with some annotations[3]:



Diagrams are interpreted as relations of vectors, and the white structure represents addition. Therefore, putting this together with our definition for composition, we can translate the left hand 'S'-shaped diagram into the relation $\left\{\left(\begin{pmatrix}u_1\\u_2\end{pmatrix},\begin{pmatrix}u_1\\u_2\end{pmatrix}\right) \mid \text{ there is } x \in \mathbb{N} \text{ such that } u_1 = v_1 + x \text{ and } x + u_2 = v_2\right\}$. The middle 'X'-shaped diagram gives us the single equation (eliminating y) $u_1 + u_2 = v_1 + v_2$, and the right-hand 'Z'-shaped one gives $u_1 + z = v_1$ and $u_2 = z + v_2$. It's tempting to automatically rearrange and solve, but remember that we are working over the natural numbers, and therefore negative numbers are not allowed. Indeed, consider assigning $u_1 = v_2 = 1$ and $u_2 = v_1 = 0$. Clearly, this satisfies the conditions for the middle diagram, and putting $x = 1$ satisfies the left one too. However, the only way to solve the right equations would be to have $z = -1$ but this is not a natural number. Swapping over the assignment shows that we can't even have an inequality: these diagrams are all just incomparable as relations, and there is no hope of recovering any kind of Frobenius structure.

On the other hand, the bimonoid laws include



[3]These annotations are *not* wire counts as they have been previously.

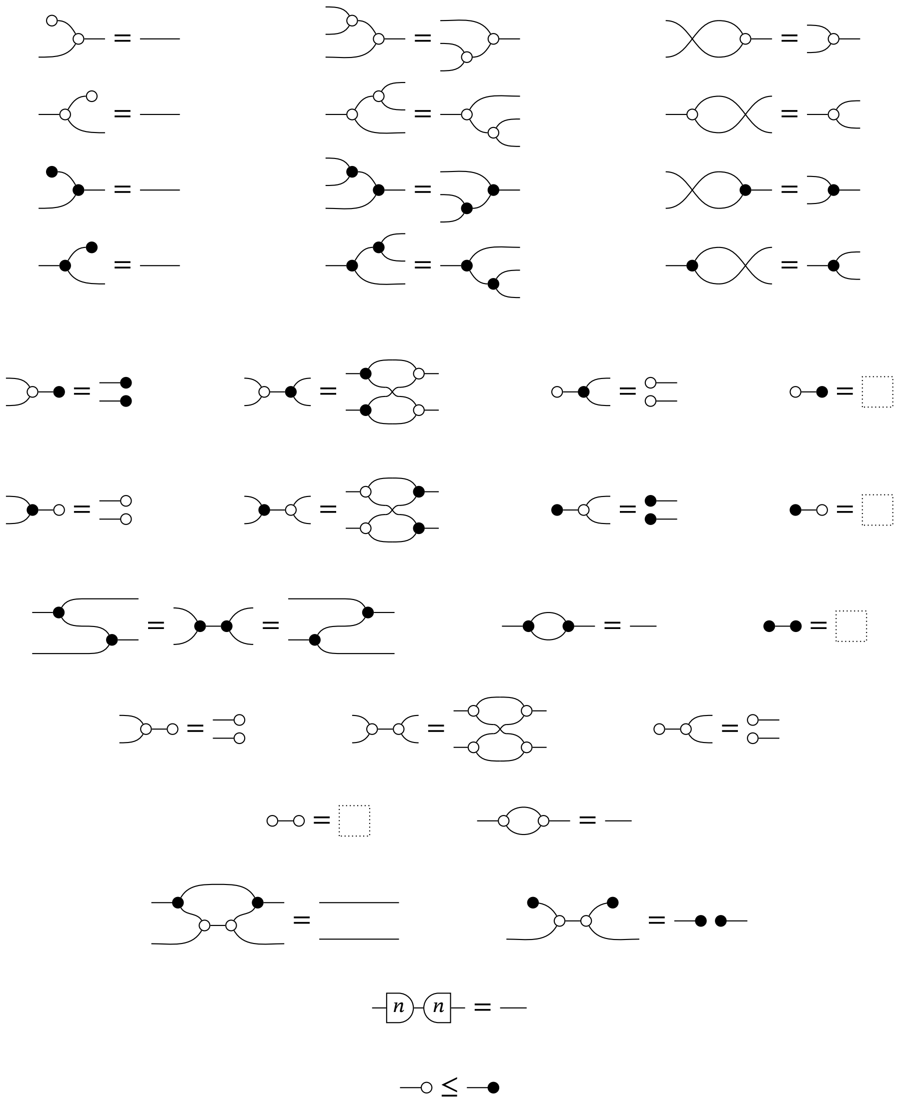
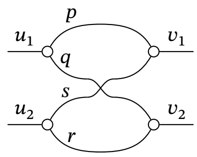


Figure 5.1: Axioms of the resource calculus. n ranges over non-zero natural numbers.

which reflect non-negativity of the natural numbers: $a + b = 0 \implies a = b = 0$. The other bimonoid law is a little more involved:



This translates into requiring the existence of $p, q, r, s \in \mathbb{N}$ satisfying four equations:

$$u_1 = p + q \qquad v_1 = p + s$$
$$u_2 = s + r \qquad v_2 = q + r$$

This should be equivalent to the 'X'-shaped equation in the centre above. One direction is obvious: if we have such p, q, r, s then we can just put $y = p + q + r + s$. On the other hand, if $u_1 + u_2 = v_1 + v_2$, then it is easy to see that when $u_2 \geq v_2$ then setting $p = u_1, q = 0, r = v_2$, and $s = u_2 - v_2$ satisfies the equations. If $v_2 \geq u_2$ then we can symmetrically put $p = v_1, q = v_2 - u_2, r = u_2$, and $s = 0$.

The last equation is an axiom scheme, parameterised over $n \in \mathbb{N}$. It uses the syntactic sugar for naturals we defined earlier, along with the obvious mirror image versions, to represent the additive relations of the form $\langle(1, n)\rangle$ and $\langle(n, 1)\rangle$. We again diverge from the case of linear relations in omitting its symmetric variant since it is not sound for AddRel and relies on the ability to *divide* by non-zero scalars.

Proposition 5.10. $\mathcal{S}_{\mathbb{RC}} : \mathbb{RC} \to \mathsf{AddRel}$ *is a morphism of ordered props.*

Proof. We must verify that the equations and the single inequality of Figure 5.1 are sound. Those involving only the black structure go through exactly as for $\mathbb{IH}$, as do the laws for the black-white bimonoids (since they encode statements like 'copying then adding is the same as adding then copying'). The only one remaining not accounted for in Example 5.8 and the above discussion is $-\boxed{n}\!-\!\boxed{n}- \;=\; -$. But, for a positive natural n,

$$\mathcal{S}_{\mathbb{RC}}(-\boxed{n}\!-\!\boxed{n}-) = \{(x, y) \mid nx = ny\}$$

which is identity because $n \neq 0$.

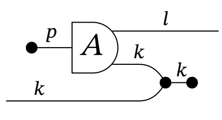
Just as in $\mathbb{IH}$, $\mathcal{S}_{\mathbb{RC}}(-\!\circ) = \{0\} \subseteq \mathbb{N} = \mathcal{S}_{\mathbb{RC}}(-\!\bullet)$. □

5.2 Completeness of additive relations

We can go further and show that $\mathcal{S}_{\mathbb{RC}}$ is actually an isomorphism of ordered props. In other words, Figure 5.1 constitutes a sound and fully complete axiomatisation of additive relations. We begin with stating the result proved in a paper at POPL:

Theorem 5.11 ([BHP+19]). $\mathcal{S}_{\mathbb{RC}} : \mathbb{RC} \to \mathsf{AddRel}$ *is an isomorphism of props.*

The proof, omitted here, establishes as a normal form N_A the diagram



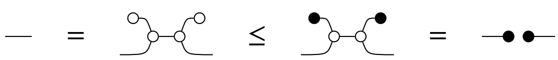
where A is the *generating matrix* of an additive relation. Informally speaking, the black counit on the left 'universally quantifies' over elements of $\mathbb{N}^p$ and we use the compact closed structure to bend the wire around to the left.

With the prop isomorphism established, we may now consider the ordering and show the full theorem:

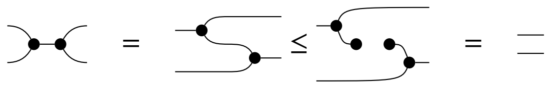
Theorem 5.12. $\mathbb{RC} \cong \mathsf{AddRel}$ *(as ordered props).*

Proof. First, we show that $\mathbb{RC}$ (abusing notation to refer to both the prop and the ordered prop by the same name) is a bicategory of relations (Definition 3.12). We do not need to prove every single one of the conditions; many of them are directly implied by equations of the SMIT, such as (3.11) and the first and third inequalities in (3.10). Moreover, for the third condition (3.12), it is enough by induction to check that each generator is a lax homomorphism, and many of these are equated in the SMIT.

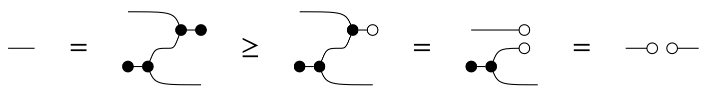
We begin with showing the various adjointness conditions. The first case of interest we tackle is $- \leq -\bullet\ \bullet-$, which we may see via the following argument:



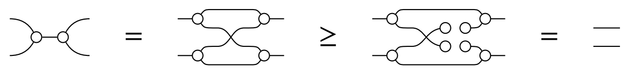
We can now use this to show the other non-trivial inequality for adjointness:



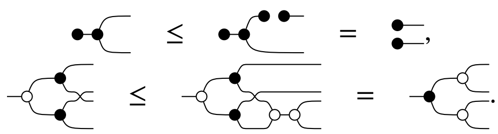
It's also useful to establish some similar adjunctions for the white structure, namely that $\dashv$ and $\circ- \dashv -\circ$. Again, the latter is key to proving the former, and only two cases are not equalities from the theory.



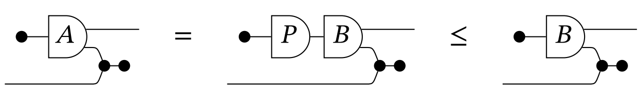
and then



We now have all the tools we need to finish the proof that $\mathbb{RC}$ is a bicategory of relations. The interesting cases for (3.12) are



Now, to show completeness of the ordering with respect to AddRel, noting that soundness of $-\circ \leq -\bullet$ is trivial, we will use the normal form. Suppose $R \subseteq S : m \to n$ are additive relations, generated by R^{m+n}-vectors $\{r_1, \ldots, r_k\}, \{s_1, \ldots, s_l\}$ respectively. Now there are $\alpha_{ij} \in \mathsf{R}$ such that for all $1 \leq j \leq k$, $r_j = \sum_{i=1}^{l} \alpha_{ij} s_i$. In other words, writing A, B for the matrices whose column vectors are the r_i and s_j respectively, there is some 'change of basis' matrix P (whose elements are α_{ij}) such that $A = BP$. Diagrammatically $-\boxed{A}- = -\boxed{P}-\boxed{B}-$. Now, in $\mathbb{RC}$, reasoning diagrammatically on the normal forms corresponding to R and S, we have (omitting types on wires)



where the final inequality follows from the definition of bicategory of relations, specifically equation (3.12). $\square$

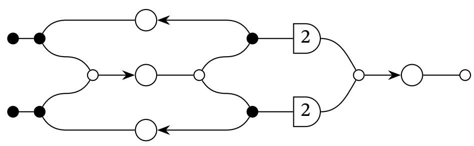
5.3 Petri nets as string diagrams

To encode Petri nets as string diagrams, we add a new generator $\rightarrow\!\bigcirc\!-\ : 1 \to 1$ to $\mathbb{RC}$. This represents a Petri net place (with a single input and output). Analogously to $-\!\boxed{x}\!-$ from SFGs, we label $\rightarrow\!\bigcirc\!-$ with a natural number (the number of tokens currently at that place) and assign this augmented generator the semantics

$$\frac{o \leq m}{(\rightarrow\!\bigcirc\!-^{m}) \xrightarrow[o]{i} (\rightarrow\!\bigcirc\!-^{m-o+i})}$$

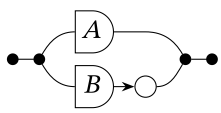
where i tokens arrive on the left and o are passed on to the right. We also use a mirrored image version $-\!\bigcirc\!\leftarrow$ to abbreviate the reflection with the black compact closed structure. The resulting prop, the quotient of this augmented syntax by the laws of $\mathbb{RC}$, is denoted Petri.

For example, the Petri net of Example 5.2 is encoded as



Multiple inputs and outputs of a place are represented by the n-ary generalisations of $\supset\!\circ\!-$ and $-\!\circ\!\subset$, and transitions are represented by black 'spiders'.

Formally, we assign to each Petri net $\mathcal{P} = (P, T, {}^\circ-, -^\circ)$ a diagram $d_\mathcal{P}$ in $\mathsf{Petri}[0,0]$. Choosing orderings on places and transitions, the pre- and post-set assignments ${}^\circ-, -^\circ\ :\ T \to \mathbb{N}^P$ can be viewed as $|P| \times |T|$ matrices over $\mathbb{N}$ A and B respectively, which, by virtue of Theorem 5.11, correspond to diagrams $-\!\boxed{A}\!-$, $-\!\boxed{B}\!-$ in $\mathbb{RC}$. $d_\mathcal{P}$ is then defined to be

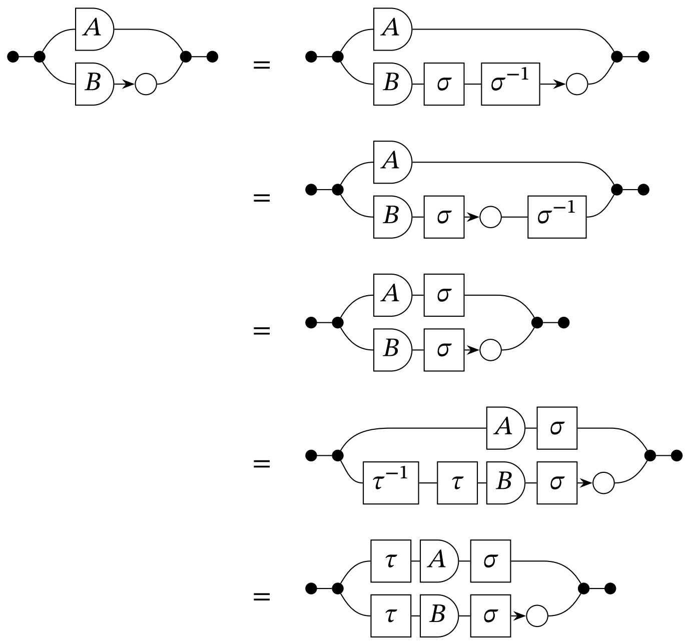


omitting (as usual) the multiplicities of the wires and similarly abbreviating $(\rightarrow\!\bigcirc\!-)^{\oplus |P|}$ as $\rightarrow\!\bigcirc\!-$.

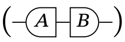
Proposition 5.13. *For every Petri net $\mathcal{P} = (P, T, {}^\circ-, -^\circ)$, $d_\mathcal{P}$ is well defined and $\mathcal{S}_{\mathbb{RC}}(d_\mathcal{P}) = [\![\mathcal{P}]\!]$.*

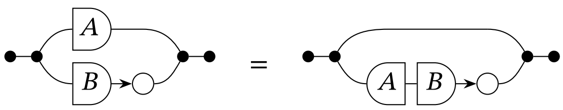
Proof. We must show that $d_\mathcal{P}$ is independent of the choices of ordering on places and transitions. If

$\sigma : |P| \to |P|$ and $\tau : |T| \to |T|$ are permutations, then



so any ordering of places gives the same diagram.

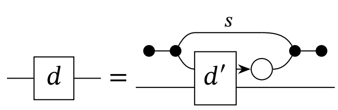
For the equivalence of semantics, note that by the definitions of Petri net semantics and of A and B we have $(\mathbf{a}, \mathbf{b}) \in [\![\mathcal{P}]\!]$ iff there is $\mathbf{f} \in \mathbb{N}^{|T|}$ such that $A\mathbf{f} \leq \mathbf{a}$ and $\mathbf{b} = \mathbf{a} - A\mathbf{f} + B\mathbf{f}$. But $\mathcal{S}_{\mathbb{RC}}$ () $= \{((A\mathbf{f}, B\mathbf{f}) \mid \mathbf{f} \in \mathbb{N}^{|T|}\}$, and it is then straightforward to calculate $\mathcal{S}_{\mathbb{RC}}(d_{\mathcal{P}})$ as equal to $[\![\mathcal{P}]\!]$ via the equality



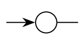
□

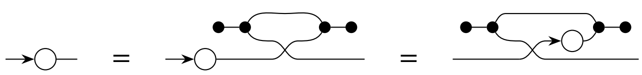
Conversely, for every diagram $d_0 \in \mathsf{Petri}[0, 0]$, we can construct a Petri net $\mathcal{P}_{d_0}$. This procedure relies on a *trace canonical form*, described in the below lemma.

Lemma 5.14. *For any diagram* $d \in \mathsf{Petri}[m, n]$*, there are a natural number* s *and* $d' \in \mathbb{RC}[m + s, n + s]$ *such that*

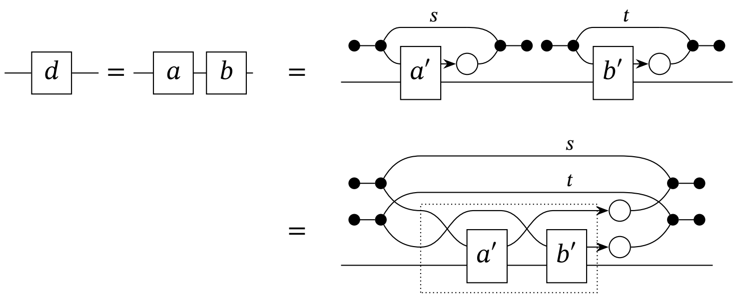


(5.2)

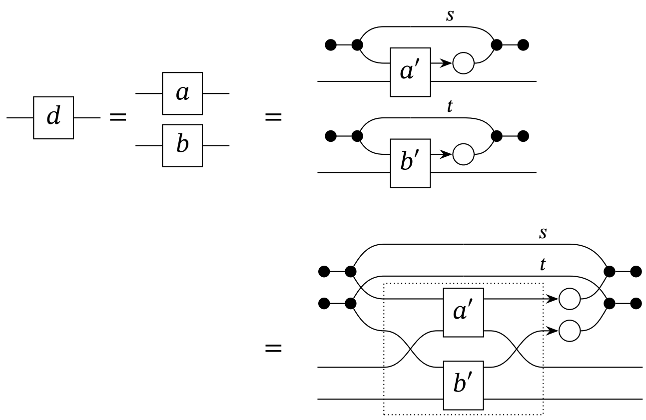
Proof. We use a straightforward structural induction on morphisms of Petri. The base cases fall into two classes: every generator of $\mathbb{RC}$ is already in trace canonical form (with $s = 0$), and for  we reason



As usual, we must consider two inductive cases, according to whether $d = a; b$ or $d = a \oplus b$. In either case, we may use the inductive hypothesis to find a' and b' in $\mathbb{RC}$ as in (5.2). For composition:



and the contents of the dotted rectangle are the required $d' \in \mathbb{RC}$. The case for $d = a \oplus b$ is similar:



and again the dotted rectangle gives an $\mathbb{RC}$ diagram to show that we have reached trace canonical form. □

Applying this to some $d \in \mathbb{RC}[0,0]$, we obtain some $d' \in \mathbb{RC}[p,p]$. By the isomorphism of Theorem 5.11, this corresponds to some additive relation, which has a Hilbert basis, say of length t. We may represent this basis as a $(p+p) \times t$ matrix $A = \begin{pmatrix} A_1 \\ A_2 \end{pmatrix}$ where A_1, A_2 are both $p \times t$ matrices. We then define $\mathcal{P}_{d_0} = (\overline{p}, \overline{t}, A_2, A_1)$.

Proposition 5.15. *For every diagram* $d \in \mathsf{Petri}[0,0]$, $[\![\mathcal{P}_d]\!] = \mathcal{S}_{\mathbb{RC}}(d)$.

Proof. By construction, $(\mathbf{a}', \mathbf{b}') \in \mathcal{S}_{\mathbb{RC}}(d')$ iff there is $\mathbf{f} \in \mathbb{N}^t$ with $A_1\mathbf{f} = \mathbf{a}'$ and $A_2\mathbf{f} = \mathbf{b}'$. □

Finally, we may combine Propositions 5.13 and 5.15 to conclude this section with a statement that string diagrams may be used to reason about Petri nets while preserving behaviour.

Theorem 5.16. *Petri nets (up to equivalence of behaviour) are in one-one correspondence with* $0 \to 0$ *morphisms of* Petri.

Chapter 6

Conclusions and Future Work

We began by exploring the landscape of symmetric monoidal theories, string diagrams, and props as applied to the analysis of concurrent systems, building up from elementary category theory to introduce the cornerstone theory of interacting Hopf algebras $\mathbb{IH}$, via a number of intermediate 'building block' theories. We also defined the more concrete props of relations which we use to represent semantics, and gave the definitions of the isomorphisms between the SMTs and their respective semantic domains.

The next chapter gave the core contribution of this thesis: the definition of the symmetric monoidal inequality theory, allowing string diagrammatic reasoning about inclusions of behaviours. We extended isomorphisms between SMTs and props to ones between SMITs and ordered props, and, along the way, solved one interesting mystery about distinguishing the black and white structure of $\mathbb{IH}$.

Having established our theories accounting for inclusion, we moved on to applying them. We used the SMITs and isomorphisms to axiomatise inclusions of behaviours for signal flow graphs, in the zero- and non-zero-initialised cases, and (one class of) Petri nets, and gave an example to show the power of proving that a signal flow graph validly implements a specification. In the case of Petri nets, since negative tokens are not allowed, we had to abandon linearity and switch to additive relations. While initially this seems like a major deviation, it turned out to require only a simple modification of our axioms; changing the white structure from a Frobenius monoid to a bimonoid and adding two extra equations was enough to maintain our isomorphisms.

However, all this barely scratches the surface of the potential of SMITs. SMTs and SMITs have the potential to be as ubiquitous as the group presentation, and their natural fit with concurrent computation is of key importance with the rise of distributed computing and the Internet of Things. For a start, it is a relatively small jump to other classes of Petri net, such as those where each place may only have a bounded number of tokens, or other semantics, such as the *banking semantics* where places may spend tokens that are arriving on the same tick. Developments building on $\mathbb{IH}$ such as 'Graphical Affine Algebra' [BPSZ19] are ripe for augmentation by inequalities.

Bibliography

[Axl97] Sheldon Axler. *Linear Algebra Done Right*. Undergraduate Texts in Mathematics. Springer New York, New York, NY, 1997.

[BHP+19] Filippo Bonchi, Joshua Holland, Robin Piedeleu, Paweł Sobociński, and Fabio Zanasi. Diagrammatic algebra: From linear to concurrent systems. In *46th ACM SIGPLAN Symposium on Principles of Programming Languages (POPL 2019)*, 2019.

[BHPS17] Filippo Bonchi, Joshua Holland, Dusko Pavlovic, and Paweł Sobociński. Refinement for signal flow graphs. In Roland Meyer and Uwe Nestmann, editors, *28th International Conference on Concurrency Theory (CONCUR 2017)*, volume 85 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 24:1–24:16, Dagstuhl, Germany, 2017. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[BPS17] Filippo Bonchi, Dusko Pavlovic, and Paweł Sobociński. Functorial semantics for relational theories. *arXiv preprint*, 2017. arXiv:1711.08699.

[BPSZ19] Filippo Bonchi, Robin Piedeleu, Paweł Sobociński, and Fabio Zanasi. Graphical affine algebra. In *2019 34th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 1–12, 2019.

[BSZ14] Filippo Bonchi, Paweł Sobociński, and Fabio Zanasi. A categorical semantics of signal flow graphs. In Paolo Baldan and Daniele Gorla, editors, *CONCUR 2014 – Concurrency Theory*, pages 435–450, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg.

[BSZ15] 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.

[BSZ17a] Filippo Bonchi, Paweł Sobociński, and Fabio Zanasi. The calculus of signal flow graphs I: Linear relations on streams. *Information and Computation*, 252:2–29, February 2017.

[BSZ17b] Filippo Bonchi, Paweł Sobociński, and Fabio Zanasi. Interacting Hopf algebras. *Journal of Pure and Applied Algebra*, 221(1):144–184, March 2017.

[Bé67] Jean Bénabou. Introduction to bicategories. In *Reports of the Midwest Category Seminar*, volume 47, pages 1–77, Berlin, Heidelberg, 1967. Springer Berlin Heidelberg.

[CF17] Brandon Coya and Brendan Fong. Corelations are the prop for extraspecial commutative Frobenius monoids. *Theory and Applications of Categories*, 32(11):380–395, November 2017.

[CPV12] Bob Coecke, Dusko Pavlovic, and Jamie Vicary. A new description of orthogonal bases. *Mathematical Structures in Computer Science*, 23(3):557–567, 2012.

[CW87] Aurelio Carboni and Robert Walters. Cartesian bicategories I. *Journal of Pure and Applied Algebra*, 49(1–2):11–32, November 1987.

[Dic13] Leonard Dickson. Finiteness of the odd perfect and primitive abundant numbers with n distinct prime factors. *American Journal of Mathematics*, 35(4):413–422, 1913.

[EML45] Samuel Eilenberg and Saunders Mac Lane. General theory of natural equivalences. *Transactions of the American Mathematical Society*, 58(2):231–294, 1945.

[FSR16] Brendan Fong, Paweł Sobociński, and Paolo Rapisarda. A categorical approach to open and interconnected dynamical systems. In *Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science*, LICS '16, page 495–504, New York, NY, USA, 2016. Association for Computing Machinery.

[FZ17] Brendan Fong and Fabio Zanasi. A universal construction for (co)relations. In Filippo Bonchi and Barbara König, editors, *7th Conference on Algebra and Coalgebra in Computer Science (CALCO 2017)*, volume 72 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 12:1–12:16, Dagstuhl, Germany, 2017. Schloss Dagstuhl–Leibniz-Zentrum für Informatik.

[GLK55] Arthur Guyton, Arthur Lindsey, and Berwind Kaufmann. Effect of mean circulatory filling pressure and other peripheral circulatory factors on cardiac output. *American Journal of Physiology*, 180(3):463–468, 1955.

[JS91] André Joyal and Ross Street. The geometry of tensor calculus, I. *Advances in Mathematics*, 88(1):55–112, July 1991.

[KL80] Max Kelly and Miguel Laplaza. Coherence for compact closed categories. *Journal of Pure and Applied Algebra*, 19:193–213, December 1980.

[Lac04] Stephen Lack. Composing PROPs. *Theory and Applications of Categories*, 13(9):147–163, 2004.

[Lat98] B. P. Lathi. *Signal Processing & Linear Systems*. Oxford University Press, 1998.

[Mas52] Samuel Mason. *On the logic of feedback*. PhD thesis, Massachusetts Institute of Technology, 1952.

[ML65] Saunders Mac Lane. Categorical algebra. *Bulletin of the American Mathematical Society*, 71:40–106, 1965.

[Pet77] James L. Peterson. Petri nets. *ACM Comput. Surv.*, 9(3):223–252, September 1977.

[Sch98] Alexander Schrijver. *Theory of linear and integer programming*. John Wiley & Sons, 1998.

[Sel11] Peter Selinger. A survey of graphical languages for monoidal categories. In Bob Coecke, editor, *New Structures for Physics*, pages 289–355. Springer Berlin Heidelberg, 2011.

[Sha42] Claude Shannon. The theory and design of linear differential equation machines. Technical Report 411, National Defence Research Council, January 1942.

[Wil86] Jan Willems. From time series to linear system — Part I. Finite dimensional linear time invariant systems. *Automatica*, 22(5):561–580, 1986.

[Zan15] Fabio Zanasi. *Interacting Hopf Algebras: the theory of linear systems*. PhD thesis, École Normale Supérieure de Lyon, 2015.