Picturing Resources in Concurrency

Robin Piedeleu

Balliol College

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

Michaelmas 2018

Abstract

Inspired by the pioneering work of Petri and the rise of diagrammatic formalisms to reason about networks of open systems, we introduce the resource calculus—a graphical language for distributed systems. Like process algebras, the resource calculus is modular, with primitive connectors from which all diagrams can be built. We characterise its equational theory by proving a full completeness result for an interpretation in the symmetric monoidal category of additive relations—a result that constitutes the main contribution of this thesis.

Additive relations are frequently exploited by model-checking algorithms for Petri nets. In this thesis, we recognise them as a fundamental algebraic structure of concurrency and use them as an axiomatic framework. Surprisingly, the resource calculus has the same syntax as that of interacting Hopf algebras, a diagrammatic formalism for linear (time-invariant dynamical) systems. Indeed, the approach stems from the simple but fruitful realisation that, by replacing values in a field with values in the semiring of non-negative integers, concurrent behaviour patterns emerge. This change of model reflects the interpretation of diagrams as systems manipulating limited and discrete resources instead of continuous signals.

We also extend the resource calculus in two orthogonal directions. First, by adding an affine primitive to express access to a constant quantity of resources. The extended calculus is remarkably expressive and allows the formulation of non-additive patterns of behaviour, like mutual exclusion. Once more, we characterise it—this time as the equational theory of the symmetric monoidal category of polyhedral relations, discrete analogues of polyhedra in convex geometry. Secondly, we add a synchronous register to model stateful systems. The stateful resource calculus is expressive enough to faithfully capture the behaviour of Petri nets while being strictly more expressive. It is also shown to axiomatise a category of open Petri nets, in the style of the connector algebras of nets with boundaries first studied by Bruni, Melgratti, Montanari and Sobociński.

Contents

List of Figures

Chapter 1

Introduction

1.1 Motivation

This thesis is an effort in the pursuit of canonical algebraic structures in concurrency theory. Before delving into the technical aspects of our work, we would like to provide some motivation for our contribution and a short summary of our guiding principles—a manifesto of sorts.

In the broadest sense, computer science is concerned with generating controlled behaviour from dynamical systems. Typically, the observable behaviour and state space of the underlying physical process is coarse-grained into an—often discrete—collection of events corresponding to state-changes of interest to the observer.

Initially, theoretical computer science studied mostly extensional phenomena, that is, the extent to which certain functions on common mathematical objects could be computed. The first models of computation were models of *computability*, limited to describing the properties of algorithmic processes through which a system transforms inputs into outputs in finite time. The implementation describes exactly how to stir a single idealised dynamical system from a start state to a final state, to obtain the desired result. In this rather restricted sense, the meaning of a computation is a partial function (often just $\mathbb{N} \to \mathbb{N}$). The sequential behaviour of the implementation mattered only to control the amount of resources (typically, space and time) that the process needed to successfully complete its task.

As the use of computers became more widespread, and with the rise of networks and distributed computation, thinking of computers as isolated systems failed to capture the rich tapestry of *behaviours* that they could exhibit. Increasingly, the meaning of a computation had to take into account the interaction of the computer with its environment. This includes its interaction with human agents whose use of computers had changed too. In the early days of computing, users were a limited pool of programmers and scientists, whose role often involved waiting patiently for the result of long calculations. As the circle of users widened, so did their use: they are now constantly reacting and responding to the machine's behaviour, controlling the dynamics of the system through designated input and output channels[1]. At the time of writing of this thesis, the overwhelming majority of computations do not implement a function in the traditional mathematical sense. This does not mean that one cannot encode the activity of computers as functions on the natural numbers, but this would be absurdly reductionistic and miss the intended meaning of the computation. As Samson Abramsky asks somewhat facetiously "What function does the internet compute?"[Abr06]. We could ask the same question of most modern computer applications, from operating systems to social-media websites. All of these examples are not meant to terminate with a single output (if they do, it is only when they crash). It is their interactive behaviour that is of interest.

As a result, models of computation were cut off from the safe Platonic realm of conventional mathematical structures. Computer scientists were faced with the challenge of integrating the way in which spatially-distributed systems communicate, interact with each other or with their users, and share resources, all within a consistent formalism. In this new setting, the usual notions of expressiveness, like Turing completeness, while still applicable, lost some of their relevance as an universal yardstick. New paradigms had to be developed. If *process calculi* were successful in providing versatile tools to describe complex systems from simple connectors, they were limited by the idiosyncratic choices of their syntax. Despite Robin Milner's quest for the λ-calculus of concurrency, no existing language can claim to have identified the right primitives to specify and reason about the behaviour of concurrent processes. Instead, there is an abundance of calculi, with sometimes incomparable expressive power and no clear unifying picture. Different paradigms offer toolkits for different purposes, while no definite theory of concurrency has prevailed. In the words of Samson Abramsky, we still do not know what the fundamental structures of concurrency are [Abr06].

This is not to say there are no grounding concepts of theoretical importance in distributed computing: starting with the insights of Petri, the field has sought to carve out fundamental notions, such as causality, nondeterminism, (a)synchrony and concurrency, independently of any particular syntax. Petri nets provide an intuitive graphical formalism in which to study the behaviour of spatially distributed systems

[1]The logician Jean-Yves Girard credits computer science for restoring the central place of the first-person subject in logic [Gir06]. In this sense, computers do not compute—we do, using them.

and processes. Petri himself described his nets as a kind of theoretical framework for discrete physics, an idea that has been revitalised by the emerging connections between quantum physics and computation. However, in contrast to the modular approach of process calculi, Petri nets are monolithic objects, describing closed systems with no obvious way of integrating their interactions with an environment.

One of the guiding lines of this work is that, insofar as the behaviour of computers can only be understood in reference to how they interact with an environment, they have to be treated as *open systems*. Here, the environment denotes any external system whose behaviour is uncontrolled, whether it is a human agent or other computers, inside a network. Correspondingly, mathematical models of computation cannot fully isolate the system whose behaviour they attempt to describe. This goes against the very successful tradition in the natural sciences of modelling physical systems in isolation. If this approach has proved unreasonably effective, especially in physics, it seems that the rise of information technology has required us to develop mathematical formalisms to analyse the dynamic behaviour of open and interconnected systems. Computer science is not the only field forcing a paradigm shift—to some extent, biology and the social sciences require a similar change of tools and perspective.

An approach to open systems that has been particularly influential in concurrency theory is that best summarised by Robin Milner's words: "The meaning of a program should reflect its history of access to resources that are not local to it" [Mil75]. Resources have always played a central role in distributed systems. Processes running concurrently may share access to resources such as network bandwidth, memory or processor cycles. The importance of coordinating mechanisms was recognised as early as the 1960s and 1970s in the work of Dijkstra, Hoare, Lamport and many others [Dij68, Hoa72, Lam74]. But the notion of resource remained informal, dictated by the practical concerns of particular applications. Later, formalisms like separation logic [Rey02] and the logic of bunched implications [OP99] recognised the foundational role of resource management in concurrent programming and built axiomatic frameworks around it to reason about concurrent processes that manipulate shared state. They were not the first resource-conscious logics. The pioneering work of Girard on linear logic [Gir87] paved the way for the development and application of substructural logic in computer science. In this case, substructural means that the usual structural rules of contraction, weakening and exchange are either missing or, at least, controlled more closely than in classical or intuitionistic logic.

For sequential computation, category theory provides a powerful unifying framework. One of the fundamental links between theoretical computer science and category theory is the Curry-Howard-Lambek isomorphism. Informally, it establishes a tight correspondence between natural deduction proofs, programs or terms in the simply-typed λ-calculus, and morphisms in Cartesian closed categories. However the correspondence does not extend to the concurrent setting. From the syntactic side—as we claimed above—the λ-calculus, which is based on sequential function composition, is ill-equipped to deal with concurrency. One of the reasons is that it is not resource-sensitive. In logical terms, intuitionistic logic allows for unlimited use of contraction and weakening; in categorical terms, Cartesian closed categories come equipped with a natural comonoid (i.e., copying) structure on each object. Some researchers have argued that the simpler setting of monoidal categories, with resource-conscious sequential and parallel composition as primitive operations, are better suited to represent concurrent phenomena [AGN96, KSW97b]. We also start from the premise that monoidal categories have something to say about concurrency and that, conversely, we need to abandon Cartesian assumptions to study concurrent phenomena.

Morphisms in monoidal categories can be depicted as *string diagrams*, a two-dimensional syntax that takes into account the intrinsic spatial structure of distributed systems, highlighting physical features such as connectivity and resource-sharing. This makes them a convenient framework in which to explore mathematical models of open systems, resting on firm algebraic foundations. A growing body of work exploits these appealing features to reason about—to cite a few examples—coordination in distributed systems [BLM06], networks of communicating automata [ASW09], the synthesis and verification of electronic circuits [Ghi13], the behaviour of (passive linear) electric circuits [BF15], equilibrium in open Markov processes [BFP16], stochastic reaction networks [BP17], quantum computation and foundations [CK17], and even distributional models of meaning in linguistics [CSC10]. Sometimes, pre-existing diagrammatic formalisms have been profitably revisited from this point of view, like Petri nets and signal flow graphs [Sha42], to mention two examples that will feature prominently in this thesis.

One of the advantages of specifying and verifying properties of distributed systems in the language of (monoidal) category theory is that we are able to do so *compositionally*: that is, deriving properties of systems by combining those of simpler sub-systems using sequential and parallel composition. This is know as the *principle of composi-*

tionality. All of the cited works rely heavily on this principle and it will constitute one of the pillars of our methodology.

Another cornerstone of our work is the emphasis on axiomatic or structural methods. We take this to mean that, whenever possible, we prefer to understand systems not just through their interpretation in a semantic universe, but by identifying the fundamental laws that regulate their interaction. Concretely, we work with presentations of symmetric monoidal categories by generators and equations. This is a form of structuralism, focusing on the relationships between systems and the interactive behaviour of concurrent processes rather than some extrinsic notion of functional form. This relative viewpoint fits well with category theory, for which the study of mathematical objects is never done in a vacuum, but through their relations with other similar objects.

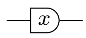
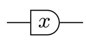
1.2 From concurrent to linear systems and back

Our work stands at the intersection of several strands of research, ranging from concurrency to control theory. What follows is an attempt to situate it within the existing literature.

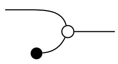
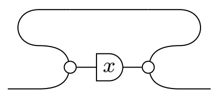
To find some middle ground between Petri's syntax-free approach to concurrency and the versatility of process calculi, some research has been done towards an algebraic study of Petri nets that treats them as open systems. Notable proposals include [Maz87, NPS95, BCEH05, Rei09]. However, the concrete starting point of this thesis is the work of Bruni et al. [BMM11, SMMB13] on *connector algebras* of nets. This work follows the investigation of the algebra of stateless connectors in [BLM06] and extends it to the stateful case of Condition/Event and Place/Transition nets. The result is a symmetric monoidal category (a product and permutations category or *prop*, in fact) of Petri nets with boundaries, which can synchronise with their environment through designated open transitions. Their operational semantics defines a functor into a sub-category of spans of graphs [KSW97b], more specifically a prop of two-sided labelled transition systems with labels for the left and right boundaries.

While efforts were made to characterise the equational theory corresponding to the semantics of Petri nets with boundaries [Sob13], the problem remained open. In the meantime, similar techniques were successfully applied to the field of control theory. One of the central objects of study in control theory are linear time-invariant dynamical systems, i.e., systems whose possible behaviours span a linear subspace of a vector space. The theory of interacting Hopf (IH) algebras [Zan15, BSZ17], gives a sound and complete axiomatisation of the prop of linear relations over a field, capturing all of these behaviours into a convincing graphical syntax, generated by the following basic connectors:

 (1.1)

For fields of fractions of polynomial rings, the diagrams are *signal flow graphs*, interpreted as linear relations over streams. First introduced by Claude Shannon [Sha42], signal flow graphs are a fundamental combinatorial model in control theory, used to represent variable coupling between different parts of cyber-physical systems. It was shown that signal flow graphs faithfully embed into the graphical calculus of IH over $\mathbb{K}\langle\langle x\rangle\rangle$, with the indeterminate  having the operational semantics of a register that stores a value for a single clock-tick, before releasing it into the circuit: computation happens synchronously and at a given time-step, if  holds value v and receives v' as input, it releases v as output and stores v'. Within this calculus, all the usual questions of control theory (and more) can be formulated, such as whether a given specification is physically realisable [BSZ15], whether a given system is controllable [FSR16], whether two systems describe the same behaviour [BSZ14] or whether one is a refinement of the other [BHPS17].

Motivated by the preliminary work of [Sob13], we aim to capitalise on the insights and success of graphical linear algebra in control theory to tackle the problem of characterising the equational theory of nets with boundaries. The graphical nature of Petri nets suggests that there are commonalities. In fact, a striking feature of our approach to concurrency is that we use the same generators as those of IH. Only their interpretation changes: we are interested in relations with values in the semiring $\mathbb{N}$ of natural numbers, instead of taking values in a field. This is a significant if subtle change, with profound consequences for the corresponding equational theory. Consider the following two examples:

$d_1 :=$  $d_2 :=$  (1.2)

As diagrams of IH over, say, the real numbers, these two examples are both equal to linear systems whose behaviour is the total relation , i.e., the entire space.

The key idea is that, when interpreted as carrying values in $\mathbb{N}$, the same two diagrams have a completely different and more interesting operational behaviour. The first diagram denotes the order on the natural numbers and the second one represents represents precisely the behaviour of a place in a Petri net!

We think of this change of model as moving from values that represent continuous *signals* to values that represent discrete *resources*. The connection with Petri nets also invites us to think of these resources as tokens. We chose the term "resource" to reflect the fact that, without additive inverses, we cannot borrow arbitrary values. Indeed, for values in a ring, a given system can access an unlimited quantity by simply spending the corresponding negative quantity. In a non-negative semiring like that of the natural numbers, one can only use resources that one already has. We will focus on the semantics of concurrent computation, but this calculus could plausibly be used for any kind of system with access to resources that are intrinsically non-negative, such as money, goods, items in data structures like stacks or queues, species populations in biology, concentrations in chemistry and more.

The resource-sensitive counterpart of linear relations are *additive relations*, (finitely-generated) sub-monoids of $\mathbb{N}^k \times \mathbb{N}^l$. Additive structures are not new in concurrency. The reader already familiar with Petri nets will know that computations of nets are additive: if we can go from marking $\mathbf{m}_1$ to marking $\mathbf{m}_1'$ through transitions $\mathbf{t}_1$, and from marking $\mathbf{m}_2$ to marking $\mathbf{m}_2'$ through transitions $\mathbf{t}_2$, then it is always possible to reach marking $\mathbf{m}_1' + \mathbf{m}_2'$ from $\mathbf{m}_1 + \mathbf{m}_2$ through transitions $\mathbf{t}_1 + \mathbf{t}_2$. Model-checking algorithms for Petri nets often exploit, at least implicitly, the additive structure of computations [LS15]. The central claim of this thesis is that additive relations constitute a fundamental semantic structure of concurrent computation, particularly well suited to model coordinated access to limited resources.

The first main contribution of this thesis is a characterisation of the equational theory of the prop of additive relations, AddRel. Like IH for linear relations, we give a presentation by generators and equations of AddRel called the *resource calculus*. This presentation shares many features with IH but, despite the presence of similar algebraic structures, their building blocks interact differently, giving insights into the differences between concurrent and linear systems.

While the resource calculus can express basic forms of synchronisation and non-determinism, there are many non-additive phenomena in concurrency for which it is not sufficiently expressive. One notable example is that of *mutual exclusion*: when two or more processes are prevented from accessing a shared resource at the same time. This is paradigmatic example of concurrency control pattern. To be able to account for mutual exclusion and more general patterns of inhibitory synchronisation, we extend the resource calculus with an affine generator $\vdash\!\!-$, whose meaning is the constant resource one. Again, the extended calculus has a linear counterpart—affine subspaces of vector spaces—but, in the additive world, the category of behaviours corresponds to that of *polyhedral relations*. The name stems from a geometric perspective: additive relations are like discrete cones while polyhedral relations can be seen as discrete polyhedra. We extend the equational theory of the resource calculus and obtain a full completeness result, proving that the affine resource calculus gives a presentation of the prop of polyhedral relations.

Finally, we need to clarify what additive relations and the resource calculus have to do with Petri nets and how we can tackle the problem we set out to solve. To do so, we introduce a stateful variant of the resource calculus and derive a presentation for it. The presentation is a very simple extension with a single additional generator $-\!\boxed{x}\!-$ and no equations. The corresponding category of behaviours is a prop of additive labelled transition systems within which we show that those corresponding to the operational semantics of Petri nets can all be expressed. The translation of a Petri net into the stateful resource calculus gives a satisfying solution to the problem of finding an axiomatisation for the nets with boundaries of [SMMB13]. Our microscopic analysis reveals that places in Petri nets (with boundaries) can be decomposed as stateful asynchronous buffers, as in the second diagram of (1.2). The prop of additive relations, axiomatised by the resource calculus, provides the algebra of stateless transitions. If the algebraic structures underpinning Petri nets motivated the development of the resource calculus, the latter is strictly more expressive than Petri nets, which constitute just one possible application of the language we develop in this thesis. Our hope is that the resource calculus will constitute a foundational assembly language for concurrency, into which various paradigms can be compiled and compared.

Of course we do not claim to have conclusively answered Samson Abramsky's question about the fundamental structures of concurrency. First of all, the stateful resource calculus models a very fine notion of process equivalence that is too intensional for most practical purposes. But even if behavioural equivalences, like trace equivalence and (weak or strong) bisimilarity [vG90] are often undecidable, we believe that the resource calculus provides a solid algebraic framework within which many of these notions can be investigated. Similarly, we do not examine fundamental concepts such as causality, conflict or mobility in this thesis but we do not see any immediate obstacle to studying them within this theoretical framework.

We have made an effort to present the content of this thesis as a unified narrative. Scientific research, however, is not linear and the narrative we present has the sort of consistency that only hindsight can afford. Indeed, in the meandering journey of his graduate studies, the author has contracted an intellectual debt to a much wider community than the story suggests. First, our emphasis on axiomatic approaches to open systems owes a lot, not just to the work on interacting Hopf algebras, but also to the related work on categorical quantum mechanics, where a lot of the same algebraic structures appear. In fact, the ZX calculus [CD08], a complete [Bac14, NW17] axiomatisation of the prop of finite-dimensional Hilbert spaces (over powers of two, to model qubit computations) and linear maps, is uncannily similar to IH. From a different but related body of work, the applications of hypergraph categories and of the decorated cospans construction [Fon16] to networks of open systems drawn from a broad range of scientific domains [BF15, BFP16, BP17] has also been an important source of inspiration.

1.3 Roadmap and original contributions

We outline here the overall flow of arguments and the main original contributions of each chapter.

In the second chapter, we give an overview of the basic theory of monoidal categories with an emphasis on the graphical calculus of string diagrams. There are no new results here even if the presentation itself may be new to some: we take a systems-first approach and take special care to show how the algebraic structure of diagrams (and their category-theoretic counterparts) are meaningful operations on distributed systems. We also introduce props and their presentations, the monoidal equivalent of Lawvere theories on which we will rely extensively in subsequent chapters. Finally, we outline briefly the theory of interacting Hopf algebras, the axiomatic approach to the theory of linear dynamical systems, which provides a reference point with much of the same primitives as in our work.

The third chapter is the technical heart of this work. It introduces the prop of additive relations and a sound and complete calculus for it, called the resource calculus. There are several original contributions in this chapter:

- To the best of our knowledge, this is the first time that the definition of the prop AddRel of additive relations, Definition 49, makes its way into print. The definition of additive relation itself was suggested to the author by Paweł Sobociński. The proof that they form a prop, in Proposition 51, is based on a similar proof for a different prop introduced in [Sob13], but appears here in detail for the first time.

- The definition of the resource calculus, the equational theory of additive relations in Definition 54 is an original contribution. Its full completeness, Theorem 59, is entirely new and is one of the most significant results of this thesis.

- All the results of Section 3.7 are, as far as we know, original contributions. We identify a subprop of AddRel that is the Kleisli category of the composite of the powerset and multiset monads and we give a complete calculus for it. This might be of independent interest to logicians studying linear logic as this prop is the dual of a well-known simple model of its Multiplicative Exponential fragment.

In the fourth chapter, we show how to extend the resource calculus to deal with affine phenomena, in order to model more complex behaviour in distributed systems, such as mutual exclusion. To do so, we introduce a simple and elegant technique to axiomatise affine theories from linear ones.

- We first show how to deal with affine transformations in the general case of a cancellative semiring. Presenting affine maps as the coKleisli category of a comonad, and the age-old homogenisation method as an adjunction, are simple but new results that some may find of independent interest. Theorem 112 provides a sound and complete calculus for the prop of affine maps.

- Next we show that we can extend the same method to the prop PolyRel of polyhedral relations. As far as we know, the notion of polyhedral relation appears in this thesis for the first time, and so does the observation that they form a prop. These can be found in Definition 118 and Proposition 119.

- As for additive relations, we characterise precisely the equational theory of PolyRel with the affine resource calculus, introduced in Definition 120. It is shown to be sound and complete in Theorem 124.

- Finally, we apply the results of the previous sections to embed the calculus of stateless connectors, a coordination language introduced in [BLM06], into the resource calculus. This embedding was suggested to the author by Filippo Bonchi and is in some sense immediate. Nonetheless it is an important case study that justifies our approach to concurrency.

- The same section contains an intriguing new result that may be interesting to researchers in categorical quantum mechanics, for whom the symmetric monoidal category of relations (and certain subcategories thereof) is an important toy (non-)model. It turns out that polyhedral relations also include all relations between finite sets and that the affine resource calculus provides a complete graphical calculus for fRel. This is the content of Theorem 131.

The last chapter explores extensions of the resource calculus to model stateful systems and formalises the link with Petri nets.

- We formalise a well-known construction that builds a prop St(T) of state-passing systems from any prop T in a natural way. This idea is commonplace in computer science, but we do not know of any paper that formalises the construction on arbitrary props.

- We show in Theorem 138 that, if T is compact closed, then St(T) is isomorphic to the coproduct of T with the prop freely-generated by a single $1 \to 1$ generator and no equations. This is a simple yet elegant result which the author was surprised not to find explicitly stated anywhere in the literature.

- It follows from Theorem 138 that, given an axiomatisation of T, we can immediately derive one for its stateful extension. We apply this idea to the resource calculus to define its stateful extension and show that it is isomorphic to a category of additive labelled transition systems.

- Equipped with the preceding results, we turn to another case study: we show in Proposition 147 that Petri nets can be faithfully encoded into the stateful resource calculus and that the embedding preserves their operational semantics.

- We also show that the stateful resource calculus is strictly more expressive than Petri nets in Corollary 149.

- In Section 5.2.2 we examine different notions of state with different semantics from the usual operational behaviour of Petri nets and show that they all fit uniformly within the stateful resource calculus.

- In Section 5.2.3 we turn to Petri nets with boundaries, as defined in [BMM11, SMMB13] and show that they are also captured within our framework.

- Next, we try to explain why the encoding of Petri nets works for additive relations but fails for linear relations. A preliminary result in this direction is the existence of a traced embedding of relations between finite sets (with the disjoint sum as monoidal product) into AddRel, Theorem 162.

1.4 Prerequisites

Throughout this thesis, we will assume familiarity with basic category theory, not only the definitions of category, functor and natural transformation, but also, more broadly, a rudimentary understanding of universal properties, of common (co)limits such as (co)products, pullbacks and pushouts or (co)equalisers, and of adjunctions and monads. The reference on these topics is still [ML13] but the author has also found [Lei14] to be a useful introductory account.

Chapter 2

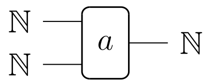
Background

2.1 Open systems as diagrams

This section gives an overview of the diagrammatic approach to the study of open systems—a field that has recently risen to prominence in a variety of scientific contexts [Ghi13, BF15, BFP16, BP17, CK17, CSC10]. We will favour diagrams over symbolic notations, not only to avoid unnecessary bureaucratic overhead, but also to highlight the spatially distributed nature of the systems we want to model. We will nonetheless recall the relevant category-theoretic notions systematically, to connect—at least in this preliminary chapter—the diagrammatic notation with standard mathematical practice. For a more comprehensive introduction to string diagrams, we recommend Selinger's survey [Sel11].

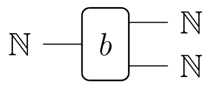
We will denote as *system* anything with a number (including zero) of ports, connection points or terminals through which it can interact with its environment, for instance, by sharing resources like energy or information. Although we have distributed models of computation in mind, we use the word "system" in a very liberal way, without referring to any specific domain of application or level of abstraction. These systems could refer to software services communicating over a network, hardware components running in parallel and potentially accessing shared memory, or even abstract models of computation, like automata and state-machines of various kinds with communicating capabilities. They could also be physical systems, interacting at a microscopic level of granularity according to the laws of quantum mechanics, or macroscopic systems that can freely exchange energy in the form of heat. They could also be biological systems interacting as part of complex chemical cycles, or even cooperating/competing social agents and institutions.

Because our syntax is two-dimensional, it is useful to partition the connection points of a given system into a left and right interface. We should not think of these as inputs or outputs, but simply as an artefact of how we compose diagrams in the plane. The distinction between right and left ports is useful in practice to describe how to connect various systems to form more complex ones, by assembling them horizontally and vertically, like two-dimensional LEGO bricks. The open ports are labelled by types in order to specify to which other ports they can be connected. For example, consider a system with two ports on the left and one on the right that can exchange discrete resources (modelled by $\mathbb{N}$) with its environment. It can be depicted as the following box with labelled wires:



(2.1)

Suppose that it constrains the observable value at its right boundary to be equal to the sum of the values observed on the two leftmost ports. It seems legitimate to interpret the system a as a functional process, which takes two natural numbers as inputs and outputs their sum. But we could have also considered a system with the converse set of constraints between the observable values at its boundary, $b = \{(p,(n,m)) \mid n,m,p \in \mathbb{N} \text{ and } p = n+m\}$, represented as



(2.2)

Is it better to think of b as a running backwards or as a process that takes a quantity of resources as input and splits it into two? In this case, which ports are inputs and which are outputs? In traditional approaches to computation, inputs capture the causal effect of the environment on a system, and outputs the converse. A computation effectively transforms inputs into outputs. Processes can be chained sequentially, feeding the output of one system as input to another. We may even allow feedback to connect some outputs to inputs of the same process, but this fundamental distinction still underpins most models.

By contrast, in our approach, we make no assumptions about the inherent directionality of causal flow between the different parts of the systems that our diagrams describe. Systems are better thought of as constraining the interactions with the environment at their connection points, independently of any a priori distinction between inputs and outputs. These notions still have a place in our framework, but they are emergent properties of how different parts of a system interact.

This position is not new and we do not claim that it is our own. Control theorist Jan Willems is perhaps one of the first to have articulated these principles and warned against the use of inputs and outputs as primitive notions when modelling physical systems [Wil07]. He contends that physical laws are not intrinsically directed, but merely express a relation between the observable variables of a system:

> "For example, the gas law states how the variables of interest, temperature, volume, and mass are related. This law does not, however, state that some of the variables generate the others. The interconnection of two physical devices means that certain variables associated with the first device are set equal to certain variables associated with the second device. Connecting two pipes of two hydraulic systems means that the pressure and flow in the first pipe at the interconnection point are set equal to the pressure and flow in the second pipe at the interconnection point. After interconnection, the two hydraulic systems share the pressure and flow variables."

This perspective gave rise to the *behavioural* approach to dynamical systems in control theory. Models of (concurrent) computation are the abstract representation of a physical system from which we wish to generate a certain behaviour and we claim that the same principles can be fruitfully applied to them. To effectively compute, we need to ensure the existence of a mapping of the model onto physical reality, and one that is impervious to irrelevant perturbations. This mapping should also be compositional, in the sense that the behaviour of the whole system can be derived from the behaviour of its parts and how they are interconnected.

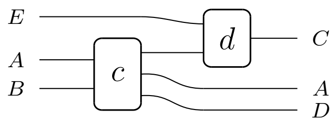
This is not to say that inputs and outputs have no place in models of computation. After all, when a user presses the return key to launch a program, there is a direct causal effect. It is not the machine launching a program that causes the user to press the key. Certain systems have an irreducible a priori directionality. For example, we will see later that places in a Petri net have input and output ports that direct the flow of tokens in the net. Willems' behavioural approach encourages us to be careful about not introducing causal relationships where they can be avoided, in order to identify genuine inputs and outputs.

2.1.1 Parallel and synchronising composition

In the behavioural approach, the composition of physical systems does not correspond to the chaining of the output signal of one process to the input of another, but is understood more simply in terms of synchronisation or variable coupling. This paradigm shift is the inevitable consequence of the absence of any inherent directionality between the ports of the systems we consider. If we have access to the two systems below

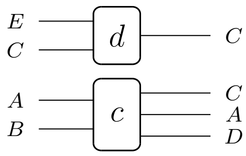
(2.3)

we can connect them to obtain



(2.4)

In what follows, we are interested in distributed systems and networks of interacting components, which can coexist in parallel, without necessarily communicating. We represent the resulting joint system as the vertical juxtaposition of the two:



(2.5)

Note that, contrary to the operation of interconnection, parallel composition is not typed, and two systems can always be joined into a larger one in which they do not interact.

From these two primitive operations, we can obtain complex networks of systems interacting in various ways, whose only relevant structure is their connectivity.

2.1.1.1 Monoidal categories

We now reveal that the diagrams that we use to represent systems are morphisms of monoidal categories. We will see that our diagrams are in fact morphisms in highly-structured monoidal categories, unveiling the additional algebraic structure progressively in the following sections.

Category theory provides a general mathematical framework to study systems and processes with a primitive typed operation of composition. Monoidal categories are a convenient abstraction to model the interaction between the usual typed composition and an untyped form of composition, called monoidal product. To guarantee that they behave as the diagrams suggest, we need the axioms in the following definition (and more, as we will see later).

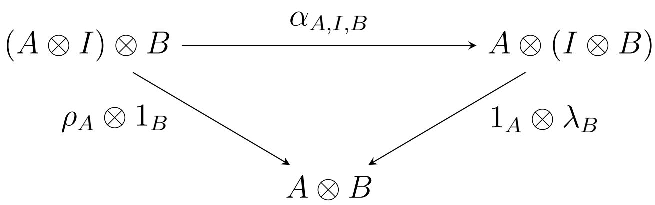
Definition 1. A *monoidal category* is a category equipped with

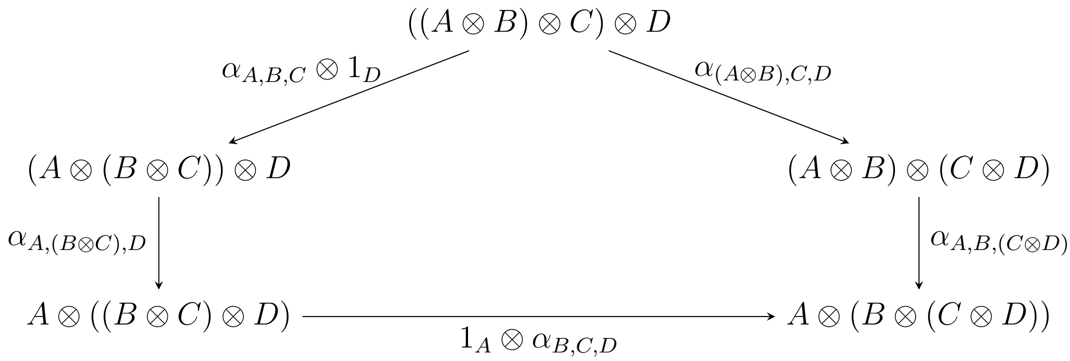
(a) a monoidal product functor $\otimes : \mathsf{C} \times \mathsf{C} \to \mathsf{C}$

(b) that is *associative*, i.e for all objects A, B and C, we have a natural isomorphism $\alpha_{A,B,C} : A \otimes (B \otimes C) \to (A \otimes B) \otimes C$, called the associator; and

(c) such that there exists a distinguished object I, called the monoidal *unit*, with two natural isomorphisms $\rho_A : A \otimes I \to A$ and $\lambda_A : I \otimes A \to A$, for all objects A, called the unitors;

(d) subject to the following coherence conditions (see [ML13]):





These two axioms are sufficient to prove that any two morphisms with the same domain and codomain, built up from the identities, associators and unitors, are equal. As a result there is a unique structural morphism between any two ways of bracketing the same list of objects, such as $A \otimes (B \otimes C)$ or $(A \otimes B) \otimes C$. This is the content of MacLane's coherence theorem [ML13, Theorem VII.2.1].

Remark 2 (Notational convention)**.** To match more closely the diagrammatic representation of morphisms in monoidal categories, we will usually write $f\,;g$ for the composition of $f\colon A \to B$ followed by $g\colon B \to C$.

Less often, we will adopt the more common symbolic order, denoting it by simply concatenating the two morphisms, as in gf. We will need this when applying certain morphisms to arguments, as in $gf(x)$, or when multiplying matrices.

Example 3. Perhaps one of the simplest examples of monoidal category is Set, the category of sets and functions, with the Cartesian product as monoidal product, and the singleton set as monoidal unit. In fact, any category with finite products is monoidal, with the categorical product as monoidal product. These categories are often called Cartesian monoidal and the terminal object is always the unit. Dually, a category with finite coproducts is also monoidal and has the initial object as unit. Such categories are called coCartesian monoidal.

Example 4. Let Rel be the category with

- sets $X, Y, \ldots$ as objects;
- binary relations $r \subseteq X \times Y$ as morphisms $X \to Y$;
- composition given by

  $$r \,;\, s = \Big\{(x,z) \mid \exists y \in Y,\, (x,y) \in r \text{ and } (y,z) \in s\Big\};$$

  for relations $r \colon X \to Y$ and $s \colon Y \to Z$;
- $1_X = \{(x,x) \,|\, x \in X\}$ as identity for X;

Rel can be given the structure of a monoidal category in at least two different ways. The Cartesian product (which is not the categorical product in this case) given by $X \times Y$ on objects, and by

$$r_1 \times r_2 = \Big\{((x_1,x_2),(y_1,y_2)) \mid (x_1,y_1) \in r_1 \text{ and } (x_2,y_2) \in r_2\Big\}$$

for relations $r_i \colon X_i \to Y_i$ $(i = 1,2)$, is a monoidal product whose unit is the singleton set $1 = \{\bullet\}$, associator is $\alpha_{X,Y,Z} = \Big\{\Big(((x,y),z),(x,(y,z))\Big)\Big\}$ and (left-)unitor $\lambda_X = \{(x,(\bullet,x))\}$. We will refer to this monoidal category as $\mathsf{Rel}_\times$.

In addition, the disjoint sum of sets is a biproduct for Rel. We will occasionally be interested in this other monoidal product and will refer to the corresponding monoidal category as Rel_+.

Most of the categories that we will consider in the next chapters are monoidal subcategories of $\mathsf{Rel}_\times$ and inherit additional algebraic structure from it, each corresponding to intuitive diagrammatic operations on systems: most notably, a symmetry, a form of feedback and the ability to copy and discard parts of systems. We investigate these notions in their full generality in Sections 2.1.2.1, 2.1.2.2 and 2.1.2.3, respectively.

2.1.1.2 Monoidal functors and natural transformations

Diagrams represent morphisms in arbitrary monoidal categories. In what follows we will consider specific systems and define their behaviour in reference to a given model. For this we will need to map diagrams functorially to a corresponding monoidal category that constitutes our semantic universe. But functors are too flexible, as they may not preserve the monoidal product. This is why we define the relevant notion of structure-preserving functor between monoidal categories and natural transformation between such functors.

Definition 5. A *monoidal functor* is a functor $F : \mathsf{C} \to \mathsf{D}$ between two monoidal categories $(\mathsf{C}, \heartsuit)$ and $(\mathsf{D}, \spadesuit)$ such that

(a) there exists a natural transformation $\phi_{A,B} : F(A)\spadesuit F(B) \to F(A\heartsuit B)$

(b) and a morphism $\phi_I : I_\mathsf{D} \to FI_\mathsf{C}$,

satisfying the following commutativity conditions for all objects A, B and C of C:

$$\begin{array}{ccccc}
(FA\spadesuit FB)\spadesuit FC & \longrightarrow & FA\spadesuit(FB\heartsuit FC) & \xrightarrow{1_{FA}\spadesuit\phi_{B,C}} & FA\spadesuit F(B\heartsuit C) \\
\downarrow{\scriptstyle \phi_{A,B}\spadesuit 1_{FC}} & & & & \downarrow{\scriptstyle \phi_{A,B\heartsuit C}} \\
F(A\otimes B)\spadesuit FC & \xrightarrow[\phi_{A\heartsuit B,C}]{} & F((A\heartsuit B)\heartsuit C) & \longrightarrow & F(A\heartsuit(B\heartsuit C))
\end{array}$$

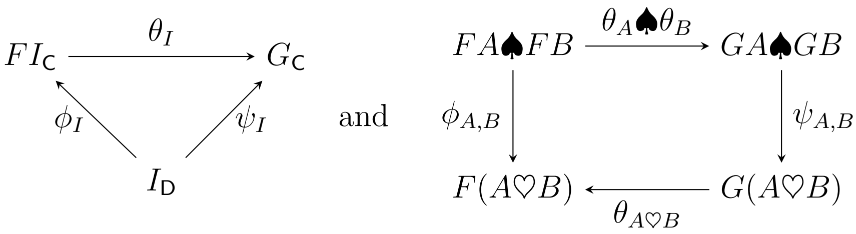
and

$$\begin{array}{ccc}
FA\spadesuit I_\mathsf{D} & \xrightarrow{FA\spadesuit\phi_I} & FA\spadesuit FI_\mathsf{C} \\
\downarrow & & \downarrow{\scriptstyle \phi_{A,I_\mathsf{C}}} \\
FA & \longleftarrow & F(A\heartsuit I_\mathsf{C})
\end{array}
\qquad
\begin{array}{ccc}
I_\mathsf{D}\spadesuit FB & \xrightarrow{\phi_I\spadesuit FB} & FI_\mathsf{C}\spadesuit FB \\
\downarrow & & \downarrow{\scriptstyle \phi_{I_\mathsf{C},B}} \\
FB & \longleftarrow & F(I_\mathsf{C}\heartsuit B)
\end{array}$$

where the unlabelled morphisms are the only sensible associators or unitors in the monoidal categories C and D.

We say that a functor is *strong monoidal* if the components of ϕ are all invertible. If they are identities, the functor is *strict monoidal.* There is also a notion of *oplax monoidal* functor with $\phi_{A,B} : F(A\heartsuit B) \to F(A)\spadesuit F(B)$ and $\phi_I : FI_\mathsf{C} \to I_\mathsf{D}$, with the associativity and unitality coherence conditions pointing in the other direction as well.

Definition 6. A *monoidal natural transformation* between two monoidal functors $(: F, \phi)$ and (G, ψ) is a natural transformation $\theta\colon F \to G$ such that



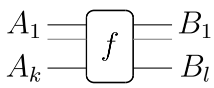
commute for all objects A and B.

Definition 7. Two monoidal categories C and D are *monoidally equivalent* when there exist strong monoidal functors $F\colon \mathsf{C} \to \mathsf{D}$ and $G\colon \mathsf{D} \to \mathsf{C}$ with monoidal natural isomorphisms $FG \to 1_{\mathsf{D}}$ and $1_{\mathsf{C}} \to GF$ (note that 1_{C} and 1_{D} are clearly strict and therefore strong monoidal functors).

2.1.1.3 Strictness or why the graphical calculus works

If its associators and unitors are all equalities, a monoidal category is said to be *strict* monoidal. In this thesis, most of the categories that we study are strict. But even when they are not, we can always find a monoidally equivalent strict monoidal category—this is another interpretation of the coherence result of MacLane. Consequently, we can write $A \otimes B \otimes C$ without ambiguity and safely ignore the associator isomorphisms. Similarly, we will always omit ρ and λ by identifying A with either $A \otimes I$ or $I \otimes A$.

The coherence theorem has another important consequence for our purposes: it justifies the correspondence between morphisms in a monoidal category and the diagrammatic representation of systems. In the graphical calculus of monoidal categories, a morphism $f\colon A_1 \otimes \cdots \otimes A_k \to B_1 \otimes \cdots \otimes B_l$ is depicted as a box with labelled wires or ports:

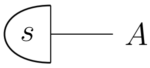


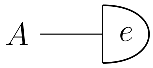
(2.6)

Note that—justified by the coherence theorem—this representation does not distinguish between different bracketings of the domain and codomain. The identity on an object A, is depicted as a single labelled wire, and the identity on $A \otimes B \otimes C$ as three parallel wires:

(2.7)

The monoidal unit is omitted from diagrams so that morphisms of type $s\colon I \to A$ (often called states) and $e\colon A \to I$ (often called effects) are represented respectively by



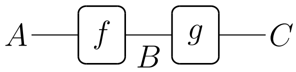
and


(2.8)

and the the identity on I is simply the empty diagram:

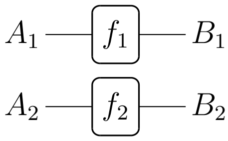
$\mathrm{id}_I =$  (2.9)

As expected, the composition $f\,;\,g$ of morphisms $f\colon A \to B$ and $g\colon B \to C$ is represented by connecting the intermediate matching wires:



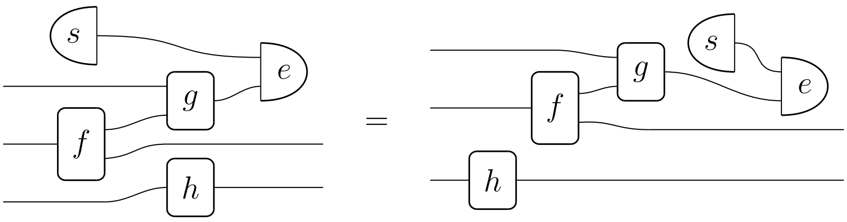
(2.10)

and the monoidal product of $f_1\colon A_1 \to B_1$ and $f_2\colon A_2 \to B_2$ is depicted as the vertical juxtaposition of the corresponding diagrams:



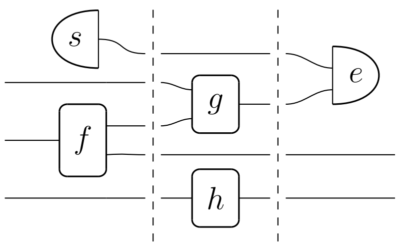
(2.11)

The graphical calculus of monoidal categories comes with an intuitive and mathematically rigorous notion of topological equivalence. For example, the following two diagrams are equal:



(2.12)

The diagram on the left above can be decomposed as



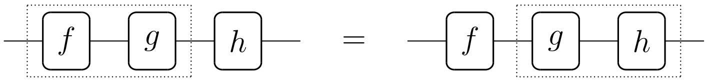
(2.13)

or, in purely symbolic notation,

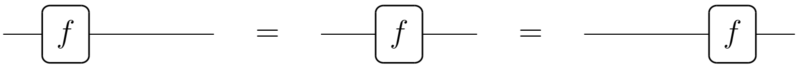
$$(s \otimes \mathsf{id} \otimes f \otimes \mathsf{id})\,;\,(\mathsf{id} \otimes g \otimes \mathsf{id} \otimes h)\,;\,(e \otimes \mathsf{id} \otimes \mathsf{id}) \tag{2.14}$$

The reader will easily see from this last expression why we prefer to use the diagrammatic notation. The first advantage is that the wiring makes the type explicit while it does not appear in the symbolic formula in (2.14). But more importantly, the diagrammatic syntax turns simple topological deformations into theorems. The defining equations of strict monoidal categories are diagrammatic tautologies (where we omit object labels for clarity):

- Associativity and unitality of composition:

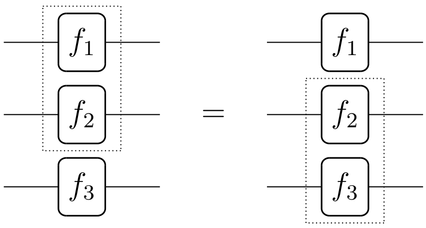


(2.15)

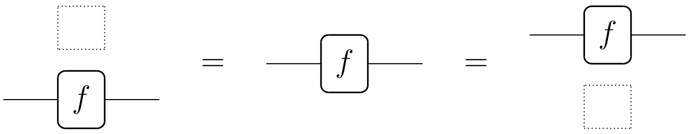


(2.16)

- Associativity and unitality of the monoidal product:

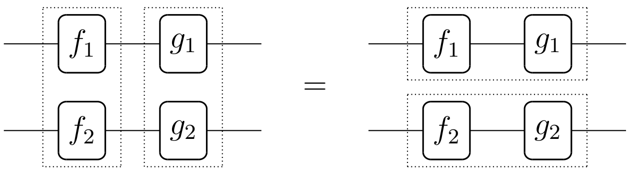


(2.17)



(2.18)

- Functoriality of the monoidal product, also known as the interchange law:



(2.19)

Finally, we emphasise that the graphical calculus is not just a convenient visual representation, but a rigorous way to derive equalities of morphisms in monoidal categories: it is sound and complete for the axioms of monoidal categories by a result of Joyal and Street [JS91]. We will not discuss the precise definition of which topological deformations are allowed between morphisms of monoidal categories, trusting that our intuition is sound. For our purpose, it is sufficient to say that the allowed deformations are precisely those obtained by applying locally any sequence of the graphical rules above. Formally, diagrams are equivalence classes of two-dimensional terms, modulo the axioms of monoidal categories. Graph rewriting for string diagrams is a subject of active research [BGK$^+$16, BGK$^+$18].

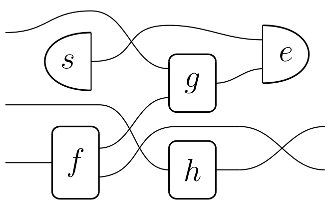
2.1.2 Only the connectivity matters

Only the connectivity of systems matters: we do not wish to keep track of where the diagrams are located on the page on which we depict them, but only of which ports are connected to each other. For many purposes, this is a considerable abstraction step. When designing integrated circuits for example, latency, power consumption and heat generation place constraints on the spatial distribution of components. In this thesis however, we will not be concerned with the physical implementation of systems. This is a choice, and we do not wish to minimise the practical importance of these concerns. Our purpose is principally theoretical and therefore it is simpler to model systems as abstract representations of physical processes, whose dynamics are coarse-grained into (mostly discrete) states and events, independent of their precise spatial location. Spatial structure still matters, insofar as systems are distributed and connected to each other with a certain topology.

Interestingly, this abstraction step allows us to draw more general and permissive kinds of diagrams, yet imposes more algebraic structure on their category-theoretic counterpart.

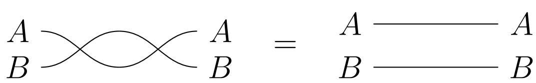
2.1.2.1 Symmetry

The first consequence of the insensitivity to spatial distribution is that we should be able to braid wires in order to connect systems in scenarios like this:



(2.20)

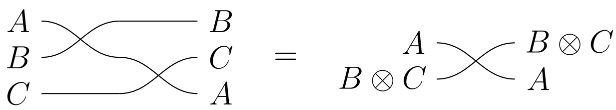
Furthermore, we do not want to keep track of which wires are going below or above so that the braiding ⤫ should be self-inverse:



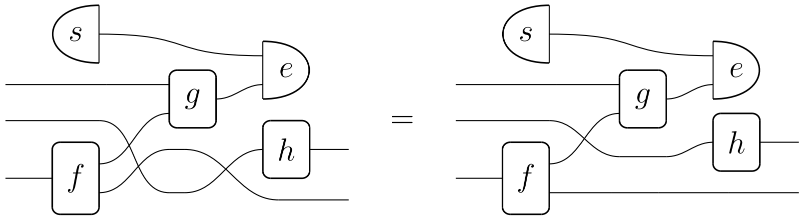
(2.21)

At the category-theoretic level, this leads to the definition of *symmetric monoidal categories*, smc for short. The graphical calculus of smc is that of plain monoidal categories along with the braiding and corresponding axioms. Again, in terms of diagrams, these axioms are either self-evident tautologies or simple deformations.

Definition 8. A *symmetric monoidal category* is a monoidal category with a natural isomorphism $\times_A^B\colon A \otimes B \to B \otimes A$ that is self-inverse and compatible with the monoidal product, in the sense that



These coherence equations allow us to deduce that the diagram in (2.20) is equal to



(2.22)

using simple deformations of the wires.

Example 9. The monoidal category $\mathsf{Rel}_\times$ is symmetric monoidal with braiding given by

$$\times_X^Y := \{((x,y),(y,x)) \mid x \in X \text{ and } y \in Y\} \tag{2.23}$$

for the sets X and Y.

Unsurprisingly, we require that structure-preserving functors between smc respect the braiding.

Definition 10. A *symmetric monoidal functor* is a monoidal functor (F, ϕ) for which

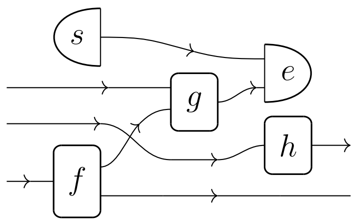
$$\begin{array}{ccc} FA\spadesuit FB & \xrightarrow{\phi_{A,B}} & F(A\heartsuit B) \\ {\scriptstyle \times_{FA}^{FB}}\downarrow & & \downarrow {\scriptstyle F(\times_A^B)} \\ FB\spadesuit FA & \xleftarrow[\phi_{B,A}]{} & F(B\heartsuit A) \end{array}$$

commutes (note that if F is stict, the condition is trivial).

Like for monoidal categories, there is a coherence theorem for smc [ML13, Theorem XI.3.1] that guarantees the soundness and completeness of the graphical calculus.

2.1.2.2 Bending wires with cups and caps

We emphasised at the beginning of this chapter that our representation of systems does not intrinsically distinguish inputs from outputs. And, since it should also be insensitive to the spatial location of systems, there are no constraints preventing us from connecting any two ports, provided that their types match. Unfortunately, the rules of smc do not allow for this generality. Morphisms in every category come equipped with an inherent distinction between domain and codomain. For us, this is a syntactic distinction, providing a convenient language to describe how to connect systems together when drawing them in two-dimensions. However, smc take this distinction more seriously than we would like: every diagram representing a morphism in a smc admits an implicit causal structure, given by a directed acyclic graph, flowing from left to right. It is therefore always possible to assign a consistent direction to the wires of a diagram, for example:

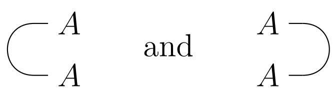


(2.24)

Conversely, one can show that a diagram can be interpreted in a smc precisely when the directed graph, whose nodes are the boxes and edges are the wires (with the direction going from left to right), is acyclic [CK15]. We will not make these statements more rigorous nor prove them here but the reader can convince themselves by noticing that morphisms in a smc are built by composing layers of monoidal products of morphisms, from left to right.

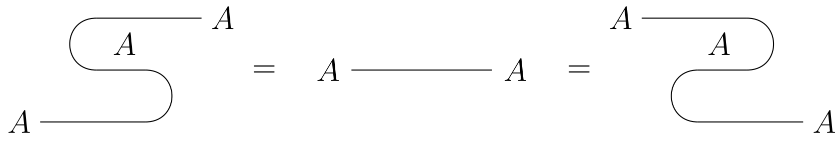
At first, it seems like the category-theoretic counterpart of general diagrams should be some generalisation of smc that forgets the directionality of morphisms. But, as with the braiding, we can obtain the desired algebraic structure by giving ourselves new distinguished morphisms satisfying equations that reflect the topological properties of diagrams. The reader may find interesting that—once again—more permissive diagrams correspond to more restrictive categories or categories with more sophisticated structure.

The additional diagrams are depicted as 90 degree turns:



(2.25)

for each object A, satisfying the triangle equalities—which, once more, are obvious topological deformations:



(2.26)

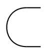
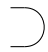
and

(2.27)

In a smc, we call two morphisms satisfying these equalities, *cups* and *caps* respectively.

Definition 11. A *self-dual compact closed category* is a smc that has cups and caps for every object.

Example 12. The smc $\mathsf{Rel}_\times$ is self-dual compact closed with cup and cap for the set X given by

 $:= \{(\bullet, (x, x)) \mid x \in X\}$ and  $:= \{((x, x), \bullet) \mid x \in X\}$ (2.28)

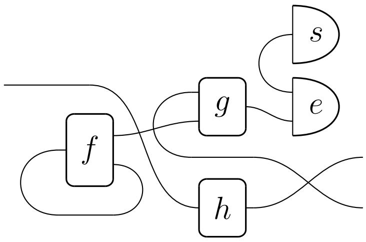
Remark 13. There is a more general notion simply called a *compact closed* category, for which cups and caps link an object A to its dual, denoted by A^*. In the graphical calculus of compact closed categories, wires carry a direction represented by an arrow and A^* has its arrow going in the opposite direction to that A; cups and caps reverse the direction:

(2.29)

This definition forces the dual of an object to be isomorphic to it, in any smc. In this thesis, all of the compact closed smc that we will consider are self-dual—in the sense that the isomorphism is an equality, $A = A^*$—so we will not need the extra level of generality and will not need to label wires with arrows.

Remark 14. The reader versed in the language of higher categories may know that monoidal categories are equivalent to 2-categories with a single 0-cell. Then, the triangle equalities for cups and caps correspond to the defining equalities of an adjunction in an arbitrary 2-category. For a given object (seen as a 1-cell in the equivalent 2-category), its associated cup and cap are the counit and unit of this adjunction. As a result, they determine each other uniquely and the existence of duals is a *property* of a given smc, rather than a *structure*. Hence, when they exist, duals are unique up to isomorphism.

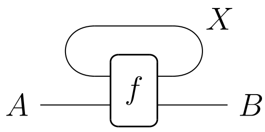
The graphical language of self-dual compact closed categories allows us to bend wires at will, treating them as unoriented edges between the connection points of individual subsystems:



(2.30)

The coherence theorem for compact closed categories guarantees the soundness of these graphical operations and can be found in [KL80], one of the first papers to contain string diagrams.

From compacity, we can derive a form of feedback called the *partial trace* of a system, by connecting two of its ports:

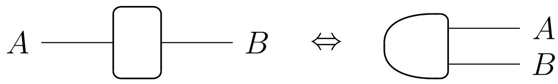


(2.31)

As a result, every (self-dual) compact closed category is also *traced monoidal*, a notion introduced in [JSV96], that we will only need in Section 5.3.1 where it will be defined.

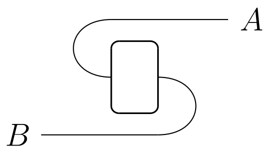
We can also move any port from the left to the right boundary and vice-versa. This has the following important consequence.

Proposition 15. *In a self-dual compact closed category, for any two objects A and B, there is a bijective correspondence between morphisms of the following form:*



(2.32)

We will make extensive use of this fact when finding normal forms for morphisms of certain self-dual compact closed categories: because of Proposition 15, it will be enough to consider systems with ports only on one side of their boundary. By applying this bijection once on each side, we can deduce that, in a self-dual compact closed category C, $\mathsf{C}(A, B) \cong \mathsf{C}(B, A)$. We call the operation that realises this bijection the *transpose*:

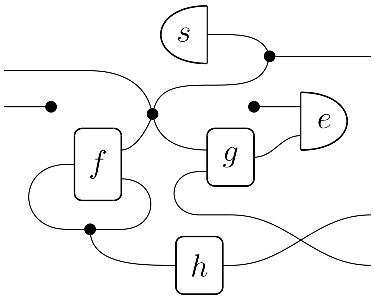


(2.33)

For a morphism d, we denote its transpose symbolically by $d^\dagger$. Note that, in $\mathsf{Rel}_\times$, the transpose computes the converse of a relation: $(x, y) \in r$ iff $(y, x) \in r^\dagger$, for a relation $r \colon X \to Y$.

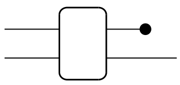
2.1.2.3 Copying and discarding with the Frobenius axioms

We started with the idea that composition of systems is variable sharing, represented as connecting *two* ports together. But there is no reason that we should limit ourselves to just two. The primary source of examples of distributed systems are networks of interacting computers. In a network, machines share channels on which they communicate and synchronise access to certain resources. Often these channels connect more than two subsystems together. This is fundamental and our diagrammatic syntax should reflect this capability, allowing us to draw systems like these:



(2.34)

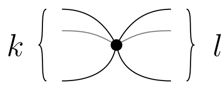
Similarly, a system may have ports that we wish to ignore so we want to be able to discard parts of a system that are irrelevant in our model. Diagrammatically, this means connecting a wire to *no* other port:



(2.35)

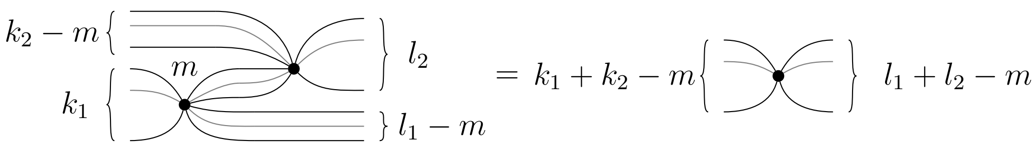
We see that diagrams are not just graphs anymore, but hypergraphs with systems as vertices and wires as hyperedges. And, once again, by identifying the algebraic structure we need, we will obtain the category-theoretic counterpart that captures the richer graphical calculus.

Frobenius monoids Intuitively we need nodes with arbitrary arity, like



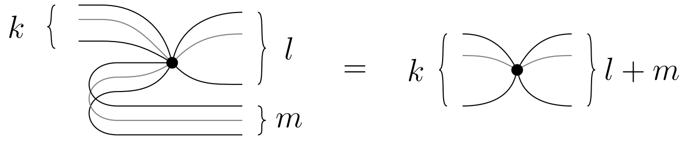
(2.36)

that compose in the obvious way:



(2.37)

and whose edges can be bent in arbitrary ways, in the sense that



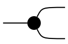
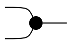
(2.38)

It is possible to obtain an algebraic structure that satisfies these laws by giving ourselves a finite number of morphisms subject to a finite number of simple equations that capture the desired topological invariance. All we need are morphims that can copy (dually, merge) and delete (dually, generate) wires:

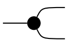
 (2.39)

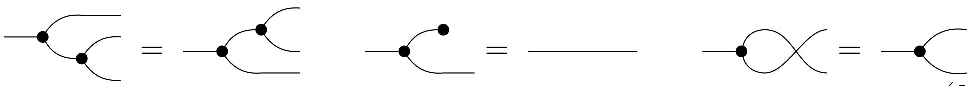
satisfying the equations of extra-special commutative Frobenius monoids.

Definition 16. An *extra-special commutative Frobenius monoid* in a smc is an object A together with morphisms

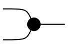
 $: A \to A \otimes A$  $: A \to I$  $: A \otimes A \to A$  $: I \to A$

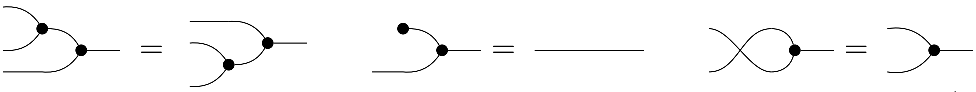
such that

(a)  and  form a cocommutative comonoid:



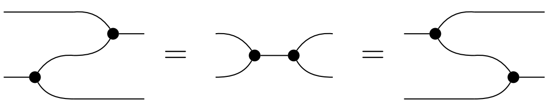
(2.40)

(b)  and  form a commutative monoid:

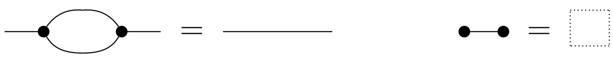


(2.41)

(c) they obey the Frobenius equations:

 (2.42)

(d) and the separability and bone equations:

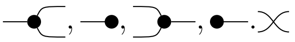
 (2.43)

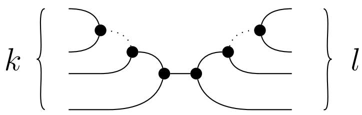
Remark 17. Originally defined in the context of representation theory [BN37], the general definition of Frobenius monoids in smc is due to [CW87], where they first appeared under the name commutative separable algebras. The Frobenius equations are a famous algebraic pattern bridging algebraic and topological phenomena, see [Koc04, Lac04].

The equations we have given contain some redundancies for the sake of symmetry. For example, any one of the two Frobenius axioms is sufficient to prove the other. Similarly, we only need the commutativity of the monoid to prove the cocommutativity of the comonoid, and vice-versa.

Most of the Frobenius monoids in this thesis will be commutative, separable and satisfy the bone equation so we will simply refer to them as Frobenius monoids by default, unless stated otherwise.

There is a coherence result for Frobenius monoids in a smc, called the *spider theorem*, guaranteeing that any two ways of connecting multiple ports together through any number of black nodes are equal.

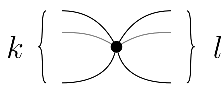
Theorem 18 (Spider theorem)**.** *Every morphism $A^{\otimes k} \to A^{\otimes l}$ constructed from*  *and the identity on A, using composition and the monoidal product, such that the corresponding diagram is* connected *as a undirected graph, is equal to*



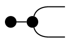
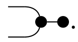
(2.44)

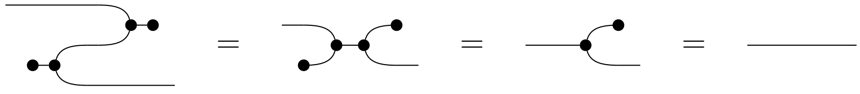
Proof. This result is an implicit corollary of the work of Lack on composing props via distributive laws [Lac04]. It appears explicitly for the first time as the spider theorem in [CPP08], where Frobenius algebras are used as the interface between quantum and classical systems. A proof by a normal form argument can be found in [HV, Theorem 4.23]. □

Thanks to the spider theorem, we can see any composition of black comonoids and monoids as an abstract vertex with dangling wires, whose only relevant structure is the number of connecting points on the left and on the right. The normal form guarantees that we can unambiguously depict any such morphism $A^{\otimes k} \to A^{\otimes l}$ as a *spider* with k legs on the right and l legs on the left, which is what we wanted:



(2.45)

Remark 19. Given Frobenius monoid, we can define  and . Notice that these two morphisms satisfy the triangle equalities:



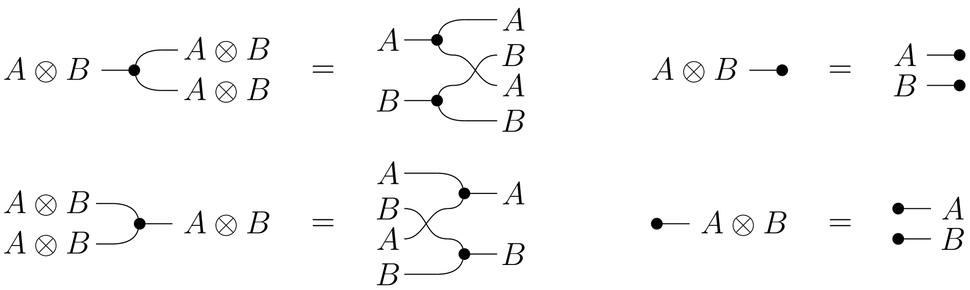
(2.46)

The other equality can be proven similarly. Finally, the cups and caps given by the Frobenius structure are clearly symmetric by the commutativity of the (co)monoid.

In a self-dual compact closed category, the cups and caps of a Frobenius monoid may or may not coincide with the existing structure. If they do not, we may not be able to bend the legs of spiders using the compact structure of the category in a way that is consistent with the notation. Luckily, the cups and caps in categories of interest to us will all come from Frobenius monoids so we will not need to pay attention to this slight annoyance.

Hypergraph categories Smc with a consistent choice of Frobenius monoid on each object have been reinvented (or rediscovered, depending on one's philosphical inclination) several times independently, receiving a different name each time: well-supported compact closed categories in [Car91] and the subsequent work of Walters and his collaborators [RSW05], dgs-monoidal categories in [GH97], dungeon categories in categorical quantum mechanics [Mor14], and the now more established hypergraph categories, in [Kis14] and [Fon16]. We will adopt the latter.

Definition 20. A *hypergraph category* is a smc with a special commutative Frobenius structure on each object, compatible with the monoidal product, in the sense that

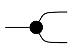
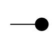


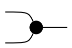
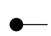
(2.47)

(2.48)

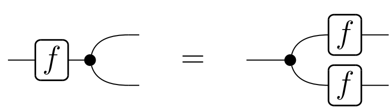
and for which the unit is coherent[1], i.e., the Frobenius structure on I is given by the unitor isomorphism and the identity: $(\rho_i^{-1}, \mathsf{id}_I, \rho_i, \mathsf{id}_I)$.

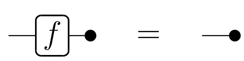
Example 21. Our running example, $\mathsf{Rel}_\times$, is a hypergraph category in which the Frobenius structure on a set X is given by

 $:= \{(x, (x, x)) \mid x \in X\}$  $:= \{(x, \bullet) \mid x \in X\}$ (2.49)

 $:= \{((x, x), x)) \mid x \in X\}$  $:= \{(\bullet, x) \mid x \in X\}$ (2.50)

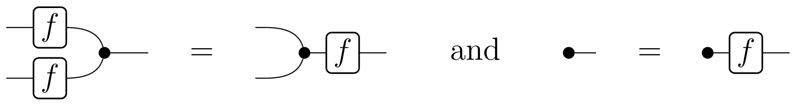
Remark 22. The definition above does not require the (co)monoid to be natural. Morphisms in a hypergraph category may or may not be (co)monoid homomorphisms:

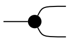


and


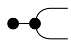
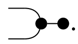
(2.51)

[1]This additional requirement was overlooked in much of the literature; it was introduced in [FS18].

 (2.52)

In fact, for a given hypergraph category it is generally instructive to look at those that do commute with the (co)monoid. For example, in $\mathsf{Rel}_\times$, the relations that preserve  are precisely the functional relations (those that are single-valued) and those that preserve  are precisely the total relations. Therefore, the comonoid homomorphisms are exactly the functions.

Proposition 23. *Every hypergraph category is self-dual compact closed.*

Proof. The cups and caps on each object are given by  and . That they satisfy the triangle equalities and are symmetric are proven in Remark 19. □

2.2 Props

We give an introductory account of props for the purpose of this thesis; for a more in-depth look, the reader is referred to [Lei04, Zan15, Lac04].

Definition 24. A *pro*duct and *p*ermutations category or *prop* is a strict symmetric monoidal category with the natural numbers as objects and addition as monoidal product. A morphism of props is a strict symmetric monoidal functor that is the identity on objects. Props form a category we call Prop.

Example 25. We give below two examples of props that will be useful later.

(a) Since every finite set is a disjoint sum of a finite number of copies of the singleton set, the full symmetric monoidal subcategory of Rel_+ on finite sets is monoidally equivalent to a prop. The symmetric monoidal functor realising this equivalence maps every finite set to its cardinality. Write $\underline{k}$ for the ordinal $\{0, \ldots, k-1\}$ and $|X|$ for the cardinality of the set X. We can fix an ordering on each finite set and use it to define the functor on morphisms: it maps a relation $X \to Y$ to a relation $\underline{k} \to \underline{l}$, where $|X| = k$ and $|Y| = l$. Call this prop fRel_+.

(b) The smc $\mathsf{Rel}_\times$ is not a prop but we can (and will) choose a set to define a symmetric monoidal subcategory of it that is a prop. Given a set X, we can consider the category whose morphisms $k \to l$ are relations $X^k \to X^l$. Note that, with the Cartesian product as monoidal product this is not a prop, as it is not *strictly* associative. But through the isomorphism $X^{k+l} \cong X^k \times X^l$ we can make this symmetric monoidal category into a prop Rel_X with monoidal product given on morphisms by:

$$r_1 \oplus r_2 = \left\{ \left(\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, \begin{pmatrix} x_1' \\ x_2' \end{pmatrix} \right) \mid (x_1, x_1') \in r \text{ and } (x_2, x_2') \in s \right\} \tag{2.53}$$

Furthermore, for every set X, Rel_X inherits the hypergraph structure of $\mathsf{Rel}_\times$. We will be particularly interested in subprops of $\mathsf{Rel}_\mathbb{N}$ whose morphisms preserve some of the algebraic structure present on $\mathbb{N}$.

There is also a non-symmetric version of a prop, called a *pro*. We will need it for technical purposes, in order to define some constructions on props rigorously.

Definition 26. A *pro* is a strict monoidal category with the natural numbers as objects and addition as monoidal product. A morphism of pro is a strict monoidal functor that is the identity on objects. Pros form a category we call Pro.

Remark 27. We can recover Prop as a full subcategory of the co-slice category P/Pro, where P is the pro of permutations: this is a disconnected groupoid whose morphisms $k \to l$ are the bijections $\underline{k} \to \underline{l}$. There is an obvious forgetful functor from Prop to the slice category P/Pro. Note that this functor is not essentially surjective and that props are the full subcategory of P/Pro for which the permutation action of P is *natural* in a sense that we will not make precise here. We refer the reader to [Zan15, Section 2] for a more comprehensive account.

Pro(p)s are often used as a syntax for one-sorted algebraic theories, generalising Lawvere's account [Law63] to the symmetric monoidal case, and originating in the work of MacLane [Mac65]. Here, one-sorted refers to the fact that the algebraic structure is defined over a single carrier object (which we depict as a single wire) unlike the theory of group actions, for example, which involves two types—a group and a set on which it acts. This is an essential feature of props as every object is isomorphic to a monoidal product of copies of the generating object. In the graphical calculus of monoidal categories, this means that every diagram is a box with n identical wires as input and m as output. Props have been used extensively to provide a resource-conscious syntax for open systems [Zan15, BSZ17, BCR17]. They are also fundamental in our approach to concurrent systems. We will be particularly interested in giving presentations (also called axiomatisations) of props in order to understand their morphisms in terms of their interactions with each other rather than in reference to a semantic universe.

2.2.1 Freely generated props

We will need the definition of a prop that is freely generated from a signature, that is, a set of generators and equations. The following is often considered folklore and, as a result, is left undefined in the literature. To our knowledge, the only formal accounts come from [Zan15] and [BCR17, section 5]. We will follow the development of the latter reference.

According to Definition 24, a prop T can be seen as a collection of operations $T(n, m)$ which we can organise into a functor $\mathbb{N} \times \mathbb{N} \to \mathsf{Set}$, where $\mathbb{N}$ is the discrete category with objects the natural numbers.

Definition 28. The category of *signatures* is the category $\mathsf{Set}^{\mathbb{N}\times\mathbb{N}}$ with functors as objects and natural transformations as morphisms.

Proposition 29. *There is a forgetful functor $U : \mathsf{Prop} \to \mathsf{Set}^{\mathbb{N}\times\mathbb{N}}$ sending a prop to its underlying signature. Moreover this functor is monadic: it has a left adjoint $F : \mathsf{Set}^{\mathbb{N}\times\mathbb{N}} \to \mathsf{Prop}$ and the category of algebras for the monad UF is equivalent to the category of props.*

Proof. This is a generalization of Lawvere's original result [Law63] on algebraic theories. The details can be found in [BCR17, Appendix A]. □

This result allows us to speak about the *free prop* $F\Sigma$ on a signature Σ. More intuitively, the prop freely generated by Σ has composites of elements of Σ, identities and braidings as morphisms, modulo the naturality laws of (strict) smc. In addition we can describe any prop using a presentation by generators and equations as the following proposition states.

Proposition 30. *Any prop is the coequaliser in* Prop *of two parallel prop morphisms of the form*

$$FE \underset{\rho}{\overset{\lambda}{\rightrightarrows}} F\Sigma$$

for two signatures E and Σ.

Proof. This is a consequence of the fact that Prop is a category of algebras over a monad and therefore the colimit completion of the associated Kleisli category [BW85, Section 3.3, Proposition 4]. Let $\epsilon : FU \to 1$ be the counit of the adjunction in proposition 29. If T is the prop at hand, let $\Sigma = U\mathsf{T}$ and $E = UFU\mathsf{T}$, with the two morphisms $\lambda = FU\epsilon_{\mathsf{T}}$ and $\rho = \epsilon_{FU\mathsf{T}}$. □

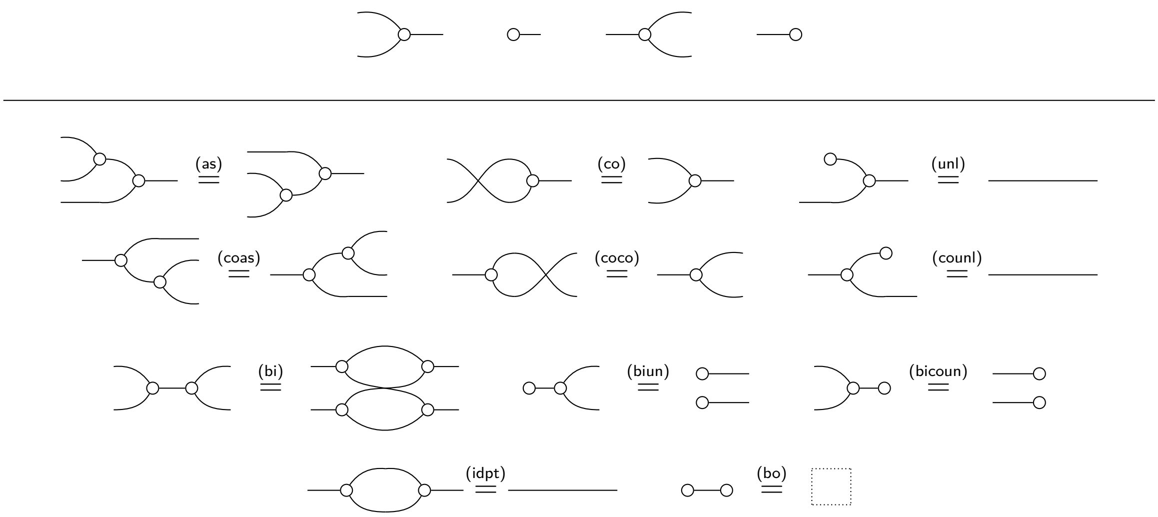


Figure 2.1: Presentation of $\mathsf{Bi}_{\mathbb{B}} \cong \mathsf{Rel}_+$.

Definition 31. We call the data of Σ, E and two parallel morphisms $\lambda, \rho : FE \to F\Sigma$, a *presentation*.

The elements of Σ serve as generators for T while those of E are the equations that hold between them. Thus, given e an equation in $E(n,m)$ the morphisms $FU\epsilon_{\mathsf{T}}(e)$ and $\epsilon_{FU\mathsf{T}}(e)$ in the free prop on Σ are equal in T. Proposition 30 guarantees that presentations of props in terms of generators and relations do indeed define props.

We give below two examples that illustrate Proposition 30.

Example 32. The prop fRel_+ is isomorphic to the prop $\mathsf{Bi}_{\mathbb{B}}$ freely generated by the signature for an idempotent commutative bimonoid. This can be seen as a consequence of Theorem 63 that we will cover in the following chapter. For reference, its presentation is given in Fig. 2.1: there are four generators at the top and ten equations below. Note that the symmetry morphism $\times$: $2 \to 2$ is not included in the generators since it is assumed in the structure of the prop. We can still have equations specifying its interaction with the generators, such as the commutativity equation (co) for the monoid.

Example 33. Another interesting prop is Frob, the prop of *extra-special commutative Frobenius algebras* in Fig. 2.2: This prop is the simplest hypergraph category as it contains only the required Frobenius structure. It is equivalent to the category CoRel of corelations [BG01, Zan16, CF17] whose objects are finite sets and morphisms $X \to Y$ are equivalence relations over $X + Y$ or, equivalently, jointly epic cospans, composed via pushout.

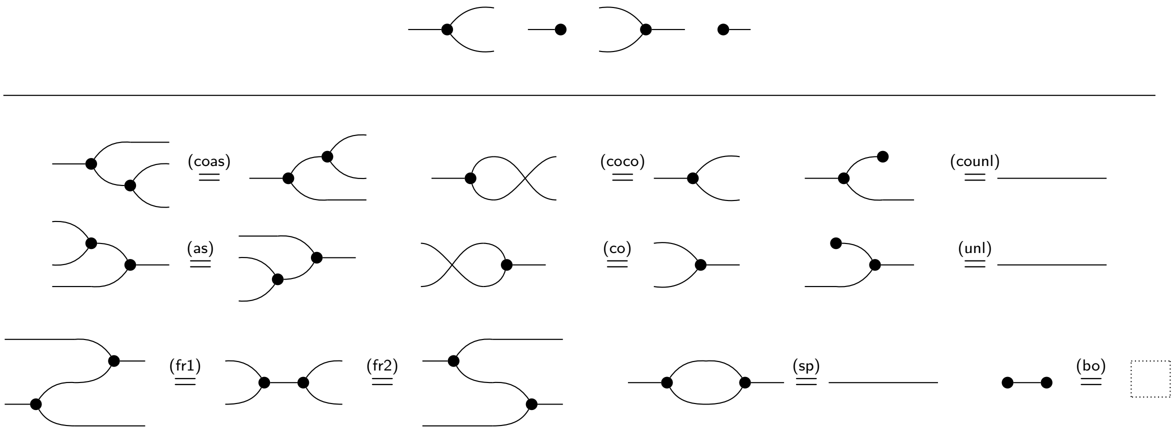
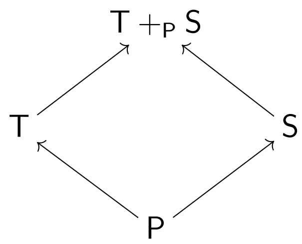


Figure 2.2: Presentation of $\mathsf{Frob} \cong \mathsf{CoRel}$.

2.2.2 Operations on props

Coproduct of props There are various ways to compose pro(p)s [Lac04]. The simplest is the sum. As we saw in remark 27, Prop is a full subcategory of the co-slice category P/Pro.

Proposition 34. *For* T *and* S *two props, the coproduct* $\mathsf{T}+\mathsf{S}$ *in* Prop *is given by the following pushout over* P *in* Cat*:*



Proof. See the proof of [Zan15, Proposition 2.10]. □

The pushout effectively identifies the action of permutations on the sum of the two props. We also have the following easy recipe for the coproduct of two props given their presentations:

Corollary 35. *If* T *and* S *are props with respective presentations* $(\Sigma_{\mathsf{T}}, E_{\mathsf{T}}, \lambda_{\mathsf{T}}, \rho_{\mathsf{T}})$ *and* $(\Sigma_{\mathsf{S}}, E_{\mathsf{S}}, \lambda_{\mathsf{S}}, \rho_{\mathsf{S}})$*, the coproduct* $\mathsf{T}+\mathsf{S}$ *is given by the prop with presentation* $(\Sigma_{\mathsf{T}} + \Sigma_{\mathsf{S}}, E_{\mathsf{T}} + E_{\mathsf{S}}, \lambda_{\mathsf{T}} + \lambda_{\mathsf{S}}, \rho_{\mathsf{T}} + \rho_{\mathsf{S}})$*.*

Proof. First, because (small) colimits commute with (small) colimits the coproduct of two coequalisers is the coequaliser of the coproduct:

$$FE_T + FE_S \underset{\rho_{\mathsf{T}}+\rho_{\mathsf{S}}}{\overset{\lambda_{\mathsf{T}}+\lambda_{\mathsf{S}}}{\rightrightarrows}} F\Sigma_{\mathsf{T}} + F\Sigma_{\mathsf{S}} \longrightarrow \mathsf{T}+\mathsf{S} \tag{2.54}$$

and since left adjoints preserve colimits, we have natural isomorphisms

$$F(\Sigma_{\mathsf{T}} + \Sigma_{\mathsf{S}}) \cong F\Sigma_{\mathsf{T}} + F\Sigma_{\mathsf{S}} \quad \text{and} \quad F(E_{\mathsf{T}} + E_{\mathsf{S}}) \cong FE_{\mathsf{T}} + FE_{\mathsf{S}} \tag{2.55}$$

from which we can conclude that $\mathsf{T}+\mathsf{S}$ is also a coequaliser for the following diagram, as required:

$$F(E_{\mathsf{T}} + E_{\mathsf{S}}) \rightrightarrows F(\Sigma_{\mathsf{T}} + \Sigma_{\mathsf{S}}) \longrightarrow \mathsf{T} + \mathsf{S} \tag{2.56}$$

□

Example 36. In [FC13] the authors use the sum of props to define a compositional theory of *directed acyclic graphs*. Taking the sum of the free props over a single $1 \to 1$ morphism with no equations, and the props for idempotent *bialgebras*, $\mathsf{Bi}_{\mathbb{B}}$ (Example 32), they are able to model the composition of directed graphs in a way that preserves the acyclicity condition. Swapping the prop of bimonoids for that of Frobenius monoids gives a prop of open graphs (without any acyclicity condition). This was first noticed in [RSW05].

Quotient of props Taking the coproduct of two props is often not very interesting because the two props do not interact. This is why we occasionally need to add more equations to enforce some behaviour or compatibility between their morphisms. At the level of the presentation, this amounts to simply adding these equations to the signature.

Definition 37. Let T be a prop presented by $(\Sigma, E, \lambda, \rho)$, and X a set of equations between morphisms of T, i.e., a pair of maps $l, r \colon X \to F\Sigma$. Then, the codomain of the coequaliser

$$FE + FX \underset{\rho+r}{\overset{\lambda+l}{\rightrightarrows}} F\Sigma \longrightarrow \mathsf{T}_{/X} \tag{2.57}$$

is $\mathsf{T}_{/X}$ *the prop* T *quotiented by the equations of* X.

2.3 Interacting Hopf algebras: the calculus of linear relations

While it is not strictly necessary, it is useful to contrast some of the work in this thesis with the graphical calculus of linear relations, developed in [BSZ17, Zan15]. This work largely inspired ours and we will sometimes refer to it when contrasting it with our calculus. Key differences will be highlighted.

Fix a field $\mathbb{K}$.

Definition 38. A *linear relation* $k \to l$ is a linear subspace $R \subseteq \mathbb{K}^k \times \mathbb{K}^l$.

Linear relations form a subprop $\mathsf{LinRel}_{\mathbb{K}}$ of $\mathsf{Rel}_{\mathbb{K}}$, i.e., they are stable under relational composition and monoidal product.

In [BSZ17], the authors give a presentation of this prop that they name *Interacting Hopf algebras*, $\mathsf{IH}_{\mathbb{K}}$. The main ingredients are two (extra-special commutative) Frobenius monoids that interact as a Hopf bimonoid.

Definition 39. Let $\mathsf{IH}_{\mathbb{K}}$ be the prop freely generated by the signature given in Fig. 2.3.

This gives a sound and fully complete axiomatisation of linear relations.

Theorem 40. $\mathsf{LinRel}_{\mathbb{K}}$ *is isomorphic to* $\mathsf{IH}_{\mathbb{K}}$.

When $\mathbb{K}$ is the field of fractions of a polynomial ring, the presentation gives a complete graphical syntax for *signal flow graphs* [BSZ14, BSZ15, FSR16], a notation commonly used in control theory to represent the behaviour of linear time-invariant dynamical systems.

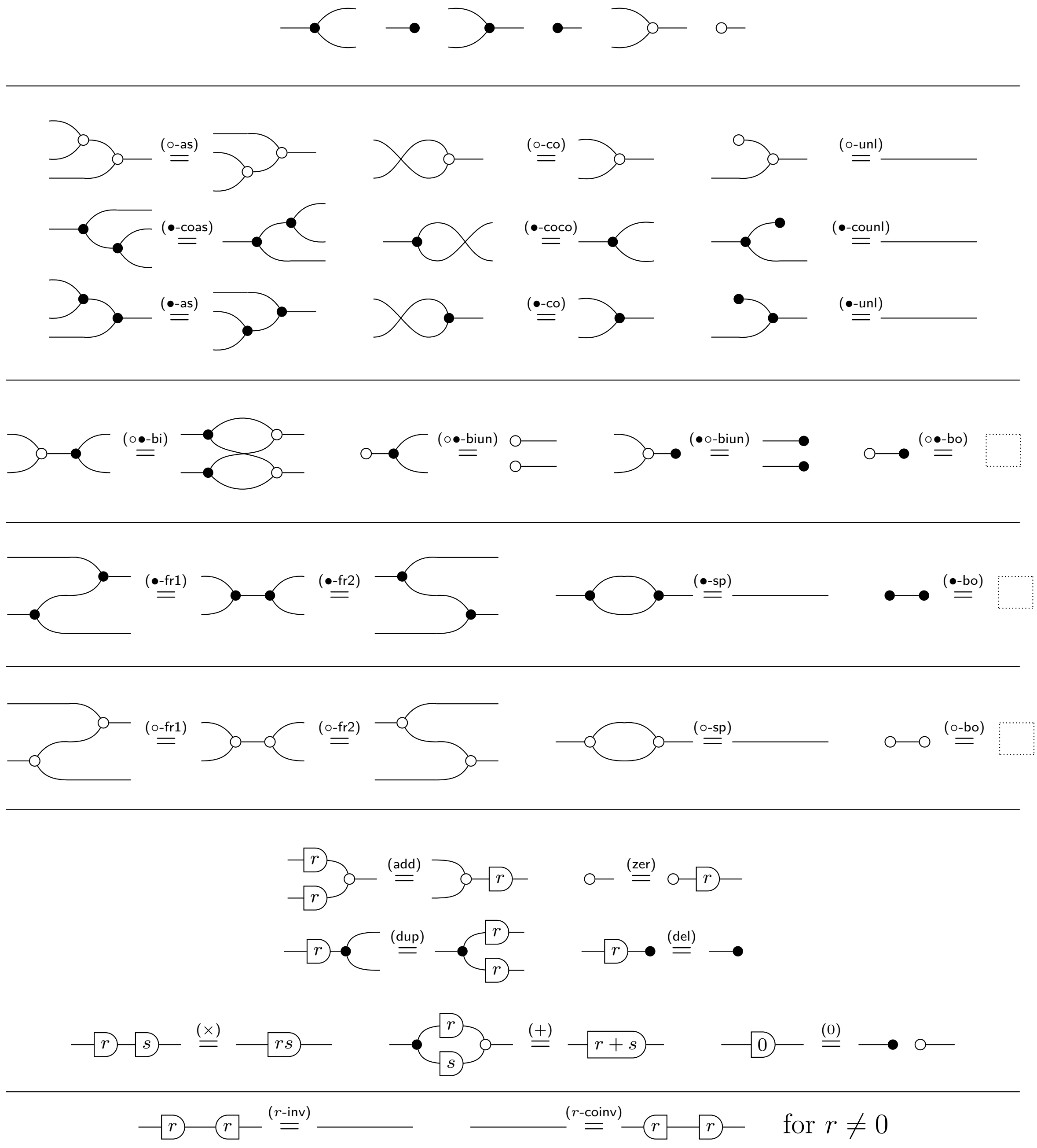


Figure 2.3: Axioms of Interacting Hopf Algebras ($\mathsf{IH}_{\mathbb{K}}$).

Chapter 3

The algebra of non-determinism and synchronisation

3.1 Overview

Concurrent programming is the art of designing synchronisation mechanisms through which different processes coordinate access to shared resources. Because of the inherent unpredictability of the order of events in a distributed system [Lam78], these synchronisation mechanisms have to deal with potentially non-deterministic threads of execution. The aim of our work is to study this interplay of synchronisation and non-determinism with algebraic tools.

Our approach takes after the component-based approach to software engineering: we separate systems into three categories of elements with different purposes and sets of concerns, namely data, processors (also called components) and connectors. In this thesis, we will focus on the latter category. Connectors provide the necessary coordination layer between potentially heterogeneous components that exchange data through their interfaces. Their behaviour has to be specified independently of any particular implementation and of the components that they connect.

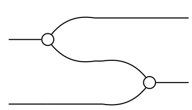
In this chapter, we develop a prop of connectors that will constitute the basis for a coordination language that we will extend in subsequent chapters. The morphisms of this prop are multiport channels that manipulate discrete resources (analogous to tokens in Petri nets) in order to specify certain synchronisation patterns at their boundary ports. Thus, they act as coordinators that specify a protocol between the components that they connect.

The main inspiration for the language we develop comes from Petri nets. In [BMM11, SMMB13] the authors develop a compositional treatment of nets in which they can synchronise with their environment through open transitions. Two nets can be composed in parallel or by connecting their boundary transitions. Thus they form a prop in which the closed morphisms (those of type $0 \to 0$) are the usual Petri nets. Some morphisms in this prop are connectors without places—just boundary transitions. This chapter can be seen as a microscopic analysis of these placeless nets, identifying the basic building blocks and their fundamental interactions. It will take us a little while, with a few detours along the way, to make the connection between the foundational work laid out in this chapter and the nets of [BMM11] precise. The reader will have to wait until Section 5.2.3, after we have introduced a notion of state. In this chapter (and the next), the language we introduce is entirely stateless and therefore with limited expressiveness for the purpose of specifying real-world systems. However, we believe that the results of this chapter highlight more clearly the fundamental algebraic structures of synchronisation protocols in concurrency.

A first look at the prop of transitions of nets with boundaries reveals that two different types of behaviours coexist: a subprop of asynchronous and non-deterministic connectors and another of synchronous connectors. The concepts of synchronisation and non-determinism, both fundamental, can be integrated into a consistent picture of concurrency when interpreted in a resource-sensitive context, as we will now see.

Independently, the algebraic structure of each of these subprops is well understood. It will be their interaction that will interest us here. To explain this, let us come back to fRel_+, the prop of relations with the disjoint sum as monoidal product. It can be seen as a primitive model of non-deterministic computation and, as we saw in Example 32, is monoidally equivalent to the free prop on an idempotent commutative bimonoid. Following Fig. 2.1, we represent its generators by:

With these components, a relation can be represented unambiguously as a bipartite graph. For example, $R = \{(1,1),(1,2),(2,2)\}$ is given by


(3.1)

We can think of the graph as specifying the evolution of a process, from left to right: its state may change from i to j if and only if $(i,j) \in R$. We can imagine that a process is represented by a single token whose state is its position on the graph. It enters from one port on the left of the diagram and is routed non-deterministically to a port on the right, provided that the two ports are connected. Notice that there is nothing preventing us from imagining that there is more than one token flowing around the network, each representing a thread or a different process whose state may evolve non-deterministically, as specified by the connectivity of the graph.

With this multi-token interpretation, if the syntax stays the same, our semantics changes: to represent systems that can support an arbitrary number of processes, the relations $k \to l$ that we consider are not subsets of $\underline{k} \times \underline{l}$ any longer, but morphisms of $\mathsf{Rel}_{\mathbb{N}}$ (see Example 25 (b)), *i.e.*, subsets of $\mathbb{N}^k \times \mathbb{N}^l$. The generating morphisms of fRel_+ admit a particularly simple and elegant interpretation in $\mathsf{Rel}_{\mathbb{N}}$. They denote addition, zero and their transpose!

$$[\![\supset\!\circ\!-]\!] = \left\{ \left(\begin{pmatrix} n \\ m \end{pmatrix}, n+m \right) \mid (n,m) \in \mathbb{N}^2 \right\} \qquad [\![\circ\!-]\!] = \{(0, \bullet)\} \tag{3.2}$$

$$[\![-\!\circ\!\subset]\!] = \left\{ \left(n+m, \begin{pmatrix} n \\ m \end{pmatrix} \right) \mid (n,m) \in \mathbb{N}^2 \right\} \qquad [\![-\!\circ]\!] = \{(\bullet, 0)\} \tag{3.3}$$

So far, the expressive power of this calculus is rather limited, as the connectors behave completely asynchronously and processes never interact with each other. Changing the interpretation of our basic syntax does not actually change what we can express in it. In concurrency, interesting behaviour arises when different processes are allowed to interact, whether it is by accessing shared resources (*e.g.* a database), merging, or spawning new copies of themselves. In other words, we would like primitives to express how concurrent processes synchronise.

To capture this behaviour, we add synchronous connectors that we depict with a black structure:

In the process interpretation, the semantics of these connectors is straightforward—they represent duplicating, deleting, merging and spawning processes. The associated relations in $\mathsf{Rel}_{\mathbb{N}}$ are:

$$[\![-\!\bullet\!\subset]\!] = \left\{ \left(n, \begin{pmatrix} n \\ n \end{pmatrix} \right) \mid n \in \mathbb{N} \right\} \qquad [\![-\!\bullet]\!] = \{(n, \bullet) \mid n \in \mathbb{N}\} \tag{3.4}$$

$$[\![\supset\!\bullet\!-]\!] = \left\{ \left(\begin{pmatrix} n \\ n \end{pmatrix}, n \right) \mid n \in \mathbb{N} \right\} \qquad [\![\bullet\!-]\!] = \{(\bullet, n) \mid n \in \mathbb{N}\} \tag{3.5}$$

In categorical terms, we notice that these are simply the monoid and comonoid of the hypergraph structure in $\mathsf{Rel}_{\mathbb{N}}$, inherited from the Frobenius structure on $\mathbb{N}$ in $\mathsf{Rel}_\times$. As such, their behaviour is captured precisely by the axioms for extra-special commutative Frobenius monoids. For Petri nets, we will see that these provide the basic infrastructure that synchronises the firing of transitions depending on the availability of tokens in places.

To summarise, we have explained how non-determinism and synchronisation can be seen as part of the same syntax, provided that we shift to a resource-sensitive interpretation. The central idea is to keep track of the number of processes that are running by reasoning in $\mathsf{Rel}_{\mathbb{N}}$. However, we have not explained how the black and white connectors interact. The purpose of this chapter is to characterise precisely the expressive power of the above graphical syntax, that is, the subcategory of $\mathsf{Rel}_{\mathbb{N}}$ it generates. More importantly, we give a presentation of it, producing a sound and complete set of equations to reason about the behaviour of the corresponding systems.

Remark 41. The results of Sections 3.3, 3.4 and 3.6 have been published in [BHP+19], co-authored with Filippo Bonchi, Josh Holland, Paweł Sobociński and Fabio Zanasi. The precise formulation of the resource calculus (Definition 54) and the proof of its completeness for additive relations (Theorem 59) is the work of the author of this thesis. The rest of [BHP+19] was written collaboratively and the present author is greatly indebted to the vision and ideas of his co-authors for the presentation of the results of this chapter.

Note that the proof of Theorem 59 as stated in [BHP+19] is incorrect: it makes use a subtly weaker axiom scheme that is not sufficient to carry out the rewriting of diagrams into normal form as explained in Section 3.6. To obtain a terminating rewriting procedure we had to modify the axiom of the resource calculus slightly and it is still open whether the *statement* of Theorem 59 found in [BHP+19] is correct, as it could be proved via a different method. However, the author strongly believes that no such proof exists, that the resource calculus as presented in [BHP+19] is incomplete for additive relations and that the equations of the resource calculus as presented in this thesis are necessary to guarantee completeness.

3.2 Additive monoids

All the basic connectors that we introduced in the previous section share an essential trait: their semantics respect the additive structure of $\mathbb{N}$, in the sense that all of the associated relations contain zero and are closed under addition in $\mathbb{N}^k \times \mathbb{N}^l$ (defined componentwise).

In this section, we introduce key notions from the theory of finitely generated commutative monoids that we will need to characterise the subprop of $\mathsf{Rel}_{\mathbb{N}}$ generated by the basic black and white connectors.

The monoids in which we are interested can be thought of as discrete analogues of convex polyhedral cones in $\mathbb{R}^d$. Notice that $\mathbb{N}$ is free as a monoid and therefore, we do not need to introduce nonnegative linear combinations to define the equivalent of cones over $\mathbb{N}$. We can rely on the corresponding notion to be simply closed under addition and this is why we will prefer the term *additive monoid*.

Remark 42 (Notational convention)**.** We use boldface $\mathbf{a}, \mathbf{b}, \mathbf{c}, \ldots$ to denote natural number vectors and capital letters $A, B, C, \ldots$ for matrices.

Definition 43. An *additive monoid* is a finitely generated submonoid of $\mathbb{N}^d$ for some nonnegative integer d.

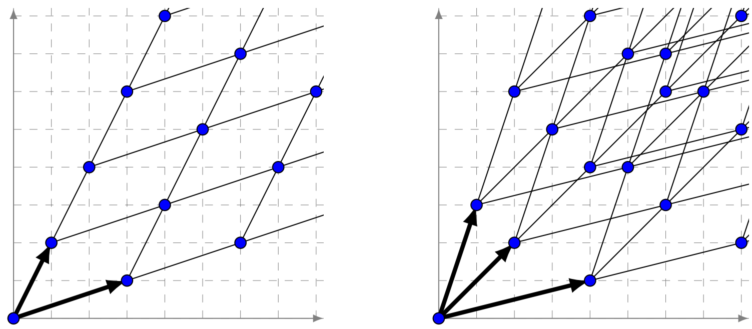
Additive monoids inherit a lot of properties from $\mathbb{N}^d$: they are

- commutative,
- nonnegative, i.e., $\mathbf{a} + \mathbf{b} = \mathbf{0}$ iff $\mathbf{a} = \mathbf{0}$ and $\mathbf{b} = \mathbf{0}$,
- cancellative, i.e., $\mathbf{a} + \mathbf{c} = \mathbf{b} + \mathbf{c}$ iff $\mathbf{a} = \mathbf{b}$, and
- torsion-free, i.e., $n\mathbf{a} = n\mathbf{b}$ iff $\mathbf{a} = \mathbf{b}$.

In fact, these properties are enough to fully characterise them (as finitely generated monoids) [RGS99, Theorem 3.11]. In the literature, additive monoids are most commonly known as *affine monoids*. We prefer the term additive, not only because it is more descriptive, but also because the term *affine* clashes with terminology that we will use later.

Since $\mathbb{N}$ is not Noetherian, not all submonoids of $\mathbb{N}^d$ are finitely generated. For example the monoid $M = \{(n, m) \mid n > m\} \cup \{(0, 0)\}$ is not. To see this, imagine that it were finitely generated, say by $(n_1, m_1), \ldots, (n_p, m_p)$. Let j be the index for which the ratio m_j/n_j is the largest of all the generating vectors (if there are several generating vectors with the same ratio, choose the one with the largest components). Then $(n_j + 1, m_j + 1)$ is in M but cannot be obtained as a sum of generating vectors, since $m_j + 1/n_j + 1 > m_j/n_j$.

Example 44. Like cones, additive monoids can be represented as subsets of $\mathbb{R}^d$. Their structure is generally more complicated than cones. The two graphs below represent the additive monoids generated by $\{(1, 2), (3, 1)\}$ and $\{(1, 3), (2, 2), (4, 1)\}$ respectively:



The reader will notice that the monoid on the right resembles a three-dimensional lattice deformed to fit a two-dimensional perspective.

As the previous example suggests, in general, we can fit lattices of arbitrary dimension into a fixed $\mathbb{N}^d$ and this is one of the features that distinguishes additive monoids from linear subspaces. In the vector space R^d, the dimension of subspaces is at most d. Not so for additive monoids: they can be generated by any number of points, potentially larger than d.

We now turn to the suitable notion of basis for additive monoids. The usual notion of linear dependence can be adapted to the context of linear algebra over the semiring $\mathbb{N}$. We say that a set of vectors is *dependent* if one of them is equal to a sum (with multiplicities) of the others. Otherwise the vectors are said to be *independent*.

Definition 45. A *basis* of an additive monoid is an independent generating set.

Differently from linear subspaces, bases of additive monoids are unique. The following theorem is well-known in the literature but we could not find a proof[1] so we give one below for completeness.

Theorem 46. *Every additive monoid admits a unique basis, called its Hilbert basis.*

Proof. Let $A \subseteq \mathbb{N}^d$ be an additive monoid. Let $A^* = A \setminus \{0\}$ and $H = A^* \setminus (A^* + A^*)$. Intuitively this is the set of *irreducible elements* of A, the non-zero elements that cannot be further decomposed as a sum of two non-zero elements of A. We claim that H is not only a generating set for A but that it is also independent. First, define $|\mathbf{a}| = \sum_{i=1}^{d} a_i$ to be the *magnitude* of $\mathbf{a} = \sum_{i=1}^{d} a_i \mathbf{e}_i$, with $\mathbf{e}_i$ the basis vectors of $\mathbb{N}^d$.

- H is a generating set: if $\mathbf{a} \in A^*$ is in H then we are done, so assume that it is not. Then it is reducible, i.e., there exists $\mathbf{b}, \mathbf{c} \in A^*$ such that $\mathbf{a} = \mathbf{b} + \mathbf{c}$. Then $|\mathbf{a}| > |\mathbf{b}|$ and $|\mathbf{a}| > |\mathbf{c}|$. Either $\mathbf{b}$ and $\mathbf{c}$ are irreducible or one of them can

[1] It is left as an exercise in the text that we used as a reference [RGS99].

be further decomposed into elements of A of smaller magnitude. Reasoning by induction on the magnitude of $\mathbf{a}$, we see that the decomposition procedure has to terminate and therefore every element of A can be expressed as a nonnegative sum of elements of H.

- H is uniquely minimal for inclusion, i.e., if G is a generating set, we have $H \subseteq G$. This is by construction, since irreducible elements cannot be expressed as the sum of any other elements of A.

□

Dickson's lemma [Dic13] (probably also known earlier to Gordan) is a remarkable result about subsets of $\mathbb{N}^d$ that we will need to prove that composite monoids are finitely generated (once we have defined composition).

Theorem 47 (Dickson-Gordan)**.** *Every subset $S \subseteq \mathbb{N}^d$ contains finitely many minimal points.*

The following lemma is equivalent to Theorem 47. Consider the set $\mathbb{N}^d$ with the product ordering inherited from $\mathbb{N}$.

Lemma 48. *Every set $\{x_i\}_{i\geq 0}$ in $\mathbb{N}^d$ such that $x_i \nleq x_j$ whenever $i < j$ is finite.*

Proof. We reason by induction on d. For $d = 1$ the result follows immediately from the fact that the usual order on $\mathbb{N}$ is a total order: the non-increasing condition means that $x_i < x_0$ for all $0 < i$. There are only finitely many such nonnegative integers.

Assume that the statement of the lemma is true for d and that there exists an infinite sequence $\{x_i\}_{i\geq 0}$ in $\mathbb{N}^{d+1}$ such that $x_i \nleq x_j$ whenever $i < j$. From this sequence we will construct an infinite non-increasing sequence in $\mathbb{N}^d$, a contradiction.

Let us write $\pi : \mathbb{N}^{d+1} \to \mathbb{N}$ for the projection onto the $(d+1)$th component. Let k_1 be the first index such that $\pi(x_{k_1})$ is minimal in the set $\{\pi(x_i)\}_{i\geq 0}$. Similarly let k_2 be the first index such that $\pi(x_{k_1})$ is minimal in the set $\{\pi(x_i)\}_{i\geq k_1+1}$. We can repeat this process to obtain a sequence $\{y_i = x_{k_i}\}_{i\geq 0}$ of elements of $\mathbb{N}^{d+1}$. Call $\pi_d : \mathbb{N}^{d+1} \to \mathbb{N}^d$ the projection onto the first d components. We claim that the sequence $\{\pi_d(y_i)\}_{i\geq 0}$ satisfies the condition in the statement of the lemma. Take i and j to be two indexes such that $i < j$. By construction $\pi(y_i) \leq \pi(y_j)$. Also by construction, we have $y_i = x_{k_i} \nleq x_{k_j} = y_j$ since $k_i < k_j$. This means that $\pi_d(y_i) \nleq \pi_d(y_j)$, as we wanted. □

Proof of Theorem 47. Using the lexicographic ordering (for instance) we can totally order the set of minimal points of S into a sequence $\{m_i\}_{i\geq 0}$ of elements of $\mathbb{N}^d$. In the product order they satisfy $m_i \nleq m_j$ whenever $i < j$ since they are incomparable. By Lemma 48, this set is finite. □

3.3 The prop of additive relations

We now show that our syntax is universal for a collection of relations that we call *additive relations*. We define them and show that they form a subprop of $\mathsf{Rel}_{\mathbb{N}}$.

Definition 49. An *additive relation* of type $k \to l$ is an additive monoid $R \subseteq \mathbb{N}^k \times \mathbb{N}^l$.

Remark 50. Note that in the definition above, we have identified $\mathbb{N}^k \times \mathbb{N}^l$ with $\mathbb{N}^{k+l}$ through the isomorphism $(\mathbf{a}, \mathbf{b}) \mapsto \begin{pmatrix}\mathbf{a}\\ \mathbf{b}\end{pmatrix}$. In general, we prefer to write additive relations as pairs of vectors to distinguish the domain from the codomain and avoid type errors. With the isomorphism $\mathbb{N}^k \times \mathbb{N}^l \cong \mathbb{N}^{k+l}$, all the usual operations on additive monoids are available: 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 $\langle(\mathbf{a}, \mathbf{b})\rangle = \{(n\mathbf{a}, n\mathbf{b}) \mid n \in \mathbb{N}\}$. More generally, for a finite set $G = \{(\mathbf{a}_1, \mathbf{b}_1), \cdots, (\mathbf{a}_p, \mathbf{b}_p)\}$ of points in $\mathbb{N}^k \times \mathbb{N}^l$,

$$\langle G \rangle = \left\{ \sum_{i=1}^{p} n_i(\mathbf{a}_i, \mathbf{b}_i) \mid n_1, \ldots, n_p \in \mathbb{N} \right\}. \tag{3.6}$$

is an additive relation.

By definition, additive relations, like additive monoids, are finitely generated. As a sanity check, we should verify that this is also the case for the semantics of our basic connectors:

$$[\![-\!\bullet\!\!\subset]\!] = \left\langle \left(1, \begin{pmatrix}1\\1\end{pmatrix}\right) \right\rangle \qquad [\![-\!\bullet]\!] = \langle (1, \bullet) \rangle \tag{3.7}$$

$$[\![\supset\!\!\circ\!-]\!] = \left\langle \left\{ \left(\begin{pmatrix}0\\1\end{pmatrix}, 1\right), \left(\begin{pmatrix}1\\0\end{pmatrix}, 1\right) \right\} \right\rangle \qquad [\![\circ\!-]\!] = \langle (0, \bullet) \rangle. \tag{3.8}$$

As they are just the converse relations, the generators for the other connectors can be obtained by reversing the order of the generating pairs.

We now show that additive relations form a subprop, AddRel, of $\mathsf{Rel}_{\mathbb{N}}$. We need to verify that they are closed under composition and monoidal product. The case of the monoidal product is straightforward: if $R : k \to l$ and $R' : k' \to l'$ are 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

$$\left\{ \left(\begin{pmatrix}\mathbf{a}_i\\ \mathbf{a}'_j\end{pmatrix}, \begin{pmatrix}\mathbf{b}_i\\ \mathbf{b}'_j\end{pmatrix} \right) \mid 1 \leq i \leq p, \text{ and } 1 \leq j \leq q \right\}. \tag{3.9}$$

The case of composition also goes through, but it is non-trivial.

Proposition 51. *The composition of two finitely generated additive relations is finitely generated.*

We first need to prove an intermediary result about the set along which two additive relations synchronise.

Definition 52. For U a $m \times k$ matrix and V a $m \times l$ matrix, both with coefficients in $\mathbb{N}$, let $T_{U,V} := \{(\mathbf{a}, \mathbf{b}) \in \mathbb{N}^k \times \mathbb{N}^l \mid U\mathbf{a} = V\mathbf{b}\}$ be the set of *transactions* of U and V.

Lemma 53. *For U an $m \times k$ matrix and V an $m \times l$ matrix, $T_{U,V}$ is an additive relation.*

Proof. $T_{U,V}$ is clearly a submonoid of $\mathbb{N}^k \times \mathbb{N}^l$ so the only property that remains to prove is that it is finitely generated.

By Dickson's lemma, $T_{U,V} \setminus \{(\mathbf{0}, \mathbf{0})\}$ has finitely many minimal elements for the product partial order. We call these minimal transactions and claim that every transaction is a sum of minimal transactions. As before, we define $|(\mathbf{a}, \mathbf{b})| = \sum_{i=1}^{k} a_i + \sum_{i=1}^{l} b_i$ to be the magnitude of the transaction $(\mathbf{a}, \mathbf{b}) = \left(\begin{pmatrix} a_1 \\ \vdots \\ a_k \end{pmatrix}, \begin{pmatrix} b_1 \\ \vdots \\ b_l \end{pmatrix}\right)$. We reason by induction on the magnitude of transactions. The statement is trivial for magnitude zero. Assume that every transaction of magnitude less than or equal to n can be decomposed as a sum of minimal transactions. Let $(\mathbf{a}, \mathbf{b}) \in T$ have magnitude $n+1$. If $(\mathbf{a}, \mathbf{b}) \in T$ is minimal we are done, so assume that it is not. Then there exists $(\mathbf{m}, \mathbf{n}) \in S$, minimal, such that $(\mathbf{m}, \mathbf{n}) \leq (\mathbf{a}, \mathbf{b})$. Hence, there also exists $(\mathbf{c}, \mathbf{d}) \in \mathbb{N}^d$ such that $(\mathbf{m}, \mathbf{n}) + (\mathbf{c}, \mathbf{d}) = (\mathbf{a}, \mathbf{b})$. We claim that $(\mathbf{c}, \mathbf{d}) \in T$: we have

$$(\mathbf{m}, \mathbf{n}) + (\mathbf{c}, \mathbf{d}) = (\mathbf{m} + \mathbf{c}, \mathbf{n} + \mathbf{d}) \tag{3.10}$$

and

$$U\mathbf{a} = U(\mathbf{m} + \mathbf{c}) = U\mathbf{m} + U\mathbf{c} \tag{3.11}$$

$$V\mathbf{b} = V(\mathbf{n} + \mathbf{d}) = V\mathbf{n} + V\mathbf{d} \tag{3.12}$$

And, since $U\mathbf{a} = V\mathbf{b}$ and $U\mathbf{m} = V\mathbf{n}$, we can deduce by cancellativity that $U\mathbf{c} = V\mathbf{d}$.

Finally, since $(\mathbf{c}, \mathbf{d})$ necessarily has smaller magnitude than $(\mathbf{a}, \mathbf{b})$, it is a linear combination of minimal transactions by the induction hypothesis and therefore so is $(\mathbf{a}, \mathbf{b})$. □

Proof of Proposition 51. Suppose that additive relations $R\colon k \to l$ and $S\colon l \to m$ have respective generating sets $\{(\mathbf{a}_1, \mathbf{b}_1), \cdots, (\mathbf{a}_p, \mathbf{b}_p)\}$ and $\{(\mathbf{c}_1, \mathbf{d}_1), \cdots, (\mathbf{c}_q, \mathbf{d}_q)\}$. We will find a generating set for $R\,;S$. Let $U = \left(\begin{smallmatrix} U_k \\ U_l \end{smallmatrix}\right)$ and $V = \left(\begin{smallmatrix} V_l \\ V_m \end{smallmatrix}\right)$ be the $(k+l) \times p$ and $(l+m) \times q$ matrices whose columns are the generating vectors of R and S, respectively. Then $(\mathbf{a}, \mathbf{b}) \in R$ iff there exists $\mathbf{f} \in \mathbb{N}^p$ such that $U\mathbf{f} = \left(\begin{smallmatrix} \mathbf{a} \\ \mathbf{b} \end{smallmatrix}\right)$, and similarly for S and V.

By Lemma 53, the commutative monoid $\{(\mathbf{e}, \mathbf{f}) \in \mathbb{N}^p \times \mathbb{N}^q \mid U_l\mathbf{e} = V_l\mathbf{f}\}$ is generated by finitely many minimal elements $(\mathbf{e}_1, \mathbf{f}_1), \ldots, (\mathbf{e}_d, \mathbf{f}_d)$. We claim that $\{(U_k\mathbf{e}_1, V_m\mathbf{f}_1), \ldots, (U_k\mathbf{e}_d, V_m\mathbf{f}_d)\}$ generates $R\,;S$.

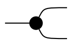
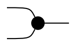
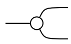
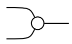
Let $(\mathbf{a}, \mathbf{c}) \in R\,;S$. By definition, there exits $\mathbf{b} \in \mathbb{N}^l$ such that $(\mathbf{a}, \mathbf{b}) \in R$ and $(\mathbf{b}, \mathbf{c}) \in S$. Thus, there exists $\mathbf{e} \in \mathbb{N}^p$ and $\mathbf{f} \in \mathbb{N}^q$ with $(U_k\mathbf{e}, U_l\mathbf{e}) = (\mathbf{a}, \mathbf{b})$ and $(V_l\mathbf{f}, V_m\mathbf{f}) = (\mathbf{b}, \mathbf{c})$. Let $(\mathbf{e}, \mathbf{f}) = \sum_{j=1}^n n_i(\mathbf{e}_i, \mathbf{f}_i)$ and finally $(\mathbf{a}, \mathbf{c}) = (U_k\mathbf{e}, V_m\mathbf{f}) = \sum_{j=1}^n n_i(U_k\mathbf{e}_i, V_m\mathbf{f}_i)$ as claimed. □

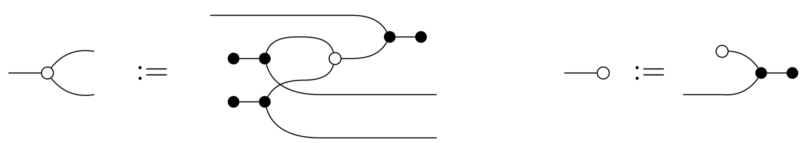
Note that AddRel is a hypergraph prop which inherits its hypergraph structure from $\mathsf{Rel}_{\mathbb{N}}$ (Example 25 (b)). It is therefore also self-dual compact closed by Proposition 23.

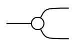
3.4 The resource calculus

We now introduce a prop, freely generated by a signature containing black and white generators, that we call the *resource calculus*. In Section 3.6 we will show that this prop is isomorphic to AddRel.

Definition 54. Let Rc be the prop freely generated over the signature in Fig. 3.1.

Remark 55. Because  ,  ,  and  form a Frobenius monoid, we can define  and  as the transpose of  and  using cups and caps:


(3.13)

This is why we do not need  and  as generators in the signature of Fig. 3.1.

A few comments are in order. As we can see, this signature is very similar to the one of $\mathsf{IH}_{\mathbb{K}}$ (Definition 39), but there remain a few crucial differences.

- In the first block, both the black and white structures are commutative monoids and comonoids, expressing fundamental properties of addition and copying.

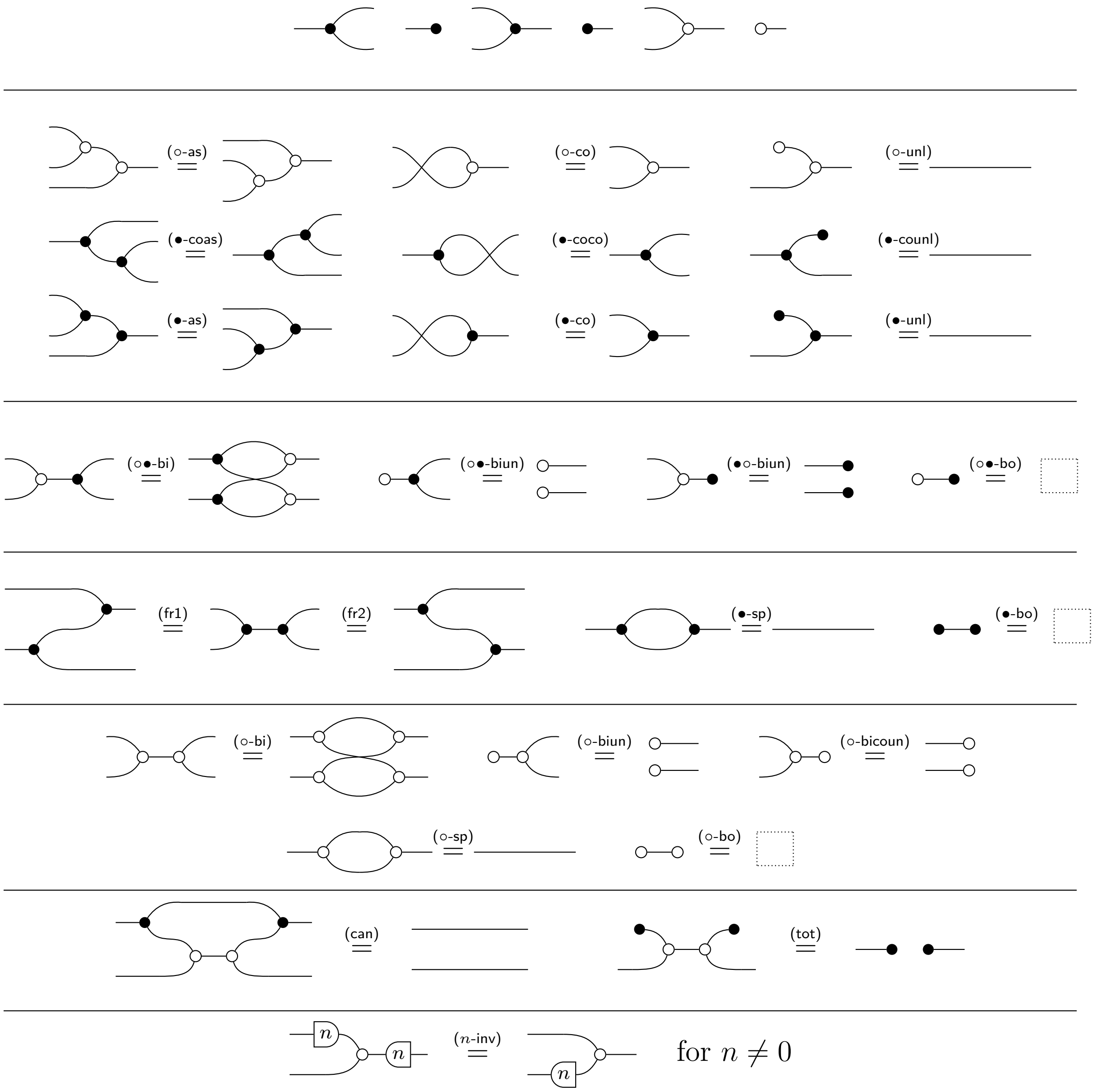
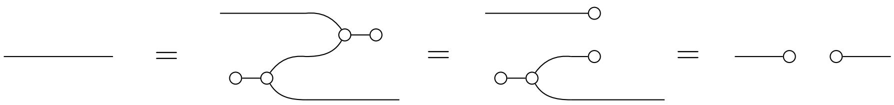


Figure 3.1: Axioms of the resource calculus.

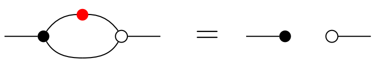
- In the second block, the white monoid and black comonoid interact as a bimonoid. Bimonoids are one of two canonical ways that monoids and comonoids interact, as shown in [Lac04].

- In the third block, the black monoid and comonoid form an extra-special Frobenius monoid (cf. Section 2.1.2.3). The Frobenius equations, together with the special equation, are the other canonical way in which monoids and comonoids interact, as identified in [Lac04].

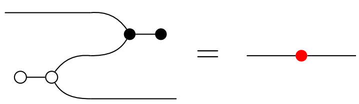
- In the fourth block, deviating from the equational theory of linear relations, the white monoid-comonoid pair forms a special bimonoid, not a Frobenius monoid. Here, the Frobenius structure—if present—would play the role of assuming the presence of additive inverses [CPV13, BPS17]. Since we are dealing with the natural numbers, the structure satisfies only the bimonoid equations. The key difference here is not the (○-bi) equation, which is also satisfied by any special commutative Frobenius monoid, but the (○-biun) equation. This equation witnesses the nonnegativity of elements in the underlying semiring: if $n + m = 0$ then both n and m must be zero. Note that this is consistent with the resource interpretation. We want to think of the numbers as processes or threads in a concurrent computation and there is no such thing as a negative process. This prevents the ○-operation from being Frobenius and therefore from giving rise to a second hypergraph structure on AddRel. Indeed if it were also Frobenius, we would get an inconsistent theory in which the identity separates and all morphisms of the same type are equal:


(3.14)

- In $\mathsf{IH}_{\mathbb{K}}$ the ●-○-bimonoid is also a Hopf monoid, i.e., there exists a morphism, called the *antipode*, satisfying

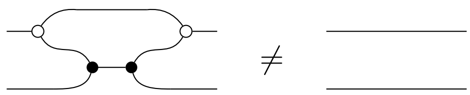


The antipode encodes the negation of $\mathbb{K}$ and comes for free from the interaction of the two Frobenius monoids:

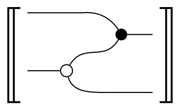


In the case of additive relations, this morphism cannot form an antipode because, as we saw earlier, the ○-cup reduces to two disconnected ○— . Once more, this is consistent with the resource interpretation of additive relations, as allowing an antipode would allow one to talk about negative resources.

- The (can) equation witnesses the cancellativity of addition. This equation holds for linear relations but the crucial difference is that, for additive relations, the colour-swap does not hold:



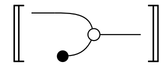
We can think of the following morphism as a form of controlled subtraction, with which we keep track of what we subtract:



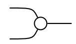
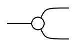
$$= \left\{ \left(\begin{pmatrix} a \\ b \end{pmatrix}, \begin{pmatrix} c \\ d \end{pmatrix} \right) \in \mathbb{N}^4 \mid a = c \text{ and } b = a + d \right\}$$

For a ring, this operation would be total but, because we lack additive inverses, it is only partial. Cancellativity is the next best thing after being a group[2]. Intuitively (can) means that adding something and then subtracting it can always be done but the converse may not be possible if we want to subtract more than we already have. Again, this makes sense when thinking of the elements of $\mathbb{N}$ as resources or processes.

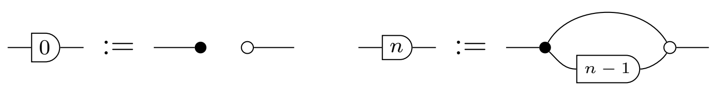
- Interestingly, the ○-structure being a bimonoid and not Frobenius means that we can reason about the natural order on the base semiring. Concretely, we can express the following relation:



$$= \{(n, m) \in \mathbb{N}^2 \mid n \leq m\}$$

The equation (tot) axiomatises its behaviour. Note that, if  and  satisfied the Frobenius equation, the diagram above would simply reduce to .

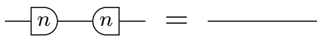
- Finally, the last equation is an axiom scheme, showing how the calculus accounts for the multiplicative structure of $\mathbb{N}$. It uses the following syntactic sugar, along with the obvious mirror image versions, defined inductively:



(3.15)

[2]All cancellative commutative monoids embed faithfully into an Abelian group by the same construction that builds the integers from the naturals, also known as the Grothendieck group construction in the context of K-theory.

They represent the additive relations of the form $\langle(1, n)\rangle$. We call these *scalars*. Semantically, the axiom scheme expresses the fact that $nk + r$ is a multiple of n if and only if r is also a mutliple of n. This provides the engine for a graphical version of Euclidean division. Note that (n-inv) implies that multiplication by n (for $n \neq 0$) has a one-sided inverse:

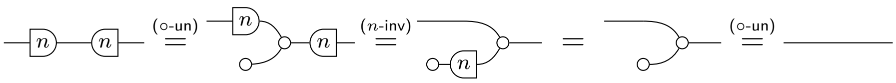


(3.16)

since we can easily prove by induction that

(3.17)

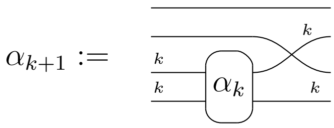
and therefore



(3.18)

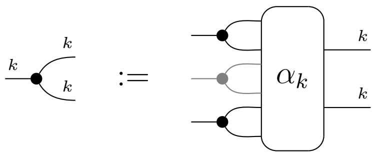
In $\mathsf{IH}_{\mathbb{K}}$, the converse axioms also hold. Here, they are not included in the signature for Rc since they are not sound for AddRel and rely on the ability to *divide* by non-zero scalars. As the semiring $\mathbb{N}$ does not contain multiplicative inverses, this is not possible in general.

Note that, since , , and form a Frobenius monoid, Rc is a hypergraph prop: for $k \in \mathbb{N}$ we can generalise , , and to k wires. Let $\alpha_0 :=$ and $\alpha_k : k + k \to k + k$ be given inductively by



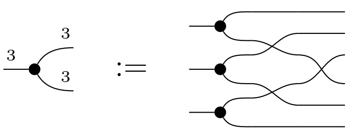
(3.19)

Then, let



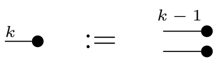
(3.20)

The dual connector can be defined similarly. For example,

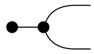
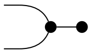


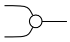
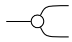
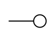
(3.21)

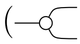
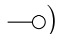
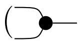
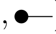
For , simply let



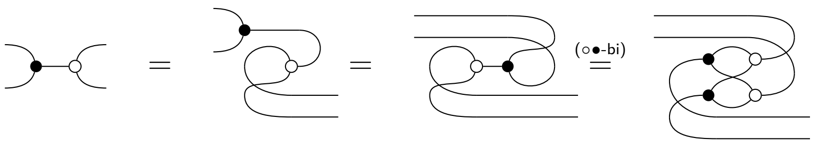
(3.22)

by induction, and similarly for the dual. And, since Rc is hypergraph, it is also self-dual compact closed by Proposition 23. We also use  and  to denote  and  respectively.

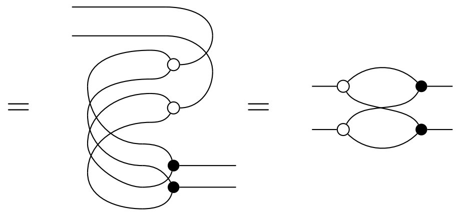
With the same construction, we can extend  ,  ,  and  to an arbitrary number of wires as well.

Proposition 56. *The monoid* (, ) *and comonoid* (, ) *form a commutative bimonoid.*

Proof. We prove the bimonoid law for the monoid-comonoid pair:



(3.23)

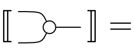
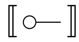


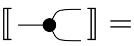
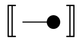
(3.24)

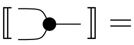
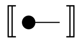
The equalities involving the unit and counit can be proven entirely analogously. □

It is time to reveal that the mapping $[\![-]\!]$ that we have been using informally so far can be made into a strict symmetric monoidal functor or, in other words, a prop morphism. First, by the universal property of free props, there is a unique way of extending the mapping of the generators to the entire resource calculus.

Definition 57. Let $[\![-]\!] : \mathsf{Rc} \to \mathsf{AddRel}$ be the unique mapping defined on the generators of Rc as:

 $= \left\{ \left(\begin{pmatrix} n \\ m \end{pmatrix}, n+m \right) \mid (n,m) \in \mathbb{N}^2 \right\}$  $= \{(0, \bullet)\}$ (3.25)

 $= \left\{ \left(n, \begin{pmatrix} n \\ n \end{pmatrix} \right) \mid n \in \mathbb{N} \right\}$  $= \{(n, \bullet) \mid n \in \mathbb{N}\}$ (3.26)

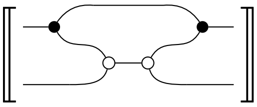
 $= \left\{ \left(\begin{pmatrix} n \\ n \end{pmatrix}, n \right) \mid n \in \mathbb{N} \right\}$  $= \{(\bullet, n) \mid n \in \mathbb{N}\}$ (3.27)

To show that it is indeed a prop morphism, we need to check that the equations of the resource calculus are sound for additive relations.

Proposition 58. $[\![-]\!] : \mathsf{Rc} \to \mathsf{AddRel}$ *is a prop morphism.*

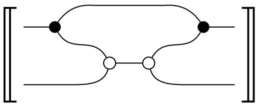
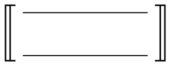
Proof. We need to verify that the equations of Rc are satisfied in AddRel. Most of them are straightforward: the •-structure is mapped to the Frobenius monoid inherited from $\mathsf{Rel}_\times$, addition is a commutative monoid and, with the Frobenius structure, it gives a bimonoid because it is a function (total and single valued). The two laws (∘-sp) and (∘-bo) are immediate; (∘-biun) and (∘-bicoun) witness the nonnegativity of natural numbers as we explained above. There are a few equations worth proving explicitly because they play an important role in the axiomatisation.

- First, we have

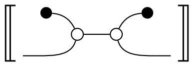


$$= \left\{ \left(\begin{pmatrix} n \\ m \end{pmatrix}, \begin{pmatrix} n' \\ m' \end{pmatrix} \right) \mid n = n' \text{ and } n + m = n' + m' \right\} \tag{3.28}$$

Substituting n for n' and using the cancellativity property of $\mathbb{N}$, $n+m = n+m'$ implies $m = m'$. Thus,

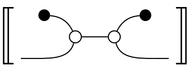
 $=$  . (3.29)

- Next, we have

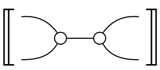


$$= \{(n, m) \mid \text{ there exist } p,\ q \text{ such that } n + p = m + q\}. \tag{3.30}$$

Clearly, any pair of naturals has (infinitely many) natural numbers greater or equal to both. It follows that the relation is total, in other words:

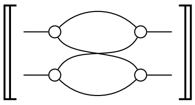
 $=$  . (3.31)

- Finally,



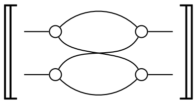
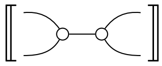
$$= \left\{ \left(\begin{pmatrix} n \\ m \end{pmatrix}, \begin{pmatrix} n' \\ m' \end{pmatrix} \right) \mid n + m = n' + m' \right\} \tag{3.32}$$

and

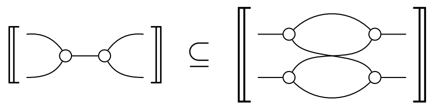


$$= \left\{ \left(\begin{pmatrix} n_1 + n_2 \\ m_1 + m_2 \end{pmatrix}, \begin{pmatrix} n_1 + m_1 \\ n_2 + m_2 \end{pmatrix} \right) \mid n_1, n_2, m_1, m_2 \in \mathbb{N} \right\} \tag{3.33}$$

By commutativity of addition, $n_1 + n_2 + m_1 + m_2 = n_1 + m_1 + n_2 + m_2$ so

 $\subseteq$  (3.34)

For the converse inclusion we can distinguish several cases. First, if $n = n'$ then $m = m'$ by cancellativity and we can just set $n_1 = n, n_2 = 0$ and $m_1 = 0, m_2 = m$. Otherwise, we can assume without loss of generality that $n' > n$. Then there exists $k > 0$ such that $n + k = n'$. So $n + m = n + k + m'$ and thus $m = m' + k$. Now, if $n = m'$, set $n_1 = n, n_2 = 0$ and $m_1 = m', m_2 = k$. Otherwise, we can assume without loss of generality that $m' > n$, i.e., that there exists p such that $n + p = m'$. In this case, set $n_1 = n, n_2 = 0$ and $m_1 = k, m_2 = n + p$. As a result,



(3.35)

□

The main technical result of this chapter is the following, establishing the full completeness of the resource calculus for additive relations.

Theorem 59. $[\![-]\!] : \mathsf{Rc} \to \mathsf{AddRel}$ *is an isomorphism of props.*

Fullness in this case means that every additive relation can be expressed as a resource calculus diagram, while completeness states that the functor $[\![-]\!]$ is faithful, i.e., that whenever two diagrams have the same interpretation as an additive relation, they are equal in the resource calculus. The proof of Theorem 59 can be found in Section 3.6. First, we need to take a detour through the simpler prop of matrices with coefficients in a semiring and explain how to obtain a sound and complete calculus for it.

3.5 Necessary detour: matrices over a semiring

We fix a semiring R which we assume commutative. This section is dedicated to a presentation of the prop Mat_R of matrices over R. We rely on the result of [Zan15]—itself an extension of the work of [Lac04]—in which the author gives a presentation of the theory of matrices over a *principal ideal domain*. However, his proof does not make use of additive inverses and can be adapted without changes to the case of a semiring. A more direct proof via a normal form argument is also found in [BE15] which is stated for the more restricted case of matrices with coefficients in a field. However, once more, the proof does not make any use of additive or multiplicative inverses and can be read directly as the proof of Theorem 63, below.

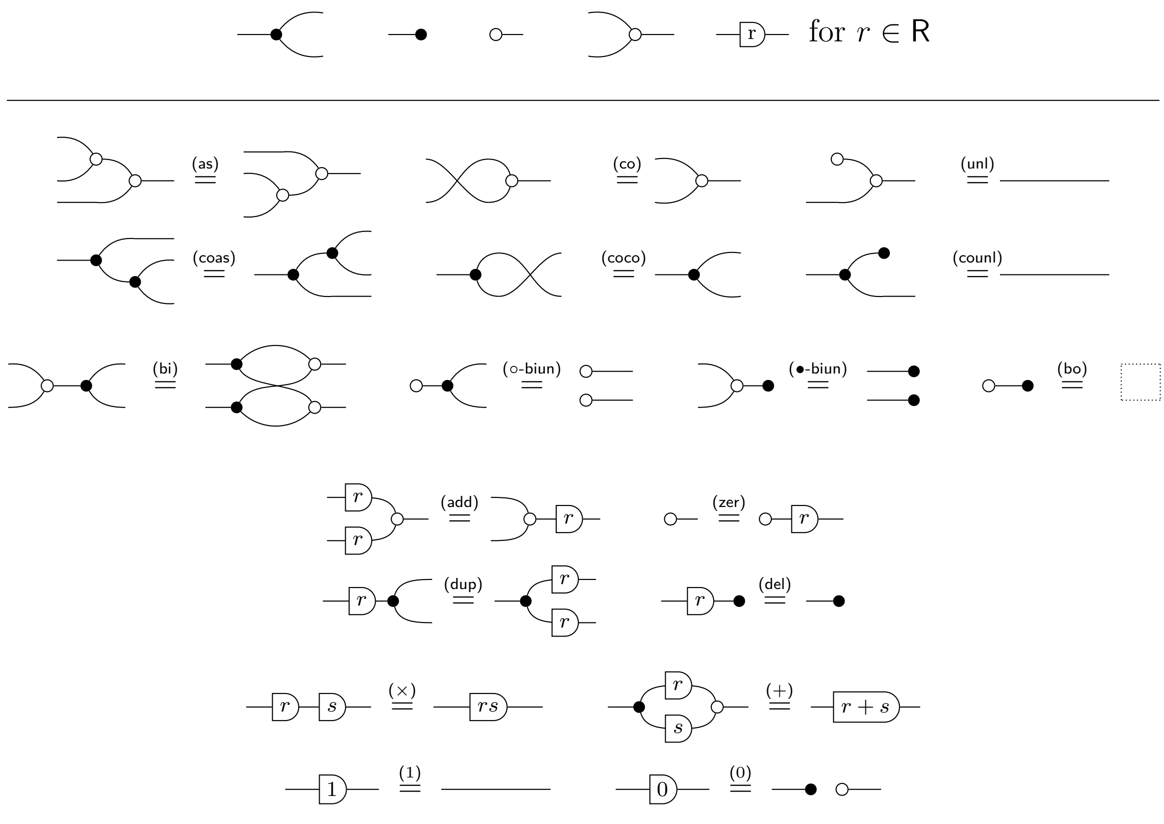


Figure 3.2: Presentation of Bi_{R}

Definition 60. Let $\mathsf{Mat}_{\mathsf{R}}$ be the prop in which morphisms $k \to l$ are $l \times k$ matrices with coefficients in R, the monoidal product is the direct sum and composition is matrix multiplication.

Proposition 61. *There is a prop embedding* $\iota : \mathsf{Mat}_{\mathsf{R}} \hookrightarrow \mathsf{AddRel}$.

Proof. The functor ι maps an $l \times k$ matrix A to the graph of the corresponding linear map: $\iota(A) = \{(\mathbf{x}, A\mathbf{x}) \mid \mathbf{x} \in \mathsf{R}^k\}$. It is clearly faithful. ☐

The corresponding equational theory is that of commutative bimonoids.

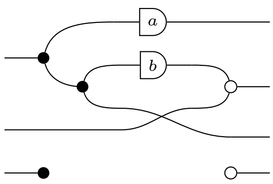
Definition 62. Bi_{R}, the prop of *commutative bimonoids* with scalars in R, is the prop freely generated by the signature in Fig. 3.2.

Simply from the signatures, we see that there is an embedding $\mathsf{Bi}_{\mathbb{N}} \hookrightarrow \mathsf{Rc}$.

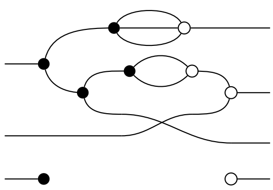
Theorem 63 (e.g. [Zan15, Prop. 3.9], or [BE15, Section 3])**.** Bi_{R} *is isomorphic to* $\mathsf{Mat}_{\mathsf{R}}$.

Remark 64. The axioms for Bi_R describe a commutative monoid, equipped with a multiplicative structure and its compatibility with the copying and deleting operations that come with any Cartesian category [Zan15, Theorem 2.42]. In [Zan15], the author deals with the more structured case of Hopf algebras for which they need the additive inverse of a ring. As we do not assume the existence of inverses, the bimonoid does not necessarily have an antipode. However, if R is a ring, the scalar $-1 : 1 \to 1$ is the antipode and the additional equation defining a Hopf algebra is a consequence of the $(+)$ law, encoding the inverse operation with the rest of the additive structure of R.

In order to develop some intuition for the isomorphism, let us demonstrate how matrices are represented diagrammatically. An $l \times k$ matrix A corresponds to a string diagram with k wires on the left and l wires on the right—the left ports can be interpreted as the columns and the right ports as the rows of A. The left jth port is connected to the ith port on the right through an r-weighted wire whenever A_{ij} is a nonzero scalar $r \in \mathsf{R}$. When the A_{ij} entry is 0, they are disconnected. Since composition along a wire carries the multiplicative structure of R, we can simply draw the connection as a plain wire if $A_{ij} = 1$. For example,

The matrix $A = \begin{pmatrix} a & 0 & 0 \\ b & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$ is drawn as  (3.36)

In the particular case of $\mathsf{R} = \mathbb{N}$, if $A_{ij} = n$ the ith and jth ports are connected through n wires so that the matrix above with $a = 3, b = 2$ is represented by



(3.37)

Thus, for the naturals, the representation of matrices reduces to counting paths between left and right ports.

Remark 65 (Notational convention)**.** We call *matrix diagram* a diagram that is in the image of the embedding of $\mathsf{Bi}_\mathbb{N}$ into Rc. In light of the isomorphism of Theorem 63 it is harmless to identify a matrix diagram with the unique matrix it represents, and we will use the same capital letter to denote both the diagram and its corresponding matrix. As a result we will also identify features of matrix diagrams with those of the matrix it represents: for example, we will freely refer to rows and columns of a matrix diagram.

3.6 Full completeness

This section is dedicated to proving Theorem 59 in order to establish the full completeness of the resource calculus.

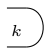
Fullness. To see that $[\![-]\!]$ is full, we simply have to notice that, for an additive relation $R\colon k \to l$ there exists $p \in \mathbb{N}$ and a $(k+l) \times p$ matrix A that represents R, in the sense that:

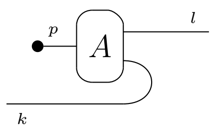
$$(\mathbf{a}, \mathbf{b}) \in R \text{ iff there exists } \mathbf{x} \in \mathbb{N}^p \text{ verifying } A\mathbf{x} = \begin{pmatrix} \mathbf{a} \\ \mathbf{b} \end{pmatrix} \tag{3.38}$$

Clearly, $\langle (\mathbf{a}_1, \mathbf{b}_1), \ldots, (\mathbf{a}_p, \mathbf{b}_p) \rangle$ is represented by the matrix

$$\begin{pmatrix} \mathbf{a}_1 & \mathbf{a}_2 & \ldots & \mathbf{a}_p \\ \mathbf{b}_1 & \mathbf{b}_2 & \ldots & \mathbf{b}_p \end{pmatrix}$$

and the generating set may be recovered from every matrix by taking its set of columns. We call such a matrix a *representing matrix* for R.

As we saw in Section 3.5, we can use the isomorphism of Theorem 63 and the embedding of $\mathsf{Bi}_{\mathbb{N}}$ into Rc to obtain a diagram for the matrix A representing R. Finally, with the cap  we can bend the last k wires from the right to the left and we can use the unit $\bullet\!-$ of the Frobenius monoid to delete (which, semantically, corresponds to universal quantification) the leftmost p wires. The resulting diagram d_A is shown below, and it follows that $R = [\![d_A]\!]$.

$d_A :=$ 

(3.39)

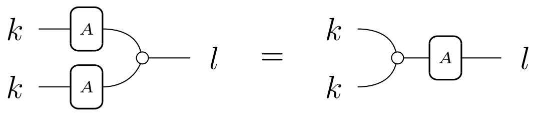
Faithfulness. Proving faithfulness of $[\![-]\!]$ is more complicated; as is often the case for completeness proofs, we use a normal form argument, inspired by the proof of completeness for linear relations in [BE15]. We proceed in two steps:

1. in Section 3.6.1 we show that every diagram can be rewritten into a prenormal form corresponding to a matrix representation, using only the equations of the resource calculus; then

2. in Section 3.6.2 we show that every diagram in prenormal form is equal to one for which the columns of the representing matrix are independent, that is, for which its columns are the Hilbert basis of the associated relation.

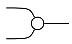
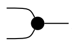
First, we need a few lemmas.

Lemma 66. *If A is the diagram of a $l \times k$ matrix, then*

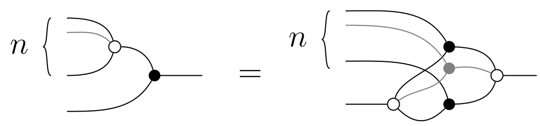


Proof. We could prove this by structural induction but it is easier to appeal to the completeness result of Theorem 63 and the fact that $A(\mathbf{x}+\mathbf{y}) = A\mathbf{x} + A\mathbf{y}$, for all $\mathbf{x}, \mathbf{y} \in \mathbb{N}^k$. □

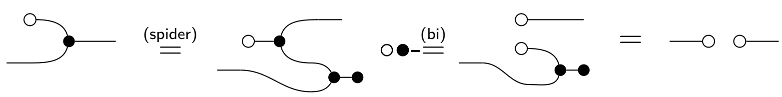
Regarding diagrams composed only of •-nodes, we will be as lax as the spider theorem allows us to be. Recall that, by Theorem 18, any two connected networks of •-monoid and •-comonoid are equal if and only if they have the same number of left and right ports.

We also need a lemma to handle interactions between  and .

Lemma 67. *For $n \in \mathbb{N}$,*

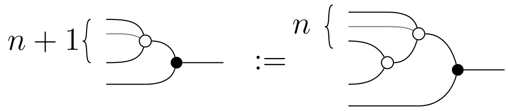


Proof. By induction on the number of input wires. The base case is simply



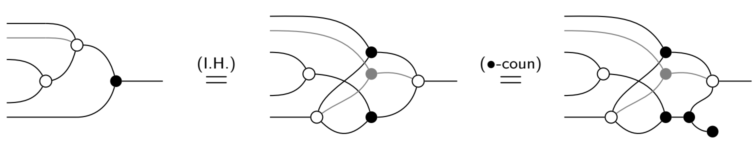
(3.40)

Assume that it holds for n legs. Then, for $n+1$ legs,

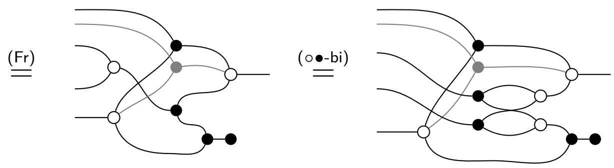


(3.41)

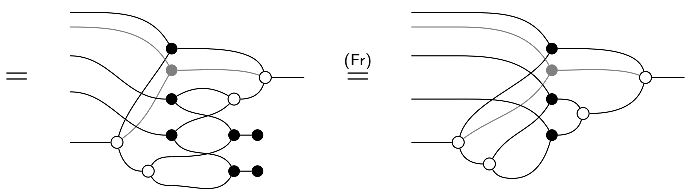
and, applying the induction hypothesis:



(3.42)

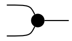


(3.43)

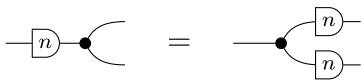


(3.44)

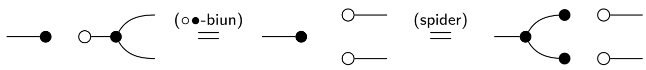
□

We will also need the fact that scalars commute with .

Lemma 68. *For $n \in \mathbb{N}$,*

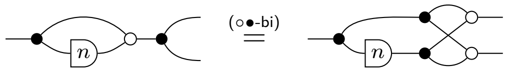


Proof. For $n = 0$,

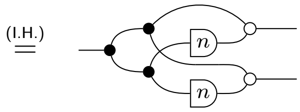


(3.45)

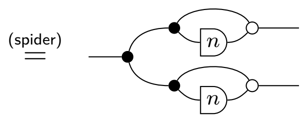
Assume that the lemma is true for $n \in \mathbb{N}$; then



(3.46)



(3.47)

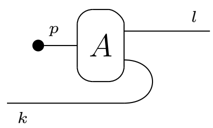


(3.48)

□

3.6.1 Prenormal Form

We want to show that every diagram in the prop is equal to one of the following form:



(prenormal form)

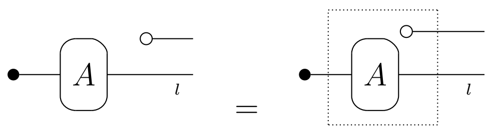
where A represents a $(k+l) \times p$ matrix (or, more precisely, is in the image of the embedding of $\mathsf{Bi}_{\mathbb{N}}$). We show this by structural induction on diagrams, where we analyse each possible case. This yields an effective procedure to rewrite every diagram in prenormal form using only the axioms of Rc.

While every matrix determines a unique additive relation, the converse is not true; there may be multiple matrices representing a given additive relation as their columns may be dependent. As a result, multiple diagrams in prenormal form correspond to the same additive relation. This is why the next section defines a normal form for which the set of generators is minimal in a suitable sense so that each diagram in normal form corresponds uniquely to an additive relation, proving the faithfulness of $[\![-]\!]$.

Remark 69. Since Rc is self-dual compact closed, there is a one-to-one correspondence between morphisms of type $k \to l$ and those of type $0 \to k + l$, by Proposition 15. Therefore, it suffices to consider diagrams with 0 domain for the rest of the proof; any other morphism can be obtained from this correspondence. In doing so, we avoid needlessly carrying around additional wires, which can clutter notation.

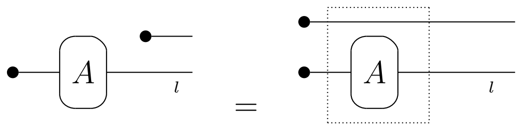
Let us come back to the prenormal form and consider the case of each generator. In the following three cases, the dotted box is in the image of the prop of $\mathbb{N}$-matrices and therefore the whole diagram is already in prenormal form.

1. For the monoidal product with zero:



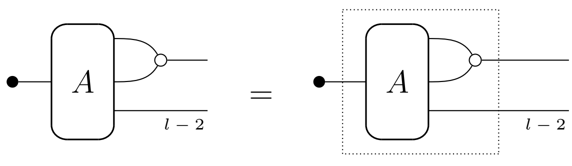
(3.49)

2. For the monoidal product with co-delete:



(3.50)

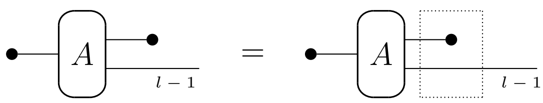
3. When two wires are added:



(3.51)

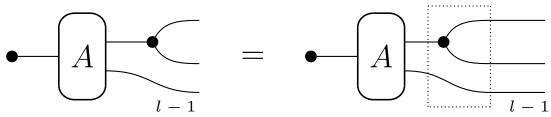
There are two other simple cases for which the matrix diagram A is composed with a generator already in matrix form. For these two cases we can appeal to the completeness of the diagrammatic calculus for matrices (Theorem 63) and compose the two matrices to obtain a diagram in prenormal form.

1. When a wire is deleted:



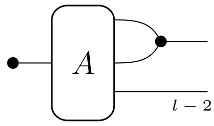
(3.52)

2. When a wire is copied:



(3.53)

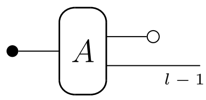
For the cases of the remaining generator, the resulting diagram is not immediately in prenormal form and some rewriting is necessary. For the case of a co-copying node joining two wires:



(3.54)

see Section 3.6.1.4. For the rewriting of the case, we will treat two subcases separately, even though they are not generators. They serve as lemmas for the co-copy case and we deal with them in their own subsections as well.

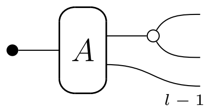
1. For when a co-zero node is connected to a wire:



(3.55)

   see Section 3.6.1.2.

2. For when a co-addition node is connected to a wire:



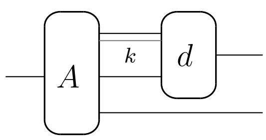
(3.56)

   see Section 3.6.1.3.

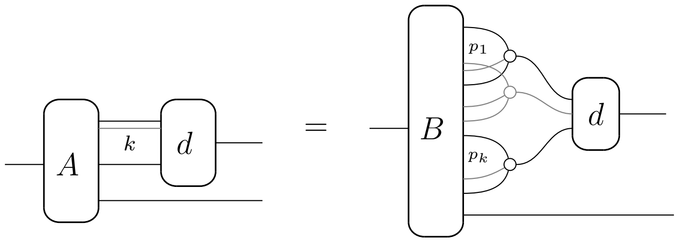
3.6.1.1 Connectedness

We need to define a simple termination measure on which we will be able to reason by induction in the next few sections. The rewriting procedure for the , and cases, relies on the use of the equations of Rc to slide certain diagrams past the layer of in the matrix block. To reason by induction and guarantee the termination of rewriting, we need to be able to count and bound the maximum number of to which a given diagram is connected, showing that it decreases after the inductive step.

Definition 70. The diagram d in



is $(p_1, \ldots, p_k)$ *A-connected*, for $p_i \in \mathbb{N}$ and $1 \leq i \leq k$, if there exists a matrix diagram B such that



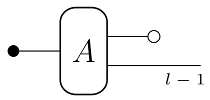
We call *A-connectedness* of d the unique maximal such k-tuple.

Remark 71. Note that, as defined, connectedness is a property of diagrams (i.e., terms of the syntax), not of the resource calculus morphism it represents. In particular, it is not invariant under the equations of Rc. In fact, reasoning by induction below, we will try to decrease the single component of the connectedness of and and alternatively each component of the connectedness of , using only the equations of Rc.

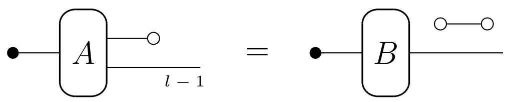
Let us come back to the treatment of the prenormal form. The next three sections deal with the three remaining cases.

3.6.1.2 Co-zero

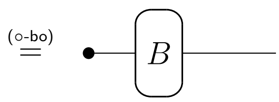
We will reason by induction on the A-connectedness of , in



For the base case, if its connectedness is zero, we have

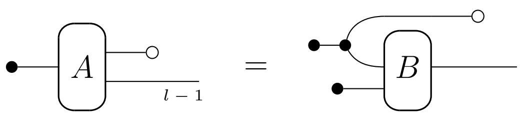


(3.57)

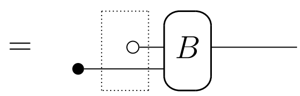


(3.58)

We also need the case where the connectedness is one:



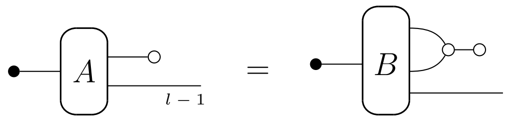
(3.59)

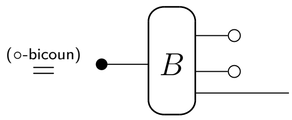


(3.60)

For the final step, we can multiply the matrix in the dotted box with B to obtain a diagram in prenormal form (appealing once more to the completeness theorem for the diagrammatic calculus of matrices).

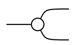
For the inductive step, assume that all diagrams of this form with up to n connectedness are equal to one in prenormal form. If  has $n+1$ A-connectedness, we have

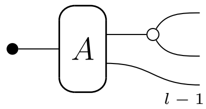
 (3.61)

 (3.62)

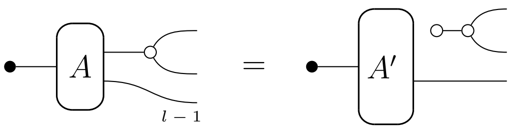
The top  has B-connectedness 1 while the lower one has B-connectedness n. By the induction hypothesis, we are done.

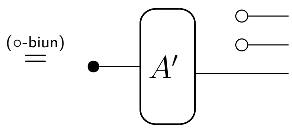
3.6.1.3 Co-addition

Again, we will reason by induction on the A-connectedness of  in

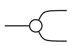


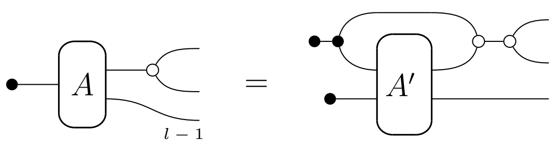
For the base case, if its connectedness is zero, we have

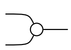
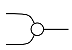
 (3.63)

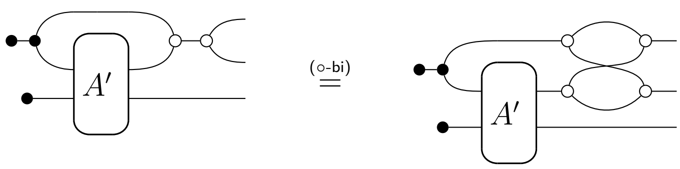
 (3.64)

which is in prenormal form.

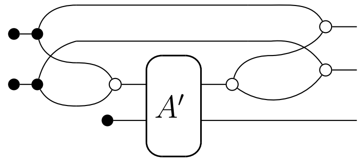
For the inductive step, assume that all diagrams of this form with n connectedness are equal to one in prenormal form. If  has $n+1$ A-connectedness, we have

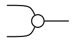
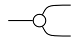
 (3.65)

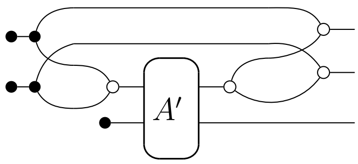
for some new matrix block A' and scalar a. Note that we have implicitly used the associativity of  to assume that all other  to the left of the rightmost one are connected to its lower leg. Now, we obtain

 (3.66)

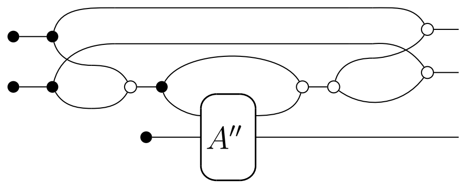
$$(3.67)$$

$=$  (3.68)

Here we should be careful of multiplying the  with the matrix diagram A' because we might obtain a matrix whose connectedness for  has increased. This can only happen if A'_{11}, the element in the first row and first column of A' is non-zero, i.e., if there is a path between the top left and right ports of the diagram for A'. If this is the case, there exists A'' such that:

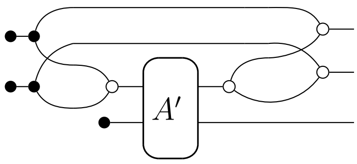


$=$

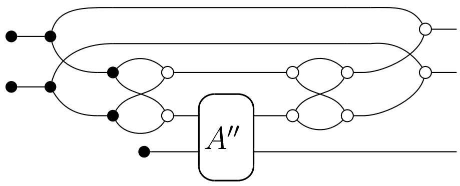


(3.69)

But then, we can iterate the process below as many times as there are paths between the two ports, until we obtain a matrix B with $B_{11} = 0$.

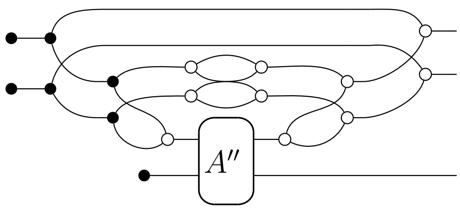


(○●-bi ;○-bi) $=$



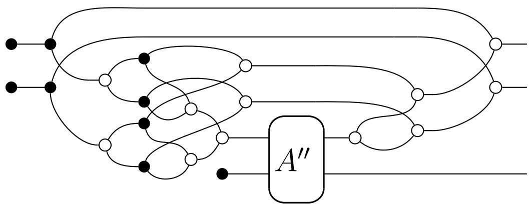
(3.70)

(○-bi) $=$



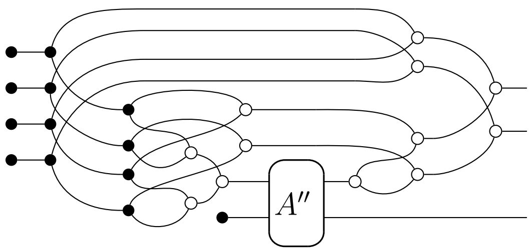
(3.71)

(Lemma 67) $=$



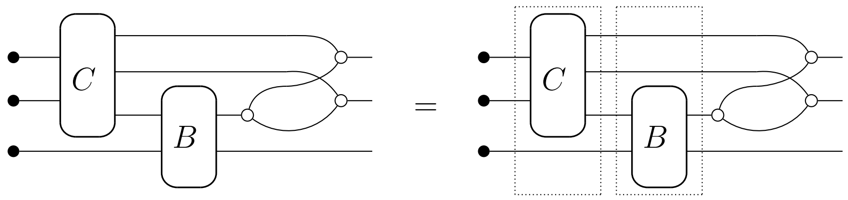
(3.72)

$=$

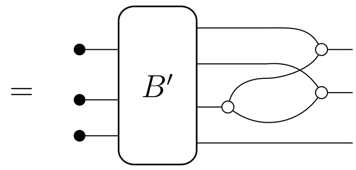


(3.73)

We repeat these steps until we have built matrix diagrams C and B as below, such that $B_{11} = 0$. When this is the case, we can safely multiply the two matrices without increasing the B-connectedness of :



(3.74)

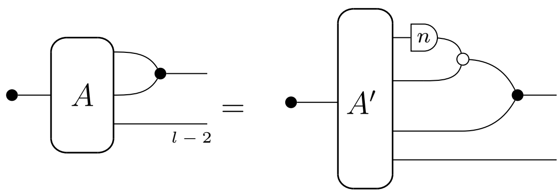


(3.75)

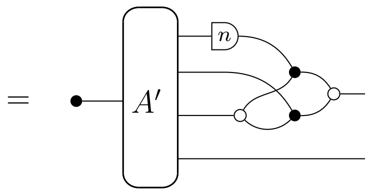
Here B' is the product of the two matrices within the dotted rectangles. Now has B'-connectedness n. We can therefore use the induction hypothesis to finish the proof.

3.6.1.4 Co-copying

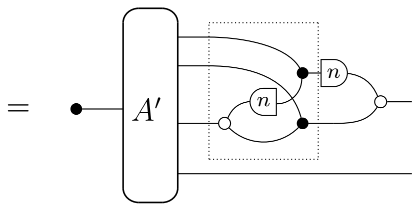
If n is a coefficient of the matrix represented by the diagram A below, we can find another matrix diagram A' such that:



(3.76)



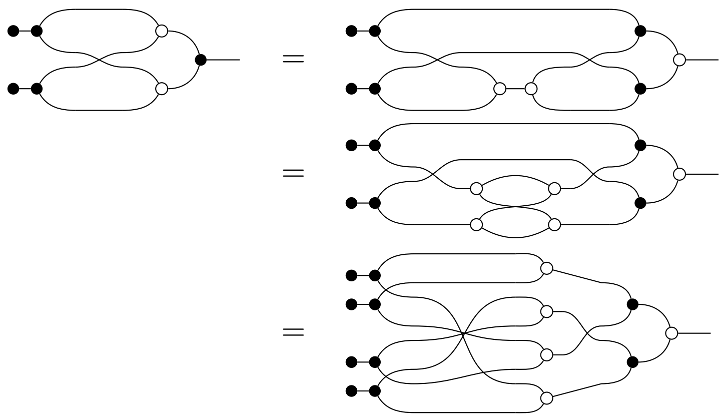
(3.77)



(3.78)

If we started with having $(n + n', k)$ A-connectedness, we end up with the highlighted diagram above having $(1, n', k)$ A'-connectedness. We would like to push and the scalar n into A' but we have to be careful that this does not increase the first two elements of the A'-connectedness (those corresponding to the top legs of the two). Indeed, we may be tempted to directly apply the procedure of Section 3.6.1.3 to absorb the , but this naïve procedure may not terminate.

To see this, we need to look at how we handled the case more closely. In equation (3.68), notice that we relied on the ability to slide along the on the left. In this process, it was rotated into a whose right port might be connected to the left leg of one of the two we have created. This may increase the connectedness of the diagram. We would find ourselves in the same situation as the one in which we started, having multiplied the number of that we need to eliminate on the right, without any bound on their connectedness. We would therefore be unable to guarantee termination. For concreteness, consider applying this naïve procedure to the following example:



(3.79)

(3.80)

(3.81)

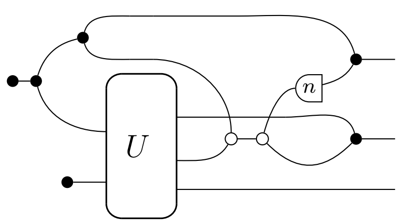
$$= \dots \tag{3.82}$$

The problem is that when we apply the elimination of of the previous section, we have to apply the same steps for every that the use of Lemma 67 in equation (3.73) has generated. But we cannot simply repeat the same procedure because the elimination of involves the application of the bimonoid law for the ○-bimonoid which might increase the connectedness of the diagram, preventing us from reasoning inductively. This is typical of the behaviour of the bimonoid law which increases the complexity of diagrams and must be handled with care in any form of rewriting.

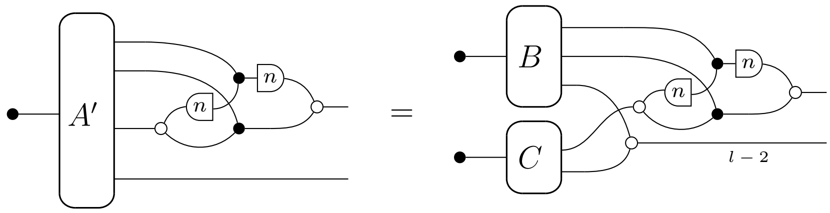
How can we circumvent this limitation? We need to be able to eliminate the that are going to increase the connectedness of the diagram. This can be done by eliminating loops in the following sense.

Definition 72. We say that a diagram d contains a *loop* when there exists a matrix diagram U such that

$d =$

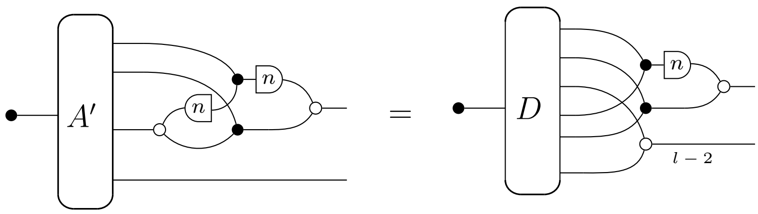


Loop-free diagrams factorise into a form that allows us to bound the connectedness of in the inductive step. This is because, if there are no loops in the diagram on the left of the equation below, there exist two matrix diagrams B and C such that



(3.83)

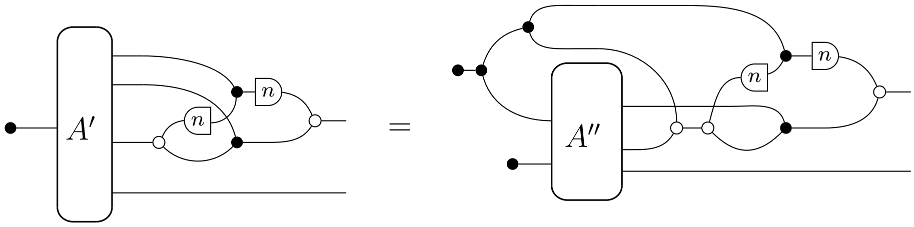
We will show below how to rewrite such a diagram into prenormal form, using the procedure on Section 3.6.1.3. Once the and scalar n are absorbed, we end up with a matrix D such that



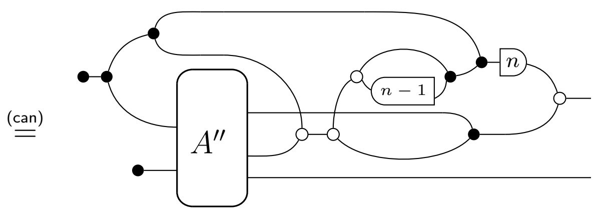
(3.84)

If we started with a with $(n + n', k)$ A-connectedness, and provided there are no loops, we end up with two : one with $(1, p)$ D-connectedness and the other with (n', p') D-connectedness for some $p, p' \in \mathbb{N}$. Thus we can apply the induction hypothesis.

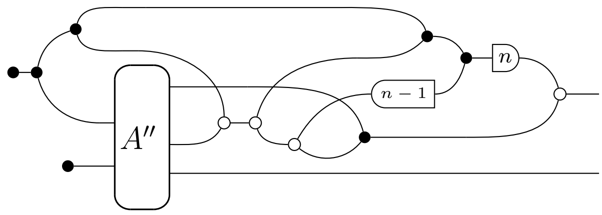
Getting rid of loops. Notice that, to avoid nontermination in the previous example, we could have used (can) in equation (3.79) instead. In fact, we can always use this strategy. If there are loops after equation (3.78), by definition there exists a matrix diagram A'' such that



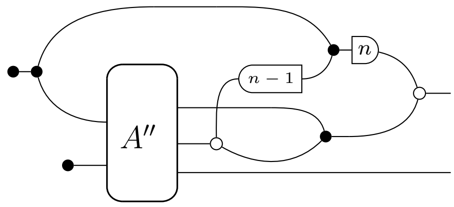
(3.85)



(3.86)

$\stackrel{\text{(can)}}{=}$ 

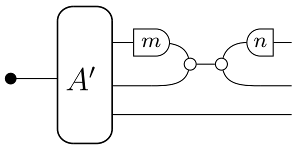
(3.87)

$\stackrel{\text{(can)}}{=}$ 

(3.88)

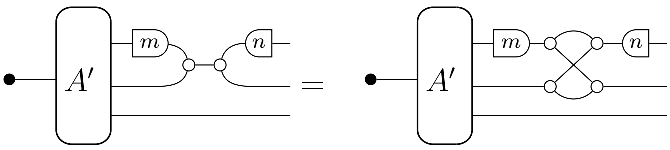
We can repeat this process in order to eliminate all loops. We will now explain how to deal with the —◦⊂ and n in a loop-free diagram.

After eliminating loops we need to be able to rewrite the following diagram into prenormal form:

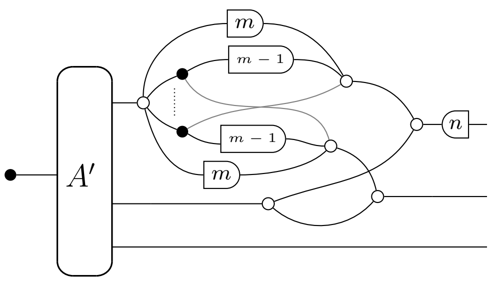


There are two cases.

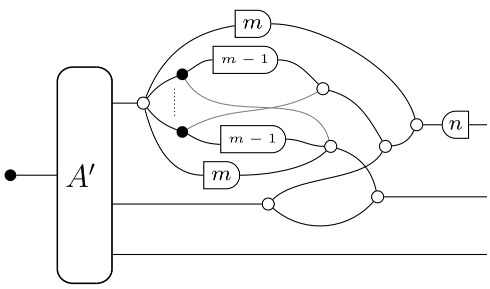
- Case 1: $m \geq n$. We have:



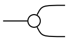
(3.89)

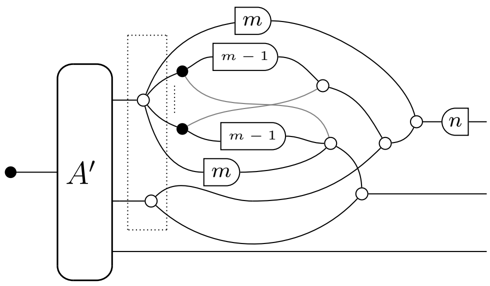
= 

(3.90)

= 

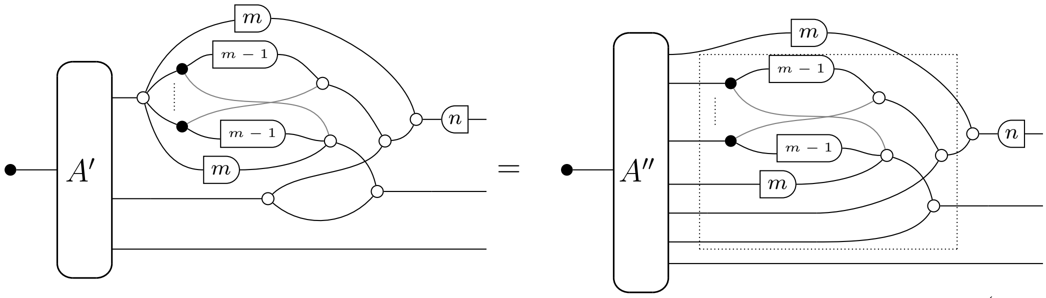
(3.91)

where (3.90) is the result of applying the rewriting procedure of Section 3.6.1.3 to the scalar followed by —◦⊂ (after potentially removing redundant generators as explained in Section 3.6.2). Then, we can apply the rewriting procedure of Section 3.6.1.3 to absorb all the  of the highlighted diagram below:



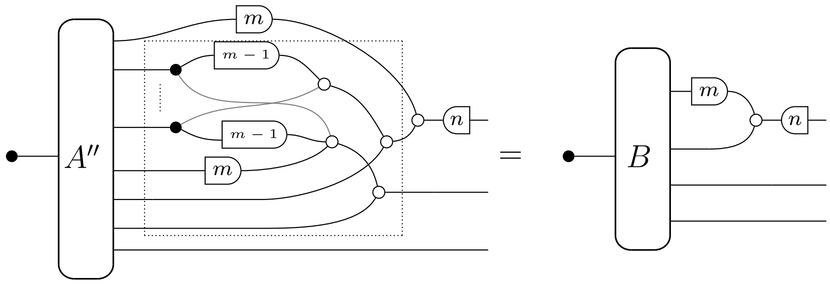
(3.92)

As a result we can find a new matrix diagram A'' such that



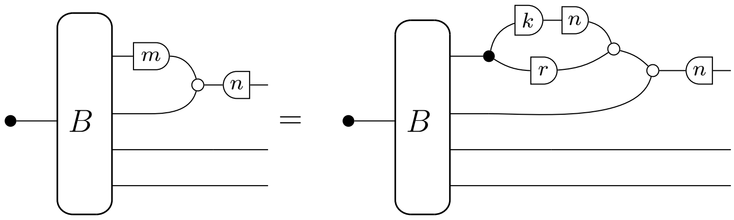
(3.93)

Now, the highlighted diagram above is in matrix form so we can find a new matrix diagram B such that

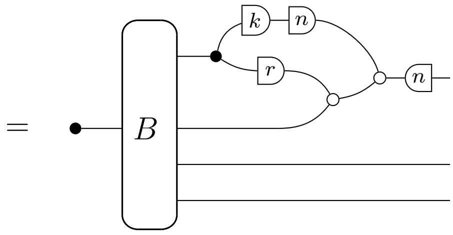


(3.94)

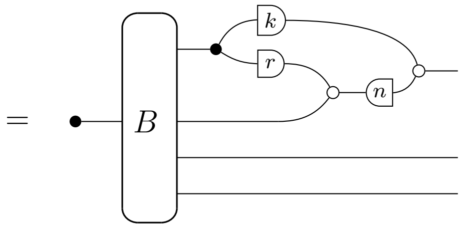
If $m \geq n$, then there exists $k, r \in \mathbb{N}$ such that $m = nk + r$ with $r < n$. So, we can apply the new axiom (n-inv) as follows:



(3.95)



(3.96)



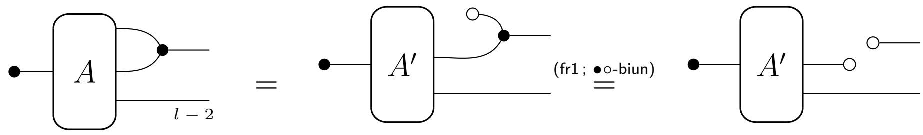
(3.97)

Since $r < n$ we need to apply the rewriting procedure of Case 2 below.

- Case 2: $m < n$. This case is simply the previous case in disguise. We can apply the rewriting steps of Case 1, mirrored: first expanding , then using Euclidean division to obtain $k, r \in \mathbb{N}$ such that $n = mk + r$ with $r < m$, and apply the axiom (n-inv) as before.

This game of ping-pong necessarily terminates because, by using Euclidean division each time, we decrease the value of the scalars to which we have to apply (n-inv) at each step.

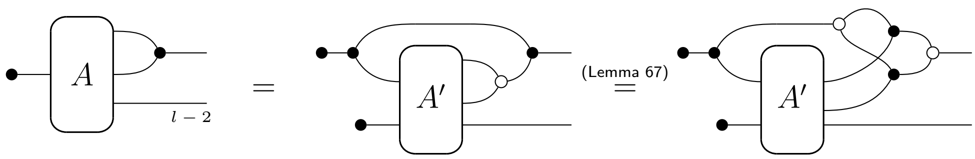
Base cases. If has A-connectedness $(0, m)$, we have



(3.98)

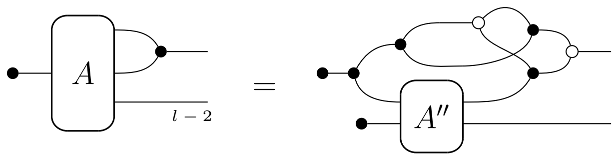
for some matrix A'. From here, we can absorb the with the result of Section 3.6.1.2.

If has A-connectedness $(1, m)$, we will reason by induction again. By hypothesis, there exists a matrix A' such that

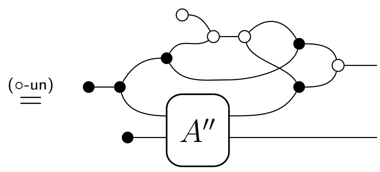


(3.99)

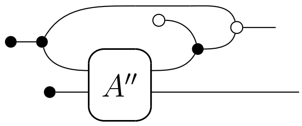
As before, we want to make sure that there are no loops. A loop implies the existence of a matrix A'' such that

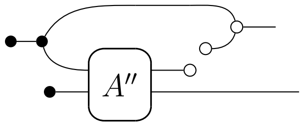


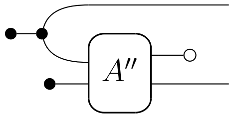
(3.100)

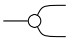


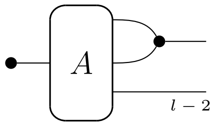
(3.101)

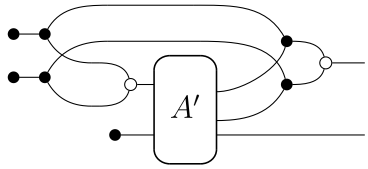
$\overset{\text{(can)}}{=}$  (3.102)

$\overset{\text{(Lemma 67)}}{=}$  (3.103)

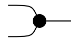
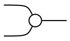
$=$  (3.104)

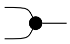
The proof can be finished using the result of Section 3.6.1.2. If there are no loops we can slide  as follows:

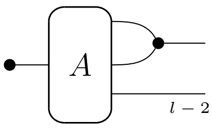


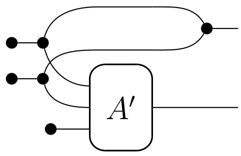
$=$ 

(3.105)

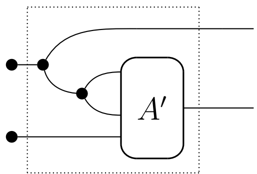
We can repeat this process until we obtain m  with connectedness $(1, 1)$. Then we can push the m  we have accumulated on the left, into the matrix A'.

In the final case, if  has $(1, 1)$ connectedness, there exists a matrix A' such that



$=$ 

(3.106)

$\overset{\text{(Spider)}}{=}$ 

(3.107)

where the last diagram is in prenormal form.

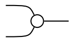
3.6.2 Normal form

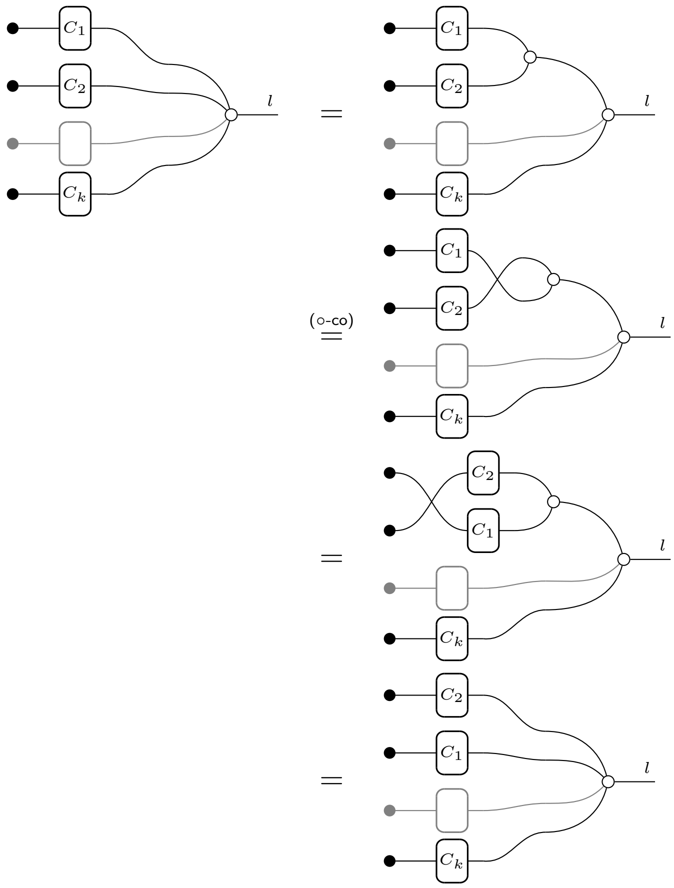
The prenormal form does not uniquely characterise the additive relation to which it is mapped. Two different matrices may generate the same additive relation as some of their columns may be dependent and therefore, redundant.

Definition 73. A diagram of Rc is in *normal form* when the columns of the matrix of its prenormal form are independent.

The following proposition proves the faithfulness of $[\![-]\!]$, provided we can show that every diagram of Rc is equal to one in normal form.

Proposition 74. *If c and d are two diagrams in normal form, $[\![c]\!] = [\![d]\!]$ implies that the representing matrices of c and d have the same set of columns.*

Proof. By Proposition 46, $[\![c]\!] = [\![d]\!]$ if and only if $[\![c]\!]$ and $[\![d]\!]$ have the same Hilbert basis. Appealing to the completeness result for matrices (Theorem 63), this is true when the representing matrices of c and d have the same *set* of independent columns. But these two matrices may still differ by a permutation of their columns. To conclude the proof, we need to show that two diagrams in normal form that differ only by a permutation of the columns of their representing matrix, are equal. This is a consequence of the commutativity of  :



(3.108)

(3.109)

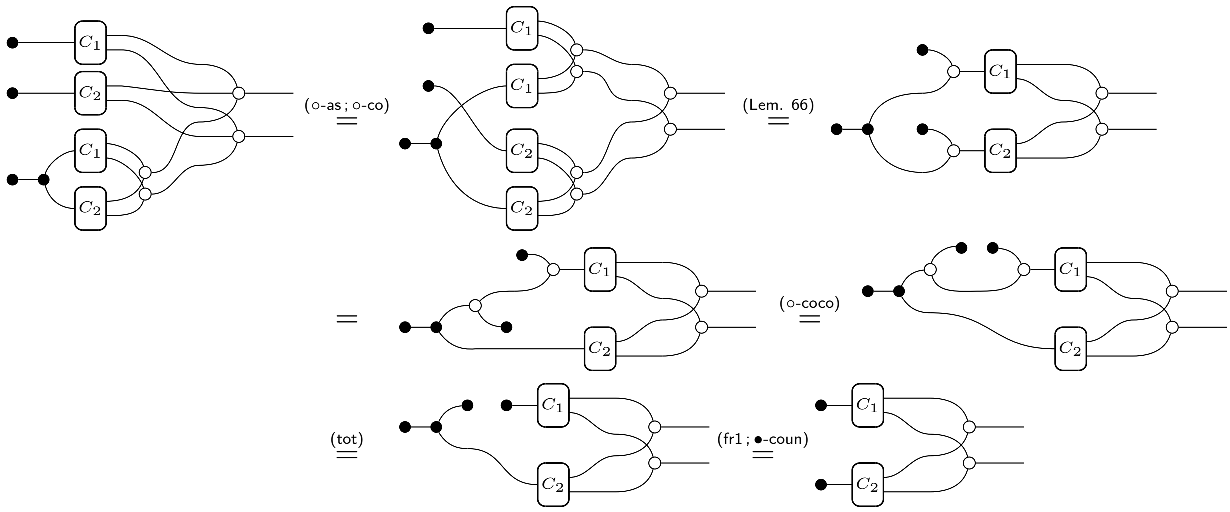
(3.110)

(3.111)

The general result follows from this by induction. □

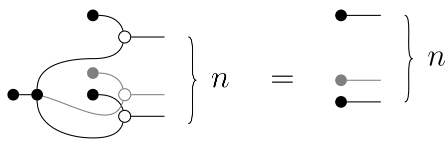
To be able to prove that diagrams are equal to their normal form, we need to make sure that redundant generators can be eliminated diagrammatically from the axioms of Rc. For linear relations over field, this step corresponds to Gaussian elimination to obtain a row-reduced form of the representing matrix. In our case, we need to check that columns that are sums of other columns of the same matrix can be eliminated, using only the equations of Rc. It is perhaps more enlightening to introduce this procedure with an example—all the other cases being a straightforward generalisation of the following simple example.

Example 75. We start with a three column matrix in which the third is the sum of the first two columns. The derivation concludes with the third column eliminated.

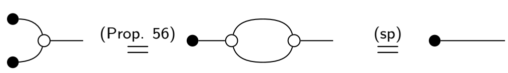


The key step is the use of the (tot) equation which we had not used previously. The general case proceeds exactly as in Example 75—we will just need two lemmas to eliminate redundant sums of columns of arbitrary size. The first one is a generalised form of (tot).

Lemma 76. *For* $n \geq 1$,

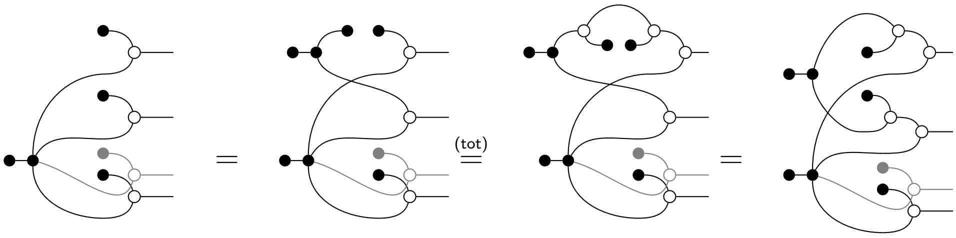


Proof. By induction on n. The base case is

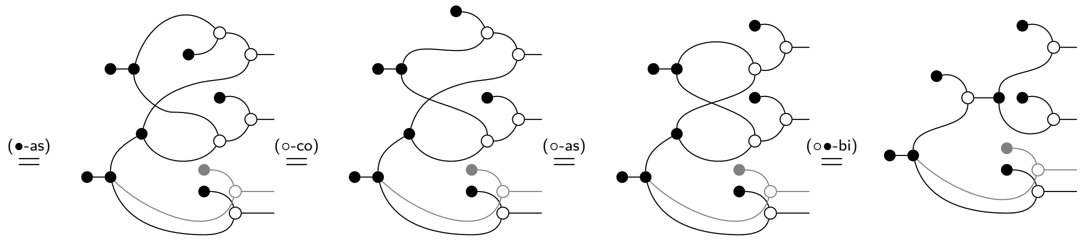


(3.112)

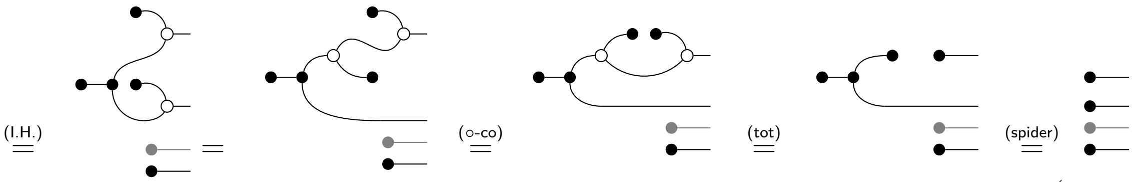
Assume that it holds for some positive integer n.



(3.113)



(3.114)

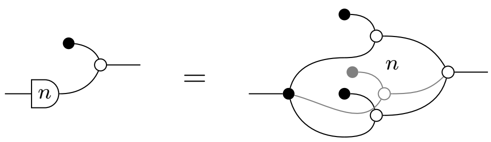


(3.115)

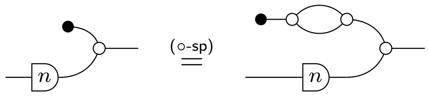
□

The next lemma allows us to handle weighted sums by reducing them to the case of lemma 76 (just like $n\mathbf{x} = \mathbf{x} + \cdots + \mathbf{x}$, n times).

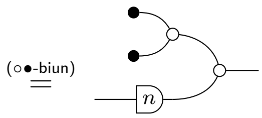
Lemma 77. *For $n \geq 1$,*



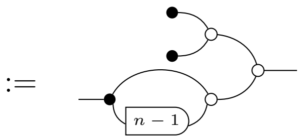
Proof. By induction on n. The base case is immediate so assume that the lemma holds for some positive integer $n - 1$. Then,



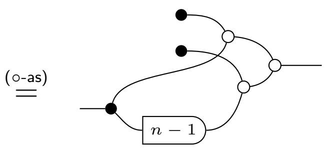
(3.116)



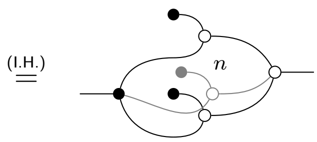
(3.117)



(3.118)



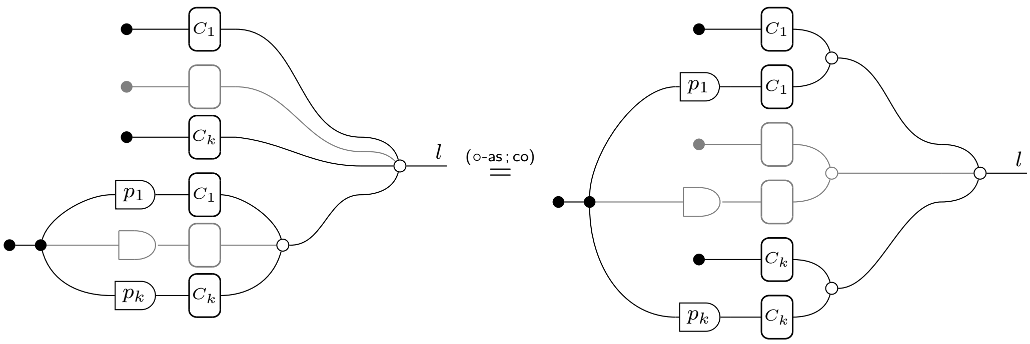
(3.119)



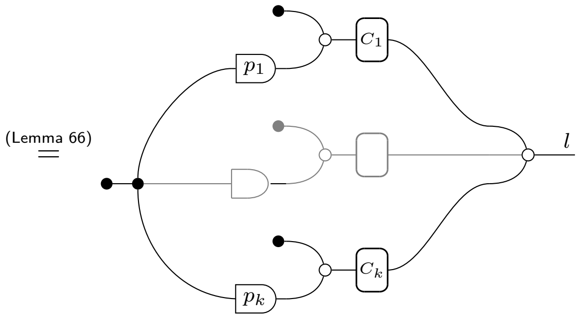
(3.120)

□

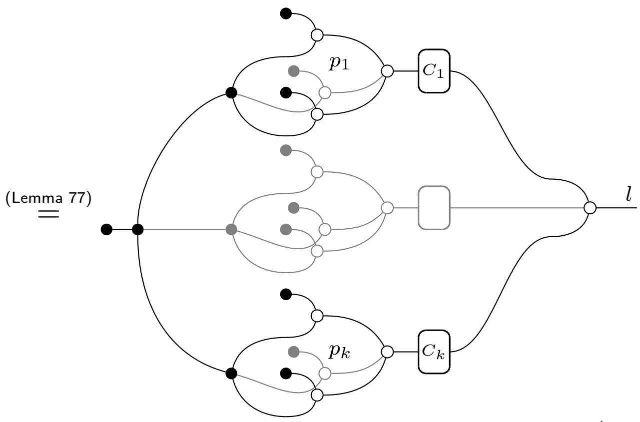
We can now proceed exactly as in Example 75 to delete redundant generating columns in the general case:



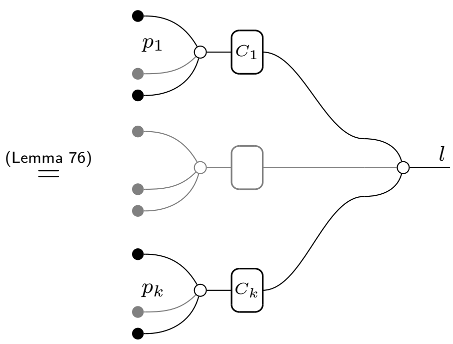
(3.121)



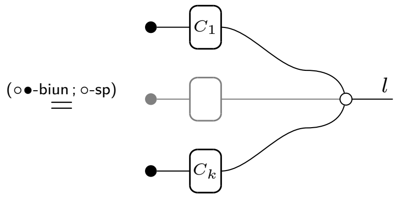
(3.122)



(3.123)



(3.124)



(3.125)

Remark 78. Note that we did not give an effective procedure to rewrite a diagram in prenormal form into one in normal form. We have only shown that, if there are redundancies, we can eliminate them with the axioms of Rc but did not show how to identify them. This is sufficient for our purposes.

3.7 The subprop of multirelations

We now present the prop of *multirelations*. It is a subprop of AddRel whose diagrams admit a consistent directionality. Left ports can be seen as inputs, right ports as outputs and the resources as flowing from inputs to outputs. This interpretation makes sense because we only have access to —●⊂ and —● (and both ○-monoid and comonoid) and therefore the resulting prop is not compact closed. We also give a presentation for it with a lot of the same algebraic structure as Rc. Interestingly, it is also the Kleisli category of PM, the composite of the finite powerset monad with the multiset monad.

3.7.1 Powerset of multisets

The functor $M : \mathsf{Set} \to \mathsf{Set}$ sends a set X to its set of multisets (or bags),

$$MX = \{\mathbf{a} \in \mathbb{N}^X \text{ with finite support}\} \tag{3.126}$$

and a map $f : X \to Y$ to $Mf : MX \to MY$ given by

$$Mf\left([x_1, \ldots, x_n]\right) = [f(x_1), \ldots, f(x_n)] \tag{3.127}$$

We write multisets using brackets, e.g., $[x, y, y, z, x]$ and denote multisets using boldface as we did for vectors with natural number coefficients. The multiset functor can be equipped with the structure of a monad with

- multiplication $\mu^M_X : MMX \to MX$ given by

$$\mu^M_X\left(\Big[[x_{1,1}, \ldots, x_{1,n}], \ldots, [x_{n,1}, \ldots, x_{n,n}]\Big]\right) = [x_{1,1}, \ldots, x_{1,n}, \ldots, x_{n,1}, \ldots, x_{n,n}] \tag{3.128}$$

- unit $\eta^M_X : X \to MX$ given by $\eta^M_X(x) = [x]$.

It is well known that the full-subcategory of the Kleisli category of M spanned by finite sets is equivalent to the prop $\mathsf{Mat}_{\mathbb{N}}$. This is a simple consequence of the product-exponential adjunction, giving a bijection between maps $X \to \mathbb{N}^Y$ and maps $X \times Y \to \mathbb{N}$. Then, fixing a total order on X and Y, we get $|Y| \times |X|$ matrices with coefficients in $\mathbb{N}$.

Proposition 79. *There is a distributive law $\lambda : MP \to PM$ of the multiset monad over the finite powerset monad, P, given by*

$$\lambda_X\left([A_1, \ldots, A_n]\right) = \Big\{[x_1, \ldots, x_n] \mid x_i \in A_i, 1 \leq i \leq n\Big\} \tag{3.129}$$

where the A_i are subsets of X.

Proof. This distributive law is mentioned frequently in the literature but we could not track down a first-principles proof that it is indeed a distributive law, so we give one below for completeness. Let X be a set. Checking that λ is a natural transformation is straightforward. We verify the commutativity of the four distributive law diagrams.

- For $[x_1, \ldots, x_n] \in MX$,

$$\begin{aligned}
\lambda_X(M\eta^P_X)([x_1, \ldots, x_n]) &= \lambda_X\left(\Big[\{x_1\}, \ldots, \{x_n\}\Big]\right) && (3.130)\\
&= \Big\{[x_1, \ldots, x_n]\Big\} && (3.131)\\
&= \eta^P_{MX}\Big([x_1, \ldots, x_n]\Big) && (3.132)
\end{aligned}$$

- For $\{x_1, \ldots, x_n\} \subseteq X$,

$$\begin{aligned}
\lambda_X\eta^M_{PX}(\{x_1, \ldots, x_n\}) &= \lambda_X\left(\Big[\{x_1, \ldots, x_n\}\Big]\right) && (3.133)\\
&= \Big\{[x_1], \ldots, [x_n]\Big\} && (3.134)\\
&= P\eta^M_X\Big(\{x_1, \ldots, x_n\}\Big) && (3.135)
\end{aligned}$$

- For $\mathcal{A}_i \subseteq PX$, with $1 \leq i \leq n$,

$$\begin{aligned}
&\mu^P_{MX}(P\lambda_X)\lambda_{PX}\Big([\mathcal{A}_1, \ldots, \mathcal{A}_n]\Big) && (3.136)\\
&= \mu^P_{MX}(P\lambda_X)\left(\Big\{[A_1, \ldots, A_n] \mid A_i \in \mathcal{A}_i, 1 \leq i \leq n\Big\}\right) && (3.137)\\
&= \mu^P_{MX}\left(\Big\{\{[x_1, \ldots, x_n] \mid x_i \in A_i\} \mid A_i \in \mathcal{A}_i, 1 \leq i \leq n\Big\}\right) && (3.138)\\
&= \Big\{[x_1, \ldots, x_n] \mid x_i \in A_i, A_i \in \mathcal{A}_i, 1 \leq i \leq n\Big\} && (3.139)\\
&= \lambda_X\left(\Big[\bigcup \mathcal{A}_1, \ldots, \bigcup \mathcal{A}_n\Big]\right) && (3.140)\\
&= \lambda_X(M\mu^P_X)\Big([\mathcal{A}_1, \ldots, \mathcal{A}_n]\Big) && (3.141)
\end{aligned}$$

- For $A_{i,j} \subseteq X$, with $1 \leq j \leq k_i$ and $1 \leq i \leq n$,

$$
\begin{aligned}
&(P\mu^M_X)\lambda_{MX}(M\lambda_X)\left(\left[[A_{1,1},\ldots,A_{1,k_1}],\ldots,[A_{n,1},\ldots,A_{n,k_n}]\right]\right) && (3.142)\\
&= (P\mu^M_X)\lambda_{MX}\left(\left[\{[x_{i,1},\ldots,x_{i,k_i}] \mid x_{i,j} \in A_{i,j}\} \mid 1 \leq i \leq n\right]\right) && (3.143)\\
&= P\mu^M_X\left(\left\{\left[[x_{i,1},\ldots,x_{i,k_i}]\right] \mid x_{i,j} \in A_{i,j}, 1 \leq i \leq n\right\}\right) && (3.144)\\
&= \{[x_{i,1},\ldots,x_{i,k_i}] \mid x_{i,j} \in A_{i,j}, 1 \leq i \leq n\} && (3.145)\\
&= \lambda_X\left(\left[\{x_{1,j} \mid 1 \leq j \leq k_1\},\ldots\{x_{n,j} \mid 1 \leq j \leq k_n\}\right]\right) && (3.146)\\
&= \mu^M_{PX}\lambda_X\left(\left[[A_{1,1},\ldots,A_{1,k_1}],\ldots,[A_{n,1},\ldots,A_{n,k_n}]\right]\right) && (3.147)
\end{aligned}
$$

□

As a result, PM is also a monad. Another way to phrase this is that the monad M lifts to the Kleisli category of the finite powerset monad, namely the category of sets and *finitely-branching* relations: we call a relation $r \subseteq X \times Y$ finitely-branching when, for every $x \in X$, there is at most a finite number of $y \in Y$ such that $(x, y) \in r$. The monad multiplication is given by $PMPM \xrightarrow{P\lambda M} PPMM \xrightarrow{\mu^P\mu^M} PM$ and the unit by $\mathsf{id} \xrightarrow{\eta^P\eta^M} PM$.

We consider the full subcategory of the Kleisli category of PM spanned by finite ordinals. Call its morphisms *multirelations*. It defines a prop which we describe explicitly below by unrolling the definition of the Kleisli composition for PM. As multirelations are more conveniently expressed in relational form, that is, as subsets of $\underline{k} \times M\underline{l}$ for multirelations of type $\underline{k} \to \underline{l}$; we adopt this convention. We abuse notation slightly by identifying multisets of elements of $\underline{k}$ (i.e., $M\underline{k} = \mathbb{N}^{\underline{k}}$) and vectors of $\mathbb{N}^k$. Then μ is simply the sum of vectors.

Definition 80. Let MRel be the prop in which

- morphisms $k \to l$ are finitely-branching relations $\underline{k} \times M\underline{l}$;
- with composition defined by $(i, \mathbf{c}) \in f \,;\, g$ iff there exists $(p_1, \mathbf{c}_1), \ldots, (p_n, \mathbf{c}_n) \in g$ such that

$$\sum_{i=1}^{n} \mathbf{c}_i = \mu^M_{\underline{m}}([\mathbf{c}_1, \ldots, \mathbf{c}_n]) = \mathbf{c} \text{ and } (i, [p_1, \ldots, p_n]) \in f \quad (3.148)$$

  for $f \colon k \to l$ and $g \colon l \to m$;
- identity $\mathsf{id}_k = \{(i, [i]) \mid i \in \underline{k}\}$;
- monoidal product $f_1 \oplus f_2 \colon k_1 + k_2 \to l_1 + l_2$ of $f_1 \colon k_1 \to l_1$ and $f_2 \colon k_2 \to l_2$ given by $(i, \mathbf{b}) \in f_1 \oplus f_2$ iff $i \leq k_1$ and $(i, \mathbf{b}) \in f_1$, or $i > k_1$ and $(i, \mathbf{b}) \in f_2$;

- braiding $\asymp_k^l := \{((i,j),([i],[j])) \mid i \in \underline{k}, j \in \underline{l}\}$.

Multirelations can be interpreted as special cases of additive relations. Morally, the functor given on morphisms by $(f: \underline{k} \to PM\underline{l}) \mapsto Mf \,;\, \lambda_{M\underline{l}} \,;\, P\mu_{\underline{l}}^M$ is the embedding we are looking for. However, this would be an abuse of notation as this functor does not have the right type.

Unrolling the abstract definition, we obtain a prop morphism $E_M : \mathsf{MRel} \to \mathsf{AddRel}$ given, in relational form, by

$$E_M f := \left\{ \left(\sum_{i=1}^n \mathbf{e}_{p_i}, \sum_{i=1}^n \mathbf{b}_i \right) \mid (p_i, \mathbf{b}_i) \in f \right\} \tag{3.149}$$

for $f: k \to l$ and where $\{\mathbf{e}_j\}_{1 \le j \le k}$, are the basis vectors of $\mathbb{N}^k$.

Proposition 81. $E_M : \mathsf{MRel} \hookrightarrow \mathsf{AddRel}$ *is a faithful prop morphism.*

Proof. For unitality, $E_M f(\mathsf{id}_k) = \left\{ \left(\sum_{i=1}^n \mathbf{e}_{p_i}, \sum_{i=1}^n \mathbf{e}_{p_i} \right) \mid n \in \mathbb{N} \right\}$ which is the identity additive relation on k.

For functoriality, let $f: k \to l$ and $g: l \to p$ be two multirelations.

- If $\left(\sum_{i=1}^n \mathbf{e}_{p_i}, \sum_{i=1}^n \mathbf{c}_i \right) \in E_M(f \,;\, g)$ for some $n \in \mathbb{N}$, then $(p_i, \mathbf{c}_i) \in f \,;\, g$ for $1 \le i \le n$ and there exists $(p_{ij}, \mathbf{c}_{ij}) \in g$, for $1 \le j \le m_i$ such that $\sum_{j=1}^{m_i} \mathbf{c}_{ij} = \mathbf{c}_i$ and $\left(p_i, \sum_{j=1}^{m_i} \mathbf{e}_{p_{ij}} \right) \in f$. Then, by additivity,

$$\left(\sum_{i=1}^n \mathbf{e}_{p_i}, \sum_{i=1}^n \sum_{j=1}^{m_i} \mathbf{e}_{p_{ij}} \right) \in E_M f \,;\, E_M g \tag{3.150}$$

- Conversely if $(\mathbf{a}, \mathbf{b}) \in E_M f \,;\, E_M g$, there exists $\mathbf{b} \in \mathbb{N}^l$ such that $(\mathbf{a}, \mathbf{b}) \in E_M f$ and $(\mathbf{b}, \mathbf{c}) \in E_M g$. By definition, we can decompose them as follows:

$$(\mathbf{a}, \mathbf{b}) = \left(\sum_{i=1}^n \mathbf{e}_{p_i}, \sum_{i=1}^n \mathbf{b}_i \right) \in E_M f \text{ and } (\mathbf{b}, \mathbf{c}) = \left(\sum_{i=1}^m \mathbf{e}_{q_i}, \sum_{i=1}^m \mathbf{c}_i \right) \in E_M g \tag{3.151}$$

  for some $n, m, p_i, q_i \in \mathbb{N}$ and $\mathbf{b}_i \in \mathbb{N}^l, \mathbf{c}_i \in \mathbb{N}^p$. Note that we necessarily have $n \le n'$ because $\sum_{i=1}^n \mathbf{b}_i = \mathbf{b} = \sum_{i=1}^m \mathbf{e}_{p_i}$ and $\mathbf{b}_i \not\le \mathbf{e}_{p_j}$. Thus, we can partition the q_i, $1 \le i \le m$ into n classes, through a surjective map $\varphi : \underline{m} \to \underline{n}$, such that $\mathbf{b}_i = \sum_{\varphi(j)=i} \mathbf{e}_{q_j}$. Hence, we have $\left(p_i, \sum_{\varphi(j)=i} \mathbf{c}_i \right) \in f \,;\, g$ and consequently,

$$(\mathbf{a}, \mathbf{c}) = \left(\sum_{i=1}^n \mathbf{e}_{p_i}, \sum_{i=1}^n \sum_{\varphi(j)=i} \mathbf{c}_i \right) \in E_M(f \,;\, g) \tag{3.152}$$

Finally, E_M is faithful. Assume that $E_M f = E_M g$ for multirelations $f, g: k \to l$. If $(i, \mathbf{b}) \in f$, then $(\mathbf{e}_i, \mathbf{b}) \in E_M f = E_M g$ and thus $(i, \mathbf{b}) \in g$ as well.

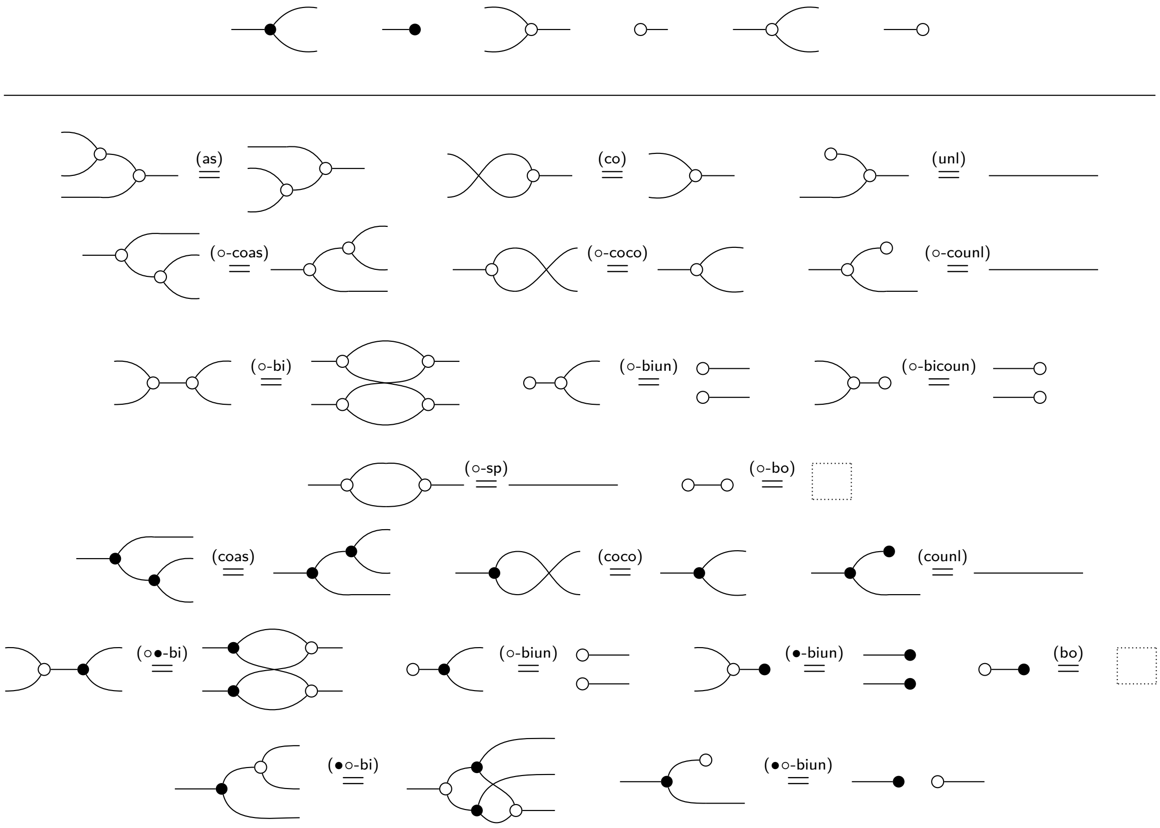


Figure 3.3: Presentation of Pc.

□

Equipped with this embedding, we can transfer the concept of Hilbert basis to multirelations in order to characterise them uniquely. The following proposition is immediate from the explicit form of E_M.

Proposition 82. *Multirelations $k \to l$ are in one-to-one correspondence with additive relations $k \to l$ whose Hilbert basis elements are all of the form $(\mathbf{e}_i, \mathbf{a})$ for some basis vector $\mathbf{e}_i \in \mathbb{N}^k$ and some arbitrary vector $\mathbf{a} \in \mathbb{N}^l$.*

3.7.2 Presenting multirelations

We have now done enough work to derive a presentation of MRel.

Definition 83. Let Pc be the prop with presentation given in Fig. 3.3

Proposition 84. *There is an embedding $E_{\mathsf{Pc}} \colon \mathsf{Pc} \hookrightarrow \mathsf{Rc}$.*

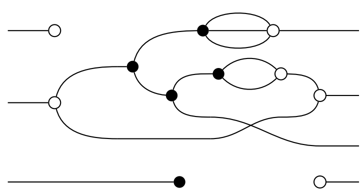
Proof. This is the unique prop morphism that sends , , and to themselves and , to their definition in Rc (see Remark 55). It is functorial because the axioms for , , and are also axioms of Rc; , , , also form a special commutative bimonoid in Rc, and the last pair of axioms are derived equations of Rc by Lemma 67. □

From the previous discussion, there are at least two different perspectives on multirelations: one can view them as additive relations without the ability to merge and generate resources or as relations with the additional ability to duplicate and delete resources. This is reflected in the diagrammatic calculus. Indeed, the lack of and prevents us from bending wires at will; MRel is not compact closed. This allows us to transfer causal intuitions to diagrams, whose left ports are now genuine inputs and right ports are genuine outputs.

It is instructive to contrast this behaviour with the case of linear relations. Remember that, in $\mathsf{IH}_{\mathbb{K}}$, both ∘ and •-structures are Frobenius monoids. Thus, any subprop of $\mathsf{IH}_{\mathbb{K}}$ that contains and (and their respective units) inherits cups and caps. This is exploited in [Zan15, Theorem 4.48], where the author shows that diagrams in $\mathsf{IH}_{\mathbb{K}}$ over the field of fractions of $\mathbb{K}[x]$ can all be directed and interpreted as rewirings (in a precise sense) of matrices over the ring of rationals (polynomial fractions whose denominator has non-zero leading coefficient). These, in turn, correspond to rational behaviours, for which the operational semantics of the corresponding diagram is realised by a weighted finite-state machine.

We chose the name Pc or *producer calculus* because it contains the part of the resource calculus from which one can produce more resources than one started with. The following example illustrates how to go between multirelations and diagrams of Pc.

Example 85. The multirelation $f\colon 3 \to 4$ given by $\{(1,[1,1,1,2,2,3]),(1,[2]),(2,[])\}$ or, in vectorial form, $\left\{\left(1,\begin{pmatrix}3\\2\\1\\0\end{pmatrix}\right),\left(1,\begin{pmatrix}0\\1\\0\\0\end{pmatrix}\right),\left(2,\begin{pmatrix}0\\0\\0\\0\end{pmatrix}\right)\right\}$, is depicted as

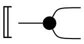
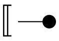


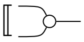
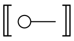
(3.153)

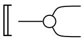
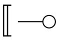
The main theorem of this section is the following.

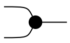
Theorem 86. *The categories* MRel *and* Pc *are isomorphic.*

Proof. We claim that the isomorphism $[\![-]\!]_M : \mathsf{Pc} \to \mathsf{MRel}$ is the unique prop morphism defined on the generating morphisms of Pc by:

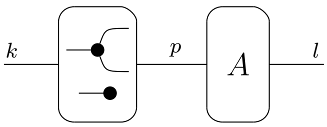
$[\![$  $]\!]_M := \left\{\left(0, \begin{pmatrix}1\\1\end{pmatrix}\right)\right\}$ $\qquad$ $[\![$  $]\!]_M := \{(0, \bullet)\}$ (3.154)

$[\![$  $]\!]_M := \{(0,1),(1,1)\}$ $\qquad$ $[\![$  $]\!]_M := \{1\}$ (3.155)

$[\![$  $]\!]_M := \left\{\left(0, \begin{pmatrix}1\\0\end{pmatrix}\right), \left(0, \begin{pmatrix}0\\1\end{pmatrix}\right)\right\}$ $\qquad$ $[\![$  $]\!]_M := \varnothing$ (3.156)

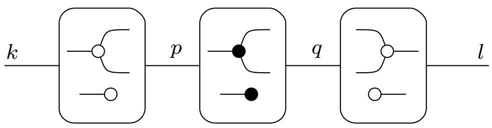
That $[\![-]\!]_M$ is functorial is another mechanical verification of the soundness of the equations of Pc for MRel and it is symmetric monoidal by construction. The interesting part of the proof is to show that it is an isomorphism. For fullness, it is straightforward to generalise the scheme described in Example 85 to construct a diagram encoding a given multirelation. Alternatively, we can reuse the method given to translate additive relations into Rc diagrams (cf. Section 3.6): first compute the unique additive relations $E_M f$ corresponding to a given multirelation f and then draw the diagram for $E_M f$. As f is a multirelation, the resulting diagram does not contain any  or , so is in the image of $E_{\mathsf{Pc}} \colon \mathsf{Pc} \hookrightarrow \mathsf{Rc}$.

For faithfulness, we can procede by a normal form argument as we have done previously. For this, we make use of Proposition 82 to show that any diagram c of Pc is equal to one from which the Hilbert basis of the associated additive relation can be read unambiguously. We say that a diagram of Pc is in normal form when it is written as

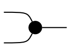
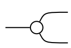
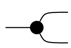
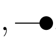


(3.157)

where A is a matrix diagram, for some $p \in \mathbb{N}$ or, equivalently,



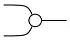
(3.158)

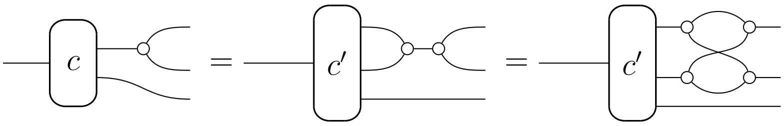
for some $p, q \in \mathbb{N}$. In these diagrams, the boxes annotated with generators represent a layer of arbitrary compositions and monoidal products of the corresponding generators (including potential permutations). So the normal form corresponds to a factorisation of diagrams in Pc into three successive layers of diagrams with restricted sets of generators. Note that this matches the normal form of diagrams in the resource calculus in its span version (see Remark 92 below). Diagrams of Rc are more general and can have a layer of ,  in between those of ,  and , .

From the normal form, we can immediately read the Hilbert basis of the associated multirelation, as we did for Rc diagrams. Here it is easier because the prenormal and normal form are one and the same. Indeed, as generating vectors are always of the form $(\mathbf{e}_i, \mathbf{b})$ for some basis vector $\mathbf{e}_i$, one of them cannot be a sum of the others and there cannot be any redundancy in the normal form.

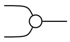
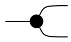
To prove that this factorisation exists for every diagram, we reason by structural induction. The base cases are immediate. Below, we give an effective procedure to rewrite c composed with every generator into normal form.

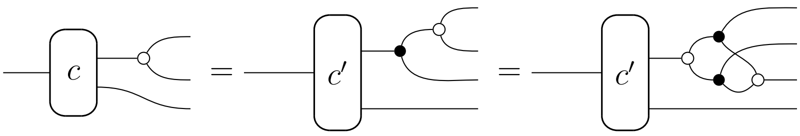
1. *Co-addition on the right.* We can either:

   (a) encounter a , in which case

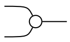
    (3.159)

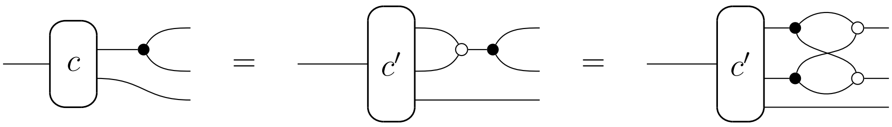
   for some diagram c' and we can apply the induction hypothesis;

   (b) or we are past the  layer and ecounter a . Then there exists c' such that

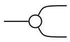
    (3.160)

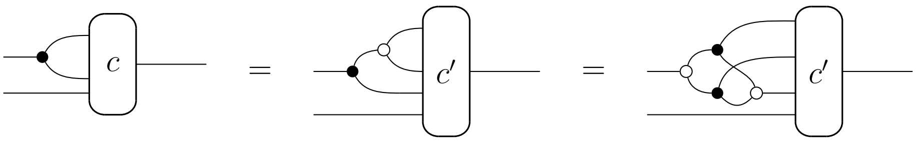
   and we can now apply the induction hypothesis.

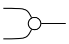
2. *Copy on the right.* If we encounter a , there exists a diagram c' such that

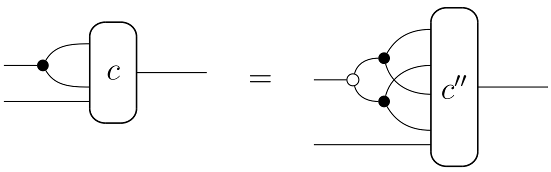
    (3.161)

   and we can apply the induction hypothesis.

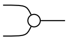
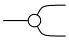
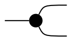
3. *Copy on the left.* If we encounter a , there exists a diagram c' such that

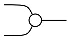
    (3.162)

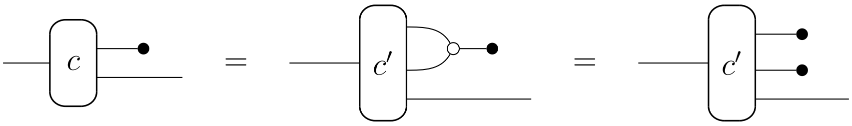
   To deal with the  that we have introduced in the last equation, we go to step 4. We obtain a diagram c'' such that

    (3.163)

at which point we can apply the induction hypothesis.

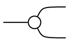
4. *Addition on the left.* We can simply apply the same steps as for 1. and 2. to push  successively through the layer of  and then  .

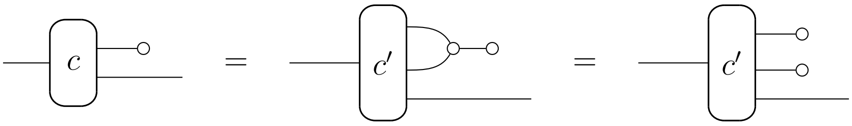
5. *Delete on the right.* If we encounter  , there exists a diagram c' such that



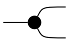
(3.164)

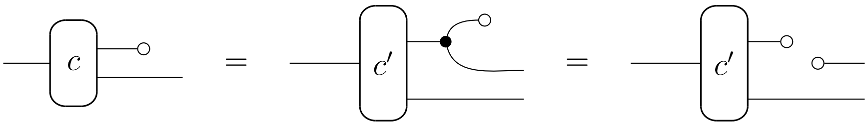
We can now apply the induction hypothesis.

6. *Co-zero on the right.* If we encounter  , there exists a diagram c' such that



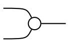
(3.165)

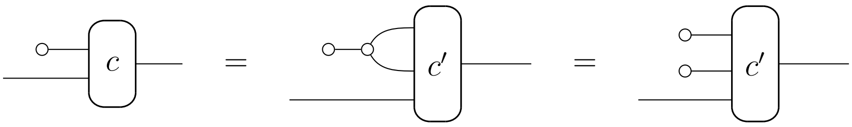
Otherwise, if we encounter  , there exists a diagram c' such that



(3.166)

and we can apply the induction hypothesis.

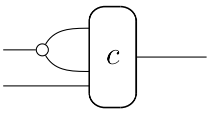
7. *Zero on the left.* If we encounter  , there exists a diagram c' such that

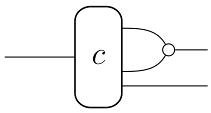


(3.167)

We can now apply the induction hypothesis.

8. *Co-addition on the left or addition on the right.* The diagrams



and


(3.168)

are already in normal form.

9. All the other cases involve taking the monoidal product of c with a generator and we can therefore apply the induction hypothesis without any rewriting.

□

3.7.3 Multirelations and linear logic

MRel is intimately related to a well-known model of linear logic. We do not wish to introduce the topic of linear logic here but simply to point out that Pc provides a diagrammatic calculus for a simple model of its multiplicative exponential fragment.

We pointed out earlier that, because of the existence of the distributive law $\lambda\colon MP \to PM$, M lifts to the Kleisli category of P, namely the category of sets and finite-branching relations. Let us call $?\colon \mathsf{Rel} \to \mathsf{Rel}$ this lifting. Remarkably, the same functor over Rel also admits the structure of a *comonad* obtained by simply transposing the two natural transformations $\mu^?$ and $\eta^?$. To distinguish these two different structures, we will write ! for the comonad, as is standard in the linear logic literature.

The relational model is one of the simplest categorical models of classical linear logic [Sch04]. It is given by interpreting the logic into $\mathsf{Rel}^!$, the coKleisli category of the multiset comonad which we describe more explicitly below: it has

- finite sets $X, Y, \ldots$ as objects;
- relations $!X \nrightarrow Y$ as morphisms $X \to Y$;
- the relation $\{([x], x) \mid x \in X\}$ as identity 1_X;
- composition defined by $(\mathbf{a}, z) \in r\,;\,s$ iff

  $$\exists(\mathbf{a}_1, y_1), \ldots, (\mathbf{a}_n, y_n) \in r \text{ such that } \mathbf{a} = \sum_{i=1}^{n} \mathbf{a}_i \text{ and } ([y_1, \ldots, y_n], z) \in s \quad (3.169)$$

  for $r : X \to Y$ and $s : Y \to Z$.

The category $\mathsf{Rel}^!$ is Cartesian closed with the disjoint sum as categorical product and $X \Rightarrow Y := !X \times Y$ as exponential. It turns out that this category is equivalent to MRel^{op}.

Proposition 87. MRel^{op} *is monoidally equivalent to* $\mathsf{Rel}^!$

Proof. There is a sequence of monoidal natural isomorphisms:

$$\mathsf{MRel}^{op}(k, l) = \mathsf{MRel}(l, k) \cong \mathsf{Rel}(\underline{l}, ?\underline{k}) \cong \mathsf{Rel}(!\underline{k}, \underline{l}) \cong \mathsf{Rel}^!(\underline{k}, \underline{l}) \quad (3.170)$$

where the third one is a consequence of the fact that Rel is self-dual, by transposition. □

3.8 Instructive failures: towards a modular account of additive relations

In [BSZ17, Zan15] the authors construct the prop of linear relations in a *modular* way: using simpler props with known presentations as building blocks and distributive laws to combine them, they obtain a presentation for $\mathsf{LinRel}_{\mathbb{K}}$. Unfortunately, none of their methods apply directly to the case of additive relations. Nonetheless, we believe that it is useful to identify some of the obstructions more precisely, so that we may one day develop new methods for composing props and eventually obtain a modular account of AddRel and its presentation. This section is a high-level discussion of where the usual approach fails. Someone reading the present work might even find enough inspiration for a solution to this problem.

Props can be seen as monads in a 2-category of bimodules over spans of monoids, as first observed in [Lac04]. Because they are monads, they can be composed via distributive laws. The distributive laws in [Zan15] come from the interaction of the prop $\mathsf{Mat}_{\mathsf{R}}$ of matrices over a PID R, with its dual, $\mathsf{Mat}_{\mathsf{R}}^{op}$. As we do not need them, we will not introduce distributive laws of props formally in this thesis. Instead, we refer the reader to [Zan15, Section 2.4] or directly to Lack's original account in [Lac04]. Intuitively, distributive laws of props specify how to slide the morphisms of two props past each other. This allows them to be combined into a new prop. They naturally induce a functorial factorisation of morphisms and are closely related to factorisation systems. They are particularly useful to derive complete sets of equations for the resulting prop.

First, notice that $\mathsf{Mat}_{\mathsf{R}}$ is finitely complete so it has pullbacks. Therefore, we can form its category of *spans*, whose objects are the same and whose morphisms are pairs of matrices $k \to p \leftarrow l$[3].

Two spans with a common foot can be composed by pullback:

$$\begin{array}{ccccc} & & p +_l q & & \\ & \overset{U}{\swarrow} & & \overset{V}{\searrow} & \\ & p & & q & \\ \overset{A}{\swarrow} & & \overset{B}{\searrow} \qquad \overset{C}{\swarrow} & & \overset{D}{\searrow} \\ k & & l & & m \end{array} \tag{3.171}$$

[3]Spans more naturally form a bicategory and the 1-category we describe is its truncation. Therefore, it would be more correct to say that its morphisms are isomorphism classes of spans. Since we do not use spans in any fundamental way, we stay slightly imprecise.

A span of matrices is the same thing as a pair from $\mathsf{Mat_R} \times \mathsf{Mat_R}^{op}$. And, when composing two spans via pullback, the two morphisms pointing the wrong way (forming a cospan) in the middle are turned into a span. This operation is functorial and defines a distributive law.

The next key idea is that, given a presentation for $\mathsf{Mat_R}$ (and therefore, of its dual as well) we only need to find a presentation of the additional equations coming from the distributive law given by pullbacks. These equations describe the different ways of sliding matrices and their opposites past each other. This produces a *local* view of the prop of spans in which the equational theory can be reduced to a complete set of basic interactions between the generators of $\mathsf{Mat_R}$ and $\mathsf{Mat_R}^{op}$.

A completely analogous procedure yields a presentation of the prop of cospans of matrices, whose morphisms are now pairs of matrices of type $k \leftarrow p \rightarrow l$, composed via pushout.

Finally, it turns out that linear relations can be constructed as a colimit of the categories of spans and cospans of matrices, gluing them in the appropriate way. The same operation at the level of the presentations provides a presentation of $\mathsf{LinRel}_{\mathbb{K}}$. The colimit can be understood as the result of choosing a well-behaved notion of *image* in the original category of matrices and identifying all spans with the same image. For a regular category, there is a canonical way to define a suitably functorial notion of image: the regular epimorphisms with the monomorphisms form the left and right class of a factorisation system so that every morphism can be factored uniquely—up to isomorphism—through its image.

There are two fundamental obstacles to applying the method above to derive a presentation for AddRel: *i)* $\mathsf{Mat}_{\mathbb{N}}$ does not have pullbacks nor pushouts and *ii)* the notion of image associated with an additive relation does not give rise to a factorisation system in $\mathsf{Mat}_{\mathbb{N}}$. We explain both points below.

Remark 88. In general, reasoning equationally about additive relations is harder than about linear relations. The graphical calculus for linear relations is fundamentally undirected as both $\bullet$ and $\circ$-structures are Frobenius monoids for which the spider theorem (Theorem 18) holds. This theorem allows us to forget a lot of the structure of monochromatic diagrams to focus on the relation between $\bullet$ and $\circ$ using the bimonoid laws. For additive relations, the presence of a second bimonoid requires some additional care. In particular it requires more effort to keep track of the causal flow through $\circ$-nodes. In addition, the cancellativity axiom of Rc is of a less local flavour than any of the axioms of $\mathsf{IH}_{\mathbb{K}}$, creating new difficulties, as we saw in the rewriting procedure of Section 3.6.1. However, it is not clear how the complexity of diagram rewriting and the construction of a prop using distributive laws are related. This is an active area of research.

First, recall that $\mathsf{Mat}_{\mathbb{N}}$ is equivalent to the Kleisli category of the multiset monad M. On the other hand, the category of algebras of M is equivalent to the category of commutative monoids, CMon. For any monad its Kleisli category is a full subcategory of its category of algebras so we have an embedding $e : \mathsf{Mat}_{\mathbb{N}} \hookrightarrow \mathsf{CMon}$.

3.8.1 Weak pullbacks

The results of this section were first noticed in [Sob13]—we fill out the missing details in the proofs below.

Theorem 89. $\mathsf{Mat}_{\mathbb{N}}$ *does not have pullbacks.*

In general, to show that certain limits do not exist in a given category C, we can embed C in a larger category and show that the limit of the relevant diagram lies outside of the essential image of the embedding. This works because limits are unique up to isomorphism. Here we apply this technique using the embedding $e : \mathsf{Mat}_{\mathbb{N}} \hookrightarrow \mathsf{CMon}$. As CMon is a category of algebras for a monad over Set, limits exist and can be computed as in Set. In particular, the pullback of two monoid homomorphisms $f\colon X \to Z$ and $g\colon Y \to Z$ is the set

$$T_{f,g} = \{(x, y) \in X \times Y \mid f(x) = g(y)\} \tag{3.172}$$

with commutative monoid operation inherited from $X \times Y$. As with Definition 52, let us call this the monoid of transactions of f and g. The following is just the reformulation of Lemma 53.

Lemma 90. *For any two matrices $A\colon k \to m$ and $B\colon l \to m$, $T_{eA,eB}$ is an additive monoid.*

Now we are equipped to prove that pullbacks do not exist in $\mathsf{Mat}_{\mathbb{N}}$.

Proof of Theorem 89. Consider the matrix $A\colon 2 \to 1$, $A = \begin{pmatrix} 1 & 1 \end{pmatrix}$. Then $T_{eA,eA}$ is generated by the minimal transactions

$$\left\{ \left(\begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right), \left(\begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right), \left(\begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix} \right), \left(\begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix} \right) \right\}. \tag{3.173}$$

This monoid is not isomorphic to a free monoid since, for example,

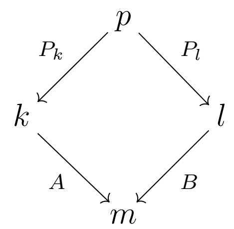
$$\left(\begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \end{pmatrix} \right) = \left(\begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right) + \left(\begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix} \right) \tag{3.174}$$

$$= \left(\begin{pmatrix}1\\0\end{pmatrix}, \begin{pmatrix}0\\1\end{pmatrix}\right) + \left(\begin{pmatrix}0\\1\end{pmatrix}, \begin{pmatrix}1\\0\end{pmatrix}\right) \tag{3.175}$$

Therefore, this pullback does not lie in the image of e and $\mathsf{Mat}_{\mathbb{N}}$ does not have pullbacks. ☐

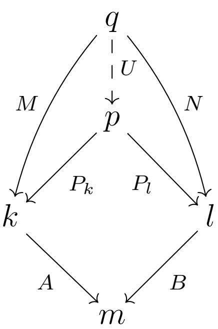
Conceptually, pullbacks do not exist in $\mathsf{Mat}_{\mathbb{N}}$ because the matrix that should satisfy the corresponding universal property is not necessarily *unique*. However, note that it always *exists*, hence the appellation "weak".

Proposition 91. $\mathsf{Mat}_{\mathbb{N}}$ *admits weak pullbacks: for every pair of matrices* $A\colon m \to k$ *and* $B\colon m \to l$*, there exists* p *and* $P_k\colon p \to k$ *and* $P_l\colon p \to l$ *such that*



(3.176)

commutes and, for every pair of matrices $M\colon q \to k$ *and* $N\colon q \to l$ *such that* $MA = NB$*, there exists a (not necessarily unique) matrix* $U\colon q \to p$ *such that*



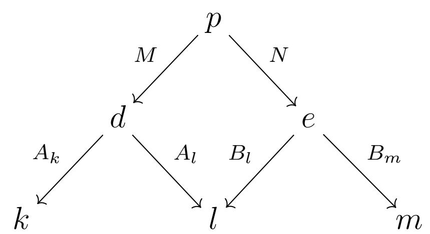
(3.177)

commutes.

Proof. Given M and N as in the statement of the proposition, $AM\mathbf{a} = BN\mathbf{a}$ by assumption and therefore $(M\mathbf{a}, N\mathbf{a})$ is an element of $T_{eA,eB}$ for every $\mathbf{a} \in \mathbb{N}^q$. Let $(\mathbf{b}_1, \mathbf{c}_1), \ldots, (\mathbf{b}_p, \mathbf{c}_p)$ be the minimal transactions (there is a finite number of them by Dickson's lemma); let P_k and P_l be the projection matrices that send a transaction to its left and right component respectively. By Lemma 53, every transaction can be obtained as a linear combination of minimal transactions, so we can define U by first choosing a decomposition of $(M\mathbf{a}, N\mathbf{a})$ into minimal elements of $T_{eA,eB}$, say $\sum_{i=1}^{p} n_i(\mathbf{b}_i, \mathbf{c}_i)$, for every $\mathbf{a} \in \mathbb{N}^q$. Then, let $U(\mathbf{a}) = \sum_{i=1}^{p} n_i\mathbf{e}_i$, for $\{\mathbf{e}_i\}_{1 \leq i \leq p}$ the basis vectors of $\mathbb{N}^p$. This mapping is not uniquely defined because there may be several decompositions of each $(M\mathbf{a}, N\mathbf{a})$ into minimal transactions but any choice of decomposition makes the relevant diagram commute. □

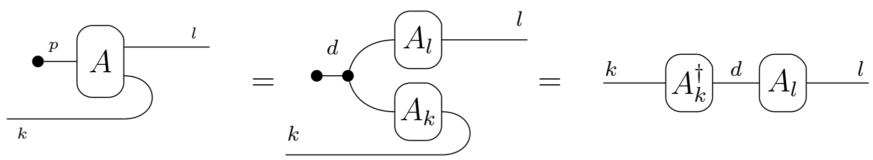
Remark 92. Weak pullbacks are not sufficient to define a category of spans of natural number matrices. We can attempt to compose such spans via weak pullbacks, but the composition is not associative, for the same reason that maps into the apex of a weak pullback are not uniquely defined by their projections, see [Sob13].

When proving that finitely generated additive relations compose (Proposition 51) we also used Dickson's lemma (packaged in Lemma 53). This is not a coincidence: the composition of additive relations works by taking the weak pullback. Indeed, since $\mathsf{Mat}_{\mathbb{N}}$ has (bi)products, matrices $A\colon d \to k + l$ are in one-to-one correspondence with pairs of matrices $A_k\colon d \to k$ and $A_l\colon d \to l$, i.e., spans $k \leftarrow d \to l$. Given two additive relations $R\colon k \to l$ and $S\colon l \to m$ with respective representing matrices $A\colon d \to k + l$ and $B\colon e \to l + m$, we can form the weak pullback $d \xleftarrow{M} p \xrightarrow{N} e$ in



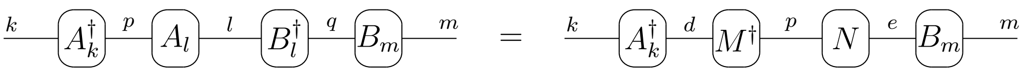
(3.178)

to obtain a representing matrix $\begin{pmatrix} A_k M \\ B_m N \end{pmatrix}$ for $R\,;\,S$. This process is simply a reformulation of the proof of Proposition 51 in more conceptual terms. At the diagrammatic level, we can also see the correspondence with the pre-normal form for Rc diagrams: it can be seen as representing additive relations as spans of matrices. Indeed, since $\mathsf{Mat}_{\mathbb{N}}$ is Cartesian, there exists A_k and A_l such that



(3.179)

Then, weak pullbacks are a recipe to compose two Rc diagrams in span form, sliding the middle cospan into a span:



(3.180)

where $d^\dagger$ represents the transpose of a diagram.

3.8.2 Minimal images

From the weak pullback, we can obtain a representing span of the composite relation $R;S$, given representing spans for R and S. However the set of minimal transactions may contain a lot of unnecessary information. In the composition of additive relations, the weak pullback is only the first step: it is followed by a factorisation step that discards all redundant transactions. In the composition of $A = \begin{pmatrix} A_k \\ A_l \end{pmatrix} : p \to k+l$ with $B = \begin{pmatrix} B_l \\ B_m \end{pmatrix} : q \to l+m$, seen as spans, the factorisation step discards all minimal transactions $(\mathbf{c}, \mathbf{d})$ of A_l and B_k, such that there exists minimal transaction $(\mathbf{e}_i, \mathbf{f}_i)$ with $A_k\mathbf{x} = \sum_{i=1}^n A_k\mathbf{e}_i$ and $B_m\mathbf{y} = \sum_{i=1}^n B_m\mathbf{f}_i$. We call such transactions *redundant*.

For relations over a regular category, this is done by factoring a span through its image. Here, the notion of image that we are looking for is not as well-behaved.

Definition 93. A *functorial factorisation system* for a category C is a pair $(\mathcal{M}, \mathcal{E})$ of collections of morphisms of C such that

- $\mathcal{M}$ and $\mathcal{E}$ contain all isomorphisms of C;
- every morphism f of C factors as $f = e;m$, with m in $\mathcal{M}$ and e in $\mathcal{E}$;
- this factorisation is functorial: given f, f' with factorisations $f = e;m$ and $f' = e';m'$, for every u and v verifying $f;v = u;f'$ there exists a unique q such that

$$\begin{array}{ccccc} \cdot & \xrightarrow{e} & \cdot & \xrightarrow{m} & \cdot \\ {\scriptstyle u}\downarrow & & \downarrow{\scriptstyle \exists ! q} & & \downarrow{\scriptstyle v} \\ \cdot & \xrightarrow[e']{} & \cdot & \xrightarrow[m']{} & \cdot \end{array} \tag{3.181}$$

  commutes.

For R a PID, Mat_R is regular and admits a well-behaved notion of image given by the (strong epimorphisms, monomorphims) factorisation system. $\mathsf{Mat}_\mathbb{N}$ is not regular but we would like the image of a matrix $A\colon k \to l$ to be the set $\mathrm{Im}A := \{\mathbf{b} \in \mathbb{N}^l \mid \exists \mathbf{a} \in \mathbb{N}^k, A\mathbf{a} = \mathbf{b}\}$. The following is immediate.

Proposition 94. *For a matrix $A\colon k \to l$, $\mathrm{Im}A$ is an additive monoid.*

The problem is that not every image in this sense is the image of a monomorphism. Monomorphisms are too rigid, as the following example demonstrates.

Example 95. Let S be the additive monoid generated by $\left\{\begin{pmatrix}2\\0\end{pmatrix}, \begin{pmatrix}1\\1\end{pmatrix}, \begin{pmatrix}0\\1\end{pmatrix}\right\}$. We claim that it is not the image of a monomorphism. Assume that there exist a matrix $A\colon k \to 2$ such that $\mathrm{Im}A = \langle G \rangle$. We will derive a contradiction.

By hypothesis, the images of the canonical basis vectors of $\mathbb{N}^k$ generate S and $\begin{pmatrix}2\\0\end{pmatrix}, \begin{pmatrix}1\\1\end{pmatrix}$ and $\begin{pmatrix}0\\1\end{pmatrix}$ are independent, so there exists three basis vectors $\mathbf{e}_1, \mathbf{e}_2$ and $\mathbf{e}_3$ such that

$$A\mathbf{e}_1 = \begin{pmatrix}2\\0\end{pmatrix}, A\mathbf{e}_2 = \begin{pmatrix}1\\1\end{pmatrix}, A\mathbf{e}_3 = \begin{pmatrix}0\\1\end{pmatrix}. \tag{3.182}$$

Since $\begin{pmatrix}2\\2\end{pmatrix} = 2\begin{pmatrix}1\\1\end{pmatrix} = \begin{pmatrix}2\\0\end{pmatrix} + 2\begin{pmatrix}0\\1\end{pmatrix}$, there would also be $\mathbf{d} \in \mathbb{N}^3$ such that $A\mathbf{d} = \begin{pmatrix}2\\2\end{pmatrix}$. Therefore we would have $A\mathbf{d} = A(2\mathbf{e}_2) = A(\mathbf{e}_1 + 2\mathbf{e}_3)$ and, because A is a monomorphism, $2\mathbf{e}_2 = \mathbf{e}_1 + 2\mathbf{e}_3$. But the $\mathbf{e}_i, 1 \le i \le 3$ are basis vectors so this is impossible.

The right notion of image is given by the collection of minimal matrices.

Definition 96. We call a matrix *minimal* when its columns are independent.

Call Min the collection of minimal matrices in $\mathsf{Mat}_{\mathbb{N}}$. Minimal matrices are a not necessarily monomorphisms but they correspond uniquely to images in $\mathsf{Mat}_{\mathbb{N}}$.

Lemma 97. *Every $\mathbb{N}$-matrix factors as a split epimorphism followed by a minimal matrix. Furthermore, the latter is unique (up to isomorphism).*

Proof. For every matrix $A\colon k \to l$, the set $\mathrm{Im}A$ is an additive monoid and therefore admits a Hilbert basis, by Theorem 46. Let $B = \{\mathbf{e}_i\}_{1 \le i \le k}$ be the canonical basis of $\mathbb{N}^k$. Since $\{f(\mathbf{e}_i), 1 \le i \le k\}$ is a generating set of $\mathrm{Im}A$, there exists a subset $H = \{\mathbf{e}_{i_1}, \ldots, \mathbf{e}_{i_h}\}$ of B such that $\{A\mathbf{e}_{i_1}, \ldots A\mathbf{e}_{i_h}\}$ is the Hilbert basis of $\mathrm{Im}A$. Define the matrix $M : h \to k$ to be restriction of A to the subspace spanned by H. It is clear that M is minimal by construction.

In addition, if H is a Hilbert basis for $\mathrm{Im}A$, it means that, for every element of $\mathbf{e}_j \in B$ there exists a decomposition $A\mathbf{e}_j = \sum_{n=1}^{h} p_n A\mathbf{e}_{i_n}$. If we fix one such decomposition for each $\mathbf{e}_j \in B$, we obtain a matrix $E : k \to h$. Moreover, E is a split epimorphism with the inclusion of H into B as section, extended by linearity. Thus we have $A = ME$ as we wanted.

Finally for unicity, suppose that A factors through another minimal matrix $M'\colon h' \to l$. Since the Hilbert basis is unique, we can exhibit an isomorphism $h' \to h$ by sending $\mathbf{e}'_i$ to $\mathbf{e}_j$ if $A\mathbf{e}'_i = A\mathbf{e}_j$ and extending by linearity. □

Unfortunately, minimal matrices do not correspond to any of the usual categorical concepts of images.

Proposition 98. Min *cannot be the right class of a factorisation system containing split epimorphisms in* $\mathsf{Mat}_{\mathbb{N}}$*.*

Proof. Consider the following counter-example:

$$\begin{array}{ccc} 4 & \xrightarrow{1_4} & 4 \\ & \searrow^{A} & \downarrow{\scriptstyle A} \\ & & 2 \end{array} \tag{3.183}$$

where A is given by

$$A = \begin{pmatrix} 0 & 2 & 1 & 2 \\ 1 & 0 & 1 & 2 \end{pmatrix} \tag{3.184}$$

First note that $A\left(0,0,0,1\right)^T$ can be decomposed in two different ways:

$$A\begin{pmatrix}0\\0\\0\\1\end{pmatrix} = \begin{pmatrix}2\\2\end{pmatrix} = 2A\begin{pmatrix}0\\0\\1\\0\end{pmatrix} = 2A\begin{pmatrix}1\\0\\0\\0\end{pmatrix} + A\begin{pmatrix}0\\1\\0\\0\end{pmatrix} \tag{3.185}$$

So A is not minimal. The set

$$\left\{A\begin{pmatrix}1\\0\\0\\0\end{pmatrix}, A\begin{pmatrix}0\\1\\0\\0\end{pmatrix}, A\begin{pmatrix}0\\0\\1\\0\end{pmatrix}\right\} = \left\{\begin{pmatrix}0\\1\end{pmatrix}, \begin{pmatrix}2\\0\end{pmatrix}, \begin{pmatrix}1\\1\end{pmatrix}\right\}$$

is the Hilbert basis of $\mathrm{Im}A$. As a result, we can factor A into a pair of a split epimorphism and minimal morphism M in two different ways. Let $E, E' \colon 4 \to 3$ be given by

$$E = \begin{pmatrix} 1 & 0 & 0 & 2 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \tag{3.186}$$

and

$$E' = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 2 \end{pmatrix} \tag{3.187}$$

Then we need to find a unique map such that the square in the following diagram commutes:

$$\begin{array}{ccc} 4 & \xrightarrow{1_4} & 4 \\ {\scriptstyle E}\downarrow & & \downarrow{\scriptstyle E'} \\ 3 & \overset{?}{\dashrightarrow} & 3 \\ & {}_{M}\searrow \quad \swarrow_{M} & \\ & 2 & \end{array} \tag{3.188}$$

But this is impossible by construction of E and E'. To see this imagine there exists such a map $B : 3 \to 3$. Then $BE \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 2 \end{pmatrix}$ which means that $B \begin{pmatrix} 2 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 2 \end{pmatrix}$. But

$$B \begin{pmatrix} 2 \\ 1 \\ 0 \end{pmatrix} = 2B \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} + B \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} = 2BE \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} + BE \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} \tag{3.189}$$

$$= 2E' \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} + E' \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 2 \\ 1 \\ 0 \end{pmatrix} \tag{3.190}$$

so we have a contradiction. □

Minimal matrices are not closed under composition either.

Proposition 99. Min *is* not *a subcategory of* $\mathsf{Mat}_{\mathbb{N}}$.

Proof. The following counter-example suffices see that minimal maps do not compose: let $A\colon 2 \to 2$ and $B\colon 2 \to 1$ be the two matrices

$$A = \begin{pmatrix} 0 & 2 \\ 3 & 0 \end{pmatrix} \qquad B = \begin{pmatrix} 3 & 2 \end{pmatrix} \tag{3.191}$$

They are minimal but $BA \begin{pmatrix} 1 \\ 0 \end{pmatrix} = 6 = BA \begin{pmatrix} 0 \\ 1 \end{pmatrix}$. □

Remark 100. If the minimal image has precisely the right granularity to obtain additive relations from spans of matrices, it is not the only notion of image from which we can obtain an associative composition. In [Sob13, Section 5], the author considers a prop $\mathsf{InjSpan}_{\mathbb{N}}$ of jointly-injective spans of matrices. Here, a matrix $k \to l$ is said to be injective if its associated map $\underline{k} \to \mathbb{N}^l$ is injective in Set. Injective maps, along with surjective maps, form a suitable factorisation system in Set. Such jointly-injective spans can then be composed via weak pullback followed by a factorisation step and this composition is associative [Sob13, Section 5, Proposition 1].

For a matrix $A\colon k \to l$, the corresponding Set map $\underline{k} \to \mathbb{N}^l$ is injective iff $A\mathbf{e}_i = A\mathbf{e}_j$ implies $i = j$, for $\{\mathbf{e}_i\}_{1\leq i\leq k}$ the canonical basis of $\mathbb{N}^k$. Clearly, minimal maps are injective in this sense but injective maps are not necessarily minimal.

In the composition of $A = \begin{pmatrix} A_k \\ A_l \end{pmatrix}\colon p \to k+l$ with $B = \begin{pmatrix} B_l \\ B_m \end{pmatrix}\colon q \to l+m$ in $\mathsf{InjSpan}_{\mathbb{N}}$, the factorisation step identifies all minimal transactions $(\mathbf{x}, \mathbf{y})$ and $(\mathbf{x}', \mathbf{y}')$ of A_l and

B_k such that $A_k\mathbf{x} = A_k\mathbf{x}'$ and $B_m\mathbf{y} = B_m\mathbf{y}'$. Call such transactions *indistinguishable*. This should be contrasted with factoring through the minimal image—a process that discards *redundant* minimal transactions. Clearly indistinguishable transactions are redundant but the converse if not true. Therefore, composition in $\mathsf{InjSpan}_{\mathbb{N}}$ retains more information about A and B.

As a result, there is a prop morphism $\mathsf{Dis}\colon \mathsf{InjSpan}_{\mathbb{N}} \to \mathsf{AddRel}$, that associates

$$\mathsf{Dis}(A) := \{(\mathbf{a}, \mathbf{b}) \mid \exists \mathbf{x} \in \mathbb{N}^p, (\mathbf{a}, \mathbf{b}) = (A_k\mathbf{x}, A_l\mathbf{x})\} \tag{3.192}$$

to $A = \begin{pmatrix} A_k \\ A_l \end{pmatrix}\colon p \to k + l$. This is functorial because composition via weak pullback is sound for AddRel, by Remark 92.

3.8.3 AddRel is almost a category of relations

Even if minimal matrices do not give rise to a well-behaved notion of image in $\mathsf{Mat}_{\mathbb{N}}$, it is possible to identify AddRel as a *subcategory* of a category of relations over a regular category, namely that of relations over the category of commutative monoids. Let us explain this further.

CMon is regular so we can form the category of relations over it—call it $\mathsf{Rel}(\mathsf{CMon})$: a relation $M \to N$ is a monomorphism $R \hookrightarrow M \times N$. Recall that monomorphisms of commutative monoids are injective monoid homomorphisms. If M and N are free and finitely generated, it means that they are isomorphic to some $\mathbb{N}^d$ for some integer $d \geq 0$, and so is their product. It follows immediately that the full subcategory of $\mathsf{Rel}(\mathsf{CMon})$ on finitely generated free monoids is (not necessarily finitely generated) that of (not necessarily finitely generated) additive relations. And finally, the subcategory of those that are finitely generated is equivalent to AddRel!

Finally, if CMon is finitely (co)complete, we can consider the category of spans of commutative monoids, $\mathsf{Span}(\mathsf{CMon})$ and, in particular its full subcategory on the free monoids, $\mathsf{Span}_M(\mathsf{CMon})$. Note that, in the latter category, the apex of the span can be a non-free monoid (otherwise, as we have seen, spans of homomorphisms of free monoids, a.k.a spans of matrices, do not form a category). Furthermore, according to [RGS99, Theorem 3.11], additive monoids are exactly the finitely generated, nonnegative, cancellative and torsion-free commutative monoids. And all of these properties are preserved by pullbacks so that we can consider the subcategory of $\mathsf{Span}(\mathsf{CMon})$ whose objects are the finitely generated free monoids and whose morphisms are spans with an additive monoid as apex. We conjecture that this smc is equivalent to a prop whose presentation is the same as that of Rc, without the special law of the Frobenius monoid. This fact might help to build a modular account of AddRel but we leave its proof for future work.

Chapter 4

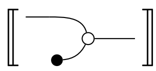
Picturing resources in stateless concurrency

While the resource calculus is a rudimentary coordination language that can express basic forms of synchronisation and nondeterminism, there are many non-additive phenomena in concurrency for which it is not sufficiently expressive. One notable example is that of *mutual exclusion*. Mutual exclusion [Dij65], one of the most influential synchronisation mechanisms, guarantees that two or more processes are prevented from accessing a given resource at the same time. It represents an inhibitory pattern of interaction and is used, for example, to prevent race-conditions in shared memory infrastructures. Numerous protocols and even hardware-assisted constructs such as *compare-and-swap* have been invented to enforce this property.

In this chapter, we extend the resource calculus to account for mutual exclusion and more general forms of inhibitory behaviour. The surprising fact is that these synchronisation mechanisms are all revealed to be instances of *affine* phenomena. Affine algebra (and geometry) is perhaps better known to the reader as linear algebra over a field where one forgot the origin. As we will see, it is possible to introduce similar notions over $\mathbb{N}$. The key is being able to express a *constant* quantity of resources different from zero, leading to the *affine resource calculus*. In fact, it is sufficient to have access to the constant 1, that we will depict as the following $0 \to 1$ diagram:

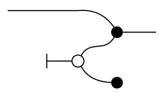
(4.1)

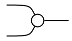
A remarkable property of the resource calculus is that it captures the order on $\mathbb{N}$. Specifically, the following diagram forces the value observed on the right to be greater than that on the right:



$$= \{(n,m) \mid \exists k \in \mathbb{N}, n+k=m\} = \{(n,m) \mid n \leq m\} \qquad (4.2)$$

Combining this diagram with the affine constant $\vdash\!\!-$, we can easily derive a connector denoting mutual exclusion. First, using $\vdash\!\!-$, we can define a wire whose bandwidth is ≤ 1, i.e., that can only carry 0 or 1:

$—\!\!+\!\!— \;:=$  (4.3)

Then, mutual exclusion can be obtained as the composition of  with $—\!\!+\!\!—$, whose denotation is the following relation:

$$\left\{ \left(\begin{pmatrix} n \\ m \end{pmatrix}, n+m \right) \mid n+m \leq 1 \right\} = \left\{ \left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, 0 \right), \left(\begin{pmatrix} 0 \\ 1 \end{pmatrix}, 1 \right), \left(\begin{pmatrix} 1 \\ 0 \end{pmatrix}, 1 \right) \right\} \tag{4.4}$$

This captures the desired behaviour: the two wires on the right cannot be activated simultaneously. Characterising the sort of relations that this new syntax allows us to express as well as deriving a sound and complete equational theory for it is the task of this chapter.

We start with a preparatory section showing how the usual graphical calculus for matrices can be extended with a single constant and two equations to obtain a sound and complete calculus for affine maps (Section 4.1). Following this simple result, we tackle the more intricate case of extending the syntax of the whole resource calculus with the same affine constant. Once again, we characterise its semantics—in terms of so-called *polyhedral relations* (Section 4.2)–and exhibit a presentation for it (Section 4.3). Finally, as a case study, we show how the calculus of *stateless connectors*, an existing coordination language for the design and specification of distributed systems, embeds naturally into our syntax (Section 4.4).

Remark 101. The results of Section 4.3 were formulated and proved in collaboration with Filippo Bonchi, Paweł Sobociński and Fabio Zanasi. The author is also indebted to Josh Holland for noticing that a first version of the affine resource calculus was incomplete. Finally Filippo Bonchi suggested the encoding of the *stateless connectors* appearing in Section 4.4 to the author.

4.1 From linear to affine maps

In this section, we define a prop of affine transformations over a semiring R and show how to obtain a presentation for it. We will assume that the additive operation of R is cancellative, as it is necessary to derive a normal form for affine maps from Theorem 103.

Definition 102. A map $f : \mathsf{R}^k \to \mathsf{R}^l$ is *affine* if there exists a $m \times n$ matrix A and $\mathbf{b} \in \mathsf{R}^l$ such that

$$f(\mathbf{x}) = A\mathbf{x} + \mathbf{b}$$

for all $\mathbf{x} \in \mathsf{R}^k$. We call the pair $(A, \mathbf{b})$ the representation of f.

The next theorem is key to lift the normal form for the prop of matrices to that of affine maps.

Theorem 103. *The representation of an affine map is unique.*

Proof. Assume that $(A, \mathbf{b})$ and $(A', \mathbf{b}')$ are representations for an affine map f. First, $f(\mathbf{0}) = A\mathbf{0} + \mathbf{b} = \mathbf{b}$ and for the same reason $f(\mathbf{0}) = \mathbf{b}'$ so $\mathbf{b} = \mathbf{b}'$. Now, $f(\mathbf{x}) = A\mathbf{x} + \mathbf{b} = A'\mathbf{x} + \mathbf{b}$ and, by cancellativity of R, $A\mathbf{x} = A'\mathbf{x}$ for all $\mathbf{x} \in \mathsf{R}^k$. Evaluating at the elements of the canonical basis of R^k, we conclude that $A = A'$. □

We can now identify affine maps with their representation and will write $(A, \mathbf{b})$ directly for the map it represents.

Proposition 104. *For* $(A, \mathbf{b})\colon k \to l$ *and* $(A', \mathbf{b}')\colon l \to m$,

$$(A, \mathbf{b})\,;\,(A', \mathbf{b}') = A'A\mathbf{x} + (A'\mathbf{b} + \mathbf{b}')$$

Proof. For all $\mathbf{x} \in \mathsf{R}^k$, $A'(A\mathbf{x} + \mathbf{b}) + \mathbf{b}' = A'A\mathbf{x} + (A'\mathbf{b} + \mathbf{b}')$ □

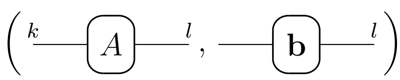
Corollary 105. *The composition of two affine maps is affine.*

We can thus define a prop of affine transformations.

Definition 106. Let Aff be the prop with

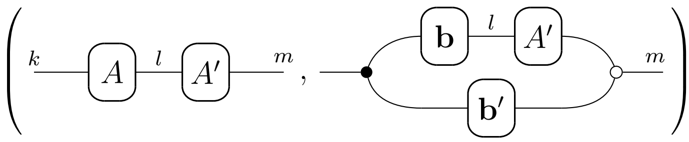
- affine transformations $\mathsf{R}^k \to \mathsf{R}^l$ as morphisms $k \to l$;
- composition given by function composition (or–which is the same–by the formula of Proposition 104);
- the monoidal product of two affine maps given by $\left(A_1 \oplus A_2, \begin{pmatrix}\mathbf{b}_1\\ \mathbf{b}_2\end{pmatrix}\right)$ for $(A_1, \mathbf{b}_1)\colon k_1 \to l_1$ and $(A_2, \mathbf{b}_2)\colon k_2 \to l_2$.

Using the graphical calculus for matrices, we can represent affine maps as pairs of diagrams,



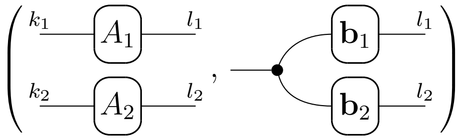
(4.5)

with composition depicted by



(4.6)

and monoidal product given by



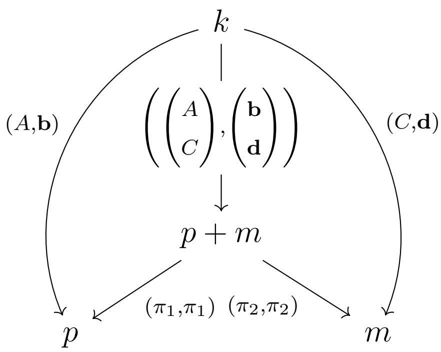
(4.7)

Theorem 107. Aff *is Cartesian.*

Proof. Let $(A, \mathbf{b})\colon k \to l$ and $(C, \mathbf{d})\colon k \to m$ be two affine maps. Let

$$\langle (A, \mathbf{b}), (C, \mathbf{d}) \rangle = \left(\begin{pmatrix} A \\ C \end{pmatrix}, \begin{pmatrix} \mathbf{b} \\ \mathbf{d} \end{pmatrix} \right) \tag{4.8}$$

with projections as in Mat_R for each component. Clearly, for the same reason as in Mat_R, the following diagram commutes:



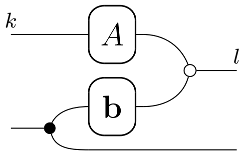
(4.9)

Moreover, it satisfies the universal property of the product for the same reason that the direct sum is a categorical product in $\mathsf{Mat}_\mathbb{N}$ (it is sufficient to compose the resulting map with the usual projections to show uniqueness). □

Homogenisation. Linear maps are affine but, in general, affine maps are not linear. However there is a simple procedure to obtain a linear map from an affine map by introducing a dummy variable. It is a well-known technique of affine and convex geometry.

Definition 108. Let $(A, \mathbf{b})\colon k \to l$ be an affine map. The *homogenisation* of $(A, \mathbf{b})$ is the matrix $Lf : k + 1 \to l + 1$ defined by

$$L(A, \mathbf{b}) = \begin{pmatrix} A & \mathbf{b} \\ 0 & 1 \end{pmatrix} =$$



(4.10)

Proposition 109. $L\colon \mathsf{Aff}_\mathsf{R} \to \mathsf{Mat}_\mathsf{R}$ *is an oplax monoidal functor.*

Proof. L is defined on objects by $Lk = k+1$. That L is functorial follows from $L(A',\mathbf{b}')L(A,\mathbf{b}) = \begin{pmatrix} A' & \mathbf{b}' \\ 0 & 1 \end{pmatrix}\begin{pmatrix} A & \mathbf{b} \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} A'A & A'\mathbf{b}+\mathbf{b}' \\ 0 & 1 \end{pmatrix} = (A,\mathbf{b})\,;\,(A',\mathbf{b}')$ and $L(\mathsf{id}_k, \mathbf{0}) = \mathsf{id}_{k+1}$. In addition, L is oplax monoidal with the natural transformations $\phi_{k,l}\colon k+l+1 \to k+1+l+1$ given by $\phi_{k,l}\begin{pmatrix} \mathbf{x} \\ \mathbf{y} \\ r \end{pmatrix} = \begin{pmatrix} \mathbf{x} \\ r \\ \mathbf{y} \\ r \end{pmatrix}$ and $\phi_0\colon 1 \to 0$, for $r \in \mathsf{R}$, the only possible such linear map. □

In fact, the categorically inclined reader may recognise this construction as an adjunction. Let us flesh out this claim. Firstly, note that there is a forgetful functor $U\colon \mathsf{Mat}_\mathsf{R} \to \mathsf{Aff}_\mathsf{R}$ that takes a matrix to the affine map represented by $(A, \mathbf{0})$.

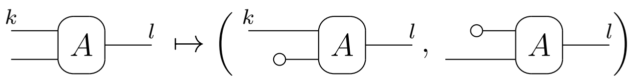
Proposition 110. $L\colon \mathsf{Aff}_\mathsf{R} \to \mathsf{Mat}_\mathsf{R}$ *is left adjoint to* U.

Proof. We construct a natural bijection

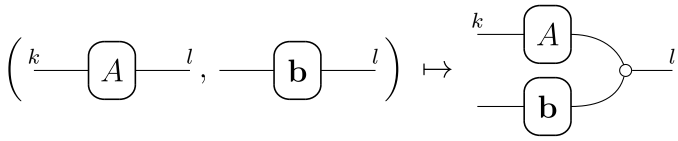
$$\mathsf{Mat}_\mathsf{R}(k+1, l) \cong \mathsf{Aff}_\mathsf{R}(k, l) \tag{4.11}$$

Let $f\colon \mathsf{Mat}_\mathsf{R}(k+1,l) \to \mathsf{Aff}_\mathsf{R}(k,l)$ be defined by $f\begin{pmatrix} A \\ \mathbf{b} \end{pmatrix} = (A, \mathbf{b})$ and $g\colon \mathsf{Aff}_\mathsf{R}(k,l) \to \mathsf{Mat}_\mathsf{R}(k+1,l)$ defined by $g(A,\mathbf{b}) = \begin{pmatrix} A \\ \mathbf{b} \end{pmatrix}$. These maps are clearly inverses to each other and natural in both variables. □

In graphical terms, the adjunction is witnessed by the following bijections:

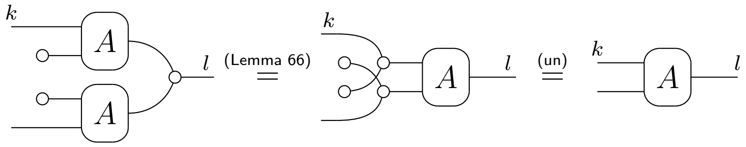


(4.12)

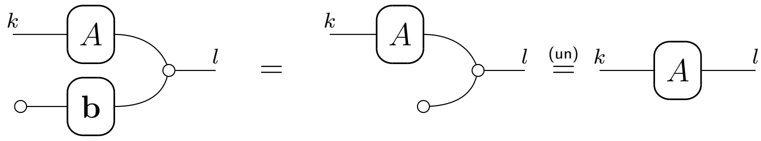


(4.13)

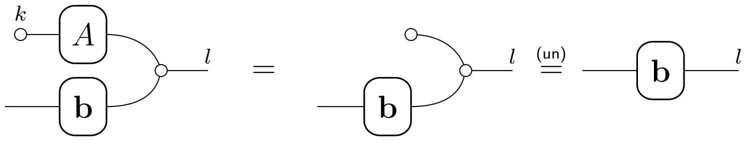
which are one-to-one since



(4.14)



(4.15)

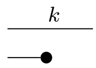


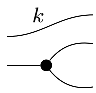
(4.16)

Every adjunction induces a monad and a comonad. We are interested in the latter, whose underlying functor LU is just $(-) + 1 \colon \mathsf{Mat_R} \to \mathsf{Mat_R}$, with counit and comultiplication given by the linear maps

$$\iota_k : \mathsf{R}^{k+1} \to \mathsf{R}^k, \iota_k \begin{pmatrix} \mathbf{x} \\ r \end{pmatrix} = \mathbf{x} \text{ and } \delta_k : \mathsf{R}^{k+1} \to \mathsf{R}^{k+2}, \delta_k \begin{pmatrix} \mathbf{x} \\ r \end{pmatrix} = \begin{pmatrix} \mathbf{x} \\ r \\ r \end{pmatrix} \tag{4.17}$$

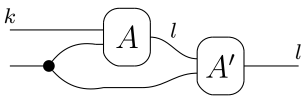
Or, graphically,

$\iota_k =$ 

and $\delta_k =$ 

(4.18)

Intuitively, the counit deletes the additional component and the comultiplication duplicates it. This is the infrastructure that takes care of the book-keeping required by the extra wire of the homogenisation construction. This is best illustrated by the coKleisli composition of $A \colon k+1 \to l$ and $A' \colon l+1 \to m$:



(4.19)

Proposition 111. $\mathsf{Aff_R}$ *is isomorphic to the coKleisli category of LU.*

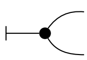
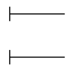
Proof. Let $\mathsf{Mat}_\mathsf{R}^{LU}$ be the coKleisli category of LU. The maps realising the homset bijection of Proposition 110 extend to strict monoidal functors $\mathsf{Mat}_\mathsf{R}^{LU} \to \mathsf{Aff_R}$ and $\mathsf{Aff_R} \to \mathsf{Mat}_\mathsf{R}^{LU}$ that are inverses of each other. □

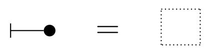
4.1.1 Presenting affine transformations

Homogenisation means, roughly speaking, that affine maps can be thought of as matrices with an extra dangling wire for the additional dimension.

In the diagrammatic syntax, we need to introduce a formal notation for these dangling wires. To this effect, we introduce a new generator $\vdash$ that we interpret as the constant 1. We need to find the right equations to axiomatise its behaviour. Let U be the free prop on the single generator $\vdash$ and no equations.

The characterisation of $\mathsf{Aff_R}$ as the coKleisli category of $(-)+1$ suggests that we should quotient $\mathsf{Mat_R} + \mathsf{U}$ with two additional equations:

 $=$  (dup)

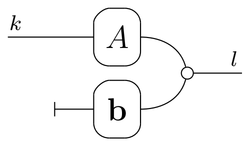
 (del)

These equations say that $\vdash\!\!-$ can be deleted and copied by the comonoid structure, just like $\circ\!\!-$. This is because several hanging wires represent the same extra dimension. It can be understood as the axiomatic counterpart of the coKleisli composition depicted in equation (4.27).

Let us prove that this is enough. Call Ac_R the prop $\mathsf{Mat}_\mathsf{R} + \mathsf{U}$ quotiented by (dup) and (del); $I_\mathsf{R} \colon \mathsf{Ac}_\mathsf{R} \to \mathsf{Aff}_\mathsf{R}$ given by extending the prop morphism $\mathsf{Bi}_\mathsf{R} \to \mathsf{Mat}_\mathsf{R}$ of Theorem 63 with $I_\mathsf{R}(\vdash\!\!-) := 1$. Checking functoriality amounts to verifying the soundness of the additional two equations: $\begin{pmatrix}1\\1\end{pmatrix} 1 = \begin{pmatrix}1\\1\end{pmatrix}$ and $\begin{pmatrix}0\end{pmatrix} 1 = 0$.

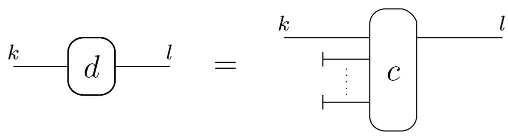
Theorem 112. $I_\mathsf{R} \colon \mathsf{Ac}_\mathsf{R} \to \mathsf{Aff}_\mathsf{R}$ *is an isomorphism of props.*

Proof. First, I_R is full since the affine map with representation $(A, \mathbf{b})$ is the image of the diagram

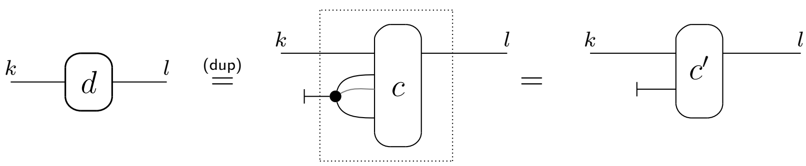
 (4.20)

identifying A, $\mathbf{b}$ and their diagrams in Bi_R (cf. Theorem 63).

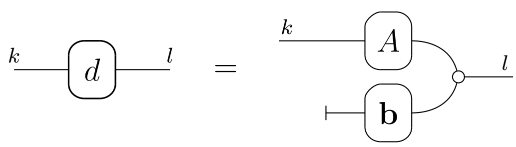
Secondly, I_R is faithful. To prove this, let d be a diagram in Ac_R. Using the naturality of the symmetric monoidal structure, we can write d as

 (4.21)

for some matrix c, i.e., a diagram in the image of the embedding $\mathsf{Bi}_\mathsf{R} \hookrightarrow \mathsf{Ac}_\mathsf{R}$. Then,

 (4.22)

for c' the diagram in the dotted box. Finally we can find matrix diagrams $\mathbf{b} : 1 \to m$ and $A : n \to m$ such that

 (4.23)

By Theorems 103 and 63, this form uniquely characterises the corresponding affine transformation. □

4.2 Polyhedral relations

4.2.1 Discrete polyhedra

We now define the affine (or inhomogeneous) counterpart of additive monoids. This notion is not standard and we are not aware of any existing reference for it.

First some notation: let

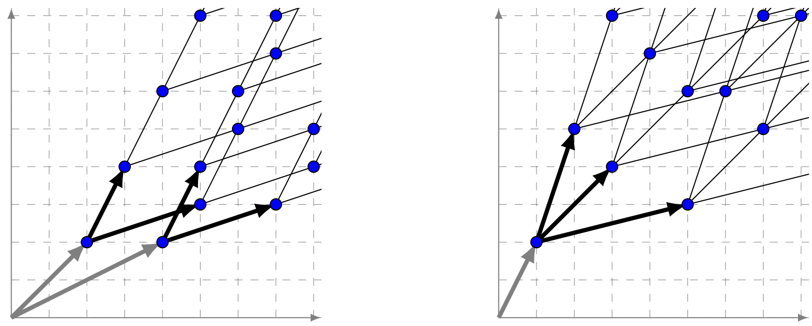
$$P(B, D) = \bigcup_{\mathbf{b} \in B} \Big\{ \mathbf{b} + \langle D \rangle \Big\}. \tag{4.24}$$

We say that this set is the *polyhedron* generated by $B, D \subseteq \mathbb{N}^d$. Elements of B are called *base points* and those of D *directions*.

Definition 113. A *discrete polyhedron* is a set $Q \subseteq \mathbb{N}^d$ for which there exist finite $B, D \subseteq \mathbb{N}^d$ such that $Q = P(B, D)$.

If $B = \{\mathbf{a}\}$ is a singleton, $Q = P(B, D)$ is simply the translation of an additive monoid by $\mathbf{a}$); if $B = \{\mathbf{0}\}$, Q is an additive monoid. Thus every additive monoid Q is a polyhedron: take $B = \{\mathbf{0}\}$ and D to be a generating set of Q. A polyhedron can be seen as a finite union of translated additive monoids.

Example 114. As for additive monoids, we can plot polyhedra in the lattice $\mathbb{N}^d$. For example, the two discrete polyhedra $P(\{(2,2),(4,2)\},\{(3,1),(1,2)\})$ and $P(\{(1,2)\}, \{(4,1),(2,2),(1,3)\})$ can be plotted in the plane, respectively as:



Note that every *finite* subset S of $\mathbb{N}^d$ is also polyhedral (by setting $B = S$ and $D = \{\mathbf{0}\}$). Finally, $\varnothing$ is polyhedral (e.g. by taking $B = \varnothing$) but not additive.

Homogenisation. As for polyhedra and cones in $\mathbb{R}$, many facts about discrete polyhedra can be deduced from their homogenenous counterparts, namely additive monoids. In this paragraph, we present a version of the homogenisation construction for discrete polyhedra that will allow us to relate the two notions systematically. In particular, homogenisation will allow us to adapt the normal form from Section 3.6 to a normal form for polyhedral relations, and to derive the completeness of the equational theory given below, in Section 4.3. As before, we can associate an additive monoid to every discrete polyhedron in a canonical way by embedding it into a lattice with an additional dimension. For a set $X \subseteq \mathbb{N}^d$, let

$$X^1 = \left\{ \begin{pmatrix} 1 \\ \mathbf{a} \end{pmatrix} \mid \mathbf{a} \in X \right\} \subseteq \mathbb{N}^{d+1} \quad \text{and} \quad X^0 = \left\{ \begin{pmatrix} 0 \\ \mathbf{a} \end{pmatrix} \mid \mathbf{a} \in X \right\} \subseteq \mathbb{N}^{d+1} \tag{4.25}$$

Definition 115. Let $Q = P(B, D) \subseteq \mathbb{N}^d$. Its *homogenisation* is the additive monoid $\widehat{Q}$ defined by $\widehat{Q} = \left\langle B^1 \cup D^0 \right\rangle$.

The homogenisation satisfies the following immediate property.

Lemma 116. *For a discrete polyhedron Q, $\mathbf{a} \in Q$ if and only if $\begin{pmatrix} 1 \\ \mathbf{a} \end{pmatrix} \in \widehat{Q}$.*

Using Theorem 46 we can eliminate redundancy in $B^1 \cup D^0$ to find the Hilbert basis $H(\widehat{Q})$. This characterises Q (not just $\widehat{Q}$) uniquely as long as it is *nonempty.*

Corollary 117. *Given nonempty polyhedral relations Q and Q', $Q = Q'$ iff $\widehat{Q}$ and $\widehat{Q'}$ have the same Hilbert basis.*

Proof. A consequence of the uniqueness of the Hilbert basis (Theorem 46) and of Lemma 116. □

4.2.2 The prop of polyhedral relations

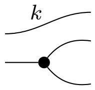
As before, we can identify subsets of $\mathbb{N}^k \times \mathbb{N}^l$ with those of $\mathbb{N}^{k+l}$ through the canonical isomorphism.

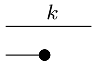
Definition 118. A *polyhedral relation* $R\colon k \to l$ is a discrete polyhedron $R \subseteq \mathbb{N}^k \times \mathbb{N}^l$.

Next we focus on the sub-prop $\mathsf{PolyRel}$ of $\mathsf{Rel}_{\mathbb{N}}$ with polyhedral relations as morphisms. For this to make sense, we need to prove that polyhedral relations are closed under composition and monoidal product, just as we did for additive relations in Section 3.3. Closure under monoidal product is straightforward. To prove closure under composition, we want to combine Proposition 51 with the homogenisation procedure. First, we need to choose whether the additional dimension is in the domain or the codomain—we choose the former to coincide with the coKleisli composition of the previous section.

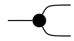
But if $R : k \to l$ and $S : l \to p$ are polyhedral relations, we cannot compose their homogenisations $\widehat{R} : k+1 \to l$ and $\widehat{S} : l+1 \to p$ in AddRel, because the types do not match.

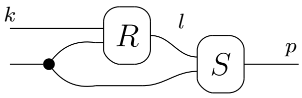
Following the same reasoning as for affine maps, we can interpret the homogenisation construction as embedding polyhedral relations into a subcategory of the coKleisli category of the comonad whose underlying functor is $(-)+1 : \mathsf{AddRel} \to \mathsf{AddRel}$, with structural natural transformations $\delta\colon (-)+1 \to (-)+2$ and $\iota\colon (-)+1 \to \mathsf{id}$:

$\mu_k =$ 

and $\epsilon_k =$ 

(4.26)

The monad laws are an immediate consequence of the comonoid equations for .
As before, for $R : k+1 \to l$ and $S : l+1 \to p$ two additive relations, their coKleisli composition $R \widehat{;} S$ can be represented in Rc as



(4.27)

We are now equipped to prove the following key property.

Proposition 119. *The composition of two polyhedral relations is a polyhedral relation.*

Proof. Let $R : k \to l$ and $S : l \to m$ be polyhedral relations and $R;S$ their composite. We can obtain the base points and directions of $R;S$ from the Hilbert basis H of the additive relation $\widehat{R} \widehat{;} \widehat{S}$.

If $(\mathbf{a}, \mathbf{c}) \in R;S$, there exists $\mathbf{b} \in \mathbb{N}^k$ such that $(\mathbf{a}, \mathbf{b}) \in R$ and $(\mathbf{b}, \mathbf{c}) \in S$. So $\left(\begin{pmatrix}1\\ \mathbf{a}\end{pmatrix}, \mathbf{b}\right) \in \widehat{R}$ and $\left(\begin{pmatrix}1\\ \mathbf{b}\end{pmatrix}, \mathbf{c}\right) \in \widehat{S}$ and therefore $\left(\begin{pmatrix}1\\ \mathbf{a}\end{pmatrix}, \mathbf{c}\right) \in \widehat{R} \widehat{;} \widehat{S}$. We can decompose this last pair into a weighted sum of elements of H:

$$\left(\begin{pmatrix}1\\ \mathbf{a}\end{pmatrix}, \mathbf{c}\right) = \left(\begin{pmatrix}1\\ \mathbf{f}\end{pmatrix}, \mathbf{g}\right) + \sum_{i=1}^{m} p_i \left(\begin{pmatrix}0\\ \mathbf{d}_i\end{pmatrix}, \mathbf{e}_i\right) \tag{4.28}$$

where

$$\left(\begin{pmatrix}1\\ \mathbf{f}\end{pmatrix}, \mathbf{g}\right), \left(\begin{pmatrix}0\\ \mathbf{d}_i\end{pmatrix}, \mathbf{e}_i\right) \in H \text{ and } p_i \in \mathbb{N} \text{ for all } 1 \le i \le m \in \mathbb{N}. \tag{4.29}$$

Then,

$$(\mathbf{a}, \mathbf{c}) = (\mathbf{f}, \mathbf{g}) + \sum_{i=1}^{m} p_i (\mathbf{d}_i, \mathbf{e}_i). \tag{4.30}$$

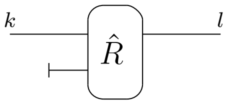
Therefore, since $(\mathbf{a}, \mathbf{c})$ was arbitrary, we conclude that $R;S = P(B, D)$, with

$$B = \left\{ (\mathbf{f}, \mathbf{g}) \mid \left(\begin{pmatrix}1\\ \mathbf{f}\end{pmatrix}, \mathbf{g}\right) \in H \right\} \text{ and } D = \left\{ (\mathbf{d}, \mathbf{e}) \mid \left(\begin{pmatrix}0\\ \mathbf{d}\end{pmatrix}, \mathbf{e}\right) \in H \right\}. \tag{4.31}$$

□

4.3 The affine resource calculus

What follows is the axiomatic counterpart of homogenisation for polyhedral relations. As for affine maps, homogenisation involves keeping track of an extra wire. We use the same generator ⊢— to plug this wire, obtaining a diagram for a polyhedral relation R with homogenisation $\widehat{R}$:



(4.32)

For the axiomatisation, homogenisation means that the equational theory of Figure 3.1 does most of the heavy lifting. Indeed, it is enough to characterise the behaviour of ⊢—. A quick semantic analysis leads us to four fundamental equations:

Definition 120. Let Rc_a be the prop freely generated over the same signature as Rc with the additional generator ⊢— and equations (dup), (del), ($\varnothing$) and (cons).

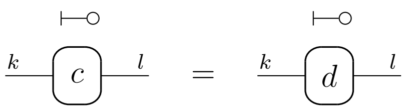
We will show that Rc_a gives a sound and complete calculus for $\mathsf{PolyRel}$. First, let us explain the four new axioms. The first two are the same as those used for affine maps and can be seen as inherited from this prop. The third equation is justified by the possibility of expressing the empty set, by, for example,

$$[\![\vdash\!\!-\!\circ]\!] = \{(\bullet, 1)\} \mathbin{;} \{(0, \bullet)\} = \varnothing. \tag{4.33}$$

As we have mentioned previously, $\varnothing$ is an example of a polyhedral relation that is not additive. Since for any R and S in $\mathsf{Rel}_{\mathbb{N}}$, $\varnothing \oplus R = \varnothing \oplus S = \varnothing$.

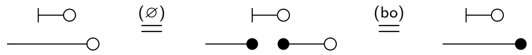
Composing or taking the monoidal product of $\varnothing$ with any relation results in $\varnothing$; $\varnothing$ is thus analogous to logical false. Indeed we can use ($\varnothing$) to derive the following lemma.

Lemma 121. *For any two $c, d : k \to l$ diagrams of Rc_a, we have*



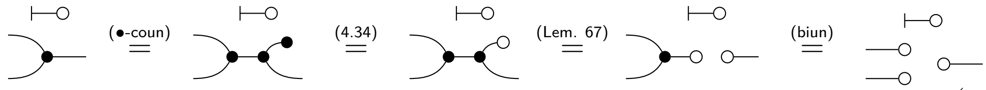
Proof. Because they represent the empty relation, all diagrams of this form should be equal in Rc_a. The proof relies on the ability to completely disconnect all diagrams using the ($\varnothing$) axiom. To verify this, we can reason by structural induction. For the base cases, we check that all generators of the same type, tensored with ⊢○, are equal.

- For the counits, we have



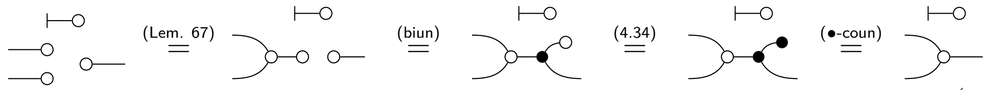
(4.34)

- For the monoids, we have



(4.35)

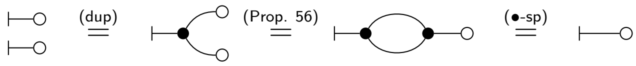
and, similarly,



(4.36)

- The reasoning for the dual generators is the same, flipped horizontally.

For the inductive case, notice that

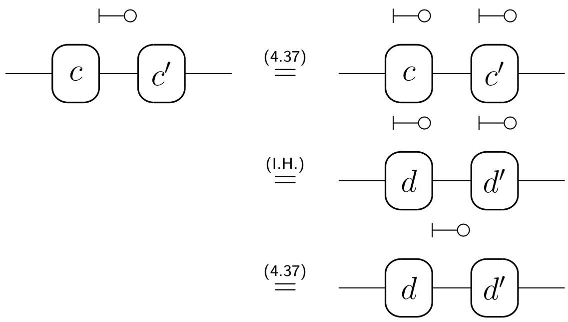


(4.37)

Then, assume that we have

(4.38)

so that their composition satisfies the expected property:



(4.39)

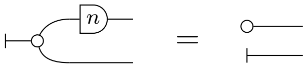
(4.40)

(4.41)

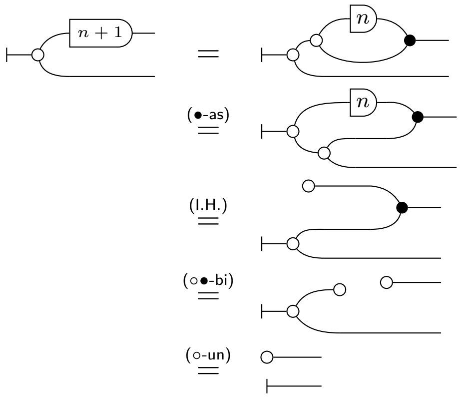
The case of the monoidal product is entirely analogous. □

The last equation, (cons), enforces the consistency of systems of non-negative integer equations. In symbolic form it guarantees that, if $2n+m=1$ then $n=0$ and $m=1$. From this simple axiom, we can prove that if $p_1n_1+\cdots+p_kn_k=1$ has a satisfying assignment iff not all the p_i, for $1\leq i\leq k$ are strictly greater than 1. The following lemmas restate this in graphical form.

Lemma 122. *For all $n > 1$,*



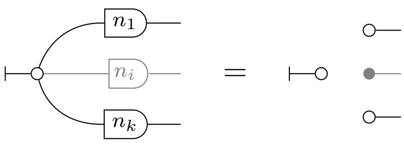
Proof. By induction on n. The base case is the axiom (cons). Assume that the statement of the lemma is true for some integer $n > 1$. Then,



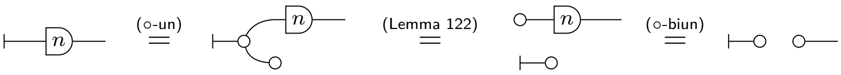
(4.42)
(4.43)
(4.44)
(4.45)
(4.46)

□

Lemma 123. *Let $n_1, \ldots, n_k$ be integers such that $n_i > 1$, for $1 \leq i \leq k$. Then*

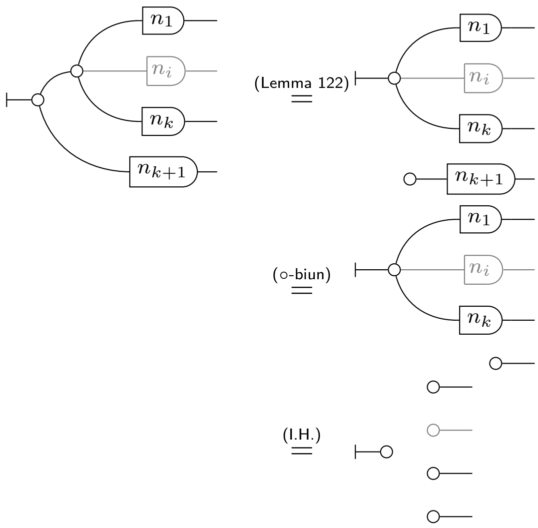


Proof. By induction on k. For $k = 1$ it is a direct consequence of Lemma 122:



(4.47)

Assume that the statement of the lemma holds for some k. Then,



(4.48)
(4.49)
(4.50)

□

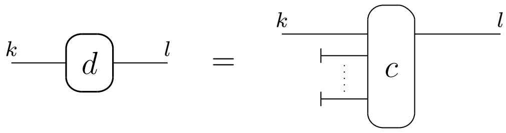
We extend $[\![-]\!] : \mathsf{Rc} \to \mathsf{AddRel}$ to $[\![-]\!]_a : \mathsf{Rc}_a \to \mathsf{PolyRel}$ by letting

$$[\![\vdash\!\!-]\!]_a = \{(\bullet, 1)\}. \tag{4.51}$$

Notice that $[\![-]\!]_a$, is a prop morphism: from the preceding discussion it is clear that the equations of Rc_a are sound for polyhedral relations. Moreover, like Rc for additive relation, Rc_a is complete for polyhedral relations.

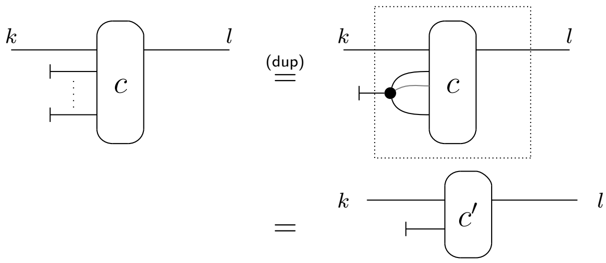
Theorem 124. $[\![-]\!]_a : \mathsf{Rc}_a \to \mathsf{PolyRel}$ *is an isomorphism of props.*

Proof. The functor $[\![-]\!]_a : \mathsf{Rc}_a \to \mathsf{PolyRel}$ is full by the representation of equation (4.32). For faithfulness we will use a normal form argument, building on the normal form of additive relations. Let $d : k \to l$ be a diagram in Rc_a. By the naturality of the symmetry we may write d as follows:



(4.52)

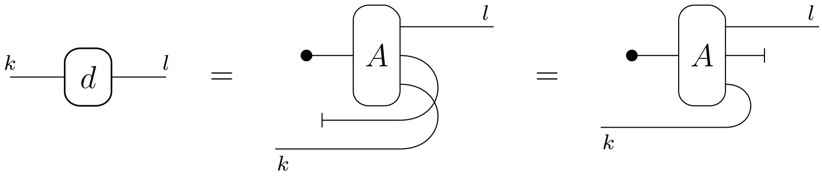
for some diagram c, in the image of the embedding $\mathsf{Rc} \hookrightarrow \mathsf{Rc}_a$. In graphical terms, we have pulled the $\vdash\!\!-$ generators up and down, past the rest of the diagram which represents some additive relation c. We may now simplify it:



(4.53)

(4.54)

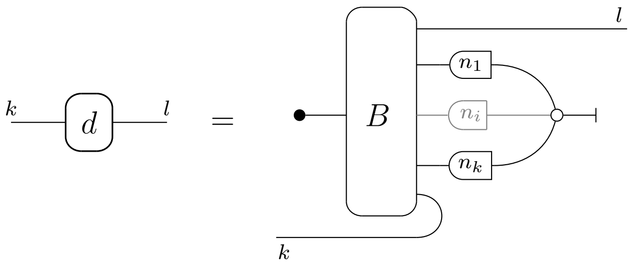
where c' is the diagram enclosed in the dotted box. Finally, from the normal form for additive relations (displayed in (3.39)), we can find a matrix diagram A such that



(4.55)

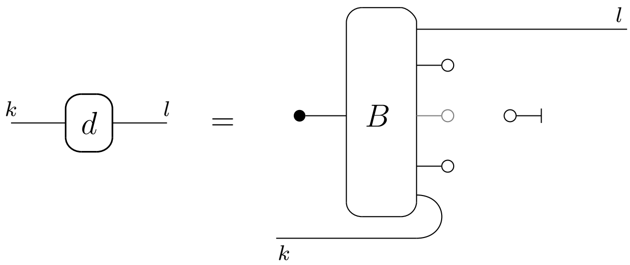
and where the columns of A are the Hilbert basis of the additive relation $[\![c']\!]$. Here, $-\!\!\dashv$ is the transpose of $\vdash\!\!-$. By Corollary 117 and the completeness of Rc for additive relations, the polyhedron $[\![d]\!]_a$ is uniquely characterised by this decomposition if it is nonempty.

If it is empty however, this means that the row of A to which —⊣ is connected is $\begin{pmatrix} n_1 & n_2 & \dots & n_k \end{pmatrix}$ with either all $n_i > 1$ or $n_i = 0$, for $1 \leq i \leq k$. In the first case, there exists a matrix B such that



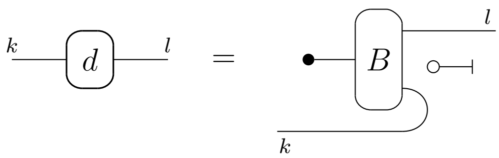
(4.56)

and therefore, by Lemma 123,



(4.57)

In the second case, A is disconnected from —⊣. We can deduce that d is of the following form, for some matrix B:



(4.58)

By Lemma 121 we know that all of these are equal in Rc_a, which concludes the proof. □

Remark 125. The same technique can be applied to interacting Hopf algebras to axiomatise affine relations (affine subspaces of $\mathbb{K}^k \times \mathbb{K}^l$ for a field $\mathbb{K}$). A field has additive and multiplicative inverses so the axiom (cons) is not sound, but the other three (dup, del, ∅) are sufficient to obtain a complete calculus. The proof is in fact simpler because the equation $a_1x_1 + \cdots + a_kx_k = 1$ always has a solution in a field, for $a_1, \dots, a_k$ not all zero. Thus, the only case of interest in the proof above for the linear case is that of (4.58).

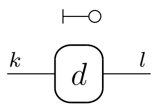
The affine resource calculus is remarkably expressive: finding whether there exists a single element of the relation corresponding to a diagram (a *satisfying assignment*) is NP-hard [Kar72], as a function of the number of open ports. Rewriting a diagram in normal form is of course more complex but we do not yet know of an upper bound for the number of steps.

The presence of $\vdash\!\circ$ and its associated axiom provides a purely graphical counterpart to an integer form of the Farkas lemma in linear programming. It is stated below in its logical form which can be shown equivalent to other more geometric formulations.

Theorem 126 (Farkas lemma)**.** *Whenever a linear system $A\mathbf{x} \leq \mathbf{b}$ is inconsistent, in the sense that no $\mathbf{x} \in \mathbb{R}^d$ satisfies it, we can derive the absurd statement $1 \leq 0$ from it by taking sums of rows of A.*

Proof. There exist many proofs of this statement and of equivalent theorems. This form of the Farkas lemma is particularly well-suited to a proof using Fourier-Motzkin elimination, the analog of Gaussian elimination for systems of linear inequalities. See [Nie13] for an elementary introduction. □

Theorem 127. *If a diagram in Rc_a represents the empty relation, it is equal to one of the form*

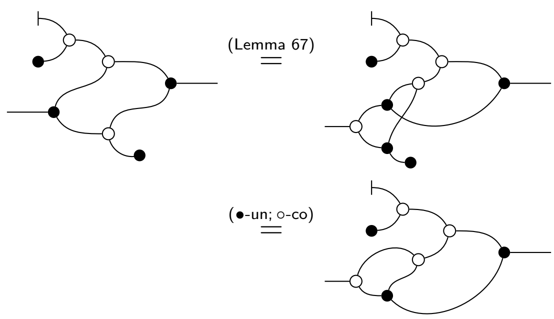


for a diagram d.

Proof. This is an immediate consequence of the completeness Theorem 124. □

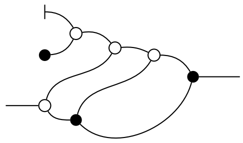
This means that, for an empty polyhedron, using solely the axioms of Rc_a we can derive the inconsistency $\vdash\!\circ$, analogous to $1 \leq 0$ in the statement of the Farkas lemma. Let us see how this works on a simple example that is not obviously inconsistent at first sight.

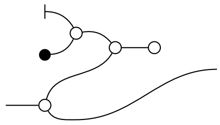
Example 128.

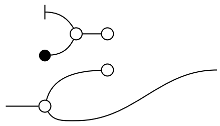


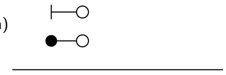
(4.59)

(4.60)

$\overset{(\circ\text{-as})}{=}$  (4.61)

$\overset{(\circ\text{-un; can})}{=}$  (4.62)

$\overset{(\circ\text{-bicoun})}{=}$  (4.63)

$\overset{(\circ\text{-biun; }\circ\text{-un})}{=}$  (4.64)

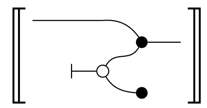
Remark 129. We can think of polyhedral relations as nondeterministic sums of additive relations. From this point of view, the affine constant endows the resource calculus with a form of nondeterministic choice. In categorical quantum mechanics, mixed states (convex sums of pure quantum states) can be encoded using a technique called the CPM construction [Sel07]. This construction has been axiomatised in terms of *environment structures* [Coe08, CP10]. They can be understood as adding a constant that represents the process of discarding parts of a system. While no compatibility with a copying process is required (since there is no canonical Frobenius structure for quantum systems), the effect is similar: the additional constant introduces nondeterminism. The precise nature of this relationship remains to be investigated.

4.4 Bounded connectors

In [BLM06], the authors describe a coordination language [GC92] called the *Calculus of Stateless Connectors*. Their aim is to separate local computation in distributed systems from the interactions between different sub-systems. Stateless connectors in this context are simple components that provide a glue to build distributed communicating systems according to a specification, in the same spirit as practical languages like Reo [BSAR06] or CommUnity [FM97]. They are generated by a small set of primitives for synchronisation, mutual exclusion, nondeterministic choice and hiding. The calculus of stateless connectors admits an intuitive operational interpretation and a fully abstract denotational semantics as the smc of relations over the two-element set, i.e., subsets of $2^k \times 2^l$, that contain $(\mathbf{0}, \mathbf{0})$, which the authors call *tick-tables*. Like the resource calculus, stateless connectors admit a graphical syntax and the authors provide a sound and complete equational theory to reason about their behaviour.

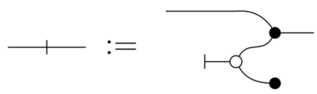
Stateless connectors can be faithfully encoded in PolyRel: there is a faithful functor between the two props. In fact, this is almost immediate because any finite subset of $\mathbb{N}^k \times \mathbb{N}^l$ is a polyhedral relation so, in particular, subsets of $2^k \times 2^l \subseteq \mathbb{N}^k \times \mathbb{N}^l$ are polyhedral. Nonetheless it is useful to write down the translation explicitly in order to illustrate how the same connectors can be represented in the resource calculus and, ultimately, how the resource calculus could be used as the foundation for a similar coordination language. It is also important to note that the equational theory of [BLM06] has several complicated axioms including an axiom scheme with no clear interpretation. While our axiomatisation also contains an axiom scheme, it is arguably much more natural.

Intuitively, the translation relies on the possibility of bounding the resources that wires transmit, with the affine part of PolyRel. In the calculus of stateless connectors of [BLM06], these resources are not modelled as natural numbers, but rather as a binary synchronisation *signal*, that can be either on or off (0 or 1). Bounding the resources available is achieved with the following relation:



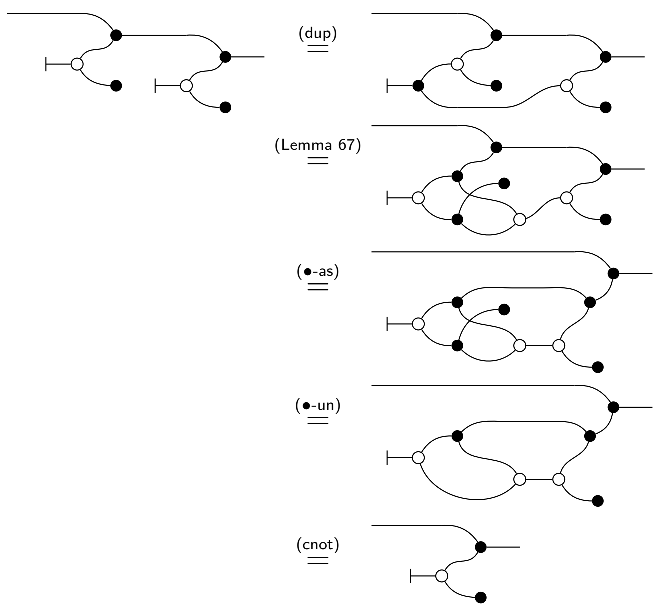
$= \{(0,0),(1,1)\}$ (4.65)

To avoid carrying this diagram around explicitly when it is not needed, we can introduce *ticked wires* as syntactic sugar for 1-bounded wires:



(4.66)

This notation is justified by the idempotence of the 1-bounded wires:



(4.67)

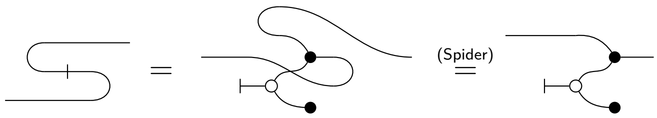
(4.68)

(4.69)

(4.70)

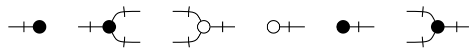
(4.71)

The use of a horizontally symmetric notation is further justified by the fact that ⊣⊢ is self-transpose:



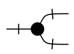
$$(4.72)$$

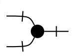
The stateless connectors are then precisely the 1-bounded counterpart of those of Rc,

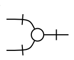


$$(4.73)$$

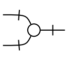
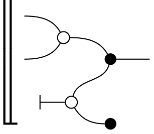
with semantics given below:

$\llbracket$  $\rrbracket = \left\{\left(0, \begin{pmatrix}0\\0\end{pmatrix}\right), \left(1, \begin{pmatrix}1\\1\end{pmatrix}\right)\right\}$ $\quad$ $\llbracket$  $\rrbracket = \{(0, \bullet), (1, \bullet)\}$ (4.74)

$\llbracket$  $\rrbracket = \left\{\left(\begin{pmatrix}0\\0\end{pmatrix}, 0\right), \left(\begin{pmatrix}1\\1\end{pmatrix}, 1\right)\right\}$ $\quad$ $\llbracket$  $\rrbracket = \{(\bullet, 0), (\bullet, 1)\}$ (4.75)

$\llbracket$  $\rrbracket = \left\{\left(\begin{pmatrix}0\\0\end{pmatrix}, 0\right), \left(\begin{pmatrix}1\\0\end{pmatrix}, 1\right), \left(\begin{pmatrix}0\\1\end{pmatrix}, 1\right)\right\}$ $\quad$ $\llbracket$  $\rrbracket = \{(\bullet, 0)\}$ (4.76)

The $\bullet$-structure is the usual Frobenius structure restricted to 0 and 1. The $\circ$-structure is more interesting. It encodes *mutual exclusion*:

$\llbracket$  $\rrbracket = \llbracket$  $\rrbracket = \left\{\left(\begin{pmatrix}n\\m\end{pmatrix}, n+m\right) \mid n+m \leq 1\right\}$ (4.77)

The port on the right is activated precisely when at most one of the ports on the left is activated. In other words, the two ports on the left cannot be activated at the same time; they exclude each other. Note that the ticked $\circ$ and $\bullet$-structures do not interact through the bimonoid law any more since, for example, ($\bullet\circ$-biun) does not hold:

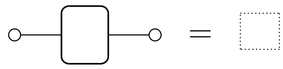
$$(4.78)$$

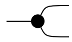
Since the calculus of stateless connectors is the sub-prop of Rel_2 generated by the morphisms above, we immediately get a faithful encoding. This is a consequence of the completeness of Rc_a and the obvious embedding $\mathsf{Rel}_2 \hookrightarrow \mathsf{Rel}_\mathbb{N}$ induced by the set inclusion $2 \hookrightarrow \mathbb{N}$. Note that the embedding $\mathsf{Rel}_2 \hookrightarrow \mathsf{Rel}_\mathbb{N}$ is not a functor since it does not preserve identities. It is however, a functor up to an idempotent in a sense that Definition 130 and the discussion following it will make precise.

Definition 130. The *Karoubi envelope* (or idempotent completion) of a category C is the category $\hat{\mathsf{C}}$ with objects pairs $(A, e\colon A \to A)$ of an object and an idempotent of C. Its morphisms $(A, e) \to (B, f)$ are morphisms $\varphi\colon A \to B$ of C such that $\varphi = e \,;\, \varphi \,;\, f$. Composition is inherited from C and the identity on (A, e) is e.

The envelope $\hat{\mathsf{C}}$ contains C as the full subcategory spanned by objects of the form (A, id_A). When C is a smc, it is not difficult to see that its Karoubi envelope is also a smc.

We can see that stateless connectors are morphisms of $\widehat{\mathsf{PolyRel}}$ between monoidal products of $(1, -\!\!+\!\!-)$. An important property of stateless connectors is that they can always stay idle, not synchronising with their environment. This is witnessed by the fact that the associated relation always contains $(\mathbf{0}, \mathbf{0})$. Therefore, stateless connectors are precisely those morphisms verifying

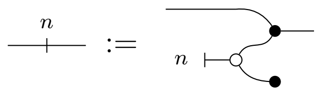
 (4.79)

Note that there is some redundancy in the translation because, for example, simply adding one tick to the left port of  (that is, pre-composing with the 1-bounded wire) is enough to obtain the desired relation. However it is instructive to see the stateless connectors as those of Rc up to an idempotent. In fact, following this train of thought, there is no reason to limit ourselves to 1-bounded wires. We can generalise them to model, not only connectors that are n-bounded for a fixed n, but also the interaction between connectors with access to resources with different bounds.

To each natural number n we can associate an idempotent as we did for 1. Let

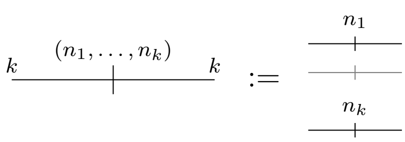
(4.80)

and



(4.81)

Similarly, to any list of natural numbers $n_1, \ldots, n_k$ we associate the idempotent



(4.82)

We restrict our attention to the sub-smc of $\widehat{\mathsf{PolyRel}}$ with objects $(k, (n_1, \ldots, n_k))$, where $(n_1, \ldots, n_k)$ denotes the associated idempotent, and morphisms $(k, (n_1, \ldots, n_k)) \to (l, (m_1, \ldots, n_l))$ those verifying equation (4.79). We can think of them as relations with access to bounded resources specified by the two bounds $(n_1, \ldots, n_k)$ and $(m_1, \ldots, m_l)$.

Finally, if we drop the requirement of equation (4.79), we can express *all* relations between finite sets. Let $\mathsf{fRel}_\times$ be the full sub-smc of $\mathsf{Rel}_\times$ spanned by finite sets (note that this is not a prop).

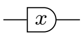
Theorem 131. *The full sub-smc* Bound *of* $\widehat{\mathsf{PolyRel}}$ *spanned by objects* $(k, (n_1, \ldots, n_k))$ *is monoidally equivalent to* $\mathsf{fRel}_\times$*.*

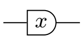
Proof. Clearly, morphisms $(k, (n_1, \ldots, n_k)) \to (l, (m_1, \ldots, n_l))$ in Bound are in bijective correspondence with relations $\underline{n_1} \times \cdots \times \underline{n_k} \to \underline{m_1} \times \cdots \times \underline{m_l}$. Let us describe the equivalence explicitly: let $K : \mathsf{Bound} \to \mathsf{fRel}$ be given by $K(k, (n_1, \ldots, n_k)) = \underline{n_1} \times \cdots \times \underline{n_k}$ and $K(P) = P$. This is functorial because morphisms of $\widehat{\mathsf{PolyRel}}$ inherit composition from $\mathsf{PolyRel}$ which itself inherits it from Rel. Finally, this is an equivalence because every finite set of cardinality n is isomorphic to the finite ordinal $\underline{n}$, by fixing a total order. □

As a result, we obtain a sound and complete graphical calculus for $\mathsf{fRel}_\times$, from the graphical calculus Rc_a for polyhedral relations. The reader may find it strange that this calculus exploits a total order on each set $\underline{k}$ to encode arbitrary relations. We can always do this in the same way that we can choose a basis on a finite-dimensional vector space and represent arbitrary linear transformations as matrices.

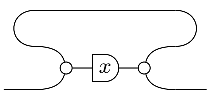
Chapter 5

Towards stateful concurrency

The resource calculus (with its affine extension) provides a convincing coordination language for stateless concurrency. However, without state, the synchronisation patterns that it can express is limited. This is why we would like to extend it to capture the behaviour of stateful systems. To this effect, we will expand our syntax with an additional generator , representing a state-holding synchronous register. Its operational behaviour is that of a simple one-step buffer: at any given moment if it holds value n it releases to the environment on the right, and stores whatever value m it synchronised with on the left.

This new piece of syntax will force us to modify our semantics slightly. We will need to introduce relations with extra state passing variable that represent the register's change of value. These can be thought of as labelled transition systems with pairs of labels on the left and on the right for each transition. We show that this corresponds to an abstract functorial construction $\mathsf{St}(-)$ that can be applied to any prop. It turns out that adding a single $1 \to 1$ generator to any prop T with sufficient structure produces a prop isomorphic to $\mathsf{St}(\mathsf{T})$. In particular, the resource calculus extended with the register  is isomorphic to $\mathsf{St}(\mathsf{AddRel})$, a prop in which morphisms are *additive labelled transition systems*.

As a case study, we show that the transition systems of (open) Petri nets can all be expressed in the stateful resource calculus. This encoding relies on the crucial realisation that the behaviour of a place in a Petri net is precisely that of the following diagram:



(5.1)

Indeed, assume some value $s \in \mathbb{N}$ is currently in the register. Regardless of what value $n \in \mathbb{N}$ arrives on the left, in order to observe $m \in \mathbb{N}$ on the right one just needs to find n' such that $m + n' = s$; this is only possible if $m \leq s$. This observation finally connects the work of this thesis with one of its original motivations, as promised in Section 3.1.

Note that the stateful resource calculus is strictly more expressive than (open) Petri nets. The latter are fundamentally asynchronous in the sense that they can always stay idle and arbitrarily delay the firing of transitions. Only the causal ordering of transition matters. The stateful resource calculus, however, is synchronous: the register forces the synchronisation of its left and right ports with a delay of precisely one timestep. Through the feedback loop of the diagram above (5.1), we can encode the asynchronous place into our synchronous calculus, but this is not the only possible use of our results. We envisage the resource calculus as an assembly language for concurrency into which different formalisms can be compiled and compared. The possibility of embedding Petri nets is here to validate our approach but is not the fullest application of the calculus, even if it is the one of the only concrete case studies of this thesis.

Remark 132. The results of Sections 5.1 and 5.2 have been published in [BHP$^+$19], co-authored with Filippo Bonchi, Josh Holland, Paweł Sobociński and Fabio Zanasi. As for Chapter 3, the author of this thesis is greatly indebted to his collaborators for the presentation of the results of this chapter. In particular, a lot of the results of Sections 5.1 and 5.2 that also appear in [BHP$^+$19] were formalised and written jointly with Filippo Bonchi.

5.1 An axiomatic approach to stateful systems

We would like to extend the stateless calculi Rc and Rc_a to include systems with internal state. First, we adopt a more abstract perspective to demonstrate that:

(a) from any prop T one may obtain a prop $\mathsf{St}(\mathsf{T})$ in which morphisms are stateful systems;

(b) in the presence of a compact closed structure on T, moving to $\mathsf{St}(\mathsf{T})$ amounts to extending T with a single generator $-\boxed{x}-$ with no equations.

From this, it follows that the stateful resource calculus and its affine extension are sound and fully complete axiomatisations of the props $\mathsf{St}(\mathsf{AddRel})$ and $\mathsf{St}(\mathsf{PolyRel})$, respectively.

5.1.1 Adding internal state

From a given prop representing stateless systems, the construction $\mathsf{St}(-)$ in Definition 133 below results in a prop in which the morphisms are stateful systems. This way of adding state is a natural and well-known technique that appears in several places in the literature, for example, in the setting of Cartesian bicategories [KSW97a] and the Geometry of Interaction [HMH14].

Definition 133. Let T be a prop. Define $\mathsf{St}(\mathsf{T})$ as the prop where:

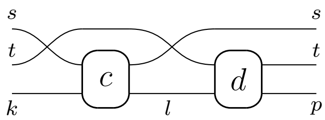
- morphisms $k \to l$ are pairs (s, c) where $s \in \mathbb{N}$ and $c\colon s+k \to s+l$ is a morphism of T, quotiented by the smallest equivalence relation including every instance of

  (5.2)

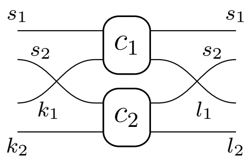
  for permutations $\sigma : s \to s$;

- the composition of $(s, c)\colon k \to l$ and $(t, d)\colon l \to p$ is $(s + t, e)$ where e is the arrow of T given by

  (5.3)

- the monoidal product of $(s_1, c_1)\colon k_1 \to l_1$ and $(s_2, c_2)\colon k_2 \to l_2$ is $(s_1 + s_2, e)$ where e is given by

  (5.4)

- the identity on j is $(0, id_j)$ and the symmetry of k, l is $(0, \sigma_{k,l})$.

Composition and the monoidal product are strictly associative because those of T are. Moreover, they are both immediately seen to be congruences for the equivalence relation of (5.17).

Example 134. Thinking of ordinary functions between sets as stateless deterministic transducers, the morphisms of $\mathsf{St}(\mathsf{Set})$ are functions whose output does not just depend on their input, but also on a set of internal states that is updated at every application. In other words, a stateful function $A \to B$ is a transducer $f\colon S \times A \to S \times B$.

In the setting of sets and *relations*, stateful morphisms are (2-)labelled transition systems (lts) with labels in A and B. For additive relations, $\mathsf{St}(\mathsf{AddRel})$ is a prop of additive lts.

The equivalence relation of Definition 133 ensures that internal state remains anonymous. The set of stateful wires is hidden and ought to serve only as a reference for internal state. We do not want to keep track of the states' labels but only of the states themselves. This is why we equate transducers that only differ by a bijective relabelling of their set of states. We can think of this as a syntactic form of equivalence akin to α-equivalence.

Remark 135. The reader familiar with (co)ends might have noticed that the $\mathsf{St}(-)$ construction can be expressed concisely as the following coend:

$$\mathsf{St}(\mathsf{T}) \cong \int^{s:\mathsf{P}} \mathsf{T}(\iota(s)+k, \iota(s)+l) \tag{5.5}$$

where P is the prop of permutations and $\iota\colon \mathsf{P} \to \mathsf{T}$ the canonical inclusion of permutations in T (c.f. Remark 27). Unrolling the definition, we see that

$$\int^{s:\mathsf{P}} \mathsf{T}(\iota(s)+k, \iota(s)+l) = \sum_{\sigma\colon s\to s} \mathsf{T}(s+k, s+l)/\sim \tag{5.6}$$

for which the equivalence relation is precisely the one given by (5.17).

There exists an obvious embedding $\mathsf{T} \hookrightarrow \mathsf{St}(\mathsf{T})$ since $f\colon k \to l$ can always be seen as a stateful morphism $f\colon 0+k \to 0+l$. In more concrete terms, a stateless system can always be seen as stateful with trivial state.

Finally, the state construction is itself functorial.

Proposition 136. $\mathsf{St}(-)$ *extends to an endofunctor on* Prop.

Proof. Let $F\colon \mathsf{T} \to \mathsf{T}'$ be a prop morphism. Define $\mathsf{St}(F)\colon \mathsf{St}(\mathsf{T}) \to \mathsf{St}(\mathsf{T}')$ by $\mathsf{St}(F)(k) = F(k)$ on objects and $\mathsf{St}(F)(c) = F(d)$ for a morphism $d\colon s+k \to s+l$ of T in the equivalence class c. This is well-defined because, if $d \sim d'$, there exists a permutation $\sigma\colon s \to s$ such that $d = (\sigma \oplus 1_k) \,;\, d' \,;\, (\sigma^{-1} \oplus 1_l)$, then $F(d) = (\sigma \oplus 1_k) \,;\, F(d') \,;\, (\sigma^{-1} \oplus 1_l)$ (since F is a prop morphism, it is strict monoidal).

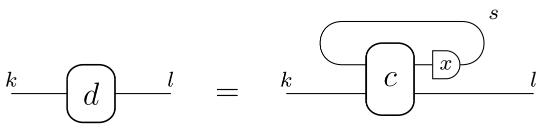
$\mathsf{St}(-)$ is immediately seen to be functorial from its definition. ∎

5.1.2 Presenting $\mathsf{St}(-)$

Let X be the prop freely generated by a signature with a single generator, $-\boxed{x}-$, and no equations. Given a prop T, one can form the coproduct $\mathsf{T}+\mathsf{X}$ of T and X (cf. Section 2.2.2). Intuitively, the prop $\mathsf{T}+\mathsf{X}$ is simple to describe: it arises by freely pasting together sequentially and in parallel the morphisms of T with the single generator $-\boxed{x}-$ of X.

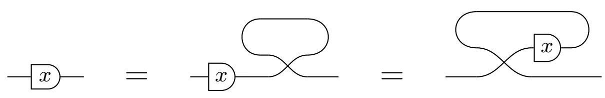
In Theorem 138 below, we give a simple characterisation of $\mathsf{St}(\mathsf{T})$: it is *isomorphic* to $\mathsf{T}+\mathsf{X}$, assuming that T has compact closed structure[1]. This result relies on the simple technical lemma below, which is a useful characterisation of the morphisms of $\mathsf{T}+\mathsf{X}$.

Lemma 137 (Trace canonical form)**.** *Suppose that* T *is a compact closed prop. For every* $d : k \to l$ *in* $\mathsf{T}+\mathsf{X}$ *there exists a morphism* $c : s+k \to s+l$ *of* T *such that*



(5.7)

Proof. By structural induction on morphisms of $\mathsf{T}+\mathsf{X}$. For the base case, if a morphism d of $\mathsf{T}+\mathsf{X}$ is in either T or X, the statement holds since

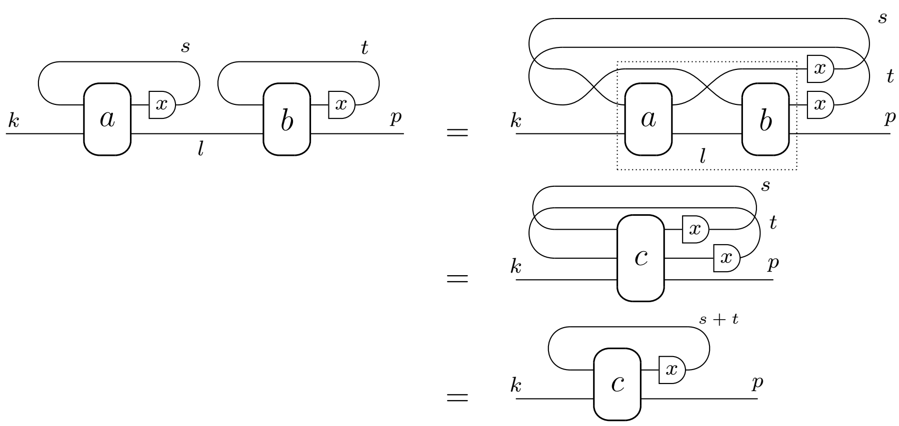


(5.8)

and every morphism of T is trivially in trace canonical form with the trace taken over the 0 object.

There are two inductive cases to consider:

- d is given by the sequential composition of two morphisms $a\colon k \to l$ and $b\colon l \to p$ in trace canonical form:



(5.9)

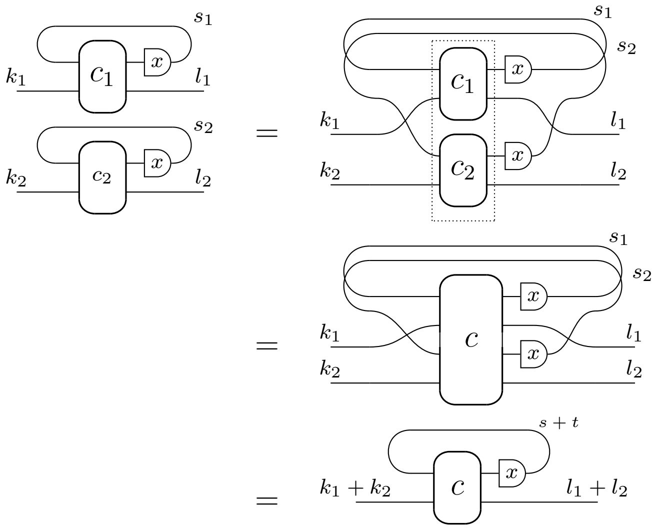
(5.10)

(5.11)

- d is given as the monoidal product of two morphisms $c_1\colon k_1 \to l_1$ and $c_2\colon k_2 \to l_2$,

[1]Here, we can assume that T is compact closed in the general sense (Remark 14) and not necessarily self-dual compact closed. In fact the theorem holds in the more general setting of traced symmetric monoidal categories (Definition 157) as the proof only makes use of the properties of the trace.

both in trace canonical form:



(5.12)

(5.13)

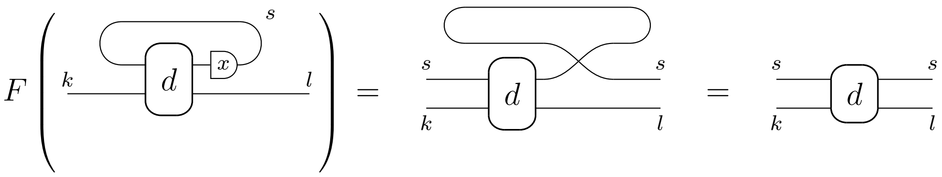
(5.14)

□

To exhibit the isomorphism $\mathsf{St}(\mathsf{T}) \cong \mathsf{T}+\mathsf{X}$, we define two monoidal functors, $R\colon \mathsf{X} \to \mathsf{St}(\mathsf{T})$ and $Z\colon \mathsf{T} \to \mathsf{St}(\mathsf{T})$. For X it suffices to say where its single generator is mapped: set $R(-\boxed{x}-) = (1, \times)$. The second functor $Z\colon \mathsf{T} \to \mathsf{St}(\mathsf{T})$ is defined as $Z(t) = (0, t)$ for all arrows t of T. Let

$$F := \langle Z, R\rangle : \mathsf{T}+\mathsf{X} \to \mathsf{St}(\mathsf{T})$$

be the prop morphism obtained from the universal property of the coproduct. Intuitively, we can think of F as an operation on diagrams that cuts the $-\boxed{x}-$ out and pulls the wires to which it was connected into state-passing wires. Indeed, since $-\boxed{x}-$ is mapped to $\times$, we have, for a morphism of $\mathsf{T}+\mathsf{X}$ in trace canonical form:

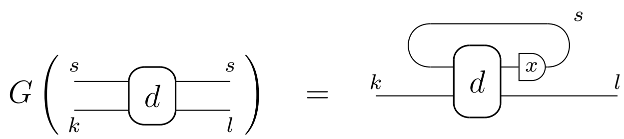


(5.15)

If we interpret the operations of $\mathsf{T}+\mathsf{X}$ as processes, the canonical interpretation of $-\boxed{x}-$ is that of a register that holds its state until it is set to a new value, at which point it releases the content of its memory to the rest of the system. This is precisely the meaning of $\times$.

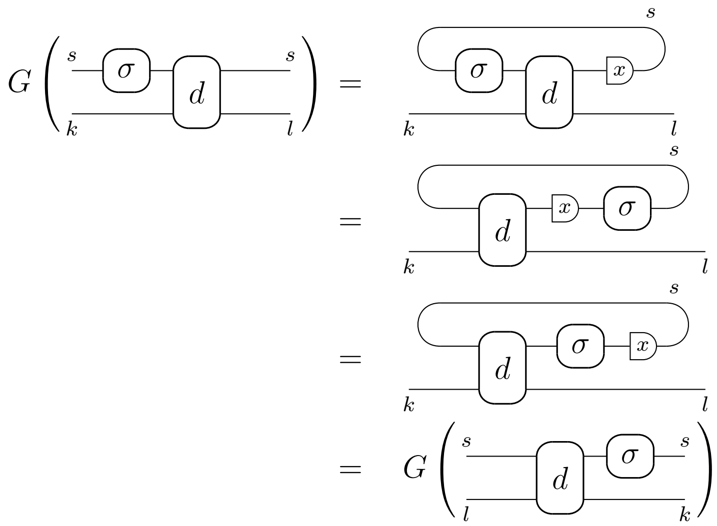
Theorem 138. *If T is a compact closed prop then $F := \langle Z, R\rangle$ is an isomorphism.*

Proof. We construct the inverse of F explicitly. Let $G\colon \mathsf{St}(\mathsf{T}) \to \mathsf{T}+\mathsf{X}$ be defined by



(5.16)

Thus, while F cuts out the $-\!\boxed{x}\!-$, G takes a stateful d and feeds back the state guarded by $-\!\boxed{x}\!-$. This mapping is well defined because, for $\sigma : s \to s$ a permutation, we have



(5.17)

(5.18)

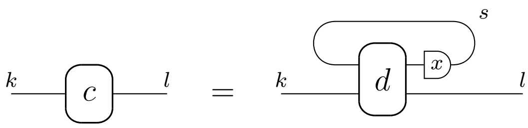
(5.19)

(5.20)

The penultimate equality is only valid because σ is a permutation and, by naturality of the symmetry in a smc, it commutes with the s parallel copies of $-\!\boxed{x}\!-$.

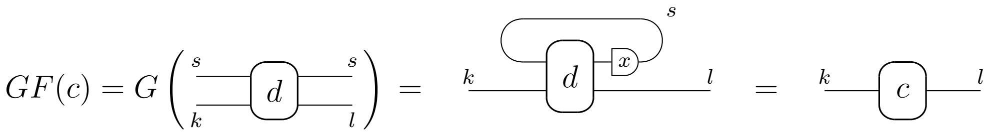
Monoidal functoriality follows from the compact structure; the argument is similar to the proof of Lemma 137.

We now prove that F and G are inverse. Given d in $\mathsf{St}(\mathsf{T})$, the fact that $FG(d) = d$ is immediate by (5.15). Conversely, given c in $\mathsf{T}+\mathsf{X}$, we can use the conclusion of Lemma 137 to obtain d in T such that



(5.21)

Thus, we have



(5.22)

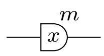
□

We can now extend the resource calculus with a register to define the *stateful resource calculus*, $\mathsf{Rc}_s := \mathsf{Rc} + X$. Its morphisms admit an operational semantics in terms of additive lts via $[\![-]\!]_s \colon \mathsf{Rc}_s \to \mathsf{St}(\mathsf{AddRel})$ given by the composite

$$\mathsf{Rc}_s \xrightarrow{\langle Z,R\rangle} \mathsf{St}(\mathsf{Rc}) \xrightarrow{\mathsf{St}([\![-]\!])} \mathsf{St}(\mathsf{AddRel}) \tag{5.23}$$

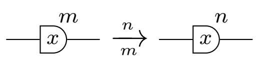
The stateful resource calculus constitutes a basic assembly language for distributed systems. To showcase its expressiveness we will demonstrate how to embed Petri nets into it in Section 5.2.

Remark 139. What we mean by "operational semantics". There is a long tradition, starting with the work of Plotkin [Plo04], of defining the behaviour of a program as a function of the behaviour of its parts. Structural operational semantics defines behaviour inductively, as the small-step evolution of each piece of syntax, given by a set of inference rules on terms. This specifies how a machine should execute a program, like β-reduction for the λ-calculus. For Rc_s, we do not specify how to execute diagrams but instead we translate them into labelled transition systems, in $\mathsf{St}(\mathsf{AddRel})$. We use the expression "operational semantics" in contrast to less intensional notions of process equivalence, like trace equivalence or bisimilarity. We could translate between our approach and a more traditional view of structural operational semantics by first introducing a new piece of syntax. For every $m \in \mathbb{N}$, let



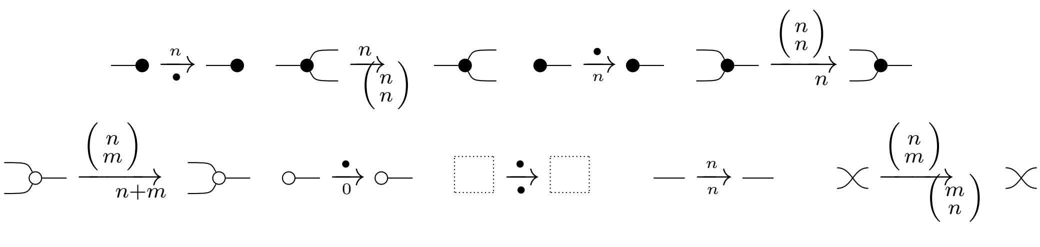
(5.24)

be the register holding value m, with operational semantics given by the following structural rule:



(5.25)

The operational meaning of the other terms of our syntax are given recursively by



(5.26)

with inference rules for sequential and parallel compositions displayed below:

$$\frac{s \xrightarrow[\mathbf{b}]{\mathbf{a}} s' \quad t \xrightarrow[\mathbf{c}]{\mathbf{b}} t'}{s \,;\, t \xrightarrow[\mathbf{c}]{\mathbf{a}} s' \,;\, t'} \qquad \frac{s \xrightarrow[\mathbf{b}_1]{\mathbf{a}_1} s' \quad t \xrightarrow[\mathbf{b}_2]{\mathbf{a}_2} t'}{s \oplus t \xrightarrow[\binom{\mathbf{b}_1}{\mathbf{b}_2}]{\binom{\mathbf{a}_1}{\mathbf{a}_2}} s' \oplus t'} \tag{5.27}$$

where m, n range over the natural numbers $\mathbb{N}$ and $\mathbf{a}, \mathbf{b}, \mathbf{c}$ over natural number vectors. Note that this operational semantics is not meant to be executable directly, as relational composition contains unbounded nondeterminism.

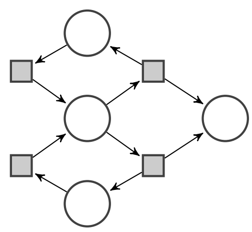
Operational semantics like ours are closely related to the *tile model* [GM00]. In fact, we could also arrange $\mathsf{St}(\mathsf{AddRel})$ as a symmetric monoidal *double* category, which would correspond more closely to the tile calculus developed in [SMMB13]. We leave the precise correspondence between the two for future work.

5.2 Petri nets

Definition 140. 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 $\mathbf{a}, \mathbf{a}' \in \mathbb{N}^P$, we write $\mathbf{a} \to \mathbf{a}'$ if there exists $\mathbf{t} \in \mathbb{N}^T$ such that ${}^\circ\mathbf{t} \leq \mathbf{a}$ and $\mathbf{a}' = \mathbf{a} - {}^\circ\mathbf{t} + \mathbf{t}^\circ$. The operational semantics (also known as the firing semantics) of $\mathcal{P}$ is the relation

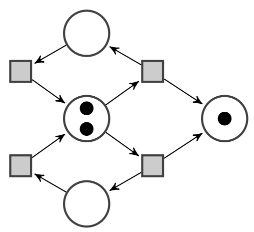
$$\mathsf{Fire}\,(\mathcal{P}) = \{(\mathbf{a}, \mathbf{a}') \mid \mathbf{a} \to \mathbf{a}'\} \subseteq \mathbb{N}^P \times \mathbb{N}^P. \tag{5.28}$$

Example 141. Consider the Petri net displayed below with the usual graphical notation, as bipartite graphs with circles representing places and squares representing transitions:

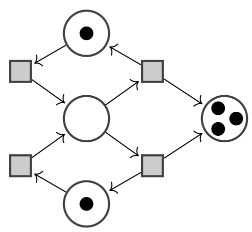


The multisets in the firing semantics are represented by tokens in the places. A given state is also called a *marking*. Transitions move tokens from one place to another. For example, from the state on the left below, the two rightmost transitions are enabled and can fire to give:

(5.29)



$\longrightarrow$



The following is an important observation.

Proposition 142. *For a Petri net* $\mathcal{P} = (P, T, {}^\circ-, -^\circ)$, $\mathsf{Fire}(\mathcal{P})$ *is an additive relation* $|P| \to |P|$.

Proof. This can be seen as an immediate consequence of the encoding of Petri nets into the resource calculus, Proposition 147 below. But this simple fact can be proven without referring to the resource calculus. We sketch a direct proof here. If $(\mathbf{a}_1, \mathbf{a'}_1) \in \mathsf{Fire}(\mathcal{P})$ and $(\mathbf{a}_2, \mathbf{a'}_2) \in \mathsf{Fire}(\mathcal{P})$, then there exists $\mathbf{t}_1, \mathbf{t}_2 \in \mathbb{N}^T$ such that ${}^\circ\mathbf{t}_1 \leq \mathbf{a}_1$, $\mathbf{a'}_1 = \mathbf{a}_1 - {}^\circ\mathbf{t}_1 + \mathbf{t}_1{}^\circ$ and ${}^\circ\mathbf{t}_2 \leq \mathbf{a}_2$, $\mathbf{a'}_2 = \mathbf{a}_2 - {}^\circ\mathbf{t}_2 + \mathbf{t}_2{}^\circ$. So ${}^\circ\mathbf{t}_1 + {}^\circ\mathbf{t}_2 \leq \mathbf{a}_1 + \mathbf{a}_2$ and $\mathbf{a'}_1 + \mathbf{a'}_2 = \mathbf{a}_1 + \mathbf{a}_2 - {}^\circ\mathbf{t}_1 - {}^\circ\mathbf{t}_2 + \mathbf{t}_1{}^\circ + \mathbf{t}_2{}^\circ$. Therefore, $(\mathbf{a}_1 + \mathbf{a}_2, \mathbf{a'}_1 + \mathbf{a'}_2) \in \mathsf{Fire}(\mathcal{P})$. Finally to see that $\mathsf{Fire}(\mathcal{P})$ is finitely generated, let ${}^\circ-$ and $-^\circ$ be given by the matrices A_1 and A_2 respectively. Then $\begin{pmatrix} A_1 \\ A_2 \end{pmatrix}$ is a representing matrix for $\mathsf{Fire}(\mathcal{P})$. □

Petri nets are combinatorial objects, usually studied monolithically rather than compositionally. We introduce a syntax Petri that *(i)* extends the resource calculus in a straightforward way and *(ii)* whose *closed* diagrams (arrows in $\mathsf{Petri}(0,0)$) capture precisely the firing semantics of Petri nets. Moreover, arbitrary diagrams are open nets in the style of [SMMB13] as we will show in Section 5.2.3.

The syntax is that of the resource calculus extended with one extra generator $\rightarrow\!\bigcirc\!-$: $1 \to 1$ to represent a place. Formally, its semantics reproduces the operational behaviour of Definition 140 and is given by the following rule:

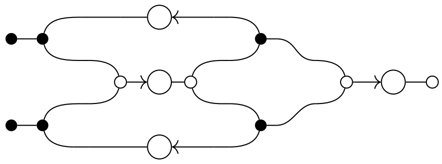
$$[\![\rightarrow\!\bigcirc\!-]\!]_p := \left\{ \left(\begin{pmatrix} p \\ m \end{pmatrix}, \begin{pmatrix} m + p - n \\ n \end{pmatrix} \right) \mid n \leq p \right\} \tag{5.30}$$

Observe that Petri is the same as the coproduct $\mathsf{Rc} + \mathsf{Pl}$ in Prop, where Pl is the prop whose arrows are string diagrams on the signature with the single $\rightarrow\!\bigcirc\!-$ generator and no equations. Then $[\![-]\!]_p$ extends to a prop morphism $\mathsf{Petri} \to \mathsf{St}(\mathsf{AddRel})$ which maps all generators of Petri inherited from Rc to their semantics in $\mathsf{St}(\mathsf{AddRel})$ through the isomorphism $\mathsf{Rc} \cong \mathsf{AddRel}$ and the embedding $\mathsf{AddRel} \hookrightarrow \mathsf{St}(\mathsf{AddRel})$. Since we impose no equations on $\rightarrow\!\bigcirc\!-$, this mapping is immediately seen to be functorial and symmetric monoidal by construction.

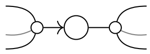
5.2.1 Encoding nets into Petri

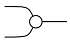
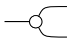
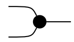
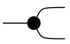
Before rigorously defining the encoding, it is more instructive to illustrate the role of Petri with an example.

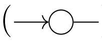
Example 143. The diagram of $\mathsf{Petri}(0,0)$ corresponding to the net of Example 141 is:



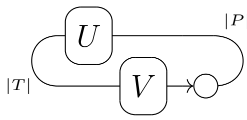
More generally, a place with multiple inputs and outputs is depicted as



using  and , while transitions are represented with the help of  and .

It is useful to remark that, contrary to the usual depiction of Petri nets as directed bipartite graphs, in our formalism it is the places—not the edges of transitions—which are directed. In Petri, the transitions are simply additive relations and, as such, do not admit a consistent directionality. The addition of the places is what directs the flow of tokens, since they have a well-defined notion of inputs and outputs. This is why we choose to depict them with an arrow () in order to highlight this distinction.

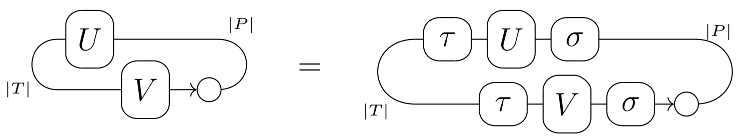
Any ordinary Petri net $\mathcal{P}$ can be encoded as a diagram $d_{\mathcal{P}}$ in $\mathsf{Petri}(0,0)$. By choosing an ordering on places and transitions, the functions ${}^\circ-, -^\circ : T \to \mathbb{N}^P$ can be regarded as matrices with coefficients in $\mathbb{N}$, of type $|T| \to |P|$. Such matrices can be seen as special cases of additive relations (cf. Theorem 63, Section 3.5): let U and V be the matrix corresponding to ${}^\circ-$, $-^\circ$ respectively. We also identify them with their diagrams in Rc. Let $d_{\mathcal{P}}$ be



(5.31)

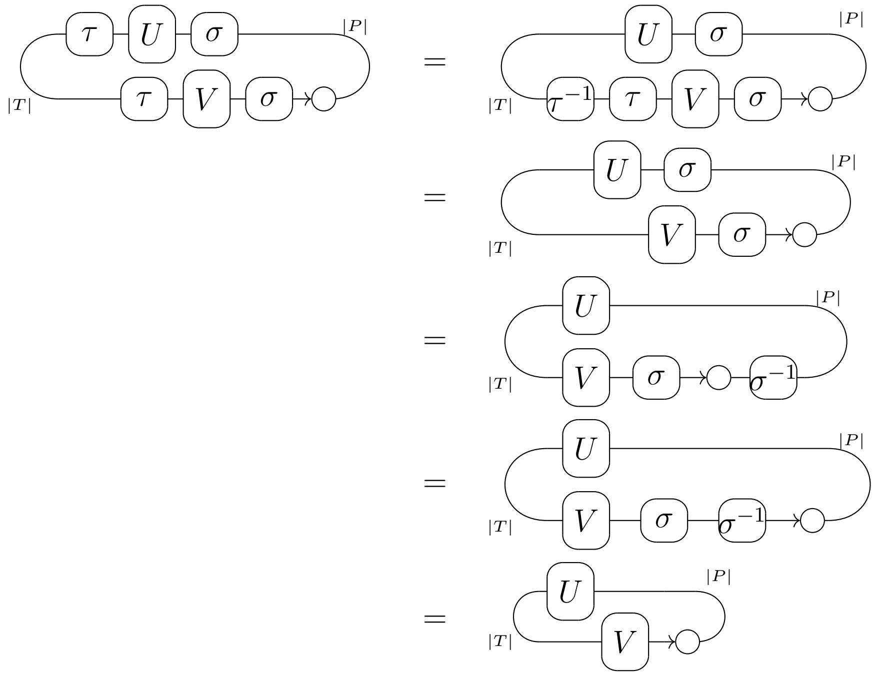
The following lemma ensures that the assignment $\mathcal{P} \mapsto d_{\mathcal{P}}$ is well defined, namely that it is independent from the chosen ordering on places and transitions.

Lemma 144. *Let $\sigma\colon |P| \to |P|$ and $\tau\colon |T| \to |T|$ be two permutations. Then*



(5.32)

Proof.



(5.33)

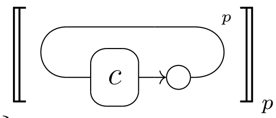
(5.34)

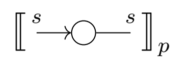
(5.35)

(5.36)

(5.37)

□

Lemma 145. *Let* $c \in \mathsf{Rc}(p,p)$. *Then*  $= \{(\mathbf{a},\mathbf{a}') \mid \exists(\mathbf{x},\mathbf{y}) \in [\![\, c \,]\!]$ *such that* $\mathbf{x} \leq \mathbf{a}$ *and* $\mathbf{a}' = \mathbf{a} - \mathbf{x} + \mathbf{y}\}$.

Proof. We can easily extend by induction the semantics of a single place, given in (5.30), to s places in parallel: $\left(\begin{pmatrix}\mathbf{a}\\ \mathbf{y}\end{pmatrix}, \begin{pmatrix}\mathbf{a}'\\ \mathbf{x}\end{pmatrix}\right) \in$  iff $\mathbf{x} \leq \mathbf{a}$ and $\mathbf{a}' = \mathbf{a} - \mathbf{x} + \mathbf{y}$. The statement of the lemma follows immediately. □

We show that $\mathcal{P}$ and $d_{\mathcal{P}}$ have the same operational behaviour.

Proposition 146. *Given a Petri net* $\mathcal{P}$, *we have* $\mathsf{Fire}(\mathcal{P}) \sim [\![\, d_{\mathcal{P}} \,]\!]_p$ *(with* $\sim$ *the equivalence relation on morphisms of* $\mathsf{St}(\mathsf{AddRel})$*).*

Proof. Let $\mathcal{P} = (P, T, {}^\circ-, -^\circ)$ be a Petri net. Since the operational equivalence is stated modulo $\sim$, we can fix a total order on P and T. Let $\mathbf{a}, \mathbf{a}' \in \mathbb{N}^{|P|}$. Denote by U and V the matrices corresponding to ${}^\circ-$ and $-^\circ$ in the definition of $d_{\mathcal{P}}$. Thus we have $(\mathbf{a},\mathbf{a}') \in \mathsf{Fire}(\mathcal{P})$ iff there exists $\mathbf{f} \in \mathbb{N}^{|T|}$ such that $U\mathbf{f} \leq \mathbf{a}$ and $\mathbf{a}' = \mathbf{a} - U\mathbf{f} + V(\mathbf{f})$. Since $U^\dagger \,;\, V = \{(\mathbf{x},\mathbf{y}) \mid \mathbf{x} = U\mathbf{f} \text{ and } \mathbf{y} = V\mathbf{f}\}$, we have that $(\mathbf{a},\mathbf{a}') \in \mathsf{Fire}(\mathcal{P})$ iff there exists $(\mathbf{x},\mathbf{y}) \in U^\dagger \,;\, V$ such that $\mathbf{x} \leq \mathbf{a}$ and $\mathbf{a}' = \mathbf{a} - \mathbf{x} + \mathbf{y}$. By Lemma 145, we conclude that $(\mathbf{a},\mathbf{a}') \in \mathsf{Fire}(\mathcal{P})$ iff $(\mathbf{a},\mathbf{a}') \in [\![\, d_{\mathcal{P}} \,]\!]_p$. □

Similarly, for every diagram $d \in \mathsf{Petri}(0,0)$, one can construct a Petri net $\mathcal{P}_d$ with the following recipe: by Lemma 137, d can be written in trace canonical form, namely there exists a diagram $c \in \mathsf{Rc}(p,p)$ such that (5.7) holds. The diagram c denotes an additive relation $[\![c]\!] \subseteq \mathbb{N}^p \times \mathbb{N}^p$ that, by Proposition 46, has an Hilbert basis. This basis can be represented as a matrix $A \colon t \to p + p$ for some $t \in \mathbb{N}$ representing the dimension of the basis. The matrix A can be decomposed into two matrices $U, V \colon t \to p$ such that $A = \begin{pmatrix} U \\ V \end{pmatrix}$. We define $\mathcal{P}_d := (\underline{p}, \underline{t}, U, V)$, that is, $\underline{p}$ and $\underline{t}$ are the sets of places and transitions, while U and V play the role of ${}^\circ-$ and $-^\circ$, respectively.

Again, we can prove that the operational behaviour is preserved.

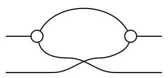
Proposition 147. *For all* $d \in \mathsf{Petri}(0,0)$, $[\![d]\!]_p \sim \mathsf{Fire}(\mathcal{P}_d)$

Proof. First, observe that by construction $(\mathbf{x}, \mathbf{y}) \in [\![c]\!]$ iff there exists $\mathbf{f} \in \mathbb{N}^t$ such that $U\mathbf{f} = \mathbf{x}$ and $V\mathbf{f} = \mathbf{y}$. Therefore, by Lemma 145, $(\mathbf{a}, \mathbf{a}') \in [\![d]\!]_p$ iff there exists $\mathbf{f} \in \mathbb{N}^t$ such that $U\mathbf{f} \le \mathbf{a}$ and $\mathbf{a}' = \mathbf{a} - U\mathbf{f} + V\mathbf{f}$. That is $(\mathbf{a}, \mathbf{a}') \in [\![d]\!]_p$ iff $(\mathbf{a}, \mathbf{a}') \in \mathsf{Fire}(\mathcal{P}_d)$. □

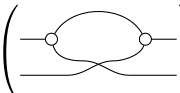
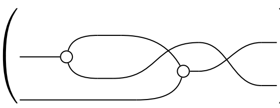
By virtue of Propositions 146 and 147 together, the lts coming from Petri nets and diagrams in $\mathsf{Petri}(0,0)$ are in one-to-one correspondence (modulo $\sim$).

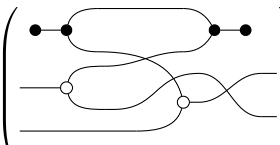
5.2.2 Classifying notions of state

Because Rc_s is isomorphic to $\mathsf{St}(\mathsf{Rc})$ which is itself isomorphic to $\mathsf{St}(\mathsf{AddRel})$, the semantics $[\![-]\!]_p \colon \mathsf{Petri} \to \mathsf{St}(\mathsf{AddRel})$ factors through a prop morphism $P \colon \mathsf{Petri} \to \mathsf{Rc}_s$. Once we know that the behaviour of $\to\!\!\bigcirc\!-$ is an arrow of $\mathsf{St}(\mathsf{AddRel})$, the encoding of $\to\!\!\bigcirc\!-$ in Rc_s is completely determined. First, we can write the place's semantics as a $\mathsf{St}(\mathsf{Rc})$ diagram[2]:

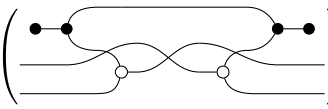
$$[\![\to\!\!\bigcirc\!-]\!]_p := \left\{ \left(\begin{pmatrix} p \\ m \end{pmatrix}, \begin{pmatrix} m+p-n \\ n \end{pmatrix} \right) \mid n \le p \right\} = [\![$$  $$]\!] \qquad (5.38)$$

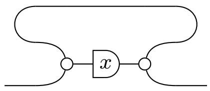
Using the isomorphism $\langle Z, R \rangle^{-1} = G \colon \mathsf{St}(\mathsf{Rc}) \to \mathsf{Rc}_s$ from the proof of Theorem 138,

$$G\Big($$  $$\Big) = G\Big($$  $$\Big) \qquad (5.39)$$

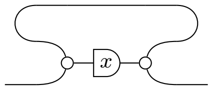
$$= G\Big($$  $$\Big) \qquad (5.40)$$

[2]Technically, a morphism of $\mathsf{St}(\mathsf{T})$ is defined as a pair (s, d) of a number of state-passing wires, and a morphism of T. We omit the number of stateful wires where no confusion can arise.

$= G\Big($  $\Big)$ (5.41)

$=$  (5.42)

This shows that P: Petri $\to$ Rc_s is given by

$P\,(\rightarrow\!\bigcirc\!-) :=$  (5.43)

and the identity on Rc.

We have shown that the stateful resource calculus is at least as expressive as Petri (and thus Petri nets). In fact Petri is *strictly less expressive* than Rc_s, in the sense that not all lts definable in Rc_s correspond to diagrams of Petri. Intuitively, this is because the register of the resource calculus is *synchronous* whereas the place of Petri nets is *asynchronous*: it can always perform a transition that does not change its internal state. This property holds for all diagrams of the Petri calculus:

Proposition 148. *Let d be in* Petri. *Then* $\left(\begin{pmatrix}\mathbf{a}\\ \mathbf{0}\end{pmatrix}, \begin{pmatrix}\mathbf{a}\\ \mathbf{0}\end{pmatrix}\right) \in [\![\, d\,]\!]_p$ *for all* $\mathbf{a}$.

Proof. By induction. The only case of interest is $\rightarrow\!\bigcirc\!-$, and $\left(\begin{pmatrix}a\\ 0\end{pmatrix}, \begin{pmatrix}a\\ 0\end{pmatrix}\right) \in [\![\rightarrow\!\bigcirc\!-]\!]_p$ by (5.30). □

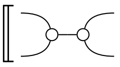
Given that $\left(\begin{pmatrix}a\\ 0\end{pmatrix}, \begin{pmatrix}a\\ 0\end{pmatrix}\right) \notin [\![-\!\boxed{x}\!-]\!]_s$ we immediately obtain:

Corollary 149. *For all diagrams d in* Petri, $[\![\, d\,]\!]_p \neq [\![-\!\boxed{x}\!-]\!]_s$. □

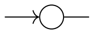
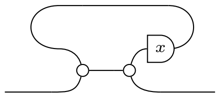
The expressiveness result strengthens the claim that $-\!\boxed{x}\!-$ is a more canonical than $\rightarrow\!\bigcirc\!-$ to introduce state to Rc.

Moreover, we can use the example of $\rightarrow\!\bigcirc\!-$ to show in a more precise sense how the resource calculus plays the role of a yardstick, guiding the space of design choices for process calculi. Indeed, the same kind of analysis can be performed with other interpretations of Petri nets.

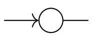
Banking semantics. One example from the literature is the *banking semantics* [BMM11, SMMB13], which defines behaviour differently. In our formalism, the banking semantics interprets the place as

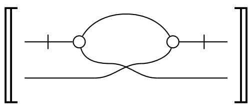
$$\rightarrow\!\bigcirc\!- \;\mapsto \left\{ \left(\begin{pmatrix}p\\ m\end{pmatrix}, \begin{pmatrix}p'\\ n\end{pmatrix}\right) \mid p + m = p' + n \right\} = [\![$$  $$]\!]$$ (5.44)

And a similar computation yields a diagram in Rc_s with the same operational behaviour:

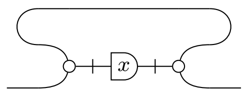
 $\mapsto$  (5.45)

C/E nets. The Petri nets that have appeared so far are sometimes called P/T nets, in contrast with *C/E nets* (also known as elementary net systems). For C/E nets, the places can hold *at most one* token each. For this, the stateful resource calculus is not sufficient—we need the affine constant as well. Let $\mathsf{Rc}_{sa} := \mathsf{Rc}_a + X$; using the ticked wires of Chapter 4, Section 4.4 to represent resources bounded by one, the place of a C/E net can be interpreted as

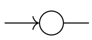
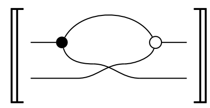
 $$\mapsto \left\{ \left(\begin{pmatrix} p \\ m \end{pmatrix}, \begin{pmatrix} p-n+m \\ n \end{pmatrix} \right) \mid p \leq 1, p-n+m \leq 1, \right\} \quad (5.46)$$

$\mapsto$  (5.47)

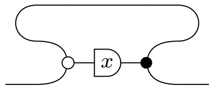
As a Rc_{sa} diagram, this is represented by

 (5.48)

Misc. We could also explore new stateful extensions. One example that we did not find in the literature interprets the place as an *accumulator*:

 $$\mapsto \left\{ \left(\begin{pmatrix} p \\ m \end{pmatrix}, \begin{pmatrix} p+m \\ p+m \end{pmatrix} \right) \right\} =$$  (5.49)

This corresponds to the Rc_s diagram

 (5.50)

For each notion of state, Figure 5.1 summarises the associated diagrams in $\mathsf{St}(\mathsf{Rc})$ (on the left) and Rc_s or Rc_{sa} (on the right). For each of these variants, the (affine) stateful resource calculus provides a complete calculus within which we can express faithfully the lts that capture their behaviour. The proofs of these facts are reformulations of those above, adapted to each specific interpretation.

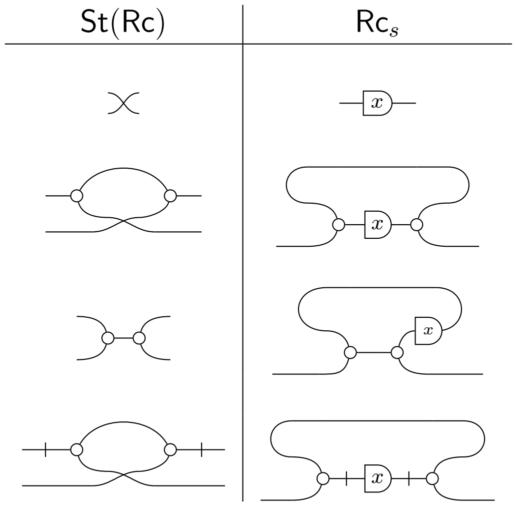


Figure 5.1: Comparison of different stateful extensions.

5.2.3 Open Petri nets

The prop Petri generalises the usual notion of Petri net to *open* Petri nets. Arbitrary morphisms $k \to l$ correspond to Petri nets with a left and right boundary along which they can be composed. This idea is not new. Open Petri nets with various forms of composition, tailored to different purposes, abound in the literature. In what follows, we relate the resource calculus to the connector algebra of [BMM11, SMMB13]. In that work, Petri nets have open transitions along which they can synchronise with their environment. This form of composition is very similar to composition in Petri, as we will see.

A lot of the results of this section imply those of the previous ones on Petri nets. We chose to clarify the correspondence with regular Petri nets before that of open nets—at the risk of repeating some of the same arguments—because the notion of net with boundaries is less standard.

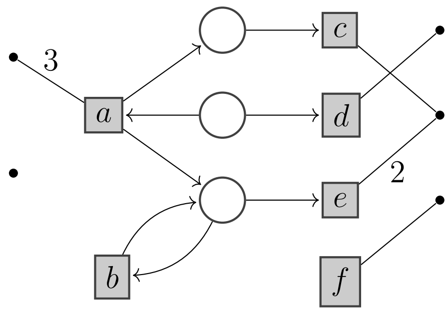
As we hinted in the introduction, this approach is by no means the only formalism that treats Petri nets as open systems. There are many more proposals to deal with Petri nets compositionally [Maz87, NPS95, BCEH05, Rei09, BP17, BM18]. The precise relationship between all of these and the resource calculus will be the object of future work.

The following is Definition 4.1 in [SMMB13].

Definition 150. A *Petri net with boundaries* $\mathcal{N} \colon k \to l$ is a Petri net $(P, T, {}^\circ-, -^\circ)$, equipped with functions ${}^\bullet- \colon T \to \mathbb{N}^k$ and $-^\bullet \colon T \to \mathbb{N}^l$, such that $\langle {}^\bullet-, -^\bullet \rangle \colon T \to \mathbb{N}^k \times \mathbb{N}^l$ is injective (as a map of Set).

The maps ${}^\bullet-$ and $-^\bullet$—understood as Kleisli arrows for the multiset monads or, equivalently, as matrices—associate a weight to each boundary transition. The added condition that they be jointly injective means that we do not allow indistinguishable transitions.

Example 151. The net $2 \to 3$



(5.51)

has boundary maps given by (the non-zero values)

$$ {}^\bullet a = \begin{pmatrix} 3 \\ 0 \end{pmatrix}, c^\bullet = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, d^\bullet = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, e^\bullet = \begin{pmatrix} 0 \\ 2 \\ 0 \end{pmatrix}, f^\bullet = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} $$

Note that the assignment of transitions to the boundaries is undirected.

In order to compose nets with boundaries, [SMMB13] implicitly uses the existence of weak pullbacks in $\mathsf{Mat}_{\mathbb{N}}$ (Proposition 91). In fact, composition is defined as in $\mathsf{InjSpan}_{\mathbb{N}}$ from Remark 100: first take the weak pullback of the two boundary maps, then factor through the surjective-injective factorisation system in Set. Given two nets with boundaries $\mathcal{M}\colon k \to l$ and $\mathcal{N}\colon l \to m$, the authors define $\mathcal{M} ; \mathcal{N}$ as the net with

- places $P_{\mathcal{M}} + P_{\mathcal{N}}$;
- transitions the minimal elements of the set of transactions of ${}^\bullet(-)_{\mathcal{M}}$ and $(-)^\bullet{}_{\mathcal{N}}$ with indistinguishable transactions identified (cf. Remark 100);
- pre and post-condition functions are given by ${}^\circ(\mathbf{a}, \mathbf{b})_{\mathcal{M};\mathcal{N}} = {}^\circ(\mathbf{a})_{\mathcal{M}} + {}^\circ(\mathbf{b})_{\mathcal{N}}$ and $(\mathbf{a}, \mathbf{b})_{\mathcal{M};\mathcal{N}}{}^\circ = (\mathbf{a})_{\mathcal{M}}{}^\circ + {}^\circ(\mathbf{b})_{\mathcal{N}}$;
- boundary functions are given by ${}^\bullet(\mathbf{a}, \mathbf{b})_{\mathcal{M};\mathcal{N}} = {}^\bullet(\mathbf{a})_{\mathcal{M}} \in \mathbb{N}^k$ and $(\mathbf{a}, \mathbf{b})_{\mathcal{M};\mathcal{N}}{}^\bullet = (\mathbf{b})_{\mathcal{N}}{}^\bullet \in \mathbb{N}^m$.

Petri nets with boundaries can also be composed in parallel: for $\mathcal{N}_1\colon k_1 \to l_1$ and $\mathcal{N}_2\colon k_2 \to l_2$, $\mathcal{N}_1 \oplus \mathcal{N}_2$ has $P_1 + P_2$ as places, $T_1 + T_2$ as transitions, ${}^\circ-$ defined as the direct sum of ${}^\circ(-)_1$ and ${}^\circ(-)_2$ and similarly for $-^\circ$, ${}^\bullet-$ and $-^\bullet$.

Remark 152. To obtain a prop (i.e., a strict monoidal category), nets with boundaries should be defined as isomorphism classes of nets. In [SMMB13, Section 4.1] the authors introduce morphisms of nets $\mathcal{N} \to \mathcal{M}$ as pairs of maps $f_P : P_{\mathcal{N}} \to P_{\mathcal{M}}$ and $f_T : T_{\mathcal{N}} \to T_{\mathcal{M}}$ that commute with the pre and post-condition functions and with the boundary maps. A morphism is an isomorphism when its two components are bijections.

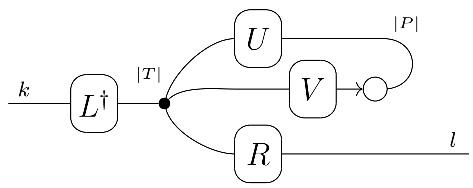
With these two operations, (isomorphism classes of) nets with boundaries form a prop [SMMB13, Proposition 5.1] that we denote by PTNet, as in the original paper.

Petri nets with boundaries have an operational semantics with witnesses for the boundary transitions.

Definition 153. Let $\mathcal{N} : k \to l$ be a net with boundaries. $\mathsf{Fire}\,(\mathcal{N}) \subseteq \mathbb{N}^{|P|+k} \times \mathbb{N}^{|P|+l}$ is the relation given by

$$\left(\begin{pmatrix}\mathbf{a}\\ \mathbf{b}\end{pmatrix}, \begin{pmatrix}\mathbf{a}'\\ \mathbf{c}\end{pmatrix}\right) \in \mathsf{Fire}\,(\mathcal{P}) \text{ iff } \exists \mathbf{t} \in \mathbb{N}^T \text{ such that } \begin{cases} {}^\circ\mathbf{t} \leq \mathbf{b},\ \mathbf{a}' = \mathbf{a} - {}^\circ\mathbf{t} + \mathbf{t}^\circ \\ {}^\bullet\mathbf{t} = \mathbf{b},\ \mathbf{t}^\bullet = \mathbf{c} \end{cases}$$

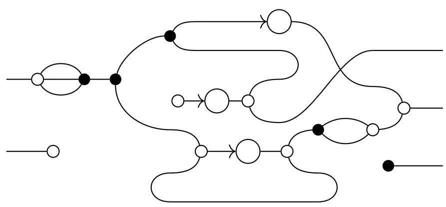
To each net with boundaries we can associate a diagram in Petri. We can essentially proceed as we did for regular Petri nets. Given a net with boundaries $\mathcal{N} : k \to l$, by choosing a total order on places and transitions, the pre and post-conditions ${}^\circ-$, $-^\circ$ can be interpreted as matrices of type $|T| \to |P|$ and, similarly, ${}^\bullet-$, $-^\bullet$ as matrices $|T| \to k$ and $|T| \to l$ respectively. All of these can be represented as Rc diagrams, say U and V for ${}^\circ-$ and $-^\circ$; L and R for ${}^\bullet-$ and $-^\bullet$, respectively. Let

$D\mathcal{N} =$ 

(5.52)

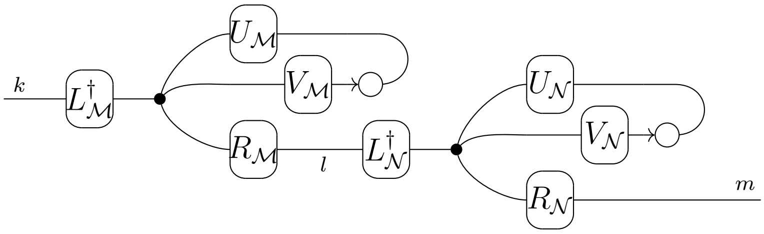
That this encoding is independent from the choice of ordering on P and T follows from a reasoning completely analogous to Lemma 144 and we will omit it.

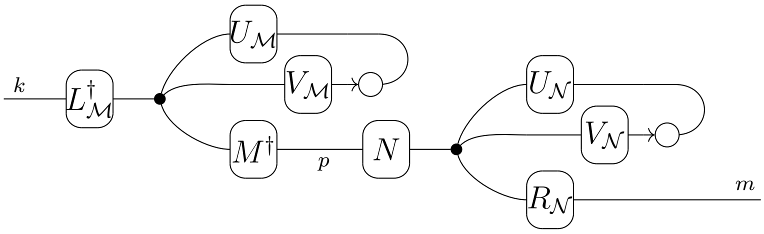
Example 154. The net of Example 151 corresponds to the following Petri diagram:


(5.53)

Proposition 155. $D : \mathsf{PTNet} \to \mathsf{Petri}$ *is a prop morphism.*

This is true for the same reason that there is a prop morphism $\mathsf{Dis}\colon \mathsf{InjSpan}_{\mathbb{N}} \to \mathsf{AddRel}$ (defined in (3.192), Remark 100). D is a functor because composition of nets with boundaries is defined through a weak pullback of matrices, which, by (3.180) is sound for composition of Rc diagrams. Diagrammatically,

$D\mathcal{M}\,;\,D\mathcal{N} =$  (5.54)

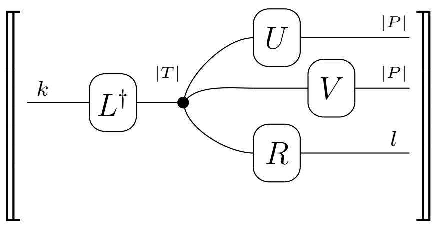
$=$  (5.55)

$= \quad D(\mathcal{M}\,;\,\mathcal{N})$ (5.56)

where M and N are the matrices obtained from taking the weak pullback of $R_{\mathcal{M}}$ and $L^{\dagger}_{\mathcal{N}}$. Furthermore, D is clearly strictly monoidal.

Proposition 156. *Let $\mathcal{N}\colon k \to l$ be a net with boundaries. Then* $\mathsf{Fire}(\mathcal{N}) \sim [\![\, D\mathcal{N} \,]\!]_p$.

Proof. We reason exactly as in the proof of Proposition 146. As usual, fix a total order on P and T; call U, V, L and R the matrices corresponding to $\circ\!-$, $-\!\circ$, $\bullet\!-$ and $-\!\bullet$, respectively. By definition $\left(\begin{pmatrix}\mathbf{a}\\\mathbf{b}\end{pmatrix}, \begin{pmatrix}\mathbf{a}'\\\mathbf{c}\end{pmatrix}\right) \in \mathsf{Fire}(\mathcal{P})$ iff there exists $\mathbf{t} \in \mathbb{N}^T$ such that $U\mathbf{t} \le \mathbf{b}$, $\mathbf{a}' = \mathbf{a} - U\mathbf{t} + V\mathbf{t}$ and $L\mathbf{t} = \mathbf{b}$, $R\mathbf{t} = \mathbf{c}$. We also have



$$= \left\{ \left(\mathbf{b}, \begin{pmatrix}\mathbf{x}\\\mathbf{y}\\\mathbf{c}\end{pmatrix}\right) \mid \exists \mathbf{t} \in \mathbb{N}^{|T|},\ \begin{matrix}\mathbf{x} = U\mathbf{t}, \mathbf{y} = V\mathbf{t}, \\ \mathbf{b} = L\mathbf{t}, \mathbf{c} = R\mathbf{t}\end{matrix} \right\}$$

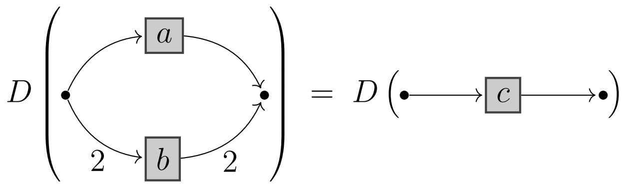
so that, by Lemma 145, we can conclude that

$$\left(\begin{pmatrix}\mathbf{a}\\\mathbf{b}\end{pmatrix}, \begin{pmatrix}\mathbf{a}'\\\mathbf{c}\end{pmatrix}\right) \in \mathsf{Fire}(\mathcal{N}) \quad \text{iff} \quad \left(\begin{pmatrix}\mathbf{a}\\\mathbf{b}\end{pmatrix}, \begin{pmatrix}\mathbf{a}'\\\mathbf{c}\end{pmatrix}\right) \in [\![\, D\mathcal{N} \,]\!]_p.$$

□

We can also translate a diagram of Petri into a net with boundaries as follows. Let $d\colon k \to l$ be a Petri morphism. By Lemma 137, d can be written in trace canonical form, i.e., there exists a diagram $c \in \mathsf{Rc}(p+k, p+l)$ such that (5.7) holds. Choose a representing matrix $A\colon t \to p+k+p+l$ for c and decompose it into $U\colon t \to p$, $V\colon t \to p$, $L\colon t \to k$ and $R\colon t \to l$. Then $\mathcal{N}_d :== (\underline{p}, \underline{t}, U, V, L, R)$ is a Petri net with boundaries. Reasoning as in Proposition 147, it is not hard to see that the translation preserves the operational semantics.

While we would like an isomorphism, D is not faithful. This is because PTNet allows nets with redundant transitions, i.e., transitions that are the sum of other transitions. For example, these two open nets are different in PTNet but have the same interpretation in Petri:



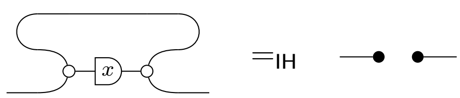
(5.57)

From an operational point of view, redundant transitions are irrelevant as they can always be simulated by firing the other transitions into which they decompose. In the example above, we can simulate the firing of transition b by firing c twice simultaneously. This explains why we still obtain a semantic equivalence between morphisms of Petri and PTNet.

5.3 Trace and traces

In this final section we discuss the role of feedback in the resource calculus. The trace in this prop seems to have a privileged status that sets it apart from its close cousin, $\mathsf{IH}_{\mathbb{K}}$.

The buffer which, as we have seen, plays a central role in the encoding of Petri nets, requires the specific structure of additive relations and collapses to the total relation when interpreted as a *linear relation*:



(5.58)

So, if the departure from linear to additive relations seems subtle at first, it does capture something important about concurrency.

The following two subsections analyse the role of the trace in the stateless and stateful setting, respectively.

We have already pointed out that fRel_+ embeds into AddRel. We can think of it as the linear[3] or resource-preserving sub-prop of AddRel. From the presentation of both props, the embedding is clear: we can just map the single bimonoid of fRel_+ to the ○-bimonoid of AddRel. Moreover, fRel_+ can be equipped with the structure of a trace, implemented via the transitive closure operation. Interestingly, the embedding $\mathsf{fRel}_+ \hookrightarrow \mathsf{AddRel}$ also preserves the trace, as we show in Section 5.3.1. This means that the trace of fRel_+ can be seen as the shadow of the compact structure (and therefore, also of the hypergraph structure) of AddRel.

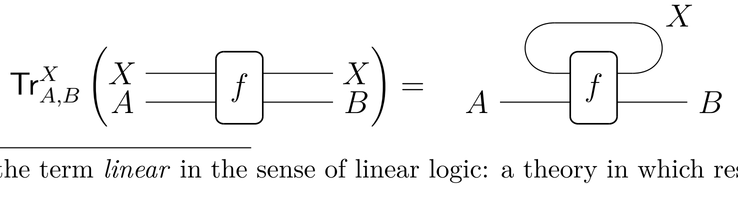
Section 5.3.3 is simply an extended discussion about the role of the trace in the stateful setting. We examine more behavioural notions of equivalence than $\mathsf{St}(\mathsf{AddRel})$, taking inspiration from the semantics of signal flow graphs. There are many possibilities to choose from but we limit ourselves to the finest such notion—trace equivalence—whereby two processes are deemed equivalent when the set of possible traces of observable values at their boundary are equal. For the first time, we do not obtain a precise characterisation of the sets of behaviours. Instead, we just look at the fundamental role that feedback seems to play in this setting, contrasting it with the case of linear relations.

This is a more exploratory section whose results are only loosely knit together by a set of intuitions that will require further work to make precise.

5.3.1 Traced monoidal categories

If (self-dual) compact closed categories allow any port to be connected to any other, irrespective of whether they are in the domain or codomain of morphisms, traced monoidal categories retain the distinction between inputs and outputs, yet allow for domains to be connected to codomains (but not domains to domains or codomains to codomains). The operation that allows this form of feedback is called a trace. The corresponding notion of traced monoidal category was introduced by Joyal, Street and Verity in [JSV96].

Definition 157. A *traced monoidal category* is a smc $(\mathsf{C}, \otimes)$ with a family of maps $\mathsf{Tr}^X_{A,B} \colon \mathsf{C}(X \otimes A, X \otimes B) \to \mathsf{C}(A, B)$ called the trace (or partial trace) that we depict as

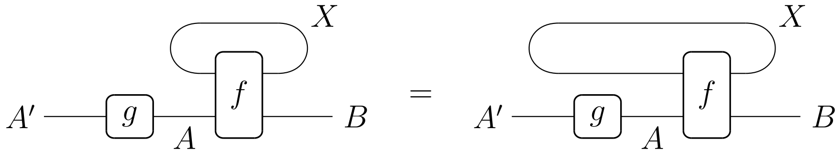


(5.59)

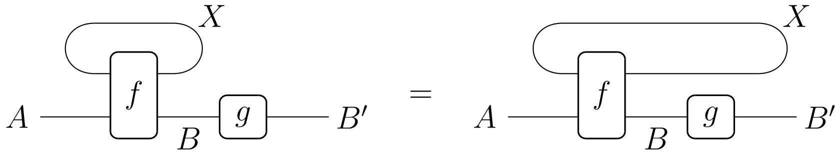
[3]Here, we use the term *linear* in the sense of linear logic: a theory in which resources cannot be copied or deleted.

satisfying

Naturality

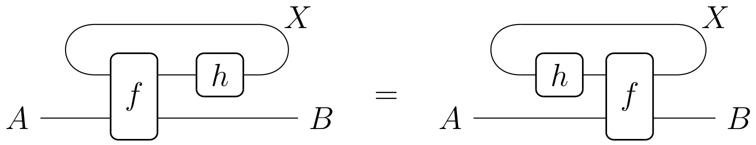


(5.60)



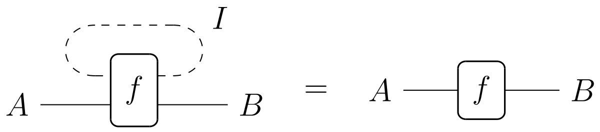
(5.61)

Dinaturality

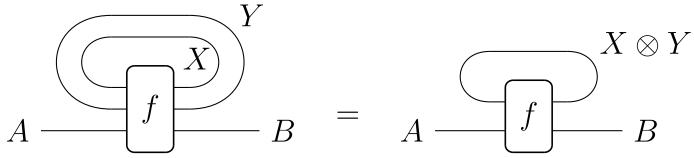


(5.62)

Vanishing

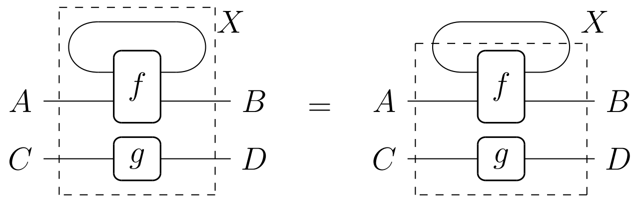


(5.63)



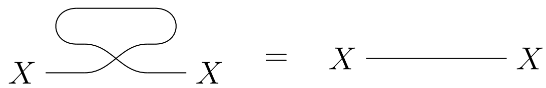
(5.64)

Superposing



(5.65)

Yanking



(5.66)

Every compact closed category is traced using the cup and cap to connect the X in the domain to the one in the codomain. The converse statement is not true.

Example 158. The category fRel_+ admits a trace, even though it is not compact closed. Denote by r^* the reflexive transitive closure of a relation $r : m \to m$. Then, for $r : k + m \to l + m$ we can define the trace by

$$\mathrm{Tr}^m_{k,l}(r) = r^l_k \cup r^m_k ; (r^m_m)^* ; r^l_m \tag{5.67}$$

where r_i^i is the restriction to the relevant components: for example $r_k^m = \iota_k; r; \pi_m$ with ι_k the insertion of $\underline{k}$ in $\underline{k+m}$ and π_m the projection of $\underline{l+m}$ onto $\underline{m}$. Even more explicitly, $r_k^m = r \cap (\underline{k} \times \underline{m})$. The relation r can be decomposed in this way because the disjoint sum is a *biproduct* in Rel.

That (5.67) defines a trace satisfying all the axioms of Definition 157 is well-known and can be found in the original paper [JSV96][Proposition 6.3].

The point of this section is to show that fRel_+ can be embedded into AddRel, by an embedding that preserves the trace.

The formula in (5.67) is consistent with the interpretation of morphisms in fRel_+ in terms of a token flowing around a network, sketched in Chapter 3, Section 3.1. Recall that each element of $\underline{k}, \underline{l}$ and $\underline{m}$ constitutes a port at which it can enter or exit. The relation r specifies the connectivity of the network: the particle can go from port i to port j iff $(i, j) \in r$. The trace is interpreted as a feedback operation sending the particle exiting at a port $i \in \underline{m}$ back to the same i in input position. This operation is iterated until it finally exits on some $j \in \underline{l}$. The following proposition makes the link between formula (5.67) and this interpretation clearer.

Proposition 159. *For a relation $r : m \to m$,*

$$r^* = \bigcup_{n \in \mathbb{N}} r^n$$

where r^n is the nth power of r, defined inductively by $r^0 = 1$ and $r^{n+1} = r^n; r$.

Proof. We need to prove that r^* as defined above is the least reflexive and transitive relation containing r.

- It contains r because it contains all the r^n, in particular $r^1 = r$.
- It is reflexive because for all $a \in \underline{m}$, $(a, a) \in 1_m = r^0 \subseteq r^*$.
- It is transitive. Let (a, b) and (b, c) be two pairs of r^*. By definition, they are in one of the r^n, say $(a, b) \in r^i$ and $(b, c) \in r^j$. Then $(a, c) \in r^{i+j} \subseteq r^*$.
- It is minimal. Let s be a reflexive transitive relation containing r. If we want to show that $r^* \subseteq s$, it is enough to show that $r^n \subseteq s$ for all $n \geq 0$. We can reason by induction on n. First $r^0 = 1 \subseteq s$ by hypothesis. Now assume that $r^n \subseteq s$ for some $n \geq 0$. By the monotony of relational composition $r^n; r \subseteq s; r$. Moreover because $r \subseteq s$, $r^n; r \subseteq s; s$ and since s is transitive, $r^n; r \subseteq s; s \subseteq s$. Therefore s contains r^*.

$\square$

Corollary 160. *$(a, b) \in \mathrm{Tr}^m_{k,l}(r)$ iff there exists a finite (possibly empty) sequence $p_1, \ldots, p_n \in U$ such that $(a, p_1) \in r$, $(p_k, p_{k+1}) \in r$ and $(p_n, b) \in r$.*

Proof. We can substitute the result of Proposition 159 into the defining equation of the trace in fRel_+ to get

$$\mathrm{Tr}^m_{k,l}(r) = r^l_k \cup r^m_k ; \left(\bigcup_{n \in \mathbb{N}} (r^m_m)^n \right) ; r^l_m \tag{5.68}$$

This tells us that $(a, b) \in \mathrm{Tr}^m_{k,l}(r)$ iff either $(a, b) \in r^l_k$ or there exists a sequence $p_1, \ldots, p_n \in U$ such that $(a, p_1) \in r^m_k$, $(p_i, p_{i+1}) \in r^m_m$ and $(p_n, b) \in r^l_m$. $\square$

Remark 161. As stated above, we can think of this sequence as the trajectory of a token starting at a whose dynamics is given by iterating r and looping back n times, as long as the token is in $\underline{m}$, before exiting at b. Note that we could have $n = 0$, in which case $(a, b) \in r^l_k$ and there is no looping through $\underline{m}$.

5.3.2 Embedding relations

Abusing notation slightly, we can also view the multiset functor as an identity-on-object embedding $M : \mathsf{fRel}_+ \hookrightarrow \mathsf{AddRel}$ which sends a relation $r : k \to l$ to its additive closure Mr, the smallest additive relation containing r. It can be described more explicitly as

$$Mr = \left\{ \left(\sum_{i=1}^{n} \mathbf{e}_{p_i}, \sum_{i=1}^{n} \mathbf{f}_{q_i} \right) \middle| (p_i, q_i) \in r, 1 \leq i \leq n \right\} \tag{5.69}$$

with $\mathbf{e}_{p_i}$ and $\mathbf{f}_{q_i}$ basis vectors of $\mathbb{N}^k$ and $\mathbb{N}^l$, respectively. Note how the number of components of both sums is the same $n \in \mathbb{N}$.

Pursuing the particle metaphor, we can interpret M as giving us the ability to observe the flow of several (indistinguishable) particles at once. They can start at the same port or at different ports. The next theorem implies that our particle interpretation is consistent with the feedback operation as well.

The trace in AddRel is given explicitly by

$$\mathsf{Tr}^m_{k,l}(R) = \left\{ (\mathbf{a}, \mathbf{b}) \mid \left(\begin{pmatrix} \mathbf{u} \\ \mathbf{a} \end{pmatrix}, \begin{pmatrix} \mathbf{u} \\ \mathbf{b} \end{pmatrix} \right) \in R \right\} \tag{5.70}$$

for $R \colon m + k \to m + l$.

Theorem 162. *As defined above, M is a traced symmetric monoidal embedding.*

Proof. First, to show that M is an embedding, we need to show is that it is faithful. Let $r, s : k \to l$ be two relations such that $Mr = Ms$. Then, as they are completely determined by their restriction to singleton multisets, $r = s$. And since M is straightforwardly monoidal, all that remains is to prove that it is traced, i.e., that $M\left(\mathrm{Tr}^m_{k,l}(r)\right) = \mathrm{Tr}^m_{k,l}(Mr)$. We show the double inclusion.

- First, assume that $(\mathbf{a}, \mathbf{b}) \in M\left(\mathrm{Tr}^m_{k,l}(r)\right)$. By the definition of the additive closure in (5.69), there exists $p_1, \dots, p_n \in \underline{k}$, $q_1, \dots, q_n \in \underline{l}$ such that

$$\mathbf{a} = \sum_{i=1}^{n} \mathbf{e}_{p_i}, \quad \mathbf{b} = \sum_{i=1}^{n} \mathbf{f}_{q_i} \quad \text{with } (p_i, q_i) \in \mathrm{Tr}^m_{k,l}(r), 1 \le i \le n. \tag{5.71}$$

By Corollary 160, for each i we can find a finite sequence $t_i^0, \dots, t_i^{h(i)}$ verifying $t_i^0 = p_i$, $t_i^{h(i)} = q_i$, $t_i^j \in \underline{m}$ for $1 \le j \le h(i)-1$ and $(t_i^{j-1}, t_i^j) \in r^m_m$ for $1 \le j \le h(i)$. We can take sums (the reader is invited to visualise them happening concurrently) of these trajectories ; write

$$\mathbf{u} = \sum_{i=1}^{n} \sum_{j=1}^{h(i)-1} \mathbf{d}_{t_i^j} \tag{5.72}$$

with $\mathbf{d}_{t_i^j}$ elements of the canonical basis of $\mathbb{N}^m$. Let

$$p'_{s_i+j} = t_i^{j-1} \text{ for } 1 \le i \le n \text{ and } 0 \le j \le h(i) - 1 \tag{5.73}$$

$$q'_{s_i+j} = t_i^{j-1} \text{ for } 1 \le i \le n \text{ and } 1 \le j \le h(i) \tag{5.74}$$

where $s_i = \sum_{k=1}^{i} h(i)$ is the partial sum of the lengths of trajectories up to i, i.e., $p'_1 = t_1^0, p'_2 = t_1^1, \dots, p'_{h(1)} = t_1^{h(1)}, p'_{h(1)+1} = t_2^0, p'_{h(1)+2} = t_2^1, \dots$ and similarly for the q'_i. In plain text, p' is the concatenation of all trajectories t_i^j in which we leave out the exit port in $\underline{l}$ and q' is the concatenation of all the same trajectories, omitting the entry point in $\underline{k}$. They both contain all of the intermediate elements in $\underline{m}$. Then, we have

$$\left(\sum_{i=1}^{h} \mathbf{e}_{p'_i}, \sum_{i=1}^{h} \mathbf{f}_{q'_i}\right) = \left(\begin{pmatrix}\mathbf{u}\\ \mathbf{a}\end{pmatrix}, \begin{pmatrix}\mathbf{u}\\ \mathbf{b}\end{pmatrix}\right) \text{ where } h = \sum_{i=1}^{n} h(i). \tag{5.75}$$

so $\left(\begin{pmatrix}\mathbf{u}\\ \mathbf{a}\end{pmatrix}, \begin{pmatrix}\mathbf{u}\\ \mathbf{b}\end{pmatrix}\right) \in Mr$. Hence, $(\mathbf{a}, \mathbf{b}) \in \mathrm{Tr}^m_{k,l}(Mr)$, which is what we wanted to prove.

- Conversely, assume that $(\mathbf{a}, \mathbf{b}) \in \mathrm{Tr}^m_{k,l}(Mr)$. Then there exists $\mathbf{u} \in \mathbb{N}^m$ such that $\left(\begin{pmatrix}\mathbf{u}\\ \mathbf{a}\end{pmatrix}, \begin{pmatrix}\mathbf{u}\\ \mathbf{b}\end{pmatrix}\right) \in Mr$ and by definition of Mr, we can decompose this pair into a sum of basis elements, say

$$\left(\begin{pmatrix}\mathbf{u}\\ \mathbf{a}\end{pmatrix}, \begin{pmatrix}\mathbf{u}\\ \mathbf{b}\end{pmatrix}\right) = \left(\sum_{i=1}^{n} \mathbf{e}_{p_i}, \sum_{i=1}^{n} \mathbf{f}_{q_i}\right) \tag{5.76}$$

with $(p_i, q_i) \in r$ for $1 \leq i \leq n$. Note first that, by definition of Mr, there must be an equal number of p_i and q_i that are less than m, for each component of the pair. Thus, there is also the same number of p_i and q_i greater than m. Let p_j be the index with the smallest j such that $p_j > m$. We can associate a trajectory to p_j. Let $t^0_j = p_j$ and $t^1_j = q_j$. If $m < q_j \leq m + l$ we are done. If $q_j \leq m$ there exists a smallest j' such that $p_{j'} = q_j$. Let $t^2_j = q_{j'}$.

  We can repeat this process until we reach an element greater than m, which necessarily happens in less than n rounds. Then we can start again, to associate a trajectory to all the $m < p_i \leq m + k$. This process must also reach all the $q_i > m$ because we associate a q_i to each p_i in this way and there are the same number of each that are greater than m. Thus we have built a permutation $\sigma\colon \{p_i \mid p_i > m\} \to \{q_i \mid q_i > m\}$ such that $(p_i, \sigma(p_i)) \in \mathsf{Tr}^m_{k,l} r$ for all $p_i > m$. Thus

$$\left(\sum_{p_i>m} \mathbf{e}_{p_i}, \sum_{p_i>m} \mathbf{f}_{\sigma(p_i)}\right) = \left(\sum_{p_i>m} \mathbf{e}_{p_i}, \sum_{q_i>m} \mathbf{f}_{q_i}\right) = \left(\begin{pmatrix}\mathbf{0}\\ \mathbf{a}\end{pmatrix}, \begin{pmatrix}\mathbf{0}\\ \mathbf{b}\end{pmatrix}\right) = \tag{5.77}$$

  and finally $(\mathbf{a}, \mathbf{b}) \in M\left(\mathsf{Tr}^m_{k,l} r\right)$.

□

5.3.3 Stream semantics

We now come to the traces part of this section (and the more speculative part of this thesis). By "traces" we mean the possibly infinite stream of values that denotes the pattern of interaction of a process with its environment.

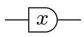
The semantics of stateful systems in St(AddRel) or St(PolyRel) is highly intentional, closer in spirit to an operational semantics (cf. Remark 139), and too discriminatory to constitute a useful behavioural equivalence. We would like to find a coarser notion of equivalence that does not explicitly keep track of any internal state, but only of the behaviour that processes display at their boundary.

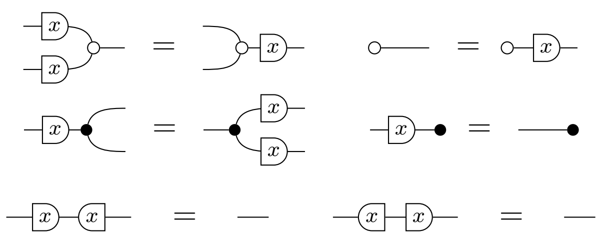
This has been done successfully for linear systems. In the theory of signal flow graphs, the diagrams are interpreted as stream transformers. These ideas were already present in the work of Shannon and were reformulated in categorical terms by [BSZ14, BSZ15, FSR16]. It is natural to ask whether diagrams of Rc_s admit a similar stream semantics. Because we are missing both additive and multiplicative inverses, the answer to this question will necessarily be more delicate than for signal flow graphs.

The first difficulty is in circumscribing precisely the sort of behaviours that we can express. We can turn to signal flow graphs for some inspiration: they admit two interpretations in terms of linear relations.

(a) As linear subspaces of formal Laurent series (finite in the past, possibly infinite in the future) [BSZ14, BSZ15]. In operational terms, all registers are initialised with the value 0.

(b) As linear subspaces of bi-infinite streams (infinite in both directions) [FSR16]. In operational terms, the registers are allowed to contain an arbitrary value at the start of the computation.

These two interpretations differ fundamentally in what constitute their scalars: for (a) scalars are polynomial fractions (elements of the field of fractions $\mathbb{K}(x)$ of the ring $\mathbb{K}[x]$) while, for (b), they are polynomials over an indeterminate x and its formal inverse x^{-1} (elements of the ring $\mathbb{K}[x, x^{-1}]$). In both cases, to achieve complete axiomatisations,  interacts nontrivially with the other connectors. As a result, they define quotients of $\mathsf{IH}_{\mathbb{F}} + X \cong \mathsf{St}(\mathsf{IH}_{\mathbb{F}})$. Both approaches share fundamental equations:



(5.78)

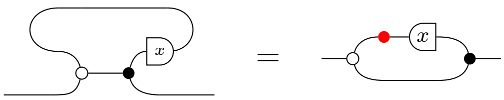
(5.79)

(5.80)

For (a), there is a realisability theorem [BSZ15, Section 5, Theorem 5] demonstrating that all sets of streams that can be expressed in $\mathsf{IH}_{\mathbb{K}(x)}$ are precisely the rational behaviours, i.e., matrices over the ring of rationals $\mathbb{K}\langle x\rangle$ (polynomial fractions whose denominator has non-zero leading coefficient). This correspondence holds at the level of the underlying sets and not for the relations themselves: every linear relation over $\mathbb{K}(x)$ is equal—as a set—to the graph of a matrix over $\mathbb{K}\langle x\rangle$, possibly with its input and output ports reshuffled. This result is important in operational terms because it connects $\mathsf{IH}_{\mathbb{K}(x)}$ with automata theory. Indeed, rational behaviours are those recognised by $\mathbb{K}$-weighted finite-state machines. This is a consequence of the fact that

$$\sum_{n\geq 0} x^n = \frac{1}{1-x} \tag{5.81}$$

This equality reduces iteration to taking multiplicative inverses of polynomials (with non-zero leading coefficient). Its diagrammatic translation is

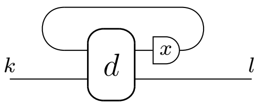


(5.82)

Recall that the scalar denoted by a dot is -1.

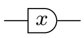
From this, we can prove ([Zan15, Theorem 4.21]) that $\mathsf{Mat}_{\mathbb{K}\langle x\rangle}$ is isomorphic to the subprop SF of $\mathsf{IH}_{\mathbb{K}(x)}$ defined inductively as follows:

- if d is in image of the embedding of $\mathsf{Mat}_{\mathbb{K}[x]}$, then it is in SF;
- if $d\colon k+1 \to l+1$ is in SF, then

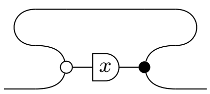
  

  (5.83)

  is also in SF;
- $c\colon k \to l$ and $d\colon l \to m$ are in SF, then so is $c\,;d$;
- $d_1\colon k_1 \to l_1$ and $d_2\colon k_2 \to l_2$ are in SF, then so is $d_1 \oplus d_2$.

Note that the notion of trace that defines SF is not the categorical trace induced by the compact structure but a guarded version, with  as guard. The correspondence between SF and $\mathsf{Mat}_{\mathbb{K}\langle x\rangle}$ is interesting because traced monoidal categories are not generally presented by a symmetric monoidal signature.

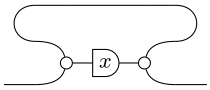
For $\mathbb{N}$, because we lack additive and multiplicative inverses, equation (5.81) does not hold and the behaviour of the buffer



(5.84)

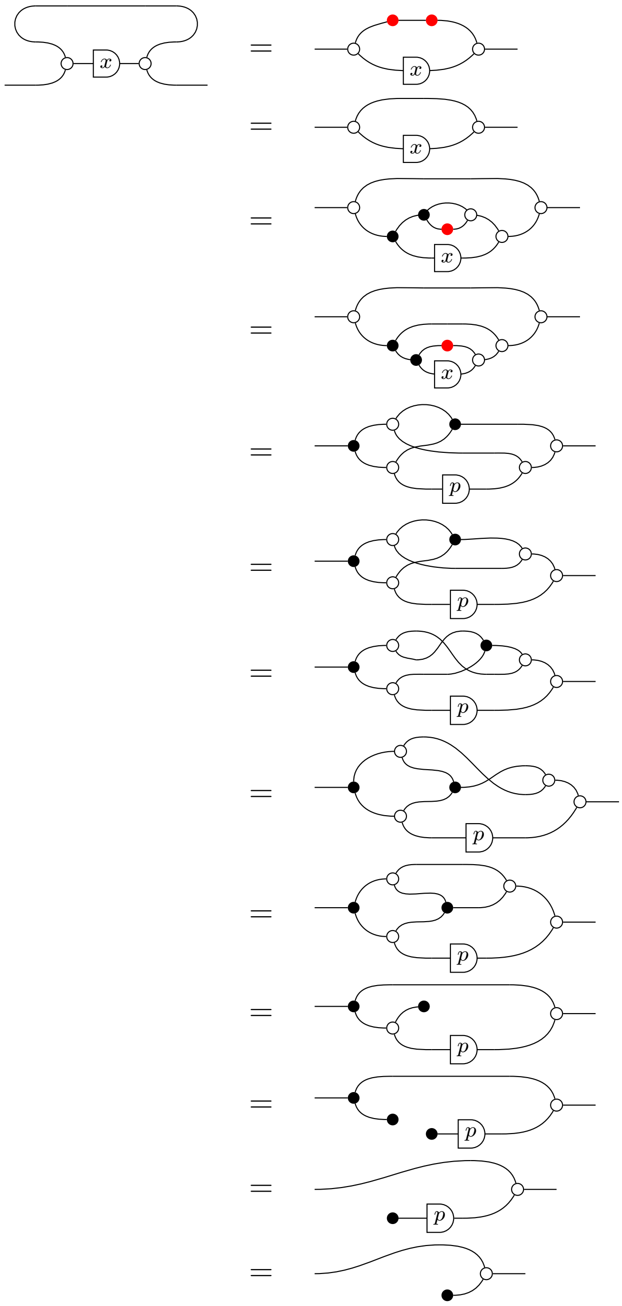
cannot be reduced to division by the scalar $1-x$. As a result, the guarded trace of (the equivalent of) SF over the naturals cannot be further decomposed. We do not know how to present this traced monoidal category by a monoidal signature (in fact, we conjecture that this is not possible).

Furthermore, as we mentioned above, the asynchronous buffer



(5.85)

reduces to the total relation, when interpreted as a linear relation. Indeed, the possibility of storing negative values in the register affords transitions with arbitrary numbers on the left and on the right ports. The reader is invited to refer back to the equations of $\mathsf{IH}_{\mathbb{F}}$ in Fig. 2.3, with $\mathbb{F}$ the field of fractions of some polynomial ring $\mathbb{K}[x]$, to follow the graphical proof below.



(5.86)

(5.87)

(5.88)

(5.89)

where $p = 1 - x$ (5.90)

(5.91)

(5.92)

(5.93)

(5.94)

(5.95)

(5.96)

(5.97)

(5.98)

= (5.99)

The preceding discussion suggests that the sets of behaviours captured by interpreting Rc_s on streams are richer than those of IH and that the trace plays an essential role. The bad news is that this makes it a lot harder to axiomatise. However, we believe that it will lead to new and fruitful connections with automata theory, in particular with the axiomatisation of Kleene algebras, as given in [Koz94]. Indeed, many of the axioms in this work can be derived purely diagrammatically and those that resist a diagrammatic interpretation point us in interesting directions for new research.

Chapter 6

Conclusion

We set out with the objective of characterising the equational theory of the connector algebras of [BMM11] and this has led us to develop Rc.

The resource calculus combines the advantages of syntax-free approaches to the specification of distributed systems, such as Petri nets, with the modularity of process algebras. Its semantics, in terms of additive relations, recognises the central role of resources in concurrency. It is formulated within an elegant graphical formalism that borrows its basic constituents from signal flow graphs. Furthermore, the addition of constant resources and state to the language extends its expressive power to capture a wider class of behaviours, from the stateless primitives of coordination languages to the labelled transition systems of Petri nets.

From a theoretical perspective, the resource calculus undoubtedly has something fundamental to say about the structures of concurrency, as our two main case studies demonstrate. From a more practical point of view, it is perhaps more difficult to evaluate our contribution at this point. We will have to wait and see whether the resource calculus is adopted by others to shed light on old problems and, maybe, contribute to solve them.

We believe that the resource calculus opens up several interesting avenues of research, some of which we have discussed in the main development. These are the subjects of ongoing work.

Develop a modular perspective on additive/polyhedral relations. The completeness of the resource calculus for additive relations relies on a somewhat monolithic proof technique. In particular, it presents a challenge for the modular approach of [Zan15]. With these methods, it is possible to build complete calculi for categories of relations progressively, from those of sub-categories, as long as they admit colimits and compatible factorisation systems [FZ17]. Unfortunately, as highlighted in Section

3.8, this is far from being the case for the category of matrices over an arbitrary semiring. However, we still have access to weak colimits and minimal representations. We hope to investigate how to exploit the existing structure to derive distributive laws or any new mathematically principled way to combine presentations of sub-theories [Had17].

Design efficient rewriting procedures for the resource calculus. If string diagrams are powerful and intuitive tools to reason about open systems, the complexity of the equational theory of the resource calculus can be prohibitive when dealing with large diagrams. Therefore, it is natural to look for rewriting systems and reduction strategies to assist or even automate the derivation of complicated equalities. In recent years, several authors have developed a solid foundation for diagram rewriting [BGK+16], especially modulo the axioms of Frobenius monoids [BGK+18]. And, contrary to $\mathsf{IH}_{\mathbb{K}}$, the resource calculus only contains one Frobenius monoid, thus providing a more immediate application of existing results. In addition, the same authors have proposed a terminating procedure for rewriting with the bimonoid axioms. We would like to leverage this procedure in future work. Finally, this project is intimately related to the outcome of the first line of work, as distributive laws do not only give modular perspective on complex theories but correspond to weakly normalising and confluent rewriting systems.

Enrich the existing calculi to capture coarser notions of equivalences. The stateful resource calculus satisfactorily models of a very intensional notion of process equivalence. We would like to quotient it to capture coarser forms of equivalence, like trace equivalence and bisimilarity. it is not clear what the corresponding semantics should be. Clarifying this would be an important breakthrough. Another related problem is to look for the equations that the asynchronous buffer should satisfy directly—without encoding it with the register—to understand the differences between asynchronous and synchronous state at the axiomatic level.

Extend existing applications to the verification of concurrent systems. Various model-checking problems for distributed systems can be formulated in terms of Petri net reachability. While most existing approaches favour a non-compositional and globally-specified model, Paweł Sobociński and his collaborators have developed `Penrose`, a compositional reachability checker for C/E nets nets [SS13, RSS14]. Their algorithm exploits compositionality by decomposing nets into smaller components, mapping them to NFA that encode the reachability problem for each of them and combining the results at the semantic level. It also exploits process equivalence for efficiency, in order to discard redundant internal states without changing the behaviour of the automaton. Provided with the right decomposition, this procedure has been shown to outperform many of the existing tools. We believe that the resource calculus can provide a basis to reason about process equivalence for larger classes of Petri nets and therefore extend the range of applications of `Penrose`. Beyond reachability, there are many other properties worth tackling from a compositional point of view: liveness, safety, coverability and fairness to name only a few.

Explore links to linear and convex optimisation. Additive and polyhedral relations over $\mathbb{R}_+$ correspond respectively to polyhedral cones and polyhedra in the sense of convex geometry. These are the basic building blocks of constraint sets in linear programming. While we have little hope that optimisation itself is compositional, it would be worthwhile to study duality theorems from this category-theoretic perspective. Furthermore, we could carry the same exploratory work to try to reformulate some fundamental results of integer linear programming from the point of view of additive/polyhedral relations and understand their relative complexity at this level.

Bibliography

[Abr06] Samson Abramsky. What are the fundamental structures of concurrency? we still don't know! *Electronic Notes in Theoretical Computer Science*, 162:37–41, 2006.

[AGN96] Samson Abramsky, Simon J. Gay, and Rajagopal Nagarajan. Interaction categories and the foundations of typed concurrent programming. In *Proceedings of the NATO Advanced Study Institute on Deductive Program Design, Marktoberdorf, Germany*, pages 35–113, 1996.

[ASW09] Luisa Albasini, Nicoletta Sabadini, and Robert F. C. Walters. Cospans and spans of graphs: a categorical algebra for the sequential and parallel composition of discrete systems. *Preprint arXiv:0909.4136*, 2009.

[Bac14] Miriam Backens. The ZX-calculus is complete for stabilizer quantum mechanics. *New Journal of Physics*, 16(9):093021, 2014.

[BCEH05] Paolo Baldan, Andrea Corradini, Hartmut Ehrig, and Reiko Heckel. Compositional semantics for open Petri nets based on deterministic processes. *Mathematical Structures in Computer Science*, 15(1):1–35, 2005.

[BCR17] John Baez, Brandon Coya, and Franciscus Rebro. Props in network theory. *Preprint arXiv:1707.08321*, 2017.

[BE15] John Baez and Jason Erbele. Categories in control. *Theory and Applications of Categories*, 30:836–881, 2015.

[BF15] John Baez and Brendan Fong. A compositional framework for passive linear networks. *Preprint arXiv:1504.05625*, 2015.

[BFP16] John Baez, Brendan Fong, and Blake Pollard. A compositional framework for markov processes. *Journal of Mathematical Physics*, 57(033301), 2016.

[BG01] Roberto Bruni and Fabio Gadducci. Some algebraic laws for spans. *Electronic Notes in Theoretical Computer Science*, 44(3):175–193, 2001.

[BGK+16] Filippo Bonchi, Fabio Gadducci, Aleks Kissinger, Paweł Sobociński, and Fabio Zanasi. Rewriting modulo symmetric monoidal structure. In *Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science (LiCS)*, pages 710–719, 2016.

[BGK+18] Filippo Bonchi, Fabio Gadducci, Aleks Kissinger, Paweł Sobociński, and Fabio Zanasi. Rewriting with Frobenius. In *Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science (LiCS)*, pages 165–174, 2018.

[BHP+19] Filippo Bonchi, Joshua Holland, Robin Piedeleu, Paweł Sobociński, and Fabio Zanasi. Diagrammatic algebra: from linear to concurrent systems. *Proceedings of the 46th ACM SIGPLAN Symposium on Principles of Programming Languages*, 3:25, 2019.

[BHPS17] Filippo Bonchi, Joshua Holland, Dusko Pavlovic, and Paweł Sobociński. Refinement for signal flow graphs. In *Proceedings of the 28th International Conference on Concurrency Theory (CONCUR)*, pages 24:1–24:16, 2017.

[BLM06] Roberto Bruni, Ivan Lanese, and Ugo Montanari. A basic algebra of stateless connectors. *Theoretical Computer Science*, 366(1–2):98–120, 2006.

[BM18] John Baez and Jade Master. Open Petri nets. *Preprint arXiv:1808.05415*, 2018.

[BMM11] Roberto Bruni, Hernán C. Melgratti, and Ugo Montanari. A connector algebra for P/T nets interactions. In *Proceedings of the 22nd International Conference on Concurrency Theory*, pages 312–326, 2011.

[BN37] Richard Brauer and Cecil Nesbitt. On the regular representations of algebras. *Proceedings of the National Academy of Sciences*, 23(4):236–240, 1937.

[BP17] John Baez and Blake Pollard. A compositional framework for reaction networks. *Reviews in Mathematical Physics*, 29(09):1750028, 2017.

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

[BSAR06] Christel Baier, Marjan Sirjani, Farhad Arbab, and Jan Rutten. Modeling component connectors in reo by constraint automata. *Science of Computer Programming*, 61(2):75–113, 2006.

[BSZ14] Filippo Bonchi, Paweł Sobociński, and Fabio Zanasi. A categorical semantics of signal flow graphs. In *Proceedings of the 25th International Conference on Concurrency Theory (CONCUR)*, pages 435–450. Springer, 2014.

[BSZ15] Filippo Bonchi, Pawel Soboci'nski, and Fabio Zanasi. Full abstraction for signal flow graphs. In *Proceedings of the 42nd Annual ACM SIGPLAN Symposium on Principles of Programming Languages POPL*, pages 515–526, 2015.

[BSZ17] Filippo Bonchi, Paweł Sobociński, and Fabio Zanasi. Interacting Hopf algebras. *Journal of Pure and Applied Algebra*, 221(1):144–184, 2017.

[BW85] Michael Barr and Charles Wells. *Toposes, triples and theories*, volume 278. Springer-Verlag New York, 1985.

[Car91] Aurelio Carboni. Matrices, relations, and group representations. *Journal of Algebra*, 136(2):497–529, 1991.

[CD08] Bob Coecke and Ross Duncan. Interacting quantum observables. In *Proceedings of the 35th International Colloquium on Automata, Languages and Programming (ICALP)*, pages 298–310, 2008.

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

[CK15] Bob Coecke and Aleks Kissinger. Categorical quantum mechanics i: causal quantum processes. *Categories for the Working Philosopher*, 2015.

[CK17] Bob Coecke and Aleks Kissinger. *Picturing Quantum Processes - A first course in Quantum Theory and Diagrammatic Reasoning*. Cambridge University Press, 2017.

[Coe08] Bob Coecke. Axiomatic description of mixed states from Selinger's CPM-construction. *Electronic Notes in Theoretical Computer Science*, 210:3–13, 2008.

[CP10] Bob Coecke and Simon Perdrix. Environment and classical channels in categorical quantum mechanics. In *Proceedings of the 24th International Workshop on Computer Science Logic (CSL)*, pages 230–244, 2010.

[CPP08] Bob Coecke, Eric Oliver Paquette, and Dusko Pavlovic. Classical and quantum structuralism. *Semantic Techniques in Quantum Computation*, pages 29–69, 2008.

[CPV13] Bob Coecke, Dusko Pavlovic, and Jamie Vicary. A new description of orthogonal bases. *Mathematical Structures in Computer Science*, 23(3):555–567, 2013.

[CSC10] Bob Coecke, Mehrnoosh Sadrzadeh, and Stephen Clark. Mathematical foundations for a compositional distributional model of meaning. *Preprint arXiv:1003.4394*, 2010.

[CW87] Aurelio Carboni and R. F. C. Walters. Cartesian bicategories i. *Journal of pure and applied algebra*, 49(1-2):11–32, 1987.

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

[Dij65] EW Dijkstra. Solution of a problem in concurrent programming control. *Communications of the ACM*, 8(9):569, 1965.

[Dij68] Edsger W. Dijkstra. Cooperating sequential processes. In *The origin of concurrent programming*, pages 65–138. Springer, 1968.

[FC13] Marcelo Fiore and Marco Devesas Campos. The algebra of directed acyclic graphs. In *Computation, Logic, Games, and Quantum Foundations. The Many Facets of Samson Abramsky*, pages 37–51. Springer, 2013.

[FM97] José Luiz Fiadeiro and Tom Maibaum. Categorical semantics of parallel program design. *Science of Computer Programming*, 28(2-3):111–138, 1997.

[Fon16] Brendan Fong. *The Algebra of Open and Interconnected Systems.* PhD thesis, University of Oxford, 2016.

[FS18] Brendan Fong and David I. Spivak. Hypergraph categories. *Preprint arXiv:1806.08304*, 2018.

[FSR16] Brendan Fong, Pawel Soboci'nski, 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)*, pages 495–504, 2016.

[FZ17] Brendan Fong and Fabio Zanasi. A universal construction for (co)relations. In *Proceedings of the 7th Conference on Algebra and Coalgebra in Computer Science (CALCO)*, pages 12:1–12:16, 2017.

[GC92] David Gelernter and Nicholas Carriero. Coordination languages and their significance. *Communications of the ACM*, 35(2):96, 1992.

[GH97] Fabio Gadducci and Reiko Heckel. An inductive view of graph transformation. In *Proceedings of the 12th International Workshop on Recent Trends in Algebraic Development Techniques (WADT)*, pages 223–237, 1997.

[Ghi13] Dan R Ghica. Diagrammatic reasoning for delay-insensitive asynchronous circuits. In *Computation, Logic, Games, and Quantum Foundations. The Many Facets of Samson Abramsky*, pages 52–68. Springer, 2013.

[Gir87] Jean-Yves Girard. Linear logic. *Theoretical computer science*, 50(1):1–101, 1987.

[Gir06] Jean-Yves Girard. Le point aveugle. *Cours de logique*, 1, 2006.

[GM00] Fabio Gadducci and Ugo Montanari. The tile model. In *Proof, Language, and Interaction, Essays in Honour of Robin Milner*, pages 133–166, 2000.

[Had17] Amar Hadzihasanovic. *The algebra of entanglement and the geometry of composition*. PhD thesis, University of Oxford, 2017.

[HMH14] Naohiko Hoshino, Koko Muroya, and Ichiro Hasuo. Memoryful geometry of interaction: from coalgebraic components to algebraic effects. In *Proceedings of the Joint Meeting of the 23rd EACSL Annual Conference on Computer Science Logic (CSL) and the 29th Annual ACM/IEEE Symposium on Logic in Computer Science (LiCS)*, pages 52:1–52:10, 2014.

[Hoa72] Charles Antony Richard Hoare. Towards a theory of parallel programming. In *The origin of concurrent programming*, pages 231–244. Springer, 1972.

[HV] Chris Heunen and Jamie Vicary. *Introduction to Categorical Quantum Mechanics*. Oxford University Press (to appear).

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

[JSV96] André Joyal, Ross Street, and Dominic Verity. Traced monoidal categories. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 119, pages 447–468. Cambridge University Press, 1996.

[Kar72] Richard M. Karp. Reducibility among combinatorial problems. In *Complexity of Computer Computations*, pages 85–103. Springer, 1972.

[Kis14] Aleks Kissinger. Finite matrices are complete for (dagger-) hypergraph categories. *Preprint arXiv:1406.5942*, 2014.

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

[Koc04] Joachim Kock. *Frobenius algebras and 2D topological quantum field theories*, volume 59. Cambridge University Press, 2004.

[Koz94] Dexter Kozen. A completeness theorem for Kleene algebras and the algebra of regular events. *Information and Computation*, 110(2):366–390, 1994.

[KSW97a] Piergiulio Katis, Nicoletta Sabadini, and Robert F. C. Walters. Bicategories of processes. *Journal of Pure and Applied Algebra*, 115(2):141–178, 1997.

[KSW97b] Piergiulio Katis, Nicoletta Sabadini, and Robert F. C. Walters. Span(graph): A categorial algebra of transition systems. In *Proceedings of the 6th International Conference on Algebraic Methodology and Software Technology (AMAST)*, pages 307–321, 1997.

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

[Lam74] Leslie Lamport. A new solution of Dijkstra's concurrent programming problem. *Communications of the ACM*, 17(8):453–455, 1974.

[Lam78] Leslie Lamport. Time, clocks, and the ordering of events in a distributed system. *Communications of the ACM*, 21(7):558–565, 1978.

[Law63] F William Lawvere. Functorial semantics of algebraic theories. *Proceedings of the National Academy of Sciences of the United States of America*, 50(5):869, 1963.

[Lei04] Tom Leinster. *Higher Operads, Higher Categories*, volume 298. Cambridge University Press, 2004.

[Lei14] Tom Leinster. *Basic Category Theory*, volume 143. Cambridge University Press, 2014.

[LS15] Jérôme Leroux and Sylvain Schmitz. Demystifying reachability in vector addition systems. In *Proceedings of the 30th Annual ACM/IEEE Symposium on Logic in Computer Science (LiCS)*, pages 56–67, 2015.

[Mac65] Saunders Mac Lane. Categorical algebra. *Bulletin of the American Mathematical Society*, 71:40–106, 1965.

[Maz87] Antoni W. Mazurkiewicz. Compositional semantics of pure Place/Transition systems. In *Proceedings of the 8th European Workshop on Applications and Theory of Petri Nets*, pages 307–330, 1987.

[Mil75] Robin Milner. Processes: a mathematical model of computing agents. In *Studies in Logic and the Foundations of Mathematics*, volume 80, pages 157–173. Elsevier, 1975.

[ML13] Saunders Mac Lane. *Categories for the Working Mathematician*, volume 5. Springer Science & Business Media, 2013.

[Mor14] Jason Morton. Belief propagation in monoidal categories. *Preprint arXiv:1405.2618*, 2014.

[Nie13] Lauritzen Niels. *Undergraduate convexity: from Fourier and Motzkin to Kuhn and Tucker*. World Scientific, 2013.

[NPS95] Mogens Nielsen, Lutz Priese, and Vladimiro Sassone. Characterizing behavioural congruences for Petri nets. In *Proceedings of the 6th International Conference on Concurrency Theory (CONCUR)*, pages 175–189, 1995.

[NW17] Kang Feng Ng and Quanlong Wang. A universal completion of the ZX-calculus. *Preprint arXiv:1706.09877*, 2017.

[OP99] Peter W. O'Hearn and David J. Pym. The logic of bunched implications. *Bulletin of Symbolic Logic*, 5(2):215–244, 1999.

[Plo04] Gordon D. Plotkin. A structural approach to operational semantics. *Journal of Logic and Algebraic Programming*, 60-61:17–139, 2004.

[Rei09] Wolfgang Reisig. Simple composition of nets. In *Proceedings of the 30th International Conference on Applications and Theory of Petri Nets*, pages 23–42, 2009.

[Rey02] John C. Reynolds. Separation logic: A logic for shared mutable data structures. In *Proceedings of the 17th ACM/IEEE Symposium on Logic in Computer Science (LiCS)*, pages 55–74, 2002.

[RGS99] José Carlos Rosales and Pedro A. García-Sánchez. *Finitely Generated Commutative Monoids*. Nova Publishers, 1999.

[RSS14] Julian Rathke, Pawel Soboci'nski, and Owen Stephens. Compositional reachability in Petri nets. In *Proceedings of the 8th International Workshop on Reachability Problems (RP)*, pages 230–243, 2014.

[RSW05] Robert Rosebrugh, Nicoletta Sabadini, and Robert F. C. Walters. Generic commutative separable algebras and cospans of graphs. *Theory and Application of Categories*, 17(6):164–177, 2005.

[Sch04] Andrea Schalk. What is a categorical model of linear logic. *Manuscript, available from* ***http://www.cs.man.ac.uk/~schalk/work.html***, 2004.

[Sel07] Peter Selinger. Dagger compact closed categories and completely positive maps. *Electronic Notes in Theoretical Computer Science*, 170:139–163, 2007.

[Sel11] Peter Selinger. A survey of graphical languages for monoidal categories. *Springer Lecture Notes in Physics*, 13(813):289–355, 2011.

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

[SMMB13] PawełSobociński, Ugo Montanari, Hernan Melgratti, and Roberto Bruni. Connector algebras for C/E and P/T nets' interactions. *Logical Methods in Computer Science*, 9, 2013.

[Sob13] Paweł Sobociński. Nets, relations and linking diagrams. In *Proceedings of the 5th International Conference on Algebra and Coalgebra in Computer Science (CALCO)*, pages 282–298, 2013.

[SS13] Pawel Soboci'nski and Owen Stephens. Penrose: Putting compositionality to work for Petri net reachability. In *Proceedings of the 5th International Conference on Algebra and Coalgebra in Computer Science (CALCO)*, pages 346–352, 2013.

[vG90] Rob J. van Glabbeek. The linear time-branching time spectrum (extended abstract). In *Proceedings of Theories of Concurrency: Unification and Extension*, pages 278–297, 1990.

[Wil07] Jan C Willems. The behavioral approach to open and interconnected systems. *IEEE Control Systems Magazine*, 27(6):46–99, 2007.

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

[Zan16] Fabio Zanasi. The algebra of partial equivalence relations. volume 325, pages 313–333, 2016.