Causal Types for Higher-Order Quantum Theory

Will Simmons

Worcester College

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

Michaelmas 2024

To people who invite me to events in person. How dare you. You know I dislike the outside world, and yet I am helpless to the call of free food. How [redacted] dare you.

Acknowledgements

Starting with the academic side, none deserve more thanks than Aleks Kissinger, whose work with his previous student, Sander Uijlen, inspired much of this thesis and laid the foundations so that I could place a few more bricks. On top of that, being a supervisor, teaching advisor, co-author, and last minute volunteer whenever I couldn't find a speaker for Quantum Lunch seminars, I am incredibly grateful for his support and guidance.

As every idea is influenced by the environment which births it, so too was my work shaped by the fascinating conversations, seminars, and opportunities with the academic community both in and outside of Oxford. Thanks to Nobuko Yoshida and Lutz Straßburger for examining this thesis and for the helpful discussions and feedback, and likewise to Jon Barrett, Stefano Gogioso during my transfer and confirmation of status assessments. For both past and present members of the Quantum group (James, Matt, Cole, John, Chris, Tein, Alex, Lia, Amin, Razin, and many more), I will always be grateful for the variety of talks and topics, insightful whiteboard sessions, and welcoming vibes. Beyond the dreaming spires, I would like to give particular thanks to Alessio Guglielmi, Ross Horne, Tito Nguyễn, and Timothée Hoffreumon for some inspiring conversations that were critical to shaping some of the ideas here, and to the Bath Mathematical Foundations and CHoCoLa seminars for the opportunities to present my work and the great follow-on discussions.

Alongside this PhD I have held a position at Quantinuum, and I would like to express my thanks for their generosity and support in funding this work. Thanks in particular to Ross Duncan for his advice, encouragement, and paper reviews alongside Sean Tull and Robin Lorenz. I am additionally grateful to my team there, from the OG boostboiz to newer colleagues and friends, for providing a working environment which helps developers and researchers to thrive.

For the personal acknowledgements, I am most grateful for my fellow Lamarshians, Julia, Joel, Max, Paddy, Arnold (the neighbour's cat, whose name most definitely is not "Arnold"), and associates, for the daily company, respite, and jests that got me through. Additional thanks to the support from my family, Folkestone and Cambridge friends, my counsellor, Oxford Juggling Club, and the many others who would fall under the catch-all of "drinking buddies".

If I am permitted just one more paragraph of silliness before the serious matter, it will be thanking the little things. To the many milkshakes and spicy potato wraps available across Oxford, thanks for the pick-me-ups. To The Toxhards, thanks for blessing us with "Angus, The Prize-Winning Hog", the soundtrack to my paper-writing delirium. And thanks to the clothing store Aloha from Deer, for enabling the bizarre experience of being approached more at conferences to talk about fashion than my research. Finally, to Diplomat Immune Tea from Aldi, never change[1].

[1] A mere handful of days after submitting this thesis, I found that Aldi replaced this product with one of the same name but changed from orange and peach flavour to the vastly inferior blackberry, elderberries, and echinacea. I give them praise and they twist my words to a cruel fate. This betrayal cuts deep and leaves a taste... well... not of orange and peach.

Abstract

The emerging study of higher-order causal structures in the setting of quantum theory has made crucial developments in our understanding of potential impacts of quantum gravity on how processes can compose, but has historically been treated in a somewhat ad-hoc way that is specific to quantum theory. This thesis is part of an ongoing movement to put the composition of higher-order causal structures front and centre via categorical frameworks, which helps in the generation of theory-independent definitions for causal structures. This thesis builds on Kissinger and Uijlen's Caus $[-]$ construction to add more expressive causal structures and obtain a tight correspondence with a formal logic precisely describing consistency of composite causal structures via proof-nets.

More specifically, the resulting theory features a number of monoidal structures to describe different causal relations between local systems (which may have some internal causal structure themselves), including non-signalling, one-way signalling, bidirectional signalling, compatibility with a directed graph, classical (probabilistic) choice, and combinations of these. We define *causal consistency* of a closed system of black boxes with fixed connections to mean that the composite can be realised with unit probability regardless of the implementations within the boxes, and we investigate a number of logics whose proofs can be translated into causally consistent setups. Completeness is given by proof-nets of *causal logic*, a new logic extending pomset with directed axiom links to describe first-order systems (the degenerate systems which can only transmit information in one direction) in which the proof-net criterion guarantees causal consistency via the absence of any cycle of information flow into which a paradox could be encoded.

After a detour into Measurement-Based Quantum Computing to present a new algorithm for extraction of an ancilla-free circuit from a Pauli Flow, the thesis ends by showing how measurement patterns and parameterised quantum circuits can be formally attributed with causal structures in the Caus $[-]$ framework.

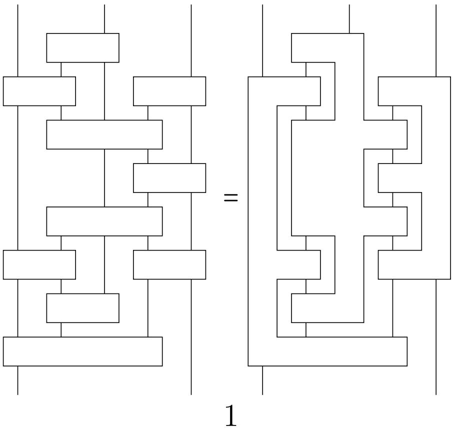
Contents

Chapter 1

Introduction

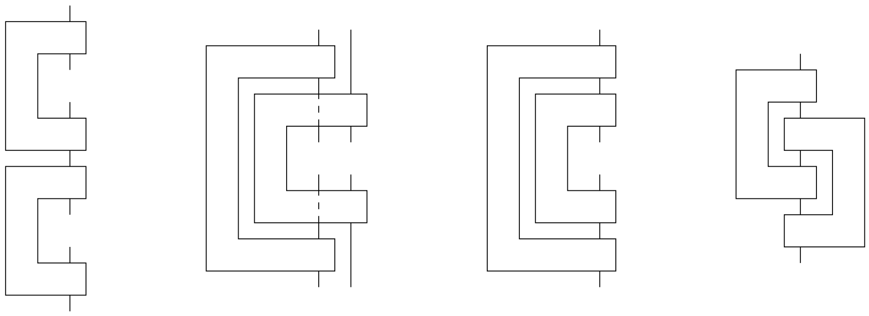
A first-order *process theory* [31] describes a collection of *systems* (equivalently, the space of possible states a physical system can take), and *processes* with designated inputs and outputs which can be combined in sequence or in parallel. Modelling them as *(strict) symmetric monoidal categories* allows us to picture the processes as boxes (morphisms) connected in a string diagram with a consistent arrow of time (in the case of the diagrams below, we read from bottom to top). We interpret each elementary process as *atomic* - they wait until they receive all of their inputs, perform their action, and produce all of their outputs simultaneously.

There are many applications where we may, instead, wish to consider processes that alternate multiple rounds of inputs and outputs. For example, in a multi-party communication protocol we may wish to group all actions of a chosen agent into a single process. This idea of capturing *locality* can carry over into theories of physical processes to describe operations that could be performed within some region of spacetime - if the systems enter and exit the region at different points in time, we can similarly build sets of regions that pass systems back and forth between them. In both examples, we benefit from using a single *black box* per agent/region since, from the perspective of another agent, both the elementary processes used and the way in which they are composed might be unknown.



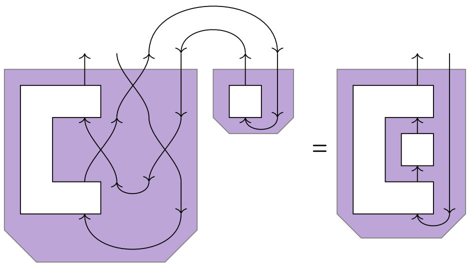
(1.1)

These new, irregularly-shaped boxes can be formalised as kinds of *higher-order maps* [22, 119], transforming input processes to output processes in the sense of higher-order programming: by inserting a regular box into the hole between each output and the next input, we obtain a new process. These higher-order processes are closed under series, parallel, and functional/nested applications, in addition to some more elaborate compositions that cannot be generated by these simple cases.



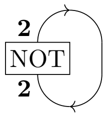
(1.2)

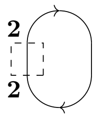
The problem of capturing these compositions mathematically is trivialised when we move to *compact closed categories*, where the wires in our diagrams may be "bent around" through the visual language of *string diagrams*. Then all possible shapes of black boxes can be encoded into simple states, and any composition can be described by some pattern of connecting the output wires.



(1.3)

However, if we were to allow *all* compositions in this way, we may introduce time loops and logical paradoxes, as witnessed e.g. by probabilistic processes that are no longer normalised. Examples include connecting together the input and output of a NOT gate, describing a simple instance of the Grandfather Paradox [11] that "occurs" with probability 0, or doing the same for an identity for the Bootstrap Paradox with probability generally exceeding 1.



$= 0 \neq \mathrm{id}_I \neq \mathrm{id}_I + \mathrm{id}_I =$


(1.4)

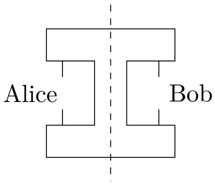
This immediately raises the question: "which string diagrams always preserve normalisation?" This thesis coins the term *causal consistency* to refer to this property of string diagrams.

Definition 1.0.1: Causal consistency (informal)

A string diagram consisting of black boxes is *causally consistent* if, whenever the implementation of each black box respects its normalisation conditions, the overall process is normalised.

If we interpret scalars - processes with no inputs or outputs - as (abstract) probabilities, causal consistency of a closed diagram states that it evaluates to the unit scalar, i.e. the combination of processes can be successfully performed in a laboratory with probability 1.

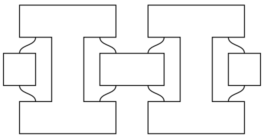
We can find some motivating examples of causally (in)consistent compositions from the study of *process matrices* [96] in quantum foundations. These second-order processes are depicted as boxes with two holes representing locations where local agents, Alice and Bob, can receive an input and perform some process to generate an output.



(1.5)

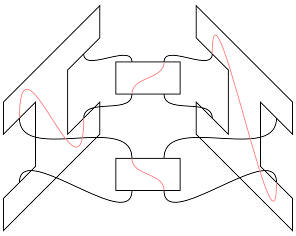
Such a black box abstracts away the order in which Alice and Bob's processes are composed, permitting Alice before Bob, Bob before Alice, probabilistic mixtures/quantum superpositions of the two orders, or even processes with *indefinite causal structure* [96] which locally agree with quantum theory but are incompatible with any of the above.

Given a pair of process matrices, it is straightforward to prove that applying them locally on either side of a single bipartite process is causally consistent:



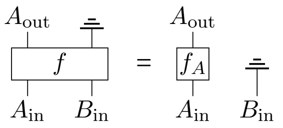
(1.6)

But when we consider composing them with two bipartite processes in parallel, we can break causal consistency by the possible introduction of paradoxical scenarios [75], such as the cycle induced by the red paths below.



(1.7)

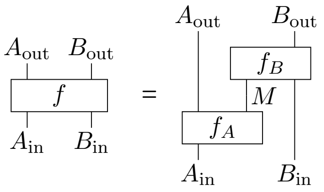
This thesis studies how causal consistency can be pieced together from the *causal structure* of the components (the temporal relations between the different sets of inputs and outputs). The shapes of these boxes are informal descriptions of causal structure which are formalised a number of ways in the literature. Taking the example of a bipartite process with one side acting before the other, one can formulate the family of *one-way signalling* (a.k.a. semi-causal) processes [12] by positing the existence of a preferred *discarding map* $\bar{\top}$ and requiring that the first output is independent of the second input:



(1.8)

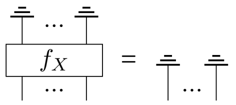
for some reduced process f_A. Intuitively, this captures the notion that an agent who only has access to the right input/output pair of f cannot use the process f to send a message to another agent that only has access to the left input/output pair.

A related concept to one-way signalling is *semi-localisability* [12]. A process is semi-localisable if it can factorise as:



(1.9)

where each process satisfies the causality equation:



(1.10)

Using the equations above, it is straightforward to show that any semi-localisable process is one-way signalling. More generally, one can enumerate the non-signalling-type equations for a black-box process with many input/output pairs that factorises according to a generic directed acyclic graph [77]. We can express such results in a theory-independent manner by treating them as equations between morphisms in a symmetric monoidal category $\mathcal{C}$ equipped with a family of discarding maps $\bar{\top}_A : A \to I$ from any object into the monoidal unit. Notable examples are the symmetric monoidal categories of stochastic matrices, where discarding corresponds to marginalisation, and categories of quantum channels, where discarding is given by the trace functional (the quantum analogue to marginalisation).

The first step this thesis takes towards characterising causal consistency is to build a categorical model of causally consistent processes, aiming for the following three goals:

- *Theory-independence/abstraction* to distinguish facts of causal consistency from the phenomena of a given physical theory;
- The morphisms describe *causally consistent processes* of a particular kind and include every such process;
- It is as *freely compositional* as possible, supporting any string diagram that satisfies causal consistency.

These are met by the $\mathrm{Caus}[-]$ construction of Kissinger and Uijlen [78], which takes a category of "unnormalised" processes and restricts morphisms to those that preserve a normalisation condition, guaranteeing that every closed diagram evaluates to probability 1 and that composition exactly captures causal consistency. It can be applied abstractly to suitable base theories like $\mathrm{Mat}[\mathbb{R}^+]$ (the category of finite matrices of positive reals) or CP^* (the category of finite-dimensional C*-algebras and completely positive maps) to give us deterministic theories of higher-order probabilistic or mixed quantum processes respectively. One can construct objects in the category that encode causal structures between a collection of local systems, such as bipartite first-order causal processes with no signalling in either direction between the parties, one-way signalling, or with no signalling constraints (see Theorem 2.3.23).

The first novel results in this thesis focus on expanding upon the $\mathrm{Caus}[-]$ construction with new monoidal products and new ways to interpret existing monoidal products in terms of causal structures. These new notions of causal structure generalise existing definitions from first-order processes to higher-order - for example, the seq product $\mathbf{A} < \mathbf{B}$ describes states compatible with "$\mathbf{A}$ before $\mathbf{B}$", generalising both one-way signalling and semi-localisability. Other new operators include the additives $\times, \oplus$ capturing probabilistic choice, a closed union $\cup$ capturing union of sets, and graph types $\mathbf{Gr}_G^\Gamma$ to directly encode any fixed causal structure. The highlight results from this exploration are:

Theorem 1.0.2: Seq Equivalence Theorem (informal)

For any choice of local systems $\mathbf{A}$ and $\mathbf{B}$ (which need not resemble first-order processes), a process is one-way signalling from $\mathbf{A}$ to $\mathbf{B}$ iff it can be written as a quasi-probabilistic mixture of semi-localisable processes.

Theorem 1.0.3: First-Order Theorem (informal)

First-order systems are characterised by a collapse of the hierarchy of signalling types.

Theorem 1.0.4: Sum of Orders Theorem (informal)

Bipartite processes that can signal information in both directions (including those with indefinite causal structure) can be expressed as a quasi-probabilistic mixture of processes which each signal information in one direction or the other.

Theorem 1.0.5: Graph Equivalence Theorem (informal)

The Seq Equivalence Theorem generalises to any causal structure: a process satisfies all signalling constraints of a graph iff it can be written as a quasi-probabilistic mixture of processes which each factorise in the shape of the graph.

Theorem 1.0.6: Affine-Bit Sufficiency Theorem (informal)

Channels with arbitrary (quantum or classical) information capacity can be reconstructed as quasi-probabilistic mixtures of a single classical bit.

Theorem 1.0.7: Graph Compatibility Theorem (informal)

A pair of definite causal structures specified by graphs are causally consistent with one another iff the union of the edges gives an acyclic graph.

Kissinger and Uijlen [78] also showed that any Caus$[\mathcal{C}]$ is a model of MLL+Mix [2, 51] with isomorphic units (a.k.a. an ISOMIX category [28]), meaning proofs in MLL+Mix can be used to verify causal consistency of corresponding diagrams. However, MLL+Mix is not sufficient to describe *all* causally consistent string diagrams. Perhaps most notably, there is no special status given to first-order systems, despite the First-Order Theorem generating string diagrams which are only causally consistent when the wires are interpreted as first-order systems.

The most significant contribution made in this thesis takes this result to the extreme and provides a new logic, called *causal logic*, whose proofs are "witnessed" by a family of (causally consistent) string diagrams in Caus$[\mathcal{C}]$ for any base theory $\mathcal{C}$ and, conversely, each string diagram in Caus$[\mathcal{C}]$ has a corresponding proof in the logic. The logic is based on a proof-net criterion that we build synthetically to match the semantics, which identifies that causal *in*consistency can always be shown by witnessing a cycle of information flow into which we can encode a paradoxical process, such as in our process matrix example (Equation 1.7).

The study into causal logic gives rise to some more important results:

Lemma 1.0.8: Switching Lemma (informal)

Given a formula F, the processes of type F^* can be generated from processes with definite causal structures. In particular, the switching graphs of a causal proof-net for F gives a generating set of such causal structures.

Theorem 1.0.9: Causal Characterisation Theorem (informal)

The string diagram corresponding to a formula F is causally consistent in $\mathrm{Caus}\left[\mathcal{C}\right]$ with some interpretation Φ iff there exists a proof-net for F in causal logic. Therefore, causal consistency is independent of the base theory $\mathcal{C}$ and the specific interpretation Φ (up to the identification of degenerate systems).

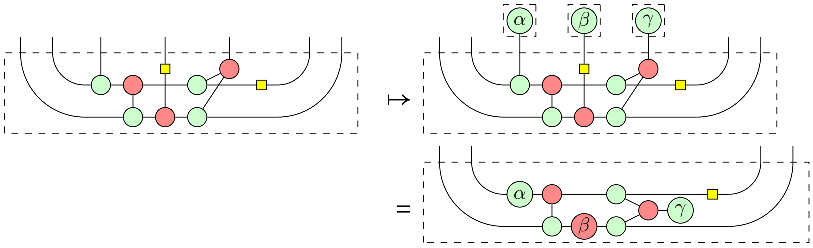
Proposition 1.0.10: Propositions 3.3.5, 3.4.4, & 3.4.7 (informal)

Causal logic is a conservative extension of pomset [100, 102], and can be faithfully encoded into pomset or a variant of MLL+Mix with polarised atoms.

The final work in this thesis grounds this study of causal structures in a practical context, looking at two paradigms of programmable quantum processes: *measurement patterns* [99] and *parameterised quantum circuits*. A measurement pattern performs computation by initialising a quantum system in a fixed *resource state*, then carefully choosing measurements to apply on each qubit to influence the state of the remaining qubits. On the other hand, the more conventional quantum circuits apply operations directly onto the input qubits to modify their state in-place.

Partial orders play a part in both of these by describing limitations on how some operations can be reordered. A *flow* [38, 19] for a measurement pattern contains a partial order over the measurements indicating whether the measurements must occur in a given order or if they can be performed simultaneously. Similarly, gates in quantum circuits may sometimes commute with each other (applying them in either order gives the same effect), and otherwise are forced to appear in a given order to preserve the overall effect. Both of these intuitively feel related to causal structure, specifying temporal constraints on the order in which some things occur, but do not clearly fit the formal definitions based on signalling conditions or factorisations.

To compare these partial orders with genuine causal structures, we can describe parameterised quantum circuits as multi-party open systems with one party per parameter and the actions available to that party correspond to choosing a value for the parameter (shown below through the language of the ZX-calculus [30, 114]).



(1.11)

The same can be done for measurement patterns, where the parties decide the angles of measurements. Using these open systems, we can phrase signalling conditions as whether one party's choice of parameter can influence the marginal state available to another party.

The final major novel result in this thesis is an *extraction algorithm* for measurement patterns which finds a parameterised (pure) quantum circuit implementing the same linear map. The specifics of this algorithm have interesting impacts on how we view the flow and commutation partial orders as causal structure.

Theorem 1.0.11: Circuit Extraction Theorem (informal)

There exists an efficient algorithm that finds an equivalent parameterised quantum circuit for any measurement pattern. Comparing to the main alternative extraction algorithm in the literature [7]:

- It can be applied to a larger class of measurement patterns, including those with different numbers of input and output qubits, and where some measurements may be restricted to a particular basis.

- It is far easier to see how the structure of the measurement pattern (in particular, the corrections related to each measurement) impact the final circuit. In particular, two measurements can be performed simultaneously iff the corresponding rotations in the circuit can commute past each other, so the notions of partial orders for measurement patterns and parameterised circuits coincide.

Theorem 1.0.12: Flow Causality Theorem (informal)

When viewing the resource state of a measurement pattern as an open system, it satisfies all the non-signalling conditions associated with the partial order of its flow.

1.1 Content of the Thesis

This thesis combines and elaborates on the content of the following publications:

- W Simmons: *Relating Measurement Patterns to Circuits via Pauli Flow.* In 18th International Conference on Quantum Physics and Logic (QPL 2021), Electronic Proceedings in Theoretical Computer Science, EPTCS, 343:50-101 [105]
- W Simmons, A Kissinger: *Higher-Order Causal Theories are Models of BV-Logic.* In 47th International Symposium on Mathematical Foundations of Computer Science (MFCS 2022), Leibniz International Proceedings in Informat- ics (LIPIcs), 241:80:1–80:14 [106]
- W Simmons, A Kissinger: *A complete logic for causal consistency* (preprint)[107]

In both [106] and [107], my supervisor and co-author, Aleks Kissinger, proposed some initial questions to me based on feedback he had received on his previous work [78]. He proposed some early candidate definitions for $<$ and otherwise assisted in providing feedback and further direction for work during discussions and presentational improvements to the papers. All proofs in the papers (and all presented in this thesis), including the constructions for causal logic, are my own work.

The thesis is arranged into sections that distinguish novel results from existing work, with the latter marked as "Background" or "Related Work" sections and every borrowed definition or result cites the original text.

Chapter 2 begins with an overview of concepts in quantum causality, how these have previously been modelled in category theory, and a justification of the $\mathrm{Caus}[-]$ construction as originally presented by Kissinger and Uijlen [78]. The rest of the chapter takes a tour of the new operators added to the $\mathrm{Caus}[-]$ framework in the papers [106, 107] and how to relate them to causal structures, including classical (probabilistic) choice, one-way signalling and individual non-signalling conditions, combinations of conditions and compatibilities with graph causal structures, and notions of partial maps.

With the content of $\mathrm{Caus}[\mathcal{C}]$ introduced, we move on to characterising the compositional structure that exists between the operators in the form of a Curry-Howard-Lambek-style correspondence with a logic in Chapter 3. A primer is provided to introduce the relevant logics. We formally introduce causal consistency as a property of string diagrams in $\mathrm{Caus}[\mathcal{C}]$ and show how this entails all inclusions and equalities between operators in $\mathrm{Caus}[\mathcal{C}]$. We give the definitions for *causal logic*, and prove that it precisely characterises causal consistency (over a limited fragment of operators). Studying the proof-nets of *causal logic*, we show that it is both a conservative extension of and completely determined by pomset. The chapter ends with a discussion of how to extend this logic to account for (almost) all operators introduced in Chapter 2.

The majority of Chapter 4 takes a detour away from categorical interpretations of causal structures into Measurement-Based Quantum Computing for the contents of the paper [105]. We provide some background on *measurement patterns* and how correcting measurement errors can implement a pure channel when the pattern has a property called *flow*. We give an algorithm for identifying whether a pattern admits a *Pauli flow* (a very general version of flow which allows qubits to be measured in either a given plane of the Bloch sphere or in a particular Pauli basis). Then, we build up the extraction algorithm implementing the linear map of a given pattern as a product of parameterised rotations determined completely by a focussed Pauli flow. Finally, the chapter concludes by considering measurement patterns within the framework of $\mathrm{Caus}\left[\mathrm{CP}^*\right]$ to see how we can infer compatibility of a pattern with a particular causal structure, using extraction of measurements to witness the required non-signalling conditions, and formalising the view of flow as a causal structure.

1.2 Prerequisites

The work in this thesis treats a number of topics across computer science, mathematics, and physics, making use of a range of mathematical formalisms and concepts. To keep things concise, we will assume some level of familiarity with the common tools and provide background sections to introduce more advanced topics. The list below details the assumed knowledge and some recommended texts for full introductions or reference.

- Finite-dimensional quantum theory and quantum computing [95, 31]: linear algebra, Hilbert spaces, Pauli matrices, density operators, quantum circuits, the stabilizer formalism.

- Category theory and diagrammatic calculi [52, 31, 63, 114]: monoidal categories, enrichment, compact closed categories, string diagrams, limits, colimits, ZX-calculus.

- Logic and proof theory [71]: propositional logic, natural deduction, semantic interpretations, structural induction.

Several of the fields on which this thesis builds are known for having a large amount of conflict in notation and nomenclature. For example, it is common to use *one-way non-signalling* to refer to either Definition 2.1.2 (no information can pass from Bob to Alice) or Definition 2.1.3 (factorising into local processes with Alice before Bob) since they are equivalent in quantum theory, though it is important in this work to disambiguate between definitions using distinct names in order to study them in a broader context. In addition to introducing the essential content for the unfamiliar, the provided background section will also clarify the terminology to be used throughout this thesis. As always, I appreciate the patience of readers in adapting to the notation and names used in this thesis where it differs from their preferences.

Chapter 2

Causal Structures in $\mathrm{Caus}[\mathcal{C}]$

The informal definition of causal consistency so far is just the requirement that composition preserves meaningful notions of normalisation, so it will very obviously depend on a number of factors of the specific model we look at:

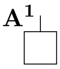
- What exactly are the processes of the model? E.g. functions, relations, linear maps, etc.
- What kinds of normalisation conditions should we consider?
- How do we encode processes of different shapes?
- How do we calculate the outcome of a composition?

This chapter is focussed on giving a detailed presentation of the models I chose to look at in this thesis.

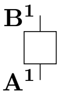
We find the language of *category theory* very useful when answering these questions about these models, since specifying a category immediately tells us the kinds of processes we have and how to compose them. With our main motivating examples coming from probability theory and quantum theory, we will mostly be working with categories where the morphisms are *linear maps* between (finite) vector spaces with the usual notions of sequential and parallel composition. Each setting may place some restrictions on which linear maps are valid, such as a requirement on the scalar field or positivity conditions - we will largely be abstracting away such details, making a few practical assumptions (see Definitions 2.3.1 and 2.4.1).

Taking a category of "unnormalised" processes, we can then define our normalisation conditions. Borrowing nomenclature from programming languages, each normalisation condition defines a *type*, characterised by the states/programs of that type. Types are not considered unique, so each morphism in the unnormalised category may admit multiple types, telling us that it satisfies multiple normalisation conditions simultaneously. The Caus $[-]$ construction [78] is one way to take such a category of unnormalised processes $\mathcal{C}$ and build a category Caus $[\mathcal{C}]$ of normalised/typed processes.

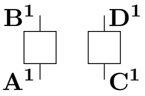
When we draw a shape for a process as in Equation 1.1, we are using the shape as a shorthand for the set of assumptions on relative positions of the interfaces in space and time, a.k.a. their *causal structure*. The quantum foundations literature has many well-established results characterising the processes consistent with a particular causal structure. We can adopt these characterising properties as our normalisation conditions in Caus $[\mathcal{C}]$ to give types for different causal structures. For example, the types below capture processes of the corresponding shapes in a first-order picture with time flowing upwards in the diagrams.



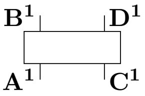
$: \mathbf{A^1}$



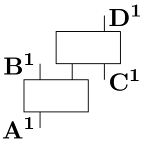
$: \mathbf{A^1} \multimap \mathbf{B^1}$ (2.1)



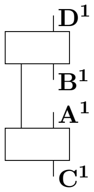
$: \left(\mathbf{A^1} \multimap \mathbf{B^1}\right) \otimes \left(\mathbf{C^1} \multimap \mathbf{D^1}\right)$



$: \left(\mathbf{A^1} \multimap \mathbf{B^1}\right) \parr \left(\mathbf{C^1} \multimap \mathbf{D^1}\right)$ (2.2)



$: \left(\mathbf{A^1} \multimap \mathbf{B^1}\right) < \left(\mathbf{C^1} \multimap \mathbf{D^1}\right)$



$: \left(\mathbf{A^1} \multimap \mathbf{B^1}\right) \multimap \left(\mathbf{C^1} \multimap \mathbf{D^1}\right)$ (2.3)

Throughout this chapter, it is assumed that the reader is familiar with category theory and linear algebra, and we begin with an overview of some key concepts of quantum causal structures to give a reference point for the constructions we use for them in Caus $[\mathcal{C}]$. Similarly, Section 2.2 provides context with a brief review of the literature of alternative categories for studying causal structures. As we are building directly on top of the work of Kissinger and Uijlen, 2.3 gives a brief summary of the definitions and highlight results from their paper [78] introducing the Caus $[-]$ construction.

The novel results begin in Section 2.4 which introduces some alternative assumptions on the base category that enable us to use more tools from linear algebra within the category theoretic framework. Each of 2.5-2.8 introduces new type constructors to Caus $[\mathcal{C}]$, increasing the variety of causal structures we can consider until we reach their direct embedding via *graph types*.

The chapter ends with a discussion section where we compare the $\mathrm{Caus}[-]$ construction to *effectus theory* [73, 25] - another set of categorical constructions which relate between unnormalised (a.k.a. *partial*) morphisms and normalised (*total*) morphisms - to look at ways to reintroduce unnormalised processes back into $\mathrm{Caus}[\mathcal{C}]$.

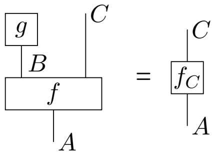
2.1 Background: Quantum Causal Structures

As per the Introduction, we will begin with the framework of *process theories* [31]: collections of processes with designated input and output system types and a means of composing processes in sequence and parallel. This is formalised in the definition of a (strict) symmetric monoidal category $(\mathcal{C}, \otimes, I)$.

We will interpret the processes as physical operations that transform the input system into the output system, with sequential composition implying that one operation happens after the other in time. In the equational theory, we unify processes when their physical operations are interchangeable without any observable difference in the way they relate the input and output system, i.e. we will ignore details on how the operations are done (the exact activities that occur in enacting the operation) to focus on the effect they have on the states they interact with.

Relativity constrains the transmission of information by the speed of light and, in particular, this means no information can flow into the past. Suppose we take a process $f : A \to B \otimes C$ with two outputs, and to B we choose to apply another physical process that runs to completion and leaves nothing left - that is, we compose with some $g : B \to I$. Since g happens after f, the state we observe at C immediately following f must be independent of the choice of morphism chosen for g. This gives the idea that there must be a unique marginal process $f_C : A \to C$ induced by f.

$$\forall f : A \to B \otimes C.\ \exists f_C : A \to C.\ \forall g : B \to I.$$



(2.4)

In the special case where $C = I$ is trivial, $A = B$, and $f : A \to A \otimes I \cong A$ is just the identity id_A, we have $f_C = f \fatsemi g = g$. This simplistic consequence of relativity implies that each object A in our process theory must have a unique effect, which we will denote as $\bar{\bar{\top}}_A : A \to I$ and refer to as the *discarding* process. For example, this would take the form of the summation operator in probability theory or the trace operator in quantum theory.

The uniqueness property fixes how discarding interacts with the monoidal structure.

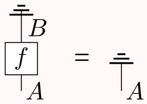
$$\bar{\top}_{A \otimes B} = \bar{\top}_A \otimes \bar{\top}_B \tag{2.5}$$

$$\bar{\top}_I = \mathrm{id}_I \tag{2.6}$$

It also constrains the kinds of other processes that can exist in the theory. For any $f : A \to B$, we can always consider discarding the output B. This composite $f \fatsemi \bar{\top}_B : A \to I$ is an effect of A, so by uniqueness it must be $\bar{\top}_A$. This shows that every process must be *discard-preserving*: if you discard all outputs of a process f, the result is indistinguishable from discarding the inputs without applying f at all. We will also refer to such processes as *(first-order) causal*[2].

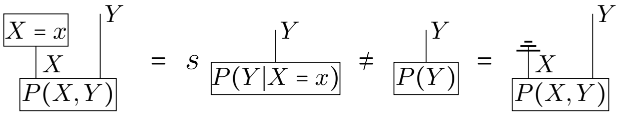
Definition 2.1.1: (First-order) causal [23, Definition 27, Lemma 5]

A *(first-order) causal* morphism $f \in \mathcal{C}(A \to B)$ is one that satisfies:



(2.7)

It is important to note that these constraints only apply when the process theory is describing operations that can be performed *reliably*, excluding operations that can only be performed with some probability of success. We may still have a rich meta-theory for analysis in which the things we describe are not physical operations. For example, conditional probabilities and postselection work by fixing some output to a particular value; they are interesting precisely because they can sometimes cause changes to other outputs when they are correlated, which would violate the relativistic information bounds if they could be performed reliably.

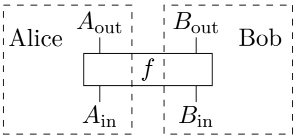


(2.8)

The causality principle refers to the result originally from Chiribella, D'Ariano, & Perinotti [23] stating that the uniqueness of effects is necessary and sufficient for a theory to satisfy "no-signalling from the future". The uniqueness of effects may also be referred to as *terminality*, since it makes the unit object I a terminal object in the category $\mathcal{C}$.

[2] The term "first-order" here comes from the usage in programming terms referring to elementary data, distinguishing it from `higher-order` content which acts on functions, and has no intended connection to first-order logics.

Returning to relativity, let us now consider a setup with two parties, Alice and Bob, who each have access to one side of a shared process.

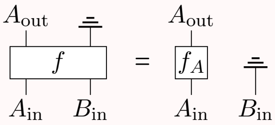


(2.9)

Depending on the space-time relationship between Alice and Bob, compatibility with relativity will mean that not every process of type $A_{\text{in}} \otimes B_{\text{in}} \to A_{\text{out}} \otimes B_{\text{out}}$ will be realisable. Suppose first that Alice and Bob are time-like separated with Alice before Bob. Excluding the possibility of closed time-like curves, this means there is no path that light could possibly take from Bob to Alice and hence no information can be sent in this direction - when we marginalise out Bob's output, the marginal process at Alice must be independent of Bob's input, i.e. equivalent to discarding Bob's input and applying some local process at Alice.

Definition 2.1.2: One-way (non-)signalling [12][3]

A bipartite process $f : A_{\text{in}} \otimes B_{\text{in}} \to A_{\text{out}} \otimes B_{\text{out}}$ is *non-signalling from* $(B_{\text{in}}, B_{\text{out}})$ *to* $(A_{\text{in}}, A_{\text{out}})$ if there exists some $f_A : A_{\text{in}} \to A_{\text{out}}$ such that:



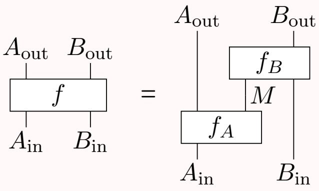
(2.10)

Alternatively, we could imagine the background space-time as divided into small regions, each with a local process and wiring up two processes corresponds to a physical system leaving one region and entering the next. Relativity then constrains these physical systems to only move along time-like curves. f must now factorise into a local process occurring with Alice, which may pass some intermediate system to a local process occurring with Bob.

Definition 2.1.3: Semi-localisability [12]

A bipartite process $f : A_{\text{in}} \otimes B_{\text{in}} \to A_{\text{out}} \otimes B_{\text{out}}$ is *semi-localisable* if there exist some system M and processes $f_A : A_{\text{in}} \to A_{\text{out}} \otimes M$, $f_B : M \otimes B_{\text{in}} \to B_{\text{out}}$ such that:

[3] These have historically also been called *semi-causal* processes [12, 49].



(2.11)

If we change the relationship so that Alice and Bob are now space-like separated, there is no time-like path between them in either direction. Definition 2.1.2 generalises by symmetrising the non-signalling condition.

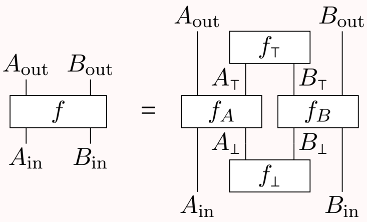
Definition 2.1.4: Non-signalling [12][4]

A bipartite process $f : A_{\text{in}} \otimes B_{\text{in}} \to A_{\text{out}} \otimes B_{\text{out}}$ is *non-signalling* if it is both (one-way) non-signalling from $(A_{\text{in}}, A_{\text{out}})$ to $(B_{\text{in}}, B_{\text{out}})$ and vice versa.

Generalising Definition 2.1.3, the lack of time-like path between Alice and Bob would mean that there is no physical system passing from one of their local processes to the other. However, it is possible that their past light-cones will intersect each other, so we must consider an additional process to describe any shared history that might set up correlations between Alice and Bob, as well as a shared future when their future light cones intersect.

Definition 2.1.5: Localisability [12]

A bipartite process $f : A_{\text{in}} \otimes B_{\text{in}} \to A_{\text{out}} \otimes B_{\text{out}}$ is *localisable* if there exist some systems $A_\perp, A_\top, B_\perp, B_\top$ and processes $f_\perp$, f_A, f_B, $f_\top$ such that:



(2.12)

For a bipartite scenario, the only configurations we need to consider are time-like separation (in either direction) and space-like separation, but this number increases greatly as we consider scenarios with more parties. A *causal structure* abstractly

[4] Non-signalling processes were also historically referred to simply as *causal* [12], though this conflicts with Definition 2.1.1.

describes the configuration of parties by the existence of time-like paths between each pair.

Definition 2.1.6: (Definite) Causal Structure [80]

A *causal structure* over a set V of points is a partial order $\leq \subseteq V \times V$.

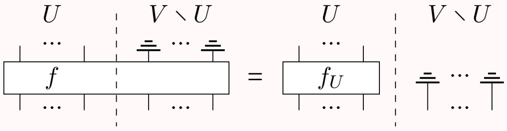
The use of a partial order here assumes that the background spacetime does not admit any closed timelike curves which are beyond the scope of this thesis, though one could just as easily consider preorders instead.

Our previous notions of non-signalling and localisability conditions generalise straightforwardly to arbitrary causal structures.

Definition 2.1.7: Signal-consistency [22]

Given a causal structure $\leq \subseteq V \times V$, a set $U \subseteq V$ is *down-closed* if $\forall u, v \in V. v \in U \wedge u \leq v \Rightarrow u \in U$.

A multi-partite process $f : \bigotimes_{v \in V} A_v \to \bigotimes_{v \in V} A'_v$ is *signal-consistent*[5] with respect to a causal structure $\leq \subseteq V \times V$ if, for every down-closed set $U \subseteq V$ there exists some $f_U : \bigotimes_{v \in U} A_v \to \bigotimes_{v \in U} A'_v$ such that:



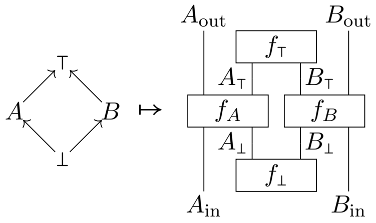
(2.13)

Definition 2.1.8: Causal Realisability [22]

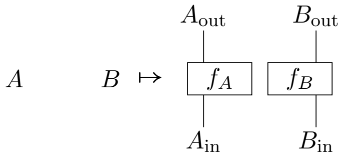
A multi-partite process $f : \bigotimes_{v \in V} A_v \to \bigotimes_{v \in V} A'_v$ is *causally realisable*[5] with respect to a causal structure $\leq \subseteq V \times V$ if there exists a set $\{M_{u,v}\}_{(u,v) \in \leq}$ of intermediate systems and a set $\{f_v : A_v \otimes \bigotimes_{u \leq v} M_{u,v} \to A'_v \otimes \bigotimes_{w \geq v} M_{v,w}\}_{v \in V}$ of processes such that composing along the intermediate systems recovers f.

It is common to add dummy points to a causal structure to give a unique global minimum $\bot$ and maximum $\top$ when we wish to account for actions in a shared past or future. This distinguishes Definition 2.1.5 for localisability from separability of a bipartite channel (i.e. when it factorises as a tensor product of local channels without any connections) as causal realisability for the different causal structures shown below.

[5] The names for these two concepts are not standardised across the literature at the moment, both often being used interchangeably as definitions of compatibility with causal structure. In the cited example [22], these properties are referred to as *causally ordered channels* versus *channels with memory*. The particular names in the definitions here are owed to Tein van der Lugt from discussions on his research where it is also important to distinguish between the definitions.



(2.14)



(2.15)

When uniqueness of effects is assumed, the dummy maximum ⊤ can be omitted, since the process there must be discarding which factorises (Equation 2.5) and can be incorporated into the predecessor processes.

A more general form of the causality principle states that the uniqueness of effects is necessary and sufficient for all causally realisable processes to be signal-consistent [77, 29]. Even in theories with unique effects, there may exist processes that are signal-consistent but lack any realisable decomposition, such as the PR-box which is non-signalling but not localisable [98]. In this case, the following alternative characterisation of non-signalling processes holds in finite-dimensional quantum theory [60, 24], or indeed in any Generalised Probabilistic Theory (GPT) with local tomography [21].

Theorem 2.1.9: [21, Theorem 5.1]

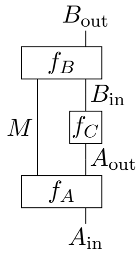
In any GPT with local tomography, any non-signalling first-order causal process can be written as an affine combination of separable first-order causal processes.

This theorem is of particular importance here since we will directly be generalising it later in the thesis to drop the assumption of first-order processes (the Non-signalling Theorem), as well as generalising it to arbitrary causal structures (the Graph Equivalence Theorem).

It is an open problem in the field to give a full characterisation of those causal structures in which signal-consistency and causal realisability coincide exactly. For example, in both classical and quantum theory it is known that the two conditions are equivalent for linear causal structures (those where the points are totally ordered) [22], in particular one-way non-signalling is equivalent to semi-localisability in the bipartite case [23].

Processes compatible with linear causal structures admit an equivalent characterisation as a certain class of higher-order processes called *combs*, though their definition requires us to introduce some tools for representing higher-order processes.

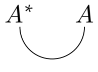
Suppose Alice and Bob share a semi-localisable process, i.e. a pair of local processes f_A and f_B connected by some memory channel M, and suppose that Charlie lies on some time-like path between Alice and Bob. Since f_A must be completed before f_B begins, we could imagine a situation in which Alice's output is passed to Charlie who uses it to prepare Bob's input.

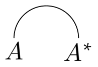


(2.16)

In this way, we can view the combination of f_A and f_B as a higher-order transformation, mapping any first-order process $f_C : A_{\text{out}} \to B_{\text{in}}$ to a new first-order process of type $A_{\text{in}} \to B_{\text{out}}$. However, we cannot always internalise such higher-order transformations as morphisms of the symmetric monoidal category. The closest we can achieve is the form in Equation 2.11 using a morphism $A_{\text{in}} \otimes B_{\text{in}} \to A_{\text{out}} \otimes B_{\text{out}}$, but then the limitation to sequential and parallel composition means we cannot compose this with f_C to obtain a morphism of type $A_{\text{in}} \to B_{\text{out}}$. We would need some means of connecting an output back round to an input in order to make this composition representable.

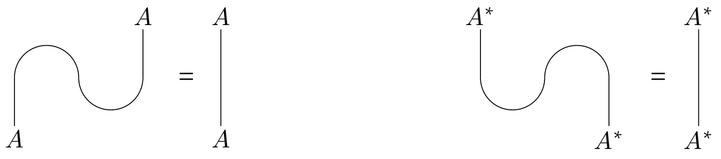
Suppose, then, that our category of processes is *compact closed*: each object A has a dual A^*, and "cup" and "cap" morphisms

$\eta_A :=$ 

$\epsilon_A :=$ 

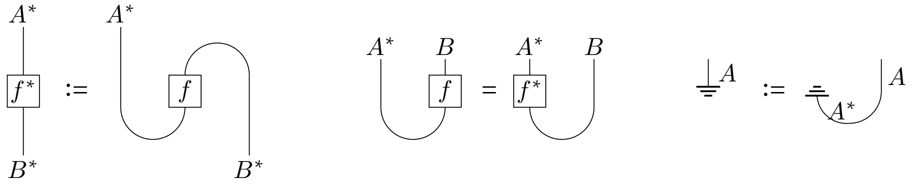
(2.17)

that satisfy the following "yanking" equations:



(2.18)

Drawing these morphisms as curved identity wires is convenient for demonstrating how they permit any morphism f on one side to slide over to the other much like naturality of an identity, up to the transposition into f^*. This specialises to mapping between states and effects, such as transposing the discard map ⊤ to give a *uniform state* ⏚ for any object.



(2.19)

When sliding multiple morphisms in this way, their order reverses (i.e. transposition is a contravariant functor $(-)^* : \mathcal{C}^{\mathrm{op}} \to \mathcal{C}$). This allows us to dissociate from the idea of a canonical direction of time in our diagrams, or the distinction of input and output roles. One can formalise this time-reversing sliding behaviour in the form of extranatural transformations, generalising the time-directed sense of naturality.

Definition 2.1.10: Extranatural Transformation [50]

Given functors $F : \mathcal{A} \times \mathcal{B} \times \mathcal{B}^{\mathrm{op}} \to \mathcal{D}$ and $G : \mathcal{A} \times \mathcal{C} \times \mathcal{C}^{\mathrm{op}} \to \mathcal{D}$, a family of morphisms

$$\{\alpha_{A,B,C} : F(A,B,B) \to G(A,C,C)\}_{A\in\mathrm{Ob}(\mathcal{A}),B\in\mathrm{Ob}(\mathcal{B}),C\in\mathrm{Ob}(\mathcal{C})} \tag{2.20}$$

is *natural* in A and *extranatural*[6] in B and C, if for all $f \in \mathcal{A}(A,A')$, $g : \mathcal{B}(B,B')$, $h : \mathcal{C}(C,C')$:

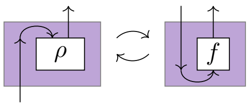
$$F(f,\mathrm{id},\mathrm{id}) \fatsemi \alpha_{A',B,C} = \alpha_{A,B,C} \fatsemi G(f,\mathrm{id},\mathrm{id}) \tag{2.21}$$

$$F(\mathrm{id},g,\mathrm{id}) \fatsemi \alpha_{A,B',C} = F(\mathrm{id},\mathrm{id},g) \fatsemi \alpha_{A,B,C} \tag{2.22}$$

$$\alpha_{A,B,C} \fatsemi G(\mathrm{id},h,\mathrm{id}) = \alpha_{A,B,C'} \fatsemi G(\mathrm{id},\mathrm{id},h) \tag{2.23}$$

When useful, we may draw directional arrow markings on the wires to reinforce an intuitive temporal reading of the diagram that reflects the reordering under duality - i.e. arrows pointing up for wires of A and down for wires of A^*.

For finite-dimensional Hilbert spaces, the cup morphisms are a choice of maximally entangled states (e.g. Bell states) and the caps are corresponding Bell effects, giving rise to the Choi-Jamiołkowski isomorphism [26, 74] between maps $A \to B$ and states $I \to A^* \otimes B$ - it is common to refer to such a state as the *Choi operator* of the corresponding process.

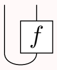


(2.24)

[6] These are a special case of *dinatural transformations* [43] where F and G can both co- and contravariantly depend on the same category $F, G : \mathcal{C} \times \mathcal{C}^{\mathrm{op}} \to \mathcal{D}$. Dinaturality is important in logic [16], but there is no general way to compose them.

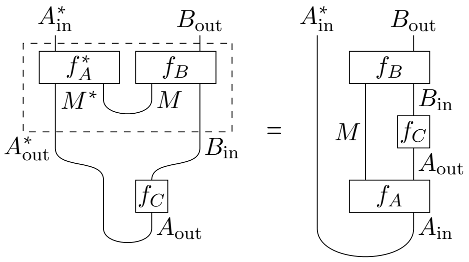
Definition 2.1.11: Choi operator [26]

The *Choi operator* of a linear map $f : A \to B$ is the state



$$: I \to A^* \otimes B \tag{2.25}$$

Returning to our example from Equation 2.16, if we encode f_C as a state of $A^*_{\text{out}} \otimes B_{\text{in}}$ in this way, we can then represent the higher-order transformation as a morphism of type $A^*_{\text{out}} \otimes B_{\text{in}} \to A^*_{\text{in}} \otimes B_{\text{out}}$.



(2.26)

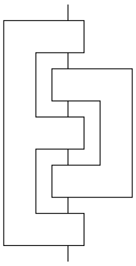
We could similarly encode this transformation as a state of type $A^*_{\text{in}} \otimes A_{\text{out}} \otimes B^*_{\text{in}} \otimes B_{\text{out}}$ and iterate to obtain representations of an infinite hierarchy of higher-order transformations. As mentioned in the Introduction, the flexibility of compact closure permits many more ways to compose these process encodings than via just sequential, parallel, and higher-order composition, such as the example in Equation 1.3.

Starting with the first-order causal processes, the hierarchy of combs defines a class of higher-order processes that preserve this property inductively.

Definition 2.1.12: Comb [22, Definition 4]

A 1-*comb* on (A_0, A_1) is the Choi operator $I \to A^*_0 \otimes A_1$ of a first-order causal morphism. For $N \geq 2$, an N-*comb* on $(A_0, \dots, A_{2N-1})$ is the Choi operator $I \to \bigotimes_{i<N} A^*_{2i} \otimes A_{2i+1}$ of a morphism transforming $(N-1)$-combs on $(A_1, \dots, A_{2N-2})$ into 1-combs on (A_0, A_{2N-1}).

Returning to causal structures, N-combs on $(A_0, A'_0, \dots, A_{N-1}, A'_{N-1})$ are precisely characterised as the Choi operators of processes $\bigotimes_{i<N} A_i \to \bigotimes_{i<N} A'_i$ which are signal-consistent (equiv. causally realisable) with respect to the linear causal structure $0 \leq \cdots \leq N-1$ [22]. This result gives rise to the diagrammatic notation used for combs as an ordered sequence of connected blocks (resembling the teeth of a hair comb), or equivalently a box with a sequence of holes in it into which we can place other combs.



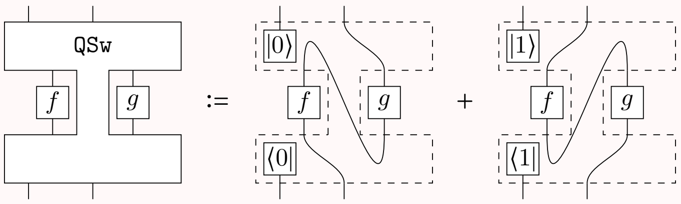
(2.27)

Looking back at the kind of diagrams from Equation 1.1 in the Introduction, we now have the tools to be able to combine a collection of first-order processes and wirings between them into an abstract shape describing the overall process with some internal causal structure between its interfaces. A key interest in quantum causality is to question what is permissible when we similarly abstract away the connections between our processes, i.e. the causal ordering of the elementary processes is not fixed but may itself exhibit quantum effects.

The standard example of this is to exploit quantum gravity to make light travel in a superposition of paths which pass through some points in different orders. Suppose a large mass can be placed into a superposition of distinct positions in space, then the induced gravitational field and subsequent space-time geometry around it will also be in a superposition of configurations. The path between two points of interest may be time-like in one configuration and space-like or time-like in the reverse order in another configuration. This is summarised by the *quantum switch* [24].

Definition 2.1.13: Quantum Switch [24, Equation 6]

The *quantum switch* is the higher-order process QSw transforming two first-order causal processes $f, g : A \to A$ into $\texttt{QSw}(f, g) : \mathbb{C}^2 \otimes A \to \mathbb{C}^2 \otimes A$, defined by its action on unitaries as:



(2.28)

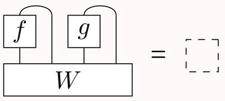
One can imagine alternative versions of this where the control system is classical or a fixed probabilistic mixture of the orderings, or where we are switching over the orderings of more than two processes. The switch is incompatible with any definite causal structure in the sense that there is no process with definite causal structure into which a single instance of f and g can be inserted to produce $\texttt{QSw}(f, g)$ [24]. This incompatibility can be stratified further to talk about processes with *random causal structure* (they can be expressed as a convex mixture of processes with different definite causal structures), *dynamic causal structure* (the causal ordering is determined by the value of some input state), or *indefinite causal structure* (incompatible with either of the previous classes, shown by violating an inequality satisfied by all processes with definite causal structure [96]).

Removing the assumption of definite causal structure, we instead represent the manner in which all processes are composed in a given scenario by a *process matrix* [96]. This is a higher-order process acting on the collection of process instances to be placed in our scenario and were constructed specifically to study causal structures in quantum foundations. Wlog, we assume that the overall scenario is *closed* (there are no remaining inputs or output), so the resulting morphism is a scalar. In line with the Born rule of quantum mechanics, the scalar gives the probability with which the closed scenario could be successfully performed. With our interest in operations that can be performed reliably, the process matrix must specifically send all inputs to the scalar 1 (represented by the empty diagram).

$$:= \mathrm{id}_I \tag{2.29}$$

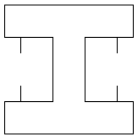
Definition 2.1.14: Process Matrix [96]

A (bipartite) *process matrix* is a state $W : I \to A_0 \otimes A_0'^* \otimes A_1 \otimes A_1'^*$ such that for all first-order causal $f : A_0 \to A_0'$, $g : A_1 \to A_1'$ it satisfies:



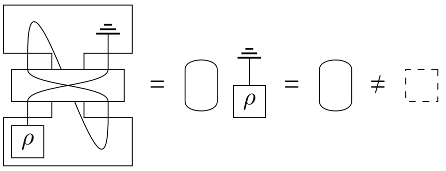
(2.30)

From their usage, we also get the typical diagrammatic presentation for process matrices as:



(2.31)

The separation of the two holes reinforces that it cannot be applied to any bipartite process - it is only required to guarantee normalisation on separable processes, though by linearity this also holds for any affine combination of them (i.e. non-signalling processes). The same applies to the quantum switch. For example, consider the simple process matrix that applies the inputs in a fixed order to some initial state and discards the result. If we then apply this to a swap, the resulting scalar corresponds to the dimension of the system.

 (2.32)

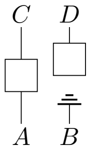
2.2 Related Work: Categories for Causality

This section will present a brief literature review of frameworks for studying quantum causality within the language of category theory, with a critical focus to motivate the solution of the Caus [−] construction in Section 2.3.

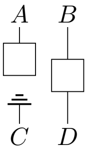
In the causal categories of Coecke and Lal [32], objects are treated as space-like slices of a scenario within a first-order picture of causality - that is, in diagrams with a canonical direction of time, a cut through the wires of the diagram such that none of the cut wires have a time-directed path between them; this cut could feasibly represent a view of the system at some fixed point in time. Since the systems are all simultaneously available, morphisms out of a given object can describe generic circuits where all systems can interact with one another. Given two space-like curves, there is no guarantee that they combine to give something space-like, so they consider categories with *partial monoidal structures*. Causality itself is imposed by terminality of the tensor unit.

Morphisms of $A \to B$ don't necessarily imply a time-like dependence of A before B, instead they merely describe a functional relation between their states - when there is no time-like path from A to B, there still exist morphisms $A \to B$ but they will be constant, i.e. factorise via the unit. This induces a causal structure over the objects, where A has no causal influence on B if every morphism $A \to B$ disconnects in this way. To make sure monoidal products of objects act like space-like slices, we require that $A \otimes B$ exists iff A and B are causally independent.

Within such causal categories, there may exist objects that are *causally intertwined* in the sense that there exist non-separating morphisms in both directions. The authors gave a helpful intuition for such objects as *crossed spatial slices*: suppose A causally precedes C and D precedes B, then $A \otimes B$ and $C \otimes D$ can both be space-like curves but neither causally precedes the other since we have morphisms of the following forms:



$: A \otimes B \to C \otimes D$



$: C \otimes D \to A \otimes B$ (2.33)

Using this definition of a causal category works great when considering a scenario with a fixed background space-time that is known in advance, but we find this too heavy of an assumption for exploring quantum causality in settings that permit indefinite causal structure. The restriction to considering morphisms between space-like slices greatly restricts us to first-order processes; for example, the four interfaces to a 2-comb necessarily carry a linear causal structure, meaning it is impossible to express a comb as a morphism between two sets of causally independent systems. The same criticism here can be applied to other frameworks with terminality as a key feature including the following:

- Markov categories [53] abstract notions of probability theory in a way that permits convenient and intuitive diagrammatic presentations. Each object is equipped with a commutative comonoid for copying/broadcasting and deleting.

- Quantum/involutive Markov categories [97, 54] generalise this for quantum information, taking features of C*-algebras to describe copying and deleting with respect to a basis. However, these copy and delete morphisms in general will fail to be causal processes, so whilst it gives a convenient notational framework in which we can study quantum information, one has to manually assert causality of morphisms under consideration.

- Effectus theory [25] describes categories with coproducts + and a terminal object 1 where the morphisms represent "total" morphisms (i.e. those that can be performed without failure) from which the Kleisli category of the Maybe monad $((-) + 1)$ contains "partial" morphisms (i.e. those that can be performed with some chance of error). Monoidal effectuses require the monoidal unit to be terminal, making the total morphisms correspond to first-order causal processes. We will address effectuses in more detail in Section 2.9 where we discuss a few different ways to apply the same principles to higher-order processes.

Whilst not expressed in the language of category theory, Bisio and Perinotti's higher-order operational theory for quantum processes [15] was devised independently of Caus$[-]$ construction but amounts to a particular subcategory of Caus$[\mathrm{CP}^*]$

(where CP^* is the category of finite-dimensional C^* algebras and completely-positive maps) relevant for describing physical scenarios. The types of the theory are inductively defined from finite-dimensional Hilbert spaces, populated by density matrices, and higher-order transformation types for Choi operators of channels that map elements of the input type to those of the output type. Dual types and two monoidal structures exist on top of this as derived structures.

Each space is characterised as the direct sum of a fixed scalar of the identity matrix (i.e. the uniform state) and a subspace of traceless operators. They show that, for each type constructor $T[A, B, \ldots]$ in the theory, the traceless subspace of $T[A, B, \ldots]$ always decomposes by direct sum into products of the identity and traceless subspaces of $A, B, \ldots$, with the particular set of product combinations used giving a unique characterisation of the type.

This framework was extended by Hoffreumon and Oreshkov [64], adding new operators for a one-way signalling product, union, and intersection of types, each matching the corresponding operator in $\mathrm{Caus}[\mathrm{CP}^*]$ but similarly developed independently. They also reformulated the framework to characterise each space by a projection and a fixed trace. Every type in their framework can be expressed canonically as a union of intersections of linear causal structures over the elementary types, following from distributivity laws between union, intersection, and the one-way product (though such distributivity between union and intersection does not hold in general in the full category $\mathrm{Caus}[\mathrm{CP}^*]$, see 2.7.8).

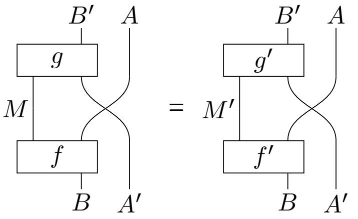
Whilst expressed in terms of quantum theory in their papers, one could very easily generalise it to other base physical theories to match the $\mathrm{Caus}[-]$ construction more broadly. It also achieves many of the goals we want, presenting a unified theory of higher-order processes including those with indefinite causal structure. The types built inductively by combining first-order spaces are relevant from the perspective of studying physical scenarios, though restricting to this can omit non-physical systems that may still be useful to study from a computational perspective (see Definition 4.4.1).

The $\mathrm{Caus}[-]$ construction itself is by no means a perfect framework for studying causal structures and higher-order processes, mostly limited by the heavy assumptions made on the base category meaning we exclude a number of interesting theories we would like to be able to study (discussed in Section 2.4.1). We will finish this section by reviewing some recent work that has aimed to recreate a similar categorical framework for higher-order physical processes that work with an arbitrary symmetric monoidal category.

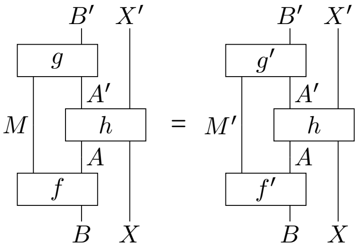
The most significant assumption to overcome is the need for the base category to be compact closed to represent higher-order processes with Choi operators. Looking for a generic alternative to represent, for instance, a 2-comb on (B, A, A', B'), we could start with functions between the homsets $\mathcal{C}(A, A') \to \mathcal{C}(B, B')$. It may be the case that the process $A \to A'$ actually has some additional inputs and outputs that it exchanges with the environment on which the 2-comb should have no effect, so we instead need families of functions $\{\mathcal{C}(A \otimes X, A' \otimes X') \to \mathcal{C}(B \otimes X, B' \otimes X')\}_{X, X' \in \mathrm{Ob}(\mathcal{C})}$ related by commutation with actions on the extension systems. Wilson, Chiribella, and Kissinger call such families of functions *locally-applicable transformations* [119], showing that they are equivalent to 2-combs for finite-dimensional quantum theory. A *strongly* locally-applicable transformation is additionally required to commute with any other locally-applicable transformation applied on the extension system, which is then enough to recover 2-combs for the category of unitary maps [118]. These now form a symmetric monoidal category and their multi-partite generalisation (acting on separable inputs, as seen with process matrices or the quantum switch) form a polycategory.

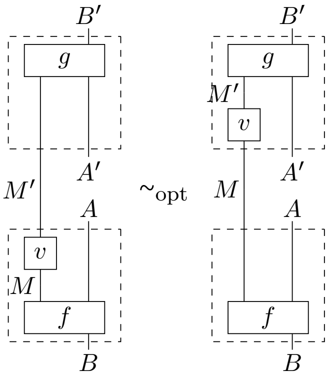
Starting with arbitrary functions can lead to (strongly) locally-applicable transformations which aren't representable as pre- and post-composition with a pair of processes as in Equation 2.16. The alternative is to build a category of 2-combs out of such pairs to guarantee, by construction, that everything we handle is representable. Hefford and Comfort [61] consider quotienting these by the equivalence relations below.

$$(f, g) \sim_\sigma (f', g') \iff \tag{2.34}$$



$$(f, g) \sim_{\text{comb}} (f', g') \iff \begin{array}{c} \forall X, X'. \\ \forall h : A \otimes X \to A' \otimes X'. \end{array} \tag{2.35}$$





(2.36)

These need not coincide in general, though we always have $\sim_{\text{opt}} \subseteq \sim_{\text{comb}} \subseteq \sim_{\sigma}$ and they are guaranteed to coincide for compact closed categories. Equivalence of Choi operators is akin to $\sim_{\sigma}$, though this need not be a congruence with respect to nesting of combs which makes it unsuitable for building a category of combs. The others give rise to categories $\text{Optic}\left(\mathcal{C}\right)$ (the category of coend optics) and $\text{Comb}\left(\mathcal{C}\right)$ which similarly generalise to polycategories for multi-partite variants.

An ultimate meta-theory for higher-order causal structure would allow us to mix and match definitions of structures based on externally observable properties like signal-consistency or locally-applicable transformations, and definitions based on decompositions like causal realisability or optics, thus allowing us to study both what happens in settings when they are equivalent and when distinct. The recent work of Hefford and Wilson [62] achieves this in the category $\text{StProf}\left(\mathcal{C}\right)$ of strong endo-profunctors on a monoidal category $\mathcal{C}$ and strong natural transformations between them. For example, locally-applicable transformations are strong natural transformations $\mathcal{C}\left(A \otimes -, A' \otimes =\right) \Rightarrow \mathcal{C}\left(B \otimes -, B' \otimes =\right)$, and optics are $\mathcal{C}\left(B, -\right) \times \mathcal{C}\left(B', =\right) \Rightarrow \mathcal{C}\left(A, -\right) \times \mathcal{C}\left(A', =\right)$ following the Yoneda embedding.

$\text{StProf}\left(\mathcal{C}\right)$ admits two monoidal structures resembling semi-localisability and separability, with the appropriate interchange law to make it a closed normal duoidal category (matching the same for $\text{Caus}\left[\mathcal{C}\right]$, see Proposition 2.6.21). These being closed monoidal induces a weak duality on objects. A highlight result from their work shows that locally-applicable transformations and optics are equivalent iff the weak duality is involutive on profunctors of the form $\mathcal{C}\left(A, -\right) \times \mathcal{C}\left(A', =\right)$, drawing a striking relation between the logical content of a theory and its physical properties in terms of decomposition theorems. Comparing this logic to that of $\text{Caus}\left[\mathcal{C}\right]$, it is possible to define an analogue of $\parr$, but it is no longer guaranteed to be associative or unital (and hence not a functor). Under the conjecture that this correspondence between logical and decompositional structure extends, we can view $\text{Caus}\left[\mathcal{C}\right]$ as an extreme case with very strong and general decomposition theorems (such as the Graph Equivalence Theorem) which grants it such a rich logic.

Where the $\mathrm{Caus}[-]$ construction conjures an inextricable link between the origin of causal structures and preservation of normalisation conditions, $\mathrm{StProf}(-)$ provides causal structures without links to normalisation, though it may only be truly meaningful when the base category is interpreted as precisely the first-order processes. Promising future directions to investigate the overlap involve applying the profunctor constructions to suitably enriched base categories to express non-signalling conditions and compare them to the existing causal structures on optics.

2.3 Background: the $\mathrm{Caus}[-]$ Construction

Given a category $\mathcal{C}$ describing an underlying physical theory, one can summarise the causal category $\mathrm{Caus}[\mathcal{C}]$ as a process theory of higher-order causal processes through which interpretations of causal structures emerge. After motivating the goals of the construction, this section will give a brief summary of the essential material from the original paper by Kissinger and Uijlen [78], presenting the construction itself, some basic structural operators, and results relating them to causal structures.

2.3.1 Deriving the Construction

The identity of the $\mathrm{Caus}[-]$ construction within the landscape of higher-order process theories can be characterised by the following three goals:

1. Theory-independence/abstraction - being able to reuse as much as possible between the study of different theories really helps to understand what properties are truly fundamental versus determined by the specifics of a given theory. Physical motivations drive us to at least want versions for both classical probability theory and (finite-dimensional) quantum theory.

2. Causally consistent processes - each morphism should describe some process that could be performed in a lab with probability 1, i.e. we guarantee that every closed diagram evaluates to the unit scalar. Each process may be subject to different normalisation conditions dependent on how we are interpreting it, which we will describe as the process being *causal* for a given type. Moreover, we should consider maximal sets of causal processes; that is, if we wish to know whether a state ρ matches a type $\mathbf{A}$, we can enumerate all causally consistent

closed diagrams featuring a black box of type **A** and check that plugging ρ in gives 1 in every case.

3. Freely compositional - we wish to impose minimal a priori constraints on how processes may be composed. In particular, this means we should support as many string diagrams as possible without causal consistency. Having open diagrams with composition only limited by the matching of interfaces means we can make reasonable abstractions such as grouping the operations within a protocol into a single process for each participant, or combining complex, interleaved interactions between physical processes occurring over non-convex regions of spacetime into a single interaction.

For Goal 1, we start with a base category $\mathcal{C}$ of unnormalised processes within which we will work (expecting a forgetful functor $\mathrm{Caus}[\mathcal{C}] \to \mathcal{C}$). We require $\mathcal{C}$ to be compact-closed to support the string diagrams imposed by Goal 3. Setting the standard examples of $\mathrm{Mat}[\mathbb{R}^+]$ to investigate finite-dimensional classical probability theory and CP^* for quantum theory, Definition 2.3.1 abstractly presents the properties of these categories assumed in [78] [7].

Definition 2.3.1: Precausal category [78, Definition 3.1]

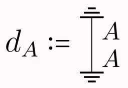
A compact closed category $\mathcal{C}$ is a *precausal category* if:

PC1. $\mathcal{C}$ has a discarding process $\stackrel{=}{\top}_A \in \mathcal{C}(A, I)$ for every object A, compatible with the monoidal structure as below;

$$\stackrel{=}{\top}_{A\otimes B} = \stackrel{=}{\top}_A \otimes \stackrel{=}{\top}_B \tag{2.37}$$

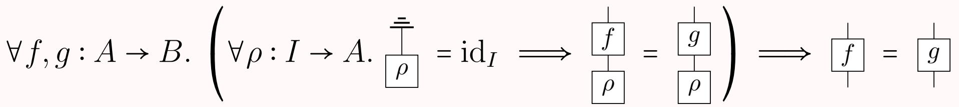
$$\stackrel{=}{\top}_I = \mathrm{id}_I \tag{2.38}$$

PC2. The scalar[8] d_A is invertible for all non-zero A;



(2.39)

PC3. $\mathcal{C}$ has enough causal states,

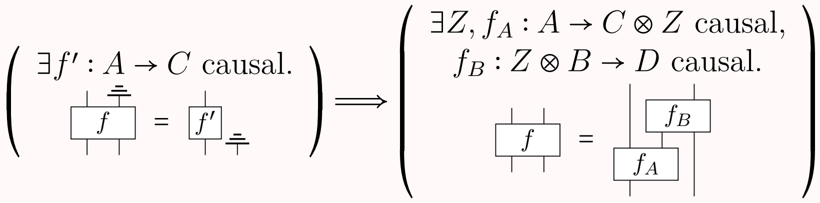


(2.40)

[7] Note that in [78], PC4 and PC5 are rolled into a single axiom, which is then proven equivalent to PC4 and PC5.

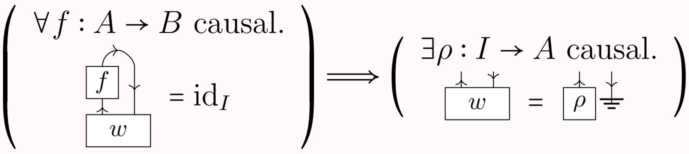
[8] This scalar is often called the "dimension", though even when the scalar d_A can be interpreted as an integer it may not match the cardinality of a basis for A. For example, $\dim(\mathcal{L}(\mathbb{C}^n)) = d^2_{\mathcal{L}(\mathbb{C}^n)}$ in CP, but $\dim(N) = d_N$ in $\mathrm{Mat}[\mathbb{R}^+]$.

PC4. First-order causal one-way signalling processes are semi-localisable: for any causal $f : A \otimes B \to C \otimes D$,



$$\left(\exists f' : A \to C \text{ causal.} \right) \Longrightarrow \left(\begin{array}{c} \exists Z, f_A : A \to C \otimes Z \text{ causal}, \\ f_B : Z \otimes B \to D \text{ causal.} \end{array} \right) \qquad (2.41)$$

PC5. For all $w : I \to A \otimes B^*$:



$$\left(\forall f : A \to B \text{ causal.} \quad = \mathrm{id}_I \right) \Longrightarrow \left(\exists \rho : I \to A \text{ causal.} \right) \qquad (2.42)$$

Example 2.3.2: Classical probability theory

Mat[$\mathbb{R}^+$] is a precausal category, whose objects are natural numbers and whose morphisms $M : m \to n$ are $n \times m$ matrices. Then, $\otimes$ is given by tensor product (a.k.a. Kronecker product) of matrices and consequently the tensor unit is the natural number 1. Hence, states are column vectors, discarding maps $\overset{=}{\top}_m : m \to 1$ are given by row vectors of all 1's, and the causality condition for states $\rho \fatsemi \overset{=}{\top} = \mathrm{id}_1$ imposes the condition that the entries of ρ sum to 1. The conditions PC1, PC2, and PC3 are easily checked, whereas PC4 and PC5 follow from the product rule for conditional probability distributions (see [78]).

Example 2.3.3: Finite-dimensional quantum theory

CP is a precausal category, whose objects are algebras $\mathcal{L}(H)$ of linear operators from a finite-dimensional Hilbert space to itself, and whose morphisms are completely positive maps (CP-maps). $\otimes$ is given by tensor product and consequently the tensor unit is the 1D algebra $\mathcal{L}(\mathbb{C}) \cong \mathbb{C}$. States are CP-maps $\mathbb{C} \to \mathcal{L}(H)$, which correspond to positive semidefinite operators $\rho \in \mathcal{L}(H)$. Discarding is given by the trace, hence causal states are the trace-1 positive semidefinite operators, a.k.a. quantum (mixed) states. Again the conditions PC1, PC2, and PC3 are easily checked, whereas PC4 and PC5 follow from the essential uniqueness of purification for CP-maps (see [78]).

Let's return to the remaining goal (2) which requires us to limit closed diagrams to the causally consistent ones. Given a closed string diagram featuring a single black box, we can always deform the wires to split the diagram into the black box and the *context*; for example, if the black box is a state $I \to A$, then we can always frame the rest of the closed diagram as an effect $A \to I$.

Definition 2.3.4: Causal scenarios

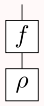
A *closed scenario* is a process description with no open systems (i.e. a scalar), and a *context* for a morphism f describes a diagram into which f can be placed to form a closed scenario.

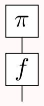
For any state $\rho \in \mathcal{C}(I, A)$ and effect $\pi \in \mathcal{C}(A, I)$, we can interpret the scalar $\rho \fatsemi \pi \in \mathcal{C}(I, I)$ as the (abstract) probability with which the closed scenario represented by this diagram can be realised, in the vain of the Born rule for quantum theory. When the scalar is id_I, we say that the scenario described by the composite diagram is *causally consistent*, or that ρ is *causal in the context of* π (and vice versa).

Double-glueing along a homfunctor [72] provides an all-purpose construction for enforcing generic normalisation conditions such as causal consistency.

Definition 2.3.5: Double-glueing along a homfunctor (simplified) [72]

Given a symmetric monoidal closed $\mathcal{C}$, the glued category $\mathbf{G}_I(\mathcal{C})$ has objects of the form $\mathbf{A} = (A \in \mathrm{Ob}(\mathcal{C}), c_{\mathbf{A}} \subseteq \mathcal{C}(I, A), d_{\mathbf{A}} \subseteq \mathcal{C}(A, I))$ and morphisms $f : \mathbf{A} \to \mathbf{B}$ are some $f \in \mathcal{C}(A, B)$ which preserve both the chosen states and effects:

$\forall \rho \in c_{\mathbf{A}} \;.$  $\in c_{\mathbf{B}}$ (2.43)

$\forall \pi \in d_{\mathbf{B}} \;.$  $\in d_{\mathbf{A}}$ (2.44)

The compositional features of $\mathbf{G}_I(\mathcal{C})$ will often resemble those of $\mathcal{C}$ - for each of symmetric monoidal closed structures, $*$-autonomy, finite products, and finite coproducts, if they exist in $\mathcal{C}$, then they exist in $\mathbf{G}_I(\mathcal{C})$ and are preserved by the forgetful functor $\mathbf{G}_I(\mathcal{C}) \to \mathcal{C}$ [72]. We will encounter this in more detail across Sections 2.3.2-2.3.6 and 2.5.1 when reviewing how these structures lift for the $\mathrm{Caus}[-]$ construction.

There are multiple intuitions one can use to view the objects of $\mathbf{G}_I(\mathcal{C})$ which we will freely appeal to throughout when explaining concepts:

- The direct mathematical interpretation views objects $\mathbf{A}$ as the pair of a set of states $c_{\mathbf{A}}$ and effects $d_{\mathbf{A}}$ which we deem *normalised* against some criterion associated with $\mathbf{A}$.

- Taking an operational perspective, an object describes a black-box system which an external agent may interact with. The black-box has an internal state taken from $c_{\mathbf{A}}$ which the agent can update by appending morphisms. Monoidal products like $\mathbf{A} \otimes \mathbf{B}$ represent composite or multi-partite systems. An agent who only has access to $\mathbf{A}$ can now only apply morphisms of the form $f \otimes \mathrm{id}_{\mathbf{B}}$ as they must commute with every action another agent could take on $\mathbf{B}$. The effects $d_{\mathbf{A}}$ describe the possible complete sequences of actions the agent can take which yield a closed scenario. In general there may be multiple distinct marginal states at $\mathbf{B}$ induced by the different effects, from which we can define non-signalling conditions and new objects whose states are admissible with respect to some causal structure. The freedom of picking any state space $c_{\mathbf{A}}$ and effect space $d_{\mathbf{B}}$ allows us to offer agents an artificially restricted set of actions as needed - for example, in $\mathbf{G}_I(\mathrm{CP})$ we will not only have a qubit object that supports all (mixed) states, but also an object that restricts states to specific planes or axes of the Bloch sphere or, dually, allows an agent to postselect within a given plane. In Section 2.5.2 we will introduce a way to encode observations via binary tests, representing the agent's knowledge as an auxiliary object they also have access to on which we can condition future actions (e.g. to internalise the agent's decision-making). With this in mind, the effects $d_{\mathbf{A}}$ involve marginalising away any such knowledge as the other agent on $\mathbf{B}$ would not have access to it.

- We may view an object as a choice of protocol or interface. We can view a state and effect pair as implementations of two parties participating in the protocol to completion, one as the party in focus and the other as the environment or external party. The analogue of causal structure here is the format and ordering requirements of the messages that are sent between the parties as part of the protocol.

Applying double-glueing to a precausal category is close to fulfilling our goals, but we need to restrict the objects such that the conditions of Equations 2.43-2.44 enforce causality. If we consider fixing some set $c \subseteq \mathcal{C}(I, A)$, we can construct the maximal set of effects that are causal in the context of every state in c.

Definition 2.3.6: Dual sets [78, Definition 4.1]

Given $c \subseteq \mathcal{C}(I, A)$, the *dual set* is

$$c^* := \left\{ \pi \in \mathcal{C}(A, I) \;\middle|\; \forall \rho \in c.\; \begin{array}{c} \boxed{\pi} \\ | \\ \boxed{\rho} \end{array} = \mathrm{id}_I \right\} \tag{2.45}$$

By compact closure and transposing elements, we may choose to equivalently interpret c/c^* as either a set of states in $\mathcal{C}(I, A)/\mathcal{C}(I, A^*)$ or a set of effects $\mathcal{C}(A^*, I)/\mathcal{C}(A, I)$. A set $c \subseteq \mathcal{C}(I, A)$ is *closed* if $c = c^{**}$.

$(-)^*$ is an instance of a focussed orthogonality [72] which immediately gives the following, implying that $(-)^{**}$ is idempotent and acts as a kind of closure operation:

$$(-)^{***} = (-)^* \tag{2.46}$$

$$\forall c.\; c \subseteq c^{**} \tag{2.47}$$

Additionally, duality reverses the ordering of inclusion between any two sets:

$$c \subseteq c' \implies c'^* \subseteq c^* \tag{2.48}$$

Example 2.3.7

We can easily see that closure $(-)^{**}$ forms the *smallest closed set* containing the input. That is, whenever c is a subset of some closed set c'^{**}, we have:

$$\begin{aligned} c \subseteq c'^{**} &\implies c'^{***} \subseteq c^* \\ &\implies c^{**} \subseteq c'^{****} = c'^{**} \end{aligned} \tag{2.49}$$

For an object $(A, c_{\mathbf{A}}, d_{\mathbf{A}}) \in \mathbf{G}_I(\mathcal{C})$ to represent a causal system, we need each combination of a state $\rho \in c_{\mathbf{A}}$ and an effect $\pi \in d_{\mathbf{A}}$ to give $\rho \fatsemi \pi = \mathrm{id}_I$, i.e. $c_{\mathbf{A}} \subseteq d_{\mathbf{A}}^*$ and $d_{\mathbf{A}} \subseteq c_{\mathbf{A}}^*$. If we ask that such sets are maximal (i.e. the only restriction we make on morphisms is causality), we have $c_{\mathbf{A}} = d_{\mathbf{A}}^* = c_{\mathbf{A}}^{**}$ (the tight orthogonality subcategory). In this case, it is sufficient to just provide one of the sets $c_{\mathbf{A}}$ or $d_{\mathbf{A}}$ to identify an object.

The additional symmetry here also makes Equations 2.43, 2.44, and 2.50 equivalent, further driving the idea that causal morphisms are those that can be applied with probability 1 in any context.

$$\forall \rho \in c_{\mathbf{A}}, \pi \in d_{\mathbf{B}}\,(= c^*_{\mathbf{B}})\,.\ \begin{array}{c}\boxed{\pi}\\ | \\ \boxed{f}\\ | \\ \boxed{\rho}\end{array} = \mathrm{id}_I \tag{2.50}$$

The only remaining edge case that does not necessarily lead to causality is when either $c_{\mathbf{A}}$ or $c^*_{\mathbf{A}}$ is empty (the other will be the full homset $\mathcal{C}(I,A)$ or $\mathcal{C}(A,I)$ by duality, notably including the zero morphism which is not causal in any context). The final step to get to the $\mathrm{Caus}\,[-]$ construction is to ask that the states are at least causal against some uniform effect (and, dually, ask for the existence of a uniform state).

Definition 2.3.8: Flat sets [78, Definition 4.2]

A set $c \subseteq \mathcal{C}(I,A)$ is *flat* if there exist invertible scalars $\mu, \theta \in \mathcal{C}(I,I)$ such that $\mu \cdot \underline{\underline{\perp}}_A \in c$ and $\theta \cdot \overline{\overline{\top}}_A \in c^*$. When these scalars exist, they are necessarily unique.

Combining the steps so far, we can define $\mathrm{Caus}\,[\mathcal{C}]$ as the full subcategory of $\mathbf{G}_I(\mathcal{C})$ on objects $(A, c_{\mathbf{A}}, d_{\mathbf{A}})$ with $c_{\mathbf{A}} = d^*_{\mathbf{A}} = c^{**}_{\mathbf{A}}$ flat. For simplicity, we will use the following explicit definition:

Definition 2.3.9: The $\mathrm{Caus}\,[-]$ Construction [78, Definition 4.3]

Given a precausal category $\mathcal{C}$, the causal category $\mathrm{Caus}\,[\mathcal{C}]$ has as objects pairs $\mathbf{A} := (A \in \mathrm{Ob}\,(\mathcal{C})\,, c_{\mathbf{A}} \subseteq \mathcal{C}(I,A))$ where $c_{\mathbf{A}}$ is closed and flat. A morphism $f : \mathbf{A} \to \mathbf{B}$ is a morphism $f : A \to B$ in $\mathcal{C}$ such that $\forall \rho \in c_{\mathbf{A}}.\ \rho \,{\fatsemi}\, f \in c_{\mathbf{B}}$.

This has an obvious forgetful functor $\mathcal{U} : \mathrm{Caus}\,[\mathcal{C}] \to \mathcal{C}$ which drops the sets from the objects and is identity on morphisms. We may refer to objects of $\mathrm{Caus}\,[\mathcal{C}]$ as *causal types*, where the judgement $\rho : \mathbf{A}$ means the state ρ is *causal for* $\mathbf{A}$ ($\rho \in c_{\mathbf{A}}$), resembling typing for a program. We introduce notation $\mu_{\mathbf{A}}, \theta_{\mathbf{A}} \in \mathcal{C}(I,I)$, ${\downarrow}_{\mathbf{A}} := \mu_{\mathbf{A}} \cdot \underline{\underline{\perp}}_A \in c_{\mathbf{A}}$, ${\uparrow}_{\mathbf{A}} := \theta_{\mathbf{A}} \cdot \overline{\overline{\top}}_A \in c^*_{\mathbf{A}}$ for the normalisation constants and normalised uniform state and effect required by flatness.

Remark 2.3.10

For any object $\mathbf{A} \in \mathrm{Ob}\,(\mathrm{Caus}\,[\mathcal{C}])$ and non-zero scalar $\lambda \in \mathcal{C}(I,I)$, we also have the isomorphic object $\lambda\mathbf{A}\,(\cong \mathbf{A}) \in \mathrm{Ob}\,(\mathrm{Caus}\,[\mathcal{C}])$ where we scale the states $c_{\lambda\mathbf{A}} := \{\lambda \cdot \rho \mid \rho \in c_{\mathbf{A}}\}$ (and inversely scale effects $c^*_{\lambda\mathbf{A}}$). The scalar is meaningless for physical interpretations as it will commute with all actions or observations made by an agent, and will always meet its inverse in a closed scenario.

Remark 2.3.11

Throughout this thesis, whenever we present isomorphisms of objects $\mathbf{A} \cong \mathbf{B}$ without explicitly providing the isomorphism, the underlying isomorphism in $\mathcal{C}$ is the unique coherent morphism $f : A \to B$ between the two carrier objects (i.e. the identity up to permutations, associators and unitors). From the definition of Caus$[\mathcal{C}]$, this isomorphism must preserve causal states in both directions, i.e. $\forall \rho_A \in c_{\mathbf{A}}.\rho_A \mathbin{⨾} f \in c_{\mathbf{B}}$ and $\forall \rho_B \in c_{\mathbf{B}}.\rho_B \mathbin{⨾} f^{-1} \in c_{\mathbf{A}}$. Equalities $\mathbf{A} = \mathbf{B}$ are the special case where f is precisely an identity (i.e. the carrier objects coincide $A = B$). These coherent isomorphisms are similarly implicit in (extra)natural transformations $\mathbf{A} \Rightarrow \mathbf{B}$, where f is causal but f^{-1} need not be (when f is an identity, $c_{\mathbf{A}} \subseteq c_{\mathbf{B}}$). Many of these natural transformations and isomorphisms may resemble theorems of linear logic, though they are not guaranteed to match the expectations from linear logic exactly (e.g. some isomorphisms from linear logic will be strict equalities here). In Chapter 3, we will generalise these notions to describe when any string diagram from $\mathcal{C}$ can lift to Caus$[\mathcal{C}]$ (see Section 3.2) as well as make the connections to linear logic more explicit.

The other parts of this section will dive into the operators of Caus$[\mathcal{C}]$ to both see how they lift the compositional structure of $\mathcal{C}$ and generate interesting examples.

2.3.2 Monoidal Unit

The unit I of $\mathcal{C}$ can be lifted to the following object $\mathbf{I}$ in Caus$[\mathcal{C}]$. We claim here that this will be the monoidal unit for several monoidal structures, and will prove it when we introduce each of them.

Definition 2.3.12: Monoidal unit [78]

$$\mathbf{I} := (I, \{\mathrm{id}_I\})$$

$\mathbf{I}$ has a single valid state, which we typically view as an empty diagram. This means that for any $\mathbf{A}$, its states are precisely the homset Caus$[\mathcal{C}](\mathbf{I}, \mathbf{A}) = c_{\mathbf{A}}$ and effects are Caus$[\mathcal{C}](\mathbf{A}, \mathbf{I}) = c_{\mathbf{A}}^*$. Again, this reinforces the concept of causal normalisation since the composition of any state $\rho : \mathbf{I} \to \mathbf{A}$ and effect $\pi : \mathbf{A} \to \mathbf{I}$ must be id_I, the unit (abstract) probability.

2.3.3 Dual Objects

Compact closure of $\mathcal{C}$ allows us to transpose any morphism $f \in \mathcal{C}(A, B)$ to $f^* \in \mathcal{C}(B^*, A^*)$ as defined in Equation 2.19, giving us a time-symmetric view of our processes. Transposition exchanges notions of states and effects and formally expresses a duality between viewing morphisms as transforming states or transforming effects. This duality is lifted to $\mathrm{Caus}\left[\mathcal{C}\right]$ as a similar (strong monoidal) functor $(-)^* : \mathrm{Caus}\left[\mathcal{C}\right]^{\mathrm{op}} \to \mathrm{Caus}\left[\mathcal{C}\right]$. We will also prove that this is a proper duality when we look at the monoidal structures.

Definition 2.3.13: Dual objects [78]

The duality functor $(-)^* : \mathrm{Caus}\left[\mathcal{C}\right]^{\mathrm{op}} \to \mathrm{Caus}\left[\mathcal{C}\right]$ is defined on objects as

$$\mathbf{A}^* := (A^*, c^*_{\mathbf{A}})$$

and sends morphisms to their transpose in $\mathcal{C}$.

When viewing an object as describing a set of processes or implementations of an interface, the dual object describes the set of effects/contexts that can be applied to them with probability 1 or ways to consume or interact with a black-box presenting that interface.

It is straightforward to see that the monoidal unit is self-dual $\mathbf{I} = \mathbf{I}^*$.

2.3.4 First-order Objects

First-order systems $\mathbf{A^1}$ and their duals $\mathbf{A^{1*}}$ are the simplest atomic components of a causal scenario, capturing degenerate interfaces that respectively represent outputs with (in general) multiple distinguishable states but only one choice of effect $\overset{=}{\top}_A$ that an external agent can apply, or inputs where an external agent can choose how to interact with the system but they cannot receive information from it.

In most theories, states of a first-order system will correspond to basic data (distributions over a finite set) or descriptions of a physical system (density matrices of a finite-dimensional Hilbert space) at a single point in time.

Definition 2.3.14: First-order objects [78, Definition 5.1]

An object $\mathbf{A} \in \mathrm{Caus}\left[\mathcal{C}\right]$ is *first-order* when there is a unique causal effect, $|c^*_{\mathbf{A}}| = 1$. By the assumption of flatness, the unique effect must be $\overset{\bullet}{|}_{\mathbf{A}}$, i.e. $\overset{=}{\top}_A$ up to some invertible scalar. For each $A \in \mathrm{Ob}\left(\mathcal{C}\right)$, there is a unique first-order object (up to isomorphism) denoted as $\mathbf{A^1}$.

$$\mathbf{A^1} := \left(A, \left\{\overset{=}{\top}_A\right\}^*\right)$$

Since the morphisms in $\mathrm{Caus}[\mathcal{C}]$ can be fully characterised as those that preserve effects (Equation 2.44), it follows immediately from this definition that morphisms $f : \mathbf{A^1} \to \mathbf{B^1}$ between two first-order objects precisely match Definition 2.1.1 for first-order causal morphisms.

If a system is both first-order and first-order dual, i.e. it has a single state and effect, then it is isomorphic to the trivial system $\mathbf{I}$.

Definition 2.3.15: First-order subcategory

We denote the full subcategory of first-order objects as $\mathrm{FO}(\mathrm{Caus}[\mathcal{C}])$. Within this subcategory, $\mathbf{I}$ is terminal ($\mathrm{Caus}[\mathcal{C}](\mathbf{A^1}, \mathbf{I}) = \left\{\bar{\top}_A\right\}$).

2.3.5 Tensor and Par

In $\mathcal{C}$, $A \otimes B$ represents parallel composition of two systems. There are multiple ways of lifting this to $\mathrm{Caus}[\mathcal{C}]$ depending on what kinds of states and effects we permit on the joint system.

$\mathbf{A} \otimes \mathbf{B}$ is the smallest closed space containing all separable states over the joint system, which may include some states that are not separable or even localisable.

Definition 2.3.16: Tensor [78]

The *tensor* product in $\mathrm{Caus}[\mathcal{C}]$ is the bifunctor $\otimes : \mathrm{Caus}[\mathcal{C}] \times \mathrm{Caus}[\mathcal{C}] \to \mathrm{Caus}[\mathcal{C}]$ defined on objects as

$$\mathbf{A} \otimes \mathbf{B} := \left(A \otimes B, \left\{ \boxed{\rho_A}\ \boxed{\rho_B} \ \middle|\ \rho_A \in c_{\mathbf{A}}, \rho_B \in c_{\mathbf{B}} \right\}^{**} \right)$$

and on morphisms identically to $\otimes : \mathcal{C} \times \mathcal{C} \to \mathcal{C}$.

Theorem 2.3.17: [78, Theorem 4.7]

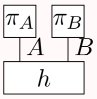
$(\otimes, \mathbf{I})$ forms a symmetric monoidal structure in $\mathrm{Caus}[\mathcal{C}]$.

At the other extreme, $\mathbf{A} \parr \mathbf{B}$ is the full space of bipartite processes that act locally like $\mathbf{A}$ and $\mathbf{B}$, yielding an object with more states and fewer effects.

Definition 2.3.18: Par [78]

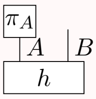
The *par* product in $\mathrm{Caus}[\mathcal{C}]$ is the bifunctor $\parr : \mathrm{Caus}[\mathcal{C}] \times \mathrm{Caus}[\mathcal{C}] \to \mathrm{Caus}[\mathcal{C}]$ defined on objects as

$$\mathbf{A} \parr \mathbf{B} := \left(A \otimes B, \left\{ h \;\middle|\; \forall \pi_A \in c^*_{\mathbf{A}}, \pi_B \in c^*_{\mathbf{B}}. \right.\right.$$



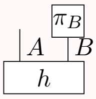
$$\left.\left. = \mathrm{id}_I \right\} \right) \tag{2.51}$$

$$= \left(A \otimes B, \left\{ h \;\middle|\; \forall \pi_A \in c^*_{\mathbf{A}}. \right.\right.$$



$$\left.\left. \in c_{\mathbf{B}} \right\} \right) \tag{2.52}$$

$$= \left(A \otimes B, \left\{ h \;\middle|\; \forall \pi_B \in c^*_{\mathbf{B}}. \right.\right.$$



$$\left.\left. \in c_{\mathbf{A}} \right\} \right) \tag{2.53}$$

and on morphisms identically to $\otimes : \mathcal{C} \times \mathcal{C} \to \mathcal{C}$.

These three definitions for $\mathbf{A} \parr \mathbf{B}$ are equivalent by Definition 2.3.6.

Whilst the local marginals show that this acts like an $\mathbf{A}$ and $\mathbf{B}$ in parallel, causal contexts can only act locally on them (or at least are generated by closure $(-)^{**}$ of the separable contexts). This allows us to model distributed protocols or physical systems separated by a substantial distance in space or time, where there may not be a way to present the $\mathbf{A}$ and $\mathbf{B}$ interfaces simultaneously for a localised agent to interact with both.

In contrast, we can think of $\mathbf{A} \otimes \mathbf{B}$ as a system where the $\mathbf{A}$ and $\mathbf{B}$ are simultaneously accessible, so the contexts can freely pass information between them. For example, if $\mathbf{B} = \mathbf{A}^*$ then a separable state is precisely a pair of state and effect for $\mathbf{A}$ which we can plug together since the cap $\epsilon_A \in \mathcal{C}(A \otimes A^*, I)$ of the compact structure is causal $\epsilon_A \in \mathrm{Caus}[\mathcal{C}](\mathbf{A} \otimes \mathbf{A}^*, \mathbf{I})$. In the multi-round communication protocol picture, ϵ_A would play a copycat, forwarding each input from the $\mathbf{A}$ side to the corresponding output on the $\mathbf{A}^*$ side and vice versa.

The relationship between separable states of $\mathbf{A} \otimes \mathbf{B}$ (up to closure) and separable effects of $\mathbf{A} \parr \mathbf{B}$ is exhibited in the de Morgan duality between $\otimes$ and $\parr$:

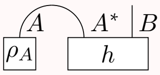
$$(\mathbf{A} \otimes \mathbf{B})^* = \mathbf{A}^* \parr \mathbf{B}^* \tag{2.54}$$

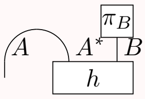
We can understand $\parr$ a little better by looking at the internal hom $\mathbf{A} \multimap \mathbf{B}$ which represents the space of transformations from $\mathbf{A}$ to $\mathbf{B}$.

Definition 2.3.19: Internal hom [78]

$\multimap$: $\mathrm{Caus}[\mathcal{C}]^{\mathrm{op}} \times \mathrm{Caus}[\mathcal{C}] \to \mathrm{Caus}[\mathcal{C}]$ is the bifunctor on $\mathrm{Caus}[\mathcal{C}]$ defined on objects as

$$\mathbf{A} \multimap \mathbf{B} := \mathbf{A}^* \parr \mathbf{B} \tag{2.55}$$

$= \left(A^* \otimes B, \left\{ h \,\middle|\, \forall \rho_A \in c_{\mathbf{A}}. \right.\right.$  $\left.\left. \in c_{\mathbf{B}} \right\}\right)$ (2.56)

$= \left(A^* \otimes B, \left\{ h \,\middle|\, \forall \pi_B \in c^*_{\mathbf{B}}. \right.\right.$  $\left.\left. \in c^*_{\mathbf{A}} \right\}\right)$ (2.57)

and on morphisms identically to $(-)^* \otimes (=) : \mathcal{C}^{\mathrm{op}} \times \mathcal{C} \to \mathcal{C}$.

Each state of $\mathbf{A} \multimap \mathbf{B}$ is an encoding of a transformation which we can view in either direction: composing it with any state of $\mathbf{A}$ gives a state of $\mathbf{B}$, or dually applying any effect $\mathbf{B}^*$ yields an effect $\mathbf{A}^*$. $\multimap$ makes Caus$[\mathcal{C}]$ monoidal closed wrt $\otimes$, with encodings and evaluations given by composing with the cup $\eta_A : \mathbf{I} \to \mathbf{A}^* \parr \mathbf{A}$ and cap $\epsilon_A : \mathbf{A} \otimes \mathbf{A}^* \to \mathbf{I}$, precisely capturing the Choi-Jamiołkowski isomorphism.

As the symbols $\otimes, \parr, \multimap$ suggest, these form a model of *multiplicative linear logic*, or more specifically an extension called *ISOMIX* logic.

Theorem 2.3.20: [78, Theorem 4.10]

For any precausal category $\mathcal{C}$, Caus$[\mathcal{C}]$ is an ISOMIX category; that is, a $*$-autonomous category with a coherent isomorphism $I \cong I^*$.

Corollary 2.3.21

$(\parr, \mathbf{I})$ forms a symmetric monoidal structure in Caus$[\mathcal{C}]$, and $\otimes$ is closed monoidal with right-adjoint $\multimap$ and duality functor $(-)^*$.

Proof. Immediate from 2.3.20. □

This immediately generates equalities and natural transformations between expressions over the operators. Examples of this include Equations 2.54 and 2.55 and the natural transformations below representing subset inclusions on the state sets, which corresponds to the *mix* and *switch* rules of deep inference for linear logic [58, 109] (see Section 3.1).

$$\mathbf{A} \otimes \mathbf{B} \Rightarrow \mathbf{A} \parr \mathbf{B} \tag{2.58}$$

$$\mathbf{A} \otimes (\mathbf{B} \parr \mathbf{C}) \Rightarrow (\mathbf{A} \otimes \mathbf{B}) \parr \mathbf{C} \tag{2.59}$$

However, additional equations exist specifically for first-order objects.

Proposition 2.3.22: [78, Corollary 5.5]

For any first-order objects $\mathbf{A^1}$, $\mathbf{B^1}$, their tensor and par coincide.

$$\mathbf{A^1} \otimes \mathbf{B^1} \cong \mathbf{A^1} \parr \mathbf{B^1}$$

Furthermore, the result is the first-order object on $A \otimes B$.

We will further address the implications of the logical structure of $\mathrm{Caus}[\mathcal{C}]$ in Section 3.2.3.

A final distinction we can make between $\otimes$ and $\parr$ is how they impact the level of information signalling across their states and how this can be used to construct the spaces of processes compatible with some simple causal structures.

Theorem 2.3.23: [78, Theorems 6.2, 6.3, 6.4, Corollary 6.9]

Fix some first-order objects $\mathbf{A^1}, \mathbf{B^1}, \mathbf{C^1}, \mathbf{D^1}$ and consider some Choi operator $h \in \mathcal{C}(I, A^* \otimes B \otimes C^* \otimes D)$. The following statements hold up to the appropriate permutation of the systems:

FO$\otimes$. $h : (\mathbf{A^1} \multimap \mathbf{B^1}) \otimes (\mathbf{C^1} \multimap \mathbf{D^1})$ iff h is the Choi operator of a first-order causal non-signalling process $A \otimes C \to B \otimes D$.

FO$\parr$. $h : (\mathbf{A^1} \multimap \mathbf{B^1}) \parr (\mathbf{C^1} \multimap \mathbf{D^1})$ iff h is the Choi operator of a first-order causal process $A \otimes C \to B \otimes D$.

FO$\multimap$. $h : \mathbf{A^1} \multimap (\mathbf{B^1} \multimap \mathbf{C^1}) \multimap \mathbf{D^1}$ iff h is the Choi operator of a first-order causal one-way signalling process $A \otimes C \to B \otimes D$.

These all generalise to the multi-partite case accordingly. In the latter case, the states of $\mathbf{A^1_0} \multimap (\ldots(\mathbf{A^1_{N-1}} \multimap \mathbf{A^1_N})\ldots) \multimap \mathbf{A^1_{2N-1}}$ coincide with the N-combs on $(A_0, \ldots, A_{2N-1})$.

The most surprising of these is the characterisation of $\otimes$. Unpacking the definition of $(\mathbf{A^1} \multimap \mathbf{B^1}) \otimes (\mathbf{C^1} \multimap \mathbf{D^1})$, we start with the separable first-order causal channels and then apply the closure operator $(-)^{**}$ which would add additional processes. This result shows that this closure is capable of introducing correlations between the left and right subsystems, but that the correlations it can introduce are too weak to allow for information to pass between them. This is the first-step towards characterising what the closure operator *actually* does from a linear algebra perspective, which we will refine later (see the Affine Closure Theorem).

In addition to combs, $\mathrm{Caus}[\mathcal{C}]$ is home to many other higher-order processes of interest. In particular, whilst $\otimes$ seems to limit us to non-signalling processes - a definite causal structure - dualising it to $\parr$ no only permits information signalling in both directions but covers processes with indefinite causal structure.

Theorem 2.3.24: [78, Theorem 7.2] (simplified)

Fix some first-order objects $\mathbf{A^1}, \mathbf{B^1}, \mathbf{C^1}, \mathbf{D^1}$. $h \in \mathcal{C}(I, A^* \otimes B \otimes C^* \otimes D)$ is a process matrix iff $h : ((\mathbf{A^1} \multimap \mathbf{B^1}) \otimes (\mathbf{C^1} \multimap \mathbf{D^1}))^*$.

Example 2.3.25

The Choi operator `QSw` of the quantum switch over $A \in \mathrm{CP}$ is a morphism of type $((\mathbf{A^1} \multimap \mathbf{A^1}) \otimes (\mathbf{A^1} \multimap \mathbf{A^1})) \multimap (\mathbf{C^1} \otimes \mathbf{A^1} \multimap \mathbf{C^1} \otimes \mathbf{A^1})$ in $\mathrm{Caus}[\mathrm{CP}]$ where $\mathbf{C^1}$ is the first-order object on $\mathbb{C}^2$.

2.3.6 Intersections

The final operation investigated in [78] was an *intersection* $\cap$ of causal objects, allowing us to combine constraints to generate more general causal structures. Rather than corresponding to a logical connective, this is intended to model intersection of sets similarly to intersection types [35, 8] in programming languages.

One complication is that, if we try to take the intersection of two sets with different underlying objects or different constants of proportionality $(\mu_{\mathbf{A}}, \theta_{\mathbf{A}})$, we will get the empty set which is not flat.

Theorem 2.3.26: [78, Theorem 6.15] (simplified)

For any objects $\mathbf{A} = (A, c_{\mathbf{A}})$, $\mathbf{A}' = (A, c_{\mathbf{A}'})$ with the same underlying object $A \in \mathrm{Ob}(\mathcal{C})$ and the same normalisation scalars $\mu_{\mathbf{A}} = \mu_{\mathbf{A}'}$ and $\theta_{\mathbf{A}} = \theta_{\mathbf{A}'}$, the intersection object

$$\mathbf{A} \cap \mathbf{A}' := (A, c_{\mathbf{A}} \cap c_{\mathbf{A}'}) \tag{2.60}$$

yields a pullback for the causal cospan

$$\begin{tikzcd}
\mathbf{A} \cap \mathbf{A}' \arrow[r, "\mathrm{id}_A"] \arrow[d, "\mathrm{id}_{A'}"'] \arrow[dr, phantom, "\lrcorner", very near start] & \mathbf{A} \arrow[d, "\mathrm{id}_A"] \\
\mathbf{A}' \arrow[r, "\mathrm{id}_{A'}"] & \mathbf{A}''
\end{tikzcd} \tag{2.61}$$

Example 2.3.27

For any $\mathbf{A}$ and $\mathbf{B}$, the objects $\mathbf{A} \otimes \mathbf{B}$ and $\mathbf{A} \parr \mathbf{B}$ both have the same underlying object ($A \otimes B \in \mathrm{Ob}(\mathcal{C})$) and normalisation scalars ($\mu_{\mathbf{A}} \cdot \mu_{\mathbf{B}}$ and $\theta_{\mathbf{A}} \cdot \theta_{\mathbf{B}}$), so we can construct their intersection

$$(\mathbf{A} \otimes \mathbf{B}) \cap (\mathbf{A} \parr \mathbf{B}) = (A \otimes B, c_{\mathbf{A} \otimes \mathbf{B} \cap c_{\mathbf{A} \parr \mathbf{B}}}) \tag{2.62}$$

However, the natural transformation $\mathbf{A} \otimes \mathbf{B} \Rightarrow \mathbf{A} \parr \mathbf{B}$ (Equation 2.58) tells us that $c_{\mathbf{A} \otimes \mathbf{B}} \subseteq c_{\mathbf{A} \parr \mathbf{B}}$, so this intersection just collapses to the tensor:

$$(\mathbf{A} \otimes \mathbf{B}) \cap (\mathbf{A} \parr \mathbf{B}) = \mathbf{A} \otimes \mathbf{B} \tag{2.63}$$

Remark 2.3.28

The full version of [78, Theorem 6.15] also accounts for permutations, e.g. if A and A' are monoidal products of objects in different orders. For notational ease, when writing $\mathbf{A} \cap \mathbf{A}'$ we will assume that the appropriate permutation isomorphisms are implicitly applied to ensure the carrier objects in $\mathcal{C}$ match exactly. At times, we may clarify by fixing the target permutation to, for example, some ordering over the elements of a causal structure.

Using the following result, we can generate objects describing an arbitrary causal structure $\leq$ from intersections of linear causal structures (characterised by the comb construction FO$\multimap$). This matches the standard philosophy of using partial orders to represent multiple possible total orderings over elements, such as in event orderings for distributed systems [82].

Theorem 2.3.29: [78, Theorem 6.12]

Given a causal structure $\leq \subseteq V \times V$, a *totalisation* $< : V \times V$ is a total order with the same elements, i.e. $u \leq v \implies u < v$.

A process f is signal-consistent with respect to $\leq$ iff f is signal-consistent with respect to every totalisation of $\leq$.

Example 2.3.30

Non-signalling can be expressed as the intersection of one-way non-signalling in each direction.

$$\left(\mathbf{A^1} \multimap \mathbf{B^1}\right) \otimes \left(\mathbf{C^1} \multimap \mathbf{D^1}\right) = \begin{array}{c} (\mathbf{A^1} \multimap (\mathbf{B^1} \multimap \mathbf{C^1}) \multimap \mathbf{D^1}) \\ \cap\ (\mathbf{C^1} \multimap (\mathbf{D^1} \multimap \mathbf{A^1}) \multimap \mathbf{B^1}) \end{array} \tag{2.64}$$

2.4 Assumptions on the Underlying Theory

When investigating higher order causal theories, it is useful to strengthen the definition of a precausal category in a handful of ways. This section will present an alternative set of axioms to assume with some brief justifications, before demonstrating how they provide additional mathematical tools from linear algebra and how this affects our understanding of the structure of $\mathrm{Caus}[\mathcal{C}]$.

2.4.1 Additive Precausal Categories

For the remainder of this thesis, we will adapt the definition of $\mathrm{Caus}[\mathcal{C}]$ to be built from an *additive precausal category* $\mathcal{C}$.

Definition 2.4.1: Additive precausal category

Let $\mathcal{C}$ be a compact closed category with products. Such a category always has biproducts and additive enrichment [69]: each homset is a commutative monoid, writing summation of $f, g \in \mathcal{C}(A, B)$ as $f + g \in \mathcal{C}(A, B)$ and zero morphisms as $0_{A,B} \in \mathcal{C}(A, B)$. $\mathcal{C}$ is an *additive precausal category* if:

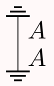
APC1. $\mathcal{C}$ has a discarding process $\bar{\top}_A \in \mathcal{C}(A, I)$ for every system A, compatible with the monoidal and biproduct structures as below;

$$\bar{\top}_{A\otimes B} = \bar{\top}_A \otimes \bar{\top}_B \tag{2.65}$$

$$\bar{\top}_I = \mathrm{id}_I \tag{2.66}$$

$$\bar{\top}_{A\oplus B} = \left[\bar{\top}_A, \bar{\top}_B\right] \tag{2.67}$$

APC2. The scalar d_A is invertible for all non-zero A;

$d_A :=$  (2.68)

APC3. Each object $A \in \mathrm{Ob}(\mathcal{C})$ has a finite causal basis: a minimal finite set $\{\rho_i\}_{i\in\mathfrak{B}_A^{\mathcal{C}}} \subseteq \left\{\bar{\top}_A\right\}^*$ of causal states (indexed by some set $\mathfrak{B}_A^{\mathcal{C}}$ with $|\mathfrak{B}_A^{\mathcal{C}}| = \dim(A)$) which are sufficient to distinguish morphisms;

$$\forall B.\ \forall f, g \in \mathcal{C}(A, B).\ \left(\forall i \in \mathfrak{B}_A^{\mathcal{C}}.\ \begin{array}{c}\boxed{f}\\ \boxed{\rho_i}\end{array} = \begin{array}{c}\boxed{g}\\ \boxed{\rho_i}\end{array}\right) \Longrightarrow \boxed{f} = \boxed{g} \tag{2.69}$$

APC4. Addition of scalars is *cancellative* ($\forall x, y, z.\ x + z = y + z \Longrightarrow x = y$), *totally preordered* ($\forall x, y.\ \exists z.\ x = y + z \vee y = x + z$), and all non-zero scalars have a multiplicative inverse.

APC5. All effects have a complement with respect to discarding: for any $\pi \in \mathcal{C}(A, I)$, there exists some $\pi' \in \mathcal{C}(A, I)$ and scalar λ such that $\pi + \pi' = \lambda \cdot \overset{=}{\top}_A$.

The first 3 axioms relate closely to the corresponding ones in Definition 2.3.1, whereas the last two are quite different in flavour, and are in some sense more elementary, as their proofs don't rely on any particularly deep facts about our main classical and quantum examples (see Examples 2.4.2 and 2.4.4 below).

The axioms APC1 and APC2 above are essentially identical to PC1 and PC2, with the additional requirement that discarding be compatible with biproducts as well as tensor products.

APC3 is a strengthening of condition PC3. Rather than requiring us to check a pair of processes agree on *all* causal states to be equal, we only require agreement on some fixed finite set of states. In other words, each system has a set of states that behaves like a basis spanning all the others, for the purposes of distinguishing maps. In the quantum foundations literature, this is sometimes called a *fiducial set of states*.

APC4 says that the semiring of scalars $\mathcal{C}(I, I)$ behaves somewhat like the set of non-negative real numbers $\mathbb{R}^+$, allowing us to interpret them as (pseudo)-probabilities. While the scalars need not be a field (indeed our main example $\mathbb{R}^+$ is not), any field satisfies this axiom as well.

APC5 allows us to interpret effects (up to some renormalisation) as testing some predicate. To see how this works, first assume for simplicity that $\lambda = \mathrm{id}_I$. We can think of an effect $\pi : A \to I$ as some predicate over A, and π' as its negation. For any causal state ρ, we can think of the composition $p_1 := \rho \fatsemi \pi$ as the probability that π is true for ρ and $p_2 := \rho \fatsemi \pi'$ as the probability that π is false. The fact that $\pi + \pi' = \overset{=}{\top}_A$ lets us conclude that those probabilities sum to 1:

$$p_1 + p_2 = \rho \fatsemi \pi + \rho \fatsemi \pi' = \rho \fatsemi (\pi + \pi') = \rho \fatsemi \overset{=}{\top}_A = \mathrm{id}_I \tag{2.70}$$

If $\lambda \neq \mathrm{id}_I$, the previous reasoning holds after re-normalising, i.e. replacing π and π' with $\lambda^{-1} \cdot \pi$ and $\lambda^{-1} \cdot \pi'$.

Example 2.4.2: Classical probability theory

$\mathrm{Mat}[\mathbb{R}^+]$ defined as in Example 2.3.2 is also an additive precausal category, where $\oplus$ is given by the direct sum of matrices. The standard basis of unit vectors gives a basis for APC3, the semiring of scalars $\mathrm{Mat}[\mathbb{R}^+](I, I) \cong \mathbb{R}^+$ satisfies APC4, and for APC5, we just need to choose a suitably large λ such that $\pi' := \lambda \cdot \overset{=}{\top}_A - \pi$ contains only positive numbers.

Example 2.4.3: Affine probability theory

In addition to $\mathbb{R}^+$, we can construct an additive precausal category Mat$[K]$ for any field of characteristic 0. In particular, Mat$[\mathbb{R}]$ is an additive precausal category that is identical to Mat$[\mathbb{R}^+]$ but without any positivity constraint, describing affine or “quasi-probabilistic” maps where negative probabilities are permitted.

Example 2.4.4: Finite-dimensional quantum theory

The quantum example is very nearly the category CP, as defined in Example 2.3.3, but CP doesn’t have biproducts. If we freely add biproducts, we obtain a category CP* whose objects are all finite-dimensional C*-algebras (or equivalently, algebras of the form $\mathcal{L}(H_1) \oplus \ldots \oplus \mathcal{L}(H_k)$) and completely positive maps. Discarding is again given by the trace operator, so APC1 and APC2 are straightforward to verify. For APC3, we can fix a (non-orthogonal) basis of states for each type. As in the classical case, the scalars are $\mathbb{R}^+$, so APC4 is immediate and since $\overset{=}{\top}_A$ is an interior point in the cone of positive effects, π' can be defined as $\lambda \cdot \overset{=}{\top}_A - \pi$ for suitably large λ.

We note that the restrictions of (additive) precausal categories still omit a number of interesting alternative settings, including:

- Categories of non-deterministic processes such as Rel, where the scalars are boolean values with a non-cancellative addition given by disjunction. Rel is also known to fail PC5 (see [78, Appendix A.3]).

- Settings with infinite-dimensional systems such as Hilb, which are rarely compact closed.

- Real quantum mechanics (i.e. the subcategory of CP* containing the matrices with real-valued entries) which is compact-closed but doesn’t admit local discrimination, failing PC3/APC3.

2.4.2 Subtractive Closure

Note that, with the help of bases (APC3), we can promote additive cancellativity of scalars to additive cancellativity for all processes.

Lemma 2.4.5

In an additive precausal category:

$$\forall f, g, h \in \mathcal{C}(A, B) \,.\; f + h = g + h \implies f = g \qquad \text{(APC4a)}$$

Proof. By APC3, we can fix bases of causal states $\{\rho_i\}_{i \in \mathfrak{B}^{\mathcal{C}}_A}$, $\{\upsilon_j\}_{j \in \mathfrak{B}^{\mathcal{C}}_{B^*}}$ for the systems A and B^* respectively. By transposing the υ_j, we can regard them as effects $\upsilon^*_j : B \to I$. Now, suppose $f + h = g + h$. Then, for all i, j:

$$\rho_i ⨾ (f+h) ⨾ \upsilon^*_j = \rho_i ⨾ (g+h) ⨾ \upsilon^*_j \implies \rho_i ⨾ f ⨾ \upsilon^*_j + \rho_i ⨾ h ⨾ \upsilon^*_j = \rho_i ⨾ g ⨾ \upsilon^*_j + \rho_i ⨾ h ⨾ \upsilon^*_j \quad (2.71)$$

Applying cancellation for scalars APC4, we conclude that $\rho_i ⨾ f ⨾ \upsilon^*_j = \rho_i ⨾ g ⨾ \upsilon^*_j$. Transposing and applying APC3 to the basis $\{\upsilon_j\}_{j \in \mathfrak{B}^{\mathcal{C}}_{B^*}}$ gives $\rho_i ⨾ f = \rho_i ⨾ g$. Then, a second application of APC3 to $\{\rho_i\}_{i \in \mathfrak{B}^{\mathcal{C}}_A}$ gives $f = g$. □

This condition allows us to define the free *subtractive closure* $\mathrm{Sub}(\mathcal{C})$, which extends $\mathcal{C}$ with all negatives, and prove that there exists a faithful embedding $[-] : \mathcal{C} \to \mathrm{Sub}(\mathcal{C})$.

Definition 2.4.6: Subtractive closure

Given an additive precausal category $\mathcal{C}$, the free subtractive closure $\mathrm{Sub}(\mathcal{C})$ formed by the objects of $\mathcal{C}$ and morphisms $A \to B$ are equivalence classes of pairs of morphisms $f, g \in \mathcal{C}(A, B)$ under the relation $(f, g) \sim (f', g') \overset{\text{def}}{\iff} f + g' = f' + g$. Composition of morphisms is defined as $(f, g) ⨾ (x, y) := (f ⨾ x + g ⨾ y, f ⨾ y + g ⨾ x)$.

The intent behind this construction is that a morphism (f, g) should represent the expression $f - g$, and hence this should be seen as equivalent to $f' - g'$ when $(f - g) + g + g' = f + g' = f' + g = (f' - g') + g + g'$ by cancellativity.

Proposition 2.4.7

$\mathrm{Sub}(\mathcal{C})$ is an Ab-enriched category.

Proof. Firstly, we need $\sim$ to be an equivalence relation. Reflexivity and symmetry are both trivial. For transitivity, given $(f, g) \sim (a, b) \sim (x, y)$ we have $f + b = a + g$ and $a + y = x + b$. Hence, $f + y + b = a + g + y = x + b + g$, so by cancellativity $f + y = x + g$, i.e. $(f, g) \sim (x, y)$.

Since morphisms of $\mathrm{Sub}(\mathcal{C})$ are equivalence classes, we need composition to be well-defined with respect to our choices of representatives. That is, if $(f,g) \sim (f',g')$ and $(x,y) \sim (x',y')$, then $(f,g) \fatsemi (x,y) \sim (f',g') \fatsemi (x',y')$. From $f + g' = f' + g$ and $x + y' = x' + y$ we have:

$$\begin{aligned} f \fatsemi x + g' \fatsemi x &= f' \fatsemi x + g \fatsemi x \\ f' \fatsemi y + g \fatsemi y &= f \fatsemi y + g' \fatsemi y \\ f' \fatsemi y' + g \fatsemi y' &= f \fatsemi y' + g' \fatsemi y' \\ f \fatsemi x' + g' \fatsemi x' &= f' \fatsemi x' + g \fatsemi x' \\ f \fatsemi x + f \fatsemi y' &= f \fatsemi x' + f \fatsemi y \\ g \fatsemi x' + g \fatsemi y &= g \fatsemi x + g \fatsemi y' \\ g' \fatsemi x' + g' \fatsemi y &= g' \fatsemi x + g' \fatsemi y' \\ f' \fatsemi x + f' \fatsemi y' &= f' \fatsemi x' + f' \fatsemi y \end{aligned} \tag{2.72}$$

We note that, since $\mathcal{C}$ is compact closed, tensor products distribute over summation. In particular, $\forall f.\ f + f = 2 \cdot f$ where 2 is the scalar $\mathrm{id}_I + \mathrm{id}_I$. Summing these equations and using cancellativity:

$$2 \cdot (f \fatsemi x + g \fatsemi y + f' \fatsemi y' + g' \fatsemi x') = 2 \cdot (f \fatsemi y + g \fatsemi x + f' \fatsemi x' + g' \fatsemi y') \tag{2.73}$$

By APC4, 2 is invertible. Cancelling it out, the resulting equation exactly gives us $(f,g) \fatsemi (x,y) \sim (f',g') \fatsemi (x',y')$.

Next, composition must be unital. The identity on A is $(\mathrm{id}_A, 0_{A,A})$ since:

$$\begin{aligned} (f,g) \fatsemi (\mathrm{id}_B, 0_{B,B}) &\sim (f \fatsemi \mathrm{id}_B + g \fatsemi 0_{B,B}, f \fatsemi 0_{B,B} + g \fatsemi \mathrm{id}_B) \\ &\sim (f + 0_{A,B}, 0_{A,B} + g) \\ &\sim (f,g) \\ &\sim (\mathrm{id}_A \fatsemi f + 0_{A,A} \fatsemi g, \mathrm{id}_A \fatsemi g + 0_{A,A} \fatsemi f) \\ &\sim (\mathrm{id}_A, 0_{A,A}) \fatsemi (f,g) \end{aligned} \tag{2.74}$$

We also need composition to be associative. Given $(f,g) : A \to B$, $(x,y) : B \to C$, and $(u,v) : C \to D$:

$$\begin{aligned} ((f,g) \fatsemi (x,y)) \fatsemi (u,v) &\sim (f \fatsemi x + g \fatsemi y, f \fatsemi y + g \fatsemi x) \fatsemi (u,v) \\ &\sim \begin{pmatrix} f \fatsemi x \fatsemi u + g \fatsemi y \fatsemi u + f \fatsemi y \fatsemi v + g \fatsemi x \fatsemi v, \\ f \fatsemi y \fatsemi u + g \fatsemi x \fatsemi u + f \fatsemi x \fatsemi v + g \fatsemi y \fatsemi v \end{pmatrix} \\ &\sim (f,g) \fatsemi (x \fatsemi u + y \fatsemi v, x \fatsemi v + y \fatsemi u) \\ &\sim (f,g) \fatsemi ((x,y) \fatsemi (u,v)) \end{aligned} \tag{2.75}$$

Finally, Ab-enrichment comes from inheriting the enrichment in commutative monoids from $\mathcal{C}$ via $(f, g) + (x, y) := (f + x, g + y)$ and we have inverses $-(f, g) := (g, f)$. We can similarly show that these are well-defined regardless of the choice of representatives by showing $\sim$ is preserved. Invertibility comes from $(f, g) + -(f, g) \sim (f, g) + (g, f) \sim (f + g, g + f) \sim (0_{A,B}, 0_{A,B})$. □

Proposition 2.4.8

There is a faithful embedding $[-] : \mathcal{C} \to \mathrm{Sub}(\mathcal{C})$ that is the identity on objects and maps morphisms as $[f] := (f, 0_{A,B})$.

Proof. For functoriality, $(f, 0_{A,B}) \,⨾\, (g, 0_{B,C}) \sim (f \,⨾\, g, 0_{A,C})$ and identities are $(\mathrm{id}_A, 0_{A,A})$. The identity on objects is injective, so this functor is an embedding. For faithfulness, if we have $f, g : A \to B$ such that $(f, 0_{A,B}) \sim (g, 0_{A,B})$, then $f = f + 0_{A,B} = g + 0_{A,B} = g$. □

To help distinguish between working in different categories, we will use $\sim$ for both equivalence of representative pairs and equality of morphisms of $\mathrm{Sub}(\mathcal{C})$ as equivalence classes, keeping $=$ for equality of morphisms in $\mathcal{C}$. The faithfulness of the embedding ($[f] \sim [g] \implies f = g$) allows us to use $\mathrm{Sub}(\mathcal{C})$ as a mathematical environment with more structure and still always draw useful conclusions in $\mathcal{C}$ itself.

Example 2.4.9

Applying $\mathrm{Sub}(-)$ to our standard examples of additive precausal categories:

- $\mathrm{Sub}(\mathrm{Mat}[\mathbb{R}^+]) \cong \mathrm{Mat}[\mathbb{R}]$ since for each dimension the standard basis vectors have non-negative entries, i.e. they live in $\mathrm{Mat}[\mathbb{R}^+]$, but under linear combination can generate arbitrary real-valued vectors;

- $\mathrm{Sub}(\mathrm{Mat}[\mathbb{R}]) \cong \mathrm{Mat}[\mathbb{R}]$ since $\mathrm{Mat}[\mathbb{R}]$ already contains all negations;

- Looking at CP^*, each elementary object $\mathcal{L}(\mathbb{C}^n)$ is a vector space of dimension n^2 in which the completely positive maps span only the Hermitian maps under linear combination, forming a real vector space despite being built from complex matrices. For biproducts, the complete positivity requirements act independently on each subspace, so we still obtain real vector spaces. We then obtain arbitrary finite dimensions as biproducts of $\mathcal{L}(\mathbb{C})$, concluding that $\mathrm{Sub}(\mathrm{CP}^*)$ is also equivalent to $\mathrm{Mat}[\mathbb{R}]$.

The next few results all show that a lot of categorical structure lifts from $\mathcal{C}$ to $\mathrm{Sub}(\mathcal{C})$.

Proposition 2.4.10

The embedding $[-] : \mathcal{C} \to \mathrm{Sub}(\mathcal{C})$ is strong monoidal, where the monoidal product in $\mathrm{Sub}(\mathcal{C})$ acts on objects identically to $\mathcal{C}$ and on morphisms as $(f, g) \otimes (x, y) \coloneqq (f \otimes x + g \otimes y, f \otimes y + g \otimes x)$.

Proof. Firstly, this monoidal product is well-defined, i.e. if $(f, g) \sim (f', g')$ and $(x, y) \sim (x', y')$ then $(f, g) \otimes (x, y) \sim (f', g') \otimes (x', y')$. This can be proved identically to the case for sequential composition.

We inherit the unit, associators, and unitors from $\mathcal{C}$ as I, $[\alpha_{A,B,C}] : (A \otimes B) \otimes C \to A \otimes (B \otimes C)$, $[\lambda_A] : I \otimes A \to A$, $[\rho_A] : A \otimes I \to A$. These will still be invertible by functoriality of the embedding. For naturality:

$$\begin{aligned}
&((f, g) \otimes (x, y)) \otimes (u, v) \fatsemi (\alpha_{D,E,F}, 0) \\
\sim &\begin{pmatrix} ((f \otimes x + g \otimes y) \otimes u + (f \otimes y + g \otimes x) \otimes v) \fatsemi \alpha_{D,E,F}, \\ ((f \otimes x + g \otimes y) \otimes v + (f \otimes y + g \otimes x) \otimes u) \fatsemi \alpha_{D,E,F} \end{pmatrix} \\
\sim &\begin{pmatrix} \alpha_{A,B,C} \fatsemi (f \otimes (x \otimes u + y \otimes v) + g \otimes (x \otimes v + y \otimes u)), \\ \alpha_{A,B,C} \fatsemi (f \otimes (x \otimes v + y \otimes u) + g \otimes (x \otimes u + y \otimes v)) \end{pmatrix} \\
\sim &(\alpha_{A,B,C}, 0) \fatsemi (f, g) \otimes ((x, y) \otimes (u, v))
\end{aligned} \tag{2.76}$$

$$\begin{aligned}
&(f, g) \otimes (\mathrm{id}_I, 0_{I,I}) \fatsemi (\rho_B, 0_{B \otimes I,B}) \\
\sim &(f \otimes \mathrm{id}_I + g \otimes 0_{I,I}, f \otimes 0_{I,I} + g \otimes \mathrm{id}_I) \fatsemi (\rho_B, 0_{B \otimes I,B}) \\
\sim &(f \otimes \mathrm{id}_I, g \otimes \mathrm{id}_I) \fatsemi (\rho_B, 0_{B \otimes I,B}) \\
\sim &((f \otimes \mathrm{id}_I) \fatsemi \rho_B, (g \otimes \mathrm{id}_I) \fatsemi \rho_B) \\
\sim &(\rho_A \fatsemi f, \rho_A \fatsemi g) \\
\sim &(\rho_A, 0_{A \otimes I,A}) \fatsemi (f, g)
\end{aligned} \tag{2.77}$$

and similarly for $(\lambda_A, 0_{I \otimes A,A})$. In the above, we have used compact closure of $\mathcal{C}$ to infer that $f \otimes 0_{C,D} = 0_{A \otimes C,B \otimes D}$. We may also inherit the triangle and pentagon equations by functoriality of the embedding. We now have that $\mathrm{Sub}(\mathcal{C})$ is a monoidal category.

For the embedding to be strong monoidal, the maps $[-]_0 : I^{\mathrm{Sub}(\mathcal{C})} \xrightarrow{\cong} [I^{\mathcal{C}}]$ and $([-]_2)_{A,B} : [A] \otimes [B] \xrightarrow{\cong} [A \otimes B]$ are given by identities. Naturality of $[-]_2$ is just $(f, 0) \otimes (g, 0) \sim (f \otimes g + 0 \otimes 0, f \otimes 0 + 0 \otimes g) \sim (f \otimes g, 0)$. The compatibility equations reduce to $\alpha_{[A],[B],[C]} \sim [\alpha_{A,B,C}]$, $\lambda_{[A]} \sim [\lambda_A]$ and $\rho_{[A]} \sim [\rho_A]$. □

Proposition 2.4.11

$\mathrm{Sub}\left(\mathcal{C}\right)$ is compact closed, and the embedding from $\mathcal{C}$ preserves symmetry, duals, cups, and caps.

Proof. The symmetry in $\mathrm{Sub}\left(\mathcal{C}\right)$ is $[\sigma_{A,B}]$. The hexagon equation and invertibility follow by functoriality. For naturality:

$$\begin{aligned}
&(f,g) \otimes (x,y) ⨾ (\sigma_{C,D}, 0) \\
\sim\ & (f \otimes x + g \otimes y, f \otimes y + g \otimes x) ⨾ (\sigma_{C,D}, 0) \\
\sim\ & ((f \otimes x) ⨾ \sigma_{C,D} + (g \otimes y) ⨾ \sigma_{C,D}, (f \otimes y) ⨾ \sigma_{C,D} + (g \otimes x) ⨾ \sigma_{C,D}) \\
\sim\ & (\sigma_{A,B} ⨾ (x \otimes f) + \sigma_{A,B} ⨾ (y \otimes g), \sigma_{A,B} ⨾ (x \otimes g) + \sigma_{A,B} ⨾ (y \otimes f)) \\
\sim\ & (\sigma_{A,B}, 0) ⨾ (x \otimes f + y \otimes g, x \otimes g + y \otimes f) \\
\sim\ & (\sigma_{A,B}, 0) ⨾ (x,y) \otimes (f,g)
\end{aligned} \tag{2.78}$$

We appeal to the standard result that monoidal functors preserve duals [63, Theorem 3.14], which constructs the cup $[-]_0 ⨾ [\eta_A] ⨾ ([-]_2)^{-1}_{A^*,A} \sim [\eta_A] : I \to A^* \otimes A$ and cap $([-]_2)_{A^*,A} ⨾ [\epsilon_A] ⨾ [-]_0^{-1} \sim [\epsilon_A] : A \otimes A^* \to I$ for all objects of $\mathrm{Sub}\left(\mathcal{C}\right)$ since the embedding is bijective on objects. Hence, $\mathrm{Sub}\left(\mathcal{C}\right)$ is compact closed. □

Proposition 2.4.12

$\mathrm{Sub}\left(\mathcal{C}\right)$ has biproducts, and the embedding from $\mathcal{C}$ preserves the biproduct structure.

Proof. We can inherit the biproduct $A \oplus B$ with injections $[\iota_A] : A \to A \oplus B$, $[\iota_B] : B \to A \oplus B$ and projections $[p_A] : A \oplus B \to A$, $[p_B] : A \oplus B \to B$. The characteristic equations

$$\iota_{A_i} ⨾ p_{A_j} = \begin{cases} \mathrm{id}_{A_i} & i = j \\ 0_{A_i,A_j} & i \neq j \end{cases} \tag{2.79}$$

$$p_A ⨾ \iota_A + p_B ⨾ \iota_B = \mathrm{id}_{A \oplus B} \tag{2.80}$$

are preserved by functoriality. □

The next few proofs show that the ordering and invertibility of scalars from APC4 also lift to $\mathrm{Sub}\left(\mathcal{C}\right)$.

Lemma 2.4.13

Every scalar in $\mathrm{Sub}(\mathcal{C})$ is either a scalar of $\mathcal{C}$ or a negation of one: $\forall s \in \mathrm{Sub}(\mathcal{C})(I,I).\ \exists \lambda \in \mathcal{C}(I,I).\ s \sim [\lambda] \vee s \sim -[\lambda]$.

Proof. Pick some candidate representatives $s \sim [s^+] - [s^-]$. As $\mathcal{C}(I,I)$ is totally ordered, either $s^+ \geq s^-$ or $s^+ \leq s^-$. If there exists some λ such that $s^+ = s^- + \lambda$ then $s \sim [s^-] + [\lambda] - [s^-] \sim [\lambda]$, and similarly for the other ordering. □

Proposition 2.4.14

The total preorder on $\mathcal{C}(I,I)$ generalises to a preorder on any homset $\mathcal{C}(A,B)$. $\mathrm{Sub}(\mathcal{C})(A,B)$ admits a similar preorder $f \leq g \overset{\text{def}}{\iff} \exists h \in \mathcal{C}(A,B).\ f + [h] \sim g$ which satisfies the following:

1. The functor $[-] : \mathcal{C} \to \mathrm{Sub}(\mathcal{C})$ is monotone wrt the preorders.

2. $\mathrm{Sub}(\mathcal{C})(I,I)$ is totally preordered.

3. A scalar $s \in \mathrm{Sub}(\mathcal{C})(I,I)$ satisfies $0 \leq s$ iff $s \sim [\lambda]$ for some $\lambda \in \mathcal{C}(I,I)$. Hence, the following notions of convex combinations coincide:

   - $\sum_i s_i \cdot [f_i]$ with $0 \leq s_i \in \mathrm{Sub}(\mathcal{C})(I,I)$ and $\sum_i s_i \sim \mathrm{id}_I$;
   - $[\sum_i \lambda_i \cdot f_i]$ with $\lambda_i \in \mathcal{C}(I,I)$ and $\sum_i \lambda_i = \mathrm{id}_I$.

Proof.

1. If $f \leq g \in \mathcal{C}(A,B)$ then we have some $h \in \mathcal{C}(A,B)$ with $f + h = g$. Since $[-]$ preserves the additive enrichment, $[g] \sim [f+g] \sim [f] + [g]$ so $[f] \leq [g]$.

2. Given any $s, t \in \mathrm{Sub}(\mathcal{C})(I,I)$, consider the scalar $s-t$. By Lemma 2.4.13, either $s - t \sim [r]$ or $s - t \sim -[r]$ for some $r \in \mathcal{C}(I,I)$. This gives either $s \sim t + [r]$ or $s + [r] \sim t$.

3. $0 \leq s$ iff there exists some $\lambda \in \mathcal{C}(I,I)$ such that $s \sim 0 + [\lambda] \sim [\lambda]$. The equivalence between the notions of convex combinations can then be obtained by using this to map between the presentations of the scalars; pulling the sum equation back into $\mathcal{C}(I,I)$ also uses the preservation of the additive enrichment and faithfulness of the embedding $[-]$. □

Proposition 2.4.15

For any additive precausal category $\mathcal{C}$, the scalars $K := \mathrm{Sub}(\mathcal{C})(I, I)$ are a field, and hence $\mathrm{Sub}(\mathcal{C})$ is enriched over K-vector spaces.

Proof. As the subtractive closure of the semiring $\mathcal{C}(I, I)$, we already know that K is a ring, so it suffices to show that K has multiplicative inverses. Take a non-zero element $k \in K$. By Lemma 2.4.13, it is either $[\lambda]$ or $-[\lambda]$ for some $\lambda \in \mathcal{C}(I, I)$, which must also be non-zero by faithfulness of the embedding. Hence, we can invert it by APC4 and take k^{-1} to be $[\lambda^{-1}]$ or $-[\lambda^{-1}]$ as appropriate. Enrichment in K-vector spaces then follows immediately. □

$\mathrm{Sub}(\mathcal{C})$ trivially has complements for all morphisms in the sense of APC5. We can still extend APC5 to all morphisms of $\mathcal{C}$ by using compact closure to turn them into effects and back again.

Lemma 2.4.16

In an additive precausal category, all morphisms $f : A \to B$ have a complement $f' : A \to B$ with respect to the *uniform noise process* $\overset{=}{\top}_A \fatsemi \underset{=}{\bot}_B$; that is, there exists a scalar λ such that:

$$f + f' = \lambda \cdot \overset{=}{\top}_A \fatsemi \underset{=}{\bot}_B \tag{APC5a}$$

Proof. We can first make f into an effect $\pi_f : A \otimes B^* \to I$ using the compact structure: $\pi_f := (f \otimes \mathrm{id}_{B^*}) \fatsemi \epsilon_B$. Then, applying APC5, we get a complement π' satisfying $\pi_f + \pi' = \lambda \cdot \left(\overset{=}{\top}_A \otimes \overset{=}{\top}_{B^*}\right)$. Then, taking $f' := (\mathrm{id}_A \otimes \eta_B) \fatsemi (\pi' \otimes \mathrm{id}_B)$ satisfies the required property APC5a. □

The final results of this section concern the treatment of bases in APC3. In particular, they are preserved by subtractive closure, and more usefully that $\mathrm{Sub}(\mathcal{C})$ is expressive enough to have dual effects for any linearly independent set of states.

Proposition 2.4.17

Any set of states $\{\rho_i\}_{i \in \mathfrak{B}_A^{\mathcal{C}}} \subseteq \mathcal{C}(I, A)$ is a basis for A in $\mathcal{C}$ (in the sense of being a minimal set of states that can distinguish any morphisms $A \to B$ as in APC3) iff $\{[\rho_i]\}_{i \in \mathfrak{B}_A^{\mathcal{C}}}$ is a basis for A in $\mathrm{Sub}(\mathcal{C})$.

Proof. It is sufficient to just show equivalence between the ability to distinguish morphisms in each category, since preservation of minimality follows as a basic consequence of this (any subset of the $\{\rho_i\}_{i \in \mathfrak{B}^{\mathcal{C}}_A}$ would distinguish all morphisms in $\mathcal{C}$ iff it does so in $\text{Sub}(\mathcal{C})$).

$\Longrightarrow$: Suppose $\{\rho_i\}_{i \in \mathfrak{B}^{\mathcal{C}}_A}$ form a basis for A in $\mathcal{C}$. Consider an arbitrary B and some $(f^+, f^-), (g^+, g^-) \in \text{Sub}(\mathcal{C})(A, B)$. Suppose that $\forall i \in \mathfrak{B}^{\mathcal{C}}_A.\ [\rho_i] ⨾ (f^+, f^-) \sim [\rho_i] ⨾ (g^+, g^-)$. Unpacking this, we have $\forall i \in \mathfrak{B}^{\mathcal{C}}_A.\ \rho_i ⨾ (f^+ + g^-) = \rho_i ⨾ (g^+ + f^-)$. Since $\{\rho_i\}_{i \in \mathfrak{B}^{\mathcal{C}}_A}$ forms a basis in $\mathcal{C}$, we have $f^+ + g^- = g^+ + f^-$, i.e. $(f^+, f^-) \sim (g^+, g^-)$.

$\Longleftarrow$: Suppose $\{[\rho_i]\}_{i \in \mathfrak{B}^{\mathcal{C}}_A}$ forms a basis for A in $\text{Sub}(\mathcal{C})$. Consider an arbitrary B and some $f, g \in \mathcal{C}(A, B)$ such that $\forall i \in \mathfrak{B}^{\mathcal{C}}_A.\ \rho_i ⨾ f = \rho_i ⨾ g$. By functoriality of the embedding we get $\forall i \in \mathfrak{B}^{\mathcal{C}}_A.\ [\rho_i] ⨾ [f] \sim [\rho_i] ⨾ [g]$, and then we can use the basis property to derive that $[f] \sim [g]$, which implies $f = g$ by faithfulness. □

Lemma 2.4.18

Given any set of states in $\mathcal{C}(I, A)$ that are linearly independent in $\text{Sub}(\mathcal{C})$, they can be extended to a basis in $\mathcal{C}$ with a dual basis in $\text{Sub}(\mathcal{C})$.

Proof. Let $\{\rho_i\}_{i \in \mathfrak{B}^{\mathcal{C}}_A} \subseteq \mathcal{C}(I, A)$ be a minimal finite basis which must exist wlog from APC3. We will start by showing that this has a dual basis, i.e. a set of effects $\{e_i\}_{i \in \mathfrak{B}^{\mathcal{C}}_A} \subseteq \text{Sub}(\mathcal{C})(A, I)$ such that:

$$[\rho_i] ⨾ e_j \sim \delta_{i,j} \sim \begin{cases} \text{id}_I & i = j \\ 0_{I,I} & i \neq j \end{cases} \tag{2.81}$$

The vector of scalars $\rho_i ⨾ \pi$ must uniquely identify any effect $\pi \in \mathcal{C}(A, I)$, giving us a coordinate system for effects. Similarly, the set of effects $\{\pi_j\}_{j \in \mathfrak{B}^{\mathcal{C}}_{A^*}} \subseteq \mathcal{C}(A, I)$ formed as the transpose of a minimal basis for A^* from APC3 can yield coordinates $\rho ⨾ \pi_j$ that uniquely describe any state $\rho \in \mathcal{C}(I, A)$.

We can build a matrix of the scalars (in $\text{Sub}(\mathcal{C})$) formed by the inner products $m_{i,j} = [\rho_i ⨾ \pi_j]$, so the rows describe coordinates of the states $\{\rho_i\}_{i \in \mathfrak{B}^{\mathcal{C}}_A}$ and columns for the effects $\{\pi_j\}_{j \in \mathfrak{B}^{\mathcal{C}}_{A^*}}$. Performing column operations such as rescaling by a non-zero scalar s or summing columns j and k generates the coordinates of the effect $s \cdot \pi_j$ or $\pi_j + \pi_k$. By cancellative addition and invertibility of non-zero scalars from APC4, we can still represent each effect in $\{\pi_j\}_{j \in \mathfrak{B}^{\mathcal{C}}_{A^*}}$ as a linear combination of the new column effects. Since the scalars of $\text{Sub}(\mathcal{C})$ form a field, we can apply Gaussian elimination with column operations to yield column effects $\{e_j\}_{j \in \mathfrak{B}^{\mathcal{C}}_{A^*}} \subseteq \mathrm{Sub}(\mathcal{C})(A, I)$ such that $[\rho_i] \fatsemi e_j \sim \delta_{i,j}$. Gaussian elimination completes since zero columns would imply $\{\pi_j\}_{j \in \mathfrak{B}^{\mathcal{C}}_{A^*}}$ were linearly dependent and unsolved rows would similarly contradict minimality of $\{\rho_i\}_{i \in \mathfrak{B}^{\mathcal{C}}_{A}}$, and by the same argument the matrix is square ($\dim(A) = \dim(A^*)$) so we can reindex the column effects as $\{e_i\}_{i \in \mathfrak{B}^{\mathcal{C}}_{A}}$.

The new column effects still represent a basis for A^* in $\mathrm{Sub}(\mathcal{C})$ under transposition. The $\{[\pi_j^*]\}_{j \in \mathfrak{B}^{\mathcal{C}}_{A^*}}$ form a basis for A^* in $\mathrm{Sub}(\mathcal{C})$ by Proposition 2.4.17, and one output of Gaussian elimination (with back substitution) is a matrix of constants $\{s_{i,j}\}_{i \in \mathfrak{B}^{\mathcal{C}}_{A}, j \in \mathfrak{B}^{\mathcal{C}}_{A^*}} \subseteq \mathrm{Sub}(\mathcal{C})(I, I)$ such that $[\pi_j] \sim \sum_{i \in \mathfrak{B}^{\mathcal{C}}_{A}} s_{i,j} \cdot e_i$. Hence, if any two morphisms $f, g \in \mathrm{Sub}(\mathcal{C})(A^*, B)$ agree on $\{e_i^*\}_{i \in \mathfrak{B}^{\mathcal{C}}_{A}}$, they must agree on any linear combination, meaning they agree on each of $\{[\pi_j^*]\}_{j \in \mathfrak{B}^{\mathcal{C}}_{A^*}}$, and so they are equal. Minimality comes from the need to distinguish $\{\rho_i\}_{i \in \mathfrak{B}^{\mathcal{C}}_{A}}$ under transposition (if some e_i^* were excluded, the remainder would not be able to determine $[\rho_i^*] \not\sim [0_{A^*,I}] \in \mathrm{Sub}(\mathcal{C})(A^*, I)$ since all inner products will be zero in both cases). We now have that $\{e_i\}_{i \in \mathfrak{B}^{\mathcal{C}}_{A}}$ is a dual basis to $\{\rho_i\}_{i \in \mathfrak{B}^{\mathcal{C}}_{A}}$.

Given any set of states $\{v_k\}_k \subseteq \mathcal{C}(I, A)$ that are linearly independent in $\mathrm{Sub}(\mathcal{C})$, we can extend the set to $\{v_i'\}_{i \in \mathfrak{B}^{\mathcal{C}}_{A}}$ by adding any terms from $\{\rho_i\}_{i \in \mathfrak{B}^{\mathcal{C}}_{A}}$ that preserve linear independence. We can then similarly diagonalise the inner products with $\{e_i\}_{i \in \mathfrak{B}^{\mathcal{C}}_{A}}$ to represent each ρ_i as a linear combination of $\{v_i'\}_{i \in \mathfrak{B}^{\mathcal{C}}_{A}}$ and prove that they form a basis and construct a dual basis for it. □

The existence of a dual basis notably implies that any state $\rho \in \mathrm{Sub}(\mathcal{C})(I, A)$ can be uniquely characterised by an expansion in terms of basis states $\sum_{i \in \mathfrak{B}^{\mathcal{C}}_{A}} \alpha_i \cdot [\rho_i]$, making our notion of a basis fit the usual linear algebraic definitions.

2.4.3 Impacts on $\mathrm{Caus}[\mathcal{C}]$

Using an additive precausal category makes it easier to devise interesting closed sets or interpret the impact of the closure operator since it just corresponds to taking affine combinations of states. For this to make sense, we should say precisely what we mean to take affine combinations of states in $\mathcal{C}$.

Definition 2.4.19: Affine closure

For a set of states $c \subseteq \mathcal{C}(I, A)$, we define sets $\mathsf{aff}(c) \subseteq \mathrm{Sub}(\mathcal{C})(I, A)$ and $\mathsf{aff}^+(c) \subseteq \mathcal{C}(I, A)$ as follows:

$$\mathsf{aff}(c) := \left\{ \rho \in \mathrm{Sub}\left(\mathcal{C}\right)\left(I, A\right) \;\middle|\; \begin{array}{c} \exists \left\{\rho_i\right\}_i \subseteq c, \left\{s_i\right\}_i \subseteq \mathrm{Sub}\left(\mathcal{C}\right)\left(I, I\right). \\ \sum_i s_i \sim \mathrm{id}_I, \rho \sim \sum_i s_i \cdot \left[\rho_i\right] \end{array} \right\} \tag{2.82}$$

$$\mathsf{aff}^{+}(c) := \left\{\rho \in \mathcal{C}\left(I, A\right) \mid \left[\rho\right] \in \mathsf{aff}\left(c\right)\right\} \tag{2.83}$$

If we identify the set $\mathcal{C}\left(I, A\right)$ with its image under $[-]$, we can think if $\mathsf{aff}^{+}\left(c\right)$ as the intersection of the affine closure of c with the set $\mathcal{C}\left(I, A\right) \subseteq \mathrm{Sub}\left(\mathcal{C}\right)\left(I, A\right)$ of "positive" states embedded in the subtractive closure. In the classical and quantum cases, $\mathsf{aff}^{+}\left(-\right)$ arises from taking all the affine combinations of elements of c, then intersecting the resulting set with the positive cone of (unnormalised) probability distributions or quantum states, respectively.

To help prove that $\left(-\right)^{**}$ is affine closure, we attribute each flat set c with a *preferred basis* which partitions into a subset which spans c and those disjoint from c.

Definition 2.4.20: Preferred basis

A *preferred basis* for a flat set of states $c \subseteq \mathcal{C}\left(I, A\right)$ is a basis $\left\{\rho_i\right\}_{i \in \mathfrak{B}^{\mathcal{C}}_{A}} \subseteq \mathcal{C}\left(I, A\right)$ such that $\forall i \in \mathfrak{B}^{\mathcal{C}}_{A}.\ \rho_i \fatsemi \theta \cdot \overset{=}{\top}_A = \mathrm{id}_I$ (where $\theta \cdot \overset{=}{\top}_A \in c^*$ by flatness) containing a maximal linearly independent subset of c. These always exist by Lemma 2.4.18, picking the additional states from the default first-order causal basis (APC3) and renormalising. We write $\mathfrak{B}_c := \left\{i \in \mathfrak{B}^{\mathcal{C}}_{A} \mid \rho_i \in c\right\}$ for the indices of the basis elements in c and $\overline{\mathfrak{B}}_c = \mathfrak{B}^{\mathcal{C}}_{A} \setminus \mathfrak{B}_c$.

A *preferred basis* for $\mathbf{A} \in \mathrm{Ob}\left(\mathrm{Caus}\left[\mathcal{C}\right]\right)$ is a preferred basis for $c_{\mathbf{A}}$ and we write $\mathfrak{B}_{\mathbf{A}} := \mathfrak{B}_{c_{\mathbf{A}}}$ and $\overline{\mathfrak{B}}_{\mathbf{A}} := \overline{\mathfrak{B}}_{c_{\mathbf{A}}}$.

Example 2.4.21

Let's look at some preferred bases for some simple objects. For any first-order object $\mathbf{A^1}$, APC3 immediately gives us a basis for $A \in \mathrm{Ob}\left(\mathcal{C}\right)$ where every state is causal. For example, the binary object $\mathbf{2}$ has a preferred basis given by the two boolean states $\mathfrak{B}_{\mathbf{2}} = \mathfrak{B}^{\mathcal{C}}_{2} = \left\{\mathtt{t}, \mathtt{f}\right\}$. By definition, we then have $\overline{\mathfrak{B}}_{\mathbf{2}} = \varnothing$.

Dually, $\mathbf{2}^*$ has a single causal state, $\overset{\shortmid}{\bullet}_{\mathbf{2}^*}$, so $\mathfrak{B}_{\mathbf{2}^*} = \left\{\underset{=}{\bot}_{2^*}\right\}$. We can then pick any other state of $2^* \cong 2 \in \mathrm{Ob}\left(\mathcal{C}\right)$ with the same normalisation to extend this to a full basis, such as $\mathfrak{B}^{\mathcal{C}}_{2^*} = \left\{\underset{=}{\bot}_{2^*}, \mu_{\mathbf{2}}^{-1} \cdot \mathtt{t}\right\}$, in which case $\overline{\mathfrak{B}}_{\mathbf{2}^*} = \left\{\mu_{\mathbf{2}}^{-1} \cdot \mathtt{t}\right\}$.

When we compare $\mathbf{2}^* \otimes \mathbf{2}$ and $\mathbf{2}^* \parr \mathbf{2} = \mathbf{2} \multimap \mathbf{2}$, we know that they share the same underlying object $2^* \otimes 2 \in \mathrm{Ob}\left(\mathcal{C}\right)$ and normalisation scalars, so let's try constructing a single basis for $2^* \otimes 2$ that is preferred by both $\mathbf{2}^* \otimes \mathbf{2}$ and $\mathbf{2}^* \parr \mathbf{2}$.

From the definition of $\otimes$, we can start with the separable states consisting of the elements from preferred bases of $\mathbf{2}^*$ and $\mathbf{2}$.

$$\left\{\mathtt{t} \otimes \underset{=}{\bot}_{2^*}, \mathtt{f} \otimes \underset{=}{\bot}_{2^*}, \mathtt{t} \otimes \mu_{\mathbf{2}}^{-1} \cdot \mathtt{t}, \mathtt{f} \otimes \mu_{\mathbf{2}}^{-1} \cdot \mathtt{t}\right\} \tag{2.84}$$

The first two of these generate the states of $\mathbf{2}^* \otimes \mathbf{2}$, but this is not preferred for $\mathbf{2}^* \parr \mathbf{2}$: neither of the latter two states are causal for $\mathbf{2}^* \parr \mathbf{2}$ and yet they are both required to generate $(\mathtt{t} \otimes \mathtt{t}) + (\mathtt{f} \otimes \mathtt{f}) : \mathbf{2}^* \parr \mathbf{2}$ (the Choi operator of the identity). If we replace one of the basis elements with this term, we reach a point where three basis elements generate every state of $\mathbf{2}^* \parr \mathbf{2}$.

$$\mathfrak{B}^{\mathcal{C}}_{2^* \otimes 2} = \left\{\mathtt{t} \otimes \underset{=}{\bot}_{2^*}, \mathtt{f} \otimes \underset{=}{\bot}_{2^*}, (\mathtt{t} \otimes \mathtt{t}) + (\mathtt{f} \otimes \mathtt{f}), \mathtt{t} \otimes \mu_{\mathbf{2}}^{-1} \cdot \mathtt{t}\right\} \tag{2.85}$$

$$\mathfrak{B}_{\mathbf{2}^* \otimes \mathbf{2}} = \left\{\mathtt{t} \otimes \underset{=}{\bot}_{2^*}, \mathtt{f} \otimes \underset{=}{\bot}_{2^*}\right\} \tag{2.86}$$

$$\overline{\mathfrak{B}}_{\mathbf{2}^* \otimes \mathbf{2}} = \left\{(\mathtt{t} \otimes \mathtt{t}) + (\mathtt{f} \otimes \mathtt{f}), \mathtt{t} \otimes \mu_{\mathbf{2}}^{-1} \cdot \mathtt{t}\right\} \tag{2.87}$$

$$\mathfrak{B}_{\mathbf{2}^* \parr \mathbf{2}} = \left\{\mathtt{t} \otimes \underset{=}{\bot}_{2^*}, \mathtt{f} \otimes \underset{=}{\bot}_{2^*}, (\mathtt{t} \otimes \mathtt{t}) + (\mathtt{f} \otimes \mathtt{f})\right\} \tag{2.88}$$

$$\overline{\mathfrak{B}}_{\mathbf{2}^* \parr \mathbf{2}} = \left\{\mathtt{t} \otimes \mu_{\mathbf{2}}^{-1} \cdot \mathtt{t}\right\} \tag{2.89}$$

Appendix B presents Python code for exploring Caus$[\mathrm{Mat}[\mathbb{R}]]$, where we represent each type $\mathbf{A}$ by a matrix whose column vectors describe $\mathfrak{B}_{\mathbf{A}}$ for some choice of preferred basis. It contains a number of additional examples of preferred bases for different objects, and the code demonstrates how one can algorithmically generate them.

Lemma 2.4.22

A preferred basis $\{\rho_i\}_{i \in \mathfrak{B}^{\mathcal{C}}_A}$ for a flat set $c \subseteq \mathcal{C}(I, A)$ $(\theta \cdot \overset{=}{\top}_A \in c^*)$ with a dual basis $\{e_i\}_{i \in \mathfrak{B}^{\mathcal{C}}_A}$ (as in Lemma 2.4.18) satisfies:

PB0. $\forall i \in \overline{\mathfrak{B}}_c.\ \forall v \in c.\ [v] \mathbin{;} e_i \sim 0$;

PBΣ. $\left[\theta \cdot \overset{=}{\top}_A\right] \sim \sum_{i \in \mathfrak{B}^{\mathcal{C}}_A} e_i$;

PBc. $\{\rho_i \mid i \in \mathfrak{B}_c\}^{**} = c^{**}$;

PB$*$. $c^* = \left\{\pi \in \mathcal{C}(A, I) \;\middle|\; \exists \{s_i\}_{i \in \overline{\mathfrak{B}}_c} \subseteq \mathrm{Sub}(\mathcal{C})(I, I).\ [\pi] \sim \left[\theta \cdot \overset{=}{\top}_A\right] + \sum_{i \in \overline{\mathfrak{B}}_c} s_i \cdot e_i\right\}$;

PB$\mathcal{C}$. $\forall i \in \overline{\mathfrak{B}}_c.\ \exists \pi \in c^*, \alpha \in \mathcal{C}(I, I).\ \alpha \neq 0 \wedge [\pi] \sim \left[\theta \cdot \overset{=}{\top}_A\right] + [\alpha] \cdot e_i$.

Proof. PB0: Since the basis contains a maximal linearly independent subset of c, we must be able to express any $v \in c$ as a linear combination $[v] \sim \sum_{i \in \mathfrak{B}_c} s_i \cdot [\rho_i]$, and hence for any $i \in \overline{\mathfrak{B}}_c$. $[v] \fatsemi e_i \sim 0$.

PB$\sum$: This follows by a resolution of the identity and normalisation of the basis elements.

$$\begin{aligned} \left[\theta \cdot \bar{\bar{\top}}_A\right] &\sim \left(\sum_{i \in \mathfrak{B}_A^{\mathcal{C}}} e_i \fatsemi [\rho_i] \right) \fatsemi \left[\theta \cdot \bar{\bar{\top}}_A\right] \\ &\sim \sum_{i \in \mathfrak{B}_A^{\mathcal{C}}} e_i \fatsemi \mathrm{id}_I \\ &\sim \sum_{i \in \mathfrak{B}_A^{\mathcal{C}}} e_i \end{aligned} \tag{2.90}$$

PBc: We will show the dual, $\{\rho_i \mid i \in \mathfrak{B}_c\}^* = c^*$. The $\supseteq$ direction is immediate from Equation 2.48. If an effect π satisfies $\rho_i \fatsemi \pi = 0$ for any $\rho_i \in c$, then it also satisfies this for any linear combinations, and any state in c is expressible in this way by maximality of the linearly independent subset contained in the basis.

PB$*$: Since $\{e_i\}_{i \in \mathfrak{B}_A^{\mathcal{C}}}$ form a basis for $\mathrm{Sub}(\mathcal{C})(I, A^*)$ under transposition, any effect in $\mathrm{Sub}(\mathcal{C})(A, I)$ has a unique expression as a linear combination of them. Given the dual of PBc, each condition $\rho_i \fatsemi \pi = \mathrm{id}_I$ (from $\pi \in c^*$) is equivalent to fixing one of the coefficients in this linear combination to match that of $\theta \cdot \bar{\bar{\top}}_A \in c^*$. This leaves the coefficients of each index in $\overline{\mathfrak{B}}_c$ as free.

PB$\mathcal{C}$: Choose some representative decomposition $e_i \sim [e_i^+] - [e_i^-]$ with $e_i^+, e_i^- \in \mathcal{C}(A, I)$. By APC5, there is some invertible $\lambda \in \mathcal{C}(I, I)$ and $e' \in \mathcal{C}(A, I)$ such that $e' + e_i^- = \lambda \cdot \bar{\bar{\top}}_A$, i.e.

$$\left[\theta\lambda^{-1} \cdot (e' + e_i^+)\right] \sim \left[\theta \cdot \bar{\bar{\top}}_A\right] + \left[\theta\lambda^{-1}\right] \cdot e_i \tag{2.91}$$

This is in c^* by PB$*$. □

Theorem 2.4.23: Affine Closure Theorem

Given any flat set $c \subseteq \mathcal{C}(I, A)$ for a non-zero A, $c^{**} = \mathsf{aff}^+(c)$.

Proof. Fix a preferred basis $\{\rho_i\}_{i \in \mathfrak{B}_A^{\mathcal{C}}}$ for c with a dual basis $\{e_i\}_{i \in \mathfrak{B}_A^{\mathcal{C}}}$.

$\supseteq$: For any affine combination $\sum_{i \in \mathfrak{B}_c} s_i \cdot [\rho_i]$ and $\pi \in c^*$, linearity gives

$$\begin{aligned}\left(\sum_{i \in \mathfrak{B}_c} s_i \cdot [\rho_i]\right) \fatsemi \pi &= \sum_{i \in \mathfrak{B}_c} s_i \cdot [\rho_i \fatsemi \pi] \\ &= \sum_{i \in \mathfrak{B}_c} s_i \\ &= [\mathrm{id}_I]\end{aligned} \tag{2.92}$$

Hence any $\upsilon \in \mathcal{C}(I, A)$ that is such an affine combination in $\mathrm{Sub}(\mathcal{C})$ is in c^{**}.

$\subseteq$: Consider an arbitrary $\upsilon \in c^{**}$ and $i \in \overline{\mathfrak{B}}_c$. By PB$\mathcal{C}$, let $\pi \in c^*$ satisfy $[\pi] \sim \left[\theta \cdot \overset{=}{\top}_A\right] + [\alpha] \cdot e_i$.

$$\begin{aligned}[\upsilon] \fatsemi e_i &\sim \left[\alpha^{-1} \cdot \upsilon \fatsemi \theta \cdot \overset{=}{\top}_A\right] + [\upsilon] \fatsemi e_i - \left[\alpha^{-1}\right] \\ &\sim \left[\alpha^{-1} \cdot \upsilon\right] \fatsemi \left(\left[\theta \cdot \overset{=}{\top}_A\right] + [\alpha] \cdot e_i\right) - \left[\alpha^{-1}\right] \\ &\sim \left[\alpha^{-1} \cdot \upsilon \fatsemi \pi\right] - \left[\alpha^{-1}\right] \\ &\sim \left[\alpha^{-1}\right] - \left[\alpha^{-1}\right] \\ &\sim 0\end{aligned} \tag{2.93}$$

Repeating for each $i \in \overline{\mathfrak{B}}_c$ and expanding υ, we now have $[\upsilon] \sim \sum_{i \in \mathfrak{B}^{\mathcal{C}}_A} ([\upsilon] \fatsemi e_i) \cdot [\rho_i] \sim \sum_{i \in \mathfrak{B}_c} ([\upsilon] \fatsemi e_i) \cdot [\rho_i]$, i.e. υ is a linear combination of terms in c. This combination is affine since:

$$\begin{aligned}\sum_{i \in \mathfrak{B}_c} [\upsilon] \fatsemi e_i &\sim \sum_{i \in \mathfrak{B}_c} ([\upsilon] \fatsemi e_i) \cdot \left([\rho_i] \fatsemi \left[\theta \cdot \overset{=}{\top}_A\right]\right) \\ &\sim \left(\sum_{i \in \mathfrak{B}_c} ([\upsilon] \fatsemi e_i) \cdot [\rho_i]\right) \fatsemi \left[\theta \cdot \overset{=}{\top}_A\right] \\ &\sim [\upsilon] \fatsemi \left[\theta \cdot \overset{=}{\top}_A\right] \\ &\sim \mathrm{id}_I\end{aligned} \tag{2.94}$$

□

With this in mind, we can now rephrase each of the operators in $\mathrm{Caus}[\mathcal{C}]$ in terms of the preferred bases of the component systems.

Proposition 2.4.24

Let $\{\rho_i\}_{i \in \mathfrak{B}^{\mathcal{C}}_A}$ be a preferred basis for $\mathbf{A} \in \mathrm{Ob}(\mathrm{Caus}[\mathcal{C}])$ with dual basis $\{e_i\}_{i \in \mathfrak{B}^{\mathcal{C}}_A}$. For each $i \in \overline{\mathfrak{B}}_{\mathbf{A}}$, let $\pi_i \in c^*$ be an effect such that $[\pi_i] \sim [\overset{\bullet}{\mid}_{\mathbf{A}}] + [\alpha_i] \cdot e_i$ from PB$\mathcal{C}$. Then

$$c^*_{\mathbf{A}} = \left(\{\overset{\bullet}{\mid}_{\mathbf{A}}\} \cup \{\pi_i\}_{i \in \overline{\mathfrak{B}}_{\mathbf{A}}}\right)^{**}$$

Proof. Affine combinations allow us to generate any expression $[\text{⫯}_{\mathbf{A}}]+\sum_{i\in\overline{\mathfrak{B}_{\mathbf{A}}}} s_i{\cdot}e_i$, coinciding with c^* by PB$\ast$. □

Proposition 2.4.25

Let $\{\rho_i^A\}_{i\in\mathfrak{B}_A^{\mathcal{C}}}$ and $\left\{\rho_j^B\right\}_{j\in\mathfrak{B}_B^{\mathcal{C}}}$ be preferred bases for objects $\mathbf{A},\mathbf{B}\in\mathrm{Ob}\left(\mathrm{Caus}\left[\mathcal{C}\right]\right)$. Then

$$c_{\mathbf{A}\otimes\mathbf{B}}=\left\{h\;\middle|\;\begin{array}{c}[h]\sim\sum_{i,j}s_{i,j}\cdot\left[\rho_i^A\otimes\rho_j^B\right],\\ \sum_{i,j}s_{i,j}\sim\mathrm{id}_I,\\ \forall i,j.\ i\in\overline{\mathfrak{B}_{\mathbf{A}}}\vee j\in\overline{\mathfrak{B}_{\mathbf{B}}}\implies s_{i,j}=0\end{array}\right\}$$

Proof. $\supseteq$ is immediate from the Affine Closure Theorem. For $\subseteq$, by PBc the states of $\mathbf{A}$ are affine combinations of $\{\rho_i^A\}_{i\in\mathfrak{B}_{\mathbf{A}}}$ and the same for $\mathbf{B}$. Then every product state $\rho_A\otimes\rho_B$ is an affine combination of the product basis states $\left\{\rho_i^A\otimes\rho_j^B\right\}_{i\in\mathfrak{B}_{\mathbf{A}},j\in\mathfrak{B}_{\mathbf{B}}}$. □

Proposition 2.4.26

Let $\{\rho_i^A\}_{i\in\mathfrak{B}_A^{\mathcal{C}}}$ and $\left\{\rho_j^B\right\}_{j\in\mathfrak{B}_B^{\mathcal{C}}}$ be preferred bases for objects $\mathbf{A},\mathbf{B}\in\mathrm{Ob}\left(\mathrm{Caus}\left[\mathcal{C}\right]\right)$. Then

$$c_{\mathbf{A}\parr\mathbf{B}}=\left\{h\;\middle|\;\begin{array}{c}[h]\sim\sum_{i,j}s_{i,j}\cdot\left[\rho_i^A\otimes\rho_j^B\right],\\ \sum_{i,j}s_{i,j}\sim\mathrm{id}_I,\\ \forall i,j.\ i\in\overline{\mathfrak{B}_{\mathbf{A}}}\wedge j\in\overline{\mathfrak{B}}_B\implies s_{i,j}=0\\ \forall i\in\overline{\mathfrak{B}_{\mathbf{A}}}.\ \sum_j s_{i,j}\sim 0,\\ \forall j\in\overline{\mathfrak{B}_{\mathbf{B}}}.\ \sum_i s_{i,j}\sim 0\end{array}\right\}$$

Proof. Consider an arbitrary $h\in\mathcal{C}\left(I,A\otimes B\right)$ with expansion $[h]\sim\sum_{i,j}s_{i,j}\cdot\left[\rho_i^A\otimes\rho_j^B\right]$.

Equation 2.51 describes $c_{\mathbf{A}\parr\mathbf{B}}$ as those states that are causal with respect to separable effects. By Proposition 2.4.24

$$\begin{aligned}c_{\mathbf{A}}^*&=\left(\{\text{⫯}_{\mathbf{A}}\}\cup\left\{\pi_i^A\right\}_{i\in\overline{\mathfrak{B}_{\mathbf{A}}}}\right)^{**}\\ \forall i\in\overline{\mathfrak{B}_{\mathbf{A}}}.\ \left[\pi_i^A\right]&\sim[\text{⫯}_{\mathbf{A}}]+\left[\alpha_i^A\right]\cdot e_i^A\\ c_{\mathbf{B}}^*&=\left(\{\text{⫯}_{\mathbf{B}}\}\cup\left\{\pi_j^B\right\}_{j\in\overline{\mathfrak{B}_{\mathbf{B}}}}\right)^{**}\\ \forall j\in\overline{\mathfrak{B}_{\mathbf{B}}}.\ \left[\pi_j^B\right]&\sim[\text{⫯}_{\mathbf{B}}]+\left[\alpha_i^B\right]\cdot e_i^B\end{aligned}\tag{2.95}$$

We can then equate the conditions on the coefficients $\{s_{i,j}\}_{i,j}$ to causality with respect to particular effects. Firstly, the combination is affine by causality with respect to the uniform effect.

$$\begin{aligned}[h \fatsemi (\text{⫯}_{\mathbf{A}} \otimes \text{⫯}_{\mathbf{B}})] &\sim \sum_{i,j} s_{i,j} \cdot \left[\left(\rho_i^A \otimes \rho_j^B\right) \fatsemi (\text{⫯}_{\mathbf{A}} \otimes \text{⫯}_{\mathbf{B}})\right] \\ &\sim \sum_{i,j} s_{i,j} \cdot [\mathrm{id}_I \otimes \mathrm{id}_I] \\ &\sim \sum_{i,j} s_{i,j}\end{aligned} \tag{2.96}$$

$$h \fatsemi (\text{⫯}_{\mathbf{A}} \otimes \text{⫯}_{\mathbf{B}}) = \mathrm{id}_I \iff \sum_{i,j} s_{i,j} \sim \mathrm{id}_I \tag{2.97}$$

Under the assumption that $\sum_{i,j} s_{i,j} \sim \mathrm{id}_I$, then each $i \in \overline{\mathfrak{B}}_{\mathbf{A}}$ gives another constraint.

$$\begin{aligned}\left[h \fatsemi \left(\pi_i^A \otimes \text{⫯}_{\mathbf{B}}\right)\right] &\sim [h] \fatsemi \left(\left([\text{⫯}_{\mathbf{A}}] + \left[\alpha_i^A\right] \cdot e_i^A\right) \otimes [\text{⫯}_{\mathbf{B}}]\right) \\ &\sim [h \fatsemi (\text{⫯}_{\mathbf{A}} \otimes \text{⫯}_{\mathbf{B}})] + \left[\alpha_i^A\right] \cdot [h] \fatsemi \left(e_i^A \otimes \sum_j e_j^B\right) \\ &\sim [\mathrm{id}_I] + \left[\alpha_i^A\right]\left(\sum_j s_{i,j}\right)\end{aligned} \tag{2.98}$$

$$h \fatsemi \left(\pi_i^A \otimes \text{⫯}_{\mathbf{B}}\right) = \mathrm{id}_I \iff \sum_j s_{i,j} \sim 0 \tag{2.99}$$

Similarly, each $j \in \overline{\mathfrak{B}}_{\mathbf{B}}$ gives

$$h \fatsemi \left(\text{⫯}_{\mathbf{A}} \otimes \pi_j^B\right) = \mathrm{id}_I \iff \sum_i s_{i,j} \sim 0 \tag{2.100}$$

Assuming all the above constraints are met, we then have a final constraint for each pair of $i \in \overline{\mathfrak{B}}_{\mathbf{A}}$ and $j \in \overline{\mathfrak{B}}_{\mathbf{B}}$.

$$\begin{aligned}\left[h \fatsemi \left(\pi_i^A \otimes \pi_j^B\right)\right] &\sim [h] \fatsemi \left(\left(\left[\alpha_i^A\right] \cdot e_i^A + \sum_{i'} e_{i'}^A\right) \otimes \left(\left[\alpha_j^B\right] \cdot e_j^B + \sum_{j'} e_{j'}^B\right)\right) \\ &\sim \left(\sum_{i',j'} s_{i',j'}\right) + \left[\alpha_j^B\right]\left(\sum_{i'} s_{i',j}\right) + \left[\alpha_i^A\right]\left(\sum_{j'} s_{i,j'}\right) + \left[\alpha_i^A \alpha_j^B\right] s_{i,j} \\ &\sim [\mathrm{id}_I] + 0 + 0 + \left[\alpha_i^A \alpha_j^B\right] s_{i,j} \\ &\sim [\mathrm{id}_I] + \left[\alpha_i^A \alpha_j^B\right] s_{i,j}\end{aligned} \tag{2.101}$$

$$h \fatsemi \left(\pi_i^A \otimes \pi_j^B\right) = \mathrm{id}_I \iff s_{i,j} = 0 \tag{2.102}$$

Causality is preserved under affine combinations of effects, so these conditions are equivalent to causality under separable effects, i.e. $h \in c_{\mathbf{A} \parr \mathbf{B}}$. □

Proposition 2.4.27

Suppose $\mathbf{A}$ and $\mathbf{A}'$ share the same underlying object $A \in \mathrm{Ob}(\mathcal{C})$ and normalisation scalars, and let $\{\rho_i\}_{i\in\mathfrak{B}^{\mathcal{C}}_{A}}$ be a preferred basis for both $\mathbf{A}$ and $\mathbf{A}'$. Then

$$c_{\mathbf{A}\cap\mathbf{A}'} = \left\{ v \;\middle|\; \begin{array}{c} [v] \sim \sum_i s_i \cdot [\rho_i], \\ \sum_i s_i \sim \mathrm{id}_I, \\ \forall i \in \overline{\mathfrak{B}_{\mathbf{A}}} \cup \overline{\mathfrak{B}_{\mathbf{A}'}}.\ s_i \sim 0 \end{array} \right\}$$

Proof. Membership in $c_{\mathbf{A}}$ and $c_{\mathbf{A}'}$ respectively mean that our state's unique expansion as an affine combination of the basis terms only uses the terms from $\mathfrak{B}_{\mathbf{A}}$ ($\forall i \in \overline{\mathfrak{B}_{\mathbf{A}}}.\ s_i \sim 0$) and only those from $\mathfrak{B}_{\mathbf{A}'}$ ($\forall i \in \overline{\mathfrak{B}_{\mathbf{A}'}}.\ s_i \sim 0$). □

A simple corollary of Proposition 2.4.25 captures what [117] refers to as the principle of "no interaction with trivial degrees of freedom". In particular, it recovers the precausal category axiom PC5 by showing that every state of $(\mathbf{A^1} \multimap \mathbf{B^1})^* = (\mathbf{A^1}^* ⅋ \mathbf{B^1})^* = \mathbf{A^1} \otimes \mathbf{B^1}^*$ decomposes into a product of a state of $\mathbf{A^1}$ and $⏚_{B^*}$.

Corollary 2.4.28

If $\mathbf{A} \in \mathrm{Ob}(\mathrm{Caus}[\mathcal{C}])$ is first-order dual ($c_{\mathbf{A}} = \{\downarrow_{\mathbf{A}}\}$), then every $h \in c_{\mathbf{A}\otimes\mathbf{B}}$ is a product morphism of the form $\downarrow_{\mathbf{A}} \otimes g$ for some $g \in c_{\mathbf{B}}$.

Proof. In any preferred basis for $\mathbf{A}$, $\{\rho^A_i\}_{i\in\mathfrak{B}_{\mathbf{A}}} = \{\downarrow_{\mathbf{A}}\}$. From Proposition 2.4.25, $[h] \sim [\downarrow_{\mathbf{A}}] \otimes \sum_{j\in\mathfrak{B}_{\mathbf{B}}} s_j \cdot [\rho^B_j]$. Then

$$\begin{aligned} \sum_{j\in\mathfrak{B}_{\mathbf{B}}} s_j \cdot [\rho^B_j] &\sim [\downarrow_{\mathbf{A}} ⨾ \uparrow_{\mathbf{A}}] \otimes \sum_{j\in\mathfrak{B}_{\mathbf{B}}} s_j \cdot [\rho^B_j] \\ &\sim \left([\downarrow_{\mathbf{A}}] \otimes \sum_{j\in\mathfrak{B}_{\mathbf{B}}} s_j \cdot [\rho^B_j]\right) ⨾ [\uparrow_{\mathbf{A}} \otimes \mathrm{id}_B] \\ &\sim [h ⨾ (\uparrow_{\mathbf{A}} \otimes \mathrm{id}_B)] \end{aligned} \tag{2.103}$$

Hence $h = \downarrow_{\mathbf{A}} \otimes (h ⨾ (\uparrow_{\mathbf{A}} \otimes \mathrm{id}_B))$ with $h ⨾ (\uparrow_{\mathbf{A}} \otimes \mathrm{id}_B) \in c_{\mathbf{B}}$ since it is expressible as an affine combination of states of $c_{\mathbf{B}}$. □

With that, we have almost completely recovered all the precausal category axioms from those of additive precausal categories. The one remaining condition, PC4, was only significant for Kissinger and Uijlen's proofs of FO$\otimes$ and FO$\multimap$ (FO⅋ did not require this so it still holds).

We can already recover FO$\otimes$ since the Affine Closure Theorem characterises the states $c_{(\mathbf{A^1}\multimap\mathbf{B^1})\otimes(\mathbf{C^1}\multimap\mathbf{D^1})}$ as the affine closure of separable processes, from which we can follow the proof of Theorem 2.1.9 to equate this with the non-signalling processes. We will later generalise this beyond first-order to arbitrary $\mathbf{A} \otimes \mathbf{B}$ (see the Non-signalling Theorem).

In Section 2.6.4 we will recover a slightly weaker version of PC4 by the equivalence of one-way signalling with the affine closure of semi-localisability (see the Seq Equivalence Theorem). Through this, we obtain another monoidal product which degenerates to $\mathbf{A^1} \multimap (\mathbf{B^1} \multimap \mathbf{C^1}) \multimap \mathbf{D^1}$ on first-order systems.

2.5 Additive Operators

As well as the monoidal product of $\mathcal{C}$ lifting to multiple non-degenerate monoidal products on $\mathrm{Caus}[\mathcal{C}]$, we can lift the biproduct structure to distinct products and coproducts, modelling the additives of linear logic. In this section, we apply the standard constructions from double-glueing and examine how the results give rise to probabilistic choice and conditional operation. These operators will be key to encoding (abstract) probability distributions in arbitrary causal categories.

2.5.1 Products and Coproducts

The following constructions for products and coproducts are the standards from double-glueing [72] adapted to account for closure, where we can either describe them in terms of states or effects. For completeness, we will explicitly prove that the constructions do in fact give categorical products and coproducts for $\mathrm{Caus}[\mathcal{C}]$, along with the equivalence between the alternative forms.

Definition 2.5.1

We define bifunctors $\times, \oplus : \mathrm{Caus}[\mathcal{C}] \times \mathrm{Caus}[\mathcal{C}] \to \mathrm{Caus}[\mathcal{C}]$ by the following actions on objects.

$$\mathbf{A} \times \mathbf{B} := \left(A \oplus B, \begin{pmatrix} \{p_A \fatsemi \pi_A \mid \pi_A \in c_{\mathbf{A}}^* \subseteq \mathcal{C}(A, I)\} \\ \cup \{p_B \fatsemi \pi_B \mid \pi_B \in c_{\mathbf{B}}^* \subseteq \mathcal{C}(B, I)\} \end{pmatrix}^* \right) \tag{2.104}$$

$$= \left(A \oplus B, \{\langle \rho_A, \rho_B \rangle \mid \rho_A \in c_{\mathbf{A}}, \rho_B \in c_{\mathbf{B}}\} \right) \tag{2.105}$$

$$\mathbf{A} \oplus \mathbf{B} := \left(A \oplus B, \left(\{\rho_A \fatsemi \iota_A \mid \rho_A \in c_{\mathbf{A}}\} \cup \{\rho_B \fatsemi \iota_B \mid \rho_B \in c_{\mathbf{B}}\} \right)^{**} \right) \tag{2.106}$$

$$= \left(A \oplus B, \{[\pi_A, \pi_B] \mid \pi_A \in c_{\mathbf{A}}^* \subseteq \mathcal{C}(A, I), \pi_B \in c_{\mathbf{B}}^* \subseteq \mathcal{C}(B, I)\}^* \right) \tag{2.107}$$

They respectively act on morphisms $f : \mathbf{A} \to \mathbf{B}$, $g : \mathbf{C} \to \mathbf{D}$ as $\langle p_A \fatsemi f, p_C \fatsemi g \rangle : \mathbf{A} \times \mathbf{C} \to \mathbf{B} \times \mathbf{D}$ and $[f \fatsemi \iota_B, g \fatsemi \iota_D] : \mathbf{A} \oplus \mathbf{C} \to \mathbf{B} \oplus \mathbf{D}$.

Proposition 2.5.2

The alternative definitions of $\mathbf{A} \times \mathbf{B}$ of 2.104 and 2.105 are equivalent and, furthermore, yield a categorical product in $\mathrm{Caus}\left[\mathcal{C}\right]$.

Proof. $\subseteq$: For any $\rho \in \left(\{p_A \,⨾\, \pi_A \mid \pi_A \in c_{\mathbf{A}}^*\} \cup \{p_B \,⨾\, \pi_B \mid \pi_B \in c_{\mathbf{B}}^*\}\right)^*$, the η-rule for products allows us to expand it as $\rho = \langle \rho \,⨾\, p_A, \rho \,⨾\, p_B \rangle$. Then $\rho \,⨾\, p_A \in c_{\mathbf{A}}$ since ρ satisfies $\rho \,⨾\, p_A \,⨾\, \pi_A = \mathrm{id}_I$ for all $\pi_A \in c_{\mathbf{A}}^*$, and similarly $\rho \,⨾\, p_B \in c_{\mathbf{B}}$.

$\supseteq$: For any states $\rho_A \in c_{\mathbf{A}}, \rho_B \in c_{\mathbf{B}}$ and effects $\pi_A \in c_{\mathbf{A}}^*, \pi_B \in c_{\mathbf{B}}^*$, we have $\langle \rho_A, \rho_B \rangle \,⨾\, p_A \,⨾\, \pi_A = \rho_A \,⨾\, \pi_A = \mathrm{id}_I$ and $\langle \rho_A, \rho_B \rangle \,⨾\, p_B \,⨾\, \pi_B = \rho_B \,⨾\, \pi_B = \mathrm{id}_I$.

To show that we have a product, suppose we are given some $f : \mathbf{C} \to \mathbf{A}$ and $g : \mathbf{C} \to \mathbf{B}$. The existence and uniqueness of $\langle f, g \rangle$ can be inherited from the fact that $A \oplus B$ is a (bi)product in $\mathcal{C}$, so we just need to show that $\langle f, g \rangle$, p_A, and p_B are all causal.

Given any state $\rho \in c_{\mathbf{C}}$, since f and g are causal we have $\rho ⨾ f \in c_{\mathbf{A}}$ and $\rho ⨾ g \in c_{\mathbf{B}}$. The product definition of $c_{\mathbf{A} \times \mathbf{B}}$ now gives that $\rho \,⨾\, \langle f, g \rangle = \langle \rho \,⨾\, f, \rho \,⨾\, g \rangle \in c_{\mathbf{A} \times \mathbf{B}}$, so $\langle f, g \rangle : \mathbf{C} \to \mathbf{A} \times \mathbf{B}$.

For the projections, we know that all states of $\mathbf{A} \times \mathbf{B}$ are of the form $\langle \rho_A, \rho_B \rangle$ for some $\rho_A \in c_{\mathbf{A}}$ and $\rho_B \in c_{\mathbf{B}}$. Since $\langle \rho_A, \rho_B \rangle \,⨾\, p_A = \rho_A \in c_{\mathbf{A}}$, we have $p_A : \mathbf{A} \times \mathbf{B} \to \mathbf{A}$ and similarly $p_B : \mathbf{A} \times \mathbf{B} \to \mathbf{B}$. □

Proposition 2.5.3

The alternative definitions of $\mathbf{A} \oplus \mathbf{B}$ of 2.106 and 2.107 are equivalent and, furthermore, yield a categorical coproduct in $\mathrm{Caus}\left[\mathcal{C}\right]$.

Proof. Dualising the proof of Proposition 2.5.2 (i.e. swapping the roles of states/products/projections with effects/coproducts/injections) gives us the required results. □

As expected, these give rise to a de Morgan duality.

Corollary 2.5.4

$$(\mathbf{A} \times \mathbf{B})^* = \mathbf{A}^* \oplus \mathbf{B}^*$$

Proof. This is immediate from the symmetric definitions of $c_{\mathbf{A} \times \mathbf{B}}$ and $c_{\mathbf{A} \oplus \mathbf{B}}$. For example,

$$
\begin{aligned}
c^*_{\mathbf{A}\oplus\mathbf{B}} &= \{[\pi_A, \pi_B] \mid \pi_A \in c^*_{\mathbf{A}} \subseteq \mathcal{C}(A, I), \pi_B \in c^*_{\mathbf{B}} \subseteq \mathcal{C}(B, I)\} \\
&= \{[\rho^*_A, \rho^*_B] \mid \rho_A \in c_{\mathbf{A}^*} \subseteq \mathcal{C}(I, A^*), \rho_B \in c_{\mathbf{B}^*} \subseteq \mathcal{C}(I, B^*)\} \\
&= \{p_A \fatsemi \rho^*_A + p_B \fatsemi \rho^*_B \mid \rho_A \in c_{\mathbf{A}^*}, \rho_B \in c_{\mathbf{B}^*}\} \\
&= \{(\rho_A \fatsemi \iota_{A^*} + \rho_B \fatsemi \iota_{B^*})^* \mid \rho_A \in c_{\mathbf{A}^*}, \rho_B \in c_{\mathbf{B}^*}\} \\
&= \{\langle \rho_A, \rho_B \rangle^* \mid \rho_A \in c_{\mathbf{A}^*}, \rho_B \in c_{\mathbf{B}^*}\} \\
&= c_{\mathbf{A}^* \times \mathbf{B}^*} \subseteq \mathcal{C}(A \oplus B, I)
\end{aligned}
\tag{2.108}
$$

Here we have used that injections and projections may be treated as transposes of one another when we suppose that the compact structure for the duality $A \oplus B \dashv A^* \oplus B^*$ in $\mathcal{C}$ is built from those of $A \dashv A^*$ and $B \dashv B^*$ in the canonical way [63]. □

2.5.2 Probabilistic Choice

The intuition for how we can think of × and ⊕ operationally follow from their standard interpretations in linear logic.

Instead of presenting a pair of interfaces in parallel, $\mathbf{A} \oplus \mathbf{B}$ and $\mathbf{A} \times \mathbf{B}$ present a single interface which can be chosen to act either like $\mathbf{A}$ or $\mathbf{B}$. The distinction is whether this choice is made in advance during state preparation for $\mathbf{A} \oplus \mathbf{B}$, or if the context is allowed to make the choice of projecting into $\mathbf{A}$ or $\mathbf{B}$ for $\mathbf{A} \times \mathbf{B}$. As a linear resource, the choice is made to be exactly one of these, i.e. we can't choose to interact with both the $\mathbf{A}$ and $\mathbf{B}$ components of $\mathbf{A} \times \mathbf{B}$ simultaneously, but we can make the choice probabilistically since convex combinations can always be constructed in Caus$[\mathcal{C}]$. This interpretation allows us to view $\mathbf{A} \oplus \mathbf{B}$ instead as a kind of probabilistic test which sometimes gives one result and prepares an $\mathbf{A}$, and otherwise prepares a $\mathbf{B}$; dually, we could view $\mathbf{A} \times \mathbf{B}$ as a system which accepts a binary test result and conditionally prepares an $\mathbf{A}$ or a $\mathbf{B}$.

Remark 2.5.5

Probabilistic choice has been handled slightly differently in the logic literature through sub-additives [68], in which $A \oplus_p B$ describes something that yields A with probability $0 < p < 1$ and B with probability $1 - p$. These assumptions ensure that both outcomes are actually possible.

The additives in Caus$[\mathcal{C}]$ simultaneously capture the traditional interpretation of $A \oplus B$ as either *definitely* A or *definitely* B (with the outcome possibly dependent on some previous input), and sub-additives for any possible choice of probability. This suits our desire to capture black-box systems, since an external agent may not know the probability distribution generated by a particular process up front.

Thinking of first-order types as describing systems with no input (i.e. no choice in how to interact with them), both the product and coproduct have interesting interactions with first-order types because of where the classical choice happens. For coproducts, the choice is already fixed in the creation of a state, so we expect it to preserve the first-order property. However, products introduce freedom of choice in effects, allowing us to view the projections as inputs to the system dictating whether it should prepare the left or the right state.

Proposition 2.5.6

If $\mathbf{A}$ and $\mathbf{B}$ are both first-order, then so is $\mathbf{A} \oplus \mathbf{B}$.

Proof. If $c_{\mathbf{A}} = \left\{\overset{=}{\top}_A\right\}^*$ and $c_{\mathbf{B}} = \left\{\overset{=}{\top}_B\right\}^*$, then we have

$$c_{\mathbf{A}\oplus\mathbf{B}} = \left\{[\pi_A, \pi_B] \mid \pi_A \in c_{\mathbf{A}}^*, \pi_B \in c_{\mathbf{B}}^*\right\}^* = \left\{\left[\overset{=}{\top}_A, \overset{=}{\top}_B\right]\right\}^* = \left\{\overset{=}{\top}_{A\oplus B}\right\}^* \tag{2.109}$$

using APC1. □

Proposition 2.5.7

$\mathbf{A} \times \mathbf{B}$ is never first-order.

Proof. By flatness, both $p_A \fatsemi \text{¡}_{\mathbf{A}}$ and $p_B \fatsemi \text{¡}_{\mathbf{B}}$ are in $c_{\mathbf{A}\times\mathbf{B}}$. They are distinct as morphisms of $\mathcal{C}$ since they can be distinguished using $\langle \rho_A, 0_{I,B} \rangle$ for any $\rho_A \in c_{\mathbf{A}}$ (projecting on B will give zero whereas projecting on A will give id_I). □

A particular first-order object that will play a special role throughout this thesis is the binary object $\mathbf{2}$.

Definition 2.5.8

The *binary object* is defined as

$$\mathbf{2} \coloneqq \mathbf{I} \oplus \mathbf{I} \tag{2.110}$$

for which we denote the two injections as $\mathtt{t} \coloneqq \iota_1$ and $\mathtt{f} \coloneqq \iota_2$.

This is the smallest non-trivial state space (i.e. a space with multiple distinguishable states), and hence the simplest system that can be used to send information. By definition, the states of **2** are affine mixtures of the left and right injections which we can model as two-dimensional vectors of scalars summing to id_I (our interpretation of the abstract probability 1). In our standard examples for $\mathcal{C}$, the scalars for $\mathrm{Mat}[\mathbb{R}^+]$ and CP^* are $\mathbb{R}^+$ and the (complete) positivity requirement constrains each element of the vector to be positive, giving two-outcome probability distributions. For $\mathrm{Mat}[\mathbb{R}]$ we have $\mathbb{R}$ without any such positivity restriction, giving two-outcome pseudo-probabilities. This construction generalises to $\mathbf{I}^{\oplus \mathbf{n}}$ for n-outcome (pseudo-)probability distributions.

Recall that we can view effects in $\mathcal{C}$ as testing a predicate on a given state, using APC5 to obtain its negation. APC5a allows us to generalise this beyond effects to view any morphism in $\mathcal{C}$ as something that can be performed with some probability (which may depend on the context, and need not be in the range $[0,]$), i.e. one branch of a *binary test*. The following Lemma shows that we can encode the success or failure of such a test into a separate output with type **2**.

Lemma 2.5.9: Binary Test Lemma

For any $f \in \mathcal{C}(A,B)$ and $\mathbf{A} = (A, c_{\mathbf{A}}), \mathbf{B} = (B, c_{\mathbf{B}}) \in \mathrm{Ob}(\mathrm{Caus}[\mathcal{C}])$, there exists some non-zero scalar $\lambda \in \mathcal{C}(I,I)$, morphism $f' \in \mathcal{C}(A,B)$ and causal morphism $t_f \in \mathrm{Caus}[\mathcal{C}](\mathbf{A}, \mathbf{B} \parr \mathbf{2})$ such that $t_f = (\lambda \cdot f \otimes \mathtt{t}) + (f' \otimes \mathtt{f})$.

Proof. Apply APC5a to f to get $f' \in \mathcal{C}(A,B)$ and $\lambda \in \mathcal{C}(I,I)$ where $f + f' = \lambda \cdot \overset{=}{\top}_A \fatsemi \underset{=}{\bot}_B$ which is causal up to a scalar by flatness. We then pick $t_f = (\theta_{\mathbf{A}}\mu_{\mathbf{B}}\lambda^{-1} \cdot f \otimes \mathtt{t}) + (\theta_{\mathbf{A}}\mu_{\mathbf{B}}\lambda^{-1} \cdot f' \otimes \mathtt{f})$.

To check $t_f \in \mathrm{Caus}[\mathcal{C}](\mathbf{A}, \mathbf{B} \parr \mathbf{2})$, we consider an arbitrary effect $\pi \otimes \text{⫯}_{\mathbf{2}} \in c^*_{\mathbf{B} \parr \mathbf{2}}$.

$$\begin{aligned} t_f \fatsemi (\pi \otimes \text{⫯}_{\mathbf{2}}) &= \theta_{\mathbf{A}}\mu_{\mathbf{B}}\lambda^{-1} \cdot (f \otimes \mathtt{t} \fatsemi \pi \otimes \text{⫯}_{\mathbf{2}} + f' \otimes \mathtt{f} \fatsemi \pi \otimes \text{⫯}_{\mathbf{2}}) \\ &= \theta_{\mathbf{A}}\mu_{\mathbf{B}}\lambda^{-1} \cdot (f \fatsemi \pi + f' \fatsemi \pi) \\ &= \theta_{\mathbf{A}}\mu_{\mathbf{B}} \cdot \overset{=}{\top}_A \fatsemi \underset{=}{\bot}_B \fatsemi \pi \qquad (2.111) \\ &= \text{⫯}_{\mathbf{A}} \fatsemi \text{⫰}_{\mathbf{B}} \fatsemi \pi \\ &= \text{⫯}_{\mathbf{A}} \in c^*_{\mathbf{A}} \end{aligned}$$

□

In a time-oriented picture, the causality condition implies that the probability of the test outcome must be independent of whatever the environment does with the output, so one might have expected the causal type of the binary test to be $\mathbf{A} \to \mathbf{B} \otimes \mathbf{2}$. However, $\mathrm{Caus}\left[\mathcal{C}\right]$ is not time-oriented like this; $\mathbf{B}$ could be some higher-order system including an input which might affect the test outcome. In the special case of first-order systems, we have $\mathbf{B^1} \parr \mathbf{2} \cong \mathbf{B^1} \otimes \mathbf{2}$ which recovers this independence, but $\parr$ is necessary in general.

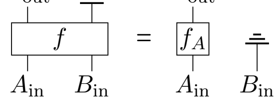
2.6 The Seq Operator

In dropping the assumption of equivalence of one-way signalling and semi-localisability from PC4, we lost out on the characterisation of $\mathbf{A^1} \multimap (\mathbf{B^1} \multimap \mathbf{C^1}) \multimap \mathbf{D^1}$ as first-order causal one-way signalling processes from FO-$\multimap$. This section will recover this result by explicitly building a monoidal structure for one-way signalling on higher-order systems and showing that it degenerates in the appropriate way for first-order systems. We initially consider several alternative constructions for the set of valid states based on one-way signalling, semi-localisability, and an asymmetric sum of products inspired by a construction for probabilistic coherence spaces [17], before showing that they all coincide up to affine closure.

2.6.1 One-Way Signalling

Let us return to the familiar causal structure between two parties, Alice and Bob, where Bob is completely in the future of Alice. Rather than each being limited to the first-order picture where they each receive a single input from their environment and yield a single output, we now consider arbitrary protocols under which they interact with the environment as described by some causal types $\mathbf{A}, \mathbf{B} \in \mathrm{Ob}\left(\mathrm{Caus}\left[\mathcal{C}\right]\right)$. The joint process performed by Alice and Bob must, primarily, act locally like $\mathbf{A}$ at Alice and $\mathbf{B}$ at Bob, so any valid state must be a state of $\mathbf{A} \parr \mathbf{B}$ in addition to being consistent with the causal structure.

One-way signalling states that the marginal process at Alice is independent of any input given to Bob by the environment.

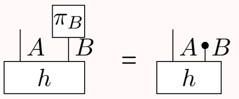


(2.112)

This generalises to independence of any choice over Bob's context.

Definition 2.6.1: One-way signalling product

$$c_{\mathbf{A}} < c_{\mathbf{B}} := \left\{ h \;\middle|\; \forall \pi_B \in c^*_{\mathbf{B}}. \right.$$



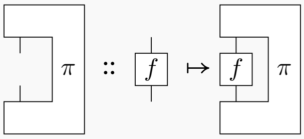
$$\left. \in c_{\mathbf{A}} \right\}$$

This is clearly a refinement of the condition in Equation 2.53 for $c_{\mathbf{A} \parr \mathbf{B}}$. To see that this is generalises one-way signalling, we must first convince ourselves that this recovers Definition 2.1.2 when the local systems describe first-order processes.

Proposition 2.6.2

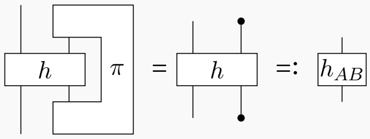
Fix any first-order objects $\mathbf{A^1}, \mathbf{B^1}, \mathbf{C^1}, \mathbf{D^1}$ and consider some Choi operator $h \in \mathcal{C}(I, A^* \otimes B \otimes C^* \otimes D)$. Then $h \in c_{\mathbf{A^1} \multimap \mathbf{B^1}} < c_{\mathbf{C^1} \multimap \mathbf{D^1}}$ iff h is the Choi operator of a first-order causal one-way signalling process $A \otimes C \to B \otimes D$.

Proof. First, consider what the effects $\pi \in c^*_{\mathbf{A}^1 \multimap \mathbf{B}^1}$ look like. As these are dual to the set of first-order processes $\mathbf{A}^1 \multimap \mathbf{B}^1$, we can visualise them this way:



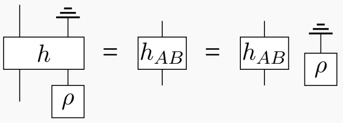
(2.113)

Consequently, the condition in the Definition 2.6.1 applied to $c_{\mathbf{A}^1 \multimap \mathbf{B}^1} < c_{\mathbf{C}^1 \multimap \mathbf{D}^1}$ says that for any causal effect $\pi \in c_{\mathbf{C}^1 \multimap \mathbf{D}^1}$ we have:



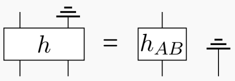
(2.114)

However, the effects in $c^*_{\mathbf{C}^1 \multimap \mathbf{D}^1}$ always take the form of plugging a causal state of $\mathbf{C^1}$ into the process and discarding the result by Corollary 2.4.28. Hence, the equation above is equivalent to saying that for all states $\rho \in c_{\mathbf{C^1}}$, there exists h_{AB} such that:



(2.115)

which, by APC3 recovers one-way non-signalling as shown in Definition 2.1.2:

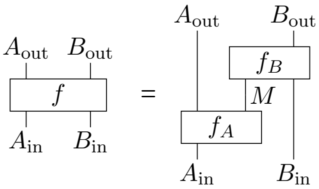


(2.116)

□

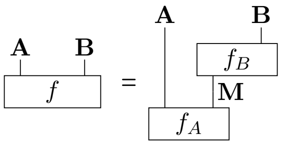
2.6.2 Semi-Localisability

Recall that semi-localisability asks the process to factorise into two local processes with an intermediate system passed from one to the other.



(2.117)

Taking this diagram and directly mapping it to higher-order systems would give the following with $f_A : \mathbf{A} \parr \mathbf{M}$ and $f_B : \mathbf{M} \to \mathbf{B}$:

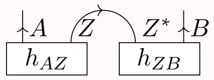


(2.118)

This is not quite what we are looking for since an arbitrary intermediate system could permit information flow in both directions. Instead, we restrict it to a first-order object, following our understanding of the causality principle as limiting a backwards information flow.

Definition 2.6.3

$c_{\mathbf{A}} \lhd c_{\mathbf{B}} := \Bigg\{$



$\Bigg| \; h_{AZ} \in c_{\mathbf{A} \parr \mathbf{Z^1}}, \; h_{ZB} \in c_{\mathbf{Z^{1*}} \parr \mathbf{B}} \Bigg\}$

Again, we recover the original definition of semi-localisability if we fix $\mathbf{A}$ and $\mathbf{B}$ to first-order processes.

Proposition 2.6.4

Fix any first-order objects $\mathbf{A^1}, \mathbf{B^1}, \mathbf{C^1}, \mathbf{D^1}$ and consider some Choi operator $h \in \mathcal{C}\left(I, A^* \otimes B \otimes C^* \otimes D\right)$. Then $h \in c_{\mathbf{A^1} \multimap \mathbf{B^1}} \lhd c_{\mathbf{C^1} \multimap \mathbf{D^1}}$ iff h is the Choi operator of a first-order causal semi-localisable process $A \otimes C \to B \otimes D$.

Proof. Using $*$-autonomy and Proposition 2.3.22, we can exhibit the following equivalences:

$$\left(\mathbf{A^1} \multimap \mathbf{B^1}\right) \parr \mathbf{Z^1} \cong \mathbf{A^1} \multimap \left(\mathbf{B^1} \parr \mathbf{Z^1}\right) \tag{2.119}$$

$$\cong \mathbf{A^1} \multimap \left(\mathbf{B^1} \otimes \mathbf{Z^1}\right)$$

$$\mathbf{Z^{1*}} \parr \left(\mathbf{C^1} \multimap \mathbf{D^1}\right) \cong \left(\mathbf{Z^1} \otimes \mathbf{C^1}\right) \multimap \mathbf{D^1} \tag{2.120}$$

So any h_{AZ} and h_{ZB} from Definition 2.6.3 can equivalently be expressed as first-order causal processes $A \to B \otimes Z$ and $Z \otimes C \to D$ respectively. From this, composing along Z immediately gives the form of a semi-localisable process. □

2.6.3 Asymmetric Sum of Products

The final definition we will consider is inspired by an asymmetric monoidal product on *probabilistic coherence spaces*.

Definition 2.6.5: Probabilistic coherence space [56, Theorem 2]

A *probabilistic coherence space* is a pair (X, A) of a finite set X and a subset A of the functions $X \to \mathbb{R}^+$ onto the non-negative reals such that:

- A is non-empty;
- A is closed under convex combinations;
- A is downward closed under the pointwise order on $\mathbb{R}^+$.

Definition 2.6.6: [17, Definition 6.9]

The *seq product* of probabilistic coherence spaces (X, A) and (Y, B) is given by $(X \times Y, A \obslash B)$ where

$$A \obslash B := \left\{ \sum_{i \in \mathcal{I}} f_i \otimes g_i \;\middle|\; \sum_{i \in \mathcal{I}} f_i \in A,\ \forall i.\ g_i \in B \right\}$$

To adapt this for $\mathrm{Caus}[\mathcal{C}]$, we keep the form of a sum of products, requiring that each term on the right is individually causal, but only the sum of the terms on the left need be causal. We also suppose the terms on the left are taken from $\mathrm{Sub}(\mathcal{C})$ to make use of the affine-linear structure so long as their sum exists in $\mathcal{C}$ itself.

Definition 2.6.7

$$c_{\mathbf{A}} \obslash c_{\mathbf{B}} := \left\{ h \;\middle|\; \begin{array}{c} \exists \mathcal{I}, \{f_i\}_{i \in \mathcal{I}} \subseteq \mathrm{Sub}(\mathcal{C})(I, A), \{g_i\}_{i \in \mathcal{I}} \subseteq c_{\mathbf{B}}, f \in c_{\mathbf{A}}. \\ [h] \sim \sum_{i \in \mathcal{I}} f_i \otimes [g_i], \\ [f] \sim \sum_{i \in \mathcal{I}} f_i \end{array} \right\}$$

This is easy to convert to a form based on preferred bases, which clearly sits somewhere on a spectrum between $c_{\mathbf{A} \otimes \mathbf{B}}$ and $c_{\mathbf{A} \parr \mathbf{B}}$.

Lemma 2.6.8

Let $\{\rho_i^A\}_{i\in\mathfrak{B}_A^{\mathcal{C}}}$ and $\{\rho_j^B\}_{j\in\mathfrak{B}_B^{\mathcal{C}}}$ be preferred bases for objects $\mathbf{A},\mathbf{B}\in\mathrm{Ob}\left(\mathrm{Caus}\left[\mathcal{C}\right]\right)$. Then

$$c_{\mathbf{A}} \obslash c_{\mathbf{B}} = \left\{ h \;\middle|\; \begin{array}{c} [h] \sim \sum_{i,j} s_{i,j}\cdot\left[\rho_i^A\otimes\rho_j^B\right], \\ \sum_{i,j} s_{i,j}\sim \mathrm{id}_I, \\ \forall i,j.\ j\in\overline{\mathfrak{B}_{\mathbf{B}}} \implies s_{i,j}\sim 0 \\ \forall i\in\overline{\mathfrak{B}_{\mathbf{A}}}.\ \sum_j s_{i,j}\sim 0 \end{array} \right\}$$

Proof. $\supseteq$ is straightforward by picking the indices $\mathcal{I}=\mathfrak{B}_{\mathbf{B}}$ and states $\{g_j\}_{j\in\mathfrak{B}_{\mathbf{B}}} = \{\rho_j^B\}_{j\in\mathfrak{B}_{\mathbf{B}}}$ and $\{f_j\}_{j\in\mathfrak{B}_{\mathbf{B}}} = \{\sum_i s_{i,j}\cdot[\rho_i^A]\}_{j\in\mathfrak{B}_{\mathbf{B}}}$. The conditions guarantee that $\sum_j f_j$ is an affine combination of causal basis elements and hence is causal.

For $\subseteq$, we expand each f_i and g_i in terms of the basis states and group matching terms.

$$f_k \sim \sum_i s_{k,i}\cdot\left[\rho_i^A\right] \tag{2.121}$$

$$[g_k] \sim \sum_j s'_{k,j}\cdot\left[\rho_j^B\right] \tag{2.122}$$

$$\begin{aligned}[h] &\sim \sum_k f_k\otimes[g_k] \qquad\qquad (2.123)\\ &\sim \sum_{i,j}\left(\sum_k s_{k,i}s'_{k,j}\right)\cdot\left[\rho_i^A\otimes\rho_j^B\right]\end{aligned}$$

As always, the combination is affine by flatness.

$$\begin{aligned}\sum_{i,j}\left(\sum_k s_{k,i}s'_{k,j}\right) &\sim \sum_{i,j}\left(\sum_k s_{k,i}s'_{k,j}\right)\cdot\left[\left(\rho_i^A\otimes\rho_j^B\right)\fatsemi\left(⫯_{\mathbf{A}}\otimes⫯_{\mathbf{B}}\right)\right]\\ &\sim \left[h\fatsemi\left(⫯_{\mathbf{A}}\otimes⫯_{\mathbf{B}}\right)\right]\\ &\sim \sum_k\left(f_k\fatsemi[⫯_{\mathbf{A}}]\right)\otimes\left[g_k\fatsemi⫯_{\mathbf{B}}\right] \qquad\qquad (2.124)\\ &\sim \sum_k f_k\fatsemi[⫯_{\mathbf{A}}]\\ &\sim \left[f\fatsemi[⫯_{\mathbf{A}}]\right]\sim\mathrm{id}_I\end{aligned}$$

Each g_k is causal and hence an affine combination of only causal basis elements - that is, for any $j\in\overline{\mathfrak{B}_{\mathbf{B}}}$, $s'_{k,j}\sim 0$.

$$\sum_k s_{k,i}s'_{k,j}\sim\sum_k s_{k,i}0\sim 0 \tag{2.125}$$

Similarly, f is causal - for any $i\in\overline{\mathfrak{B}_{\mathbf{A}}}$,

$$\sum_j\left(\sum_k s_{k,i}s'_{k,j}\right)\sim\sum_{i',j}\left(\sum_k s_{k,i'}s'_{k,j}\right)\cdot\left(\left[\rho_{i'}^A\right]\fatsemi e_i\right)\otimes\left[\rho_j^B\fatsemi⫯_{\mathbf{B}}\right]$$

$$
\begin{aligned}
&\sim [h] ⨟ (e_i \otimes [⫯_{\mathbf{B}}]) \\
&\sim \sum_k (f_k ⨟ e_i) \otimes [g_k ⨟ ⫯_{\mathbf{B}}] \qquad (2.126) \\
&\sim \sum_k f_k ⨟ e_i \\
&\sim [f] ⨟ e_i \sim 0
\end{aligned}
$$

□

2.6.4 Equivalence of Definitions

Knowing that one-way signalling and semi-localisability are equivalent for first-order processes in both quantum theory and classical probability theory, we anticipate there being some general relation between $c_{\mathbf{A}} < c_{\mathbf{B}}$ and $c_{\mathbf{A}} \lhd c_{\mathbf{B}}$. By following a sequence of inclusions, we find that $c_{\mathbf{A}} < c_{\mathbf{B}}$ coincides with $c_{\mathbf{A}} \obslash c_{\mathbf{B}}$, preserves closure, and is self-dual. It is unknown whether $c_{\mathbf{A}} \lhd c_{\mathbf{B}}$ preserves closure in general, but manually taking its affine closure $(c_{\mathbf{A}} \lhd c_{\mathbf{B}})^{**}$ coincides with $c_{\mathbf{A}} < c_{\mathbf{B}}$.

We will start with stating the equivalence and the chain of inclusions used to derive it to give context for the particular inclusions we will provide here.

Theorem 2.6.9: Seq Equivalence Theorem

$$
c_{\mathbf{A}} < c_{\mathbf{B}} = (c_{\mathbf{A}}^* < c_{\mathbf{B}}^*)^* = (c_{\mathbf{A}} \lhd c_{\mathbf{B}})^{**} = c_{\mathbf{A}} \obslash c_{\mathbf{B}}
$$

Proof. The following inclusions are annotated with the corresponding Lemma references. Subscript $*$'s apply the Lemma to $c_{\mathbf{A}}^*$ and $c_{\mathbf{B}}^*$, and superscript $*$'s dualise the Lemma's inclusion via Equation 2.48.

$$
\begin{tikzcd}
c_{\mathbf{A}} < c_{\mathbf{B}} \arrow[r, "2.6.10"] & c_{\mathbf{A}} \obslash c_{\mathbf{B}} \arrow[r, "\text{closure}"] & (c_{\mathbf{A}} \obslash c_{\mathbf{B}})^{**} \arrow[r, "2.6.11^*_*"] & (c_{\mathbf{A}}^* < c_{\mathbf{B}}^*)^* \arrow[d, "2.6.12^*"] \\
& (c_{\mathbf{A}}^* \lhd c_{\mathbf{B}}^*)^* \arrow[ul, "2.6.12_*"] & (c_{\mathbf{A}} < c_{\mathbf{B}})^{**} \arrow[l, "2.6.14^{**}"'] & (c_{\mathbf{A}} \lhd c_{\mathbf{B}})^{**} \arrow[l, "2.6.13^{**}"']
\end{tikzcd}
\qquad (2.127)
$$

□

Lemma 2.6.10

$$
c_{\mathbf{A}} < c_{\mathbf{B}} \subseteq c_{\mathbf{A}} \obslash c_{\mathbf{B}}
$$

Proof. Consider an arbitrary $h \in c_{\mathbf{A}} < c_{\mathbf{B}}$ and expand it in terms of a preferred basis as $[h] \sim \sum_{i,j} s_{i,j} \cdot \left[\rho_i^A \otimes \rho_j^B\right]$, aiming for the form of Lemma 2.6.8. The constant marginal is causal $h ⨟ (\mathrm{id}_A \otimes ⫯_{\mathbf{B}}) \in c_{\mathbf{A}}$, giving us the affine sum.

$$\begin{aligned}\sum_{i,j} s_{i,j} &\sim \sum_{i,j} s_{i,j} \cdot \left[\left(\rho^A_i \otimes \rho^B_j\right) \fatsemi (\text{⫯}_{\mathbf{A}} \otimes \text{⫯}_{\mathbf{B}})\right] \\ &\sim \left[h \fatsemi (\mathrm{id}_A \otimes \text{⫯}_{\mathbf{B}}) \fatsemi \text{⫯}_{\mathbf{A}}\right] \\ &\sim \mathrm{id}_I\end{aligned} \tag{2.128}$$

For each $j \in \overline{\mathfrak{B}}_{\mathbf{B}}$, use PB$\mathcal{C}$ to obtain some $\pi_B \in c^*_{\mathbf{B}}$ such that $[\pi_B] \sim [\text{⫯}_{\mathbf{B}}] + [\alpha] \cdot e^B_j$ with some invertible scalar $\alpha \in \mathcal{C}(I, I)$. For any i, uniqueness of the marginal gives the next condition.

$$\begin{aligned}s_{i,j} &\sim \sum_{i',j'} s_{i',j'} \cdot \left[\rho^A_{i'} \otimes \rho^B_{j'}\right] \fatsemi \left(e^A_i \otimes e^B_j\right) \\ &\sim \left[\alpha^{-1}\right] \cdot [h] \fatsemi \left([\alpha] \cdot e^A_i \otimes e^B_j\right) \\ &\sim \left[\alpha^{-1}\right] \cdot \left([h] \fatsemi \mathrm{id}_A \otimes [\pi_B] - [h] \fatsemi \mathrm{id}_A \otimes \text{⫯}_{\mathbf{B}}\right) \fatsemi e^A_i \\ &\sim \left[\alpha^{-1}\right] \cdot 0 \fatsemi e^A_i \\ &\sim 0\end{aligned} \tag{2.129}$$

For each $i \in \overline{\mathfrak{B}}_{\mathbf{A}}$, we similarly pick $\pi_A \in c^*_{\mathbf{A}}$ such that $[\pi_A] \sim [\text{⫯}_{\mathbf{B}}] + [\beta] \cdot e^A_i$, giving us the remaining condition from causality of the marginal.

$$\begin{aligned}\sum_j s_{i,j} &\sim \sum_{i',j} s_{i',j} \cdot \left[\rho^A_{i'} \otimes \rho^B_j\right] \fatsemi \left(e^A_i \otimes [\text{⫯}_{\mathbf{B}}]\right) \\ &\sim [h] \fatsemi \left(e^A_i \otimes [\text{⫯}_{\mathbf{B}}]\right) \\ &\sim \left[h \fatsemi (\mathrm{id}_A \otimes \text{⫯}_{\mathbf{B}})\right] \fatsemi \left([\pi_A] - [\text{⫯}_{\mathbf{A}}]\right) \\ &\sim [\mathrm{id}_I] - [\mathrm{id}_I] \\ &\sim 0\end{aligned} \tag{2.130}$$

□

Lemma 2.6.11

$$c_{\mathbf{A}} < c_{\mathbf{B}} \subseteq \left(c^*_{\mathbf{A}} \oslash c^*_{\mathbf{B}}\right)^*$$

Proof. Take any $h \in c_{\mathbf{A}} < c_{\mathbf{B}}$ and any $k \in c^*_{\mathbf{A}} \oslash c^*_{\mathbf{B}}$, decomposing it as $[k] \sim \sum_i f_i \otimes [g_i]$ with $\{g_i\}_i \subseteq c^*_{\mathbf{B}}$ and $f \in c^*_{\mathbf{A}}$ such that $[f] \sim \sum_i f_i$.

$$\begin{aligned}[h \fatsemi k^*] &\sim [h] \fatsemi \left(\sum_i f^*_i \otimes [g^*_i]\right) \\ &\sim \sum_i \left[h \fatsemi (\mathrm{id}_A \otimes g^*_i)\right] \fatsemi f^*_i \\ &\sim \sum_i \left[h \fatsemi (\mathrm{id}_A \otimes \text{⫯}_{\mathbf{B}})\right] \fatsemi f^*_i\end{aligned} \tag{2.131}$$

$$
\begin{aligned}
&\sim [h \mathbin{⨟} (\mathrm{id}_A \otimes ⫯_{\mathbf{B}}) \mathbin{⨟} f^*] \\
&\sim [\mathrm{id}_I]
\end{aligned}
$$

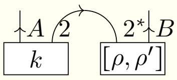
Faithfulness of the embedding pulls this back to $h \mathbin{⨟} k^* = \mathrm{id}_I$. □

Lemma 2.6.12

$$(c_{\mathbf{A}} \lhd c_{\mathbf{B}})^* \subseteq c^*_{\mathbf{A}} < c^*_{\mathbf{B}}$$

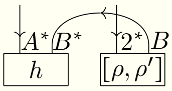
Proof. Consider any $h \in (c_{\mathbf{A}} \lhd c_{\mathbf{B}})^*$.

For any two $\rho, \rho' \in c_{\mathbf{B}}$, their copairing is $[\rho, \rho'] \in c_{\mathbf{2} \multimap \mathbf{B}} = c_{\mathbf{2}^* \parr \mathbf{B}}$ up to partial transpose. For any $k \in c_{\mathbf{A} \parr \mathbf{2}}$:



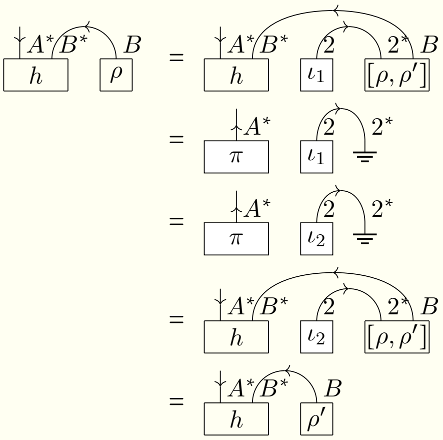
$$\in c_{\mathbf{A} \lhd \mathbf{B}} \tag{2.132}$$

Taking the inner product of this morphism with h will give id_I. Since this holds for any $k \in c_{\mathbf{A} \parr \mathbf{2}}$:



$$\in c^*_{\mathbf{A} \parr \mathbf{2}} = c_{\mathbf{A}^* \otimes \mathbf{2}^*} \tag{2.133}$$

2 is first-order, so $c^*_{\mathbf{2}} = \left\{ \overset{=}{\top}_2 \right\}$. By Corollary 2.4.28, the above morphism must separate into $\pi \otimes \underset{=}{\perp}_2$ for some $\pi \in c^*_{\mathbf{A}}$.



(2.134)

Selecting $\rho' = ⫰_{\mathbf{B}}$ and keeping ρ arbitrary, we obtain uniqueness of the marginal. The marginal is also causal (in $c^*_{\mathbf{A}}$) since, for any $v \in c_{\mathbf{A}}$, the product $v \otimes ⫰_{\mathbf{B}} \in c_{\mathbf{A}} \lhd c_{\mathbf{B}}$ is trivially semi-localisable ($\mathbf{I}$ is first-order), and so $h \mathbin{⨟} (v^* \otimes ⫯_{\mathbf{B}^*}) = \mathrm{id}_I$. With this, we conclude that $h \in c^*_{\mathbf{A}} < c^*_{\mathbf{B}}$. □

Lemma 2.6.13

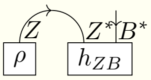
$$c_{\mathbf{A}} \lhd c_{\mathbf{B}} \subseteq c_{\mathbf{A}} < c_{\mathbf{B}}$$

Proof. Let $h \in c_{\mathbf{A}} \lhd c_{\mathbf{B}}$ decompose into $h_{AZ} \in c_{\mathbf{A} \parr \mathbf{Z^1}}$ and $h_{ZB} \in c_{\mathbf{Z^{1*}} \parr \mathbf{B}}$. Applying any effect $\pi \in c^*_{\mathbf{B}}$ locally to h_{ZB} must result in a causal state of $\mathbf{Z^{1*}}$. Since $\mathbf{Z^1}$ is first-order, this must be $\bullet_{\mathbf{Z^{1*}}}$ regardless of the choice of π. Thus, $h \fatsemi (\mathrm{id}_A \otimes \pi) = h_{AZ} \fatsemi (\mathrm{id}_A \otimes \bullet_{\mathbf{Z^1}})$ gives the unique witness in $c_{\mathbf{A}}$ (from $h_{AZ} \in c_{\mathbf{A} \parr \mathbf{Z^1}}$), so $h \in c_{\mathbf{A}} < c_{\mathbf{B}}$. □

Lemma 2.6.14

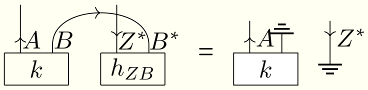
$$c_{\mathbf{A}} < c_{\mathbf{B}} \subseteq (c^*_{\mathbf{A}} \lhd c^*_{\mathbf{B}})^*$$

Proof. Consider any $k \in c_{\mathbf{A}} < c_{\mathbf{B}}$ and $h \in c^*_{\mathbf{A}} \lhd c^*_{\mathbf{B}}$ which decomposes into $h_{AZ} \in c_{\mathbf{A^*} \parr \mathbf{Z^1}}$ and $h_{ZB} \in c_{\mathbf{Z^{1*}} \parr \mathbf{B^*}}$. For any $\rho \in c_{\mathbf{Z^1}}$ (in particular, the elements of the canonical basis from APC3), the following state is in $c^*_{\mathbf{B}}$:



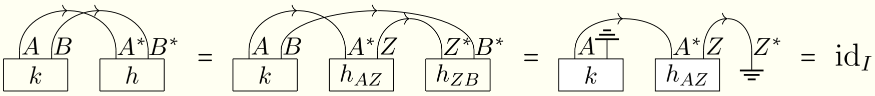
(2.135)

Composing this with k must yield the unique marginal regardless of the choice of ρ. By equality on the basis states:



(2.136)

This product state is causal for $\mathbf{A} \otimes \mathbf{Z^{1*}} \cong (\mathbf{A}^* \parr \mathbf{Z^1})^*$ and therefore is causal in the context h_{AZ}. In summary:



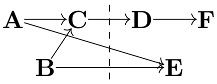
(2.137)

□

This was a minimal number of inclusions required to obtain all equalities in the Seq Equivalence Theorem, though there will exist alternative direct proofs for some inclusions obtained by transitivity. It may be insightful for the reader to think about how other inclusions between these operators may be proved directly - some straightforward examples include $c_{\mathbf{A}} \oslash c_{\mathbf{B}} \subseteq c_{\mathbf{A}} < c_{\mathbf{B}}$ and $c_{\mathbf{A}} \oslash c_{\mathbf{B}} \subseteq (c^*_{\mathbf{A}} \oslash c^*_{\mathbf{B}})^*$. The converse of the latter is a little more involved, but can be done using Lemma 2.6.8. Likewise, all proofs in the remainder of this thesis will freely choose whichever form makes for the simplest proof, though additional insight may be found from thinking about alternative proofs using the others.

Example 2.6.15

In this example, we will highlight the inclusion $c_{\mathbf{A}<\mathbf{B}} \subseteq c^{**}_{\mathbf{A}\lhd\mathbf{B}}$. Suppose we begin with a first-order causal process f that is signal-consistent with respect to the following causal structure (each local system is first-order causal, e.g. $\mathbf{A} = \mathbf{A^1}_{\text{in}} \multimap \mathbf{A^1}_{\text{out}}$):

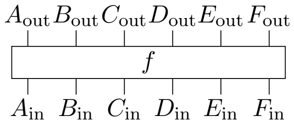


(2.138)

Using the above cut, such a process will satisfy the one-way signalling constraint with $\{\mathbf{A}, \mathbf{B}, \mathbf{C}\}$ before $\{\mathbf{D}, \mathbf{E}, \mathbf{F}\}$. Making similar observations on each half, this shows it is of the following type:

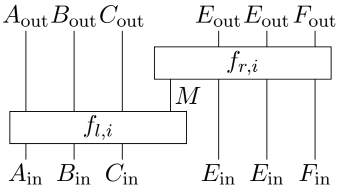
$$((\mathbf{A} \otimes \mathbf{B}) < \mathbf{C}) < ((\mathbf{D} < \mathbf{F}) \otimes \mathbf{E}) \tag{2.139}$$

By Seq Equivalence Theorem, we can express f as an affine-combination of semi-localisable processes, each of which factorise into local processes for the left and right half.



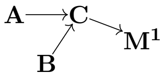
A_{out} B_{out} C_{out} D_{out} E_{out} F_{out}
f
A_{in} B_{in} C_{in} D_{in} E_{in} F_{in}

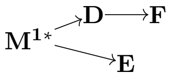
$$\sim \sum_i \alpha_i \tag{2.140}$$



A_{out} B_{out} C_{out}
E_{out} E_{out} F_{out}
$f_{r,i}$
M
$f_{l,i}$
A_{in} B_{in} C_{in}
E_{in} E_{in} F_{in}

The power in the equivalence comes from the higher-order definition of $\lhd$, where we can find particular first-order causal processes $f_{l,i}$ consistent with the same local causal structure within $\{\mathbf{A}, \mathbf{B}, \mathbf{C}\}$ in the original causal structure, and similarly the $f_{r,i}$ can preserve the causal structure within $\{\mathbf{D}, \mathbf{E}, \mathbf{F}\}$.





(2.141)

$$f_{l,i} : ((\mathbf{A} \otimes \mathbf{B}) < \mathbf{C}) \parr \mathbf{M^1} \qquad f_{r,i} : \mathbf{M^{1*}} \parr ((\mathbf{D} < \mathbf{F}) \otimes \mathbf{E}) \tag{2.142}$$

Remark 2.6.16

During the development of the Seq Equivalence Theorem, I initially began with trying to prove it using the original precausal category assumptions (Definition 2.3.1). Lemmas 2.6.13 and 2.6.14 were straightforward, and so were Lemmas 2.6.11 and 2.6.12 with the minimal added assumption of biproducts.

Proving Lemma 2.6.10, however, seemed to require heavier assumptions on the structure of the base category $\mathcal{C}$. After proving the result in the specific case of $\mathcal{C} = \mathrm{CP}^*$, I picked out the relevant abstract properties used by the proof which gave rise to conditions APC3-APC5 and the definition of an additive pre-causal category. Any future attempt to weaken the assumptions may struggle to preserve this property, and therefore the equivalence of one-way signalling and semi-localisability. Even without this Lemma, though, we still have a de Morgan duality $c_{\mathbf{A}<\mathbf{B}} = c^*_{\mathbf{A}^* \lhd \mathbf{B}^*}$, which looks similar to the operators of Slavnov's *semi-commutative multiplicative linear logic* [108].

2.6.5 Seq is a Monoidal Structure

One output of the Seq Equivalence Theorem is that $c_{\mathbf{A}} < c_{\mathbf{B}}$ is closed, and flatness is straightforward. We can therefore elevate it to a genuine object in $\mathrm{Caus}[\mathcal{C}]$ or, more specifically, a new monoidal structure.

Definition 2.6.17: Seq

The *seq product* in $\mathrm{Caus}[\mathcal{C}]$ is the monoidal structure $(<, \mathbf{I})$ from the bifunctor $< : \mathrm{Caus}[\mathcal{C}] \times \mathrm{Caus}[\mathcal{C}] \to \mathrm{Caus}[\mathcal{C}]$ defined on objects as

$$\mathbf{A} < \mathbf{B} \coloneqq (A \otimes B, c_{\mathbf{A}} < c_{\mathbf{B}}) \tag{2.143}$$

and on morphisms identically to $\otimes : \mathcal{C} \times \mathcal{C} \to \mathcal{C}$.

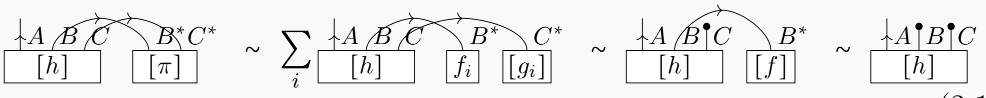
Proposition 2.6.18

The seq product is a monoidal structure.

Proof. Bifunctoriality of $⅋$ means product morphisms like $f < g$ will preserve semi-localisability of states, giving causality of $f < g : \mathbf{A} < \mathbf{B} \to \mathbf{C} < \mathbf{D}$.

The associator α and unitors λ and ρ are inherited from $\mathcal{C}$, which gives naturality and the coherence identities for free. We just need to show that each of these and their inverses are causal when using the seq product.

For $\alpha_{A,B,C} : (\mathbf{A} < \mathbf{B}) < \mathbf{C} \to \mathbf{A} < (\mathbf{B} < \mathbf{C})$, consider any one-way signalling $h \in c_{(\mathbf{A}<\mathbf{B})<\mathbf{C}} = (c_{\mathbf{A}} < c_{\mathbf{B}}) < c_{\mathbf{C}}$ and effect $\pi : c^*_{\mathbf{B}<\mathbf{C}} = c^*_{\mathbf{B}} \oslash c^*_{\mathbf{C}}$ with an asymmetric sum of product form $[\pi] \sim \sum_i f_i \otimes [g_i]$ with $[f] \sim \sum_i f_i$. We need to show that composing h with π (via the associator) yields a constant marginal in $c_{\mathbf{A}}$ regardless of the choice of π.



(2.144)

For $\lambda_A : \mathbf{I} < \mathbf{A} \to \mathbf{A}$, consider any $h \in c_{\mathbf{I}<\mathbf{A}}$ and effect $\pi \in c^*_{\mathbf{A}}$. The marginal of h must be id_I because it is the only causal state of $\mathbf{I}$, hence adjoining h and π (via λ_A) will yield id_I for normalisation.

For $\rho_A : \mathbf{A} < \mathbf{I} \to \mathbf{A}$, consider any $h \in c_{\mathbf{A}<\mathbf{I}}$. The unique effect from $\mathbf{I}$ is id_I, so by causality of the marginal we have $h \fatsemi \rho_A = h \fatsemi (\mathrm{id}_A \otimes \top_I) \fatsemi \rho_A \in c_{\mathbf{A}}$.

The inverses α^{-1}, λ^{-1}, and ρ^{-1} are all causal by dual arguments. □

Unlike $\otimes$ and $\parr$, this monoidal structure is not braided/symmetric, owing to the asymmetry in the definitions and, more broadly, in the directed nature of time in causal structures. We will use $\mathbf{A} > \mathbf{B}$ to refer to the symmetrically defined monoidal structure, describing "$\mathbf{A}$ after $\mathbf{B}$".

We can think of a process of type $\mathbf{A} < \mathbf{B}$ as something that presents an $\mathbf{A}$ interface at some time and then presents a $\mathbf{B}$ interface at some later time, possibly requiring the $\mathbf{A}$ protocol to be completed beforehand. The self-duality from the Seq Equivalence Theorem states that the valid ways to interact with such a process are therefore to first completely consume the $\mathbf{A}$, before later consuming the $\mathbf{B}$.

$$(\mathbf{A} < \mathbf{B})^* = \mathbf{A}^* < \mathbf{B}^* \tag{2.145}$$

We see $\mathbf{A} < \mathbf{B}$ as sitting somewhere in between $\mathbf{A} \otimes \mathbf{B}$ and $\mathbf{A} \parr \mathbf{B}$; the product states that span $c_{\mathbf{A}\otimes\mathbf{B}}$ are trivially semi-localisable, and Equation 2.53 defines $c_{\mathbf{A}\parr\mathbf{B}}$ as identical to $c_{\mathbf{A}} < c_{\mathbf{B}}$ without requiring the uniqueness of the marginal. These set inclusions turn identities $\mathrm{id}_{A\otimes B}$ into natural transformations, refining the embedding of $\mathbf{A} \otimes \mathbf{B}$ into $\mathbf{A} \parr \mathbf{B}$ from Equation 2.58.

$$\mathbf{A} \otimes \mathbf{B} \Rightarrow \mathbf{A} < \mathbf{B} \Rightarrow \mathbf{A} \parr \mathbf{B} \tag{2.146}$$

Additionally, $<$ combines with either $\otimes$ or $\parr$ to give a *duoidal structure* on $\mathrm{Caus}[\mathcal{C}]$.

Definition 2.6.19: Duoidal Structure [4, Definition 6.1]

A *duoidal structure* is a pair of monoidal structures $(\mathcal{C}, \diamond, I)$, $(\mathcal{C}, \star, J)$ on the same category, along with a natural transformation

$$(A \star B) \diamond (C \star D) \to (A \diamond C) \star (B \diamond D)$$

called the *interchange*, and morphisms

$$I \to I \star I \qquad\qquad J \diamond J \to J \qquad\qquad I \to J$$

satisfying some coherence conditions.

Remark 2.6.20

Structures similar to duoidal categories are common amongst the logic literature. In particular, rules corresponding to the interchange in deep inference formalisms are often called *medial* rules [20, 110], and can be seen between a wide variety of multiplicative and additive connectives. One translation of such a rule into the language of category theory as a *BV-category* [17] gives something very similar to the definition of a duoidal structure but with a different set of coherence conditions.

Proposition 2.6.21

$(\mathrm{Caus}[\mathcal{C}], \otimes, \mathbf{I})$ and $(\mathrm{Caus}[\mathcal{C}], <, \mathbf{I})$ form a duoidal structure. The unit morphisms are given by unitors and id_I, and the interchange is the natural transformation:

$$\mathrm{id}_A \otimes \sigma_{B,C} \otimes \mathrm{id}_D : (\mathbf{A} < \mathbf{B}) \otimes (\mathbf{C} < \mathbf{D}) \Rightarrow (\mathbf{A} \otimes \mathbf{C}) < (\mathbf{B} \otimes \mathbf{D})$$

Another duoidal structure between $(\mathrm{Caus}[\mathcal{C}], <, \mathbf{I})$ and $(\mathrm{Caus}[\mathcal{C}], \parr, \mathbf{I})$ is obtained by duality, with interchange:

$$\mathrm{id}_A \otimes \sigma_{B,C} \otimes \mathrm{id}_D : (\mathbf{A} \parr \mathbf{B}) < (\mathbf{C} \parr \mathbf{D}) \Rightarrow (\mathbf{A} < \mathbf{C}) \parr (\mathbf{B} < \mathbf{D})$$

Proof. Naturality and the coherence conditions can be inherited from naturality and coherence in $\mathcal{C}$, so it suffices to show that the interchange is causal.

Let $h \in c_{\mathbf{A}} \obslash c_{\mathbf{B}}$ and $k \in c_{\mathbf{C}} \obslash c_{\mathbf{D}}$ be states with decompositions

$$[h] \sim \sum_i a_i \otimes [b_i] \qquad \forall i.\ b_i \in c_{\mathbf{B}} \tag{2.147}$$

$$\exists a \in c_{\mathbf{A}}.\ [a] \sim \sum_i a_i \tag{2.148}$$

$$[k] \sim \sum_j c_j \otimes [d_j] \qquad \forall j.\ d_j \in c_{\mathbf{D}} \tag{2.149}$$

$$\exists c \in c_{\mathbf{C}}.\ [c] \sim \sum_j c_j \tag{2.150}$$

Then

$$\begin{aligned} [(h \otimes k) \mathbin{⨟} (\mathrm{id}_A \otimes \sigma_{B,C} \otimes \mathrm{id}_C)] \sim \sum_{i,j} a_i \otimes c_j \otimes [b_i \otimes d_j] &\in c_{\mathbf{A} \otimes \mathbf{C}} \obslash c_{\mathbf{B} \otimes \mathbf{D}} \\ \sum_{i,j} a_i \otimes c_j \sim [a \otimes c] &\in c_{\mathbf{A} \otimes \mathbf{C}} \qquad (2.151) \\ \forall i, j.\ b_i \otimes d_j &\in c_{\mathbf{B} \otimes \mathbf{D}} \end{aligned}$$

□

Duoidal categories are commonly used to describe settings that allow for both an independent composition (e.g. ⊗) and a dependent composition (e.g. <) [17, 104, 46], fitting our interpretation of space-like and time-like separations in causal structures.

From the interchanges, fixing either object in the middle to **I** gives natural transformations for *linear distribution* laws like Equation 2.59.

$$\mathbf{A} \otimes (\mathbf{B} < \mathbf{C}) \Rightarrow (\mathbf{A} \otimes \mathbf{B}) < \mathbf{C} \qquad (2.152)$$

$$(\mathbf{A} < \mathbf{B}) \otimes \mathbf{C} \Rightarrow \mathbf{A} < (\mathbf{B} \otimes \mathbf{C}) \qquad (2.153)$$

$$\mathbf{A} < (\mathbf{B} \parr \mathbf{C}) \Rightarrow (\mathbf{A} < \mathbf{B}) \parr \mathbf{C} \qquad (2.154)$$

$$(\mathbf{A} \parr \mathbf{B}) < \mathbf{C} \Rightarrow \mathbf{A} \parr (\mathbf{B} < \mathbf{C}) \qquad (2.155)$$

Products and coproducts also form duoidal structures with any other monoidal structure [104], giving yet more interchanges. These express the ideas that multiple sources of external choice can always be amalgamated into a single global external choice, and dually any internal choice about the global system (where each option has the same relationship between the local systems) can be split into independent local choices.

$$\forall \square \in \{\otimes, \parr, <, >, \times, \oplus\}.\ (\mathbf{A} \times \mathbf{B}) \square (\mathbf{C} \times \mathbf{D}) \Rightarrow (\mathbf{A} \square \mathbf{C}) \times (\mathbf{B} \square \mathbf{D}) \qquad (2.156)$$

$$\forall \square \in \{\otimes, \parr, <, >, \times, \oplus\}.\ (\mathbf{A} \square \mathbf{B}) \oplus (\mathbf{C} \square \mathbf{D}) \Rightarrow (\mathbf{A} \oplus \mathbf{C}) \square (\mathbf{B} \oplus \mathbf{D}) \qquad (2.157)$$

2.6.6 Causality as a Type Equality

Recall Proposition 2.3.22, which demonstrated that there exist some equations between the operators of Caus $[\mathcal{C}]$ which hold on first-order objects but not in general. It is natural to ask whether more equations similarly hold with seq when applied to first-order objects. We can go one better and give an equation which holds precisely when the subsystems are first-order, completely characterising them.

Theorem 2.6.22: First-Order Theorem

$\mathbf{A}^* \parr \mathbf{A} = \mathbf{A}^* < \mathbf{A} \iff |c^*_{\mathbf{A}}| = 1$. Consequently, the following equations hold for any first-order objects $\mathbf{A^1}, \mathbf{A'^1}$ and any object $\mathbf{B}$:

$$\mathbf{A^1} \parr \mathbf{B} = \mathbf{A^1} > \mathbf{B} \tag{2.158}$$

$$\mathbf{A^1} \otimes \mathbf{B} = \mathbf{A^1} < \mathbf{B} \tag{2.159}$$

$$\mathbf{A^1} \parr \mathbf{A'^1} = \mathbf{A^1} < \mathbf{A'^1} = \mathbf{A^1} > \mathbf{A'^1} = \mathbf{A^1} \otimes \mathbf{A'^1} \tag{2.160}$$

$$\mathbf{A^{1*}} \parr \mathbf{B} = \mathbf{A^{1*}} < \mathbf{B} \tag{2.161}$$

$$\mathbf{A^{1*}} \otimes \mathbf{B} = \mathbf{A^{1*}} > \mathbf{B} \tag{2.162}$$

$$\mathbf{A^{1*}} \parr \mathbf{A'^{1*}} = \mathbf{A^{1*}} < \mathbf{A'^{1*}} = \mathbf{A^{1*}} > \mathbf{A'^{1*}} = \mathbf{A^{1*}} \otimes \mathbf{A'^{1*}} \tag{2.163}$$

Proof. $c_{\mathbf{A}^* < \mathbf{A}} \subseteq c_{\mathbf{A}^* \parr \mathbf{A}}$ always holds, so it is sufficient to just look at the converse.

$\Longrightarrow$: The cup $\eta_A : I \to A^* \otimes A$ is always causal for $\mathbf{A}^* \parr \mathbf{A}$ up to some invertible scalar. Composing this with any effect $\pi \in c^*_{\mathbf{A}}$ reduces to its transpose (up to scalar). If it is one-way signalling, then $\pi^* = \text{⫯}^*_{\mathbf{A}}$ means we have a unique effect for $\mathbf{A}$.

$\Longleftarrow$: If $\mathbf{A}$ has a unique effect, then any $h : \mathbf{A}^* \parr \mathbf{A}$ trivially has a unique marginal and hence is in $c_{\mathbf{A}^*} < c_{\mathbf{A}}$.

As for the additional equations: 2.160 holds by the same argument as Proposition 2.3.22; for 2.158, the unique effect implies a unique marginal by construction; and for 2.161, the unique state must be the unique marginal. The rest hold by duality. □

It is interesting to compare this result to the causality principle: that a theory has no signalling from the future iff every system has a unique effect, i.e. the theory describes first-order systems. We can more directly encode the proof of the causality principle by considering probabilistic tests with input $\mathbf{A}$ and output $\mathbf{B}$ using $\text{Caus}[\mathcal{C}](\mathbf{A}, \mathbf{B} \parr \mathbf{O})$ where $\mathbf{O}$ is an object into which we encode the outcome of the test (generalising the Binary Test Lemma from binary output $\mathbf{2}$ to any system). The test outcome is independent of any future handling of the output when the test is in $\text{Caus}[\mathcal{C}](\mathbf{A}, \mathbf{B} > \mathbf{O})$. Their equivalence is captured by the equation $\mathbf{A} \multimap (\mathbf{B} \parr \mathbf{O}) \cong \mathbf{A} \multimap (\mathbf{B} > \mathbf{O})$. So long as $\mathbf{O}$ has at least two distinct states, one can adapt the proof of the First-Order Theorem to show that this holds if and only if $\mathbf{B}$ is first-order.

The additional equations shown also help us to finally recapture the characterisation of first-order causal one-way signalling processes from FO$\multimap$ without assuming PC5. We can achieve this by simply rewriting the type expression using the equations until it is of the form of Proposition 2.6.2.

$$\begin{aligned}\mathbf{A^1} \multimap \left(\mathbf{B^1} \multimap \mathbf{C^1}\right) \multimap \mathbf{D^1} &= \mathbf{A^{1*}} \parr \left(\mathbf{B^{1*}} \parr \mathbf{C^1}\right)^* \parr \mathbf{D^1} \\ &= \mathbf{A^{1*}} < \left(\mathbf{B^{1*}} < \mathbf{C^1}\right)^* < \mathbf{D^1} \\ &\cong \mathbf{A^{1*}} < \mathbf{B^1} < \mathbf{C^{1*}} < \mathbf{D^1} \\ &\cong \left(\mathbf{A^{1*}} \parr \mathbf{B^1}\right) < \left(\mathbf{C^{1*}} < \mathbf{D^1}\right) \\ &= \left(\mathbf{A^1} \multimap \mathbf{B^1}\right) < \left(\mathbf{C^1} \multimap \mathbf{D^1}\right)\end{aligned} \tag{2.164}$$

In later sections, it will become useful to have a means of generating counterexample morphisms that fail non-signalling conditions in the sense that they can actively signal information. We can view this as a constructive contrapositive to the First-Order Theorem, working when we have at least two distinct states into which we can encode information to produce a (noisy) binary channel. Given such pairs of states in $\mathbf{A}$ and $\mathbf{B}$ for representing "true" and "false", we call $f : \mathbf{A} \to \mathbf{B}$ a *binary encoding* if it acts as a noisy binary channel over these states.

Definition 2.6.23: Binary Encoding

Let $\mathbf{A}, \mathbf{B} \in \mathrm{Ob}\left(\mathrm{Caus}\left[\mathcal{C}\right]\right)$ be objects that are not first-order dual $\left(\left|c_{\mathbf{A}}\right|, \left|c_{\mathbf{B}}\right| > 1\right)$. A *binary encoding* $f : \mathbf{A} \to \mathbf{B}$ for a choice of states $\rho_A \neq ⫰_{\mathbf{A}} \in c_{\mathbf{A}}$, $\rho_B \neq ⫰_{\mathbf{B}} \in c_{\mathbf{B}}$, is a morphism which actively signals some information:

$$\begin{gathered} ⫰_{\mathbf{A}} \fatsemi f = ⫰_{\mathbf{B}} \\ \exists \alpha.\ \left[\rho_A \fatsemi f\right] \sim \alpha \cdot \left[\rho_B\right] + \left(\mathrm{id}_I - \alpha\right) \cdot \left[⫰_{\mathbf{B}}\right] \not\sim \left[⫰_{\mathbf{B}}\right] \end{gathered} \tag{2.165}$$

Lemma 2.6.24

For any pair of objects $\mathbf{A}, \mathbf{B} \in \mathrm{Ob}\left(\mathrm{Caus}\left[\mathcal{C}\right]\right)$ and choices of distinct states $\rho_A \neq ⫰_{\mathbf{A}} \in c_{\mathbf{A}}$, $\rho_B \neq ⫰_{\mathbf{B}} \in c_{\mathbf{B}}$, there exists a binary encoding $f : \mathbf{A} \to \mathbf{B}$.

Proof. The key idea behind this proof is to use a dual basis to produce a perfect binary channel in $\mathrm{Sub}\left(\mathcal{C}\right)$, i.e. a process that sends ρ_A to ρ_B and $⫰_{\mathbf{A}}$ to $⫰_{\mathbf{B}}$. The final noisy channel in $\mathcal{C}$ will be some probabilistic combination of this perfect channel and the completely noisy channel $⫯_{\mathbf{A}} \fatsemi ⫰_{\mathbf{B}}$, which exists by APC5a. We finish by showing that this final channel is causal $\mathbf{A} \to \mathbf{B}$.

Let $\{\rho_i^A\}_{i \in \mathfrak{B}^{\mathcal{C}}_A}$ be a preferred basis for $\mathbf{A}$ with dual basis $\{e_i^A\}_{i \in \mathfrak{B}^{\mathcal{C}}_A}$ such that $\rho_1^A = \rho_A$ and $\rho_2^A = ⫰_{\mathbf{A}}$ (we can fix these states before applying Lemma 2.4.18 when generating the basis), and similarly $\left\{\rho_j^B\right\}_{j \in \mathfrak{B}^{\mathcal{C}}_B}$, $\left\{e_j^B\right\}_{j \in \mathfrak{B}^{\mathcal{C}}_B}$ for $\mathbf{B}$ with $\rho_1^B = \rho_B$ and

$\rho_2^B = \underset{\bullet}{|}_{\mathbf{B}}$.

We start by defining the following morphism:

$$\begin{aligned} t &:= e_1^A \fatsemi \left[\rho_1^B\right] + \sum_{i \neq 1} e_i^A \fatsemi \left[\rho_2^B\right] \\ &\sim e_1^A \fatsemi \left[\rho_1^B\right] + \left(\left[\overset{\bullet}{|}_{\mathbf{A}}\right] - e_1^A\right) \fatsemi \left[\rho_2^B\right] \\ &\in \mathrm{Sub}\left(\mathcal{C}\right)\left(A, B\right) \end{aligned} \tag{2.166}$$

We can view this as a binary test for e_1^A, with outcomes encoded into $\mathbf{B}$ instead of $\mathbf{2}$. t might not have a corresponding morphism in $\mathcal{C}$ (i.e. if $\mathcal{C}$ is $\mathrm{Mat}[\mathbb{R}^+]$ or CP^*, t may not be a positive operator), but we can still represent it as an affine combination of morphisms from $\mathcal{C}$: pick some representation $t \sim [t^+] - [t^-]$ and apply APC5a to $t^- \in \mathcal{C}(A, B)$ to give some invertible λ and $t' \in \mathcal{C}(A, B)$ such that $[t'] \sim \left[\lambda \cdot \overset{=}{\top}_A \fatsemi \underset{=}{\bot}_B\right] - [t^-]$ (wlog suppose λ is such that $\lambda\theta_{\mathbf{A}}^{-1}\mu_{\mathbf{B}}^{-1} + \mathrm{id}_I$ is invertible, otherwise we could freely pick a larger λ and t'). We then define

$$f := \left(\lambda\theta_{\mathbf{A}}^{-1}\mu_{\mathbf{B}}^{-1} + \mathrm{id}_I\right)^{-1} \cdot \left(t^+ + t'\right) \in \mathcal{C}\left(A, B\right) \tag{2.167}$$

so that t is an affine combination of f and $\overset{\bullet}{|}_{\mathbf{A}} \fatsemi \underset{\bullet}{|}_{\mathbf{B}}$, and we claim that it is causal $f \in \mathrm{Caus}\left[\mathcal{C}\right]\left(\mathbf{A}, \mathbf{B}\right)$. Consider an arbitrary effect $\pi \in c_{\mathbf{B}}^*$.

$$\begin{aligned} &[f \fatsemi \pi] \\ &\sim \left[\left(\lambda\theta_{\mathbf{A}}^{-1}\mu_{\mathbf{B}}^{-1} + \mathrm{id}_I\right)^{-1}\right] \cdot \left(\left[\lambda \cdot \overset{=}{\top}_A \fatsemi \underset{=}{\bot}_B\right] + t\right) \fatsemi [\pi] \\ &\sim \left[\left(\lambda\theta_{\mathbf{A}}^{-1}\mu_{\mathbf{B}}^{-1} + \mathrm{id}_I\right)^{-1}\right] \cdot \begin{pmatrix} \left[\lambda\theta_{\mathbf{A}}^{-1}\mu_{\mathbf{B}}^{-1} \cdot \overset{\bullet}{|}_{\mathbf{A}}\right] \fatsemi \left[\rho_2^B\right] \\ +e_1^A \fatsemi \left[\rho_1^B\right] \\ +\left[\overset{\bullet}{|}_{\mathbf{A}}\right] \fatsemi \left[\rho_2^B\right] \\ -e_1^A \fatsemi \left[\rho_2^B\right] \end{pmatrix} \fatsemi [\pi] \\ &\sim \left[\left(\lambda\theta_{\mathbf{A}}^{-1}\mu_{\mathbf{B}}^{-1} + \mathrm{id}_I\right)^{-1}\right] \cdot \begin{pmatrix} \left[\lambda\theta_{\mathbf{A}}^{-1}\mu_{\mathbf{B}}^{-1} \cdot \overset{\bullet}{|}_A\right] \\ +e_1^A \\ +\left[\overset{\bullet}{|}_{\mathbf{A}}\right] \\ -e_1^A \end{pmatrix} \\ &\sim \left[\left(\lambda\theta_{\mathbf{A}}^{-1}\mu_{\mathbf{B}}^{-1} + \mathrm{id}_I\right)^{-1}\right] \cdot \left[\lambda\theta_{\mathbf{A}}^{-1}\mu_{\mathbf{B}}^{-1} + \mathrm{id}_I\right] \cdot \left[\overset{\bullet}{|}_{\mathbf{A}}\right] \\ &\sim \left[\overset{\bullet}{|}_{\mathbf{A}}\right] \end{aligned} \tag{2.168}$$

So $f \fatsemi \pi \in c_{\mathbf{A}}^*$, i.e. f is causal, and there is no signalling from the output to the input (its partial transpose is in $c_{\mathbf{A}^* < \mathbf{B}}$). We can verify that it signals from the input to output.

$$\begin{aligned}
&[\overset{|}{\bullet}_{\mathbf{A}} \fatsemi f] \\
&\sim \left[\left(\lambda\theta_{\mathbf{A}}^{-1}\mu_{\mathbf{B}}^{-1} + \mathrm{id}_I\right)^{-1}\right] \cdot [\overset{|}{\bullet}_{\mathbf{A}}] \fatsemi \left(\left[\lambda \cdot \overset{=}{\top}_A \fatsemi \underset{=}{\bot}_B\right] + t\right) \\
&\sim \left[\left(\lambda\theta_{\mathbf{A}}^{-1}\mu_{\mathbf{B}}^{-1} + \mathrm{id}_I\right)^{-1}\right] \cdot [\rho_2^A] \fatsemi \begin{pmatrix} \quad [\lambda\theta_{\mathbf{A}}\mu_{\mathbf{B}}^{-1} \cdot \underset{|}{\bullet}_{\mathbf{A}}] \fatsemi [\rho_2^B] \\ +e_1^A \fatsemi [\rho_1^B] \\ +[\underset{|}{\bullet}_{\mathbf{A}}] \fatsemi [\rho_2^B] \\ -e_1^A \fatsemi [\rho_2^B] \end{pmatrix} \\
&\sim \left[\left(\lambda\theta_{\mathbf{A}}^{-1}\mu_{\mathbf{B}}^{-1} + \mathrm{id}_I\right)^{-1}\right] \cdot \left[\lambda\theta_{\mathbf{A}}^{-1}\mu_{\mathbf{B}}^{-1} + \mathrm{id}_I\right] \cdot [\rho_2^B] \\
&\sim [\overset{|}{\bullet}_{\mathbf{B}}]
\end{aligned} \tag{2.169}$$

$$\begin{aligned}
&[\rho_A \fatsemi f] \\
&\sim \left[\left(\lambda\theta_{\mathbf{A}}^{-1}\mu_{\mathbf{B}}^{-1} + \mathrm{id}_I\right)^{-1}\right] \cdot [\rho_A] \fatsemi \left(\left[\lambda \cdot \overset{=}{\top}_A \fatsemi \underset{=}{\bot}_B\right] + t\right) \\
&\sim \left[\left(\lambda\theta_{\mathbf{A}}^{-1}\mu_{\mathbf{B}}^{-1} + \mathrm{id}_I\right)^{-1}\right] \cdot [\rho_1^A] \fatsemi \begin{pmatrix} \quad [\lambda\theta_{\mathbf{A}}^{-1}\mu_{\mathbf{B}}^{-1} \cdot \underset{|}{\bullet}_{\mathbf{A}}] \fatsemi [\rho_2^B] \\ +e_1^A \fatsemi [\rho_1^B] \\ +[\underset{|}{\bullet}_{\mathbf{A}}] \fatsemi [\rho_2^B] \\ -e_1^A \fatsemi [\rho_2^B] \end{pmatrix} \\
&\sim \left[\left(\lambda\theta_{\mathbf{A}}^{-1}\mu_{\mathbf{B}}^{-1} + \mathrm{id}_I\right)^{-1} \cdot \left(\rho_B + \lambda\theta_{\mathbf{A}}^{-1}\mu_{\mathbf{B}}^{-1} \cdot \overset{|}{\bullet}_{\mathbf{B}}\right)\right]
\end{aligned} \tag{2.170}$$

We additionally note for the benefit of future proofs that f maps e_1^B to e_1^A (up to an invertible scalar).

$$\begin{aligned}
&[f] \fatsemi e_1^B \\
&\sim \left[\left(\lambda\theta_{\mathbf{A}}^{-1}\mu_{\mathbf{B}}^{-1} + \mathrm{id}_I\right)^{-1}\right] \cdot \begin{pmatrix} \quad [\lambda\theta_{\mathbf{A}}^{-1}\mu_{\mathbf{B}}^{-1} \cdot \underset{|}{\bullet}_{\mathbf{A}}] \fatsemi [\rho_2^B] \\ +e_1^A \fatsemi [\rho_1^B] \\ +[\underset{|}{\bullet}_{\mathbf{A}}] \fatsemi [\rho_2^B] \\ -e_1^A \fatsemi [\rho_2^B] \end{pmatrix} \fatsemi e_1^B \\
&\sim \left[\left(\lambda\theta_{\mathbf{A}}^{-1}\mu_{\mathbf{B}}^{-1} + \mathrm{id}_I\right)^{-1}\right] \cdot e_1^A
\end{aligned} \tag{2.171}$$

□

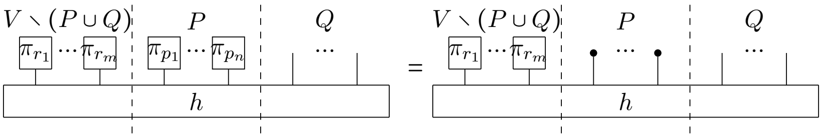
2.6.7 General Signalling Constraints

If we look back at Definition 2.1.7 for signal-consistency, we see that each non-signalling condition always partitions the points in the causal structure into two sets: viewing the condition as picking a space-like slice across the setup, we have the set of points that occur in the past of the slice and another set for those in the future. Under other circumstances, we may wish to consider more fine-grained non-signalling conditions where we need not slice through the entire setup.

Fix a set V of agents and objects $\{\mathbf{A}_v\}_{v\in V} \subseteq \mathrm{Ob}\left(\mathrm{Caus}\left[\mathcal{C}\right]\right)$ describing the local system provided to each agent. A multi-partite process h is a state of $\bigparr_{v \in V} \mathbf{A}_v$ precisely when it acts locally according to $\mathbf{A}_v$ from the perspective of each agent v, regardless of the actions of the other agents - i.e. the marginal at v, which need not be independent of the other systems, is causal for $\mathbf{A}_v$.

Given disjoint subsets $P, Q \subseteq V$, we say that h is one-way non-signalling from P to Q if, regardless of the actions chosen by $V \setminus (P \cup Q)$, the local behaviour at Q is independent of any choices of actions by agents in P.

$$\forall \{\pi_r\}_{r \in V \setminus (P \cup Q)} \,.\ \forall \{\pi_p\}_{p \in P} \,.$$



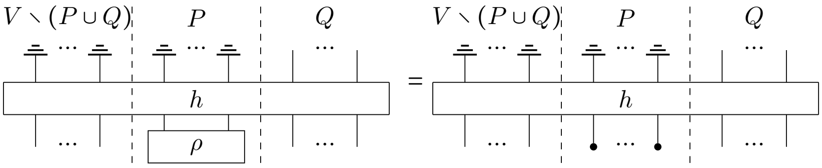
(2.172)

Combining the definitions of $\parr$ via Equation 2.52 and $<$ via Definition 2.6.1, this is precisely captured by the typing:

$$h : \left(\bigparr_{r \in V \setminus (P \cup Q)} \mathbf{A}_r \right) \parr \left(\left(\bigparr_{p \in P} \mathbf{A}_p \right) > \left(\bigparr_{q \in Q} \mathbf{A}_q \right) \right) \tag{2.173}$$

Again, if we pick each $\mathbf{A}_v$ to be a first-order channel $\mathbf{B}_v^{\mathbf{1}} \multimap \mathbf{C}_v^{\mathbf{1}}$ this exactly coincides with the usual presentation of a non-signalling condition, where the outputs of Q are independent of the inputs of P, but possibly dependent on the inputs at Q and $V \setminus (P \cup Q)$. By taking affine combinations of local actions at P, this even accounts for correlated and entangled states at their inputs.

$$\forall \rho : \bigotimes_{p \in P} \mathbf{B}_p^{\mathbf{1}} .$$



(2.174)

2.7 Unions and Intersections

We have already examined the prior work on intersections of objects in Section 2.3.6. In this section, we will dualise this to a *union* of types based on union of sets and examine the interplay between intersection, union, and the monoidal structures.

Similar to intersections, flatness of a set will generally not be preserved by taking the union. We need to impose the same assumption on the underlying object and normalisation scalars, which we will now denote as *set-compatibility* of the objects.

[9] As this is a new definition for this thesis, the $\pitchfork$ symbol was chosen for the main reason that it

Definition 2.7.1: Set-compatibility

Objects $\mathbf{A}, \mathbf{A}' \in \mathrm{Ob}\left(\mathrm{Caus}\left[\mathcal{C}\right]\right)$ are *set-compatible* $\mathbf{A} \text{⫛} \mathbf{A}'$[9] when they share the same carrier object $A \in \mathrm{Ob}\left(\mathcal{C}\right)$ and normalisation scalars $\mu_{\mathbf{A}} = \mu_{\mathbf{A}'}$ and $\theta_{\mathbf{A}} = \theta_{\mathbf{A}'}$.

Proposition 2.7.2

For any objects $\mathbf{A}, \mathbf{A}' \in \mathrm{Ob}\left(\mathrm{Caus}\left[\mathcal{C}\right]\right)$, the following are equivalent:

1. $\mathbf{A} \text{⫛} \mathbf{A}'$;
2. They share the same carrier object $A = A' \in \mathrm{Ob}\left(\mathcal{C}\right)$ and $c_{\mathbf{A}} \cap c_{\mathbf{A}'} \neq \varnothing$;
3. The identities on their carrier objects $\mathrm{id}_A \in \mathcal{C}\left(A, A\right)$, $\mathrm{id}_{A'} \in \mathcal{C}\left(A', A'\right)$ form a *causal cospan*, i.e. there exists some $\mathbf{B} \in \mathrm{Ob}\left(\mathrm{Caus}\left[\mathcal{C}\right]\right)$ such that $\mathrm{id}_A : \mathbf{A} \to \mathbf{B}$ and $\mathrm{id}_{A'} : \mathbf{A}' \to \mathbf{B}$;
4. The identities on their carrier objects $\mathrm{id}_A \in \mathcal{C}\left(A, A\right)$, $\mathrm{id}_{A'} \in \mathcal{C}\left(A', A'\right)$ form a *causal span*, i.e. there exists some $\mathbf{B} \in \mathrm{Ob}\left(\mathrm{Caus}\left[\mathcal{C}\right]\right)$ such that $\mathrm{id}_A : \mathbf{B} \to \mathbf{A}$ and $\mathrm{id}_{A'} : \mathbf{B} \to \mathbf{A}'$.

Proof. $1 \implies 2$: $c_{\mathbf{A}} \ni \text{⫰}_{\mathbf{A}} = \mu_{\mathbf{A}} \cdot \text{⏚}_A = \mu_{\mathbf{A}'} \cdot \text{⏚}_A = \text{⫰}_{\mathbf{A}'} \in c_{\mathbf{A}'}$ gives an element in the intersection.

$1 \impliedby 2$: By flatness, $\text{⫰}_{\mathbf{A}} \in c_{\mathbf{A}}$ and $\text{⫯}_{\mathbf{A}} \in c^*_{\mathbf{A}}$.

$$\mu_{\mathbf{A}}\theta_{\mathbf{A}}d_A = \mu_{\mathbf{A}}\theta_{\mathbf{A}} \cdot \text{⏚}_A \fatsemi \text{⍑}_A = \text{⫰}_{\mathbf{A}} \fatsemi \text{⫯}_{\mathbf{A}} = \mathrm{id}_I \tag{2.175}$$

Similarly $\mu_{\mathbf{A}'}\theta_{\mathbf{A}'}d_A = \mathrm{id}_I$, so $\mu_{\mathbf{A}} = \mu_{\mathbf{A}'}$ iff $\theta_{\mathbf{A}} = \theta_{\mathbf{A}'}$ by invertibility and hence we only need to show one.

Suppose there exists some $\rho \in c_{\mathbf{A}} \cap c_{\mathbf{A}'}$. In particular, $\rho \fatsemi \text{⫯}_{\mathbf{A}} = \mathrm{id}_I = \rho \fatsemi \text{⫯}_{\mathbf{A}'}$. Then

$$\begin{aligned}
\theta_{\mathbf{A}} &= \theta_{\mathbf{A}} \cdot \rho \fatsemi \text{⫯}_{\mathbf{A}'} \\
&= \theta_{\mathbf{A}}\theta_{\mathbf{A}'} \cdot \rho \fatsemi \text{⍑}_A \\
&= \theta_{\mathbf{A}'} \cdot \rho \fatsemi \text{⫯}_{\mathbf{A}} \\
&= \theta_{\mathbf{A}'}
\end{aligned} \tag{2.176}$$

$1 \implies 3$: Since the carrier object is the same, $\mathrm{id}_A = \mathrm{id}_{A'} \in \mathcal{C}\left(A, A\right) = \mathcal{C}\left(A', A'\right)$. Let $\mathbf{B} = \left(A, \{\text{⫯}_{\mathbf{A}}\}^*\right)$. We can show causality by preservation of

does not clash with any other existing notation in the relevant literature. It also reinforces that this is a symmetric relation and has visual similarities to ∩ and ∈ which connects it to sets.

effects (Equation 2.44), i.e. that $\text{⫯}_{\mathbf{A}} \in c^*_{\mathbf{A}}$ and $\text{⫯}_{\mathbf{A}} = \theta_{\mathbf{A}} \cdot \overset{=}{\top}_A = \theta_{\mathbf{A}'} \cdot \overset{=}{\top}_A = \text{⫯}_{\mathbf{A}'} \in c^*_{\mathbf{A}'}$, which are both given by flatness.

$1 \Longleftarrow 3$: $\mathrm{id}_A : \mathbf{A} \to \mathbf{B}$ implies the carrier object of $\mathbf{B}$ is A, and the same for $\mathrm{id}_{A'} : \mathbf{A}' \to \mathbf{B}$ implies $A = A'$.

Using the same argument from the $1 \Longleftarrow 2$ case, it suffices to show $\theta_{\mathbf{A}} = \theta_{\mathbf{A}'}$. By preservation of effects, $\text{⫯}_{\mathbf{B}} \in c^*_{\mathbf{A}}$ and $\text{⫯}_{\mathbf{B}} \in c^*_{\mathbf{A}'}$, so by uniqueness of flatness scalars $\theta_{\mathbf{A}} = \theta_{\mathbf{B}} = \theta_{\mathbf{A}'}$.

$1 \Longleftrightarrow 4$: Similar, using $\mathbf{B} = (A, \{\text{⫰}_{\mathbf{A}}\})$. □

Proposition 2.7.3

For any set-compatible objects $\mathbf{A} \pitchfork \mathbf{A}'$, the union object

$$\mathbf{A} \cup \mathbf{A}' := \left(A, (c_{\mathbf{A}} \cup c_{\mathbf{A}'})^{**}\right) \tag{2.177}$$

$$= (\mathbf{A}^* \cap \mathbf{A}'^*)^* \tag{2.178}$$

yields a pushout for the causal span

$$\begin{tikzcd}
\mathbf{A}'' \arrow[r, "\mathrm{id}_A"] \arrow[d, "\mathrm{id}_{A'}"'] & \mathbf{A} \arrow[d, "\mathrm{id}_A"] \arrow[dl, phantom, "\ulcorner"] \\
\mathbf{A}' \arrow[r, "\mathrm{id}_{A'}"] & \mathbf{A} \cup \mathbf{A}'
\end{tikzcd} \tag{2.179}$$

Proof. The union definition is the de Morgan dual of intersection

$$\begin{aligned}
\pi \in c^*_{\mathbf{A}\cup\mathbf{A}'} &\Longleftrightarrow \pi \in (c_{\mathbf{A}} \cup c_{\mathbf{A}'})^* \\
&\Longleftrightarrow (\forall \rho \in c_{\mathbf{A}} . \rho \fatsemi \pi = \mathrm{id}_I) \wedge (\forall \rho' \in c_{\mathbf{A}'} . \rho' \fatsemi \pi = \mathrm{id}_I) \\
&\Longleftrightarrow \pi \in c^*_{\mathbf{A}} \cap c^*_{\mathbf{A}'} \\
&\Longleftrightarrow \pi \in c_{\mathbf{A}^* \cap \mathbf{A}'^*}
\end{aligned} \tag{2.180}$$

so the pushout property holds by duality. □

Example 2.7.4

In Example 2.3.27, we saw that $\mathbf{A} \otimes \mathbf{B} \pitchfork \mathbf{A} \parr \mathbf{B}$ but their intersection degenerates to $\mathbf{A} \otimes \mathbf{B}$. We can similarly use $c_{\mathbf{A}\otimes\mathbf{B}} \subseteq c_{\mathbf{A}\parr\mathbf{B}}$ to show that their union degenerates:

$$(\mathbf{A} \otimes \mathbf{B}) \cup (\mathbf{A} \parr \mathbf{B}) = \mathbf{A} \parr \mathbf{B} \tag{2.181}$$

Just taking a union $c_{\mathbf{A}} \cup c_{\mathbf{A}'}$ will generally not yield a closed set as it will fail to account for any affine combinations of terms from different sets. We choose to still use standard union notation for objects since they arise from the same pushout construction and it can still be viewed as a strict union with respect to a preferred basis.

Proposition 2.7.5

Suppose $\mathbf{A} \pitchfork \mathbf{A}'$, and let $\{\rho_i\}_{i \in \mathfrak{B}^{\mathcal{C}}_{A}}$ be a preferred basis for both $\mathbf{A}$ and $\mathbf{A}'$. Then

$$c_{\mathbf{A} \cup \mathbf{A}'} = \left\{ v \;\middle|\; \begin{array}{c} [v] \sim \sum_i s_i \cdot [\rho_i], \\ \sum_i s_i \sim \mathrm{id}_I, \\ \forall i \in \overline{\mathfrak{B}_{\mathbf{A}}} \cap \overline{\mathfrak{B}_{\mathbf{A}'}}.\ s_i \sim 0 \end{array} \right\}$$

Proof. This is immediate from viewing the states of $\mathbf{A} \cup \mathbf{A}'$ as affine combinations of states of $\mathbf{A}$ and states of $\mathbf{A}'$. □

One of the most significant results we will show about unions and intersections is that all the monoidal structures of $\mathrm{Caus}[\mathcal{C}]$ will distribute over them. This means that any object built from these operators can always be expressed in a form where unions and intersections have been lifted to the top level.

Lemma 2.7.6: Setwise Distributivity

Given objects $\mathbf{A}, \mathbf{A}', \mathbf{B} \in \mathrm{Ob}(\mathrm{Caus}[\mathcal{C}])$ with $\mathbf{A} \pitchfork \mathbf{A}'$, each $\square \in \{\otimes, \parr, <, >, \times, \oplus\}$ distributes over each $\diamond \in \{\cup, \cap\}$.

$$(\mathbf{A} \diamond \mathbf{A}') \square \mathbf{B} = (\mathbf{A} \square \mathbf{B}) \diamond (\mathbf{A}' \square \mathbf{B})$$

Proof.

- $(\mathbf{A} \cap \mathbf{A}') < \mathbf{B} = (\mathbf{A} < \mathbf{B}) \cap (\mathbf{A}' < \mathbf{B})$:

$$\begin{aligned} \rho \in c_{(\mathbf{A} \cap \mathbf{A}') < \mathbf{B}} &\iff \forall \pi \in c^*_{\mathbf{B}}.\ \rho \fatsemi (\mathrm{id}_A \otimes \pi) = \rho \fatsemi (\mathrm{id}_A \otimes \top_{\mathbf{B}}) \in c_{\mathbf{A} \cap \mathbf{A}'} \\ &\iff \forall \pi \in c^*_{\mathbf{B}}.\ \rho \fatsemi (\mathrm{id}_A \otimes \pi) = \rho \fatsemi (\mathrm{id}_A \otimes \top_{\mathbf{B}}) \in c_{\mathbf{A}} \\ &\qquad \wedge \forall \pi \in c^*_{\mathbf{B}}.\ \rho \fatsemi (\mathrm{id}_A \otimes \pi) = \rho \fatsemi (\mathrm{id}_A \otimes \top_{\mathbf{B}}) \in c_{\mathbf{A}'} \qquad (2.182) \\ &\iff \rho \in c_{\mathbf{A} < \mathbf{B}} \wedge \rho \in c_{\mathbf{A}' < \mathbf{B}} \\ &\iff \rho \in c_{(\mathbf{A} < \mathbf{B}) \cap (\mathbf{A}' < \mathbf{B})} \end{aligned}$$

- $(\mathbf{A} \cup \mathbf{A}') < \mathbf{B} = (\mathbf{A} < \mathbf{B}) \cup (\mathbf{A}' < \mathbf{B})$: Dual to above.
- $(\mathbf{A} \cap \mathbf{A}') > \mathbf{B} = (\mathbf{A} > \mathbf{B}) \cap (\mathbf{A}' > \mathbf{B})$: Note that equality of carrier objects

and normalisation scalars give $⫯_{\mathbf{A}\cap\mathbf{A}'} = ⫯_{\mathbf{A}} = ⫯_{\mathbf{A}'}$.

$$\begin{aligned}
\rho \in c_{(\mathbf{A}\cap\mathbf{A}')>\mathbf{B}} \iff & \forall \pi \in c^*_{(A\cap A')}. \\
& \quad \rho \fatsemi (\pi \otimes \mathrm{id}_B) = \rho \fatsemi (⫯_{\mathbf{A}\cap\mathbf{A}'} \otimes \mathrm{id}_B) \in c_{\mathbf{B}} \\
\iff & \forall \pi \in \left(c^*_{\mathbf{A}} \cup c^*_{\mathbf{A}}\right)^{**}. \\
& \quad \rho \fatsemi (\pi \otimes \mathrm{id}_B) = \rho \fatsemi (⫯_{\mathbf{A}\cap\mathbf{A}'} \otimes \mathrm{id}_B) \in c_{\mathbf{B}} \\
\iff & \forall \pi \in c^*_{\mathbf{A}} \cup c^*_{\mathbf{A}}. \\
& \quad \rho \fatsemi (\pi \otimes \mathrm{id}_B) = \rho \fatsemi (⫯_{\mathbf{A}\cap\mathbf{A}'} \otimes \mathrm{id}_B) \in c_{\mathbf{B}} \\
\iff & \forall \pi \in c^*_{\mathbf{A}}.\ \rho \fatsemi (\pi \otimes \mathrm{id}_B) = \rho \fatsemi (⫯_{\mathbf{A}} \otimes \mathrm{id}_B) \in c_{\mathbf{B}} \\
& \wedge \forall \pi \in c^*_{\mathbf{A}'}.\ \rho \fatsemi (\pi \otimes \mathrm{id}_B) = \rho \fatsemi (⫯_{\mathbf{A}'} \otimes \mathrm{id}_B) \in c_{\mathbf{B}} \\
\iff & \rho \in c_{\mathbf{A}>\mathbf{B}} \wedge \rho \in c_{\mathbf{A}'>\mathbf{B}} \\
\iff & \rho \in c_{(\mathbf{A}>\mathbf{B})\cap(\mathbf{A}'>\mathbf{B})}
\end{aligned} \tag{2.183}$$

- $(\mathbf{A} \cup \mathbf{A}') > \mathbf{B} = (\mathbf{A} > \mathbf{B}) \cup (\mathbf{A}' > \mathbf{B})$: Dual to above.

- $(\mathbf{A} \cap \mathbf{A}') \otimes \mathbf{B} = (\mathbf{A} \otimes \mathbf{B}) \cap (\mathbf{A}' \otimes \mathbf{B})$: Consider each direction of inclusion of causal states. For $\subseteq$, given any $\rho_A \in c_{\mathbf{A}} \cap c_{\mathbf{A}'}$ and $\rho_B \in c_{\mathbf{B}}$, we have $\rho_A \otimes \rho_B \in c_{\mathbf{A}\otimes\mathbf{B}}$ and $\rho_A \otimes \rho_B \in c_{\mathbf{A}'\otimes\mathbf{B}}$. All other causal states follow by affine closure. For $\supseteq$, we expand $h \in c_{\mathbf{A}\otimes\mathbf{B}} \cap c_{\mathbf{A}'\otimes\mathbf{B}}$ in terms of a joint preferred basis for both $\mathbf{A}$ and $\mathbf{A}'$ and another preferred basis for $\mathbf{B}$. By Proposition 2.4.25, the affine combination excludes any basis elements on the left that aren't causal for $\mathbf{A}$, and excludes any that aren't causal for $\mathbf{A}'$, leaving only basis elements in $c_{\mathbf{A}\cap\mathbf{A}'}$. This gives a representation as an affine combination of product states from $c_{(\mathbf{A}\cap\mathbf{A}')\otimes\mathbf{B}}$.

- $(\mathbf{A} \cup \mathbf{A}') \otimes \mathbf{B} = (\mathbf{A} \otimes \mathbf{B}) \cup (\mathbf{A}' \otimes \mathbf{B})$: By $*$-autonomy, there is an adjunction between $\otimes$ and $\multimap$, so $\otimes$ preserves all colimits which includes pushouts.

- $(\mathbf{A} \cap \mathbf{A}') \parr \mathbf{B} = (\mathbf{A} \parr \mathbf{B}) \cap (\mathbf{A}' \parr \mathbf{B})$: Dual to $\otimes, \cup$ case.

- $(\mathbf{A} \cup \mathbf{A}') \parr \mathbf{B} = (\mathbf{A} \parr \mathbf{B}) \cup (\mathbf{A}' \parr \mathbf{B})$: Dual to $\otimes, \cap$ case.

- $(\mathbf{A} \cap \mathbf{A}') \times \mathbf{B} = (\mathbf{A} \times \mathbf{B}) \cap (\mathbf{A}' \times \mathbf{B})$:

$$\begin{aligned}
c_{(\mathbf{A}\cap\mathbf{A}')\times\mathbf{B}} &= \left\{ \langle \rho_A, \rho_B \rangle \mid \rho_A \in c_{\mathbf{A}\cap\mathbf{A}'}, \rho_B \in c_{\mathbf{B}} \right\} \\
&= \left\{ \langle \rho_A, \rho_B \rangle \mid \rho_A \in c_{\mathbf{A}} \cap c_{\mathbf{A}'}, \rho_B \in c_{\mathbf{B}} \right\}
\end{aligned}$$

$$\begin{aligned}&= \{\langle \rho_A, \rho_B \rangle \mid \rho_A \in c_{\mathbf{A}}, \rho_B \in c_{\mathbf{B}}\} \\ &\quad \cap \{\langle \rho_A, \rho_B \rangle \mid \rho_A \in c_{\mathbf{A}'}, \rho_B \in c_{\mathbf{B}}\} \\ &= c_{\mathbf{A} \times \mathbf{B}} \cap c_{\mathbf{A}' \times \mathbf{B}} \\ &= c_{(\mathbf{A} \times \mathbf{B}) \cap (\mathbf{A}' \times \mathbf{B})}\end{aligned} \tag{2.184}$$

- $(\mathbf{A} \cup \mathbf{A}') \times \mathbf{B} = (\mathbf{A} \times \mathbf{B}) \cup (\mathbf{A}' \times \mathbf{B})$: We also break this down into inclusions in each direction. For $\subseteq$, if we are given a pairing where the left term is an affine combination of causal states mixing $c_{\mathbf{A}}$ and $c_{\mathbf{A}'}$, we can lift the sum up to give an affine combination of pairings from $c_{\mathbf{A} \times \mathbf{B}}$ and $c_{\mathbf{A}' \times \mathbf{B}}$.

$$\begin{aligned}\left\langle \sum_i s_i \cdot \left[\rho_i^A\right], [\rho_B] \right\rangle &\sim \left\langle \sum_i s_i \cdot \left[\rho_i^A\right], \sum_i s_i \cdot [\rho_B] \right\rangle \\ &\sim \sum_i s_i \cdot \left[\left\langle \rho_i^A, \rho_B \right\rangle\right]\end{aligned} \tag{2.185}$$

  As for $\supseteq$, expand some $h \in (c_{\mathbf{A} \times \mathbf{B}} \cup c_{\mathbf{A}' \times \mathbf{B}})^{**}$ as an affine combination of pairs. We push the summation into the pairings and recognise the affine combination of terms from $c_{\mathbf{B}}$ as an element of $\mathcal{C}(I, B)$, hence it is an element of $c_{\mathbf{B}}$ itself. We are then left with a single pair of an element of $c_{\mathbf{B}}$ and an affine combination of elements from $c_{\mathbf{A}}$ and $c_{\mathbf{A}'}$.

$$\begin{aligned}\sum_i s_i \cdot \left[\left\langle \rho_i^A, \rho_i^B \right\rangle\right] &\sim \left\langle \sum_i s_i \cdot \left[\rho_i^A\right], \sum_i s_i \cdot \left[\rho_i^B\right] \right\rangle \\ &\sim \left\langle \sum_i s_i \cdot \left[\rho_i^A\right], \left[h \fatsemi p_B\right] \right\rangle\end{aligned} \tag{2.186}$$

- $(\mathbf{A} \cap \mathbf{A}') \oplus \mathbf{B} = (\mathbf{A} \oplus \mathbf{B}) \cap (\mathbf{A}' \oplus \mathbf{B})$: Dual to $\times, \cup$ case.

- $(\mathbf{A} \cup \mathbf{A}') \oplus \mathbf{B} = (\mathbf{A} \oplus \mathbf{B}) \cup (\mathbf{A}' \oplus \mathbf{B})$: Dual to $\times, \cap$ case. □

An additional distributivity equation holds for $\multimap$, but the mixed variance switches between union and intersection in line with the de Morgan duality.

$$(\mathbf{A} \cup \mathbf{A}') \multimap \mathbf{B} = (\mathbf{A} \multimap \mathbf{B}) \cap (\mathbf{A}' \multimap \mathbf{B}) \tag{2.187}$$

$$(\mathbf{A} \cap \mathbf{A}') \multimap \mathbf{B} = (\mathbf{A} \multimap \mathbf{B}) \cup (\mathbf{A}' \multimap \mathbf{B}) \tag{2.188}$$

$$\mathbf{B} \multimap (\mathbf{A} \cup \mathbf{A}') = (\mathbf{B} \multimap \mathbf{A}) \cup (\mathbf{B} \multimap \mathbf{A}') \tag{2.189}$$

$$\mathbf{B} \multimap (\mathbf{A} \cap \mathbf{A}') = (\mathbf{B} \multimap \mathbf{A}) \cap (\mathbf{B} \multimap \mathbf{A}') \tag{2.190}$$

If we aren't looking for an exact equality of types, instead considering embedding of spaces via natural transformations, we can apply distributivity rules between intersection and union themselves to always yield an "intersection of unions" form.

Proposition 2.7.7

$$(\mathbf{A} \cap \mathbf{C}) \cup (\mathbf{B} \cap \mathbf{C}) \Rightarrow (\mathbf{A} \cup \mathbf{B}) \cap \mathbf{C} \tag{2.191}$$

$$(\mathbf{A} \cap \mathbf{B}) \cup \mathbf{C} \Rightarrow (\mathbf{A} \cup \mathbf{C}) \cap (\mathbf{B} \cup \mathbf{C}) \tag{2.192}$$

Proof. Any state of $(\mathbf{A} \cap \mathbf{C}) \cup (\mathbf{B} \cap \mathbf{C})$ is an affine combination $\sum_i s_i \cdot [\rho_i]$ where each ρ_i is either in $c_\mathbf{A}$ and $c_\mathbf{C}$, or it is in $c_\mathbf{B}$ and $c_\mathbf{C}$. Either way, each ρ_i is in $c_\mathbf{C}$ and so must be $\sum_i s_i \cdot [\rho_i]$, and the sum is also an affine combination of states from $c_\mathbf{A} \cup c_\mathbf{B}$. The second natural transformation is given by duality. □

Remark 2.7.8

Hoffreumon & Oreshkov showed that a theory matching an inductively-defined subcategory of Caus $[\mathrm{CP}^*]$ admitted this distribution law as an equality $(A \cup B) \cap C = (A \cap C) \cup (B \cap C)$ [64]. This allowed them to reach either an "intersection of unions" or a "union of intersections" form without loss of precision. Unfortunately, this equality doesn't generalise to the full causal category. Consider the space $3 = I \oplus I \oplus I$ in $\mathrm{Mat}[\mathbb{R}^+]$, which has a dual basis formed by the injections and projections. We can define three flat, closed spaces by $\mathbf{A_i} = \left(3, \left\{\iota_i, \frac{1}{3}⏚_3\right\}^{**}\right)$ $(i \in \{1,2,3\})$. The intersection between any pair of these spaces is the singleton space $\left(3, \left\{\frac{1}{3}⏚_3\right\}\right) \cong \mathbf{3^{1*}}$ and the union of any pair is the full first-order space $\mathbf{3^1}$. Then $(\mathbf{A_1} \cup \mathbf{A_2}) \cap \mathbf{A_3} = \mathbf{3^1} \cap \mathbf{A_3} = \mathbf{A_3}$ but $(\mathbf{A_1} \cap \mathbf{A_3}) \cup (\mathbf{A_2} \cap \mathbf{A_3}) \cong \mathbf{3^{1*}} \cup \mathbf{3^{1*}} = \mathbf{3^{1*}}$.

Conceptually, there are a lot of similarities between $\cup$ and $\oplus$. Given $\mathbf{A} \pitchfork \mathbf{A'}$, we could also view $\mathbf{A} \cup \mathbf{A'}$ as describing some form of internal choice between providing a state of $\mathbf{A}$ or a state of $\mathbf{A'}$, where this choice is fixed upon the creation of the state. Where this contrasts with $\oplus$ is that the information detailing whether we have a state of $\mathbf{A}$ or $\mathbf{A'}$ is not available (ignoring the possibility of identifying the state through tomography which usually requires access to many identical copies of the state). This forces any effect to handle each case identically (which is always possible by set-compatibility), as opposed to copairings which can use the knowledge of the state provided to choose between different effects. Given a state of $\mathbf{A} \oplus \mathbf{A'}$, we can always choose to freely "forget" the tagging information by applying the *codiagonal* morphism.

$$\nabla_A \coloneqq [\mathrm{id}_A, \mathrm{id}_A] : \mathbf{A} \oplus \mathbf{A}' \to \mathbf{A} \cup \mathbf{A}' \tag{2.193}$$

If we look at discarding for $\mathbf{A} \oplus \mathbf{A}'$, we can then see that this factorises via the codiagonal in two ways, distinguishing between discarding the classical tagging information and the physical system in either order.

$$\text{⫯}_{\mathbf{A} \oplus \mathbf{A}'} = [\text{⫯}_{\mathbf{A}}, \text{⫯}_{\mathbf{A}'}] = \nabla_A \fatsemi \text{⫯}_{\mathbf{A}} = [\text{⫯}_{\mathbf{A}} \fatsemi \mathtt{t}, \text{⫯}_{\mathbf{A}'} \fatsemi \mathtt{f}] \fatsemi \nabla_I \tag{2.194}$$

Similarly by duality, $\mathbf{A} \cap \mathbf{A}'$ describes an external choice but without the ability to act conditionally based on that choice. The *diagonal* morphism realises that this is just a special case of conditional actions where both branches act identically.

$$\Delta_A \coloneqq \langle \mathrm{id}_A, \mathrm{id}_A \rangle : \mathbf{A} \cap \mathbf{A}' \to \mathbf{A} \times \mathbf{A}' \tag{2.195}$$

2.7.1 Tensor is the Non-Signalling Space

The seq product $<$ not only faithfully captures one-way signalling processes on first-order systems, but on any choice of local systems, thus working for higher-order protocol descriptions. On the other hand, we have so far only established that $\otimes$ characterises non-signalling on first-order causal processes. Kissinger and Uijlen's proof [78, Theorem 6.2] relied on PC5 which is a specific property of first-order systems (Corollary 2.4.28). Similarly, the existing results which characterise non-signalling processes as affine combinations of separable processes [60, 24, 21] only considered first-order processes. However, by using the characterisations we already have in terms of preferred bases, we can generalise the result to hold for arbitrary objects in $\mathrm{Caus}[\mathcal{C}]$ for any additive precausal category $\mathcal{C}$.

Theorem 2.7.9: Non-signalling Theorem

$$\mathbf{A} \otimes \mathbf{B} = (\mathbf{A} < \mathbf{B}) \cap (\mathbf{A} > \mathbf{B})$$

Proof. Compare the characterisations of Proposition 2.4.25 and Lemma 2.6.8. For $\supseteq$, all conditions are immediately provided. For $\subseteq$, the only remaining conditions are of the form $\forall i \in \overline{\mathfrak{B}}_{\mathbf{A}}.\ \sum_j s_{i,j} \sim 0$ which holds because each element in the sum is itself zero. □

For any inductively-defined higher-order objects $\mathbf{A}$, $\mathbf{B}$, one could arguably obtain this result by embedding them into first-order processes (providing all inputs up front and delaying all outputs to the end) and applying the existing results to demonstrate that a process $\mathbf{A} \otimes \mathbf{B}$ is non-signalling. However, any constructive representations (e.g. marginals, decompositions via semi-localisability, affine combinations of separable processes) will only be constrained to be first-order processes and may not themselves be causal for the higher-order $\mathbf{A}$, $\mathbf{B}$. The advantage given by the Non-signalling Theorem is that it constructively preserves any local structure within $\mathbf{A}$ and $\mathbf{B}$, avoiding such issues entirely.

We now have classifications of all three of $\otimes$, $<$, and $⅋$ that describe them as different layers of a signalling hierarchy, where one embeds into the next via Equation 2.146.

- States of $\mathbf{A} \otimes \mathbf{B}$ are the bipartite states that locally act like $\mathbf{A}$ and $\mathbf{B}$ with no information passing between them;
- States of $\mathbf{A} < \mathbf{B}$ are the bipartite states that locally act like $\mathbf{A}$ and $\mathbf{B}$ with no information passing from $\mathbf{B}$ to $\mathbf{A}$ but possibly some information passed in the other direction;
- States of $\mathbf{A} ⅋ \mathbf{B}$ are the bipartite states that locally act like $\mathbf{A}$ and $\mathbf{B}$ with possible information being passed in either direction.

The extent of information signalling admitted by each level in this hierarchy can be captured equationally in other forms by distributivity laws with the additive operators.

Remark 2.7.10

Informally, distributivity between the additives and multiplicatives holds whenever the parallel content cannot influence the internal/external choices. Taking a step back to $\mathcal{C}$, the monoidal and biproduct structures distribute nicely, so we don't need to be cautious about distributivity at the level of morphisms.

$$\begin{aligned} \delta &:= [\mathrm{id}_A \otimes \iota_B, \mathrm{id}_A \otimes \iota_C] = p_{A\otimes B} ⨟ (\mathrm{id}_A \otimes \iota_B) + p_{A\otimes C} ⨟ (\mathrm{id}_A \otimes \iota_C) && (2.196)\\ &: (A \otimes B) \oplus (A \otimes C) \to A \otimes (B \oplus C) && (2.197)\\ \delta^{-1} &:= \langle \mathrm{id}_A \otimes p_B, \mathrm{id}_A \otimes p_C \rangle = (\mathrm{id}_A \otimes p_B) ⨟ \iota_{A\otimes B} + (\mathrm{id}_A \otimes p_C) ⨟ \iota_{A\otimes C} && (2.198)\\ &: A \otimes (B \oplus C) \to (A \otimes B) \oplus (A \otimes C) && (2.199) \end{aligned}$$

When does this lift to $\mathrm{Caus}[\mathcal{C}]$? For $\otimes$ and $⅋$, it follows the regular expectations of MALL (MLL with additives). Any $\star$-autonomous category has an adjunction between $\otimes$ and $\multimap$, so the left-adjunct $\otimes$ preserves all colimits and dually $⅋$ will preserve all limits.

$$(\mathbf{A} \otimes \mathbf{B}) \oplus (\mathbf{A} \otimes \mathbf{C}) \cong \mathbf{A} \otimes (\mathbf{B} \oplus \mathbf{C}) \tag{2.200}$$

$$(\mathbf{A} \parr \mathbf{B}) \times (\mathbf{A} \parr \mathbf{C}) \cong \mathbf{A} \parr (\mathbf{B} \times \mathbf{C}) \tag{2.201}$$

As for the other combinations, δ and δ^{-1} are still causal for one direction (these can be obtained by composing the interchange laws of Equations 2.156-2.157 with the diagonal $\mathbf{A} \to \mathbf{A} \times \mathbf{A}$ or codiagonal $\mathbf{A} \oplus \mathbf{A} \to \mathbf{A}$).

$$\delta : (\mathbf{A} \parr \mathbf{B}) \oplus (\mathbf{A} \parr \mathbf{C}) \to \mathbf{A} \parr (\mathbf{B} \oplus \mathbf{C}) \tag{2.202}$$

$$\delta^{-1} : \mathbf{A} \otimes (\mathbf{B} \times \mathbf{C}) \to (\mathbf{A} \otimes \mathbf{B}) \times (\mathbf{A} \otimes \mathbf{C}) \tag{2.203}$$

However, when we look at the final cases (the distributivity linear implications that are not provable in MALL), they do not preserve causality. For a concrete example, let's take $\mathbf{A} = \mathbf{2}^*$ and $\mathbf{B} = \mathbf{C} = \mathbf{I}$ and consider the following state:

$$\rho \coloneqq \mathtt{t} \otimes \mathtt{t} + \mathtt{f} \otimes \mathtt{f} : \mathbf{2}^* \parr (\mathbf{I} \oplus \mathbf{I}) = \mathbf{2} \multimap \mathbf{2} \tag{2.204}$$

This just encodes an identity channel on a classical bit, so the choice between $\mathbf{B}$ and $\mathbf{C}$ will clearly be dependent on external choices at $\mathbf{A}$. This is in contrast with $(\mathbf{A} \parr \mathbf{B}) \oplus (\mathbf{A} \parr \mathbf{C})$ where the (probabilistic) choice between $\mathbf{B}$ and $\mathbf{C}$ must be fixed for any state and hence independent of actions at $\mathbf{A}$. This is clear when we apply δ^{-1} to ρ and compare the result with $c_{(\mathbf{A} \parr \mathbf{B}) \oplus (\mathbf{A} \parr \mathbf{C})}$.

$$\rho \fatsemi \delta^{-1} = (\mathtt{t} \otimes \mathrm{id}_I) \fatsemi \iota_{A \otimes B} + (\mathtt{f} \otimes \mathrm{id}_I) \fatsemi \iota_{A \otimes C} \tag{2.205}$$

$$\notin c_{(\mathbf{A} \parr \mathbf{B}) \oplus (\mathbf{A} \parr \mathbf{C})} = \left\{ \left(\text{⏚}_{2^*} \otimes \mathrm{id}_I \right) \fatsemi \iota_{A \otimes B}, \left(\text{⏚}_{2^*} \otimes \mathrm{id}_I \right) \fatsemi \iota_{A \otimes C} \right\}^{**} \tag{2.206}$$

The argument that δ is not causal for $(\mathbf{A} \otimes \mathbf{B}) \times (\mathbf{A} \otimes \mathbf{C}) \to \mathbf{A} \otimes (\mathbf{B} \times \mathbf{C})$ follows by duality.

As for $<$, the one-way signalling condition means we always preserve internal and external choices in the past.

$$(\mathbf{A} < \mathbf{C}) \oplus (\mathbf{B} < \mathbf{C}) \cong (\mathbf{A} \oplus \mathbf{B}) < \mathbf{C} \tag{2.207}$$

$$(\mathbf{A} < \mathbf{C}) \times (\mathbf{B} < \mathbf{C}) \cong (\mathbf{A} \times \mathbf{B}) < \mathbf{C} \tag{2.208}$$

But for choices in the future, we similarly have δ and δ^{-1} only preserving causality in one direction (using the same counterexample as for $\parr$).

$$\delta : (\mathbf{A} < \mathbf{B}) \oplus (\mathbf{A} < \mathbf{C}) \to \mathbf{A} < (\mathbf{B} \oplus \mathbf{C}) \tag{2.209}$$

$$\delta^{-1} : \mathbf{A} < (\mathbf{B} \times \mathbf{C}) \to (\mathbf{A} < \mathbf{B}) \times (\mathbf{A} < \mathbf{C}) \tag{2.210}$$

Remark 2.7.11

For the interest of those familiar with Barrett's Generalised Non-Signalling Theory [10] (a GPT which supports more non-local correlations than quantum theory), we can reconstruct it perfectly as a particular subcategory of $\mathrm{Caus}\left[\mathrm{Mat}[\mathbb{R}^{+}]\right]$. States are vectors of positive real numbers describing the probabilities of each outcome in some set of fiducial measurements (i.e. a finite set of measurements with finite outcomes that are sufficient to characterise the state, fixed for each system up front). The set of allowed states of a single system include any vector that gives a normalised probability distribution for each fiducial measurement. An (n, k)-system (n fiducial measurements, each with k outcomes) is characterised by the type $(\mathbf{I}^{\oplus \mathbf{k}})^{\times \mathbf{n}}$, since states are a vector of $n \cdot k$ positive reals and the dual space of effects is generated by $\left\{p_i \fatsemi \overset{=}{\top}_{I^{\oplus k}}\right\}_{i \leq n}$ (choosing a fiducial measurement and marginalising over the outcomes). For multi-partite systems, the permissible states require both the normalisation condition and an extra non-signalling condition, matching the characterisation of $\mathbf{A} \otimes \mathbf{B}$ as the space of non-signalling processes. In terms of transformations, the theory allows every matrix that preserves the normalisation condition on all input states, i.e. yields id_I for every normalised state provided as input and normalised effect applied to the output, matching Equation 2.50 for causal morphisms. There is no equivalent realisation of the Generalised Local Theory (also from [10]) since the space of local multi-partite states is not affine-closed and therefore will not have a corresponding object in any $\mathrm{Caus}\left[\mathcal{C}\right]$.

2.7.2 Affine Combinations of Linear Orderings are Universal

As neat as the Non-signalling Theorem appears, the de Morgan dual arguably has more significant consequences for the study of quantum causality.

Theorem 2.7.12: Sum of Orders Theorem

$$\mathbf{A} \parr \mathbf{B} = (\mathbf{A} < \mathbf{B}) \cup (\mathbf{A} > \mathbf{B})$$

Proof. This equation holds immediately from the Non-signalling Theorem by duality. □

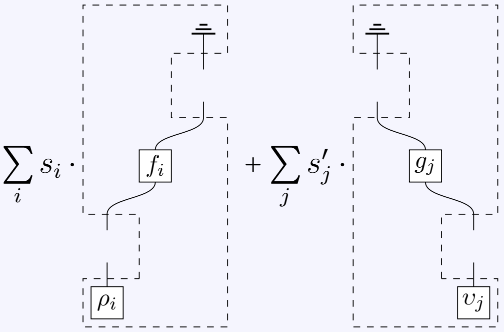
Corollary 2.7.13

Any process matrix (including those that exhibit indefinite causal structure) can be represented as an affine combination of processes with definite causal structure.

Proof. Recall Theorem 2.3.24 which characterises process matrices as the states of the following type:

$$\begin{aligned}
\left(\left(\mathbf{A^1} \multimap \mathbf{B^1}\right) \otimes \left(\mathbf{C^1} \multimap \mathbf{D^1}\right)\right)^* &\cong \left(\mathbf{A^1} \otimes \mathbf{B^{1*}}\right) \parr \left(\mathbf{C^1} \otimes \mathbf{D^{1*}}\right)\\
&\cong \left(\mathbf{A^1} \otimes \mathbf{B^{1*}}\right) < \left(\mathbf{C^1} \otimes \mathbf{D^{1*}}\right)\\
&\quad \cup \left(\mathbf{A^1} \otimes \mathbf{B^{1*}}\right) > \left(\mathbf{C^1} \otimes \mathbf{D^{1*}}\right)\\
&\cong \mathbf{A^1} < \mathbf{B^{1*}} < \mathbf{C^1} < \mathbf{D^{1*}} \qquad (2.211)\\
&\quad \cup \mathbf{C^1} < \mathbf{D^{1*}} < \mathbf{A^1} < \mathbf{B^{1*}}\\
&\cong \mathbf{A^1} \otimes \left(\mathbf{B^1} \multimap \mathbf{C^1}\right) \otimes \mathbf{D^{1*}}\\
&\quad \cup \mathbf{C^1} \otimes \left(\mathbf{D^1} \multimap \mathbf{A^1}\right) \otimes \mathbf{B^{1*}}
\end{aligned}$$

Therefore, every process matrix can be represented as an affine combination of first-order causal processes arranged in the following ways:



(2.212)

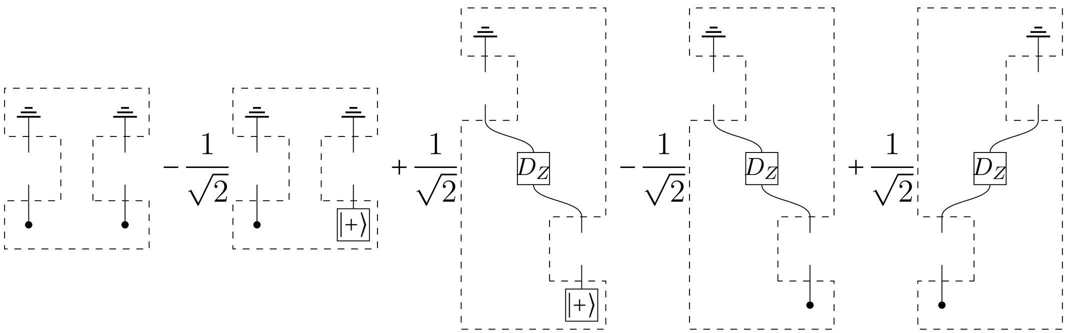
□

Example 2.7.14

The OCB process of Oreshkov, Costa, and Brukner [96] is an example process matrix on qubit systems which was shown to admit no decomposition as a convex combination of processes with definite causal structure.

$$W = \frac{1}{4}\left(\text{⏚} \otimes \text{⏚} \otimes \text{⏚} \otimes \text{⏚} + \tfrac{1}{\sqrt{2}}\left(\text{⏚} \otimes Z \otimes Z \otimes \text{⏚} + Z \otimes \text{⏚} \otimes X \otimes Z\right)\right) \qquad (2.213)$$

where Z and X are the corresponding Pauli matrices. However, it is equal to the following affine combination, where D_Z applies decoherence in the Z basis (e.g. by measuring non-destructively and discarding the classical outcome).



(2.214)

Using this result, we can view the distinction between definite and indefinite causal structures as precisely the distinction between convex and affine combinations over some basis of processes. The capability to exceed bounds on channel capacities [47] or causal inequalities [96] may all be explained by the fact that convex combinations are expected to preserve inequalities satisfied by the components of the sum but affine combinations can allow us to extrapolate outside the limits.

Since, computationally, affine combinations are just as easy to compute as convex combinations, fixing a basis of processes with definite causal structure may simplify computational tasks involved in investigating processes with indefinite causal structure.

2.8 Graph Types

Between $\otimes$ and $<$, we have a good way to interpret parallel and sequential causal structures, and combining them could feasibly give a way to describe causal structures whose partial order matches that of a series-parallel graph. Whilst such graphs are exactly those that are N-free [113] (no induced subgraph on 4 vertices is in the shape of the letter "N"), it takes infinitely many additional generators (one for each prime graph) to extend them to arbitrary graphs [3]. For the sake of generality, we will instead construct new n-ary operators, called *graph types*, in $\mathrm{Caus}[\mathcal{C}]$ to directly encode compatibility with respect to the causal structure represented by some graph.

Rather than aiming for some particular compositional properties, the approach this section takes will propose several candidate definitions for graph types that encode either signal-consistency or causal realisability, and observe the properties that emerge as consequences of these physically-motivated definitions. We set off without any expectation or requirement that, for example, nesting of graph types should relate to graph substitution, or that graph homomorphisms should induce natural transformations between the graph types, or that intersections of graph types should correspond to intersections of the edge sets of the graphs. In fact, whilst the first two of these will be satisfied, the failure of the latter reflects the fact that the signalling relations between individual parties does not determine all relations between subsets (see Section 2.8.7).

Similar to the Non-signalling Theorem and Seq Equivalence Theorem, a novel highlight of this section is the Graph Equivalence Theorem, which proves the equivalence of signal-consistency and causal realisability up to affine combinations of processes by showing that the corresponding candidate definitions for graph types coincide.

To set the common ground between these definitions, we will assume that each point v in the causal structure is assigned some object $\Gamma(v) \in \mathrm{Ob}(\mathrm{Caus}[\mathcal{C}])$ describing how a valid process should locally appear at that point, thus any definition of a graph type should embed into $\invamp_{v \in V} \Gamma(v)$. We also fix an ordering over V to give a canonical carrier object $\bigotimes_{v \in V} \mathcal{U}(\Gamma(v)) \in \mathrm{Ob}(\mathcal{C})$ (in line with Remark 2.3.28, any definition shown here is implicitly permuted to match this ordering).

Definition 2.8.1: Local interpretation

Let $G = (V, E)$ be a directed graph, and suppose V is ordered. A *local interpretation for G* is a function $\Gamma : V \to \mathrm{Ob}(\mathrm{Caus}[\mathcal{C}])$ assigning a causal type to each vertex.

2.8.1 Local Graph Types

The first direction we will look at generalises Definitions 2.3.16 and 2.6.3 based on (semi-)localisability. These show the spaces as spanned (under affine combination) by terms which factorise into local causal processes connected by wires along the edges of a graph (trivially for $\otimes$). In Definition 2.6.3 in particular, we require the intermediate system to be first-order, allowing information to travel from $\mathbf{A}$ to $\mathbf{B}$ but never in the opposite direction. This concept is one that generalises straightforwardly to factorising with wires given by the edges of a graph as in the definition of causal realisability.

Morphisms in our example categories are linear maps and can be likened to tensor nets. In this picture, wiring up any two interfaces corresponds to performing a tensor contraction. In the following definition, we use the term *contraction* to refer to the wiring up of interfaces in this way[10].

Definition 2.8.2: Graph state

Let $G = (V, E)$ be a graph with ordered vertices and a local interpretation Γ. An *edge interpretation for G* is a function $\Delta : E \to \mathrm{Ob}\left(\mathrm{FO}\left(\mathrm{Caus}\left[\mathcal{C}\right]\right)\right)$. Together, these determine a *component typing for G* $\mathrm{Comp}_G^{\Gamma;\Delta} : V \to \mathrm{Ob}\left(\mathrm{Caus}\left[\mathcal{C}\right]\right)$ by:

$$\mathrm{Comp}_G^{\Gamma;\Delta}(v) := \begin{pmatrix} \Gamma(v) \\ \parr\left(\bigparr_{(u \to v) \in E} \Delta(u \to v)^*\right) \\ \parr\left(\bigparr_{(v \to w) \in E} \Delta(v \to w)\right) \end{pmatrix} \tag{2.215}$$

The *contraction morphism for G* is the morphism

$$\epsilon_G^{\Gamma;\Delta} \in \mathcal{C}\left(\bigotimes_{v \in V} \mathcal{U}\left(\mathrm{Comp}_G^{\Gamma;\Delta}(v)\right), \bigotimes_{v \in V} \mathcal{U}\left(\Gamma(v)\right)\right) \tag{2.216}$$

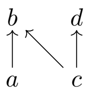
formed by the following rules:

- For each edge $(u \to v) \in E$, apply a cap $\epsilon_{\Delta(u \to v)} : \Delta(u \to v) \otimes \Delta(u \to v)^* \to \mathbf{I}$ between the $(u \to v)$ component of $\mathrm{Comp}_G^{\Gamma;\Delta}(u)$ and the $(u \to v)$ component of $\mathrm{Comp}_G^{\Gamma;\Delta}(v)$.
- For each vertex $v \in V$, apply an identity on the $\Gamma(v)$ within $\mathrm{Comp}_G^{\Gamma;\Delta}(v)$.

A *graph state over G* is a morphism $g : I \to \bigotimes_{v \in V} \mathcal{U}(\Gamma(v))$ which factorises as $g = \left(\bigotimes_{v \in V} g_v\right) ⨟ \epsilon_G^{\Gamma}$ with the component at each vertex matching the component typing $\forall v \in V. g_v : \mathrm{Comp}_G^{\Gamma;\Delta}(v)$.

Example 2.8.3

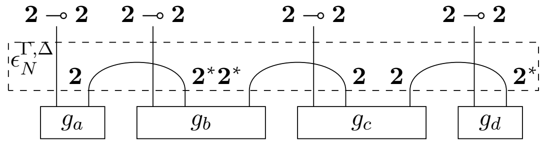
Let N be the following graph.



(2.217)

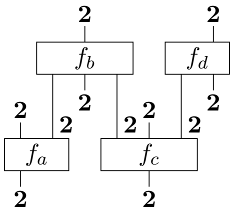
For simplicity, consider constant interpretations Γ, Δ that map vertices to bit channels $\mathbf{2} \multimap \mathbf{2}$ and each edge to a single bit $\mathbf{2}$. The graph states over N are then morphisms of the following form where each component g_v is a causal state of $\mathrm{Comp}_N^{\Gamma;\Delta}(v)$:

[10] “Contraction” is also a common term in logic, referring to the duplication/merging of identical terms - there is no intended relation between contraction in this thesis and logical contraction


$\mathbf{2} \multimap \mathbf{2}$
$\mathbf{2} \multimap \mathbf{2}$
$\mathbf{2} \multimap \mathbf{2}$
$\mathbf{2} \multimap \mathbf{2}$
$\epsilon_N^{\Gamma,\Delta}$
$\mathbf{2}$
$\mathbf{2}^*$
$\mathbf{2}^*$
$\mathbf{2}$
$\mathbf{2}$
$\mathbf{2}^*$
g_a
g_b
g_c
g_d

(2.218)

Unpacking each g_v into the corresponding channel f_v, we see more clearly that the graph states encode networks that factorise according to the graph N.



(2.219)

Such graph states resemble Bayesian networks, with some slight modifications. Understanding that many physical theories do not admit broadcasting maps for sharing the output of a given point with its descendants, each local function has additional outputs to pass on, accepting that this may not share information identically.

Whilst we have defined graph states for arbitrary graphs, whenever the graph would admit a cycle we can (noisily) encode a logical paradox which breaks causality - more specifically, there is no longer a uniform discarding effect which is causal for all graph states. This turns the anti-symmetry of the partial order for a causal structure into an emergent property by asking for flatness of the causally realisable processes.

Lemma 2.8.4: Acyclicity Lemma

The set of graph states over G with a fixed edge interpretation Δ is flat iff in every cycle of G there is some edge $(u \to v)$ for which $\Delta(u \to v) \cong \mathbf{I}$.

Proof. $\Longleftarrow$: Suppose that every cycle of G contains an $\mathbf{I}$ edge. Let G' be the graph obtained by removing each such edge from G, leaving a DAG. Since each removed edge was interpreted as $\mathbf{I}$, graph states over G are always graph states over G'. We can then show that graph states over DAGs are flat by induction on the number of vertices of the graph.

If G is acyclic, there must be some maximal vertex v with no outgoing edges. Pick the effect $\top_{\Gamma(v)} : \Gamma(v) \to \mathbf{I}$ and apply it to a graph state g at g_v.

$$g_v \fatsemi \left(\top_{\Gamma(v)} \otimes \mathrm{id}\right) : \mathbf{I} \to \bigparr_{(u \to v) \in E} \Delta(u \to v)^* \tag{2.220}$$

Since Δ can only pick first-order objects, the only causal morphism of this type is $\bigotimes_{(u \to v) \in E} \text{⫰}_{\Delta(u \to v)^*}$. Each $\text{⫰}_{\Delta(u \to v)^*}$ can be absorbed into g_u to show that $g \fatsemi \left(\mathrm{id} \otimes \text{⫯}_{\Gamma(v)} \otimes \mathrm{id}\right)$ gives a graph state over $G \setminus \{v\}$. By induction, the set of graph states over $G \setminus \{v\}$ is flat with causal effect $\bigotimes_{u \in V \setminus \{v\}} \text{⫯}_{\Gamma(u)}$. We now have $g \fatsemi \left(\bigotimes_{u \in V} \text{⫯}_{\Gamma(u)}\right) = \mathrm{id}_I$ for every graph state g, so $\bigotimes_{u \in V} \text{⫯}_{\Gamma(u)}$ is in the dual set.

$\bigotimes_{v \in V} \text{⫰}_{\Gamma(v)}$ is trivially a graph state by supplementing it with a pair of local causal state and effect for each edge of the graph.

$\Longrightarrow$: Suppose there exists a (wlog minimal) cycle σ of G for which Δ assigns an object with at least two distinct causal states to each edge.

For each edge $(u \to v) \in \sigma$, pick two distinct states $\rho_{uv} \neq \text{⫰}_{uv} \in c_{\Delta(u \to v)}$. For each segment $(u \to v \to w) \in \sigma$, let $f_v : \mathrm{Caus}[\mathcal{C}]\left(\Delta(u \to v), \Delta(v \to w)\right)$ be a binary encoding for the chosen states of $\Delta(u \to v)$ and $\Delta(v \to w)$ from Lemma 2.6.24.

We will use these binary encodings to construct two graph states that are equal up to a non-unit scalar, and so by linearity there is no effect that can send them both to id_I.

Firstly, we pick a graph state g where every $g_v : \mathbf{I} \to \mathrm{Comp}_G^{\Gamma;\Delta}(v)$ is separable:

$$\begin{aligned} g_v &= \rho_v \otimes \left(\bigotimes_{(u \to v) \in E} \pi_{uv}\right) \otimes \left(\bigotimes_{(v \to w) \in E} \rho_{vw}\right) \\ &\rho_v \in c_{\Gamma(v)} \\ &\forall (u \to v) \in E.\ \pi_{uv} \in c^*_{\Delta(u \to v)} \\ &\forall (v \to w) \in E.\ \rho_{vw} \in c_{\Delta(v \to w)} \end{aligned} \tag{2.221}$$

so $g = \bigotimes_{v \in V} \rho_v$ (the state-effect pairs for each edge merge to unit scalars).

We build another graph state g' by taking g and for every vertex v in the cycle σ ($(u \to v \to w) \in \sigma$), we obtain g'_v by replacing $\pi_{uv} \otimes \rho_{vw}$ with $\eta \fatsemi (\mathrm{id} \otimes f_v)$ (where η is the appropriate cup state). This still gives $g'_v : \mathbf{I} \to \mathrm{Comp}_G^{\Gamma;\Delta}(v)$, so we have a valid graph state. Removing the state-effect pairs for edges not in σ, we find that g' is g with an additional circular morphism following σ (we borrow the $\prod$ notation from linear algebra to refer to an indexed sequential product of linear maps).

$$g' = \left(\eta \fatsemi \left(\mathrm{id} \otimes \prod_{v \in \sigma} f_v\right) \fatsemi \epsilon\right) \cdot g \tag{2.222}$$

We expand one $f_{\hat{v}}$ in this scalar (with $(\hat{u} \to \hat{v} \to \hat{w}) \in \sigma$), and use the fact that each f_v is causal and noisily preserves the dual effect (see the proof of

Lemma 2.6.24) to simplify. Here, we use similar notations as in Lemma 2.6.24 for the chosen states, dual effects, and relevant scalars.

$$
\begin{aligned}
&\left[\eta \fatsemi \left(\mathrm{id} \otimes \prod_{v\in\sigma} f_v\right) \fatsemi \epsilon\right] \\
&\sim \left[\left(\lambda_{\hat{v}}\theta^{-1}_{\hat{u}\hat{v}}\mu^{-1}_{\hat{v}\hat{w}} + \mathrm{id}_I\right)^{-1}\right] \cdot [\eta] \fatsemi \\
&\left(\mathrm{id} \otimes \left(\left(\begin{array}{l} \quad\left[\lambda_{\hat{v}}\theta^{-1}_{\hat{u}\hat{v}}\mu^{-1}_{\hat{v}\hat{w}} \cdot \text{⫯}_{\hat{u}\hat{v}}\right] \fatsemi [\rho_2^{\hat{v}\hat{w}}] \\ +e_1^{\hat{u}\hat{v}} \fatsemi [\rho_1^{\hat{v}\hat{w}}] \\ +[\text{⫯}_{\hat{u}\hat{v}}] \fatsemi [\rho_2^{\hat{v}\hat{w}}] \\ -e_1^{\hat{u}\hat{v}} \fatsemi [\rho_2^{\hat{v}\hat{w}}] \end{array}\right) \fatsemi \left[\prod_{\hat{v}\neq v\in\sigma} f_v\right]\right)\right) \fatsemi [\epsilon] \\
&\sim [\rho_2^{\hat{v}\hat{w}}] \fatsemi \left[\prod_{\hat{v}\neq v\in\sigma} f_v\right] \fatsemi [\text{⫯}_{\hat{u}\hat{v}}] \qquad\qquad (2.223)\\
&\quad + \left[\left(\lambda_{\hat{v}}\theta^{-1}_{\hat{u}\hat{v}}\mu^{-1}_{\hat{v}\hat{w}} + \mathrm{id}_I\right)^{-1}\right] \cdot \left([\rho_1^{\hat{v}\hat{w}}] - [\rho_2^{\hat{v}\hat{w}}]\right) \fatsemi \left[\prod_{\hat{v}\neq v\in\sigma} f_v\right] \fatsemi e_1^{\hat{u}\hat{v}} \\
&\sim [\rho_2^{\hat{v}\hat{w}}] \fatsemi [\text{⫯}_{\hat{v}\hat{w}}] \\
&\quad + \left[\prod_{v\in\sigma}\left(\lambda_v\theta^{-1}_{uv}\mu^{-1}_{vw} + \mathrm{id}_I\right)^{-1}\right] \cdot \left([\rho_1^{\hat{v}\hat{w}}] - [\rho_2^{\hat{v}\hat{w}}]\right) \fatsemi e_1^{\hat{v}\hat{w}} \\
&\sim \left[\mathrm{id}_I + \prod_{v\in\sigma}\left(\lambda_v\theta^{-1}_{uv}\mu^{-1}_{vw} + \mathrm{id}_I\right)^{-1}\right]
\end{aligned}
$$

The big product must be non-zero since it is invertible, so we can conclude that g and g' are equal up to a non-unit scalar. □

Corollary 2.8.5

The set of graph states over G (with any edge interpretation) is flat iff G is acyclic.

Proof. If G contains any cycle, we can choose an edge interpretation that assigns a non-trivial object to every edge in the cycle and create counterexamples to flatness with Lemma Acyclicity Lemma. Similarly, if G is acyclic, Lemma Acyclicity Lemma shows that all graph states are causal with respect to the uniform effect (up to a constant which does not depend on the choice of edge interpretation). □

This brings us to our first candidate definition of a graph type, where the states are affine combinations of graph states with arbitrary edge interpretations. We will also consider artificially restricting all intermediate edges to the object **2**, the simplest object that can still carry information, as we will eventually see that this is sufficient to recover arbitrary systems under affine combination.

Definition 2.8.6: Local graph type

Given a DAG $G = (V, E)$ with ordered vertices and a local interpretation Γ, the *local graph type over G* is

$$\mathbf{LoGr}_G^{\Gamma} \coloneqq \left(\bigotimes_{v \in V} \mathcal{U}\left(\Gamma\left(v\right)\right), \left\{ g \;\middle|\; \begin{array}{l} g \text{ is a graph state} \\ \text{over } G \text{ for some edge} \\ \text{interpretation } \Delta \end{array} \right\}^{**} \right) \tag{2.224}$$

and the **2**-*local graph type over G* is

$$\mathbf{LoGr2}_G^{\Gamma} \coloneqq \left(\bigotimes_{v \in V} \mathcal{U}\left(\Gamma\left(v\right)\right), \left\{ g \;\middle|\; \begin{array}{l} g \text{ is a graph state} \\ \text{over } G \text{ using} \\ \Delta :: (u \to v) \mapsto \mathbf{2} \end{array} \right\}^{**} \right) \tag{2.225}$$

2.8.2 Signalling Graph Types

Next, we turn to definitions based on signal-consistency with respect to a causal structure. This expresses compatibility with the causal structure as a combination of a number of elementary non-signalling conditions, each of which can be expressed using $<$ and $\parr$ as discussed in Section 2.6.7. The concept of a down-closed subset of the causal structure corresponds to a partition of the entire set into past and future, from which we can then impose non-signalling from the future. Because this is a complete partition of the entire causal structure, we can simplify the form of a non-signalling condition from Equation 2.173.

We provide two definitions of this kind: one which imposes no constraints on the kinds of marginal states observed in the past, and one which recursively asks the past marginals to continue respecting the signalling conditions of the graph.

Definition 2.8.7: Signalling graph type

Let $G = (V, E)$ be a DAG with ordered vertices and fix a local interpretation Γ. A subset of vertices $U \subseteq V$ is *down-closed* if $\forall \left(u \to v\right) \in E.\ v \in U \implies u \in U$. The *signalling graph type over G* is

$$\mathbf{SiGr}_G^{\Gamma} \coloneqq \bigcap_{U \subset V \text{ down-closed}} \bigparr_{u \in U} \Gamma\left(u\right) < \bigparr_{v \in V \setminus U} \Gamma\left(v\right) \tag{2.226}$$

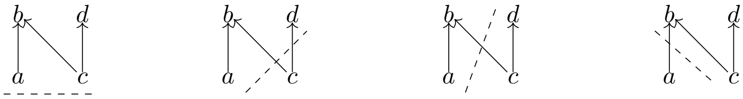
and the *recursive signalling graph type over G* is

$$\mathbf{RSiGr}_G^{\Gamma} \coloneqq \bigcap_{U \subset V \text{ down-closed}} \mathbf{RSiGr}_{G[U]}^{\Gamma[U]} < \bigparr_{v \in V \setminus U} \Gamma\left(v\right) \tag{2.227}$$

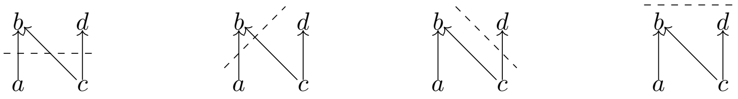
where $G[U]$ is the induced subgraph over U and $\Gamma[U]$ similarly restricts the domain of Γ to U.

Example 2.8.8

We claim that the graph states over the N graph from Example 2.8.3 satisfy the signalling graph type conditions for the same graph. There are 8 down-closed subsets to consider, specified by the following cuts through the graph:

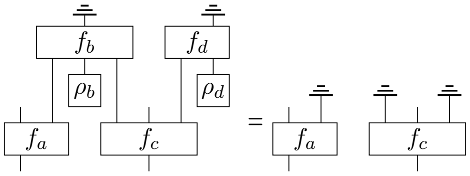


(2.228)



(2.229)

Let's look at the bottom-left one, describing $\{a, c\}$ before $\{b, d\}$ - that is, for any choice of local effects on $\{b, d\}$, there is a unique marginal over $\{a, c\}$. The local interpretation of each vertex was $\mathbf{2} \multimap \mathbf{2}$, so any choice of effect will always decompose into a pair of a binary input state and discarding the binary output. In the picture of Equation 2.219, the component at b is $f_b : (\mathbf{2} \multimap \mathbf{2}) \parr \mathbf{2}^* \parr \mathbf{2}^*$, so after applying the input and discarding the output we are left with something of type $\mathbf{2}^* \parr \mathbf{2}^*$ of which the only possibility is the discard map. The same applies to f_d to give the following:



(2.230)

This marginal is independent of our choice of effects on $\{b, d\}$. The $\mathbf{SiGr}_G^\Gamma$ definition requires this marginal to be a state of $(\mathbf{2} \multimap \mathbf{2}) \parr (\mathbf{2} \multimap \mathbf{2})$, whereas $\mathbf{RSiGr}_G^\Gamma$ requires it to recursively satisfy more non-signalling conditions - in this case, that there is no signalling in either direction between a and c, i.e. the marginal is a state of $(\mathbf{2} \multimap \mathbf{2}) \otimes (\mathbf{2} \multimap \mathbf{2})$. The latter is immediate from the fact that it is separable, and the former follows by the natural inclusion of $\otimes$ into $\parr$ (Equation 2.58).

If we instead use the semi-localisability definition of $<$ (Definition 2.6.3), compatibility with a given cut through the graph is immediate by applying the same cut to the actual graph state in Equation 2.219 and observing that it factorises via the first order system $\mathbf{2} \otimes \mathbf{2}$.

Where graph states resembled Bayesian networks, this holds more resemblance to the causal Markov condition: a probability distribution is Markov relative to a DAG iff every variable is independent of its non-descendants conditional on its parents. Since $\mathcal{C}$ might not support the ability to freely construct conditional distributions, we instead phrase it in terms of signalling (whether the choice of local effects, which may cover marginalisation and some classes of interventions, has any observable effect on the remaining state), and we look at all ancestors instead of just the parents since graph states can permit transitive signalling.

2.8.3 Ordered Graph Types

Recall Theorem 2.3.29 which characterised compatibility of first-order processes with a given causal order via compatibility with every totalisation. The final graph type definition we will consider is a higher-order generalisation of this, where the totalisations of the causal order correspond to topological sorts over the vertices of the graph.

Definition 2.8.9: Ordered graph type

Given a DAG G with ordered vertices and a local interpretation Γ, the *ordered graph type over G* is

$$\mathbf{OrGr}_G^\Gamma \coloneqq \bigcap_{[v_1,\dots,v_n] \in \mathrm{sort}(G)} \Gamma(v_1) < \dots < \Gamma(v_n) \tag{2.231}$$

where $\mathrm{sort}(G)$ is the set of all topological sorts of the graph G.

Example 2.8.10

Returning to our example of the N graph, we have to consider the following topological sorts of the graph:

$$\begin{array}{ccc} (a,c,b,d) & (a,c,d,b) & (c,d,a,b) \\ (c,a,b,d) & (c,a,d,b) & \end{array} \tag{2.232}$$

We can follow a similar argument to Example 2.8.8 to show that the graph states on N are compatible with each of these orderings in terms of non-signalling conditions or factorising via some first-order systems.

Making the same alterations to discuss non-signalling over conditional independence, this has a similar resemblance to the ordered Markov condition: conditional on its parents, each variable of a probability distribution is independent of all its predecessors in a topological sort of the graph.

Relating to the category theory literature, the ordered graph type extends $\otimes$ and $<$ from series-parallel orders to arbitrary partial orders over the vertices using the pullback construction of Shapiro & Spivak [104], elevating the duoidal structure to a dependence structure ($\mathrm{Caus}\,[\mathcal{C}]$ is a pseudo-algebra of the categorical symmetric operad of finite posets).

2.8.4 Equivalence of Definitions

The following Theorem states the equivalence of all candidate graph type definitions presented here. Notably, $\mathbf{SiGr}^{\Gamma}_{G} = \mathbf{OrGr}^{\Gamma}_{G}$ lifts Theorem 2.3.29 from first-order channels to arbitrary local systems, and $\mathbf{LoGr}^{\Gamma}_{G} = \mathbf{OrGr}^{\Gamma}_{G}$ generalises the Non-signalling Theorem and Seq Equivalence Theorem from the empty graph for $\otimes$ or linear graphs for $<$ to arbitrary DAGs.

Theorem 2.8.11: Graph Equivalence Theorem

$$\mathbf{LoGr}^{\Gamma}_{G} = \mathbf{LoGr2}^{\Gamma}_{G} = \mathbf{RSiGr}^{\Gamma}_{G} = \mathbf{SiGr}^{\Gamma}_{G} = \mathbf{OrGr}^{\Gamma}_{G}$$

Proof. We will show a cycle of inclusions between the state sets of these types.

Inclusion 1 $c_{\mathbf{LoGr2}^{\Gamma}_{G}} \subseteq c_{\mathbf{LoGr}^{\Gamma}_{G}}$: graph states using the constant edge interpretation are included in the set of all graph states.

Inclusion 2 $c_{\mathbf{LoGr}^{\Gamma}_{G}} \subseteq c_{\mathbf{OrGr}^{\Gamma}_{G}}$: for any topological order O, we proceed in the manner of the $\Longleftarrow$ direction of the Acyclicity Lemma: applying any causal effect at the maximal vertex v gives a constant marginal over $V \setminus \{v\}$ as the edges into it are first-order. We continue down the order of O inductively, showing one-way signalling at each step.

Inclusion 3 $c_{\mathbf{OrGr}^{\Gamma}_{G}} \subseteq c_{\mathbf{SiGr}^{\Gamma}_{G}}$: pick some state $h \in c_{\mathbf{OrGr}^{\Gamma}_{G}}$ and some down-closed set $U \subset V$. Since U is down-closed, there must exist some topological sort O which lists all vertices in U and then all vertices in $V \setminus U$. Therefore, $h : \left(<^{O}_{v \in U} \Gamma(v)\right) < \left(<^{O}_{v \in V \setminus U} \Gamma(v)\right)$ which is a subtype of $(\parr_{v \in U} \Gamma(v)) < (\parr_{v \in V \setminus U} \Gamma(v))$.

Inclusion 4 $c_{\mathbf{SiGr}^{\Gamma}_{G}} \subseteq c_{\mathbf{RSiGr}^{\Gamma}_{G}}$: by induction on the number of vertices in G. The cases for zero and one vertex are trivial, so consider some $h \in c_{\mathbf{SiGr}^{\Gamma}_{G}}$ ($|V| > 1$) and an arbitrary down-closed $U \subset V$. It is sufficient to show that the constant marginal $h_U : \parr_{v \in U} \Gamma(v)$ is actually in $c_{\mathbf{SiGr}^{\Gamma[U]}_{G[U]}}$ (since $|U| < |V|$, this is a subset of $c_{\mathbf{RSiGr}^{\Gamma[U]}_{G[U]}}$ by the induction hypothesis).

Pick an arbitrary $U' \subset U$ which is down-closed wrt $G[U]$ (and hence down-closed wrt G). Using the signalling graph type property of h, it has a constant marginal $h_{U'} : \bigparr_{v \in U'} \Gamma(v)$ for any choice of local effects over $V \setminus U'$. Applying any choice of effects $\{\pi_v : \Gamma(v) \to \mathbf{I}\}_{v \in U \setminus U'}$ on h_U then gives $h_{U'}$ as a constant marginal, showing that $h_U : (\bigparr_{v \in U'} \Gamma(v)) < (\bigparr_{v \in U \setminus U'} \Gamma(v))$:

$$\begin{aligned} & h_U \fatsemi \bigotimes_{v \in U} \begin{cases} \mathrm{id}_{\Gamma(v)} & v \in U' \\ \pi_v & v \in U \setminus U' \end{cases} \\ = {} & h \fatsemi \bigotimes_{v \in V} \begin{cases} \mathrm{id}_{\Gamma(v)} & v \in U' \\ \pi_v & v \in U \setminus U' \\ \text{⫯}_{\Gamma(v)} & v \in V \setminus U \end{cases} \\ = {} & h_{U'} \end{aligned} \tag{2.233}$$

Taking the intersections of these types for each down-closed $U' \subset U$ gives $h_U : \mathbf{SiGr}_{G[U]}^{\Gamma[U]}$.

Inclusion 5 $c_{\mathbf{RSiGr}_G^\Gamma} \subseteq c_{\mathbf{LoGr2}_G^\Gamma}$: This is by far the most involved direction. It is, again, by induction on the number of vertices where the cases for zero and one are trivial, so we focus on the inductive case. Firstly, we will fix a bunch of notation. For each $v \in V$:

- Let $\left\{\rho_{i_v}^v\right\}_{i_v} \subseteq \mathcal{C}(I, \mathcal{U}(\Gamma(v)))$ be a preferred basis for $\Gamma(v)$, with dual effects $\left\{e_{i_v}^v\right\}_{i_v} \subseteq \mathrm{Sub}(\mathcal{C})(\mathcal{U}(\Gamma(v)), I)$.

- Let λ_v be an invertible scalar and $\left\{\sigma_{i_v}^v\right\}_{i_v} \subseteq \mathcal{C}(I, \mathcal{U}(\Gamma(v)))$ s.t. $\forall i_v.\ \rho_{i_v}^v + \sigma_{i_v}^v = \lambda_v \cdot \text{⏚}$ (e.g. apply APC5a for each $\rho_{i_v}^v$ to get λ_{v,i_v} and take the maximum with respect to the preorder of APC4).

- Let $\left\{t_{i_v}^v : \mathbf{2}^{\otimes|\mathrm{in}(v)|} \to \Gamma(v) \parr \mathbf{2}^{\parr|\mathrm{out}(v)|}\right\}_{i_v}$ be binary tests for $\left\{\rho_{i_v}^v\right\}_{i_v}$, conditioned on all predecessors in the graph and broadcast to all successors.

$$\left(\bigotimes_{u \in \mathrm{in}(v)} \iota_{x_u}\right) \fatsemi t_{i_v}^v := \begin{cases} \mu_v \lambda_v^{-1} \cdot \begin{pmatrix} \rho_{i_v}^v \otimes \bigotimes_{w \in \mathrm{out}(v)} \mathtt{t} \\ +\sigma_{i_v}^v \otimes \bigotimes_{w \in \mathrm{out}(v)} \mathtt{f} \end{pmatrix} & \forall u \in \mathrm{in}(v).\ \iota_{x_u} = \mathtt{t} \\ \text{⫰}_{\Gamma(v)} \otimes \bigotimes_{w \in \mathrm{out}(v)} \mathtt{f} & \text{otherwise} \end{cases} \tag{2.234}$$

- Let $\left\{T_{i_v}^v\right\}_{i_v}$ be conditional preparations.

$$\left(\bigotimes_{u \in \mathrm{in}(v)} \iota_{x_u}\right) \fatsemi T_{i_v}^v := \begin{cases} \rho_{i_v}^v & \forall u \in \mathrm{in}(v).\ \iota_{x_u} = \mathtt{t} \\ \text{⫰}_{\Gamma(v)} & \text{otherwise} \end{cases} \tag{2.235}$$

  For those $i_v \in \mathfrak{B}_{\Gamma(v)}$ $(\rho_{i_v}^v \in c_{\Gamma(v)})$, $T_{i_v}^v : \mathbf{2}^{\otimes|\mathrm{in}(v)|} \to \Gamma(v)$ is causal.

Let $\max G = \{v \in V \mid \nexists (v \to w) \in E\}$ be the set of maximal vertices of the graph (since G is acyclic, this must be non-empty). Let $g_{\vec{i}} \in c_{\mathbf{LoGr2}_G^\Gamma}$ be the graph state over G wrt the constant-$\mathbf{2}$ edge interpretation $\Delta_{\mathbf{2}}$ whose components are $\left\{t_{i_v}^v \mid v \notin \max G\right\} \cup \left\{T_{i_v}^v \mid v \in \max G\right\}$; note that this is only a valid graph state when $i_v \in \mathfrak{B}_{\Gamma(v)}$, otherwise $T_{i_v}^v$ is not causal for the appropriate component type.

Now, we take an arbitrary state $h : \mathbf{RSiGr}_G^\Gamma$.

For any $v \in \max V$, we claim the following:

$$\forall i_v \in \overline{\mathfrak{B}}_{\Gamma(v)}.\ [h] \fatsemi \left(\bigotimes_{u \in V} \begin{cases} e_{i_v}^v & u = v \\ \mathrm{id} & u \neq v \end{cases} \right) \sim 0 \tag{2.236}$$

$V \setminus \{v\}$ is down-closed, so the recursive signalling graph type tells us that $h : \mathbf{RSiGr}_{G[V \setminus \{v\}]}^{\Gamma[V \setminus \{v\}]} < \Gamma(v)$. Equation 2.236 then follows from Lemma 2.6.8, as expanding h in terms of the basis for $\Gamma(v)$ will only use the causal basis states.

Now, consider the following expression:

$$\sum_{\vec{i}} \left([h] \fatsemi \bigotimes_{v \in V} e_{i_v}^v \right) \cdot \left[g_{\vec{i}}\right] \tag{2.237}$$

This is a linear combination of graph states wrt $\Delta_{\mathbf{2}}$, since any choice of $\vec{i}$ for which $g_{\vec{i}}$ is not valid ($\exists v \in \max V. \rho_{i_v}^v \notin c_{\Gamma(v)}$) would give a zero coefficient. It is also affine by Lemma 2.6.8.

The rest of the proof will show that expanding the definitions of each $t_{i_v}^v$ and $T_{i_v}^v$ in Equation 2.237 equates $g_{\vec{i}}$ to another affine combination, where precisely one term in the sum is h itself (with a non-zero coefficient) and the rest are other graph states wrt $\Delta_{\mathbf{2}}$. From this, we could solve the equation for h to represent it as an affine combination of graph states wrt $\Delta_{\mathbf{2}}$, showing that it lives in $c_{\mathbf{LoGr2}_G^\Gamma}$.

For each index i_v in the sum of Equation 2.237, only $e_{i_v}^v$ and either $t_{i_v}^v$ or $T_{i_v}^v$ depend on it. Summing over that index locally gives the following decompositions:

$$\begin{aligned} &\sum_{i_v} e_{i_v}^v \fatsemi \left[\left(\bigotimes_{u \in \mathrm{in}(v)} \iota_{x_u} \right) \fatsemi t_{i_v}^v \right] \\ \sim &\begin{cases} \left[\mu_v \lambda_v^{-1}\right] \cdot \left(\begin{array}{r} \sum_{i_v} e_{i_v}^v \fatsemi \left[\rho_{i_v}^v \otimes \bigotimes_{w \in \mathrm{out}(v)} \mathtt{t}\right] \\ + \sum_{i_v} e_{i_v}^v \fatsemi \left[\sigma_{i_v}^v \otimes \bigotimes_{w \in \mathrm{out}(v)} \mathtt{f}\right] \end{array} \right) & \forall u \in \mathrm{in}(v) \,.\ \iota_{x_u} = \mathtt{t} \\ \sum_{i_v} e_{i_v}^v \fatsemi \left[{\downarrow}\!\bullet_{\Gamma(v)} \otimes \bigotimes_{w \in \mathrm{out}(v)} \mathtt{f} \right] & \text{otherwise} \end{cases} \end{aligned}$$

$$
\begin{aligned}
&\sim \begin{cases} \begin{pmatrix} \left[\mu_v \lambda_v^{-1}\right] \cdot \left(\sum_{i_v} e^v_{i_v} \fatsemi \left[\rho^v_{i_v}\right]\right) \otimes \left[\bigotimes_{w \in \mathrm{out}(v)} \mathtt{t}\right] \\ -\left[\mu_v \lambda_v^{-1}\right] \cdot \left(\sum_{i_v} e^v_{i_v} \fatsemi \left[\rho^v_{i_v}\right]\right) \otimes \left[\bigotimes_{w \in \mathrm{out}(v)} \mathtt{f}\right] \\ +\left(\sum_{i_v} e^v_{i_v} \fatsemi \left[\underset{\bullet}{|}_{\Gamma(v)}\right]\right) \otimes \left[\bigotimes_{w \in \mathrm{out}(v)} \mathtt{f}\right] \end{pmatrix} & \forall u \in \mathrm{in}\,(v)\,.\ \iota_{x_u} = \mathtt{t} \\ \left(\sum_{i_v} e^v_{i_v} \fatsemi \left[\underset{\bullet}{|}_{\Gamma(v)}\right]\right) \otimes \left[\bigotimes_{w \in \mathrm{out}(v)} \mathtt{f}\right] & \text{otherwise} \end{cases} \\
&\sim \begin{cases} \begin{pmatrix} \left[\mu_v \lambda_v^{-1}\right] \cdot [\mathrm{id}] \otimes \left[\bigotimes_{w \in \mathrm{out}(v)} \mathtt{t}\right] \\ -\left[\mu_v \lambda_v^{-1}\right] \cdot [\mathrm{id}] \otimes \left[\bigotimes_{w \in \mathrm{out}(v)} \mathtt{f}\right] \\ +\left[\overset{\bullet}{|}_{\Gamma(v)} \fatsemi \underset{\bullet}{|}_{\Gamma(v)}\right] \otimes \left[\bigotimes_{w \in \mathrm{out}(v)} \mathtt{f}\right] \end{pmatrix} & \forall u \in \mathrm{in}\,(v)\,.\ \iota_{x_u} = \mathtt{t} \\ \left[\overset{\bullet}{|}_{\Gamma(v)} \fatsemi \underset{\bullet}{|}_{\Gamma(v)}\right] \otimes \left[\bigotimes_{w \in \mathrm{out}(v)} \mathtt{f}\right] & \text{otherwise} \end{cases}
\end{aligned}
\tag{2.238}
$$

$$
\begin{aligned}
&\sum_{i_v} e^v_{i_v} \fatsemi \left[\left(\bigotimes_{u \in \mathrm{in}(v)} \iota_{x_u}\right) \fatsemi T^v_{i_v}\right] \\
&\sim \begin{cases} \sum_{i_v} e^v_{i_v} \fatsemi \left[\rho^v_{i_v}\right] & \forall u \in \mathrm{in}\,(v)\,.\ \iota_{x_u} = \mathtt{t} \\ \sum_{i_v} e^v_{i_v} \fatsemi \left[\underset{\bullet}{|}_{\Gamma(v)}\right] & \text{otherwise} \end{cases} \\
&\sim \begin{cases} \mathrm{id} & \forall u \in \mathrm{in}\,(v)\,.\ \iota_{x_u} = \mathtt{t} \\ \left[\overset{\bullet}{|}_{\Gamma(v)} \fatsemi \underset{\bullet}{|}_{\Gamma(v)}\right] & \text{otherwise} \end{cases}
\end{aligned}
\tag{2.239}
$$

In each case from these, we get an affine combination of a local channel applied at the vertex (either the identity or completely noisy channel) and some classical outcome passed to the next vertices. So in the complete expansion of Equation 2.237, we get an affine combination of terms, each of which is h with the completely noisy channel applied to some subset of vertices.

$$
\sum_{\vec{i}} \left([h] \fatsemi \bigotimes_{v \in V} e^v_{i_v}\right) \cdot \left[g_{\vec{i}}\right] \sim \sum_{U \subseteq V} s_U \cdot \left[h \fatsemi \bigotimes_{v \in V} \begin{cases} \mathrm{id} & v \in U \\ \overset{\bullet}{|}_{\Gamma(v)} \fatsemi \underset{\bullet}{|}_{\Gamma(v)} & v \notin U \end{cases}\right]
\tag{2.240}
$$

Not every such subset is generated, so some coefficients s_U may be zero.

The $U = V$ term reduces completely to h with coefficient $s_V = \prod_{v \notin \max V} \mu_v \lambda_v^{-1}$ which is invertible and non-zero, so we really can solve this equation for h.

All that remains is to show that the other terms in this expansion are in $c_{\mathbf{LoGr2}^{\Gamma}_{G}}$. Consider the term for some $U \subset V$ with a non-zero coefficient.

U must be down-closed; for any $(u \to v) \in E$, if $v \in U$ then Equations 2.238 and 2.239 tell us v must have received $\mathtt{t}$ from each of its predecessors, and the only term from the expansion at u which yields $\mathtt{t}$ applies id at u, i.e. $u \in U$.

The U case of the recursive signalling graph type of h gives the seq type $h : \mathbf{RSiGr}^{\Gamma[U]}_{G[U]} < \parr_{v \in V \setminus U} \Gamma\,(v)$, so after applying $\overset{\bullet}{|}_{\Gamma(v)}$ on each $v \in V \setminus U$, we are left with the constant marginal $h_U : \mathbf{RSiGr}^{\Gamma[U]}_{G[U]}$. Applying the induction hypothesis gives $h_U : \mathbf{LoGr2}^{\Gamma[U]}_{G[U]}$ and hence can be represented as an affine combination of graph states over $G[U]$ wrt $\Delta_{\mathbf{2}}$. Adding the local states $\downarrow_{\Gamma(v)}$ at each $v \in V \setminus U$ gives an affine combination of graph states over G with $\Delta_{\mathbf{2}}$. □

This equivalence offers us great flexibility for proving results about graph types as we can freely switch between different characterising properties. Many of the properties in the remainder of this section have multiple proofs using the different kinds of graph types which may be interesting and provide insight in their own right. Now the equivalence is established, we may use $\mathbf{Gr}^{\Gamma}_{G}$ to refer to graph types in a canonical way, choosing between the alternative definitions as we see fit. We can observe that the following special cases hold by the definitions:

$$\mathbf{Gr}^{\varnothing}_{(\varnothing,\varnothing)} = \mathbf{I} \tag{2.241}$$

$$\mathbf{Gr}^{a\mapsto\mathbf{A}}_{(\{a\},\varnothing)} = \mathbf{A} \tag{2.242}$$

$$\mathbf{Gr}^{a\mapsto\mathbf{A},b\mapsto\mathbf{B}}_{(\{a,b\},\varnothing)} = \mathbf{A} \otimes \mathbf{B} \tag{2.243}$$

$$\mathbf{Gr}^{a\mapsto\mathbf{A},b\mapsto\mathbf{B}}_{(\{a,b\},\{a\to b\})} = \mathbf{A} < \mathbf{B} \tag{2.244}$$

Looking at other specific examples of graph types beyond $\mathbf{A} \otimes \mathbf{B}$ and $\mathbf{A} < \mathbf{B}$, we find an exact correspondence between any object generated by $\{\otimes, <\}$ and graph types of series-parallel graphs. This is important not only for meeting the expectations regarding compatibility with series-parallel causal structures, but it acts as the first step on a theory of compositionality of causal structures which we refine in Section 2.8.6.

Corollary 2.8.12

Given graph types over non-empty disjoint vertex sets V and V',

$$\mathbf{Gr}^{\Gamma}_{(V,E)} \otimes \mathbf{Gr}^{\Gamma'}_{(V',E')} = \mathbf{Gr}^{\Gamma,\Gamma'}_{(V\cup V',E\cup E')} \tag{2.245}$$

$$\mathbf{Gr}^{\Gamma}_{(V,E)} < \mathbf{Gr}^{\Gamma'}_{(V',E')} = \mathbf{Gr}^{\Gamma,\Gamma'}_{(V\cup V',E\cup E'\cup\{v\to v' \mid v\in V,v'\in V'\})} \tag{2.246}$$

Proof. For $\otimes$, consider the local graph type definition. Graph states over the combined graph $(V \cup V', E \cup E')$ are precisely the parallel composition of a graph state over (V, E) and one over (V', E'). Each type just takes affine combinations of these.

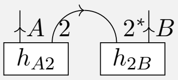
For $<$, use the ordered signalling type definition. Each topological ordering of $(V \cup V', E \cup E' \cup \{v \to v' \mid v \in V, v' \in V'\})$ is a topological ordering of (V, E)

followed by a topological ordering of (V', E'). Setwise Distributivity between $<$ and $\cap$ allows us to combine the quantification over the separate graphs into quantification over the combined graph. □

The equivalence of local and **2**-local graph types demonstrates that any channel with arbitrary capacity over any kind of data, be it classical data, quantum states, or other exotic systems, can be simulated by an affine combination of classical channels with 1 bit capacity. This is not too surprising since affine combinations need not preserve information capacity - whilst taking convex combinations of pure channels will only preserve or decrease capacity, affine combinations can invert this allowing us to recover pure channels of higher capacity from mixed channels of lower capacity. The **2**-local graph type definition gives an additional alternative definition for sequence types, refining Definition 2.6.3 to fix the intermediate system to **2**.

Theorem 2.8.13: Affine-Bit Sufficiency Theorem

$$\mathbf{A} < \mathbf{B} = \Bigg(A \otimes B, \Bigg\{$$



$$\Bigg| \begin{matrix} h_{A2} \in c_{\mathbf{A} \parr \mathbf{2}}, \\ h_{2B} \in c_{\mathbf{2}^* \parr \mathbf{B}} \end{matrix} \Bigg\}^{**} \Bigg)$$

Proof. Immediate from Equation 2.244 and the Graph Equivalence Theorem. □

Example 2.8.14

Circuit knitting is a technique in quantum computing to simulate a quantum circuit that is too large to run on a single quantum device by running a collection of smaller circuits across several devices and combining the statistics. This can focus on either cutting qubit wires [111] or cutting entangling gates to divide the circuit [48].

We can view the correctness of these methods as a consequence of the Sum of Orders Theorem and Affine-Bit Sufficiency Theorem. For example, any entangling gate between two qubits is a process of type $(\mathbf{Q} \multimap \mathbf{Q}) \parr (\mathbf{Q} \multimap \mathbf{Q})$. We can, firstly, represent this as an affine combination of one-way signalling processes $(\mathbf{Q} \multimap \mathbf{Q}) < (\mathbf{Q} \multimap \mathbf{Q}) \cup (\mathbf{Q} \multimap \mathbf{Q}) > (\mathbf{Q} \multimap \mathbf{Q})$. Each of those may still pass quantum information between the two qubits, but they can also be simulated by an affine combination of local channels that pass a single classical bit between them. Our simulation can either involve live classical communication between multiple quantum devices to perform this, or handle it offline by rerunning the later circuit multiple times with different classical inputs and combining the probabilities of observations.

2.8.5 Standard Forms

Even after showing that all definitions of graph types coincide exactly, graph types built from distinct graphs may describe the same subspace of linear maps. In this section, we will work towards a standard form for graph types which are unique and characterise inclusion and composition. Firstly, instead of the exact graph we find that only the partial order induced by it (i.e. its transitive closure) matters. Combining this with the Acyclicity Lemma gives a complete justification for using partial orders to represent causal structures.

Lemma 2.8.15

$$\mathbf{Gr}_G^\Gamma = \mathbf{Gr}_{G^+}^\Gamma$$

where G^+ is the transitive closure of G.

Proof. This is immediate from the signalling graph type definition, since $U \subseteq V$ is down-closed wrt G iff it is down-closed wrt G^+. □

Next, we look towards details in the local interpretations and how they induce equivalences between graph types. For example, if it maps every vertex to $\mathbf{I}$ (i.e. the graph states are scalars), then there is a unique state.

Corollary 2.8.16

$$\mathbf{Gr}_{(V,E)}^{\{v\mapsto\mathbf{I}\}_{v\in V}} \cong \mathbf{I}$$

Proof.

$$\mathbf{Gr}_{(V,E)}^{\{v\mapsto\mathbf{I}\}_{v\in V}} = \bigcap_{O \text{ topological sort of } (V,E)} \mathop{<}_{v\in O} \mathbf{I} = \mathop{\invamp}_{v\in O} \mathbf{I} \cong \mathbf{I} \tag{2.247}$$

□

Moreover, this can be observed piecewise by the ability to remove vertices with trivial local interpretation. We still need to take the transitive closure so we don't lose any information pathways that passed through this vertex.

Lemma 2.8.17

$$\mathbf{Gr}_G^{\Gamma, v \mapsto I} \cong \mathbf{Gr}_{G^+ \setminus \{v\}}^{\Gamma}$$

Proof. This is straightforward from the ordered graph type definition, since an ordering of $V \setminus \{v\}$ is topological wrt G iff it is topological wrt $G^+ \setminus \{v\}$. □

Suppose, instead, that a vertex is interpreted as a first-order object. Because of the degeneracy of having a single causal effect, there is no way for a local external agent to signal information to other vertices, so we can prune away any outgoing edges from that vertex and preserve the graph type. Again, we need to consider transitive closures so we don't disturb indirect signalling pathways. Dually, for first-order dual objects no information can be observed locally, so we can prune away incoming edges. Combining the two recovers Lemma 2.8.17 by removing both incoming and outgoing edges from a vertex.

Lemma 2.8.18

Given two transitive DAGs $G = (V, E)$ and $G' = (V, E \cup (u \to v))$ differing by a single edge, $\mathbf{Gr}_G^{\Gamma} = \mathbf{Gr}_{G'}^{\Gamma}$ iff $\Gamma(u)$ or $\Gamma(v)^*$ is first-order.

Proof. Again, we will focus on the signalling graph type definition. $c_{\mathbf{Gr}_G^{\Gamma}} \subseteq c_{\mathbf{Gr}_{G'}^{\Gamma}}$ is immediate since any $U \subset V$ that is down-closed wrt G' is also down-closed wrt G. We will just show that the opposite direction of inclusion holds iff $\Gamma(u)$ or $\Gamma(v)^*$ is first-order.

$\Longrightarrow$: aiming for the contrapositive, suppose neither of $\Gamma(u)$ or $\Gamma(v)^*$ is first-order, i.e. $\left|c_{\Gamma(u)}^*\right|, \left|c_{\Gamma(v)}\right| > 1$. Then there exists a binary encoding $f \in c_{\Gamma(u) < \Gamma(v)} \setminus c_{\Gamma(u) > \Gamma(v)}$ which actively signals from u to v. Let h be a parallel composition of f with some separable states $h_{v'} \in c_{\Gamma(v')}$ for each $v' \notin \{u, v\}$. $h \in c_{\mathbf{SiGr}_{G'}^{\Gamma}}$ because any down-closed set wrt G' containing v also contains u. However, $\mathrm{in}_G(v)$ is down-closed wrt G and does not contain u, so the active signalling of f means $h \notin c_{\mathbf{SiGr}_G^{\Gamma}}$.

$\Longleftarrow$: Consider an arbitrary $h \in c_{\mathbf{SiGr}_{G'}^{\Gamma}}$, so it is non-signalling into any subset down-closed wrt G'. Then consider some $U \subset V$ that is down-closed wrt G but not wrt G', i.e. $v \in U$ but $u \notin U$. Transitivity of G and G' then imply that both $U \cup \{u\}$ and $U \setminus \{v\}$ are down-closed wrt G':

- $U \cup \{u\}$: for any $(w \to u) \in G'$, $(w \to v) \in G'$ by transitivity and hence $w \in U$ by down-closure of U.

- $U \setminus \{v\}$: for any $(v \to w) \in G'$, $(u \to w) \in G'$ by transitivity and $(u \to w) \in G$ since the only edge that differs is $(u \to v)$. By down-closure of U wrt G and $u \notin U$, we have $w \notin U$.

If $\Gamma(u)$ is first-order, there is a unique effect $\text{⫯}_{\Gamma(u)}$. We already have h is non-signalling into $U \cup \{u\}$, so applying $\text{⫯}_{\Gamma(u)}$ must give a unique marginal over U.

Suppose instead $\Gamma(v)$ is first-order dual with unique state $\text{⫰}_{\Gamma(v)}$. Picking any choice of effects $\{\pi_w\}_{w \in V \setminus U}$, we consider the marginal h_U over U when we apply these. Since $\Gamma(v)$ is first-order dual, its local causal effects (the states of $\Gamma(v)^*$) include a basis. Whichever basis element we apply to h_U, we must get the constant marginal $h_{U \setminus \{v\}}$ since h is non-signalling into $U \setminus \{v\}$, so by equality on every element of a basis we have $h_U = h_{U \setminus \{v\}} \otimes \text{⫰}_{\Gamma(v)}$. This is independent of our choices $\{\pi_w\}_{w \in V \setminus U}$, so h is non-signalling into U. □

Taking the transitive closure and then pruning edges at first-order (dual) objects yields a procedure that reduces the graph G to a standard form $\overline{G}$. The final two lemmas for this section show that this form is a canonical normal form in the sense that it uniquely identifies a graph type and characterises both inclusion of graph types and dual inclusion (i.e. whether we can always compose graph states to yield the unit scalar).

Theorem 2.8.19: Graph Inclusion Theorem

Given a DAG $G = (V, E)$ and a local interpretation Γ, let $\overline{G} = \left(V, \overline{E}\right)$ be the DAG with

$$(u \to v) \in \overline{E} \overset{\text{def}}{\iff} (u \to v) \in E^+ \wedge \left|c^*_{\Gamma(u)}\right| > 1 \wedge \left|c_{\Gamma(v)}\right| > 1 \tag{2.248}$$

Then $\mathbf{Gr}^\Gamma_G = \mathbf{Gr}^\Gamma_{\overline{G}}$. Furthermore, for any $G' = (V, E')$ over the same vertices, $c_{\mathbf{Gr}^\Gamma_G} \subseteq c_{\mathbf{Gr}^\Gamma_{G'}}$ iff $\overline{E} \subseteq \overline{E'}$.

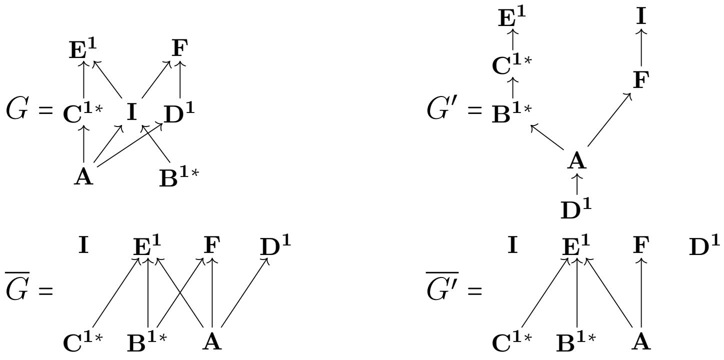
Proof. $\mathbf{Gr}^\Gamma_G = \mathbf{Gr}^\Gamma_{\overline{G}}$ follows from Lemmas 2.8.15 and 2.8.18 by pruning edges one by one from the transitive closure. The remainder now requires us to show that $c_{\mathbf{Gr}^\Gamma_{\overline{G}}} \subseteq c_{\mathbf{Gr}^\Gamma_{\overline{G'}}}$ iff $\overline{E} \subseteq \overline{E'}$.

$\Longleftarrow$: If $\overline{E} \subseteq \overline{E'}$, then the inclusion of graph types follows straightforwardly from any of the graph type definitions.

$\Longrightarrow$: Suppose, aiming for the contrapositive, that there is some $(u \to v) \in \overline{E} \smallsetminus \overline{E'}$. By definition of $\overline{E}$, $\left|c^*_{\Gamma(u)}\right| > 1$ and $\left|c_{\Gamma(v)}\right| > 1$, so we can use a binary encoding $\Gamma(u) \to \Gamma(v)$ to generate a counterexample in $c_{\mathbf{SiGr}^{\Gamma}_{\overline{E}}} \smallsetminus c_{\mathbf{SiGr}^{\Gamma}_{\overline{E'}}}$, following the same proof as the $\Longrightarrow$ direction of Lemma 2.8.18. □

Example 2.8.20

Suppose we have graph types described by the following graphs, with the vertex labels picking out some local interpretation:



(2.249)

By just looking at G and G', it may not be immediately obvious whether one structure generalises the other, whereas we can easily see that $\overline{G'}$ is a subgraph of $\overline{G}$. To get from G to $\overline{G}$, the arcs out of $\mathbf{A}$ and $\mathbf{B^{1*}}$ are introduced by taking the transitive closure. Then the first-order $\mathbf{D^1}$, $\mathbf{E^1}$, and $\mathbf{I}$ act as simple outputs of the network, so we remove their outgoing edges; it is impossible for local agents at them to signal to anything else since they only have a single choice of effect to apply (discarding). Similarly, the first-order dual $\mathbf{B^{1*}}$, $\mathbf{C^{1*}}$, and $\mathbf{I}$ act as simple inputs; an agent would see that the network acts locally like discarding regardless of actions made at other vertices, so nothing can ever signal to them.

Theorem 2.8.21: Graph Compatibility Theorem

Let $G = (V, E)$ and $G' = (V, E')$ be two DAGs over the same set of vertices, and $\Gamma, \Gamma^* : V \to \mathrm{Ob}(\mathrm{Caus}[\mathcal{C}])$ be dual local interpretation functions, $\forall v \in V.\ \Gamma^*(v) = \Gamma(v)^*$. Then $c_{\mathbf{Gr}^{\Gamma}_{G}} \subseteq c_{\left(\mathbf{Gr}^{\Gamma^*}_{G'}\right)^*}$ iff $(V, \overline{E} \cup \overline{E'})$ is acyclic.

Proof. From the Graph Inclusion Theorem, we get $c_{\mathbf{Gr}^{\Gamma}_{G}} \subseteq c_{\left(\mathbf{Gr}^{\Gamma^*}_{G'}\right)^*}$ iff $c_{\mathbf{Gr}^{\Gamma}_{\overline{G}}} \subseteq c_{\left(\mathbf{Gr}^{\Gamma^*}_{\overline{G'}}\right)^*}$. This inclusion holds exactly when, if we compose any pair of graph states over $\overline{G}$ with Γ and over $\overline{G'}$ with Γ^* by contracting the components at each vertex, then the resulting scalar is id_I.

$\Longrightarrow$: Aiming for the contrapositive, assume $\left(V, \overline{E} \cup \overline{E'}\right)$ is not acyclic. Pick a minimal cycle σ (so it visits each vertex at most once) and break it down into segments of edges in $\overline{E}$ and edges in $\overline{E'}$ (if the edge is in both, we may pick either arbitrarily). Let G_σ be the graph over $V \uplus V$ consisting of $\overline{G}$ and $\overline{G'}$ in parallel with additional edges between matching vertices marking the points where σ transitions from one graph to the other:

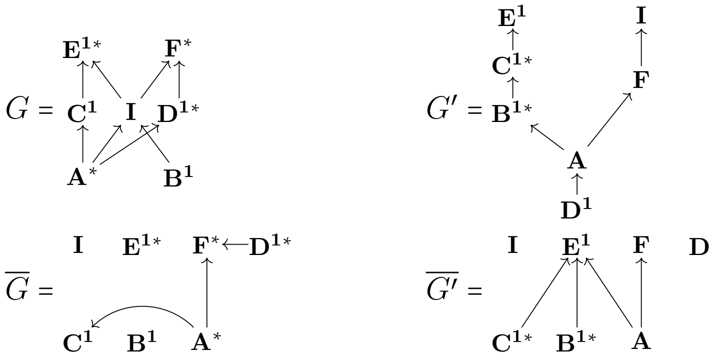
$$(u_a \to v_b) \in E_\sigma \overset{\text{def}}{\iff} \begin{cases} (u \to v) \in \overline{E} & a = b = 1 \\ (u \to v) \in \overline{E'} & a = b = 2 \\ \begin{pmatrix} \exists\, (w \to u) \in \sigma \cap \overline{E}, \\ (v \to w') \in \sigma \cap \overline{E'}.u = v \end{pmatrix} & a = 1, b = 2 \\ \begin{pmatrix} \exists\, (w' \to u) \in \sigma \cap \overline{E'}, \\ (v \to w) \in \sigma \cap \overline{E}.u = v \end{pmatrix} & a = 2, b = 1 \end{cases} \tag{2.250}$$

Graph states over G_σ resemble a pair of graph states over $\overline{G}$ and $\overline{G'}$, contracted only over the vertices where σ transitions from one graph to the other, with local effects on all other vertices. They aren't precisely the same thing, since edge interpretations must be first-order objects and the contractions may occur over some $\Gamma(v)$ which is neither first-order nor first-order dual. However, by construction of $\overline{E}$ and $\overline{E'}$, we know that any transition from $\overline{E}$ to $\overline{E'}$ at vertex v has $\left|c_{\Gamma(v)}\right| > 1$ since v must have a predecessor in $\overline{E}$ and a successor in $\overline{E'}$. This implies the existence of some binary encodings which signal to and from $\Gamma(v)$, e.g. if $(u \to v) \in \sigma \cap \overline{E}$ and $(v \to w) \in \sigma \cap \overline{E'}$, binary encodings give morphisms in $c_{\Delta(u\to v)^* < \Gamma(v)} \setminus c_{\Delta(u \to v)^* > \Gamma(v)}$ and $c_{\Gamma(v)^* < \Delta(v \to w)} \setminus c_{\Gamma(v)^* > \Delta(v \to w)}$. We can therefore apply the same cyclic feedback construction from the proof of the Acyclicity Lemma to generate two pairs of graph states in $\mathbf{Gr}^{\Gamma}_{\overline{G}}$ and $\mathbf{Gr}^{\Gamma^*}_{\overline{G'}}$ whose inner products give distinct scalars - at least one of these inner products must be not id_I.

$\Longleftarrow$: Now we suppose that $\left(V, \overline{E} \cup \overline{E'}\right)$ is acyclic. If we take any pair of graph states from $\mathbf{Gr}^{\Gamma}_{\overline{G}}$ and $\mathbf{Gr}^{\Gamma^*}_{\overline{G'}}$ and contract the components at each vertex into a single component, we obtain a valid graph state of $\mathbf{Gr}^{\{v \mapsto \mathbf{I}\}_{v \in V}}_{\left(V, \overline{E} \cup \overline{E'}\right)}$ (this is a valid object by the Acyclicity Lemma). Corollary 2.8.16 tells us there is a unique scalar formed by this composition, which is id_I. □

Example 2.8.22

Let's take the same graphs as Example 2.8.20 but dualise the local interpretation for G.



(2.251)

If we just looked at G and G', we would see a cycle in $E \cup E'$ from $\mathbf{A}^* \to \mathbf{D^{1*}}$ in G and $\mathbf{D^1} \to \mathbf{A}$ in G'. However, because $\mathbf{D^1}$ is first-order, $\mathbf{D^1} < \mathbf{A} = \mathbf{D^1} \otimes \mathbf{A}$ and $\mathbf{A}^* < \mathbf{D^{1*}} = \mathbf{A}^* \otimes \mathbf{D^{1*}}$ mean it is impossible to send any information around this cycle. Once we prune off either edge, it becomes immediately clear that the causal structures are compatible.

One can intuitively read this as saying that the graph types are compatible exactly when there is a total causal ordering between the vertices that respects both graph types. This falls in line with an exact characterisation of the dual space of a graph type as a union of topological orderings.

Proposition 2.8.23

$$\left(\mathbf{Gr}^{\Gamma}_{G}\right)^* = \bigcup_{[v_1,\dots,v_n]\in\mathrm{sort}(G)} \Gamma(v_1)^* < \dots < \Gamma(v_n)^*$$

Proof. This is immediate from the ordered graph type definition, the de Morgan dualities between $\cup$ and $\cap$, and self-duality of $<$ (Equation 2.145). □

2.8.6 Preservation of Local Structure

This section will look at further interplay between graph types and the local interpretation functions. In particular, if the local interpretation maps a vertex to an object with some structure, we will see when that structure can be lifted to the level of the graph types themselves. For example, we have a generalisation of the Setwise Distributivity laws where graph types will preserve unions and intersections in the local interpretations.

Proposition 2.8.24

$$\mathbf{Gr}_G^{\Gamma, v \mapsto \mathbf{A} \cup \mathbf{B}} = \mathbf{Gr}_G^{\Gamma, v \mapsto \mathbf{A}} \cup \mathbf{Gr}_G^{\Gamma, v \mapsto \mathbf{B}} \tag{2.252}$$

$$\mathbf{Gr}_G^{\Gamma, v \mapsto \mathbf{A} \cap \mathbf{B}} = \mathbf{Gr}_G^{\Gamma, v \mapsto \mathbf{A}} \cap \mathbf{Gr}_G^{\Gamma, v \mapsto \mathbf{B}} \tag{2.253}$$

Proof. $\cup$: Using Setwise Distributivity for $\parr$, the component type at v from $\mathbf{Gr}_G^{\Gamma, v \mapsto \mathbf{A} \cup \mathbf{B}}$ satisfies:

$$\mathrm{Comp}_G^{\Gamma, v \mapsto \mathbf{A} \cup \mathbf{B}; \Delta}(v) = (\mathbf{A} \cup \mathbf{B}) \parr (\cdots) = \mathbf{A} \parr (\cdots) \cup \mathbf{B} \parr (\cdots) \tag{2.254}$$

The v component of any graph state of $\mathbf{Gr}_G^{\Gamma, v \mapsto \mathbf{A}}$ or $\mathbf{Gr}_G^{\Gamma, v \mapsto \mathbf{B}}$ is trivially valid for the component typing above, and conversely the v component for any graph state of $\mathbf{Gr}_G^{\Gamma, v \mapsto \mathbf{A} \cup \mathbf{B}}$ is an affine combination of valid components for $\mathbf{Gr}_G^{\Gamma, v \mapsto \mathbf{A}}$ and $\mathbf{Gr}_G^{\Gamma, v \mapsto \mathbf{B}}$. Hence, any graph state of one type can always be broken down as an affine combination of graph states of the other.

$\cap$: Immediate from the ordered graph type definition and Setwise Distributivity for $<$. □

More generally, if a local interpretation maps a vertex to another graph type, then we can flatten the nesting of graphs into a single graph type by graph substitution.

Definition 2.8.25: Graph substitution

Let $G = (V, E)$ and $G' = (V', E')$ be two DAGs with disjoint vertices $V \cap V' = \varnothing$, and choose a vertex $v \in V$. We define the *substitution of G' for v in G* as the graph $G[G'/v] := ((V \setminus v) \cup V', E[E'/v])$ with

$$(u \to w) \in E[E'/v] \overset{\text{def}}{\iff} \begin{cases} u \to w \in E & u, w \in V \setminus \{v\} \\ u \to v \in E & u \in V \setminus \{v\}, w \in V' \\ v \to w \in E & u \in V', w \in V \setminus \{v\} \\ u \to w \in E' & u, w \in V' \end{cases} \tag{2.255}$$

i.e. G' takes the place of v, and each edge into and out of v is duplicated for every vertex in G'.

Proposition 2.8.26

$$\mathbf{Gr}_G^{\Gamma, v \mapsto \mathbf{Gr}_{G'}^{\Gamma'}} = \mathbf{Gr}_{G[G'/v]}^{\Gamma, \Gamma'}$$

Proof. We start with the transitive closure $\mathbf{Gr}^{\Gamma_v}_{G} = \mathbf{Gr}^{\Gamma_v}_{G^+}$ from Lemma 2.8.15 where $\Gamma_v = \Gamma \cup \left\{v \mapsto \mathbf{Gr}^{\Gamma'}_{G'}\right\}$. We can then use Lemma 2.8.17 to insert additional vertices interpreted as $\mathbf{I}$ to collect predecessors and successors of v, meaning v has precisely one predecessor p and one successor s. Let's refer to this new graph type as $\mathbf{Gr}^{\Gamma_A}_{G_A}$.

Examining the component type at v (assuming wlog the constant edge interpretation $\Delta_{\mathbf{2}}$):

$$\begin{aligned}\Gamma_A(v) \parr \mathbf{2}^* \parr \mathbf{2} &\cong \mathbf{2}^* < \mathbf{Gr}^{\Gamma'}_{G'} < \mathbf{2}\\ &= \mathbf{Gr}^{i\mapsto\mathbf{2}^*}_{(\{i\},\varnothing)} < \mathbf{Gr}^{\Gamma'}_{G'} < \mathbf{Gr}^{o\mapsto\mathbf{2}}_{(\{o\},\varnothing)}\\ &\cong \mathbf{Gr}^{\Gamma',i\mapsto\mathbf{2}^*,o\mapsto\mathbf{2}}_{(V'\cup\{i,o\},E'\cup\{i\to u,u\to o\ \mid\ u\in V'\})}\end{aligned} \tag{2.256}$$

where we have used Corollary 2.8.12 to represent this as a single graph type. This new graph, which we will denote as $i < G' < o$, extends G' with a global source i and sink o.

Considering any graph state g of $\mathbf{Gr}^{\Gamma_A}_{G_A}$, the component at v can hence be expressed as an affine combination of graph states of $i < G' < o$. By linearity, g can be expressed as an affine combination of graph states of $\mathbf{Gr}^{\Gamma,\Gamma',\{p,i,o,s\}\mapsto\mathbf{I}}_{G_B}$ where $G_B = ((V \setminus v) \cup V' \{p, i, o, s\}, E_B)$ and

$$E_B = \left(\begin{array}{l} \quad\{(u \to w) \in E \mid u, w \neq v\}\\ \cup\{(u \to p) \mid (u \to v) \in E\}\\ \cup\{(p \to i)\}\\ \cup\{(i \to w) \mid w \in V'\}\\ \cup E'\\ \cup\{(u \to o) \mid u \in V'\}\\ \cup\{(o \to s)\}\\ \cup\{(s \to w) \mid (v \to w) \in E\}\end{array}\right) \tag{2.257}$$

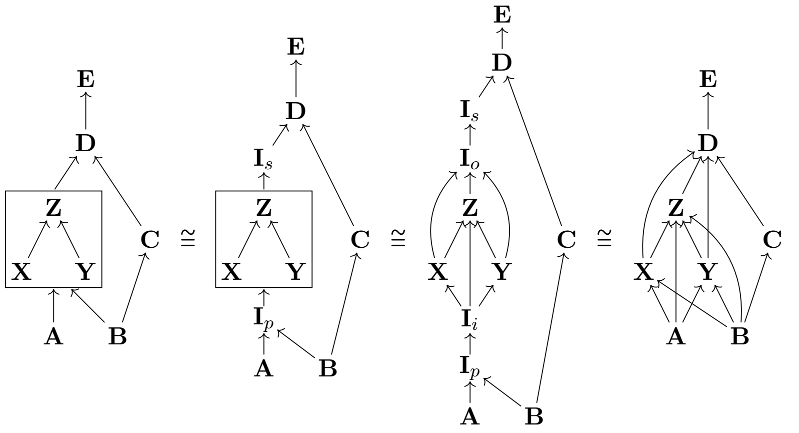
Note that the interpretations of i and o have become $\mathbf{I}$ as the $\mathbf{2}^*$ and $\mathbf{2}$ systems have been contracted according to the edges $(p \to i)$ and $(o \to s)$. Conversely, any graph state of $\mathbf{Gr}^{\Gamma,\Gamma',\{p,i,o,s\}\mapsto\mathbf{I}}_{G_B}$ is immediately a graph state of $\mathbf{Gr}^{\Gamma_A}_{G_A}$ up to unitors, hence these graph types are isomorphic.

Finally, we reach $\mathbf{Gr}^{\Gamma,\Gamma'}_{G[G'/v]}$ by removing $\{p, i, o, s\}$ with Lemma 2.8.17 and undoing the transitive closure by Lemma 2.8.15.

In summary, we have achieved an isomorphism between $\mathbf{Gr}^{\Gamma,v\mapsto\mathbf{Gr}^{\Gamma'}_{G'}}_{G}$ and $\mathbf{Gr}^{\Gamma,\Gamma'}_{G[G'/v]}$. Furthermore, it is an equality, since the underlying carrier objects are identical and the isomorphism is composed entirely out of structural morphisms (e.g. unitors, associators, permutations) which must equal the identity by coherence in $\mathcal{C}$. □

Example 2.8.27

The following sequence of graph types also shows the intermediate stages from the proof of Proposition 2.8.26. A box represents a single vertex whose interpretation is itself the graph type inside the box.



(2.258)

This result shows when it is safe to map between different levels of abstraction for a causal scenario by grouping parties together or splitting them up. We can see this as generalising Corollary 2.8.12, since both series and parallel composition of graphs are instances of graph substitution. In terms of dependence categories [104], this substitution result is exactly the natural associativity isomorphism from the pseudo-algebra definition.

2.8.7 Causal Relations Beyond Graph Types

Viewing the partial order of a causal structure as a representation for the collection of totalisations of the order immediately highlights a shortcoming of using partial orders in the first place: that there exist combinations of total orderings which do not correspond to any partial order. Analogously, Definition 2.8.9 for ordered graph types does not give a universal way to describe any intersection of linear causal structures. In particular, taking the set-wise intersection of two graphs is not precisely reflected by the intersection of their graph types.

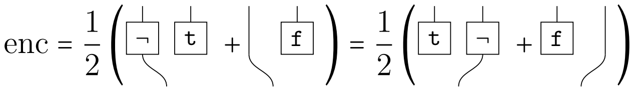
To see this, consider the encoder for the one-time-pad scheme $\mathrm{enc} : 2 \to 2 \otimes 2$:

$$\mathrm{enc}(\mathtt{t}) = \tfrac{1}{2}(\mathtt{t} \otimes \mathtt{f} + \mathtt{f} \otimes \mathtt{t}) \tag{2.259}$$

$$\mathrm{enc}(\mathtt{f}) = \tfrac{1}{2}(\mathtt{t} \otimes \mathtt{t} + \mathtt{f} \otimes \mathtt{f}) \tag{2.260}$$

This is a famous example to show that signalling relations between individual parties are not sufficient to infer the signalling relations between subsets of parties. The input cannot signal information successfully to either one of the outputs (i.e. marginalising out the other output means we can no longer recover any information about the input), but it does signal to the joint system over both outputs since we can recover the input by XOR.

We can show that enc is in the intersection of the graph types $\mathbf{Gr}^{\Gamma}_{\mathbf{In} \to \mathbf{Out_1}} \cap \mathbf{Gr}^{\Gamma}_{\mathbf{In} \to \mathbf{Out_2}}$ with a single edge from **In** to one output and the other disconnected.



(2.261)

Expanding the signalling graph type definitions, enc is compatible with:

$$\mathbf{Out_1} < (\mathbf{In} \parr \mathbf{Out_2}) \tag{2.262}$$

$$\mathbf{Out_2} < (\mathbf{In} \parr \mathbf{Out_1}) \tag{2.263}$$

$$\mathbf{In} < (\mathbf{Out_1} \parr \mathbf{Out_2}) \tag{2.264}$$

$$(\mathbf{In} \parr \mathbf{Out_1}) < \mathbf{Out_2} \tag{2.265}$$

$$(\mathbf{In} \parr \mathbf{Out_2}) < \mathbf{Out_1} \tag{2.266}$$

However, whilst traditionally we would think of the intersection of these graphs as being the graph with no edges, enc is not compatible with the corresponding graph type $\mathbf{Gr}^{\Gamma}_{\varnothing}$. This type requires all of 2.262-2.266 as well as $(\mathbf{Out_1} \parr \mathbf{Out_2}) < \mathbf{In}$ which enc fails to satisfy. Moreover, there is no graph type that exactly matches just the intersection of 2.262-2.266.

In general, we still get an inclusion relating graph intersection to the intersection of graph types, just not equality. Similarly, when $(V, E \cup E')$ is acyclic, union of graphs is only preserved in one direction.

$$c_{\mathbf{Gr}^{\Gamma}_{(V,E \cap E')}} \subseteq c_{\mathbf{Gr}^{\Gamma}_{(V,E)} \cap \mathbf{Gr}^{\Gamma}_{(V,E')}} \tag{2.267}$$

$$c_{\mathbf{Gr}^{\Gamma}_{(V,E)} \cup \mathbf{Gr}^{\Gamma}_{(V,E')}} \subseteq c_{\mathbf{Gr}^{\Gamma}_{(V,E \cup E')}} \tag{2.268}$$

Remark 2.8.28

We conjecture that the distribution of union and intersection of Remark 2.7.8 does still hold over graph types with the same local interpretation, meaning Hoffreumon and Oreshkov's standard form [64] (a union of intersections of total orderings) remains general enough to capture all causal data.

$$\left(\mathbf{Gr}^{\Gamma}_{G_1} \cup \mathbf{Gr}^{\Gamma}_{G_2}\right) \cap \mathbf{Gr}^{\Gamma}_{G_3} \overset{?}{=} \left(\mathbf{Gr}^{\Gamma}_{G_1} \cap \mathbf{Gr}^{\Gamma}_{G_3}\right) \cup \left(\mathbf{Gr}^{\Gamma}_{G_2} \cap \mathbf{Gr}^{\Gamma}_{G_3}\right) \tag{2.269}$$

However, even if we can recover a union of intersections form, it still won't be unique in general. For example, a graph type on a graph with n vertices and no edges can be written as an intersection of all $n!$ orderings using the ordered graph type definition, but by treating it as an n-fold tensor product we can recursively apply the Non-signalling Theorem to write it as an intersection of only 2^n terms. It would be interesting for future work to determine exactly when terms can be dropped from an intersection or union in this way.

2.9 Partiality in Higher-Order Theories

So far, we have only considered a theory of causal processes: those that can be physically realised with probability 1. There is a common need to instead investigate partial processes - those that may only be realised with some probability, and where this probability may depend on the context they are applied in. For example, operational theories phrased in terms of tests will want a way to talk about the process occurring on particular test outcomes, or more broadly we need to break causality in order to calculate probabilities of observations or conditionals. This section covers a few candidates for ways to incorporate partiality into the Caus $[-]$ construction and where they fall short of either interacting well with causal categories or where the adaptation to higher-order loses many key properties satisfied by first-order theories of partial maps. Two of these options will be directly inspired by *effectus theory*.

Effectus theory is a popular framework in category theory for process theories admitting views either as a category of total maps (akin to causal processes) or partial maps. Given the total map category, one can obtain the partial category using a simple construction, and vice versa. It is sufficiently general to encompass examples such as sets and total/partial functions, first-order probability theory with stochastic/substochastic maps, and first-order quantum theory with trace-preserving/trace-non-increasing maps. We refer the reader to [25] for a comprehensive introduction.

Definition 2.9.1: Effectus

An *effectus* is a category $\mathcal{B}$ with finite coproducts $(+, 0)$ and a terminal object 1, satisfying:

1. diagrams of the following forms are pullbacks in $\mathcal{B}$:

$$\begin{tikzcd}
X + Y \arrow[r, "{[\iota_1, g ⨾ \iota_2]}"] \arrow[d, "{[f ⨾ \iota_1, \iota_2]}"'] \arrow[dr, phantom, very near start, "\lrcorner"] & X + B \arrow[d, "{[f ⨾ \iota_1, \iota_2]}"] \\
A + Y \arrow[r, "{[\iota_1, g ⨾ \iota_2]}"] & A + B
\end{tikzcd} \tag{2.270}$$

$$\begin{tikzcd}
X \arrow[r, "f"] \arrow[d, "\iota_1"'] \arrow[dr, phantom, very near start, "\lrcorner"] & A \arrow[d, "\iota_1"] \\
X + Y \arrow[r, "{[f ⨾ \iota_1, g ⨾ \iota_2]}"] & A + B
\end{tikzcd} \tag{2.271}$$

2. $\mathbb{IV} = [[\iota_1, \iota_2], \iota_2] : (1 + 1) + 1 \to 1 + 1$ and $\mathbb{X} : [[\iota_2, \iota_1], \iota_2] : (1 + 1) + 1 \to 1 + 1$ are jointly monic.

$\mathcal{B}$ is referred to as the category of *total maps*, and $\mathrm{Par}(\mathcal{B})$ (the Kleisli category of the *maybe monad* $(-) + 1$) is the category of *partial maps*.

$\mathcal{B}$ is a *monoidal effectus* if it additionally carries a symmetric monoidal structure $\otimes$ where the unit is the terminal object 1 and $\otimes$ distributes over finite coproducts ($(X \otimes A) + (Y \otimes A) \cong (X + Y) \otimes A$ and $0 \cong 0 \otimes A$).

Partial maps $\mathrm{Par}(\mathcal{B})(A, B)$ are total maps $\mathcal{B}(A, B + 1)$ that can either terminate by successfully producing a state of B or otherwise fail over, modelled by a unique error state from the terminal 1. The pullback conditions guarantee a degree of independence between the subsystems of a coproduct; this notably implies that the error branch can have no influence on the successful branch and vice versa.

Within $\mathrm{Par}(\mathcal{B})$, $+$ is still a coproduct and 0 is now a zero object. This gives *partial projections* which allow us to focus on a particular outcome of some branching computation.

$$\rhd_1 := [\mathrm{id}_X, 0_{Y,X}] \in \mathrm{Par}(\mathcal{B})(X + Y, X) \tag{2.272}$$

$$\rhd_2 := [0_{X,Y}, \mathrm{id}_Y] \in \mathrm{Par}(\mathcal{B})(X + Y, Y) \tag{2.273}$$

Joint monicity of $\mathbb{IV}$ and $\mathbb{X}$ in $\mathcal{B}$ is equivalent to joint monicity of the partial projections in $\mathrm{Par}\,(\mathcal{B})$, meaning that a branching computation can be completely characterised by looking at each branch individually.

Without the distributivity law required by a monoidal effectus, a computation that branches locally to a state of $(X \oplus Y) \otimes A$ would have to unite both branches into a common outcome (giving a global state $Z \otimes A$) before the two parallel systems can interact. Elevating the local branching to a global branch $(X \otimes A) + (Y \otimes A)$ allows us to have different interactions with A dependent on the branch outcome.

These are all very natural and desirable properties which we would expect any sensible theory of partial maps to satisfy. The final requirement of a monoidal effectus is for the monoidal unit to be terminal, which perfectly captures causality for a first-order theory but is too strong for a higher-order theory.

Proposition 2.9.2

In any ISOMIX category for which the monoidal unit is terminal, all objects are isomorphic to the zero object.

Proof. By duality, 1^* is initial and the ISOMIX rule $1 \cong 1^*$ implies it is a zero object. Then any homset satisfies $\mathcal{C}(A, B) \cong \mathcal{C}(1, A^* \parr B) = \{0\}$. □

The first obvious direction to take towards finding an appropriate higher-order variant of an effectus is to simply apply the same maybe monad to some $\mathrm{Caus}\,[\mathcal{C}]$ and see whether it gives a similarly intuitive theory of partial maps.

2.9.1 Initial and Terminal Objects in $\mathrm{Caus}\,[\mathcal{C}]$

With the way we have constructed $\mathrm{Caus}\,[\mathcal{C}]$, we can't get very far with mimicking effectuses exactly since the interesting examples won't have a terminal object.

Proposition 2.9.3

$\mathrm{Caus}\,[\mathcal{C}]$ has a terminal object (equivalently by duality, an initial object) iff every object is isomorphic to $\mathbf{I}$.

Proof. Suppose that $\mathrm{Caus}\,[\mathcal{C}]$ has a terminal object $\mathbf{1}$. By flatness, there exists a uniform state $⫰_{\mathbf{1}}$ and a uniform effect $⫯_{\mathbf{1}}$ such that $⫰_{\mathbf{1}} \fatsemi ⫯_{\mathbf{1}} = \mathrm{id}_I$.

Consider an arbitrary object $\mathbf{A} \in \mathrm{Ob}\,(\mathrm{Caus}\,[\mathcal{C}])$ and any effect $\pi : \mathbf{A} \to \mathbf{I}$. By terminality, $\pi \fatsemi ⫰_{\mathbf{1}} : \mathbf{A} \to \mathbf{1}$ must equal the unique morphism $!_{\mathbf{A}} : \mathbf{A} \to \mathbf{1}$. But then $\pi = \pi \fatsemi \mathrm{id}_I = \pi \fatsemi ⫰_{\mathbf{1}} \fatsemi ⫯_{\mathbf{1}} =!_{\mathbf{1}} \fatsemi ⫯_{\mathbf{1}}$.

This shows that every object has a unique effect and also a unique state by duality, making them all isomorphic to $\mathbf{I}$. □

It turns out that the flatness condition is too strong to allow for initial and terminal objects to exist in $\mathrm{Caus}[\mathcal{C}]$. They exist in $\mathcal{C}$ by assumption via the zero object 0, but the uniqueness of the state $0_{0,I} \in \mathcal{C}(0, I)$ by initiality leaves only two candidates for the causal sets - the empty set $\varnothing$ or the singleton $\{0_{0,I}\}$ (with dual set $\{0_{0,I}\}^* = \varnothing$) - neither of which is flat. For the purpose of this section, we will consider relaxing the flatness condition slightly to permit empty sets of states or effects.

Definition 2.9.4: Thin sets

A set $c \subseteq \mathcal{C}(I, A)$ is *thin* if either c is flat or one of c and c^* is empty.

Redefining $\mathrm{Caus}[\mathcal{C}]$ to be built from thin, closed sets of states gives a very similar category. Almost all the properties shown for $\mathrm{Caus}[\mathcal{C}]$ so far will still hold, since the cases for empty $c_{\mathbf{A}}$ or $c_{\mathbf{A}}^*$ will tend to hold by some vacuous argument. Some additional care must be taken when defining $<$ as we cannot always assume a marginal exists, just requiring it to be unique if it does. One exception to this is the First-Order Theorem where the identity is now one-way non-signalling when there is *at most one* causal effect.

Much like how the monoidal structure $\otimes$ and biproduct structure $\oplus$ are degenerate in $\mathcal{C}$ but gives rise to multiple distinct monoidal structures $\{\otimes, <, \parr\}$ or additive structures $\{\oplus, \times\}$ in $\mathrm{Caus}[\mathcal{C}]$, the degenerate zero object 0 now gives distinct initial and terminal objects.

Proposition 2.9.5

The initial and terminal objects in $\mathrm{Caus}[\mathcal{C}]$ are respectively

$$\mathbf{0} := (0, \varnothing) \tag{2.274}$$

$$\mathbf{1} := (0, \{0_{I,0}\}) \tag{2.275}$$

and, furthermore, they are duals of each other and are respectively the units for $\oplus$ and $\times$.

Proof. Given any $\mathbf{A}$, there is a unique morphism $0_{0,A} \in \mathcal{C}(0, A)$. The normalisation condition for $0_{0,A} : \mathbf{0} \to \mathbf{A}$ holds vacuously since there are no states of $\mathbf{0}$ to preserve. This is hence a unique morphism in $\mathrm{Caus}[\mathcal{C}]$, making $\mathbf{0}$ initial.

An initial object is always a unit for coproducts.

Conversely, for any $\mathbf{A}$ there is a unique morphism $0_{A,0} \in \mathcal{C}(A, 0)$. For any $\rho \in c_{\mathbf{A}}$, $\rho \mathbin{⨾} 0_{A,0} = 0_{I,0} \in c_{\mathbf{1}}$ by terminality of the zero object, so $0_{A,0} : \mathbf{A} \to \mathbf{1}$. This similarly makes $\mathbf{1}$ terminal in $\mathrm{Caus}[\mathcal{C}]$, and a terminal object is always a unit for products.

For the duality, we note that the zero object is self-dual in any compact closed category $\mathcal{C}$. $\varnothing^*$ will include the full homset of effects since the normalisation condition will vacuously hold. Since there is only one effect from the zero object, $c_{\mathbf{0}}^* = c_{\mathbf{1}}$. Conversely, note that $0_{I,I} \neq \mathrm{id}_I$ since the subtractive closure of scalars is a field (Proposition 2.4.15) requiring the zero and unit to be distinct. Since composing $0_{I,0}$ with the unique effect $0_{0,I}$ gives $0_{I,I} \neq \mathrm{id}_I$, we have $c_{\mathbf{1}}^* = c_{\mathbf{0}}$. □

2.9.2 Descriptive Partiality

The first proposal we will consider is taking the Kleisli category for the maybe monad $(-) \oplus \mathbf{1}$ on $\mathrm{Caus}[\mathcal{C}]$. Mapping the error branch into the zero state means the Kleisli morphisms really only present what happens on the successful outcome.

By examining the possible states of $\mathbf{A} \oplus \mathbf{1}$, taking affine combinations with the zero state leads to there being no bound on the scale of the states we could possibly create. The weakening of flatness to thinness has created objects without any limitation on their morphisms.

Proposition 2.9.6

$$\mathrm{Caus}[\mathcal{C}](\mathbf{A}, \mathbf{B} \oplus \mathbf{1}) \cong \mathcal{C}(A, B)$$

Proof. By Equation 2.107, the effects of $\mathbf{B} \oplus \mathbf{1}$ are copairings of effects of $\mathbf{B}$ and effects of $\mathbf{1}$, of which there are none. $\mathrm{Caus}[\mathcal{C}](\mathbf{A}, \mathbf{B} \oplus \mathbf{1})$ contains all morphisms of $\mathcal{C}(A, B \oplus 0)$ which preserve the causal effects, which in this case holds vacuously for every morphism. The bijection follows from the isomorphism $B \oplus 0 \cong B$ in $\mathcal{C}$. □

This provides a way to "undo" the $\mathrm{Caus}[-]$ construction and obtain the underlying category. If we interpret the (additive) precausal category $\mathcal{C}$ as just providing the mathematical formalism within which we work, this construction provides a notion of *descriptive partiality* where a partial map can be any process that we can possibly describe using the mathematical tools available, regardless of whether one could physically realise it.

The isomorphism $0 \cong 0 \oplus 0$ for the zero object in $\mathcal{C}$ lifts to the causal category as $\mathbf{1} \cong \mathbf{1} \oplus \mathbf{1}$. In computational terms, this means that there is no way to distinguish between different error states - a computation with multiple error branches is treated exactly the same as if there was a single erroneous branch. Relating back to the properties of an effectus, $|\!/ = \times\!/$ are now the unique morphisms by terminality of $\mathbf{1} \oplus \mathbf{1}$, and (joint) monicity follows trivially from terminality of $(\mathbf{1} \oplus \mathbf{1}) \oplus \mathbf{1}$. This matches with the joint monicity of the partial projections $\rhd_i$, which are conveniently mapped into the biproduct projections p_i by the isomorphism $A \oplus 0 \cong A$ in $\mathcal{C}$.

However, this conflation of error sources means our coproduct doesn't represent truly independent branches of computation, as seen by the failure of pullback squares from Equations 2.270 and 2.271. For example, consider any cospan into $\mathbf{1} \oplus \mathbf{1}$. All pullbacks from the terminal give products, which doesn't fit the form required in general.

$$(\mathbf{1} \oplus \mathbf{Y}) \times (\mathbf{X} \oplus \mathbf{1}) \ncong \mathbf{X} \oplus \mathbf{Y} \tag{2.276}$$

$$(\mathbf{X} \oplus \mathbf{Y}) \times \mathbf{1} \ncong \mathbf{X} \tag{2.277}$$

We will soon see in Propositions 2.9.8 and 2.9.10 that these pullbacks still hold for objects with flat causal sets, meaning that we still have independence for non-error branches.

2.9.3 Testable Partiality

One could argue that the role of the terminal in an effectus is to provide a canonical notion of discarding, and it is better to prioritise the essence of discarding over uniqueness of morphisms. Despite $\mathbf{I}$ not being a terminal object, flatness would still give a canonical discarding effect ⫯ for every object. Let's now return to restricting objects of Caus$[\mathcal{C}]$ to have flat sets and look at the $(-) \oplus \mathbf{I}$ monad.

We can similarly interpret Kleisli morphisms $\mathbf{A} \to \mathbf{B} \oplus \mathbf{I}$ as computations which either successfully return some output $\mathbf{B}$ or an error state $\mathbf{I}$, but when $\mathcal{C}$ has positivity conditions (such as for Mat$[\mathbb{R}^+]$ or CP*) this is really a probabilistic combination of the two. This puts a reasonable bound on the scale of partial maps.

Whilst discarding ⫯ is a canonical effect, it is not necessarily unique in a higher-order setting and we do not require the error branch of a partial map to correspond to discarding here. The description of a partial map now not only depends on the partial operation itself, but this additional information guaranteeing that it can definitely be realised on some outcome of a test. In contrast to descriptive partiality, this gives *testable partiality* from the guarantee of a physical implementation.

We can revisit the effectus properties again here and find that we do slightly better in recovering them.

Proposition 2.9.7

The morphisms $\mathbb{V}_I = [[\iota_1, \iota_2], \iota_2] : (\mathbf{I} \oplus \mathbf{I}) \oplus \mathbf{I} \to \mathbf{I} \oplus \mathbf{I}$ and $\mathbb{X}_I = [[\iota_2, \iota_1], \iota_2] : (\mathbf{I} \oplus \mathbf{I}) \oplus \mathbf{I} \to \mathbf{I} \oplus \mathbf{I}$ are jointly monic in $\mathrm{Caus}[\mathcal{C}]$.

Proof. Consider some morphisms $f, g : \mathbf{A} \to (\mathbf{I} \oplus \mathbf{I}) \oplus \mathbf{I}$ such that $f \fatsemi \mathbb{V}_I = g \fatsemi \mathbb{V}_I$ and $f \fatsemi \mathbb{X}_I = g \fatsemi \mathbb{X}_I$. By decomposing the biproducts in $\mathcal{C}$, we have

$$f \fatsemi p_1 = g \fatsemi p_1 \tag{2.278}$$

$$f \fatsemi p_2 + f \fatsemi p_3 = g \fatsemi p_2 + g \fatsemi p_3 \tag{2.279}$$

$$f \fatsemi p_2 = g \fatsemi p_2 \tag{2.280}$$

$$f \fatsemi p_1 + f \fatsemi p_3 = g \fatsemi p_1 + g \fatsemi p_3 \tag{2.281}$$

Combining these with cancellativity of addition, we have $f \fatsemi p_3 = g \fatsemi p_3$. By equality of their biproduct decomposition, we have $f = g$. □

Proposition 2.9.8

Diagrams of the following form are pullbacks in $\mathrm{Caus}[\mathcal{C}]$.

$$\begin{tikzcd}
\mathbf{X} \oplus \mathbf{Y} \arrow[r, "{[\iota_X, g \fatsemi \iota_B]}"] \arrow[d, "{[f \fatsemi \iota_A, \iota_Y]}"'] \arrow[dr, phantom, "\lrcorner", very near start] & \mathbf{X} \oplus \mathbf{B} \arrow[d, "{[f \fatsemi \iota_A, \iota_B]}"] \\
\mathbf{A} \oplus \mathbf{Y} \arrow[r, "{[\iota_A, g \fatsemi \iota_B]}"] & \mathbf{A} \oplus \mathbf{B}
\end{tikzcd} \tag{2.282}$$

Proof. The commutative diagram holds as an equation. For the universal property, suppose we have some $k : \mathbf{Z} \to \mathbf{A} \oplus \mathbf{Y}$ and $l : \mathbf{Z} \to \mathbf{X} \oplus \mathbf{B}$ such that $k \fatsemi [\iota_A, g \fatsemi \iota_B] = l \fatsemi [f \fatsemi \iota_A, \iota_B]$. Interpreting these in $\mathcal{C}$ where $\oplus$ is a biproduct, we can project out each component to give $k \fatsemi p_A = l \fatsemi p_X \fatsemi f$ and $k \fatsemi p_Y \fatsemi g = l \fatsemi p_B$. Let $h := l \fatsemi p_X \fatsemi \iota_X + k \fatsemi p_Y \fatsemi \iota_Y \in \mathcal{C}(Z, X \oplus Y)$ be our candidate witness for the pullback.

Firstly, we need to show that h is causal for $\mathbf{Z} \to \mathbf{X} \oplus \mathbf{Y}$, so consider some arbitrary state $\rho \in c_{\mathbf{Z}}$ and effect $[\pi_X, \pi_Y] \in c^*_{\mathbf{X} \oplus \mathbf{Y}}$ (i.e. $\pi_X \in c^*_{\mathbf{X}}$ and $\pi_Y \in c^*_{\mathbf{Y}}$). Firstly, observe the following from causality of k and g:

$$\begin{aligned}\rho \fatsemi k \fatsemi p_A \fatsemi ⫯_{\mathbf{A}} + \rho \fatsemi k \fatsemi p_Y \fatsemi \pi_Y &= \rho \fatsemi k \fatsemi [⫯_{\mathbf{A}}, \pi_Y] \\ &= \mathrm{id}_I \\ &= \rho \fatsemi k \fatsemi [⫯_{\mathbf{A}}, g \fatsemi ⫯_{\mathbf{B}}] \\ &= \rho \fatsemi k \fatsemi p_A \fatsemi ⫯_{\mathbf{A}} + \rho \fatsemi k \fatsemi p_Y \fatsemi g \fatsemi ⫯_{\mathbf{B}} \\ &= \rho \fatsemi k \fatsemi p_A \fatsemi ⫯_{\mathbf{A}} + \rho \fatsemi l \fatsemi p_B \fatsemi ⫯_{\mathbf{B}}\end{aligned} \tag{2.283}$$

By cancellativity of addition (APC4), $\rho \fatsemi k \fatsemi p_Y \fatsemi \pi_Y = \rho \fatsemi l \fatsemi p_B \fatsemi ⫯_{\mathbf{B}}$. We now apply this equation to reduce causality of h to causality of l.

$$\begin{aligned}\rho \fatsemi h \fatsemi [\pi_X, \pi_Y] &= \rho \fatsemi l \fatsemi p_X \fatsemi \pi_X + \rho \fatsemi k \fatsemi p_Y \fatsemi \pi_Y \\ &= \rho \fatsemi l \fatsemi p_X \fatsemi \pi_X + \rho \fatsemi l \fatsemi p_B \fatsemi ⫯_{\mathbf{B}} \\ &= \rho \fatsemi l \fatsemi [\pi_X, ⫯_{\mathbf{B}}] \\ &= \mathrm{id}_I\end{aligned} \tag{2.284}$$

We lastly need to check that k and l factorise via h.

$$\begin{aligned}h \fatsemi [f \fatsemi \iota_A, \iota_Y] &= l \fatsemi p_X \fatsemi f \fatsemi \iota_A + k \fatsemi p_Y \fatsemi \iota_Y \\ &= k \fatsemi p_A \fatsemi \iota_A + k \fatsemi p_Y \fatsemi \iota_Y \\ &= k\end{aligned} \tag{2.285}$$

$$\begin{aligned}h \fatsemi [\iota_X, g \fatsemi \iota_B] &= l \fatsemi p_X \fatsemi \iota_X + k \fatsemi p_Y \fatsemi g \fatsemi \iota_B \\ &= l \fatsemi p_X \fatsemi \iota_X + l \fatsemi p_B \fatsemi \iota_B \\ &= l\end{aligned} \tag{2.286}$$

□

As for the pullback of some $\mathbf{X} \oplus \mathbf{Y} \xrightarrow{[f \fatsemi \iota_A, g \fatsemi \iota_B]} \mathbf{A} \oplus \mathbf{B} \xleftarrow{\iota_A} \mathbf{A}$, we need not get $\mathbf{X}$ in general. If we take some $\mathbf{X} \oplus \mathbf{Y} \xleftarrow{k} \mathbf{Z} \xrightarrow{l} \mathbf{A}$ with $l \fatsemi \iota_A = k \fatsemi [f \fatsemi \iota_A, g \fatsemi \iota_B]$, we can decompose it via the biproduct in $\mathcal{C}$ to give $l = k \fatsemi p_X \fatsemi f$ and $k \fatsemi p_Y \fatsemi g = 0$. However, in general $k \fatsemi p_Y$ need not be zero itself, just something orthogonal to g in this way. This means it may not be possible to factorise k via ι_X.

This is the case when $\mathcal{C}$ is taken to be Mat[$\mathbb{R}$], but with Mat[$\mathbb{R}^+$] and CP* we find that the pullback still works because their positivity conditions dictate that the only state which gives zero when discarded is the zero state.

Proposition 2.9.9

$$\forall \rho \in \mathcal{C}\,(I, A)\,.\ \rho \fatsemi \bar{\bar{\top}}_A = 0_{I,I} \iff \rho = 0_{I,A} \tag{2.287}$$

holds in the cases where $\mathcal{C}$ is $\mathrm{Mat}[\mathbb{R}^+]$ or CP^*.

Proof. In $\mathrm{Mat}[\mathbb{R}^+]$, states are column vectors of non-negative real numbers, and the inner product with $\bar{\bar{\top}}$ will sum the elements. The only way to sum non-negative reals to obtain zero is if every element of the sum is zero.

In CP^*, states are positive semi-definite matrices, and the inner product with $\bar{\bar{\top}}$ will take the trace of a matrix, i.e. summing the eigenvalues. Positive semi-definiteness implies all eigenvalues are non-negative, therefore a zero trace implies all eigenvalues are zero, which only holds for the zero matrix. □

Proposition 2.9.10

Suppose $\mathcal{C}$ satisfies Equation 2.287. Then diagrams of the following form are pullbacks in $\mathrm{Caus}\,[\mathcal{C}]$.

$$\begin{tikzcd}
\mathbf{X} \arrow[r, "f"] \arrow[d, "\iota_X"'] \arrow[dr, phantom, "\lrcorner", very near start] & \mathbf{A} \arrow[d, "\iota_A"] \\
\mathbf{X} \oplus \mathbf{Y} \arrow[r, "{[f \fatsemi \iota_A, g \fatsemi \iota_B]}"] & \mathbf{A} \oplus \mathbf{B}
\end{tikzcd} \tag{2.288}$$

Proof. Commutativity of the diagram is straightforward. Suppose we have some $k : \mathbf{Z} \to \mathbf{A}$ and $l : \mathbf{Z} \to \mathbf{X} \oplus \mathbf{Y}$ such that $k \fatsemi \iota_A = l \fatsemi [f \fatsemi \iota_A, g \fatsemi \iota_B]$. Projecting out the biproduct in $\mathcal{C}$ gives $k = l \fatsemi p_X \fatsemi f$ and $0_{Z,B} = l \fatsemi p_Y \fatsemi g$. If we can show that $l \fatsemi p_Y$ is zero, then $l \fatsemi p_X$ is a suitable candidate for the universal property.

From causality of l and g, we observe the following:

$$\begin{aligned}
\overset{|}{\bullet}_{\mathbf{Z}} \fatsemi l \fatsemi p_X \fatsemi \underset{|}{\bullet}_{\mathbf{X}} + \overset{|}{\bullet}_{\mathbf{Z}} \fatsemi l \fatsemi p_Y \fatsemi \underset{|}{\bullet}_{\mathbf{Y}} &= \overset{|}{\bullet}_{\mathbf{Z}} \fatsemi l \fatsemi [\underset{|}{\bullet}_{\mathbf{X}}, \underset{|}{\bullet}_{\mathbf{Y}}] \\
&= \mathrm{id}_I \\
&= \overset{|}{\bullet}_{\mathbf{Z}} \fatsemi l \fatsemi [\underset{|}{\bullet}_{\mathbf{X}}, g \fatsemi \underset{|}{\bullet}_{\mathbf{B}}] \\
&= \overset{|}{\bullet}_{\mathbf{Z}} \fatsemi l \fatsemi p_X \fatsemi \underset{|}{\bullet}_{\mathbf{X}} + \overset{|}{\bullet}_{\mathbf{Z}} \fatsemi l \fatsemi p_Y \fatsemi g \fatsemi \underset{|}{\bullet}_{\mathbf{B}} \\
&= \overset{|}{\bullet}_{\mathbf{Z}} \fatsemi l \fatsemi p_X \fatsemi \underset{|}{\bullet}_{\mathbf{X}} + \overset{|}{\bullet}_{\mathbf{Z}} \fatsemi 0_{Z,B} \fatsemi \underset{|}{\bullet}_{\mathbf{B}} \\
&= \overset{|}{\bullet}_{\mathbf{Z}} \fatsemi l \fatsemi p_X \fatsemi \underset{|}{\bullet}_{\mathbf{X}} + 0
\end{aligned} \tag{2.289}$$

by cancellativity (APC4), $\overset{|}{\bullet}_{\mathbf{Z}} \fatsemi l \fatsemi p_Y \fatsemi \underset{|}{\bullet}_{\mathbf{Y}} = 0$. We can represent this as discarding a state using compact closure, so $l \fatsemi p_Y = 0$ follows by Equation 2.287 (and invertibility of the flatness scalars).

Causality of $l \fatsemi p_X : \mathbf{Z} \to \mathbf{X}$ follows from causality of l, since for any effect $\pi_X \in c^*_{\mathbf{X}}$:

$$\begin{aligned} l \fatsemi p_X \fatsemi \pi_X &= l \fatsemi p_X \fatsemi \iota_X \fatsemi [\pi_X, \top_{\mathbf{Y}}] \qquad (2.290) \\ &= l \fatsemi [\pi_X, \top_{\mathbf{Y}}] \in c^*_{\mathbf{Z}} \end{aligned}$$

□

Despite the fact that $\mathbf{I}$ is not terminal, we have recovered the rest of the nice starting properties of an effectus. This can be pushed further in the same way to recreate weaker forms of several other useful properties in the category of partial maps (in each case the proofs similarly match those from [25] with minimal adjustment):

- $\oplus$ still forms coproducts with copairings matching those in $\mathrm{Caus}[\mathcal{C}]/\mathcal{C}$, though the lack of terminal object means we don't have a unit/zero object;
- Diagrams of the form of Proposition 2.9.8 in the category of partial maps are also pullbacks;
- We can similarly define partial projections using discarding instead of the terminal morphisms, which means we lose the naturality (since not all higher-order causal morphisms are discard preserving) but we can still pull back total maps along them.

One curious difference between this construction and effectuses comes from how we identify which partial maps are total from their interaction with effects. In an effectus, the total maps can be obtained from the partial maps as those that preserve the unique total effect. One might imagine that a higher-order setting would characterise total maps by those that transform *all* total effects to (unspecified) total effects, but it turns out that preserving totality for *any one* effect is sufficient to preserve totality on all.

Proposition 2.9.11

A partial map $f : \mathbf{A} \to \mathbf{B} \oplus \mathbf{I}$ is total ($\exists f' : \mathbf{A} \to \mathbf{B}.\ f = f' \fatsemi \iota_B$) iff there exists some effect $\pi_B : \mathbf{B} \to \mathbf{I}$ such that $f \fatsemi [\pi_B \fatsemi \iota_1, \iota_2] : \mathbf{A} \to \mathbf{I} \oplus \mathbf{I}$ is total.

Proof. The only if ($\Longrightarrow$) direction is trivial by composition, so we will focus on the if direction ($\Longleftarrow$). Suppose there exists some $\pi_B \in c^*_{\mathbf{B}}$ and $\pi_A \in c^*_{\mathbf{A}}$ such that

$f \fatsemi [\pi_B \fatsemi \iota_1, \iota_2] = \pi_A \fatsemi \iota_1$. Taking the pullback of $\mathbf{B} \oplus \mathbf{I} \xrightarrow{[\pi_B \fatsemi \iota_1, \iota_2]} \mathbf{I} \oplus \mathbf{I} \xleftarrow{\iota_1} \mathbf{I}$ using Proposition 2.9.10, the universal property means π_A and, more importantly, f both factorise via some $f' : \mathbf{A} \to \mathbf{B}$, i.e. $f = f' \fatsemi \iota_B$ is total. □

The explanation for this highlights a small issue with the monadic design for higher-order settings: if we view a partial map $\mathbf{A} \to \mathbf{B} \oplus \mathbf{I}$ as a combination of a successful branch and a failure, the probability of success/failure (and therefore whether it is total) is fixed only by the interaction with $\mathbf{A}$ and is independent of any interaction with $\mathbf{B}$! In other words, this construction is not time-symmetric. For example, postselecting on one state of $\mathbf{2}$ can be expressed as a partial map $\mathbf{2} \to \mathbf{I} \oplus \mathbf{I}$ (e.g. $\mathrm{id}_2 = [\iota_1, \iota_2]$ or $[\iota_2, \iota_1]$) but not as a partial map $\mathbf{I} \to \mathbf{2}^* \oplus \mathbf{I}$ since the probability of success depends on the state being postselected.

To define a time-symmetric version of testable partial maps $\mathbf{A} \to \mathbf{B}$, we first use $\ast$-autonomy to encode the space of total maps as effects $\mathbf{A} \otimes \mathbf{B}^* \to \mathbf{I}$, and then apply the maybe monad to give $\mathbf{A} \otimes \mathbf{B}^* \to \mathbf{I} \oplus \mathbf{I} = \mathbf{2}$, allowing the success/failure probability to depend on the entire context. If we reapply $\ast$-autonomy, this is equivalent to $\mathbf{A} \to \mathbf{B} \parr \mathbf{2}$, i.e. the space of binary tests. Crucially, this resolves the differences between descriptive and testable partiality, since any descriptive partial map (i.e. a morphism of $\mathcal{C}$) is a branch of a binary test up to some invertible scalar; modulo scalars they are the same thing!

To fully recover a setting of partial maps, we need to be able to combine two outputs $\mathbf{2} \parr \mathbf{2}$ into one, which is possible since $\mathbf{2}$ is first-order and so the outputs exert no influence on each other and $\otimes$ distributes over $\oplus$ (see Remark 2.7.10).

$$\mathbf{2} \parr \mathbf{2} = \mathbf{2} \otimes \mathbf{2} \cong (\mathbf{2} \otimes \mathbf{I}) \oplus (\mathbf{2} \otimes \mathbf{I}) \cong \mathbf{I} \oplus \mathbf{I} \oplus \mathbf{I} \oplus \mathbf{I} \xrightarrow{[\iota_1, \iota_2, \iota_2, \iota_2]} \mathbf{I} \oplus \mathbf{I} = \mathbf{2} \tag{2.291}$$

2.9.4 The Probabilistic Orthogonality

Looking outside of effectus theory, another route to a theory of partial maps is to revisit the definition of the Caus $[-]$ construction and adapt it to focus on processes that are realisable with *some* probability in any context rather than those with exactly probability 1. Since this was baked into the dual sets in Definition 2.3.6, we consider a probabilistic version of it, which will still be a focussed orthogonality [72].

Definition 2.9.12

Given $c \subseteq \mathcal{C}(I, A)$, the *sub-dual set* is

$$c^{\leq} := \left\{ \pi \in \mathcal{C}(A, I) \;\middle|\; \forall \rho \in c.\; \begin{array}{c} \boxed{\pi} \\ | \\ \boxed{\rho} \end{array} \leq \mathrm{id}_I \right\} \tag{2.292}$$

where $\leq$ refers to the preorder : $f \leq g \overset{\text{def}}{\iff} \exists h.\ f + h = g$.

Proposition 2.9.13

If $c = c^{\leq\leq} \subseteq \mathcal{C}(I, A)$, then:

1. c is non-empty;
2. c is closed under convex combinations;
3. c is downward closed under the preorder $\leq$.

Proof.

1. $0_{I,A} \in c$ as every inner product will be $0 \leq \mathrm{id}_I$.
2. Given $\sum_i \alpha_i \cdot \rho_i$ with $\{\rho_i\}_i \subseteq c$, $\sum_i \alpha_i = \mathrm{id}_I$, and some $\pi \in c^{\leq}$:

$$\begin{aligned} \left(\sum_i \alpha_i \cdot \rho_i\right) ⨟ \pi &= \sum_i \alpha_i \cdot (\rho_i + \pi) \\ &\leq \sum_i \alpha_i \cdot \mathrm{id}_I \\ &= \mathrm{id}_I \end{aligned} \tag{2.293}$$

   so $\sum_i \alpha_i \cdot \rho_i \in c^{\leq\leq} = c$.

3. If $\rho \leq \upsilon \in c$, then for any $\pi \in c^{\leq}$ we have $\rho ⨟ \pi \leq \upsilon ⨟ \pi \leq \mathrm{id}_I$. □

These properties correspond precisely to those satisfied by Probabilistic Coherence Spaces in Definition 2.6.5. In fact, a PCS is more commonly defined to be precisely some set $c^{\leq\leq}$ in $\mathrm{Mat}[\mathbb{R}^+]$ [17]. Whether the conditions of Proposition 2.9.13 *exactly* characterise sets of the form $c^{\leq\leq}$ for any other additive precausal category is currently unknown.

Generalising PCS's to arbitrary additive precausal categories may not always give something practical anyway. For example, if $\mathcal{C} = \mathrm{Mat}[\mathbb{R}]$, all negatives exist so the only down-closed sets of states are full homsets for a given object $\mathrm{Mat}[\mathbb{R}](I, A)$.

It is also hard to relate a category of such spaces to $\mathrm{Caus}[\mathcal{C}]$. By taking the equivalence classes of $\mathrm{Caus}[\mathcal{C}](\mathbf{A}, \mathbf{2})$ under the action of the projection $p_1 \in \mathcal{C}(I \oplus I, I)$

gives a set of morphisms that is closed under convex combination and down-closed under $\leq$. However, not every convex-closed, down-closed set is formed in this way. Furthermore, whilst the category of PCS's is defined in this way, it is common for operational theories from the quantum literature to prefer defining partial operations to have a concrete relationship to the total/causal operations in order to guarantee their existence in some test [23, 15]. So whilst applying double-glueing and restricting to spaces closed under $(-)^{\leq\leq}$ is a neat way to get enough flexibility to examine probabilistic operations, it is likely better to follow a construction that builds on top of $\mathrm{Caus}\left[\mathcal{C}\right]$.

Chapter 3

Logical Characterisation of $\mathrm{Caus}\left[\mathcal{C}\right]$

By now, we have a good grasp of the processes and types available in the category $\mathrm{Caus}\left[\mathcal{C}\right]$, so it is time we address causal consistency directly. The goal for this next chapter is to formalise causal consistency of string diagrams within the language of $\mathrm{Caus}\left[\mathcal{C}\right]$ and find some simple way to verify causal consistency. This is achieved by finding formal logics where the proofs can be mapped to casually consistent string diagrams. The ideal solution to this would be to find one such logic whose proofs can also generate *every* causally consistent diagram.

Each string diagram we draw is fundamentally a morphism consisting of identity, cup, and cap morphisms between typed black boxes. We can either consider specific morphisms, i.e. where each wire is interpreted with a fixed object, or view them collectively as an *extranatural transformation* (see Definition 2.1.10). We will devise definitions for causal consistency that apply in each case, so we can characterise both the behaviour shared by all interpretations into local objects by the existence of extranatural transformations, and any special case behaviour that arises for particular interpretations. For example, we know from the First-Order Theorem that $\mathrm{id}_{A^* \otimes A}$ is not causal for $\mathbf{A} \multimap \mathbf{A} \to \mathbf{A}^* < \mathbf{A}$ for any object $\mathbf{A}$, but it is causal in the special case where $\mathbf{A}$ is first-order. In fact, we will show that the degeneracy of first-order systems is the *only* way to introduce special case behaviour and therefore, beyond identification of first-order systems, causal consistency is completely independent of the interpretation.

To do this, we design *causal logic* with a new proof-net criterion to exactly match the semantics of causal consistency. Similar to the Sum of Orders Theorem, given a string diagram, we can break down both the types of the black boxes and the types of the wires into unions of graph types - in particular, the collection of graphs will correspond to the switching graphs of the proof-structure - and use the Graph Compatibility Theorem to deduce that each switching graph needs to be acyclic.

Since this logic will not incorporate any knowledge of the base category $\mathcal{C}$ or the interpretation of the local systems (beyond marking first-order systems), the tight correspondence with causal consistency immediate tells us that causal consistency is both theory- and interpretation-independent.

We assume the reader has a basic understanding of concepts in formal logic, though the first section will give a high-level overview of the specific logics we will refer to throughout this chapter, demonstrating how one constructs proofs in each of them. Novel content resumes in Section 3.2 which builds up our formal definitions of causal consistency and gives examples of how to relate proofs of some existing logics as causally consistent string diagrams.

Section 3.3 is where the bulk of the interesting results lie: we introduce causal proof-nets and prove that they precisely characterise causal consistency. We then compare it to existing logics in Section 3.4, focussing heavily on the close connection to pomset [100, 102]. Up to this point, we will have focussed on causal logic with only the connectives $\{\otimes, <, \parr\}$, so we wrap up with some discussion on extending causal logic to also handle unions and intersections, primitive graph types, and additives.

3.1 Background: MLL, BV, and pomset

Formal logics aim to provide a canonical syntax for mathematical reasoning, where a goal statement is shown to be provable in the logic by giving either a derivation from simple elementary rules or some other proof object which can be verified by some simple consistency condition. The choices of what grammar to build terms from, how one builds statements from terms (e.g. one- or two-sided, collections of terms as sets/multisets/ordered lists, etc.), and which rules to permit or consistency condition to enforce give the logic its flavour, determining the structure of proofs and what semantic models exist for it.

Linear logic [55] exposes the hidden assumption of propositional logic that everything is copyable: a proof of the implication $F \Rightarrow G$ may use its premise F multiple (or even zero) times to derive the conclusion G. The rules are “resource-aware”, ensuring that each premise is used precisely once, in line with the reading of linear implication $F \multimap G$ as “one instance of F can generate an instance of G”. Sequents use multisets Γ, Δ of terms to explicitly track the quantities of any duplicate terms. The operators of linear logic are typically grouped into the following classes:

- Duality $(-)^{*}$[11] as the correspondent of negation, which is commonly interpreted

[11] Duality is more commonly written as $(-)^{\perp}$ since this is equivalent to $(-) \multimap \perp$. We choose to use $(-)^{*}$ for a closer similarity to the duality operator in Caus $[\mathcal{C}]$.

in models as the duality between producing or consuming a use of a term. Syntactic de Morgan equations can always push duals inside formulae to yield a negation normal form (where duals are only applied on atoms).

- Multiplicatives $\{\otimes, \parr, \multimap\}$ introduce multiple uses of terms in parallel, possibly with some connection or relationship between them.
- Additives $\{\times, \oplus\}$ capture a single-use choice between two terms, distinguished as external choice (both are provided, and the choice selected at consumption) versus internal choice (only one is provided, so a consumer must be able to handle either).
- Exponentials $\{!, ?\}$ annotate terms to reintroduce resources that can be used multiple times.
- Units $\{1, \bot, 0, \top\}$ for the multiplicative and additive operators represent trivial or degenerate cases. Many extensions of linear logic will unify some of these units.

We retain the usual distinction between classical and intuitionistic variants of the logic (whether negation/duality is involutive $F^{**} \multimap F$). For this presentation, we will work with the classical logics that will be more relevant to the $\mathrm{Caus}\,[-]$ construction.

3.1.1 Sequent Calculus for MLL

Multiplicative Linear Logic (MLL) is the fragment just concerning the multiplicative operators and their units. In the classical variant, linear implication $F \multimap G$ is syntactically identified with $F^* \parr G$. The standard presentation is as a sequent calculus, inductively defined by the following rules:

$$\text{Ax}\,\frac{}{\vdash A^*, A} \qquad\qquad \text{Cut}\,\frac{\vdash \Gamma, F \qquad \vdash F^*, \Delta}{\vdash \Gamma, \Delta}$$

$$\otimes\,\frac{\vdash \Gamma, F \qquad \vdash \Delta, G}{\vdash \Gamma, \Delta, F \otimes G} \qquad\qquad \parr\,\frac{\vdash \Gamma, F, G}{\vdash \Gamma, F \parr G}$$

$$1\,\frac{}{\vdash 1} \qquad\qquad \bot\,\frac{\vdash \Gamma}{\vdash \Gamma, \bot}$$

There exists a cut-elimination procedure which, given a derivation of a sequent using the Cut rule, yields a derivation with the same conclusion that does not use Cut (a *cut-free* derivation).

Example 3.1.1

Observe that the following derivation tree yields one of the properties needed for associativity of $\otimes$:

$$\begin{prooftree}
\AxiomC{}
\LeftLabel{Ax}
\UnaryInfC{$\vdash A^*, A$}
\AxiomC{}
\LeftLabel{Ax}
\UnaryInfC{$\vdash B^*, B$}
\LeftLabel{$\otimes$}
\BinaryInfC{$\vdash A^*, B^*, A \otimes B$}
\AxiomC{}
\LeftLabel{Ax}
\UnaryInfC{$\vdash C^*, C$}
\LeftLabel{$\otimes$}
\BinaryInfC{$\vdash A^*, B^*, C^*, (A \otimes B) \otimes C$}
\LeftLabel{$\parr$}
\UnaryInfC{$\vdash A^*, B^* \parr C^*, (A \otimes B) \otimes C$}
\LeftLabel{$\parr$}
\UnaryInfC{$\vdash A^* \parr (B^* \parr C^*), (A \otimes B) \otimes C$}
\LeftLabel{$\parr$}
\UnaryInfC{$\vdash A \otimes (B \otimes C) \multimap (A \otimes B) \otimes C$}
\end{prooftree}$$

However, if we search for a cut-free derivation of $A \multimap A \otimes \bot$ (i.e. using the unit of $\parr$ with $\otimes$) the syntax-directed nature of the sequent calculus forces any solution to finish with the following rules:

$$\begin{prooftree}
\AxiomC{}
\LeftLabel{Ax}
\UnaryInfC{$\vdash A^*, A$}
\AxiomC{$\vdots$}
\UnaryInfC{$\vdash$}
\LeftLabel{$\bot$}
\UnaryInfC{$\vdash \bot$}
\LeftLabel{$\otimes$}
\BinaryInfC{$\vdash A^*, A \otimes \bot$}
\LeftLabel{$\parr$}
\UnaryInfC{$\vdash A \multimap A \otimes \bot$}
\end{prooftree}$$

But it is obvious from the rules that no cut-free proof exists for the empty sequent, so we conclude $\nvdash A \multimap A \otimes \bot$.

Some settings will consider adding additional “mixing” rules to the logic, giving rise to MLL+Mix [2, 51]:

$$\begin{prooftree}
\AxiomC{}
\LeftLabel{Mix1}
\UnaryInfC{$\vdash$}
\end{prooftree}
\qquad\qquad
\begin{prooftree}
\AxiomC{$\vdash \Gamma$}
\AxiomC{$\vdash \Delta$}
\LeftLabel{Mix2}
\BinaryInfC{$\vdash \Gamma, \Delta$}
\end{prooftree}$$

These are equivalent to adding axioms for $\vdash \bot, \bot$ (which gives $\vdash 1 \multimap \bot$) and $\vdash 1, 1$ (giving $\vdash \bot \multimap 1$) respectively, making the two units interchangeable [28]. Mix2 is also equivalent to the binary case $\vdash A^*, B^*, A, B$, which can be summarised as $\otimes$ “embedding into” $\parr$ as $\vdash A \otimes B \multimap A \parr B$.

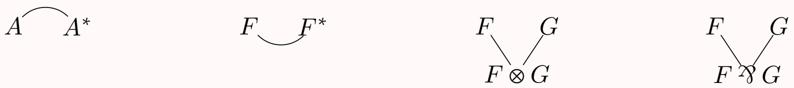
3.1.2 Proof-Nets for MLL

The primary alternative to sequent derivations as a presentation of linear logic proofs is *proof-nets* [55, 40]. These consist of graphical objects annotated with terms where correctness of the proof is determined by some global property of the graph, in contrast with the local correctness of each rule in a derivation tree. One can translate proofs back and forth, with each derivation tree yielding a unique proof-net but each proof-net matching multiple derivation trees.

To specify a proof-net framework, we first define *proof-structures* as a particular grammar of graphical objects under consideration which are composites of elementary graphs called *links*. For simplicity in this introduction to proof-nets, we will not be including units.

Definition 3.1.2: MLL links [55]

Links for MLL are graphs of the following forms, relating some premises to some conclusions:



- Axiom links with no premises and a pair of conclusions A, A^*.
- Cut links with a pair of premises F, F^* and no conclusions.
- Tensor links with a pair of premises F, G and conclusion $F \otimes G$.
- Par links with a pair of premises F, G and conclusion $F \parr G$.

Definition 3.1.3: MLL proof-structure [55]

A *proof-structure* is a graph composed of links such that every occurrence of a formula (i.e. vertex) is a conclusion of exactly one link and a premise of at most one link. The *conclusions* of the proof-structure are the formulae that are not the premises of any link.

Proof-nets are those proof-structures that correspond to "correct" proofs in the sense that they satisfy a particular correctness criterion. The Danos-Regnier criterion [40] derives a collection of variations on the proof-structure's graph, called *switching graphs*, and asks each to be acyclic.

Definition 3.1.4: MLL switchings [40]

A *switching* for a proof-structure P is a choice of one edge from each par link. The corresponding *switching graph* is a copy of P in which we omit the edges from par links that were not selected.

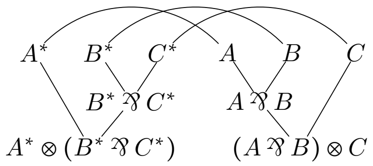
Definition 3.1.5: MLL proof-net [40]

A proof-structure is a MLL *proof-net* when every switching graph is connected and acyclic. It is a MLL+Mix proof-net when every switching graph is acyclic but need not be connected.

Proof-nets also admit cut-elimination. The syntax-directed nature of the links means that cut-free proof-structures will resemble the syntax tree of the conclusion with some choice of matching for the axiom links.

Example 3.1.6

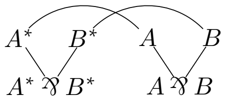
Consider the following proof-structure:



(3.1)

By looking at the switching graph where we take the paths to C^* and A, we can construct a cycle and conclude this is not a valid proof-net.

On the other hand, the following proof-structure representing the mixing rule is a proof-net for MLL+Mix but not MLL as every switching graph is disconnected.



(3.2)

There are multiple, equivalent notions of proof-nets for MLL or MLL+Mix, each of which can provide a unique intuition for the logics. For Girard's long-trip condition [55], switchings define a route for traversing over the proof-structure and requires that the entire proof is traversed. Topological criteria [86] can be very visually intuitive. Handsome proof-nets [103] even forgo switchings, instead building proof-structures as R&B-graphs and obtaining a more canonical representation of proofs in which associativity of operators is a simple equality.

3.1.3 BV, MAV, and pomset Logic

Many settings will interpret $\otimes$ and $\parr$ as a form of parallel composition, but may also carry a separate notion of time or sequential composition. BV [58] and pomset [100, 102] are two candidate solutions to this, where both extend MLL+Mix with a self-dual non-commutative operator $<$, reading $F < G$ as "F before G". They were originally speculated to express the same logic, but pomset contains strictly more theorems than BV [94].

The deductive system of BV employs *deep inference* - a way to formulate a logic where inference rules can be applied at any level of nesting within expressions, in contrast to the sequent calculus rules we saw for MLL which all apply at the top level of syntax trees. This is crucial to BV, and placing any bound on the nesting level of applying rules strictly reduces the number of theorems [112]. Deep inference rules are expressed with respect to term contexts, though the equivalent formalism of *open deduction* [59] instead allows us to combine the rules themselves via the connectives of the logic, giving a neater presentation for larger proofs as we will see in Example 3.1.7.

$$I\downarrow\frac{}{I} \qquad\qquad \mathrm{ai}\downarrow\frac{I}{A^{*}\parr A}$$

$$\mathrm{q}\downarrow\frac{(F\parr G) < (H\parr K)}{(F < H)\parr(G < K)} \qquad\qquad \mathrm{s}\,\frac{(F\parr G)\otimes H}{F\parr(G\otimes H)}$$

Instead of commutativity, associativity, and unitality (I is a unit for all of $\{\otimes, < \parr\}$) being properties which can be proved by the rules, they are permitted via a syntactic equivalence relation $\equiv$ which is a congruence so can similarly be applied at any level within terms. A deep inference version of the Cut rule also exists, along with a cut-elimination result [58].

Example 3.1.7

The following example BV proof of $(A\otimes B) < C \multimap A < (B\parr C)$ demonstrates the nested application of rules via open deduction. Dashed lines are applications of the syntactic equivalence.

$$\begin{array}{c}
I\downarrow\dfrac{}{I} \\ \hdashline
\mathrm{q}\downarrow\dfrac{\left(\mathrm{ai}\downarrow\dfrac{I}{A^{*}\parr A}\right) < \left(\begin{array}{c}
\mathrm{q}\downarrow\dfrac{\left(\begin{array}{c}\mathrm{ai}\downarrow\dfrac{I}{B^{*}\parr B} \\ \hdashline B^{*}\parr B\parr I\end{array}\right) < \left(\mathrm{ai}\downarrow\dfrac{I}{C^{*}\parr C}\right)}{((B^{*}\parr B) < C^{*})\parr(I < C)} \\ \hdashline
\left(\mathrm{q}\downarrow\dfrac{(B^{*}\parr B) < (C^{*}\parr I)}{(B^{*} < C^{*})\parr(B < I)}\right)\parr C \\ \hdashline
(B^{*} < C^{*})\parr B\parr C
\end{array}\right)}{\begin{array}{c}
\left(\left(\begin{array}{c} A^{*} < B^{*} \\ \hdashline \mathrm{q}\downarrow\dfrac{(A^{*}\parr I) < (I\parr B^{*})}{(A^{*} < I)\parr(I < B^{*})}\end{array}\right) < C^{*}\right)\parr(A < (B\parr C)) \\ \hdashline
((A^{*}\parr B^{*}) < C^{*})\parr(A < (B\parr C))
\end{array}}
\end{array}$$

The combination of sequential and parallel composition is not enough to recover all partial orders or graphs over the atoms. A recent extension called GV adds new operators based on prime directed and undirected graphs, from which we can obtain every mixed graph from its modular decomposition. We refer the reader to Acclavio et al. [3] for further details.

MAV [67] is another related logic, extending BV with additives $\{\times, \oplus\}$. This adds the relevant rules from MALL (the extension of MLL with additives) and the additional *medial* (m $\downarrow$) rule to ensure the additives integrate nicely with seq [67].

$$\mathrm{t}\downarrow \frac{I}{I \times I} \qquad \mathrm{l}\downarrow \frac{F}{F \oplus G} \qquad \mathrm{r}\downarrow \frac{G}{F \oplus G}$$

$$\mathrm{e}\downarrow \frac{(F \parr H) \times (G \parr H)}{(F \times G) \parr H} \qquad \mathrm{m}\downarrow \frac{(F \times G) < (H \times K)}{(F < H) \times (G < K)}$$

We finally turn to pomset logic, which is built on R&B-graphs, i.e. graphs with red and blue edges. We still build proof-structures from links, with the correctness of a proof-net becoming the absence of an *alternating elementary cycle*. The alternating colour requirement restricts the paths through each link in the same way as switchings.

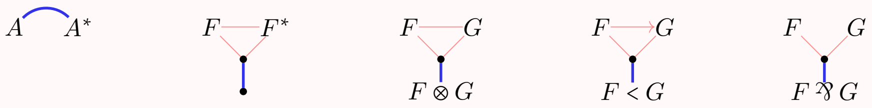
Definition 3.1.8: R&B-graphs [102]

A R&B-*graph* $G = (V, R, B)$ is an edge-bi-coloured graph where:

- V is the set of vertices;
- $R \subseteq V \times V$ is the set of (directed and undirected) red edges, expressed as an irreflexive relation;
- $B \subseteq V \times V$ is the set of (undirected) blue edges, expressed as an irreflexive, symmetric relation forming a perfect matching of the vertices $\forall x \in V.\ \exists! y \in V.\ (x, y) \in B$.

Definition 3.1.9: R&B proof-structure [102, Definition 3]

A R&B *proof-structure* is a R&B-graph composed of the following kinds of links:



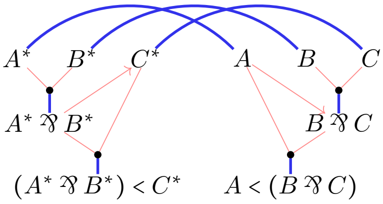
The conclusions of the proof-structure may also be connected by directed red edges according to a series-parallel partial order.

Definition 3.1.10: Pomset proof-net [102, Definition 4]

A pomset *proof-net* is a R&B proof-structure containing no alternating elementary cycle (a cycle with no repeated vertices except for the first and last, where successive edges alternate colours).

Example 3.1.11

The following is a valid proof net of $(A \otimes B) < C \multimap A < (B \parr C)$.



(3.3)

The $\parr$ links prevent any alternating path from using the B axiom link alongside either of those of A and C. The only other option for a cycle would be to use both the A and C axiom links, but the directed arcs of the $<$ links both lead from A to C preventing us from building a cycle.

On the other hand, if we were to switch out the top right $\parr$ link for a $\otimes$ link (giving overall conclusion $(A \otimes B) < C \multimap A < (B \otimes C)$), we could construct an alternating elementary cycle $[B, B^*, \cdot, A^* \parr B^*, C^*, C, B]$.

The additional edges beyond the conclusions elevate the multiset structure of MLL sequents to partially-ordered multisets, hence the name of the logic. The series-parallel restriction means we can combine the conclusions using the $\parr$ and $<$ links to form an equivalent proof-structure with a single conclusion, reminiscent of how $\parr$ naturally combines conclusions in MLL.

From the construction of the links, any path from one link to another will change colour at the transition: on each vertex labelled with a formula, the unique incident blue edge belongs to the link where this vertex is a conclusion and any red edges (if they exist) belong to the link where it is a premise. Therefore, the alternating colour requirement only constrains the possible paths within each individual link; notably, there is no alternating path between the premises of a $\parr$ link, and only one direction between the premises of a $<$ link. On the $<$-free fragment, these match the connectivities induced by switchings on MLL proof-structures, so pomset is a conservative extension of MLL+Mix because we can faithfully translate between cyclic switchings in an MLL proof-structure and alternating cycles in a R&B proof-structure.

Deciding whether a given R&B proof-structure is a pomset proof-net is coNP-complete [94] - it is easy to verify that a given alternating cycle witnesses incorrectness, but searching for such cycles is hard. Determining whether a proof-net exists for a given conclusion is even harder at Σ_2^p-complete ($\Sigma_2^p = \mathrm{NP}^{\mathrm{NP}}$ is NP extended with an oracle for NP). On the other hand, verifying BV proofs takes time linear in the size of the proof, and provability of a formula is NP-complete [76]. Pomset is even hard compared to other proof-net formulations, where proof-nets can be verified for MLL or MLL+Mix in linear time [90, 92]. Knowing these complexities will be useful for deducing the complexity of deciding causal consistency through its relationship to pomset (see Corollary 3.3.16).

3.1.4 Categorical Semantics

For any logic with a Cut rule and an algorithm for cut-elimination, we can study categories $\mathcal{C}$ of *modular proof invariants* for the logic: each formula F is assigned a denotation $[\![F]\!] \in \mathrm{Ob}(\mathcal{C})$, and to each proof $\pi : F \vdash G$ a morphism $[\![\pi]\!] : [\![F]\!] \to [\![G]\!]$ such that:

- If cut-elimination on π outputs some proof π' then $[\![\pi]\!] = [\![\pi']\!]$.
- Applying the Cut rule to $\pi : F \vdash G$ and $\pi' : G \vdash H$ gives a proof $\pi'' : F \vdash H$ such that $[\![\pi'']\!] = [\![\pi]\!] \fatsemi [\![\pi']\!]$.

Each operator in the grammar gives rise to come categorical structure based on the rules associated to the operator, such as a monoidal structure for $\otimes$. Canonical categories for the logic can be chosen as *free categories* with the required structure, though we can view any category with the same structure as a model of the logic where each proof defines some functorial way to combine morphisms in the category. The classic Curry-Howard-Lambek correspondence draws an association between intuitionistic propositional logic and cartesian closed categories, with this name given to any similar correspondence between a formal logic, a type theory for programming languages, and a kind of categorical structure.

For a deep exploration on categorical semantics for linear logic, we recommend the survey by Melliès [85]. For a brief summary, models of MLL are given by $*$-autonomous categories with the additional mixing rules relating ISOMIX categories to MLL+Mix. Then MALL just adds cartesian products and coproducts for the additives $\{\times, \oplus\}$.

For BV, Blute, Panangaden & Slavnov's BV-categories [17] combine linearly-distributive categories and a ⊗-symmetric duoidal structure [4] (minus some of the coherence equations) with isomorphic units. The examples shown include both the categories of regular (non-deterministic) and probabilistic coherence spaces. As for MAV, we again just add cartesian products and coproducts for the additives, and the additional medial rule is guaranteed to hold by the fact that the monoidal structures will always form duoidal structures with the products and coproducts [104]. A more recent study of categorical semantics for BV and MAV has made the relations to duoidal categories more concrete [6].

When it comes to pomset logic, its introduction was as a logic that precisely captured the structure of coherence spaces - that is, there exists some denotational semantics for the proof-nets in the category and, moreover, there is a 4-token coherence space Z such that, under the interpretation mapping every atomic formula to Z, a proof-structure is a proof-net iff its interpretation is a clique in the space associated with the conclusion [102]. This statement of faithfulness is not necessarily true for arbitrary interpretations, and there is yet to be any wider study of categorical semantics for pomset logic that characterise it by some kind of generic categorical structure.

3.2 Causal Consistency

This section will introduce *causal consistency* as the property of interest within causal categories that we will study as a model of different logics and aim to provide a complete characterisation for it. By relating it to macroscopic properties of the categories in terms of extranatural transformation, this characterisation will, in turn, give a logical account of the compositional structure of Caus $[\mathcal{C}]$ in the vain of the Curry-Howard-Lambek correspondence.

3.2.1 Causal String Diagrams for Black Boxes

Informally, causal consistency is a property of a collection of types in Caus $[\mathcal{C}]$ that says any valid instances of those types can be composed in a string diagram to give a closed, causal scenario. Here, we will formalise causal consistency via formulae that represent the types of the black boxes in the given diagram.

Recall Theorem 2.3.20, where Kissinger and Uijlen proved that natural transformations in Caus $[\mathcal{C}]$ form a model of ISOMIX logic. For example, Equations 2.58 and 2.59 capture the mixing and linear distributivity between ⊗ and ⅋. We can view these as statements of causal consistency, e.g. the closed diagram formed by any state of $\mathbf{A} \otimes \mathbf{B}$ and any effect of $\mathbf{A} \parr \mathbf{B}$ evaluates to the unit scalar. It is implicit in this description that we are wiring up the $\mathbf{A}$ part of the state to the $\mathbf{A}$ part of the effect, and likewise for the $\mathbf{B}$'s. We would like this scenario to be captured by the formula $(A \otimes B) \multimap (A \parr B)$, i.e. the one-sided description of the mixing property in linear logic.

To simplify how we can generalise this to give formulae for arbitrary closed diagrams, suppose we have one with just a single black box. Morphisms in our example categories are linear maps and can be likened to a tensor network, where the process of wiring up inputs and outputs of this box will act like contracting the indices of the tensor until it is reduced to a scalar - we denote the morphism of cups, caps, and identities that performs the wiring between the black boxes the *contraction morphism*, the same as we did when discussing local graph types (Definition 2.8.2). To map this to a formula, we attribute variable names to indices so that each name appears twice, implicitly specifying which pairs of indices to contract. These are combined with connectives corresponding to the monoidal products in $\mathrm{Caus}[\mathcal{C}]$ and an interpretation function maps this information to concrete objects in $\mathrm{Caus}[\mathcal{C}]$. Given the special behaviour of first-order and first-order dual objects, we will allow our formulae to specify when variables should be mapped to first-order objects and only consider interpretations that do so.

Definition 3.2.1: Causal formulae

We define causal formulae (in negation normal form) by the grammar

$$F, G ::= A \mid A^* \mid A^1 \mid A^{1*} \mid I \mid F \otimes G \mid F < G \mid F \parr G \tag{3.4}$$

The *negation* F^* of a formula is defined inductively:

$$\begin{aligned} A^{**} &= A \\ \left(A^{1*}\right)^* &= A^1 \\ I^* &= I \\ (F \otimes G)^* &= F^* \parr G^* \\ (F < G)^* &= F^* < G^* \\ (F \parr G)^* &= F^* \otimes G^* \end{aligned} \tag{3.5}$$

A formula F is *balanced* if each atom A/A^1 in F appears exactly once in each of positive and negative form.

Definition 3.2.2: Interpretations

An *interpretation* $\Phi : \mathrm{Var} \to \mathrm{Ob}\left(\mathrm{Caus}\left[\mathcal{C}\right]\right)$ *FO-respects*[12] F when, for any atom $A \in \mathrm{Var}$, if A appears in F as A^1 then $\Phi\left(A\right)$ is first-order. Φ *strongly FO-respects* F when the converse additionally holds - $\Phi\left(A\right)$ is first-order only if A^1 appears in F - and no $\Phi\left(A\right)$ is first-order dual. An interpretation can be extended to formulae inductively:

$$\begin{aligned}
\Phi\left(A^*\right) &= \Phi\left(A\right)^* \\
\Phi\left(A^1\right) &= \Phi\left(A\right) \\
\Phi\left(A^{1*}\right) &= \Phi\left(A\right)^* \\
\Phi\left(I\right) &= \mathbf{I} \\
\Phi\left(F \otimes G\right) &= \Phi\left(F\right) \otimes \Phi\left(G\right) \\
\Phi\left(F < G\right) &= \Phi\left(F\right) < \Phi\left(G\right) \\
\Phi\left(F \parr G\right) &= \Phi\left(F\right) \parr \Phi\left(G\right)
\end{aligned} \tag{3.6}$$

Definition 3.2.3: Causal consistency

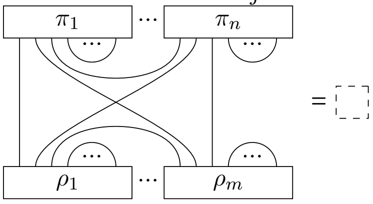
Given a balanced formula F and an interpretation Φ which FO-respects F, the *contraction morphism of* F is the morphism $\epsilon_F^\Phi \in \mathcal{C}\left(\mathcal{U}\left(\Phi\left(F\right)^*\right), I\right)$ formed by applying a cap $\epsilon_{\Phi(A)} : \Phi\left(A^*\right)^* \otimes \Phi\left(A\right)^* \to \mathbf{I}$ between the components for each variable (regular and first-order). We say that F *is causally consistent for* $\mathcal{C}$ *under* Φ ($\Vdash_{\mathcal{C}}^{\Phi} F$) when ϵ_F^Φ is causal $\Phi\left(F\right)^* \to \mathbf{I}$, i.e. $\epsilon_F^\Phi \in c_{\Phi(F)}$.

The above definition chooses the perspective that $\Phi\left(F\right)$ is describing the causal structure of the contraction morphism, viewing the black box as an effect with only inputs. This is a design choice which makes the formulae of causal consistency agree with linear logic, at the expense of dualising to $\Phi\left(F\right)^*$ when we want to talk about the causal structure within the black box.

We can safely generalise this definition to talk about causal consistency of diagrams with multiple black boxes as both states and effects using multi-sequents. In this case, $G_1, \ldots, G_m \Vdash_{\mathcal{C}}^{\Phi} F_1, \ldots, F_n$ asserts that the combination of identities, cups, and caps in between the boxes is a causal morphism in $\mathrm{Caus}\left[\mathcal{C}\right]\left(\bigotimes_j \Phi(G_j), \bigparr_i \Phi(F_i)\right)$.

[12] This term refers to the interpretation sending the designated variables to first-order objects in $\mathrm{Caus}\left[\mathcal{C}\right]$ - those representing elementary data - and has no intended relation to first-order logic.

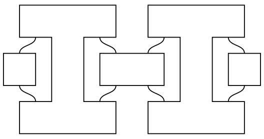
$$\forall \left\{\rho_j : \mathbf{I} \to \Phi(G_j)\right\}_j, \left\{\pi_i : \Phi(F_i) \to \mathbf{I}\right\}_i .$$



(3.7)

Such a multi-sequent can always be encoded as a single formula in $\Vdash^{\Phi}_{\mathcal{C}} \left(\bigparr_j G_j^*\right) \parr \left(\bigparr_i F_i\right)$ using $\star$-autonomy (see [81, Proposition 2.1.9]).

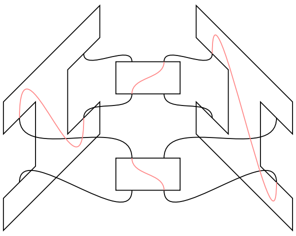
For examples of causal consistency, we return to the example diagrams drawn in the Introduction. First, we had that two process matrices can be simultaneously applied on each side of a bipartite channel. We can express this as a true statement of causal consistency:



(3.8)

$$\begin{gathered}\left(\left(A^1 \multimap B^1\right) \otimes \left(C^1 \multimap D^1\right)\right)^*, \\ \left(\left(P^1 \multimap Q^1\right) \otimes \left(R^1 \multimap S^1\right)\right)^*, \\ A^1 \multimap B^1, \left(C^1 \multimap D^1\right) \parr \left(P^1 \multimap Q^1\right), R^1 \multimap S^1 \Vdash^{\mathrm{CP}^*}_{\Phi}\end{gathered} \tag{3.9}$$

On the other hand, applying them to two bipartite channels simultaneously was *not* causally consistent [75], as we could encode a paradoxical cycle:



(3.10)

$$\begin{gathered}\left(\left(A^1 \multimap B^1\right) \otimes \left(C^1 \multimap D^1\right)\right)^*, \\ \left(\left(P^1 \multimap Q^1\right) \otimes \left(R^1 \multimap S^1\right)\right)^*, \\ \left(C^1 \multimap D^1\right) \parr \left(P^1 \multimap Q^1\right), \left(A^1 \multimap B^1\right) \parr \left(R^1 \multimap S^1\right) \nVdash^{\mathrm{CP}^*}_{\Phi}\end{gathered} \tag{3.11}$$

Recall in Remark 2.3.11 we described that causality of an identity morphism $\mathrm{id}_A \in \mathcal{C}\left(A, A\right)$ for $\mathbf{A} \to \mathbf{A}'$ corresponds to an inclusion $c_{\mathbf{A}} \subseteq c_{\mathbf{A}'}$, and showing this in both directions would demonstrate an equality $\mathbf{A} = \mathbf{A}'$. If we fix an interpretation Φ and some formulae F, G over the same set of atoms, this gives a precise correspondence between the inclusion $c_{\Phi(F)} \subseteq c_{\Phi(G)}$ and a statement of causal consistency $F \Vdash^{\Phi}_{\mathcal{C}} G$ (or into the standard right-sided form $\Vdash^{\Phi}_{\mathcal{C}} F^{*} \parr G$ by $*$-autonomy). By extension, we can encode the search for coherent isomorphisms $\Phi(F) \cong \Phi(G)$ and equalities as a pair of sequents in this way. Parameterising causal consistency by some specific $\mathcal{C}$ and Φ means that characterising causal consistency will not just get the inclusions/isomorphisms that hold for *every* choice of local system, but also those that hold in *special cases* such as the additional equations of the First-Order Theorem.

3.2.2 Extranatural Transformations

Definition 3.2.3 frames causal consistency as a property dependent on a specified interpretation of local systems as objects of $\text{Caus}[\mathcal{C}]$. If we actually want to draw conclusions about the coherent structure of causal categories, we may wish for an alternative definition of causal consistency that determines a property of the category as a whole. We can achieve this by lifting formulae to functors and interpreting a sequent as the existence of an extranatural transformation between them.

Definition 3.2.4: Causal functors

The *causal functor* $\mathcal{F}_F$ of a formula F is defined inductively:

$$\begin{aligned}
\mathcal{F}_A &= 1_{\text{Caus}[\mathcal{C}]} : \text{Caus}[\mathcal{C}] \to \text{Caus}[\mathcal{C}] \\
\mathcal{F}_{A^*} &= (-)^* : \text{Caus}[\mathcal{C}]^{\text{op}} \to \text{Caus}[\mathcal{C}] \\
\mathcal{F}_{A^1} &= \iota : \text{FO}(\text{Caus}[\mathcal{C}]) \hookrightarrow \text{Caus}[\mathcal{C}] \\
\mathcal{F}_{A^{1*}} &= \iota^{\text{op}} \fatsemi (-)^* : (\text{FO}(\text{Caus}[\mathcal{C}]))^{\text{op}} \to \text{Caus}[\mathcal{C}] \\
\mathcal{F}_I &= \mathcal{I} : 1 \to \text{Caus}[\mathcal{C}] \\
\mathcal{F}_{F \otimes G} &= (\mathcal{F}_F \times \mathcal{F}_G) \fatsemi \otimes : \text{dom}(\mathcal{F}_F) \times \text{dom}(\mathcal{F}_G) \to \text{Caus}[\mathcal{C}] \\
\mathcal{F}_{F < G} &= (\mathcal{F}_F \times \mathcal{F}_G) \fatsemi < : \text{dom}(\mathcal{F}_F) \times \text{dom}(\mathcal{F}_G) \to \text{Caus}[\mathcal{C}] \\
\mathcal{F}_{F \parr G} &= (\mathcal{F}_F \times \mathcal{F}_G) \fatsemi \parr : \text{dom}(\mathcal{F}_F) \times \text{dom}(\mathcal{F}_G) \to \text{Caus}[\mathcal{C}]
\end{aligned} \tag{3.12}$$

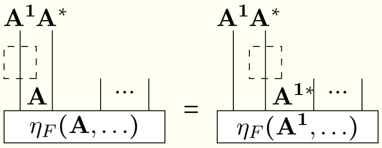
where $\mathcal{I}$ picks out $\mathcal{I}(\star) = \mathbf{I}$, $\mathcal{I}(\text{id}_\star) = \text{id}_I$. Note that the objects of the domain of $\mathcal{F}_F$ coincide with FO-respecting interpretation functions over the atoms of F, up to the unification of instances of the same atom.

Because each morphism in $\text{Caus}[\mathcal{C}]$ is a morphism in $\mathcal{C}$, we can use the existence of a basis of states/effects (APC3) to identify each term of an extranatural transformation between causal functors and determine that it is necessarily just a wiring diagram.

Lemma 3.2.5

Given a balanced formula F, there is at most one extranatural transformation $\eta_F : \mathcal{I} \to \mathcal{F}_F$ which is extranatural across the pairs of indices given by matching atoms in F. When it exists, the term attributed to an interpretation $\Phi : \mathrm{Var} \to \mathrm{Ob}\left(\mathrm{Caus}\left[\mathcal{C}\right]\right)$ is precisely $\left(\epsilon_F^\Phi\right)^*$, i.e. a selection of compact cups connecting matching atoms. By $\ast$-autonomy, this extends to extranatural transformations $\mathcal{F}_G \to \mathcal{F}_F$ for balanced $G^* \invamp F$, where the terms now consist of a mixture of cups, caps, and identity wires.

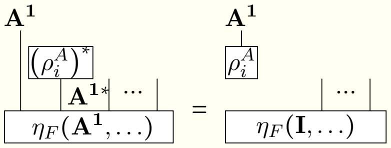
Proof. Consider an arbitrary extranatural transformation η_F of this kind and pick any component of it. For each object $\mathbf{A}$ in the component index, it embeds into some first-order object $\mathbf{A^1}$ by flatness (the causal effects of $\mathbf{A}$ includes the unique causal effect of $\mathbf{A^1}$). This embedding is witnessed by the identity from $\mathcal{C}$, $\mathrm{id}_A : \mathbf{A} \to \mathbf{A^1}$. Extranaturality applies for all causal morphisms, including the identity which we highlight in the dashed boxes below:


$\mathbf{A^1}\mathbf{A}^*$
$\mathbf{A}$
$\cdots$
$\eta_F(\mathbf{A},\ldots)$
=
$\mathbf{A^1}\mathbf{A}^*$
$\mathbf{A^{1*}}$
$\cdots$
$\eta_F(\mathbf{A^1},\ldots)$

(3.13)

Doing this for each index, we find that all components of η_F are fixed by just the first-order ones.

Let $\{\rho_i^A\}_{i \in \mathfrak{B}_A^\mathcal{C}}$ be the causal basis for A from APC3. Each ρ_i^A is a causal state of $\mathbf{A^1}$ up to a scalar, so we can apply extranaturality again.


$\mathbf{A^1}$
$\left(\rho_i^A\right)^*$
$\mathbf{A^{1*}}$
$\cdots$
$\eta_F(\mathbf{A^1},\ldots)$
=
$\mathbf{A^1}$
ρ_i^A
$\cdots$
$\eta_F(\mathbf{I},\ldots)$

(3.14)

Again, repeating this for each index reduces it to the basis state and the scalar term of η_F which (up to unitors) must be id_I, the only causal morphism $\mathbf{I} \to \mathbf{I}$. Doing this for each choice of basis states allows us to completely identify our original arbitrary component of η_F by local tomography, seeing that it acts identically to just a compact cup for each index. □

Definition 3.2.6: Causal consistency by extranatural transformation

A balanced formula F is *causally consistent for $\mathcal{C}$ by extranatural transformation* ($\Vdash_{\mathcal{C}} F$) when there exists an extranatural transformation $\eta_F : \mathcal{I} \to \mathcal{F}_F$ which is extranatural across the pairs of indices given by matching atoms in F. This generalises to two-sided multi-sequents $G_1, \ldots, G_m \Vdash_{\mathcal{C}} F_1, \ldots, F_n$ for an extranatural transformation $\left(\bigtimes_j \mathcal{F}_{G_j}\right) \fatsemi \otimes \to \left(\bigtimes_i \mathcal{F}_{F_i}\right) \fatsemi \parr$.

It is always possible to construct the morphisms $\left(\epsilon_F^\Phi\right)^*$ in $\mathcal{C}$ for any Φ, and they will always collectively satisfy extranaturality by sliding morphisms around the cups. This means the only thing left that determines $\Vdash_{\mathcal{C}} F$ is whether each ϵ_F^Φ is causal, i.e. $\Vdash_{\mathcal{C}}^\Phi F$.

Proposition 3.2.7

For any balanced formula F, $\Vdash_{\mathcal{C}} F$ iff for all interpretations Φ which FO-respect F we have $\Vdash_{\mathcal{C}}^\Phi F$. More generally, $G_1, \ldots, G_m \Vdash_{\mathcal{C}} F_1, \ldots, F_n$ iff $\Vdash_{\mathcal{C}}^\Phi \left(\bigparr_j G_j^*\right) \parr \left(\bigparr_i F_i\right)$ for every FO-respecting Φ.

Proof. By Lemma 3.2.5, $\Vdash_{\mathcal{C}} F$ iff we can build an extranatural transformation where each component (specified by an interpretation function Φ) is given by a cup connecting the pairs of matching variables from F, i.e. each such term from $\mathcal{C}$ is causal $\mathbf{I} \to \Phi(F)$. This exactly describes the transpose of ϵ_F^Φ and so occurs iff $\epsilon_F^\Phi : \Phi(F)^* \to \mathbf{I}$, i.e. iff $\Vdash_{\mathcal{C}}^\Phi F$.

In the general case of two-sided multi-sequents $G_1, \ldots, G_m \Vdash_{\mathcal{C}} F_1, \ldots, F_n$, any extranatural transformation $\left(\bigtimes_j \mathcal{F}_{G_j}\right) \fatsemi \otimes \to \left(\bigtimes_i \mathcal{F}_{F_i}\right) \fatsemi \parr$ is equivalent to one of $\mathcal{I} \to \left(\bigtimes_j \mathcal{F}_{G_j}^* \times \bigtimes_i \mathcal{F}_{F_i}\right) \fatsemi \parr$ by $*$-autonomy. The codomain functor here is equal to $\mathcal{F}_{\left(\bigparr_j G_j^*\right) \parr \left(\bigparr_i F_i\right)}$, bringing us to the definition of $\Vdash_{\mathcal{C}} \left(\bigparr_j G_j^*\right) \parr \left(\bigparr_i F_i\right)$. We can then connect this to causal consistency for all interpretations as before. □

Having now drawn this connection, we will continue to look at causal consistency with respect to an interpretation with the confidence that it will have meaningful consequences on the overall categorical structure.

3.2.3 Modelling Logics

From our existing understandings of the categorical structure of Caus $[\mathcal{C}]$ and the results mentioned in Section 3.1.4 relating them to formal logics, we can already deduce some logics which will generate true statements of causal consistency. Consequently, this can aid in greatly simplifying the task of verifying causal consistency of a given setup, since if an automated theorem prover finds a proof in any of these logics we can immediately conclude that consistency holds.

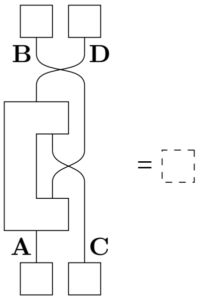
To start with, even before considering the new operators covered in this thesis, $\mathrm{Caus}[\mathcal{C}]$ is an ISOMIX category and hence any statement of MLL+Mix gives rise to a causal structural morphism representing the proof, i.e. a statement of causal consistency.

Example 3.2.8

The following example MLL+Mix proof verifies that one-way signalling processes (encoded using the 2-comb form of FO$\multimap$) embed into the space of all bipartite causal processes.

$$
\begin{prooftree}
\AxiomC{}
\LeftLabel{Ax}
\UnaryInfC{$\vdash A, A^*$}
\AxiomC{}
\LeftLabel{Ax}
\UnaryInfC{$\vdash D, D^*$}
\LeftLabel{$\otimes$}
\BinaryInfC{$\vdash A \otimes D^*, A^*, D$}
\AxiomC{}
\LeftLabel{Ax}
\UnaryInfC{$\vdash B, B^*$}
\AxiomC{}
\LeftLabel{Ax}
\UnaryInfC{$\vdash C, C^*$}
\LeftLabel{Mix2}
\BinaryInfC{$\vdash B, B^*, C, C^*$}
\LeftLabel{$\parr$}
\UnaryInfC{$\vdash B^* \parr C, B, C^*$}
\LeftLabel{$\otimes$}
\BinaryInfC{$\vdash A \otimes (B^* \parr C) \otimes D^*, A^*, B, C^*, D$}
\LeftLabel{$\parr$}
\UnaryInfC{$\vdash (A \otimes (B^* \parr C) \otimes D^*) \parr A^* \parr B \parr C^* \parr D$}
\UnaryInfC{$A \multimap (B \multimap C) \multimap D \vdash (A \multimap B) \parr (C \multimap D)$}
\end{prooftree}
$$

The corresponding statement of causal consistency says that a one-way signalling process can be placed within any context for generic bipartite causal processes - up to affine combination, this is just separable effects.



(3.15)

Proposition 2.6.21 gives the duoidal structure between $\otimes/\parr$ and $<$, from which checking that $\mathrm{Caus}[\mathcal{C}]$ is a BV-category is straightforward (all coherence equations hold from coherence in the compact closed category $\mathcal{C}$).

Proposition 3.2.9

Let F be a formula over the grammar $F ::= A|A^*|F \otimes G|F < G|F \parr G$ (i.e. F does not include any first-order atoms). If there exists a proof of F in BV, then $\Vdash^{\Phi}_{\mathcal{C}} F$ holds for any interpretation Φ.

Proof. Note that since the formula does not contain any first-order atoms, any interpretation Φ will automatically be FO-respecting for F.

We can build up the contraction morphism inductively from the steps of the proof derivation. For each elementary rule, we can attribute it with a corresponding morphism:

$$I\downarrow \qquad \mathrm{id}_I : \mathbf{I} \to \mathbf{I} \tag{3.16}$$

$$ai\downarrow \qquad \epsilon^*_{\Phi(A)} : \mathbf{I} \to \Phi\left(A^* \parr A\right) \tag{3.17}$$

$$q\downarrow \qquad \mathrm{id} \otimes \sigma \otimes \mathrm{id} : \Phi\left((F \parr G) < (H \parr K)\right) \to \Phi\left((F < H) \parr (G < K)\right) \tag{3.18}$$

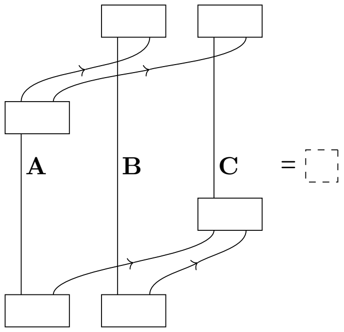
$$s \qquad \mathrm{id} \otimes \mathrm{id} \otimes \mathrm{id} : \Phi\left((F \parr G) \otimes H\right) \to \Phi\left(F \parr (G \otimes H)\right) \tag{3.19}$$

Each context $C\{-\}$ can be mapped to a functor which just tensors on appropriate identity morphisms, since $\{\otimes, <, \parr\}$ are all bifunctors. Therefore, any application of the elementary rules (with a corresponding morphism of type $\Phi(F) \to \Phi(F')$) within the context $C\{-\}$ can be sent to the relevant contraction morphism of type $\Phi\left(C\{F\}\right) \to \Phi\left(C\{F'\}\right)$.

Composing each rule in the derivation of our goal F generates the contraction morphism $\left(\epsilon^\Phi_F\right)^*$ as a causal morphism of type $\mathbf{I} \to \Phi(F)$. This concludes that $\Vdash^\Phi_C F$. □

Example 3.2.10

If we take the proof of $(A \otimes B) < C \multimap A < (B \parr C)$ from Example 3.1.7 and translate it to a statement of causal consistency, we can see it as a compatibility of graph types. Specifically, using Corollary 2.8.12, $(\mathbf{A} \otimes \mathbf{B}) < \mathbf{C}$ and $\mathbf{A}^* < (\mathbf{B}^* \otimes \mathbf{C}^*)$ are both graph types corresponding to sequence-parallel graphs.



(3.20)

Adding products and coproducts, we get the additional equations of MAV for free. Examples here are harder to visualise in the language of string diagrams due to the mixing of monoidal and cartesian structures (though possible through sheet diagrams [34]). Some simple ones include distributivity laws like those in Remark 2.7.10

$$(\mathbf{A} \parr \mathbf{B}) \times (\mathbf{A} \parr \mathbf{C}) \Rightarrow \mathbf{A} \parr (\mathbf{B} \times \mathbf{C}) \tag{3.21}$$

and the interpretation of the *medial* rule as an interchange law between < and ×.

$$(\mathbf{A} \times \mathbf{B}) < (\mathbf{C} \times \mathbf{D}) \Rightarrow (\mathbf{A} < \mathbf{C}) \times (\mathbf{B} < \mathbf{D}) \tag{3.22}$$

We note that the distributivity law merges two occurrences of $\mathbf{A}$ into one, so the witness for this won't just be a simple string diagram but is still composed of simple morphisms. For the following result, we extend Definition 3.2.4 to also apply interpretations to products and coproducts.

Proposition 3.2.11

Let F be a formula over the grammar $F ::= A | A^* | F \otimes G | F < G | F \parr G | F \oplus G | F \times G$. If there exists a proof of F in MAV, then we can translate it to a causal morphism $\mathbf{I} \to \Phi(F)$ in Caus$[\mathcal{C}]$ for any interpretation Φ.

Proof. This follows the same proof strategy as Proposition 3.2.9. The elementary rules are witnessed by the diagonal $\nabla_I = \,\overline{\overline{\top}}_{2^*} : \mathbf{I} \to \mathbf{I} \times \mathbf{I} = \mathbf{2}^*$, injections $\iota_F : \mathbf{F} \to \mathbf{F} \oplus \mathbf{G}$, $\iota_G : \mathbf{G} \to \mathbf{F} \oplus \mathbf{G}$, and the appropriate distributor for $e\downarrow$ and interchange for $m\downarrow$. We achieve deep inference through functoriality of the contexts. □

3.3 Causal Logic

However, none of these logics in the previous section are complete in the sense of being able to determine *all* true statements of causal consistency, with none of them having a way to account for the First-Order Theorem, and the additional equations satisfied in the special case of first-order systems. For this, we will synthetically build a logic up to match and then show that it can be faithfully encoded into pomset logic. This section will present the novel *causal proof-nets* as a solution and prove that they faithfully capture all equations of causal categories or, more specifically, all causally consistent scenarios.

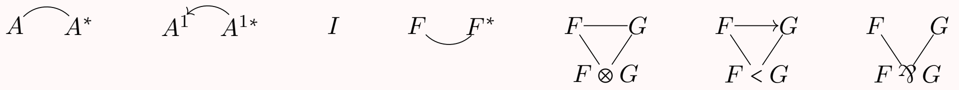
The intuition for building the logic is to use the Sum of Orders Theorem to both break the goal type into a union of graph types and break the contraction morphism into an affine combination of graph states, then use the Graph Compatibility Theorem to reduce the question of compatibility to checking for acyclicity in the composite graphs.

3.3.1 Causal Proof-Nets

This logic will follow similar conventions for proof-nets to those of MLL and pomset and will capture this idea of decomposing into a union of graph types via its switchings.

Definition 3.3.1: Causal proof-structures

A *causal proof-structure* P is a graph defined by composition of the following links:



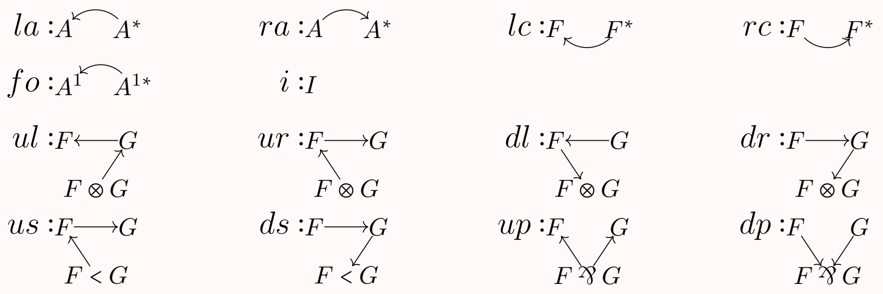
- Axiom links with no premises and a pair of conclusions A, A^*.
- FO-axiom links with no premises and a pair of conclusions A^1, A^{1*}.
- Unit links with no premises and conclusion I.
- Cut links with premises F, F^*, and no conclusions.
- Tensor links with premises F, G, and conclusion $F \otimes G$.
- Seq links with premises F, G, and conclusion $F < G$.
- Par links with premises F, G, and conclusion $F \parr G$.

A balanced formula F identifies a unique cut-free causal proof-structure P_F with conclusion F by replacing each node in its syntax tree with the corresponding link and joining pairs of matching atoms with axiom links. Conversely, for any causal proof-structure P with conclusions $\{C_i\}_i$ we define its corresponding (possibly unbalanced) formula $F_P = C_1 \parr \cdots \parr C_n$.

In the graphical notation, the use of undirected edges is just to indicate that the switchings can induce connectivity across the link in either direction, whereas directed edges indicate that connectivity will be induced in only one direction.

Definition 3.3.2: Up-down switchings

An *up-down switching* $s \in \mathcal{S}_P$ over a causal proof-structure P is a choice of one option for each link dependent on its type from the options below, which together define the *switching graph* G_s over the vertices of P built using the subgraphs:



- For axiom links, one of $\{la, ra\}$.
- For FO-axiom links, the choice is fixed $\{fo\}$.
- For unit links, the choice is fixed $\{i\}$.
- For cut links, one of $\{lc, rc\}$.
- For tensor links, one of $\{ul, ur, dl, dr\}$.
- For seq links, one of $\{us, ds\}$.
- For par links, one of $\{up, dp\}$.

An up-down switching may similarly be defined over the syntax tree of a formula, since this matches a cut-free causal proof-structure with atomic axioms in place of axiom links.

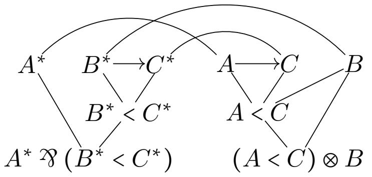
The fact that each switching graph is a directed graph will help us to connect them to graph types later on.

Definition 3.3.3: Causal proof-nets

A causal proof-structure P is a *causal proof-net* when, for every up-down switching $s \in \mathcal{S}_P$, the corresponding switching graph G_s is acyclic.

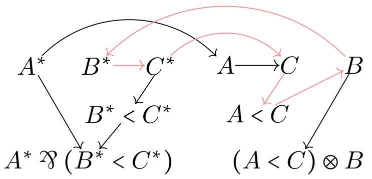
Example 3.3.4

Consider the formula $F = (A \otimes (B < C)) \multimap ((A < C) \otimes B)$ with the following corresponding proof-structure P_F.



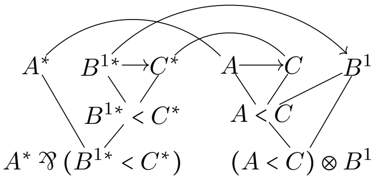
(3.23)

This fails to be a proof-net since it has a switching graph with a cycle, shown in red.



(3.24)

However, if we replace the axiom link for B with an FO-axiom link, we obtain a valid proof-net for $(A \otimes (B^1 < C)) \multimap ((A < C) \otimes B^1)$.



(3.25)

There are notable similarities amongst the switchings and correctness criteria between causal proof-nets and other acyclicity conditions for MLL+Mix (and extensions) such as long-trip, Danos-Regnier, or R&B-graph. Taking a $\parr$ link as an example, each of these switching methods (or the alternating colour restriction for R&B-graphs) allow a path between either premise and the conclusion in both directions but never a path between the premises. Rather than implementing this by connecting the conclusion to only one premise at a time, up-down switchings restrict the directions of paths to either up from the conclusion to the premises (*up*) or down from the premises to the conclusions (*dp*). Similarly, the switchings over $<$ only disallow the path between the premises in one direction, and $\otimes$ allows all paths within the link (though we still split this into multiple choices of switchings to leave directed switching graphs, similar to long-trip switchings). Using this observation, it is immediately obvious that causal logic precisely coincides with MLL+Mix and pomset logic over the corresponding fragments because any falsifying cycle for one kind of proof-net induces a cycle for the others.

Proposition 3.3.5

The logic of causal proof-nets is a conservative extension of pomset logic. Specifically, given a formula in the fragment $F, G ::= A \mid A^* \mid F \otimes G \mid F < G \mid F \parr G$, there exists a causal proof-net for F iff there exists a pomset proof-net for F.

Proof. Comparing causal proof-nets to pomset proof-nets, there is an obvious one-to-one correspondence between proof-structures for unit- and first-order-free formulae, operating link-wise over the proof-structure. The proof-net conditions both reduce down to finding cycles through the links with some connectivity constraints within the links themselves. We can show inductively that each link kind has the same connectivity between its premises and conclusions induced by the alternating colour condition for pomset or the up-down switchings for causal proof-nets, and therefore we can transform between any cycle in an up-down switching and an alternating elementary circuit.

For example, the up-down switchings of a $\parr$ link permit a path in either direction between one premise and the conclusion (the choice of switching just determines the direction), but never a path between the two premises. The corresponding bi-coloured link has a central vertex with undirected red edges to the premises and an undirected blue edge to the conclusion. The alternating-colour condition therefore prevents the path between the premises, but permits the undirected path between each premise and the conclusion.

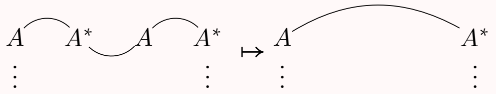
Similarly, $<$ links only forbid connections between the premises in one direction, and $\otimes$ and axiom links permit any connectivity between their components. □

In order for causal proof-nets to actually form a logic in the proof-theoretic sense, we need to demonstrate a cut-elimination result for it. Thankfully, the similarity with pomset logic means we can follow the outline of the proof of pomset cut-elimination [102, Proposition 6, Theorem 7]. We first have to define a set of local rewrites which reduce cuts down to smaller formulae, then prove that each rewrite preserves the proof-net criterion, and show that the rewrites have confluence (regardless of which rewrite we choose at each step, there are always some sequences of further rewrites that end in the same state) and strong normalisation (there are no infinite sequences of rewrites).

Definition 3.3.6: Cut-elimination Steps

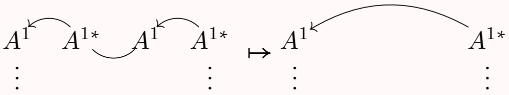
We define the following five rewrites on proof-structures based on substitution of subsets of the links:

AX/AX



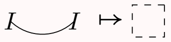
(3.26)

FO/FO



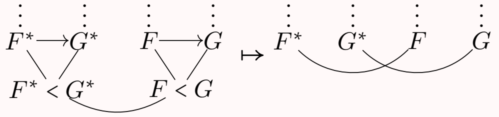
(3.27)

UN/UN



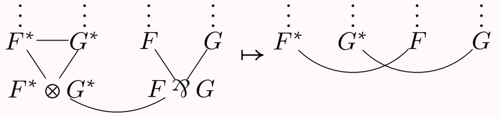
(3.28)

SQ/SQ



(3.29)

TS/PR



(3.30)

For UN/UN, note that we are using $I^* = I$ from Definition 3.2.1.

Proposition 3.3.7

The rewrites AX/AX, FO/FO, UN/UN, SQ/SQ, and TS/PR preserve the conclusions of a proof-structure and acyclicity of all up-down switchings.

Proof. The open premises and conclusions coincide between each pair of redex (left-hand side) and reduct (right-hand side), so the rest of the proof-structure will clearly yield identical conclusions.

It is clear that each redex and reduct cannot have any internal cycles for any up-down switching, so it suffices to check that the rewrites do not introduce any new connectivities between the open premises or conclusions (if a cyclic switching exists after the rewrite, then we can translate it to a cyclic switching in the original proof-structure). AX/AX and FO/FO preserve this exactly, respectively allowing connections in either direction between A and A^*, and only from A^{1*} to A^1. UN/UN is trivial as it has no open premises or conclusions in the redex or reduct. SQ/SQ and TS/PR strictly decrease the connectivity, maintaining connectivity in both directions between F^* and F, and between G and G^*, but dropping any connectivities between F^* and G^* (and between F and G for SQ/SQ). □

Proposition 3.3.8: Cut-elimination

The collection of rewrites $\{\mathrm{AX/AX}, \mathrm{FO/FO}, \mathrm{UN/UN}, \mathrm{SQ/SQ}, \mathrm{TS/PR}\}$ on causal proof-structures enjoys confluence and strong normalisation. Furthermore, the unique final proof-structure has the same conclusions and does not include any cut links.

Proof. The only redexes that can overlap are AX/AX with itself, or FO/FO with itself. In either case, this scenario looks like a chain of (first-order) axiom links and cuts. Regardless of which cut is eliminated, the resulting chain of axiom and cuts links is the same. All other redexes are disjoint, so confluence follows immediately.

For strong normalisation, the number of links in the proof-structure is strictly decreased by each rewrite application, so every sequence of rewrites must eventually terminate.

Preservation of conclusions was shown in Proposition 3.3.7. In any proof-structure with a cut, pick one of its premises F and consider the cases from the grammar of causal formulae. In each case, there is a single kind of link that generates a conclusion matching that pattern which uniquely identifies the link before the cut. Doing the same for the other premise F^* will give a match for the redex of one of the five rules. Therefore, if no more rewrites can be applied, the proof-structure is cut-free. □

We can give an interpretation of the switching options for links in terms of the flows of information between their premises. As we saw in Definition 3.2.3, causal consistency $\Vdash^{\Phi}_{\mathcal{C}} F$ means the canonical wiring map $\epsilon^{\Phi}_{F} : \Phi(F)^* \to \mathbf{I}$ sends every state in $c_{\Phi(F)^*}$ to the scalar id_I. Crucial to the proof of the characterisation, this fails to be the case precisely when plugging ϵ^{Φ}_{F} into some $\rho : \Phi(F)^*$ could result in a directed cycle of signalling relations. The signalling relations introduced by ϵ^{Φ}_{F} are relatively simple to characterise: a wire between X^* and X for a generic type X can introduce signalling in either direction, whereas information can only flow in one direction for first order types (from A^{1*} to A^1). This follows from the fact that the only thing we can "plug in" to A^1 is the unique causal effect $\overline{\overline{\top}}_{A^1}$, hence there is no way to use the choice of effect to send any non-trivial information. On the other hand, we can plug any causal state of $\Phi(A^1)$ into A^{1*}, of which there will typically be many.

We can then combine this with the signalling relations allowed by $\Phi(F)^*$. Since $\Phi(F)$ appears under a $(-)^*$, the roles of $\otimes$ and $\parr$, with respect to (non-)signalling relations, are reversed. Namely, an occurrence of $X \otimes Y$ in $\Phi\left(F\right)^*$ means that signalling can occur in either direction between the premises X and Y, generalising the case of process matrices from Theorem 2.3.24. Similarly, an occurrence of $X < Y$ in $\Phi\left(F\right)^*$ means that signalling can only occur from X to Y, whereas $X \parr Y$ in $\Phi\left(F\right)^*$ doesn't allow any signalling between premises.

In addition to fixing a flow of information between premises, we also have to consider the flow of information between a subexpression and its environment (i.e. the rest of the syntax tree): either "up" toward the leaves of the tree or "down" toward the root. Fixing this direction plus a signalling direction between premises yields all the possible choices for connectives (4 choices for $\otimes$, 2 choices each for $<$ and $\parr$). The "up" vs. "down" signalling direction is slightly harder to think about intuitively, but we can make this precise using (unions of) graph types, which we will do in the next section.

3.3.2 The Characterisation Theorem

Recall that the Sum of Orders Theorem equates any $\mathbf{A} \parr \mathbf{B}$ with a union of sequence types. Applying this recursively along with Setwise Distributivity, we can reduce any formula involving $\parr$ to a union of graph types. The up-down switchings are defined to respect this decomposition so the switching graphs over the syntax tree of a formula F inductively yield the graph types generating $\Phi(F)^*$.

Lemma 3.3.9: Switching Lemma

Given a balanced formula F and an interpretation Φ which FO-respects F and an additional object $\mathbf{E} \in \mathrm{Ob}\left(\mathrm{Caus}\left[\mathcal{C}\right]\right)$, let V be the vertices of the syntax tree of F (in negation normal form) and let $\Gamma^*_{\Phi,\mathbf{E}} : V \to \mathrm{Ob}\left(\mathrm{Caus}\left[\mathcal{C}\right]\right)$ be the function below which assigns $\Phi\left(a\right)^*$ to each leaf labelled a, $\mathbf{E}$ to the root, and $\mathbf{I}$ elsewhere (with a degenerate case for the one-vertex tree):

$$\Gamma^*_{\Phi,\mathbf{E}}\left(v\right) = \begin{cases} \Phi\left(F\right)^* \parr \mathbf{E} & v \text{ is the only vertex } F ::= A|\ A^*|A^1|A^{1*}|I \\ \Phi\left(a\right)^* & v \text{ is a leaf } a ::= A|A^*|A^1|A^{1*}|I \\ \mathbf{E} & v \text{ is the root F} \\ \mathbf{I} & \text{otherwise} \end{cases} \tag{3.31}$$

Then $\Phi\left(F\right)^* \parr \mathbf{E} \cong \bigcup_{s \in \mathcal{S}_F} \mathbf{Gr}_{G_s}^{\mathbf{\Gamma}^*_{\Phi,\mathbf{E}}}$. In words with $\mathbf{E} = \mathbf{I}$, the state space of $\Phi\left(F\right)^*$ coincides with the space generated by graph states over the switching graphs of the syntax tree of F.

Proof. We proceed inductively over the syntax tree of F (in negation normal form).

If $F ::= A|A^*|A^1|A^{1*}|I$, the syntax tree is a single node with a single trivial switching graph.

$$\Phi\left(F\right)^* \parr \mathbf{E} = \Gamma^*_{\Phi,\mathbf{E}}\left(v\right) = \mathbf{Gr}^{\Gamma^*_{\Phi,\mathbf{E}}}_{G_s} \tag{3.32}$$

If $F = G \otimes H$:

$$\begin{aligned}\Phi\left(F\right)^* \parr \mathbf{E} &= \left(\Phi\left(G\right) \otimes \Phi\left(H\right)\right)^* \parr \mathbf{E}\\ &= \left(\Phi\left(G\right)^* \parr \Phi\left(H\right)^*\right) \parr \mathbf{E}\\ &= \left(\left(\Phi\left(G\right)^* < \Phi\left(H\right)^*\right) \cup \left(\Phi\left(H\right)^* < \Phi\left(G\right)^*\right)\right) \parr \mathbf{E}\\ &= \left(\Phi\left(G\right)^* < \Phi\left(H\right)^* < \mathbf{E}\right) \cup \left(\Phi\left(H\right)^* < \Phi\left(G\right)^* < \mathbf{E}\right)\\ &\quad \cup \left(\mathbf{E} < \Phi\left(G\right)^* < \Phi\left(H\right)^*\right) \cup \left(\mathbf{E} < \Phi\left(H\right)^* < \Phi\left(G\right)^*\right)\\ &= \mathbf{Gr}^{\Gamma_\otimes}_{dr} \cup \mathbf{Gr}^{\Gamma_\otimes}_{dl} \cup \mathbf{Gr}^{\Gamma_\otimes}_{ur} \cup \mathbf{Gr}^{\Gamma_\otimes}_{ul}\end{aligned} \tag{3.33}$$

where the graphs dr, dl, ur, ul are given by the corresponding switching options for the $\otimes$ link, and $\Gamma_\otimes$ maps the premise vertices to $\Phi(G)^*$ and $\Phi(H)^*$ and the conclusion vertex to $\mathbf{E}$.

Looking at $\mathbf{Gr}^{\Gamma_\otimes}_{dr}$ as an example, the component typings for the graph states (assuming wlog the constant edge interpretation $\Delta_\mathbf{2}$) are $\Phi(G)^* \parr \mathbf{2}$, $\Phi(H)^* \parr (\mathbf{2}^* \parr \mathbf{2})$, and $\mathbf{E} \parr \mathbf{2}^*$. The inductive hypothesis identifies the premise components with affine combinations of graph states over the switching graphs of G and H. Identifying these within complete graph states gives $\mathbf{Gr}^{\Gamma_\otimes}_{dr} = \bigcup_{s \in \mathcal{S}_F, s(F) = dr} \mathbf{Gr}^{\Gamma^*_{\Phi,\mathbf{E}}}_{G_s}$ (the leaf and otherwise cases of $\Gamma^*_{\Phi,\mathbf{E}}$ handle the fact that we have moved the $\mathbf{2}$ parts of the premises from the local interpretation of the inductive graph type to edge interpretations in the larger switching graph). Repeating this for each of dl, ur, ul generates all switching graphs over F.

The cases for $<$ and $\parr$ are similar with the following decompositions:

$$\begin{aligned}\left(\Phi\left(G\right) < \Phi\left(H\right)\right)^* \parr \mathbf{E} &= \left(\Phi\left(G\right)^* < \Phi\left(H\right)^* < \mathbf{E}\right)\\ &\quad \cup \left(\mathbf{E} < \Phi\left(G\right)^* < \Phi\left(H\right)^*\right)\\ &= \mathbf{Gr}^{\Gamma_<}_{ds} \cup \mathbf{Gr}^{\Gamma_<}_{us}\end{aligned} \tag{3.34}$$

$$\begin{aligned}\left(\Phi\left(G\right) \parr \Phi\left(H\right)\right)^* \parr \mathbf{E} &= \left(\left(\Phi\left(G\right)^* \otimes \Phi\left(H\right)^*\right) < \mathbf{E}\right)\\ &\quad \cup \left(\mathbf{E} < \left(\Phi\left(G\right)^* \otimes \Phi\left(H\right)^*\right)\right)\\ &= \mathbf{Gr}^{\Gamma_\parr}_{dp} \cup \mathbf{Gr}^{\Gamma_\parr}_{up}\end{aligned} \tag{3.35}$$

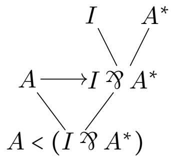
□

Example 3.3.10

Consider $F = A < (I \parr A^*)$, interpreting $\Phi(A) = \mathbf{A}$ and fixing some environment object $\mathbf{E}$. Then this lemma gives the following decomposition:

$$
\begin{aligned}
\Phi(F)^* \parr \mathbf{E} &= (\mathbf{A}^* < (\mathbf{I} \otimes \mathbf{A})) \parr \mathbf{E} \\
&\cong \mathbf{Gr}\left(\begin{tikzcd} & \mathbf{I} & & \mathbf{A} \\ \mathbf{A}^* \arrow[rr] & & \mathbf{I} \arrow[ul] \arrow[ur] & \\ & \mathbf{E} \arrow[ul] & & \end{tikzcd}\right) \cup \mathbf{Gr}\left(\begin{tikzcd} & \mathbf{I} \arrow[dr] & & \mathbf{A} \arrow[dl] \\ \mathbf{A}^* \arrow[rr] & & \mathbf{I} & \\ & \mathbf{E} \arrow[ul] & & \end{tikzcd}\right) \\
&\cup \mathbf{Gr}\left(\begin{tikzcd} & \mathbf{I} & & \mathbf{A} \\ \mathbf{A}^* \arrow[rr] & & \mathbf{I} \arrow[ul] \arrow[ur] \arrow[dl] & \\ & \mathbf{E} & & \end{tikzcd}\right) \cup \mathbf{Gr}\left(\begin{tikzcd} & \mathbf{I} \arrow[dr] & & \mathbf{A} \arrow[dl] \\ \mathbf{A}^* \arrow[rr] & & \mathbf{I} \arrow[dl] & \\ & \mathbf{E} & & \end{tikzcd}\right)
\end{aligned}
\tag{3.36}
$$

where these four graphs match the switching graphs of the syntax tree of F.



(3.37)

This lemma presents a great simplification of characterising causal consistency from needing to check causality of contractions of arbitrary types to just those for graph types. Comparing this to our discussion of $\parr$ as a union of seq types in Section 2.7.2, this holds physical significance as some types may contain processes exhibiting indefinite causal structure where the patterns of influence are inconsistent with any graph type no matter how we divide up the subsystems, yet they appear to add no new behaviour that is relevant for determining causal consistency.

As for the axiom links, they tell us which pairs of systems we will wire up, corresponding to the contractions in the analogy to tensor networks. We apply the same decomposition to the contractions

$$\epsilon_{\Phi(A)} : \Phi(A^*) \parr \Phi(A) = (\Phi(A^*) < \Phi(A)) \cup (\Phi(A^*) > \Phi(A)) \tag{3.38}$$

giving us directed switchings over the axiom links. If we ask that Φ is *strongly* FO-respecting ($\Phi(A)$ is neither first-order nor first-order dual), then $\epsilon_{\Phi(A)}$ must have the capacity to signal information in each direction. This makes both switchings necessary to cover all information flow in the contraction. Similarly, the FO-axiom links only have a single switching option since

$$\epsilon_{\Phi(A^1)} : \Phi\left(A^{1*}\right) \parr \Phi\left(A^1\right) = \Phi\left(A^{1*}\right) < \Phi\left(A^1\right) \tag{3.39}$$

is already one-way signalling. Φ being strongly FO-respecting here guarantees that $\Phi(A^1) \ncong \mathbf{I}$ and so there necessarily is at least some information signalling capacity that needs to be represented.

Combining the axiom decompositions and Switching Lemma into a single picture, we obtain the switching graphs over the whole proof-structure. If there is a cycle, we can devise an example implementation for the black box which, when contracted, encodes a paradoxical situation (i.e. an information cycle which breaks normalisation) to disprove causal consistency; otherwise, there is some linear ordering of the vertices that all information flow respects, from which we can prove that normalisation is preserved. This covers the intuition for the proof of the Causal Characterisation Theorem.

Theorem 3.3.11: Causal Characterisation Theorem

Given a balanced formula F and an interpretation Φ which strongly FO-respects F, $\Vdash^{\Phi}_{\mathcal{C}} F$ iff P_F is a causal proof-net.

Proof. Recall the definition of $\Vdash^{\Phi}_{\mathcal{C}}$ as stating whether ϵ^{Φ}_{F} is causal $\Phi\left(F\right)^{*} \to \mathbf{I}$, i.e. for all causal states $\rho \in c_{\Phi(F)^{*}}$, $\rho \fatsemi \epsilon^{\Phi}_{F} = \mathrm{id}_{I}$. We decompose ϵ^{Φ}_{F} into a permutation followed by the individual caps in parallel $\epsilon^{\Phi}_{F} = \sigma \fatsemi \bigotimes_{A \in F} \epsilon_{\Phi(A)}$ with each $\epsilon_{\Phi(A)} : \Phi\left(A^{*}\right)^{*} \otimes \Phi\left(A\right)^{*} \to \mathbf{I}$.

We first claim that $\Vdash^{\Phi}_{\mathcal{C}} F$ iff $c_{\bigotimes_{A \in F} \Phi(A^{*}) \parr \Phi(A)} \subseteq \left(c_{\Phi(F)^{*}}\right)^{*}$ up to the appropriate permutation of atomic wires. $\Longleftarrow$ is immediate from $\epsilon_{\Phi(A)} \in c_{\Phi(A^{*}) \parr \Phi(A)}$. For $\Longrightarrow$, we note that any element of $c_{\Phi(A^{*}) \parr \Phi(A)}$ can be written in the form $\epsilon^{*}_{\Phi(A)} \fatsemi \left(\mathrm{id}_{\Phi(A^{*})} \otimes f\right)$ for some $f : \Phi\left(A\right) \to \Phi\left(A\right)$ by $*$-autonomy. Given any state $\rho \in c_{\Phi(F)^{*}}$, we have $\rho \fatsemi \left(\mathrm{id} \otimes f^{*} \otimes \mathrm{id}\right) \in c_{\Phi(F)^{*}}$ since every operator used to construct $\Phi\left(F\right)^{*}$ is functorial. We can apply this for each atom A in F to reduce $c_{\bigotimes_{A \in F} \Phi(A^{*}) \parr \Phi(A)} \subseteq \left(c_{\Phi(F)^{*}}\right)^{*}$ to just checking $\epsilon^{\Phi}_{F} \in c_{\Phi(F)}$, i.e. $\Vdash^{\Phi}_{\mathcal{C}} F$.

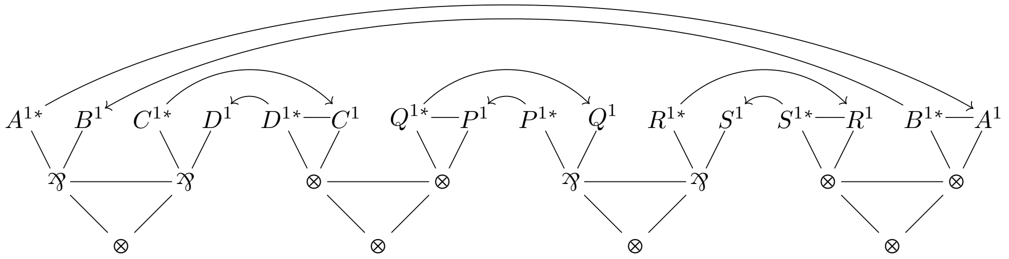
Next, the First-Order Theorem gives $\Phi\left(A^{*}\right) \parr \Phi\left(A\right) = \Phi\left(A^{*}\right) < \Phi\left(A\right)$ iff $\Phi\left(A\right)$ is first-order. Since Φ strongly FO-respects F, this happens iff A appears in F as a first-order atom A^{1}. By Equation 2.244, $\Phi\left(A^{*}\right) < \Phi\left(A\right)$ is a graph type matching the only switching option available to first-order axiom links. Similarly, the Sum of Orders Theorem gives the decomposition for generic atoms as $\Phi\left(A^{*}\right) \parr \Phi\left(A\right) = \left(\Phi\left(A^{*}\right) < \Phi\left(A\right)\right) \cup \left(\Phi\left(A^{*}\right) > \Phi\left(A\right)\right)$ which match the two switching options for regular axiom links.

Applying the Switching Lemma, $\left(\Phi\left(F\right)^{*}\right)^{*}$ is the intersection of the dual graph types for each switching graph over the syntax tree of F. Our goal $c_{\bigotimes_{A \in F} \Phi(A^{*}) \parr \Phi(A)} \subseteq \left(c_{\Phi(F)^{*}}\right)^{*}$ is now inclusion of a union within an intersection, meaning we need to quantify over both the switchings of axiom links and the switchings of the syntax tree, i.e. over all switchings of the canonical cut-free proof-structure. For each such switching s, we ask the graph type given by the axiom switching (extended to all vertices of the syntax tree with $\mathbf{I}$'s via Lemma 2.8.17) is contained in the dual graph type over the syntax tree $\left(\mathbf{Gr}^{\Gamma^*_{\Phi},\mathbf{I}}_{G_s}\right)^*$. Finally, the Graph Compatibility Theorem equates this with checking acyclicity of the combined graph, i.e. the full switching graph over the entire proof-structure (the fact that this takes the standard forms $\overline{G}$ of graphs does not matter here, since the first-order axiom links will already be directed in the appropriate direction, so any cycle can still be used to disprove causality). □

Example 3.3.12

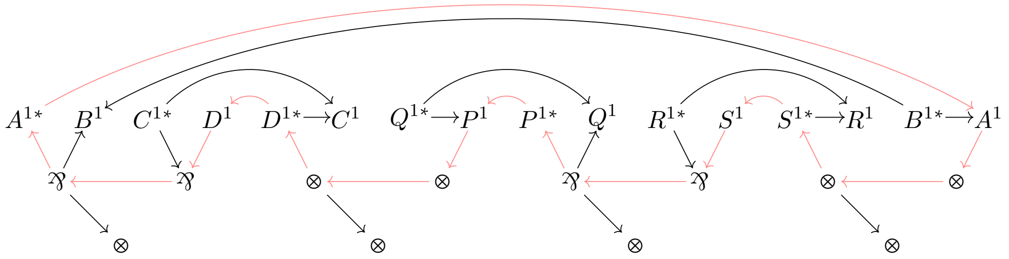
In Equation 3.11 we saw an example sequent which is not causally consistent, describing two process matrices that are applied to two bipartite channels simultaneously. Writing it as a right-handed sequent with a single term gives us the following, with the corresponding proof-structure (up to permutations of systems) shown below.

$$\begin{aligned}\nVdash^{\Phi}_{\mathcal{C}} &\left(\left(A^1 \multimap B^1\right) \otimes \left(C^1 \multimap D^1\right)\right) ⅋ \left(\left(P^1 \multimap Q^1\right) \otimes \left(R^1 \multimap S^1\right)\right) \\ &⅋ \left(\left(C^1 \multimap D^1\right)^* \otimes \left(P^1 \multimap Q^1\right)^*\right) ⅋ \left(\left(A^1 \multimap B^1\right)^* \otimes \left(R^1 \multimap S^1\right)^*\right)\end{aligned} \tag{3.40}$$



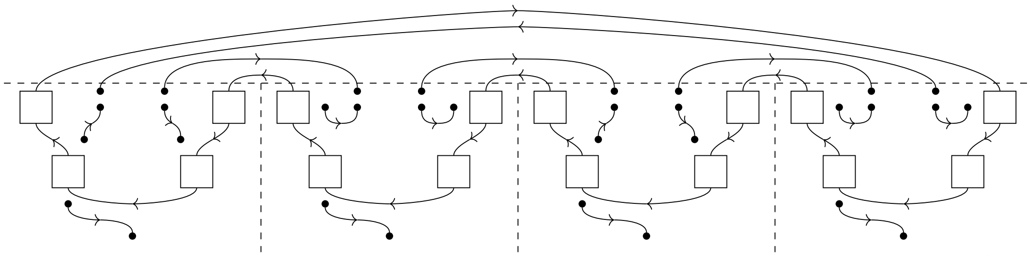
(3.41)

This fails to be a proof-net, as seen by the following switching with a cycle shown in red.



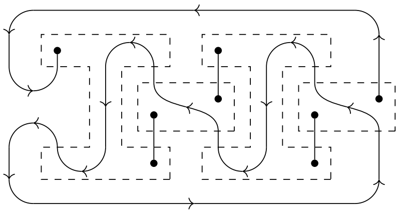
(3.42)

With the Switching Lemma we can interpret everything below the axiom links as describing a graph type, with the axioms describing some contractions. If we follow the construction used in the Acyclicity Lemma, we will build a graph state which, when contracted, gives a cycle of information flow. We depict such an example below, where the dashed lines subdivide it into the two process matrices (outer left and inner right), the two bipartite channels (inner left and outer right) and the contraction morphism (top). The individual boxes represent the (noisy) channels used to encode binary data into arbitrary systems and the dots represent local states and effects (e.g. the uniform states and effects for the appropriate objects).



(3.43)

Rearranging this back into the standard graphical shapes used for process matrices and channels for clarity (with the assumption all the systems are identical, so the boxes can simply be replaced with identity wires):



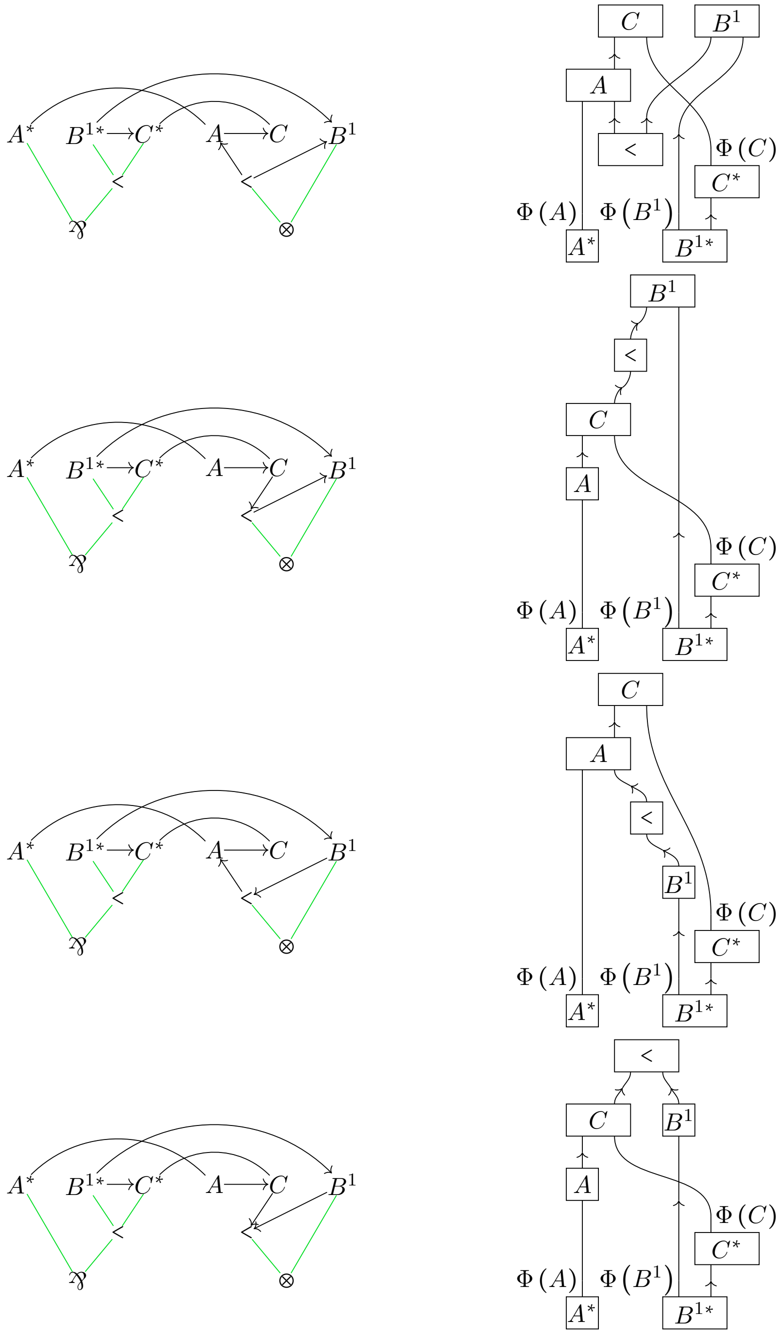
(3.44)

This matches the cycle informally suggested by Equation 1.7 in the Introduction and matches the observations of Jia and Sakharwade [75] in their analysis of such pairs of process matrices.

Example 3.3.13

Example 3.3.4 gave a proof-net for $F = (A \otimes (B^1 < C)) \multimap ((A < C) \multimap B^1)$. Rather than go through all 128 switching graphs, we will ignore the paths that can't induce connectivity between atoms (those greyed out in the graphs below) and combine the switchings over the axioms for simplicity. That leaves just 4 cases, each of which we can relate to graph states that generate the full space of

$\Phi(F)^*$ (the boxes are labelled with the corresponding vertex of the switching graph). Each graph is acyclic, allowing us to easily verify causality.



When using this theorem as a way to determine causal consistency, we still have to adhere to the restriction to *strongly* FO-respecting interpretations which won't always be guaranteed. However, we can always make minor adaptations to the formula and interpretation to guarantee strong FO-respect whilst preserving the semantics of causal consistency.

Proposition 3.3.14

Given any interpretation Φ for a formula F, there exists some formula F' and interpretation Φ' which strongly FO-respects F' such that $\Phi(F) = \Phi'(F')$ and hence $\Vdash_{\mathcal{C}}^{\Phi} F$ iff $\Vdash_{\mathcal{C}}^{\Phi'} F'$.

Furthermore, if P_F is a causal proof-net then so is $P_{F'}$.

Proof. Failing to strongly FO-respect F means there is some atom A such that $\Phi(A)$ is first-order or first-order dual and this is not reflected by the occurrences of A in F. We construct F' and Φ' to match F and Φ with the following amendments:

- If $\Phi(A) \cong \mathbf{I}$, then we replace all occurrences of A or A^1 in F with I;
- If $\Phi(A)$ is first-order but not first-order dual, we replace all occurrences of A with A^1;
- If $\Phi(A)$ is first-order dual but not first-order, we dualise the interpretation $\Phi'(A) := \Phi(A)^*$ and replace all occurrences of A with A^{1*}.

Each of these updates preserves the overall interpretation of the formula $\Phi(F) = \Phi'(F')$.

As for how these updates affect the proof-structures, a subset of the axiom links are replaced by FO-axiom links in one direction or the other, and we replace some (FO-)axiom links by pairs of unit links. In the former case we just end up with a subset of the switching graphs which means fewer of the same acyclicity checks are required, and the latter we have a single switching graph with fewer edges which cannot introduce cycles. □

Corollary 3.3.15

$\Vdash_{\mathcal{C}} F$ iff P_F is a causal proof-net.

Proof. Proposition 3.2.7 states that $\Vdash_{\mathcal{C}} F$ is equivalent to checking $\Vdash_{\mathcal{C}}^{\Phi} F$ for every FO-respecting Φ.

If P_F is a causal proof-net, then $\Vdash_{\mathcal{C}}^{\Phi} F$ for all strongly FO-respecting Φ immediately (by the Causal Characterisation Theorem), and Proposition 3.3.14 handles the rest - P_F is a proof-net $\Longrightarrow$ $P_{F'}$ is also a proof-net $\Longrightarrow$ $\Vdash_{\mathcal{C}}^{\Phi'} F' \Longrightarrow$ $\Vdash_{\mathcal{C}}^{\Phi} F$.

If P_F is not a causal proof-net, it is straightforward to construct a strongly FO-respecting interpretation which will fail causal consistency. □

Beyond the usual Curry-Howard-Lambek style results where the logic describes the behaviour common to all interpretations, this ability to make slight adaptations to guarantee strong FO-respect means this proof-net criterion can capture causal consistency for *any specific* interpretation! This has an extremely significant consequence: the only information about atomic objects which is relevant for determining causal relations is whether or not they are first-order or first-order dual. We already knew from the First-Order Theorem that they satisfied additional equations between causal types, but now we can now conclude that they are the *only* objects that do so.

In addition to being (almost entirely) independent of the interpretation, the proof-net criterion is independent of our chosen base category $\mathcal{C}$. Not only are the equations between causal types independent of the atomic objects, but they are also completely theory-independent: the inclusions between causal structures are the same for higher-order quantum theory ($\mathrm{Caus}\left[\mathrm{CP}^*\right]$), classical probability theory ($\mathrm{Caus}\left[\mathrm{Mat}[\mathbb{R}^+]\right]$), and pseudo-probability theory ($\mathrm{Caus}\left[\mathrm{Mat}[\mathbb{R}]\right]$).

Recall from Proposition 3.3.5 that causal logic conservatively extends pomset logic, improving on the lower bound of BV from Section 3.2.3. A major consequence of this comes in terms of the computational complexity of checking causal consistency, since it must be at least as hard as verifying a pomset proof-net which is coNP-complete [94]. coNP fits nicely with our picture of causality, since failure of consistency amounts to giving a cycle of information flow through the system which can be efficiently verified.

Corollary 3.3.16

Causal consistency of a balanced formula is coNP-complete.

Proof. It is coNP-complete to verify that a proof-structure is a proof-net for pomset, i.e. whether a balanced formula is satisfiable in pomset logic. Since unit- and first-order free causal proof-nets coincide with pomset proof-nets, causal consistency must be at least coNP-hard. It lies in coNP itself since giving an up-down switching and a cycle gives an efficiently verifiable refutation. □

3.3.3 Standardised Interpretations

We can see notable similarity between the Causal Characterisation Theorem and Retoré's result that pomset logic proof-nets have a sound and faithful interpretation in coherence spaces [101]. In that result, it is shown that correctness of a proof-net is not only implied by the existence of a clique for every interpretation of the atoms, but it is sufficient to only consider interpretations that map each atom to some four-token coherence spaces N and Z = N*. We can do a similar thing here by picking a standardised (almost) constant interpretation such that causal consistency precisely matches the existence of causal proof-nets. By extension, if we wish to study causal consistency by enumeration of states in the appropriate physical theory, it is sufficient to check for this standardised interpretation to know what holds for an arbitrary one.

The distinction of being "almost" constant is the need to distinguish between different classes of atoms. **2** is a good candidate for first-order variables, it being the simplest object which is first-order but not first-order dual.

For regular variables, we should associate them with an object that is neither first-order nor first-order dual, giving us a couple of options that stand out. $\mathbf{A_1} = \left(3, \left\{\iota_1, \frac{1}{3}\underline{\bot}_3\right\}^{**}\right)$ from Remark 2.7.8 is similar to N in that it is the object with the lowest dimensions (with respect to both the carrier object and state set) which is not isomorphic to any object definable from $\mathbf{I}$ with $\oplus, \times, (-)^*, \otimes, <$.

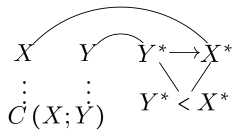
Alternatively, we could encode the ability to send information in both directions in the straightforward way by considering a first-order and a first-order dual system side-by-side. We see this in any higher-order framework by mapping atoms to function types like $\mathbf{2} \multimap \mathbf{2}$. This is appealing since we can then see the interpretation as built entirely with $\mathbf{I} \multimap \mathbf{I}$ ($\cong \mathbf{I}$), $\mathbf{I} \multimap \mathbf{2}$ ($\cong \mathbf{2}$), $\mathbf{2} \multimap \mathbf{I}$ ($\cong \mathbf{2}^*$), and $\mathbf{2} \multimap \mathbf{2}$. This reinforces the idea that causal consistency is theory-independent, showing that it is sufficient to consider stochastic maps on classical bits to even learn how causal structures compose in, for example, quantum theory.

3.4 Sufficient Fragments

3.4.1 The pomset Fragment

Causal logic extends pomset with first-order atoms. The First-Order Theorem characterises first-order objects as precisely those objects over which the identity process (and hence the contraction morphism $\epsilon_{\Phi(A^1)}$ corresponding to an axiom link) is one-way signalling. Instead of using the directed FO-axiom link, we can use this characterisation to just describe that the two atoms are linked by a one-way channel, which is something we can much more naturally express in a formula.

Consider the following component of a proof-structure:



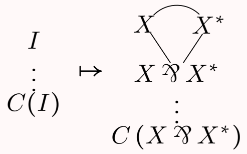
(3.45)

Enumerating the switching graphs over this component, there are no internal cycles possible and the only induced connectivity is from Y to X, matching exactly the connectivity of an FO-axiom link. Because the connectivity matches, we can convert any cycle in a proof-structure containing an FO-axiom link to a cycle in the same proof-structure with the FO-axiom link replaced by this component and vice versa, so this replacement must preserve the correctness of the proof-net criterion.

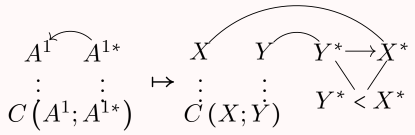
The following definition introduces this formally as a rewrite procedure between causal proof-structures (and subsequently their corresponding formulae), which always reduces down to the fragment $F, G ::= A \mid A^* \mid F \otimes G \mid F < G \mid F \parr G$ used by pomset logic.

Definition 3.4.1

Any proof-structure P induces a proof-structure $\mathrm{pom}\,(P)$ with no unit or FO-axiom links by applying the following rewrites exhaustively:



(3.46)

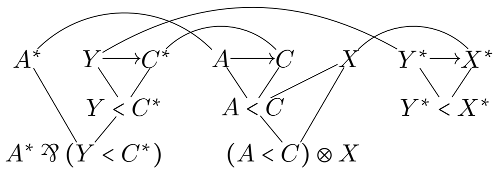


(3.47)

where X and Y are fresh variable names for each instance of the rewrite and C denotes the labels at any other link in the proof-structure as a function of the given subterms.

Example 3.4.2

Applying $\mathrm{pom}\,(-)$ to the proof-net of Equation 3.25 expands the FO-axiom link, now representing the formula $(A \otimes (Y < C)) \multimap ((A < C) \otimes X) \parr (Y^* < X^*)$:



(3.48)

Example 3.4.3

Applying this transformation just to a formula, we can encode the characteristic properties from the First-Order Theorem and show that they follow in pomset, and even in BV. For example, the interesting direction of the equality

$$\mathbf{A^1} \otimes \mathbf{B} = \mathbf{A^1} < \mathbf{B} \tag{3.49}$$

is given by the formula

$$\left(A^1 < B\right) \multimap \left(A^1 \otimes B\right) \tag{3.50}$$

which pom $(-)$ maps to the conclusion of the following BV derivation:

$$\dfrac{\dfrac{\text{ai}\downarrow\dfrac{I\downarrow\dfrac{}{I}}{B^*\parr\left(\text{s}\,\dfrac{\dfrac{B}{\left(\text{ai}\downarrow\dfrac{I}{Y^*\parr Y}\right)\otimes B}}{Y^*\parr(Y\otimes B)}\right)}}{\left(\text{q}\downarrow\dfrac{\dfrac{B^*\parr Y^*}{\left(\text{ai}\downarrow\dfrac{I}{X^*\parr X}\right)<(B^*\parr Y^*)}}{(X^*<B^*)\parr(X<Y^*)}\right)\parr(Y\otimes B)}}{(X^*<B^*)\parr(Y\otimes B)\parr(X<Y^*)}$$

Proposition 3.4.4

For any causal proof-structure P, it is a causal proof-net iff pom (P) is.

Proof. Much like Proposition 3.3.5, this amounts to checking that connectivity between the end-points is preserved through each of the selected rewrites when locally quantifying over the up-down switchings, and that the replacements don't contain internal cycles. In each case, this is straightforward. □

This draws an even closer link between the logic of causal proof-nets and pomset. Not only do we have a conservative extension, but the theorems of pomset logic completely determine the truth of all statements in causal logic. Whilst there is no new behaviour that cannot be derived from pomset logic, there are still practical benefits to the model from keeping first-order systems as part of the logic due to their important physical semantics.

At the time of writing this thesis, there is no known category-theoretic characterisation of pomset logic, but the existence of such extranatural transformations for all provable formulae makes a convincing case that Caus$[\mathcal{C}]$ should be an instance of a "pomset category" for any suitable definition.

3.4.2 Separating BV and pomset with Process Matrices

The process of expanding an axiom link to restrict it to a single direction performed by the pom$(-)$ translation is not new in the logic literature. For example, in [93, Figure 7], Nguyễn and Straßburger begin with (a formula equivalent to) the following, which is not provable in MLL+Mix (and thus not provable in the conservative extensions, BV and pomset):

$$\nvdash ((P \multimap Q) \otimes (R \multimap S))^* \multimap ((Q \multimap R) \otimes (S \multimap P)) \tag{3.51}$$

They then apply a similar rewrite to direct the axioms and break any cycles, generating a formula which is provable in pomset (and therefore causal logic) but still not in BV. Like the pom$(-)$ translation, this generates two variables in place of each one so we use new names to distinguish them (for example, the two instances of P above are replaced by A and H below).

$$\begin{aligned}\vdash &((A < B) \otimes (C < D)) \parr ((E < F) \otimes (G < H)) \\ &\parr (A^* < H^*) \parr (E^* < B^*) \parr (G^* < D^*) \parr (C^* < F^*)\end{aligned} \tag{3.52}$$

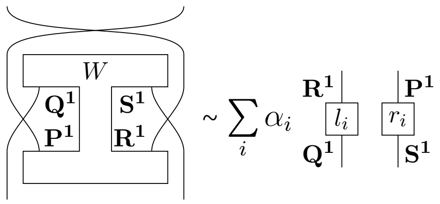
We could achieve the same result by just labelling the variables as first-order in the following way:

$$\begin{aligned}\vdash &\left(\left(P^1 \multimap Q^1\right) \otimes \left(R^1 \multimap S^1\right)\right)^* \\ &\multimap \left(\left(Q^1 \multimap R^1\right) \otimes \left(S^1 \multimap P^1\right)\right)\end{aligned} \tag{3.53}$$

Using the equality $A^1 \multimap B^1 = A^{1*} < B^1$ (Equation 2.158), this is equivalent to:

$$\begin{aligned}\vdash &\left(\left(P^{1*} < Q^1\right) \otimes \left(R^{1*} < S^1\right)\right)^* \\ &\multimap \left(\left(Q^{1*} < R^1\right) \otimes \left(S^{1*} < P^1\right)\right)\end{aligned} \tag{3.54}$$

One can verify that the formulae of Equations 3.54 and 3.52 are precisely related by pom$(-)$ translation, making this also a theorem of causal logic. Using the simpler form in Equation 3.53, we recognise this as the statement: “any process matrix can be represented as a non-signalling first-order process by swapping the outputs”.



(3.55)

3.4.3 The First-Order Inductive Fragment

Another interesting fragment of causal formulae goes in the other direction - instead of replacing all first-order atoms with regular ones, let us phrase everything in terms of first-order atoms using the grammar $F, G ::= A^1 \mid A^{1*} \mid F \otimes G \mid F \parr G$. Since first-order objects are some of the easiest to attribute physical meaning to, this sets up our statements of causal consistency to potentially have simpler intuitions, similar to what was achieved in the previous section. Inductively-defined frameworks that build up a hierarchy on top of first-order systems are also well-understood in the quantum causality literature, such as in the comb framework [22] or Bisio and Perinotti’s higher-order operational theory for quantum processes [15].

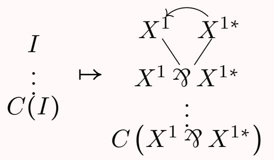
We have already shown that causal consistency of a formula is independent of the specific objects chosen by an interpretation function in the Causal Characterisation Theorem. Therefore, it should be sufficient to consider interpretations that map regular variables to channels over first-order objects $\mathbf{Y^1} \multimap \mathbf{X^1} \cong \mathbf{Y^{1*}} \parr \mathbf{X^1}$ since the channel types are neither first-order nor first-order dual. In terms of proof-structures, we can replace any regular axiom link with a pair of FO-axiom links in opposite directions which recovers connectivity in both directions.

One-way signalling processes are spanned by those which factorise by the first-order system $\mathbf{2}$ by the Affine-Bit Sufficiency Theorem; that is, any state of $\mathbf{A} < \mathbf{B}$ can be represented as a state of $(\mathbf{A} \parr \mathbf{2}) \otimes (\mathbf{2}^* \parr \mathbf{B})$ contracted along $\mathbf{2}$. Making this contraction explicit in our proof-structure introduces a new FO-axiom link to control the direction of information flow between $\mathbf{A}$ and $\mathbf{B}$. Working with right-sided sequents and black box effects, we dualise this to replace $\mathbf{A} < \mathbf{B}$ with $(\mathbf{A} \otimes \mathbf{2}^*) \parr (\mathbf{2} \otimes \mathbf{B})$.

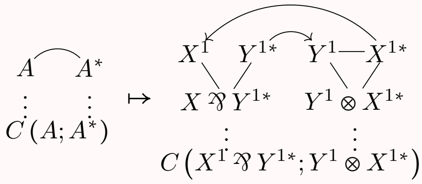
Again, we formalise these ideas as a rewrite procedure.

Definition 3.4.5

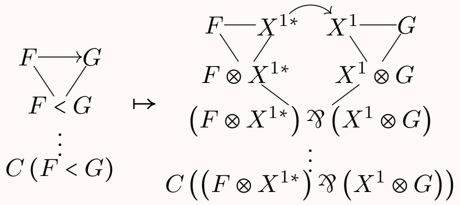
Any proof-structure P induces a proof-structure $\mathrm{fo}(P)$ with no unit, regular axiom, or seq links by applying the following rewrites exhaustively:



(3.56)



(3.57)

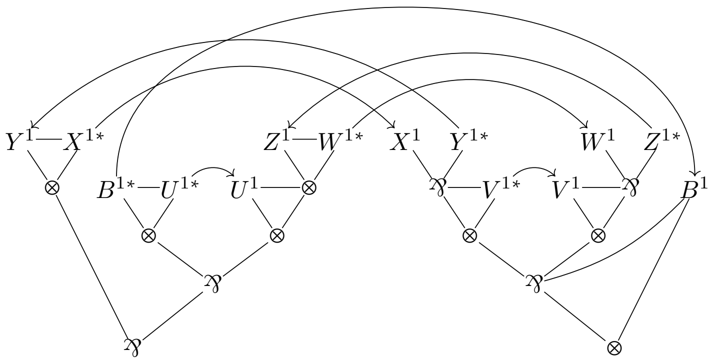


(3.58)

where X^1 and Y^1 are fresh first-order variable names.

Example 3.4.6

Applying fo $(-)$ to the proof-net of Equation 3.25 expands the two regular axiom links and the two seq links.



(3.59)

$$\left(Y^1 \otimes X^{1*}\right) \parr \left(B^{1*} \otimes U^{1*}\right) \parr \left(U^1 \otimes Z^1 \otimes W^{1*}\right) \parr \left[\left(\left[\left(X^1 \parr Y^{1*}\right) \otimes V^{1*}\right] \parr \left[V^1 \otimes \left(W^1 \parr Z^{1*}\right)\right]\right) \otimes B^1\right] \tag{3.60}$$

Proposition 3.4.7

For any causal proof-structure P, it is a causal proof-net iff fo (P) is a causal proof-net.

Proof. Similar to Proposition 3.4.4 □

Sufficiency shows that no more logical or structural behaviour exists in Caus $[\mathcal{C}]$ than in an inductively-defined framework. Despite this, Caus $[\mathcal{C}]$ retains its flexibility over such alternatives because some of the additional objects still have important physical or computational interpretations - for example, in Caus $[\mathrm{CP}^*]$ we can build qubit objects where the causal effects coincide with postselection within a chosen axis/plane (and convex combinations of these) which cannot be formed inductively but is useful for Measurement-Based Quantum Computing to describe the space of measurements whose errors can be corrected (see Section 4.4.1).

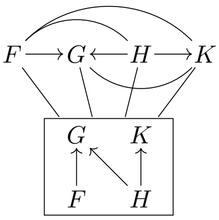
Proof-nets over the first-order inductive fragment resembles a variant of polarised MLL+Mix where only the atoms are polarised. Despite linear time verification of MLL+Mix proof-nets [92], by faithfully encoding all causal formulae (and hence pomset) we find that the seemingly small change of directing the axiom links jumps the complexity all the way to coNP-complete again.

3.5 Extensions

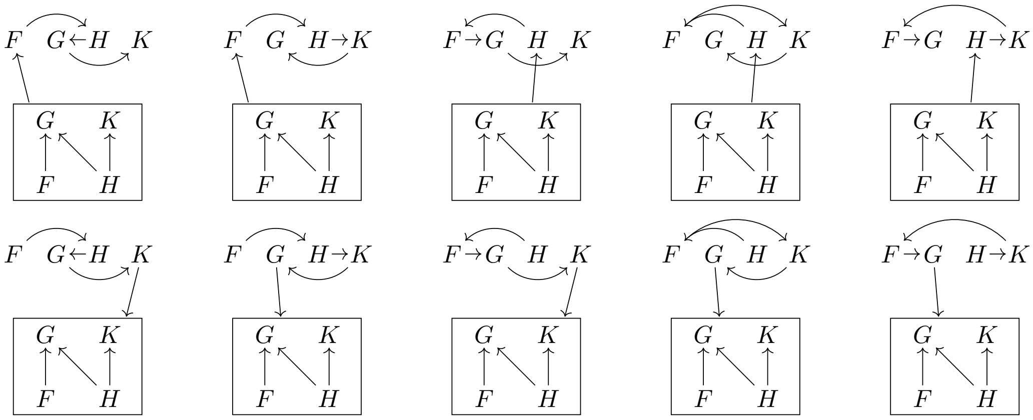
In the other direction from looking at taking fragments of the grammar of causal formulae, we could look at expanding the grammar to include arbitrary graph types, additives, unions, and intersections, and ask how causal logic should be extended to remain sound and complete.

For unions and intersections, we can always lift these to the top level of a formula (by Setwise Distributivity and Proposition 2.8.24) at which point they are just existential and universal quantification over other formulae/proof-structures. Semantically sensible formulae containing unions and intersections may involve duplicates of atoms of the same name, so we would need to carefully adjust the notion of balanced formulae to guarantee that each branch of this quantification yields balanced causal formulae.

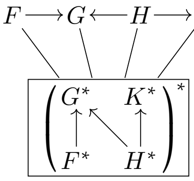
For graph types, we can either add them directly as primitives (prime graphs would be sufficient as in GV [3]) or derive them as intersections of existing types. To add graph types as primitive n-ary operators, we can generalise seq links to n premises and switch up or down to the conclusion along with switching over every linear ordering of the graph (in line with Proposition 2.8.23 and generalising the switchings of tensor links).



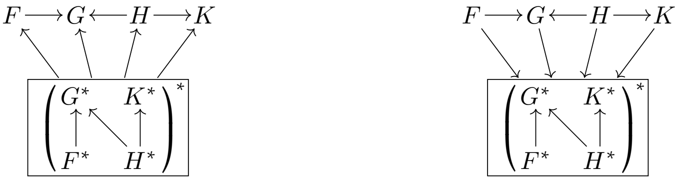
(3.61)



For their dual types, we again switch up or down to the conclusion, but have a static version of the graph connecting the premises (generalising the switchings of seq links).



(3.62)



The proof of the Switching Lemma generalises using Proposition 2.8.26.

Adding additives to causal logic can give some surprising features not present in the other logics discussed in this thesis. To elaborate on this, we need to introduce the *time-symmetry* property of a logic.

Definition 3.5.1: Time-symmetry

For any formula F, let its *time-reversal* $\text{rev}(F)$ be the formula obtained by replacing each instance of $<$ in F with $>$ and vice versa. A logic is *time-symmetric* if each formula F is provable iff $\text{rev}(F)$ is provable.

Proposition 3.5.2

pomset, BV, and MAV are all time-symmetric.

Proof. For pomset logic, proof structures of F and $\mathrm{rev}\,(F)$ are identical up to reversing the direction of seq links. Given an alternating elementary circuit in one, we find a corresponding alternating elementary circuit in the other by reversing the circuit.

For BV, the obvious inductive argument works by looking at each rule instance, e.g. deriving $F \to F'$ and deriving a rule instance for the time-reversal $\mathrm{rev}\,(F) \to \mathrm{rev}\,(F')$. To achieve this, we apply the time-reversal procedure to the context - deep inference does not care about the form of a context, so this preserves the ability to apply rules within it - and then look at the rules with trivial contexts and reverse each of them. The only rule that actually includes < is $q\downarrow$, where time-reversal transforms

$$\mathrm{q}\downarrow\frac{(F \parr G) < (H \parr K)}{(F < H) \parr (G < K)}$$

into

$$\mathrm{q}\downarrow\frac{\begin{array}{c}(F \parr G) > (H \parr K)\\ \hdashline (K \parr H) < (G \parr F)\end{array}}{\begin{array}{c}(K < G) \parr (H < F)\\ \hdashline (F > H) \parr (G > K)\end{array}}$$

Since each rule can be reversed, so can entire derivations.

MAV follows similarly to BV, where we additionally give a time-reversal for $m\downarrow$ in the same way. □

Given the close relationship between these logics and causal logic, it may be surprising to find that extending causal logic with additives would not be time-symmetric, but this is for good reason. When discussing distributivity laws in Remark 2.7.10, seq preserves internal and external choice from the past in both directions, but additives only distribute in one direction in the future. This occurs because of our interpretation of < as one-way signalling which is fundamentally asymmetric - choices in the past must be independent of future observations, but choices in the future may depend on observations in the past.

Proposition 3.5.3

Any faithful extension of causal logic with additives will *not* be time-symmetric.

Proof. In Remark 2.7.10, we demonstrated that the following isomorphism holds:

$$(\mathbf{A} \oplus \mathbf{B}) < \mathbf{C} \cong (\mathbf{A} < \mathbf{C}) \oplus (\mathbf{B} < \mathbf{C}) \tag{3.63}$$

In particular, this means that the formula

$$((A \oplus B) < C) \multimap ((A < C) \oplus (B < C)) \tag{3.64}$$

should be a valid theorem of any extension of causal logic with additives that is faithful to causal consistency. The time-reversal of this is

$$((A \oplus B) > C) \multimap ((A > C) \oplus (B > C)) \tag{3.65}$$

but Remark 2.7.10 also established that the corresponding distributor fails to be causal:

$$(\mathbf{A} \oplus \mathbf{B}) > \mathbf{C} \not\to (\mathbf{A} > \mathbf{C}) \oplus (\mathbf{B} > \mathbf{C}) \tag{3.66}$$

□

Remark 3.5.4

In Example 3.4.3, we showed that the equations characterising first-order objects can be encoded and proved in BV. Despite the simplicity of the proof of Proposition 2.5.6, showing that coproducts preserve the first-order property, we will demonstrate here that a statement for this property cannot be proved in MAV.

We haven't defined the pom $(-)$ transformation in a way that works nicely for additives, but we can argue that Proposition 2.5.6 can be boiled down to causal consistency of:

$$((A^* < B) \times (C^* < D)) \multimap ((A^* \times C^*) < (B \oplus D)) \tag{3.67}$$

Suppose we have an interpretation where $\Phi(A) = \Phi(B)$ and $\Phi(C) = \Phi(D)$. When these are first-order objects, we can interpret black boxes of $A^* < B$ and $C^* < D$ as cups/identity wires. The identity on the coproduct $\Phi(A \oplus C) = \Phi(B \oplus D)$ is formed by applying the relevant identity on each branch, which we can do since we are provided the identities as a × pair. The formula then says that this resulting identity (or tagged copairing in general) is one-way signalling, i.e. $\Phi\,(A \oplus C)$ is first-order.

Suppose, on the contrary, that this formula can be derived in MAV. Using the Cut rule, this derivation could equivalently be presented as

$$\dfrac{\dfrac{(A^* \oplus C^*) < (B \times D)}{\vdots}}{(A^* < B) \oplus (C^* < D)}$$

which we use to derive the following distribution law:

$$\dfrac{\dfrac{I\downarrow\dfrac{}{I}}{\mathrm{s}\,\dfrac{\left(\mathrm{e}\downarrow\dfrac{\mathrm{t}\downarrow\dfrac{I}{\left(\mathrm{ai}\downarrow\dfrac{I}{A^* \parr \left(\mathrm{l}\downarrow\dfrac{A}{A \oplus B}\right)}\right) \times \left(\mathrm{r}\downarrow\dfrac{B}{A \oplus B}\right)}}{(A^* \times B^*) \parr (A \oplus B)}\right) < \left(\mathrm{e}\downarrow\dfrac{\mathrm{t}\downarrow\dfrac{I}{\left(\mathrm{ai}\downarrow\dfrac{I}{C^* \parr C}\right) \times \left(\mathrm{ai}\downarrow\dfrac{I}{C^* \parr C}\right)}}{C^* \parr (C \times C)}\right)}{((A^* \times B^*) < C^*) \parr \left(\dfrac{\dfrac{(A \oplus B) < (C \times C)}{\vdots}}{(A < C) \oplus (B < C)}\right)}}}{((A \oplus B) < C) \multimap ((A < C) \oplus (B < C))}$$

By time-symmetry of MAV (Proposition 3.5.2), this would imply that we could also prove the time-reversal, $((A \oplus B) > C) \multimap ((A > C) \oplus (B > C))$. By Proposition 3.2.11, this would imply a corresponding statement of causal consistency in $\mathrm{Caus}\,[\mathcal{C}]$ which contradicts our counterexample in Proposition 3.5.3.

Unpacking this contradiction, we conclude that MAV cannot establish the fact that $\oplus$ preserves the first-order property.

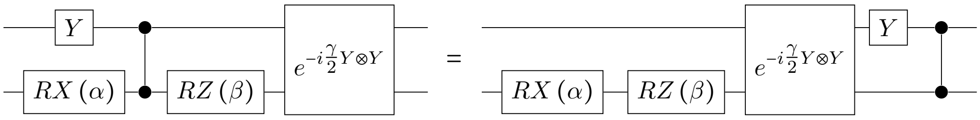
We leave the formulation of causal proof-nets with additives for future work. Methods already exist for incorporating additives into proof-nets for MALL [55, 70], and we conjecture that incorporating these into the framework of causal proof-nets should continue to faithfully capture causal consistency, but this deserves explicit verification.

Chapter 4

Causal Structures in Quantum Computing

So far, we have handled causal structures in a very abstract setting with most of our motivating examples coming from quantum foundations and arranging systems throughout spacetime, and inferring temporal relations by the capacity to signal information. Such definitions require us to have open systems, where each point of interest can both be controlled and observed in order to send and detect information flowing through the process. In many computational settings, however, the points of interest can be simple events (such as an instruction being executed, or a message sent between internal subsystems) with the only possible observation being the order in which the events occurred. We can similarly build partial orders over the events to describe whether two events always occur in a fixed order, or if it is possible for them to occur in either order. This chapter looks at a few of these settings from the field of quantum computing where the events have a degree of programmability - some parameter that we can control to influence how the event is performed - and work towards some elementary results linking the different partial orders together.

Within the *quantum circuit* model, each program consists of a collection of *gates* which are applied to qubits to update their states in-place. Whenever the unitary matrices of two gates *commute*, we can apply them to the qubits in either order to achieve the same outcome.

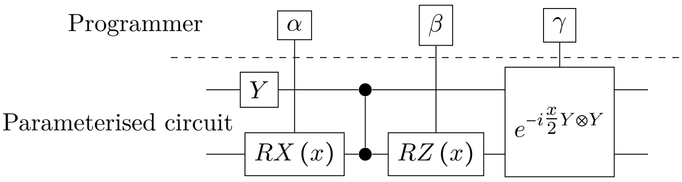

Y
$RX\,(\alpha)$
$RZ\,(\beta)$
$e^{-i\frac{\gamma}{2}Y\otimes Y}$
$=$
$RX\,(\alpha)$
$RZ\,(\beta)$
$e^{-i\frac{\gamma}{2}Y\otimes Y}$
Y

(4.1)

We can therefore build a partial order over the gates, describing whether there exists a sequence of valid commutations that reorders each pair of gates. Equivalently, whenever two gates do not commute, we record that one must occur before the other, and we take the transitive closure of this relation as the partial order.

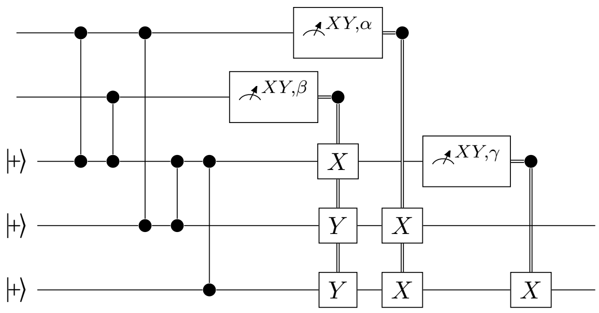
$$\begin{tikzcd}[row sep=small] Y \arrow[r] & CZ & \\ RX \arrow[r] \arrow[ur] & RZ \arrow[r] & e^{-i\frac{\gamma}{2}Y\otimes Y} \end{tikzcd} \tag{4.2}$$

Suppose that the gates are now parameterised, e.g. by the angle of rotation. We can then describe the parameterised gate as some linear map taking both the initial qubit state and a state encoding the parameter, and giving the final qubit state. This opens up our representation of the gate to give a system on which we can both input information through choosing the parameter, or observe this opening.



(4.3)

We can do a similar trick with *measurement patterns* in Measurement-Based Quantum Computing (MBQC) [99]. On their own, they look like special kinds of circuits that prepare a fixed state and then perform measurements and conditional corrections where the programmer may choose the angles of measurements to influence the linear map that is implemented.

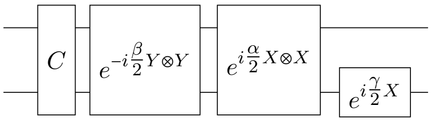


(4.4)

Similar to commuting gates, it may be possible to perform some measurements simultaneously/in either order. This is described in the partial order of a *flow* [38, 19] for the pattern, which also verifies the corrections to ensure the pattern implements a single linear map deterministically (i.e. regardless of the measurement outcomes). If we assume that corrections exist, we can instead treat the fixed resource state as our open process, onto which the programmer specifies a choice of effect to apply locally to each measured qubit.

This chapter builds up to two major new results. Firstly, we give a new algorithm for the *circuit extraction* [7] problem: given a measurement pattern, efficiently identify a pure quantum circuit which implements the same linear map (see the Circuit

Extraction Theorem). It produces the circuit as a Clifford map C followed by a rotation for each measurement, where the basis of rotation is given immediately by the corrections over the output qubits; for example, the pattern of Equation 4.4 can be extracted as:


C
$e^{-i\frac{\beta}{2}Y\otimes Y}$
$e^{i\frac{\alpha}{2}X\otimes X}$
$e^{i\frac{\gamma}{2}X}$

(4.5)

In addition to this algorithm providing potential practical benefits for automated conversion in quantum compilers and pedagogical benefits for understanding the equivalence between the paradigms, it also guarantees that the partial order of commutation between the rotations exactly matches the flow partial order.

The second result, the Flow Causality Theorem, uses this extraction algorithm to show that the open process of the resource state/parameterised circuit obeys all non-signalling conditions of the same partial order, since we can extract out and discard all future measurement angles to give a unique marginal over the past systems.

As with the other chapters, we begin with some introductory material to introduce MBQC, assuming the reader is already familiar with the more common paradigm of quantum circuits. The extraction algorithm will make heavy use of a particular kind of flow called *focussed Pauli flow*, so the first novel results focus around identifying such a flow for any pattern in Section 4.2. Section 4.3 builds up the algorithm step-by-step, finishing with a helpful visualisation of the extraction procedure using the ZX-calculus [30, 114] which some readers may prefer over the linear algebra notation used throughout the rest of the chapter. The chapter ends with a short discussion on the connection to non-signalling conditions.

4.1 Background: MBQC

The one-way model follows the typical structure of Measurement-Based Quantum Computing (MBQC) of building some resource state which is then consumed by single-qubit measurements. The particular resources considered are constructed by matching qubits (inputs and ancillas prepared in the $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ state) with vertices in a graph, and entangling them (with a CZ gate) according to the edges. The measurements are generally non-deterministic so, in order to have a deterministic effect overall, local gates can be applied to the remaining qubits to counteract the difference between the projectors for each outcome, giving the net effect of postselecting the desired outcome. This background section will build up this description formally and detail the main mechanisms used for determining how to correct for measurement errors.

4.1.1 Measurement Patterns

Single qubit measurements are characterised by the bases they project into, which themselves can be described by a vector in the Bloch sphere. For MBQC, we conventionally restrict measurements into a plane of the Bloch sphere spanned by two of the Pauli bases. Such planar measurement bases are described by the following projectors for $\alpha \in [0, 2\pi)$:

$$\begin{aligned}
\langle +_{XY,\alpha}| &= \tfrac{1}{\sqrt{2}}\left(\langle 0| + e^{-i\alpha}\langle 1|\right) & \langle -_{XY,\alpha}| &= \tfrac{1}{\sqrt{2}}\left(\langle 0| - e^{-i\alpha}\langle 1|\right) \\
\langle +_{XZ,\alpha}| &= \cos\left(\tfrac{\alpha}{2}\right)\langle 0| + \sin\left(\tfrac{\alpha}{2}\right)\langle 1| & \langle -_{XZ,\alpha}| &= \sin\left(\tfrac{\alpha}{2}\right)\langle 0| - \cos\left(\tfrac{\alpha}{2}\right)\langle 1| \\
\langle +_{YZ,\alpha}| &= \cos\left(\tfrac{\alpha}{2}\right)\langle 0| - i\sin\left(\tfrac{\alpha}{2}\right)\langle 1| & \langle -_{YZ,\alpha}| &= \sin\left(\tfrac{\alpha}{2}\right)\langle 0| + i\cos\left(\tfrac{\alpha}{2}\right)\langle 1|
\end{aligned} \tag{4.6}$$

with the following special cases for Pauli measurements for $\alpha = a\pi$ ($a \in \{0, 1\}$):

$$\begin{aligned}
\langle +_{X,a\pi}| &= \tfrac{1}{\sqrt{2}}(\langle 0| + (-1)^a\langle 1|) & \langle -_{X,a\pi}| &= \tfrac{1}{\sqrt{2}}(\langle 0| - (-1)^a\langle 1|) \\
\langle +_{Y,a\pi}| &= \tfrac{1}{\sqrt{2}}(\langle 0| - i(-1)^a\langle 1|) & \langle -_{Y,a\pi}| &= \tfrac{1}{\sqrt{2}}(\langle 0| + i(-1)^a\langle 1|) \\
\langle +_{Z,a\pi}| &= (1-a)\langle 0| + a\langle 1| & \langle -_{Z,a\pi}| &= a\langle 0| + (1-a)\langle 1|
\end{aligned} \tag{4.7}$$

For any measurement basis, the negative outcome at angle α is equivalent to the positive outcome at angle $\alpha + \pi$. We usually treat the positive measurement outcome as the desired branch, i.e. the projector we want to apply to achieve our desired end state.

For the purposes of this thesis, we will consider programming in MBQC to be the act of specifying the angles of measurement and providing an input state to some otherwise fixed protocol called a *measurement pattern*, implementing a parameterised linear map. This pattern will perform a pre-determined sequence of entanglement, measurement, and correction operations on the qubits along with some ancillas and yield some output state.

Definition 4.1.1: Measurement pattern [39, Definition 1][19]

A *measurement pattern* consists of a collection V of qubits with distinguished subsets $I, O \subseteq V$ of inputs and outputs, and a sequence of commands from:

- Preparations - initialising a qubit $u \notin I$ to $|+\rangle$;
- Entangling operators - applying a CZ gate between distinct qubits $u, v \in$

V;

- Destructive measurements - applying $\langle +_{\lambda,\alpha}|$ to qubit $u \notin O$ on outcome 0 or $\langle -_{\lambda,\alpha}|$ on outcome 1;
- Corrections - conditionally applying an X gate or a Z gate to qubit $u \in V$ if the outcome of the measurement for qubit v was 1.

A measurement pattern is *runnable* if additionally:

- All non-input qubits are prepared exactly once;
- A non-input qubit is not acted on by any other command before its preparation;
- All non-output qubits are measured exactly once;
- A non-output qubit is not acted on by any other command after its measurement;
- No correction depends on an outcome not yet measured.

We denote non-input (prepared) qubits as $\overline{I} := V \setminus I$ and non-output (measured) qubits as $\overline{O} := V \setminus O$.

A measurement pattern is *uniformly*, *strongly*, and *stepwise deterministic* if, for any choice of angles $\alpha : \overline{O} \to [0, 2\pi)$ (uniform), performing each individual measurement and its associated corrections (stepwise) applies the same linear map on each measurement outcome (strong) up to global phase.

It is most useful to visualise a measurement pattern (minus its corrections) as a *labelled open graph* Γ, whose vertices represent qubits labelled by their measurement planes and the edges indicate the entanglement. Herein, "vertex" and "qubit" will be treated as interchangeable.

Definition 4.1.2: Labelled open graph [19]

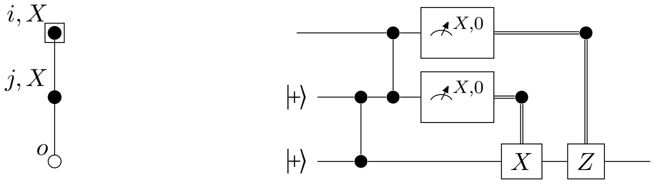
A *labelled open graph* is a tuple $\Gamma = (G, I, O, \lambda)$ where $G = (V, E)$ is an undirected graph, $I, O \subseteq V$ are (possibly overlapping) subsets of vertices representing inputs and outputs respectively, and $\lambda : \overline{O} \to \{XY, XZ, YZ, X, Y, Z\}$ is a labelling function assigning a measurement plane or Pauli to each non-output vertex.

When drawing a labelled open graph, we will distinguish between measured and output vertices as filled and unfilled points respectively, and inputs are specified by boxes around the vertices.

$u \sim_G v$ denotes vertices $u, v \in V$ being adjacent in G. Neighbour sets are given by $N_G(u) := \{w \in V \mid w \sim_G u\}$ and odd neighbourhoods by $\mathrm{Odd}_G(A) := \{w \in V \mid |N_G(w) \cap A| \text{ is odd}\}$. We may drop the subscript when the graph G is obvious from the context.

Example 4.1.3

The circuit below describes the quantum teleportation protocol [14]. We can view this as a simple example of a measurement pattern with fixed measurement angles, as described by the labelled open graph on the left.



(4.8)

Since runnable patterns can be standardised to perform all initialisations first, then all entangling gates, and finally alternate measurements and corrections, the linear map implemented by the pattern also has a standard representation given by the *intended branch* (where all measurement outcomes are 0 and hence no corrections are required).

Definition 4.1.4: Linear map of a pattern [19, Theorem 1]

The linear map associated with a measurement pattern Γ applied with angles α is given by

$$\left(\prod_{u \in \overline{O}} \langle +_{\lambda(u), \alpha(u)}|_u\right)\left(\prod_{u \sim v} CZ_{u,v}\right)\left(\prod_{u \in \overline{I}} |+\rangle_u\right) \tag{4.9}$$

Subscripts indicate a map is applied to the given qubit(s) alongside the identity on all others.

For any input state $|\psi\rangle$, the state

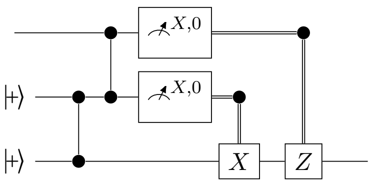
$$\left(\prod_{u \sim v} CZ_{u,v}\right)\left(\prod_{u \in \overline{I}} |+\rangle_u\right)|\psi\rangle_I \tag{4.10}$$

is referred to as the *resource state* [13].

4.1.2 Stabilizers and Corrections

So far we have brushed over how the corrections in a pattern can allow us to recover the same linear map whenever we get a measurement error. This concept can already be seen in the *quantum teleportation* protocol. If we review this example through the lens of stabilizer theory, we can see how the same method can be generalised to correct measurements on other resource states.

Let's look at the concrete example of the measurement pattern for the quantum teleportation protocol [14] from Example 4.1.3.



(4.11)

Both the preparations and entangling gates are Clifford. Using the rules for stabilizers

$$|+\rangle_u = X_u |+\rangle_u \tag{4.12}$$

$$CZ_{uv} X_u = X_u Z_v CZ_{uv} \tag{4.13}$$

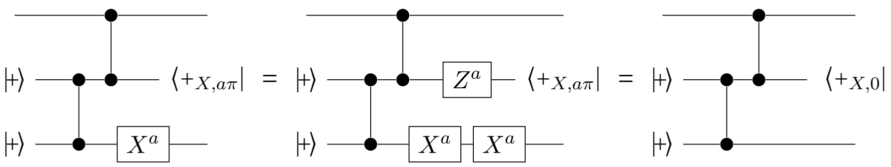
we can obtain a stabilizer per initialised vertex $w \in \overline{I}$:

$$\begin{aligned}\left(\prod_{u\sim v} CZ_{u,v}\right)\left(\prod_{u\in\overline{I}} |+\rangle_u\right) &= \left(\prod_{u\sim v} CZ_{u,v}\right) X_w \left(\prod_{u\in\overline{I}} |+\rangle_u\right) \\ &= \left(\prod_{v\in N_G(w)} Z_v\right) X_w \left(\prod_{u\sim v} CZ_{u,v}\right)\left(\prod_{u\in\overline{I}} |+\rangle_u\right)\end{aligned} \tag{4.14}$$

In the teleportation case, this gives stabilizers $Z_i X_j Z_o$ and $I_i Z_j X_o$. As stabilizers, we can freely introduce these to mathematical expressions without performing any corresponding physical operation. In particular, the Z_j in $I_i Z_j X_o$ will flip the projector of the X basis measurement, allowing us to deduce that the two branches give the same linear map. In this derivation we use a binary variable $a \in \{0, 1\}$ to specify the

[13] They are commonly also referred to in the literature as *graph states*, but we use the alternative name to avoid confusion with Definition 2.8.2.

outcome of the measurement, and therefore P^a evaluates to either $P^0 = I$ or $P^1 = P$ representing the conditional application of a Pauli P.

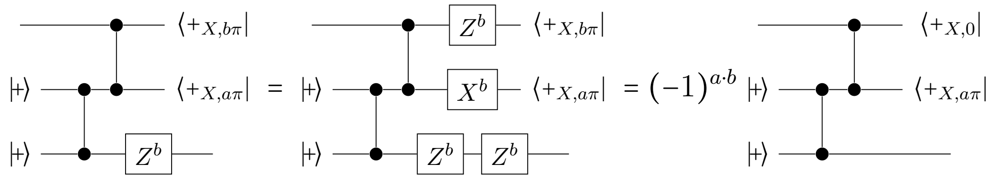


(4.15)

To correct the measurement on qubit i, we can similarly use the stabilizer $Z_i X_j Z_o$ which will flip the measurement outcome. If we reordered the measurements to handle this qubit first, we could apply X_j as a correction before the measurement on j. However, similar to the resource states admitting stabilizers, Paulis can be absorbed by projectors into the same basis (up to a global phase).

$$\begin{aligned} \langle +_{X,a\pi}|_u &= (-1)^a \langle +_{X,a\pi}|_u X_u \\ \langle +_{Y,a\pi}|_u &= (-1)^a \langle +_{Y,a\pi}|_u Y_u \\ \langle +_{Z,a\pi}|_u &= (-1)^a \langle +_{Z,a\pi}|_u Z_u \end{aligned} \tag{4.16}$$

This means that even if qubit j has already been measured, we can still correct the measurement on i. Some people like to think of this as an “implicit correction in the past”.

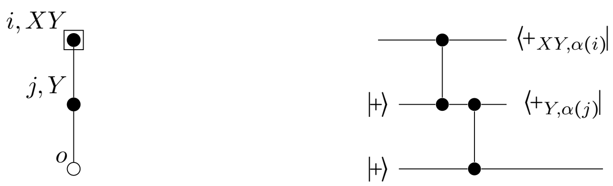


(4.17)

We can only correct in the past in this way on qubits that were measured in a Pauli basis. Applying any Pauli before a planar projector will generally update the angle either by π (the same as reintroducing a measurement error), negating it, or both.

$$\begin{aligned} \langle +_{XY,\alpha}|_u Z_u &= \langle +_{XY,\alpha+\pi}| \\ \langle +_{XY,\alpha}|_u X_u &= \langle +_{XY,-\alpha}| \\ \langle +_{XY,\alpha}|_u Y_u &\approx \langle +_{XY,-\alpha+\pi}| \end{aligned} \tag{4.18}$$

Suppose instead we changed the pattern to use different measurement labels and wish to infer what corrections to apply.



(4.19)

The same stabilizers would work to correct the measurement errors, but the X_j in $Z_i X_j Z_u$ can no longer be absorbed and must be performed explicitly. Having to perform the measurements in order increases the time for computations, so it would be better if we could avoid it. Thankfully, the stabilizers of the resource state form a group so we can combine them to give $Z_i Y_j Y_u$ where Z_i still corrects the measurement on i and Y_j can be absorbed by the Y basis measurement on j.

4.1.3 Flow

A *flow* for a labelled graph is a data structure that captures this use of stabilizers to propagate errors from unwanted measurement outcomes forward to the rest of the circuit in order to correct them, aiming for stepwise determinism. Flow is typically expressed as a purely graph-theoretic property, abstracting away any complexity from the underlying quantum processes. The different types of flow conditions start with a very basic definition that will only work for special cases of labelled graphs, and gradually incorporate more of the ideas from the previous examples to correct on more general graphs and account for more possible correction schemes.

Causal flow is the simplest case where we suppose all vertices are measured in the XY basis and each error can be corrected by considering a single stabilizer of the resource state.

Definition 4.1.5: Causal flow [38, Definition 2]

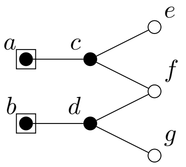
Given a labelled open graph $\Gamma = (G, I, O, \lambda)$ such that $\forall v \in \overline{O}.\ \lambda(v) = XY$, a *causal flow* for Γ is a tuple $(f, \prec)$ of a map $f : \overline{O} \to \overline{I}$ and a strict partial order $\prec$ over V such that for all $v \in \overline{O}$:

- $v \sim f(v)$
- $v \prec f(v)$
- $\forall u \in N_G(f(v)).\ u = v \vee v \prec u$

The idea here is that Z_v from the stabilizer $\left(\prod_{w \in N_G(f(v))} Z_w\right) X_{f(v)}$ will eliminate the effect of the measurement error on qubit v, meaning we can correct the error by applying $\left(\prod_{w \in N_G(f(v)) \setminus \{v\}} Z_w\right) X_{f(v)}$ and implicitly invoking the stabilizer. The partial order $\prec$ indicates a required order of the measurements, ensuring that none of the vertices required for correcting v have been measured yet.

Example 4.1.6: [38, Example 2]

The labelled open graph below admits the causal flow described in the table.



v	$\lambda(v)$	$f(v)$	$N_G(f(v))$	$\{u \mid v \prec u\}$
a	XY	c	a, e, f	c, e, f
b	XY	d	b, f, g	d, f, g
c	XY	e	c	e
d	XY	g	d	g

The first row states that the correction of the measurement at a uses the stabilizer $X_c Z_a Z_e Z_f$, where the Z_a flips the projector at a ("correcting" the measurement outcome) and the remaining terms must be cancelled out by explicitly applying the inverse operations on those qubits. This requires c, e, and f to be unmeasured at this point, and the final column states this with $a \prec c, e, f$.

Generalised flow takes a similar approach, but allows us to take combinations of the basic stabilizers. If the stabilizer of a candidate $f(v)$ would require a Z correction on some $u \in N_G(f(v))$ that has already been measured, there may exist some other stabilizer we can apply that cancels out the Z for us. Now, instead of the stabilizer being determined by a single vertex $f(v) \in \overline{I}$, we have a set $g(v) \subseteq \overline{I}$.

$$\begin{aligned}\prod_{u \in g(v)} \left(\prod_{w \in N_G(u)} Z_w \right) X_u &\approx \left(\prod_{u \in g(v)} \prod_{w \in N_G(u)} Z_w \right) \left(\prod_{u \in g(v)} X_u \right) \\ &= \left(\prod_{u \in \mathrm{Odd}(g(v))} Z_u \right) \left(\prod_{u \in g(v)} X_u \right) \\ &\approx \left(\prod_{u \in \mathrm{Odd}(g(v)) \setminus g(v)} Z_u \right) \left(\prod_{u \in g(v) \cap \mathrm{Odd}(g(v))} Y_u \right) \left(\prod_{u \in g(v) \setminus \mathrm{Odd}(g(v))} X_u \right)\end{aligned} \tag{4.20}$$

We can summarise this stabilizer as

$$\prod_u P_u^{g(v) \to u} \tag{4.21}$$

where $P^{g(v) \to u}$ is the Pauli induced on u when correcting v:

$$P^{g(v) \to u} := \begin{cases} I & u \notin g(v) \cup \mathrm{Odd}(g(v)) \\ X & u \in g(v) \setminus \mathrm{Odd}(g(v)) \\ Y & u \in g(v) \cap \mathrm{Odd}(g(v)) \\ Z & u \in \mathrm{Odd}(g(v)) \setminus g(v) \end{cases} \tag{4.22}$$

By relaxing these restrictions on the stabilizers used, we can also generate Y or X on the measured vertex, allowing the correction of measurements in the XZ and YZ planes respectively.

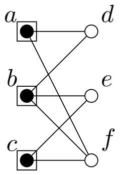
Definition 4.1.7: Generalised flow [19, Definition 3]

Given a labelled open graph $\Gamma = (G, I, O, \lambda)$ with planar labels ($\forall v \in \overline{O}.\ \lambda(v) \in \{XY, XZ, YZ\}$), a *generalised flow* (or *gflow*) for Γ is a tuple $(g, \prec)$ of a map $g : \overline{O} \to \mathcal{P}\left[\overline{I}\right]$ and a strict partial order $\prec$ over V such that for all $v \in \overline{O}$:

- $\forall u \in g(v)\,.\ v \neq u \Longrightarrow v \prec u$
- $\forall u \in \mathrm{Odd}(g(v))\,.\ v \neq u \Longrightarrow v \prec u$
- $\lambda(v) = XY \Longrightarrow v \in \mathrm{Odd}(g(v)) \setminus g(v)$
- $\lambda(v) = XZ \Longrightarrow v \in g(v) \cap \mathrm{Odd}(g(v))$
- $\lambda(v) = YZ \Longrightarrow v \in g(v) \setminus \mathrm{Odd}(g(v))$

Example 4.1.8: [19, Example 2]

Even though all measured vertices are labelled XY, the following labelled graph does not admit a causal flow but it does have gflow.



v	$\lambda(v)$	$g(v)$	$\mathrm{Odd}(g(v))$	$\{u \mid v \prec u\}$
a	XY	d	a, b	b, c, d, e, f
b	XY	e	b, c	c, d, e, f
c	XY	d, f	c	d, f

When all measurement labels are planar, every uniformly, strongly, and stepwise deterministic measurement pattern is induced by a gflow [19, 89].

There exist polynomial-time algorithms for identifying whether a labelled open graph admits causal flow or gflow [41, 88, 7], and for extracting an equivalent unitary circuit for the measurement pattern given either type of flow [42, 7].

The final form we will look at in detail is Pauli flow, which can handle both planar measurements $\lambda(v) \in \{XY, XZ, YZ\}$ and special cases for Pauli measurements $\lambda(v) \in \{X, Y, Z\}$. Similar to gflow, it gives a stabilizer that applies a Pauli X to vertices in $p(v)$ and a Pauli Z to vertices in $\mathrm{Odd}(p(v))$. However, because Equation 4.16 means Pauli corrections have no observable effect if the qubit is measured in the same basis, it doesn't matter if, for example, a vertex $u \in p(v) \setminus \mathrm{Odd}(p(v))$

with $\lambda(u) = X$ has already been measured before v. We saw this idea in play in the example of quantum teleportation to eliminate any temporal dependency between the two measurements and allow them to be performed simultaneously.

Additionally, whilst there is only one Pauli that will reliably correct planar measurements uniformly, Pauli flow recognises that each Pauli measurement lives in two such planes. For example, a measurement with $\lambda(v) = X$ can be flipped by either a Z or a Y - we don't care whether or not $v \in p(v)$ so long as $v \in \mathrm{Odd}(p(v))$.

Definition 4.1.9: Pauli flow [19, Definition 5]

Given a labelled open graph $\Gamma = (G, I, O, \lambda)$, a *Pauli flow* for Γ is a tuple $(p, \prec)$ of a map $p : \overline{O} \to \mathcal{P}\left[\overline{I}\right]$ and a strict partial order $\prec$ over V such that for all $v \in \overline{O}$:

$$[\prec .X] \quad \forall u \in p(v) \,.\, v \neq u \wedge \lambda(u) \notin \{X, Y\} \implies v \prec u$$

$$[\prec .Z] \quad \forall u \in \mathrm{Odd}(p(v)) \,.\, v \neq u \wedge \lambda(u) \notin \{Y, Z\} \implies v \prec u$$

$$[\prec .Y] \quad \forall u \preceq v.\ v \neq u \wedge \lambda(u) = Y \implies u \notin p(u) \Delta \mathrm{Odd}(p(u))$$

$$[\lambda.XY] \quad \lambda(v) = XY \implies v \in \mathrm{Odd}(p(v)) \setminus p(v)$$

$$[\lambda.XZ] \quad \lambda(v) = XZ \implies v \in p(v) \cap \mathrm{Odd}(p(v))$$

$$[\lambda.YZ] \quad \lambda(v) = YZ \implies v \in p(v) \setminus \mathrm{Odd}(p(v))$$

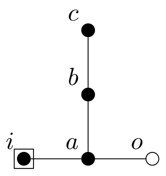
$$[\lambda.X] \quad \lambda(v) = X \implies v \in \mathrm{Odd}(p(v))$$

$$[\lambda.Z] \quad \lambda(v) = Z \implies v \in p(v)$$

$$[\lambda.Y] \quad \lambda(v) = Y \implies v \in p(v) \Delta \mathrm{Odd}(p(v))$$

where $u \preceq v := \neg(v \prec u)$ and $A \Delta B := (A \setminus B) \cup (B \setminus A)$ denotes the symmetric difference of sets.

Example 4.1.10



v	$\lambda(v)$	$p(v)$	$\mathrm{Odd}(p(v))$	$\{u \mid v \prec u\}$
i	XY	a	i, b, o	b, o
a	X	o	a	o
b	XY	c	b	$-$
c	X	b, o	c	b, o

The conditions $[\prec.X]$, $[\prec.Z]$, $[\prec.Y]$ permit corrections in the past for Pauli vertices so long as they match the Pauli of the stabilizer. This can cause a phenomenon where a qubit is entirely correctable in the past, i.e. there is no observable difference in the linear maps induced by either measurement outcome. In Example 4.1.10 we see this with qubit b. Such qubits can even be measured after yielding the output or just discarded entirely.

Similar to gflow, any labelled graph with planar and Pauli measurement labels has a uniformly, strongly, and stepwise deterministic measurement pattern iff it has a Pauli flow. However, it is not the case that every such set of corrections is induced by Pauli flow, just that *some* set of corrections matches a Pauli flow. There exists an even more general description, called *extended Pauli flow* [89], which actually generates all such correction schemes, though for the scope of this thesis regular Pauli flow is sufficient to at least give us some means of correcting the widest possible range of labelled graphs.

4.2 Identifying Pauli Flow

4.2.1 An Algorithm for Pauli Flow

Here we will build up to an algorithm for identifying a Pauli flow from a graph, which follows the key principle of existing algorithms for causal flow and gflow [88, 7] of searching for a *maximally delayed* flow, where each qubit is measured as late as possible.

Definition 4.2.1: Measurement depth

Given a labelled open graph (G, I, O, λ) with a Pauli flow $(p, \prec)$, we define the vertices at *measurement depth* k under $\prec$ by

$$V_k^{\prec} := \begin{cases} O \cup \left\{u \in \overline{O} \mid \nexists v.\ u \prec v\right\} & k = 0 \\ \left\{u \in \overline{O} \setminus V_{\cup k-1}^{\prec} \mid \forall v \succ u.\ v \in V_{\cup k-1}^{\prec}\right\} & k > 0 \end{cases} \tag{4.23}$$

where the cumulative vertices up to depth k is

$$V_{\cup k}^{\prec} := \bigcup_{i \leq k} V_i^{\prec} \tag{4.24}$$

$(p, \prec)$ is *more delayed* than $(p', \prec')$ if

$$\forall k.\ \left|V_{\cup k}^{\prec}\right| \geq \left|V_{\cup k}^{\prec'}\right| \tag{4.25}$$

and there exists a k for which this inequality is strict. $(p, \prec)$ is *maximally delayed* if there does not exist a Pauli flow on the same labelled open graph that is more delayed.

A finite graph can only admit a finite number of Pauli flows (there are only finitely many possible choices of maps p and orders $\prec$) and the delayed relation is a strict partial order, so if at least one Pauli flow exists there must be a maximally delayed one. This ensures it is safe for our identification algorithm to specifically search for a maximally delayed Pauli flow.

Qubits at measurement depth 0 are either output qubits or measurements that require no correction (all other qubits involved in the correcting stabilizer are measured in the appropriate Pauli basis for it to have no effect). The remaining qubits are arranged into layers of qubits that can be measured simultaneously because all the active corrections involved happen in a later layer (at a lower depth from the output). Since measurements at the same measurement depth are independent of each other, we can guide our search for a maximally delayed Pauli flow to just find the largest set of qubits at a given measurement depth before moving on to the next one. To do this, it is easiest to break down the cases by measurement label.

Lemma 4.2.2: Generalisation of [88, Lemma 1/3][7, Lemma C.2/C.4]

For any Pauli flow $(p, \prec)$ we subdivide the vertices at measurement depth by measurement label $L \in \{XY, XZ, YZ, X, Y, Z\}$.

$$\Lambda^L := \left\{u \in \overline{O} \,\middle|\, \lambda(u) = L\right\} \tag{4.26}$$

$$V_k^{\prec,L} := V_k^{\prec} \cap \Lambda^L \tag{4.27}$$

This gives $V_0^{\prec} = O \cup \bigcup_L V_0^{\prec,L}$ and then $V_k^{\prec} = \bigcup_L V_k^{\prec,L}$ for all $k > 0$.

If $(p, \prec)$ is a maximally delayed Pauli flow, then

$$V_k^{\prec,XY} = \left\{ u \in \Lambda^{XY} \setminus V_{\cup k-1}^{\prec} \,\middle|\, \begin{array}{l} \exists K \subseteq \mathbb{K}. \\ \mathrm{Odd}\left(K\right) \cap \mathbb{P} = \{u\}, \\ \left(K \Delta \mathrm{Odd}\left(K\right)\right) \cap \mathbb{Y} = \varnothing \end{array} \right\} \tag{4.28}$$

$$V_k^{\prec,XZ} = \left\{ u \in \Lambda^{XZ} \setminus V_{\cup k-1}^{\prec} \,\middle|\, \begin{array}{l} \exists K \subseteq \mathbb{K}. \\ \mathrm{Odd}\left(K \cup \{u\}\right) \cap \mathbb{P} = \{u\}, \\ \left(\left(K \cup \{u\}\right) \Delta \mathrm{Odd}\left(K \cup \{u\}\right)\right) \cap \mathbb{Y} = \varnothing \end{array} \right\} \tag{4.29}$$

$$V_k^{\prec,YZ} = \left\{ u \in \Lambda^{YZ} \setminus V_{\cup k-1}^{\prec} \,\middle|\, \begin{array}{l} \exists K \subseteq \mathbb{K}. \\ \mathrm{Odd}\left(K \cup \{u\}\right) \cap \mathbb{P} = \varnothing, \\ \left(\left(K \cup \{u\}\right) \Delta \mathrm{Odd}\left(K \cup \{u\}\right)\right) \cap \mathbb{Y} = \varnothing \end{array} \right\} \tag{4.30}$$

$$V_k^{\prec,X} = \left\{ u \in \Lambda^X \setminus V_{\cup k-1}^{\prec} \;\middle|\; \begin{array}{l} \exists K \subseteq \mathbb{K}. \\ \mathrm{Odd}\left(K\right) \cap \mathbb{P} = \{u\}, \\ \left(K \Delta \mathrm{Odd}\left(K\right)\right) \cap \mathbb{Y} = \varnothing \end{array} \right\} \tag{4.31}$$

$$V_k^{\prec,Y} = \left\{ u \in \Lambda^Y \setminus V_{\cup k-1}^{\prec} \;\middle|\; \begin{array}{l} \exists K \subseteq \mathbb{K}. \\ \mathrm{Odd}\left(K\right) \cap \mathbb{P} = \varnothing, \\ \left(K \Delta \mathrm{Odd}\left(K\right)\right) \cap \mathbb{Y} = \{u\} \end{array} \right\} \tag{4.32}$$

$$V_k^{\prec,Z} = \left\{ u \in \Lambda^Z \setminus V_{\cup k-1}^{\prec} \;\middle|\; \begin{array}{l} \exists K \subseteq \mathbb{K}. \\ \mathrm{Odd}\left(K \cup \{u\}\right) \cap \mathbb{P} = \varnothing, \\ \left(\left(K \cup \{u\}\right) \Delta \mathrm{Odd}\left(K \cup \{u\}\right)\right) \cap \mathbb{Y} = \varnothing \end{array} \right\} \tag{4.33}$$

where

$$\mathbb{K} \coloneqq \overline{I} \cap \left(V_{\cup k-1}^{\prec} \cup \Lambda^X \cup \Lambda^Y\right) \tag{4.34}$$

$$\mathbb{P} \coloneqq \overline{O} \setminus \left(V_{\cup k-1}^{\prec} \cup \Lambda^Y \cup \Lambda^Z\right) \tag{4.35}$$

$$\mathbb{Y} \coloneqq \Lambda^Y \setminus V_{\cup k-1}^{\prec} \tag{4.36}$$

Proof. The argument for each measurement label follows fairly similarly. Any important differences will be highlighted.

$\subseteq$: By definition, any $u \in V_k^{\prec,L}$ is in $\Lambda^L \setminus V_{\cup k-1}^{\prec}$. Let us take the set K (or $K \cup \{u\}$ for $L \in \{XZ, YZ, Z\}$) to be $p\left(u\right)$. $p\left(u\right) \subseteq \overline{I}$ because p is a function $\overline{O} \to \mathcal{P}\left[\overline{I}\right]$, and $p\left(u\right) \subseteq V_{\cup k-1}^{\prec} \cup \Lambda^X \cup \Lambda^Y \cup \{u\}$ is equivalent to $\left[\prec .X\right]$. In the $L = XY$ case, $\left[\lambda.XY\right]$ says that $u \notin p\left(u\right)$, and for $L \in \{X, Y\}$ we have $u \in \Lambda^X \cup \Lambda^Y$ and the Pauli flow conditions neither require u to be present nor absent from $p\left(u\right)$.

For the next criterion, $\left[\prec .Z\right]$ is equivalently stated as $\mathrm{Odd}\left(p\left(u\right)\right) \subseteq V_{\cup k-1}^{\prec} \cup \Lambda^Y \cup \Lambda^Z \cup \{u\}$. If $L = YZ$, $\left[\lambda.YZ\right]$ also says that $u \notin \mathrm{Odd}\left(p\left(u\right)\right)$, and similarly the conditions for each of $L \in \{XY, XZ, X\}$ require $u \in \mathrm{Odd}\left(p\left(u\right)\right)$. The remaining conditions for $L \in \{Y, Z\}$ don't impose any constraints on u's membership in $\mathrm{Odd}\left(p\left(u\right)\right)$, which is handled by the exclusion of $\Lambda^Y \cup \Lambda^Z$ from $\mathbb{P}$.

The final criterion on $p\left(u\right) \Delta \mathrm{Odd}\left(p\left(u\right)\right)$ precisely matches $\left[\prec .Y\right]$. In the case of $L = Y$, it also factors in $\left[\lambda.Y\right]$.

In summary, we can interpret $\mathbb{K}$ as the set of vertices the Pauli flow conditions constrain $p\left(u\right) \setminus \{u\}$ to live in (and hence could viably receive X's from the correction stabilizer), $\mathbb{P}$ as the set of past/present vertices that need to be unaffected by the Z's on the correction stabilizer, and $\mathbb{Y}$ as the set of past/present vertices measured in the Y basis that require special care for $\left[\prec .Y\right]$ and $\left[\lambda.Y\right]$.

$\supseteq$: We will suppose there is some $u \notin V_k^{\prec,L}$ that satisfies the corresponding conditions in the equations above and aim for a contradiction by generating a new Pauli flow that is more delayed than $(p, \prec)$. Given the set $K \subseteq \overline{I} \cap \left(V_{\cup k-1}^{\prec} \cup \Lambda^X \cup \Lambda^Y\right)$, we define a new Pauli flow $(p', \prec')$:

$$p'(v) := \begin{cases} K & v = u \wedge L \in \{XY, X, Y\} \\ K \cup \{u\} & v = u \wedge L \in \{XZ, YZ, Z\} \\ p(v) & v \neq u \end{cases} \tag{4.37}$$

$$\prec' := \prec \setminus \{(u, v) \mid u \prec v \wedge v \notin V_{\cup k-1}^{\prec}\} \tag{4.38}$$

The Pauli flow conditions on u are met by the same reasoning as in the $\subseteq$ case, and they are trivially preserved for all other $v \neq u$.

By construction, the depth of every vertex under $\prec'$ is no larger than its depth under $\prec$, plus we now have $u \in V_k^{\prec',L}$, so this is a more delayed Pauli flow. □

These characterisations of the sets $V_k^{\prec,L}$ give us an iterative method of identifying them since we can simply search for possible witness sets K for each vertex. The way in which these characterisations are written makes it easy to convert them into a linear equation system to solve for each set K. We build an equation $MX = S$ in matrices over $\mathbb{F}_2$.

- X is a vector of length $|\mathbb{K}|$ where, should a solution exist, the 1 entries will correspond to the vertices in a valid witness for K.

- M is a block matrix: the first block describes Γ the adjacency matrix of the graph, so multiplication with X calculates $\mathrm{Odd}(K)$, and the second block is the same adjacency matrix plus the identity to calculate $K \Delta \mathrm{Odd}(K)$.

- S is then a vector of length $|\mathbb{P}| + |\mathbb{Y}|$ into which we encode the constraints on $\mathrm{Odd}(K)$ and $K \Delta \mathrm{Odd}(K)$. For $L \in \{XZ, YZ, Z\}$ where the set $p(u)$ would actually correspond to $K \cup \{u\}$, we can imagine pushing the vector for u through M to yield the neighbourhood $N_\Gamma(u)$ and add incorporate this into S to encode the same condition.

$$M := \left[\frac{\Gamma \cap \mathbb{K} \times \mathbb{P}}{(\Gamma + \mathrm{Id}) \cap \mathbb{K} \times \mathbb{Y}} \right] \tag{4.39}$$

$$S = \begin{cases} \left[\begin{array}{c} \{u\} \\ \hline 0 \end{array}\right] & L \in \{XY, X\} \\ \left[\begin{array}{c} (N_\Gamma(u) \cap \mathbb{P}) \cup \{u\} \\ \hline N_\Gamma(u) \cap \mathbb{Y} \end{array}\right] & L = XZ \\ \left[\begin{array}{c} N_\Gamma(u) \cap \mathbb{P} \\ \hline N_\Gamma(u) \cap \mathbb{Y} \end{array}\right] & L \in \{YZ, Z\} \\ \left[\begin{array}{c} 0 \\ \hline \{u\} \end{array}\right] & L = Y \end{cases} \tag{4.40}$$

Such a linear equation system can be solved by Gaussian elimination and back substitution. To identify all vertices at a given measurement depth $V_k^{\prec}$, we can iterate through all the unsolved vertices and run this procedure. Fortunately, for any fixed measurement depth the matrix M is constant, meaning we can horizontally stack the S vectors for each unsolved vertex and run just one instance of Gaussian elimination to identify $V_k^{\prec}$.

Algorithm 1 details a function taking a description of a labelled open graph with adjacency matrix Γ and returns `true` with a maximally delayed Pauli flow if one exists (with the partial order encoded into a map $d : \overline{O} \to \mathbb{N}$ describing the measurement depth of vertices), and `false` otherwise.

Theorem 4.2.3: Pauli Flow Identification Algorithm

Algorithm 1 correctly identifies whether a given labelled open graph has a Pauli flow, and outputs such a Pauli flow if it exists. Moreover, this output is maximally delayed, and the algorithm completes deterministically in time that grows polynomially with the number of vertices in the graph.

Proof. At each measurement depth, it solves the above linear equation system to identify qubits that can be corrected by only applying gates to those other qubits that have already been solved. Solving this step with Gaussian elimination and back substitution takes $O\left(|V|^3\right)$ time. If no Pauli flow exists, there is no set of solutions to the above linear equation system for all vertices, so the algorithm must reach a point where no more solutions are found and returns `false`. If one does exist, this will always make some progress, so will terminate in at most V iterations, giving a total runtime of $O\left(|V|^4\right)$. The resulting Pauli flow is maximally delayed as we are constructing the largest set possible at each measurement depth by Lemma 4.2.2. □

```
PauliFlow $(V, \Gamma, I, O, \lambda)$ = begin
    $\Lambda^X := \varnothing$; $\Lambda^Y := \varnothing$; $\Lambda^Z := \varnothing$;
    forall $u \in V \setminus O$ do
        if $\lambda(u) \in \{X, Y, Z\}$ then $\Lambda^{\lambda(u)}$.insert$(u)$;
    end
    solved $:= O$;
    $\mathbb{K} := \Lambda^X \cup \Lambda^Y$;
    $\mathbb{P} := V \setminus (O \cup \Lambda^Y \cup \Lambda^Z)$;
    $\mathbb{Y} := \Lambda^Y$;
    depth $:= 0$;
    $p := \{\}$;
    $d := \{\}$;
    repeat
        $M :=$ [Equation 4.39];
        $S :=$ Matrix$(|\mathbb{P}| + |\mathbb{Y}|, 0)$;
        forall $u \in V \setminus$ solved do
            $S_u :=$ [Equation 4.40];
            $S :=$ Matrix.hstack$(S, S_u)$;
        end
        sols $:= M$.solve$(S)$;
        if |sols| = 0 then return (false, $\varnothing$, $\varnothing$);
        forall $(u, K) \in$ sols do
            if $\lambda(u) \in \{XY, X, Y\}$ then $p[u] := K$;
            else $p[u] := K \cup \{u\}$;
            $d[u] :=$ depth;
            solved.insert$(u)$;
            if $u \notin I$ then $\mathbb{K}$.insert$(u)$;
            $\mathbb{P}$.erase$(u)$;
            $\mathbb{Y}$.erase$(u)$;
        end
        if depth = 0 then $\mathbb{K} := \mathbb{K} \cup (O \setminus I)$;
        depth $:=$ depth + 1;
    until solved $= V$;
    return (true, $p$, $d$);
end
```

Algorithm 1: An algorithm for identifying whether a labelled open graph has a Pauli flow.

This algorithm may also be used to identify gflow, since the conditions for gflow and Pauli flow coincide when all measurement labels are planar.

This algorithm fits nicely into the pipeline of taking a parameterised linear map and searching for an MBQC implementation of it: start by finding a labelled open graph which implements the linear map on its intended branch (e.g. by encoding the map into the ZX-calculus and reduce it to a graphlike form [44]), then search for a Pauli flow on that graph. The guarantee of a maximally delayed Pauli flow is rather practical, as it guarantees the fewest number of rounds of simultaneous measurements possible - the fewer rounds of simultaneous measurements and corrections needed, the faster the pattern runs on a quantum computer and therefore has potential to accumulate less noise. However, such a flow may still not be unique (e.g. the back substitution step may find several solutions to the linear equation system) and it may be beneficial to search for ones with additional useful properties.

4.2.2 Focussed Sets

If we have a qubit u measured in the X basis, then any X applied to u to correct another measurement can be absorbed rather than being performed. If we can guarantee that u receives *only* X's from correcting other measurements ($\forall v \neq u.\ P^{p(v) \to u} \in \{I, X\}$), then we can always include u in the first round of measurements. A *focussed* flow asks that this is the case for any measurement label. We choose to associate planar measurements with the Pauli they coincide with at measurement angle 0, e.g. $\langle +_{XY,0}| = \langle +_{X,0}|$. At a minimum, this guarantees that the corrections applied to each qubit are always a fixed Pauli.

Focussed gflow has seen interest in prior literature to characterise flow as a right-inverse of the adjacency matrix [87] when all labels are XY.

Definition 4.2.4: Focussed flow [87] and sets

Given a labelled open graph Γ, a set $\hat{p} \subseteq \overline{I}$ is *focussed over* $S \subseteq \overline{O}$ if:

$[FX]$ $\forall u \in S \cap \hat{p}.\ \lambda(u) \in \{XY, X, Y\}$

$[FZ]$ $\forall u \in S \cap \text{Odd}(\hat{p}).\ \lambda(u) \in \{XZ, YZ, Y, Z\}$

$[FY]$ $\forall u \in S.\ \lambda(u) = Y \implies (u \in \hat{p} \iff u \in \text{Odd}(\hat{p}))$

$\hat{p}$ is a *focussed set* for Γ if it is focussed over $\overline{O}$. A Pauli flow $(p, \prec)$ is *focussed* if $p(u)$ is focussed over $\overline{O} \setminus \{u\}$ for all $u \in \overline{O}$.

Proposition 4.2.5

If a labelled open graph has a focussed Pauli flow $(p, \prec)$, then there exists a focussed Pauli flow $(p, \prec')$ which satisfies $\forall v \in \overline{O}.\ \lambda(v) \in \{X, Y, Z\} \implies \forall u \in V.\ v \preceq' u$.

Proof. Let $\prec' := \prec \setminus \left\{(u, v) \in V \times \overline{O} \;\middle|\; \lambda(v) \in \{X, Y, Z\}\right\}$. By construction, this satisfies $\forall v \in \overline{O}.\ \lambda(v) \in \{X, Y, Z\} \implies \forall u \in V.\ v \preceq' u$. To show that $(p, \prec')$ is a valid Pauli flow, we just need to show that all $[\prec .P]$ conditions are unaffected which all follow from the focussed conditions. □

The next few results work towards a way to transform any Pauli flow into a focussed Pauli flow.

Lemma 4.2.6

Given a Pauli flow $(p, \prec)$ for a labelled open graph (G, I, O, λ) with two vertices $u, v \in \overline{O}$ such that $u \prec v$, then $(p', \prec)$ is a Pauli flow where:

$$p'(w) := \begin{cases} p(u) \,\Delta\, p(v) & w = u \\ p(w) & w \neq u \end{cases} \tag{4.41}$$

Moreover, if $(p, \prec)$ is maximally delayed, then so is $(p', \prec)$.

Proof. The Pauli flow conditions hold trivially for any vertex in $\overline{O} \setminus \{u\}$ since the correction sets have not changed, so it is sufficient to show they are preserved for u. We should first observe that $\mathrm{Odd}(p'(u)) = \mathrm{Odd}(p(u) \,\Delta\, p(v)) = \mathrm{Odd}(p(u)) \,\Delta \mathrm{Odd}(p(v))$.

$[\prec .X]$: For any $w \in p'(u)$ with $w \neq u$ and $\lambda(w) \notin \{X, Y\}$, we must have either $w \in p(u)$, $w = v$, or $w \in p(v) \wedge w \neq v$. In any of these cases, we have $u \prec w$ from $[\prec .X]$ for $(p, \prec)$ and $u \prec v$.

$[\prec .Z]$: This follows similarly from $u \prec v$ and $[\prec .Z]$ on $\mathrm{Odd}(p(u))$ and $\mathrm{Odd}(p(v))$.

$[\prec .Y]$: For any $w \preceq u$ with $\lambda(w) = Y$, we also must have $w \preceq v$ since $u \prec v$. Hence by $[\prec .Y]$, $w \in p(u) \iff w \in \mathrm{Odd}(p(u))$ and the same for $p(v)$. Therefore, $w \in p'(u) = p(u) \,\Delta\, p(v) \iff w \in \mathrm{Odd}(p(u)) \,\Delta \mathrm{Odd}(p(v)) = \mathrm{Odd}(p'(u))$ as required.

$[\lambda .XY]$-$[\lambda .YZ]$: $u \notin p(v)$ and $u \notin \mathrm{Odd}(p(v))$ by $[\prec .X]$ and $[\prec .Z]$ since $u \prec v$, so the requirements are given by the corresponding conditions for $(p, \prec)$.

$[\lambda.X]$: $u \notin \mathrm{Odd}(p(v))$ by $[\prec.Z]$ and $u \in \mathrm{Odd}(p(u))$ by $[\lambda.X]$, so $u \in \mathrm{Odd}(p'(u))$.

$[\lambda.Z]$: $u \notin p(v)$ by $[\prec.X]$ and $u \in p(u)$ by $[\lambda.Z]$, so $u \in p'(u)$.

$[\lambda.Y]$: $u \in p(v) \iff u \in \mathrm{Odd}(p(v))$ by $[\prec.Y]$ and $u \in p(u) \iff u \notin \mathrm{Odd}(p(u))$ by $[\lambda.Y]$, then it is straightforward to show $u \in p'(u) \iff u \notin \mathrm{Odd}(p'(u))$ by cases.

The maximally delayed property of a Pauli flow only concerns the partial order between the vertices, so since $(p, \prec)$ and $(p', \prec)$ both use $\prec$ the property is trivially preserved. □

This gives us a mechanism to generate new Pauli flows by adding correction sets together. We now show that this can help us to make progress towards satisfying the focussed property.

Lemma 4.2.7

For any labelled open graph Γ, if sets $\hat{p}, \hat{q} \subseteq \overline{I}$ are focussed over $S \subseteq \overline{O}$, then so is $\hat{p}\Delta\hat{q}$.

Proof. $[FX]$: For any vertex $v \in (\hat{p}\Delta\hat{q}) \cap S$, we have either $v \in \hat{p}$ or $v \in \hat{q}$. Since $\hat{p}$ and $\hat{q}$ are focussed over S, the corresponding $[FX]$ condition gives $\lambda(v) \in \{XY, X, Y\}$.

$[FZ]$: Similarly, for any vertex $v \in \mathrm{Odd}(\hat{p}\Delta\hat{q}) \cap S = (\mathrm{Odd}(\hat{p})\Delta\mathrm{Odd}(\hat{q})) \cap S$, we have $\lambda(v) \in \{XZ, YZ, Y, Z\}$ from either $v \in \mathrm{Odd}(\hat{p})$ or $v \in \mathrm{Odd}(\hat{q})$ and $[FZ]$.

$[FY]$: For any $v \in S$ with $\lambda(v) = Y$, we have $v \in \hat{p} \iff v \in \mathrm{Odd}(\hat{p})$ and $v \in \hat{q} \iff v \in \mathrm{Odd}(\hat{q})$ from $[FY]$ for $\hat{p}$ and $\hat{q}$. Hence, $v \in \hat{p}\Delta\hat{q} \iff v \in \mathrm{Odd}(\hat{p})\Delta\mathrm{Odd}(\hat{q})$. □

Lemma 4.2.8

For any labelled open graph Γ, if sets $\hat{p}, \hat{q} \subseteq \overline{I}$ are not focussed over $\{v\}$ $(v \in \overline{O})$, then $\hat{p}\Delta\hat{q}$ is focussed over $\{v\}$.

Proof. We consider each case for $\lambda(v)$ and how the focussed conditions could fail for $\hat{p}$ and $\hat{q}$:

$\lambda(v) \in \{XY, X\}$: $[FX]$ and $[FY]$ are trivially satisfied, so we must have $v \in \mathrm{Odd}(\hat{p})$ and $v \in \mathrm{Odd}(\hat{q})$ to fail $[FZ]$. This means $v \notin \mathrm{Odd}(\hat{p})\Delta\mathrm{Odd}(\hat{q}) =$

$\mathrm{Odd}(\hat{p}\Delta\hat{q})$, satisfying $[FZ]$ for $\hat{p}\Delta\hat{q}$.

$\lambda(v) \in \{XZ, YZ, Z\}$: Similarly, $[FZ]$ and $[FY]$ hold trivially, so we must have $v \in \hat{p}$ and $v \in \hat{q}$ to fail $[FX]$. We hence have $v \notin \hat{p}\Delta\hat{q}$, satisfying $[FZ]$ for $\hat{p}\Delta\hat{q}$.

$\lambda(v) = Y$: Now $[FX]$ and $[FZ]$ are trivial and we have $v \in \hat{p}\Delta\mathrm{Odd}(\hat{p})$ and $v \in \hat{q}\Delta\mathrm{Odd}(\hat{q})$ to fail $[FY]$. Combined, these satisfy $[FY]$ since $v \notin (\hat{p}\Delta\mathrm{Odd}(\hat{p}))\Delta(\hat{q}\Delta\mathrm{Odd}(\hat{q})) = (\hat{p}\Delta\hat{q})\Delta\mathrm{Odd}(\hat{p}\Delta\hat{q})$. □

Combining these two lemmas, we can find combinations of correction sets that fix unfocussed vertices whilst preserving those we have already focussed.

Lemma 4.2.9: Generalisation of [7, Lemma 3.13]

Let (G, I, O, λ) be a labelled open graph with a Pauli flow $(p, \prec)$ and some vertex $v \in \overline{O}$. Then there exists $p' : \overline{O} \to \mathcal{P}[\overline{I}]$ such that:

1. $\forall w \in \overline{O}.\ v = w \vee p'(w) = p(w)$;
2. $p'(v)$ is focussed over $\overline{O} \setminus \{v\}$;
3. $(p', \prec)$ is a Pauli flow for (G, I, O, λ).

Proof. Let $J : \mathbb{Z}_{|\overline{O}|} \to \overline{O}$ be some indexing of the vertices that respects the order $\prec$ $(\forall i, j < |\overline{O}|.\ J(i) \prec J(j) \implies i < j)$. We define a sequence of functions p_k as:

$$p_0(u) := p(u) \tag{4.42}$$

$$p_{k+1}(u) := \begin{cases} p_k(u)\,\Delta\, p_k(J(k)) & \begin{pmatrix} u = v, \\ J(k) \neq v, \\ p_k(v) \text{ is not focussed over } \{J(k)\} \end{pmatrix} \\ p_k(u) & \text{otherwise} \end{cases} \tag{4.43}$$

1 is satisfied for all $(p_k, \prec)$ by construction, and 3 is also satisfied for all by Lemma 4.2.6. To work towards 2, we proceed inductively with hypothesis $\Phi(k) :=$ "$p_k(v)$ is focussed over $\{J(i)\}_{i<k} \setminus \{v\}$". The $k = 0$ case holds vacuously.

Suppose we have $\Phi(k)$. If $J(k) = v$, then $p_{k+1}(v) = p_k(v)$ and $\Phi(k+1)$ is an immediate consequence of $\Phi(k)$. If $p_k(v)$ is focussed over $\{J(k)\}$, then $p_{k+1}(v) = p_k(v)$, so $\Phi(k+1)$ follows from this assumption and $\Phi(k)$. If, on the other hand, $p_k(v)$ is not focussed over $\{J(k)\}$, we have $p_{k+1}(v) = p_k(v) \Delta\ p_k(J(k))$. From conditions $[\lambda.XY]$-$[\lambda.Y]$, $p_k(J(k))$ is also not focussed over $\{J(k)\}$, so by Lemma 4.2.8 we have $p_{k+1}(v)$ is focussed over $\{J(k)\}$. For any of the remaining $i < k$ (where $J(i) \neq v$), $\Phi(k)$ says that $p_k(v)$ is focussed over $\{J(i)\}$. Since J respects the order $\prec$, we also have $J(i) \preceq J(k)$, and hence the $[\prec.P]$ conditions imply that $p_k(J(k))$ is focussed over $\{J(i)\}$. We combine these with Lemma 4.2.7 to deduce that $p_{k+1}(v)$ is also focussed over $\{J(i)\}$.

This chain ends in $\left(p_{|\overline{O}|}, \prec\right)$ where $p_{|\overline{O}|}$ is focussed over $\overline{O} \setminus \{v\}$. □

Lemma 4.2.10: Focussing Lemma

If a labelled open graph has a Pauli flow, then it has a maximally delayed, focussed Pauli flow.

Proof. If a Pauli flow exists, then there must be a maximally delayed one $(p, \prec)$. Applying Lemma 4.2.9 for each $v \in \overline{O}$ in turn, we reach some $(p', \prec)$ where, for every $v \in \overline{O}$, $p'(v)$ is focussed over $\overline{O} \setminus \{v\}$, i.e. $(p', \prec)$ is a focussed Pauli flow. Since the partial order $\prec$ remains the same, this is still maximally delayed. □

It should be noted that the focussed conditions can be incorporated into the linear equation system subroutine of Algorithm 1 to directly obtain focussed Pauli flows without having to separately apply the above routine to focus the output.

For general measurement patterns, even focussed Pauli flows may not be unique. However, given multiple focussed Pauli flows, the differences between their correction sets are given by the focussed sets of the graph. Conversely, we can add focussed sets to a flow at any point where it doesn't mess up previous measurements.

Lemma 4.2.11

Let Γ be a labelled open graph with two focussed Pauli flows $(p, \prec)$ and $(p', \prec')$. Then for any vertex $v \in \overline{O}$, $p(v) \Delta\ p'(v)$ is a focussed set.

Proof. Each of $p(v)$ and $p'(v)$ are focussed over $\overline{O} \setminus \{v\}$, so their combination must also be by Lemma 4.2.7. For each case of $\lambda(v)$, the corresponding condition from $[\lambda.XY]$-$[\lambda.Y]$ is then enough to show that $p(v) \Delta\ p'(v)$ is also focussed over $\{v\}$. □

Lemma 4.2.12

Let $\Gamma = (G, I, O, \lambda)$ be a labelled open graph with a focussed Pauli flow $(p, \prec)$ and a focussed set $\hat{p} \subseteq \overline{I}$. Let $v \in \overline{O}$ be a vertex where $\forall w \in \hat{p} \cup \mathrm{Odd}(\hat{p})\,.\,\lambda(w) \in \{XY, XZ, YZ\} \implies w \neq v \wedge v \preceq w$. Then $(p', \prec')$ is a focussed Pauli flow, where:

$$p'(w) := \begin{cases} p(w) \,\Delta\, \hat{p} & w = v \\ p(w) & w \neq v \end{cases} \tag{4.44}$$

and $\prec'$ is the transitive closure of

$$\prec \cup \{(v, w) \mid w \in \hat{p} \cup \mathrm{Odd}(\hat{p}) \wedge \lambda(w) \in \{XY, XZ, YZ\}\} \tag{4.45}$$

Proof. Firstly, $\prec'$ is still a strict partial order. Transitivity is immediate by definition. For antisymmetry (and similarly for strictness), suppose on the contrary that we have some $a \prec' b$ and $b \prec' a$ (or directly $a \prec' a$). Unpacking the transitive closure, this gives a cycle $[a, \ldots, b, \ldots, a]$ where each step is either in $\prec$ or $\{(v, w) \mid w \in \hat{p} \cup \mathrm{Odd}(\hat{p}) \wedge \lambda(w) \in \{XY, XZ, YZ\}\}$. Since $\prec$ is a strict partial order, we cannot have such a cycle where every step is in $\prec$, so at least one step must be some such (v, w). We can freely eliminate inner cycles around v, so we can assume wlog that v only appears once in the cycle. This means the rest of the cycle is only from $\prec$, so $w \prec v$ by transitivity. However, we assumed that $v \preceq w$ since $w \in \hat{p} \cup \mathrm{Odd}(\hat{p})$ and $\lambda(w) \in \{XY, XZ, YZ\}$, giving us the contradiction we need.

The $[\prec .P]$ conditions are preserved from the extension to $\prec'$ covering planar labels and the focussed property covering Pauli labels.

Conditions $[\lambda.XY]$, $[\lambda.XZ]$, and $[\lambda.YZ]$ are preserved since $v \notin \hat{p} \cup \mathrm{Odd}(\hat{p})$ gives $v \in p'(v) \iff v \in p(v)$ and $v \in \mathrm{Odd}(p'(v)) \iff v \in \mathrm{Odd}(p(v))$.

For conditions $[\lambda.X]$, $[\lambda.Z]$, and $[\lambda.Y]$, the correction amounts to saying that $p'(v)$ is not focussed over $\{v\}$. If this were not the case, then $p'(v) \,\Delta\, \hat{p} = p(v)$ would be focussed over $\{v\}$ by Lemma 4.2.7, contradicting the corresponding Pauli flow condition for $p(v)$.

Finally, $p'(v)$ is focussed over $\overline{O} \setminus \{v\}$ using Lemma 4.2.7, so $(p', \prec')$ is focussed. □

An important consequence of this result is that, in order to fully identify the space of all focussed Pauli flows for a given labelled open graph, it is sufficient to find one using the Pauli Flow Identification Algorithm that we focus with the Focussing Lemma, and find all the focussed sets. The following proofs show that the focussed sets form a group, allowing us to reduce the search to just some generating set.

Proposition 4.2.13

The focussed sets of a labelled open graph form a group under Δ.

Proof. Closure under Δ is a direct consequence of Lemma 4.2.7 with $S = \overline{O}$. The identity is simply the empty set $\varnothing$ and each focussed set is self-inverse. □

Lemma 4.2.14

Any labelled open graph $\Gamma = (G, I, O, \lambda)$ with a focussed Pauli flow has $2^{|O|-|I|}$ distinct focussed sets.

Proof. There are $2^{|\overline{I}|}$ subsets of $\overline{I}$.

Fix some focussed Pauli flow $(p, \prec)$. For each $v \in \overline{O}$, $p(v)$ is focussed over $\overline{O} \setminus \{v\}$. By Lemmas 4.2.12 and 4.2.11, $(-) \Delta\, p(v)$ defines a bijection between those sets that are focussed for v and those that are not. So every time we restrict our subsets to those focussed over an additional vertex, we half the number of possible subsets.

Repeating this for all of $\overline{O}$ means we have $2^{|\overline{I}|-|\overline{O}|} = 2^{|O|-|I|}$ focussed sets. □

Similar to the Pauli Flow Identification Algorithm, we can encode the conditions for focussed sets into a linear equation system $MX = S$ in $\mathbb{F}_2$ which can be solved by Gaussian elimination and back substitution to obtain a single focussed set.

$$M := \left[\begin{array}{c} \Gamma \cap \mathbb{P} \times \mathbb{O} \\ \hline (\Gamma + \mathrm{Id}) \cap \mathbb{P} \times \Lambda^{Y} \end{array}\right] \tag{4.46}$$

$$S := \left[\begin{array}{c} 0 \\ \hline 0 \end{array}\right] \tag{4.47}$$

$$\mathbb{P} := \left(O \cup \Lambda^{XY} \cup \Lambda^{X} \cup \Lambda^{Y}\right) \cap \overline{I} \tag{4.48}$$

$$\mathbb{O} := \Lambda^{XY} \cup \Lambda^{X} \tag{4.49}$$

In the above, we let Γ stand for the adjacency matrix of its graph. We define $\mathbb{P}$ to be the set of vertices that could be included in a focussed set and $\mathbb{O}$ the set of vertices that can't be in the odd neighbourhood. Solutions satisfy $[FX]$ since X only ranges over $\mathbb{P}$. The top block of the system encodes $[FZ]$ since multiplying by Γ in $\mathbb{F}_2$ gives the odd neighbourhood, and similarly the bottom block encodes $[FY]$.

We need to add additional conventions to make sure that the focussed sets we obtain are non-empty and independent. Since this system is underconstrained (when there exist non-empty focussed sets), there will always be some freedom of choice during the back substitution step. Since every focussed set must be a solution, these free substitutions must generate the full set. Hence, we can obtain independent, non-empty focussed sets by taking a single free substitution for each focussed set, iterating through each free substitution in turn. Algorithm 2 demonstrates this.

```
FocussedSets$(V, \Gamma, I, O, \lambda)$ = begin
    $M$ := [Equation 4.46];
    $M'$ := reduced_row_echelon$(M)$;
    FSets := $\varnothing$;
    leading_lookup := $\{\}$;
    foreach $c < M'$.cols do
        $\hat{p} := \{c\}$;
        leading = false;
        foreach $r < M'$.rows do
            if $M'(r, c)$ then
                if $r \in$ leading_lookup then
                    $\hat{p}$.insert(leading_lookup$[r]$);
                end
                else
                    leading_lookup$[r] := c$;
                    leading := true;
                    break;
                end
            end
        end
        if !leading then FSets.insert$(\hat{p})$;
    end
    return FSets;
end
```

Algorithm 2: An algorithm for identifying generators for all focussed sets of a labelled open graph.

Lemma 4.2.15: Focussed Set Identification Algorithm

Given a labelled open graph Γ with focussed Pauli flow, Algorithm 2 identifies $|O| - |I|$ independent generators for the group of focussed sets of Γ. Furthermore, it completes in polynomial time wrt the number of vertices in Γ.

Proof. All sets found are focussed sets since the linear equations encode the focussed conditions. They are independent as each one contains a unique vertex corresponding to the column c. We obtain the right quantity by Lemma 4.2.14, since all focussed sets must be solutions to this linear equation system and hence back substitution can break them down in terms of these generators.

Since both $|\mathbb{P}|$ and $|\mathbb{O}|$ are at most $|V|$, the Gaussian elimination takes $O\left(|V|^3\right)$ time which is the dominant cost of the algorithm. □

Lemma 4.2.14 now puts a bound on the number of possible focussed Pauli flows for a given labelled open graph. In the unitary case ($|O| = |I|$) specifically, this implies the uniqueness of focussed Pauli flows, generalising the known result for the uniqueness of focussed gflow for unitaries [87].

In Sections 4.3.2 and 4.3.3, we will relate focussed flows and focussed sets to properties of the linear map implemented by a measurement pattern. They make it especially easy to construct equivalent parameterised quantum circuits for the same linear map.

4.3 Circuit Extraction

Quantum circuits are far more common than MBQC as a model of quantum computation and as a design space for quantum algorithms. Especially when working at scale, measurement patterns can become hard to design or analyse through only the linear map representation which could be alleviated with better tools for relating them to quantum circuits.

The goal of circuit extraction is to identify a sequence of gates that implements the same linear map as a given measurement pattern without calculating and decomposing the linear map. The method presented here will make use of a Pauli flow to determine the effect that each measurement angle has on the outputs. Doing so will yield a circuit in the form of a Clifford process followed by a sequence of parameterised Pauli rotations, with one rotation per planar measurement.

4.3.1 Circuits as Products of Pauli Rotations

Quantum circuits will often be presented in terms of small elementary gates, such as the universal gate set $\{CX, RZ, RX\}$, as these often very closely relate to the operations available natively on a quantum computer. This can often come with the downside of being quite difficult to identify the macroscopic structure of a circuit due to the sheer quantity and locality of the elementary gates. Pauli rotations $e^{-i\frac{\theta}{2}P}$ [14] ($P \in \{I, X, Y, Z\}^{\otimes n}$) instead provide a universal gate set for unitary quantum circuits with the ability to abstract away the influence of Clifford gates and represent non-local logical effects in a way that makes it easy to identify commutations and large-scale influence across a circuit. Let us briefly examine how these give rise to a convenient data structure for examining a pure quantum circuit by expressing it as a Clifford circuit C followed by a sequence of Pauli rotations (stored up to their commutation relation).

$$\left(\prod_j e^{-i\frac{\theta_j}{2}P_j}\right)C \tag{4.50}$$

This representation was notably covered in the work of Zhang and Chen [120] where it was used to identify possible pairs of T gates which could be merged through some sequence of gate commutations and Clifford gate relations in order to reduce the number of T gates in the circuit. Similar structures have been studied in the context of compiling circuits for lattice surgery [83] and as an intermediate for synthesis of Clifford+T circuits [57]. Some of these presentations collected the Clifford process at the end of the circuit rather than the start - for unitary circuits we can choose to have them at either end, but in the general case we may not always be able to transport qubit initialisations through the Pauli rotations to the end.

We can trivially get any circuit into this form, representing it as a set of qubit initialisations followed by the sequence of Pauli rotations formed by decomposing each gate in turn.

Gate	Equivalent Pauli rotations (up to global phase)
CX_{ct}	$e^{-i\frac{\pi}{4}Z_cX_t}e^{i\frac{\pi}{4}Z_c}e^{i\frac{\pi}{4}X_t}$
CZ_{ct}	$e^{-i\frac{\pi}{4}Z_cZ_t}e^{i\frac{\pi}{4}Z_c}e^{i\frac{\pi}{4}Z_t}$
$RZ(\theta)$	$e^{-i\frac{\theta}{2}Z}$
$RX(\theta)$	$e^{-i\frac{\theta}{2}X}$
H	$e^{-i\frac{\pi}{4}Z}e^{-i\frac{\pi}{4}X}e^{-i\frac{\pi}{4}Z}$
CCX_{abt}	$e^{-i\frac{\pi}{8}Z_aZ_bX_t}e^{i\frac{\pi}{8}Z_aZ_b}e^{i\frac{\pi}{8}Z_aX_t}e^{i\frac{\pi}{8}Z_bX_t}e^{-i\frac{\pi}{8}Z_a}e^{-i\frac{\pi}{8}Z_b}e^{-i\frac{\pi}{8}X_t}$

Converting back to elementary gates can be just as easy: if the Clifford circuit is represented via a stabilizer tableau, we start by synthesising that back into a circuit [1, 84] and then apply a standard decomposition to each Pauli rotation [9, 36],

[14] The convention of using $e^{-i\frac{\theta}{2}P}$ rather than $e^{i\alpha P}$ is to give a closer semblance to rotation gates like RZ, RX, and their multi-qubit counterparts, and to make the notation easier once we get to extraction from measurement patterns, but it is just a notational choice.

although this will typically add an extremely high amount of redundant Clifford gates. More efficient synthesis can be performed using techniques for synthesising pairs of rotations simultaneously [36] or by diagonalising sets of mutually commuting rotations [115, 37]. It is also possible that future architectures may find efficient ways to perform each Pauli rotation natively or employ lattice surgery where it is practical to just perform them directly [83].

Decomposing each elementary gate into Pauli rotations will generate a representation that is no easier to handle than the original circuit, mostly in part due to the large number of Clifford operations that will still hold their place in the rotation list. A Pauli rotation is Clifford when the angle is an integral multiple of $\frac{\pi}{2}$ (i.e. the coefficient in the exponential is a multiple of $\frac{\pi}{4}$). The action of Clifford rotations on both Paulis and arbitrary Pauli rotations can be summarised in a few equations.

Lemma 4.3.1: Commutation Rules

For any Pauli strings $P, Q \in \{I, X, Y, Z\}^{\otimes n}$ and angles θ, ϕ, if P and Q commute, then

$$e^{-i\frac{\theta}{2}P}Q = Qe^{-i\frac{\theta}{2}P} \tag{4.51}$$

$$e^{-i\frac{\theta}{2}P}e^{-i\frac{\phi}{2}Q} = e^{-i\frac{\phi}{2}Q}e^{-i\frac{\theta}{2}P} \tag{4.52}$$

and otherwise (i.e. they anticommute)

$$e^{-i\frac{\pi}{4}P}Q = (-iPQ)e^{-i\frac{\pi}{4}P} \tag{4.53}$$

$$e^{-i\frac{\pi}{4}P}e^{-i\frac{\phi}{2}Q} = e^{-i\frac{\phi}{2}(-iPQ)}e^{-i\frac{\pi}{4}P} \tag{4.54}$$

Proof. Any operator satisfying $A^2 = I$ and real α permit the decomposition $e^{i\alpha A} = \cos\alpha I + i\sin\alpha A$ by grouping terms in the Taylor expansion. Each equation follows from decomposing one of the exponentials in this way. Whilst the right-hand side of Equation 4.54 appears to contain a real exponent, remember that $-iPQ$ is a real Pauli string when P and Q anticommute. □

The above rules can be used to move any Clifford rotation to the start, leaving us with a sequence of Pauli rotations describing the actions of the non-Clifford operations in the circuit.

They also describe when rotations of any angle can commute with each other. We can see the commutations as a point of redundancy in the representation. Because anticommuting Pauli strings prevent commutation of their exponentials, any valid ordering of the rotations will preserve the relative order of any pair with anticommuting strings, inducing a temporal dependency between them. Taking the transitive closure of these dependencies gives a partial order representing the sequence up to any number of commutations.

Definition 4.3.2: Pauli graph

A *Pauli graph* is a data structure representing a quantum circuit, consisting of:

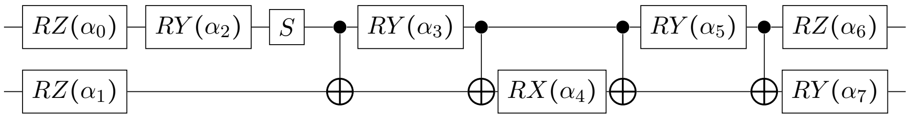
- A stabilizer tableau describing a Clifford circuit C. In general, C may be an isometry, which we can represent using the tableau of its Choi operator.
- A directed acyclic graph describing a partial order $\prec$ over pairs $\{(P_k, \theta_k)\}_k$ ($P_k \in \{I, X, Y, Z\}^{\otimes n}$, $\theta_k \in \mathbb{R}$). If (P, θ) and (P', θ') are incomparable under $\prec$, then the Pauli strings P and P' must commute (and, therefore, so do the rotations).

The linear map of the circuit is given by $\left(\prod_k^{\succ} e^{-i\frac{\theta_k}{2}P_k}\right) C$ up to global phase.

We will draw the ordering relations $\prec$ via Hasse diagrams for simplicity. Implementations of Pauli graphs as a data structure may prefer storing the full relation for practical reasons such as making the transport of Clifford operations (Equation 4.54) a local change that does not require recalculating the entire graph [33].

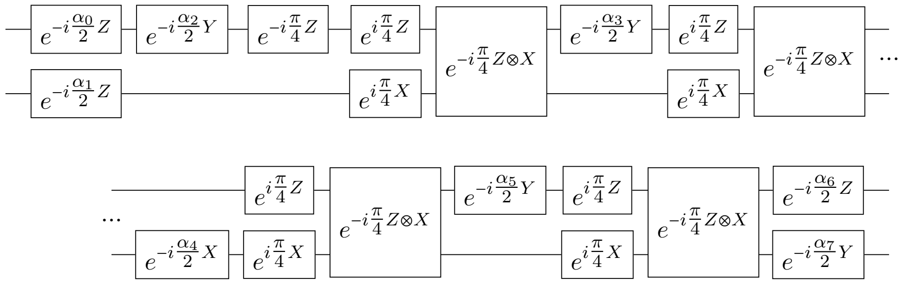
Example 4.3.3

Suppose we start with the following circuit:



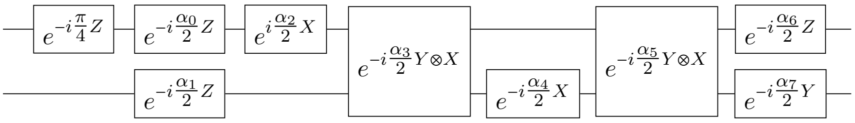
(4.55)

Decomposing each gate into Pauli rotations gives a rather long form.


$e^{-i\frac{\alpha_0}{2}Z}$
$e^{-i\frac{\alpha_2}{2}Y}$
$e^{-i\frac{\pi}{4}Z}$
$e^{i\frac{\pi}{4}Z}$
$e^{-i\frac{\pi}{4}Z\otimes X}$
$e^{-i\frac{\alpha_3}{2}Y}$
$e^{i\frac{\pi}{4}Z}$
$e^{-i\frac{\pi}{4}Z\otimes X}$
...
$e^{-i\frac{\alpha_1}{2}Z}$
$e^{i\frac{\pi}{4}X}$
$e^{i\frac{\pi}{4}X}$
...
$e^{i\frac{\pi}{4}Z}$
$e^{-i\frac{\pi}{4}Z\otimes X}$
$e^{-i\frac{\alpha_5}{2}Y}$
$e^{i\frac{\pi}{4}Z}$
$e^{-i\frac{\pi}{4}Z\otimes X}$
$e^{-i\frac{\alpha_6}{2}Z}$
$e^{-i\frac{\alpha_4}{2}X}$
$e^{i\frac{\pi}{4}X}$
$e^{i\frac{\pi}{4}X}$
$e^{-i\frac{\alpha_7}{2}Y}$

(4.56)

Now we move the Clifford-angled rotations to the start of the circuit. Let's start with the first $e^{-i\frac{\pi}{4}Z}$ from the S gate. Since Y and Z anticommute, Equation 4.54 applies giving $e^{-i\frac{\pi}{4}Z}e^{-i\frac{\alpha_2}{2}Y} = e^{-i\frac{\alpha_2}{2}(-iZY)}e^{-i\frac{\pi}{4}Z} = e^{i\frac{\alpha_2}{2}X}e^{-i\frac{\pi}{4}Z}$. It then commutes with the RZ rotations. Repeating this for the components of the CX gates (noting that they cancel), we end up with the following circuit:

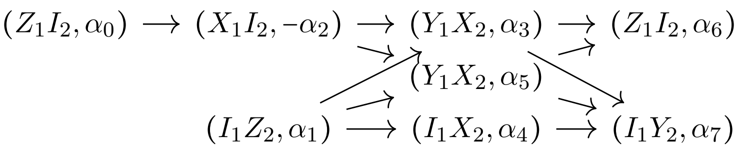

$e^{-i\frac{\pi}{4}Z}$
$e^{-i\frac{\alpha_0}{2}Z}$
$e^{i\frac{\alpha_2}{2}X}$
$e^{-i\frac{\alpha_3}{2}Y\otimes X}$
$e^{-i\frac{\alpha_5}{2}Y\otimes X}$
$e^{-i\frac{\alpha_6}{2}Z}$
$e^{-i\frac{\alpha_1}{2}Z}$
$e^{-i\frac{\alpha_4}{2}X}$
$e^{-i\frac{\alpha_7}{2}Y}$

(4.57)

This is in the product form we expect, with Clifford process $C = e^{-i\frac{\pi}{4}Z_1}$ followed by a sequence of non-Clifford rotations.

We can note that the rotations $e^{-i\frac{\alpha_3}{2}Y_1X_2}$, $e^{-i\frac{\alpha_4}{2}X_2}$, and $e^{-i\frac{\alpha_5}{2}Y_1X_2}$ mutually commute, but none can be moved before $e^{-i\frac{\alpha_1}{2}Z_2}$ or after $e^{-i\frac{\alpha_7}{2}Y_2}$. This is summarised by the graph below depicting the dependency relation $\prec$.

Ins		Outs		Sign
X		Y		+
	X		X	+
Z		Z		+
	Z		Z	+



(4.58)

Rewrite rules for Pauli graphs change the structure but preserve the linear map. The most useful rewrite for circuit reduction is rotation merging: two nodes in the graph j, k can be merged to give a single node with $(P_j, \theta_j + \theta_k)$ if $P_j = P_k$ and both $j \nprec k$ and $k \nprec j$ (i.e. there is some valid topological ordering of the graph in which these two rotations are adjacent).

Example 4.3.4

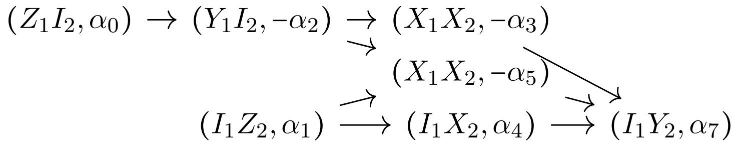
In the Pauli graph from Example 4.3.3, the nodes (Y_1X_2, α_3) and (Y_1X_2, α_5) can be merged into a single node $(Y_1X_2, \alpha_3 + \alpha_5)$. On the other hand, (Z_1I_2, α_0) and (Z_1I_2, α_6) cannot be merged since there are intervening rotations between them.

In addition to its use in constructing Pauli graphs, we can view Equation 4.54 as an explicit rewrite allowing the movement of Clifford rotations through the circuit. Combining this with merging/splitting of rotations and eliminating rotations with angle 0 permits a more rigid canonical form where the range of angles for each rotation is reduced to $\left(0, \frac{\pi}{2}\right)$. More explicitly, if (P_k, θ_k) has $\theta_k \geq \frac{\pi}{2}$, we can split it into two nodes $\left(P_k, \theta_k - \frac{\pi}{2}\right)$ and $\left(P_k, \frac{\pi}{2}\right)$ and move the latter through the graph into the tableau.

Example 4.3.5

Let's again take the Pauli graph from Example 4.3.3 and suppose that $\alpha_6 = \frac{\pi}{2}$ and we wish to move it into the tableau. The nodes (I_1X_2, α_4) and (I_1Y_2, α_7) are not predecessors of (Z_1I_2, α_6) so we can ignore them. For the rest, Z_1I_2 anticommutes with Y_1X_2 and X_1I_2 but commutes with Z_1I_2 and I_1Z_2. As we are moving a $e^{-i\frac{\pi}{2}Z_1I_2}$ rotation, Equation 4.54 updates Y_1X_2 to $-iZ_1I_2Y_1X_2 = -X_1X_2$ and X_1I_2 to $-iZ_1I_2X_1I_2 = Y_1I_2$. To update the tableau, we look at the output substring of each row and update it according to the Commutation Rules. In this case, only the top row (Y_1I_2) needs updating to $-iZ_1I_2Y_1I_2 = -X_1I_2$.

Ins		Outs		Sign
X		X		$-$
	X		X	$+$
Z		Z		$+$
	Z		Z	$+$



(4.59)

When the initial Clifford process includes some qubit initialisations, some rotations will have no observable effect. For example, $|0\rangle$ is an eigenstate for all RZ gates. The following lemma generalises this idea, showing that free stabilizers over the outputs of the Clifford process can be used to change the Pauli strings of rotations. The Commutation Rules can propagate stabilizers beyond the initial rotations to enable this rewrite to be performed elsewhere. Changing the Pauli strings of rotations in this way can affect how they commute/anticommute with their neighbours, modifying the dependency relation $\prec$.

This lemma will be of particular importance in Section 4.3.2 for its role in circuit extraction.

Lemma 4.3.6: Product Rotation Lemma

Let P and Q be commuting Pauli strings such that $QC = C$ for some linear map C. Then $e^{-i\frac{\theta}{2}P}C = e^{-i\frac{\theta}{2}PQ}C$.

Proof. For any analytic function $F(P)$, we can expand its Taylor series in $F(P)C$, then introduce and commute Q in each term to form the Taylor series for $F(PQ)C$.

In this case, using the expansion $e^{-i\frac{\theta}{2}P} = \cos\frac{\theta}{2}I - i\sin\frac{\theta}{2}P$:

$$\begin{aligned} e^{-i\frac{\theta}{2}P}C &= \cos\tfrac{\theta}{2}C - i\sin\tfrac{\theta}{2}PC \\ &= \cos\tfrac{\theta}{2}C - i\sin\tfrac{\theta}{2}PQC \qquad (4.60) \\ &= e^{-i\frac{\theta}{2}PQ}C \end{aligned}$$

□

Example 4.3.7

Taking the ongoing example from Example 4.3.3 and supposing that the second qubit is initialised to the $|0\rangle$ state at the beginning, we have the following Pauli graph:

Ins	Outs	Sign
X	Y	+
Z	Z	+
	Z	+

$$\begin{tikzcd}
(Z_1I_2, \alpha_0) \arrow[r] & (X_1I_2, -\alpha_2) \arrow[r] \arrow[dr] & (Y_1X_2, \alpha_3) \arrow[r] \arrow[ddr] & (Z_1I_2, \alpha_6) \\
 & & (Y_1X_2, \alpha_5) \arrow[ur] \arrow[dr] & \\
 & (I_1Z_2, \alpha_1) \arrow[r] \arrow[ur] \arrow[uur] & (I_1X_2, \alpha_4) \arrow[r] & (I_1Y_2, \alpha_7)
\end{tikzcd} \tag{4.61}$$

If I_1Z_2 is a stabilizer of C, then it is also a stabilizer of $e^{-i\frac{\alpha_0}{2}Z_1I_2}C$. Since it also commutes with the string X_1I_2, we can apply the Product Rotation Lemma to update that node to $(X_1Z_2, -\alpha_2)$. We need to subsequently update the graph now that X_1Z_2 commutes with Y_1X_2 and it no longer commutes with I_1X_2.

Similarly, we can update (I_1Z_2, α_1) to (I_1I_2, α_1) - this is just a global phase so we can freely remove it. Since I_1Z_2 anticommutes with Y_1X_2 and I_1X_2, we can't propagate it any further.

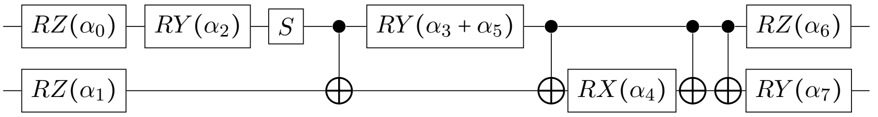
Ins	Outs	Sign
X	Y	+
Z	Z	+
	Z	+

$$\begin{tikzcd}
(Z_1I_2, \alpha_0) \arrow[rr] \arrow[dr] \arrow[drr] & & (Y_1X_2, \alpha_3) \arrow[r] \arrow[ddr] & (Z_1I_2, \alpha_6) \\
 & (X_1Z_2, -\alpha_2) \arrow[dr] & (Y_1X_2, \alpha_5) \arrow[ur] \arrow[dr] & \\
 & & (I_1X_2, \alpha_4) \arrow[r] & (I_1Y_2, \alpha_7)
\end{tikzcd} \tag{4.62}$$

Similar to the phase teleportation procedure in ZX-calculus [79] or phase folding [5], one can consider using this in circuit optimisation where we retain the layout of a circuit and just use the Pauli graph to spot where non-adjacent gates can be merged. This is typically good when the original circuit has a relatively low density of Clifford gates when it is unlikely that resynthesis will give as efficient a circuit.

Example 4.3.8

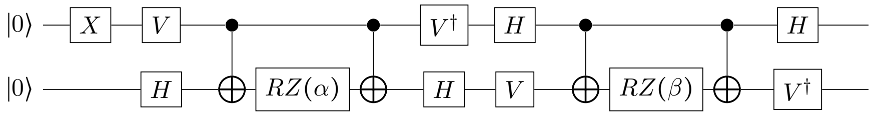
Example 4.3.4 showed that two rotations in our ongoing example could be merged. By recording which rotations came from which gates in the original circuit, we can merge the gates even though they are not adjacent. This maintains the overall structure of the circuit but enables some very simple peephole optimisation to further reduce the gate count.



(4.63)

Example 4.3.9

Suppose we have the following gates at the start of a circuit run from an initial $|0\rangle^{\otimes n}$ state.



(4.64)

This has a pretty basic Pauli graph; there are only two rotations which commute with each other.

Outs		Sign
Z		$-$
	Z	$+$

$$(Y_1X_2,\alpha)$$
$$(X_1Y_2,\beta) \qquad (4.65)$$

The product stabilizer $-Z_1Z_2$ commutes with X_1Y_2 so the Product Rotation Lemma can be applied, mapping the rotation to $(-Y_1X_2,\beta) = (Y_1X_2,-\beta)$. The two rotations can then be merged into $(Y_1X_2,\alpha-\beta)$. Reflecting this back in the circuit changes the $RZ\,(\alpha)$ gate to $RZ\,(\alpha-\beta)$ and removes the $RZ\,(\beta)$ gate. This allows the second half of the circuit to be entirely removed using gate-inverse cancellation. This example achieves the same result as reducing the terms in an excitation operator modulo the stabilizers (such as done during qubit tapering [18]) but can be applied to arbitrary circuits.

4.3.2 Extracting Measurement Rotations

We turn to the task of taking a measurement pattern described by a labelled open graph Γ and measurement angles α, and searching for an ancilla-free pure quantum circuit that implements the same linear map.

To motivate our method, recall the possible measurement bases from Equations 4.6 and 4.7. Each planar basis can be constructed as a basic rotation applied to some Pauli basis:

$$\begin{aligned} \langle \pm_{XY,\alpha}| &\approx \langle \pm_{X,0}| e^{i\frac{\alpha}{2}Z} \\ \langle \pm_{XZ,\alpha}| &\approx \langle \pm_{Z,0}| e^{i\frac{\alpha}{2}Y} \\ \langle \pm_{YZ,\alpha}| &\approx \langle \pm_{Z,0}| e^{-i\frac{\alpha}{2}X} \end{aligned} \tag{4.66}$$

In each case, the rotation is about the Pauli orthogonal to the measurement plane and therefore coincides with $P^{p(v)\to v}$ for any Pauli flow $(p,\prec)$. We can summarise these by

$$\langle +_{\lambda(v),\alpha(v)}| \approx \langle +_{\lambda(v),0}| e^{(-1)^{D_v} i\frac{\alpha(v)}{2}P^{p(v)\to v}} \tag{4.67}$$

$$D_v := \begin{cases} 1 & \lambda(v) = YZ \\ 0 & \text{otherwise} \end{cases} \tag{4.68}$$

where D_v dictates the direction of rotation about the Bloch sphere. The equation holds for Pauli measurements, where $P^{p(v)\to v}$ may be either orthogonal Pauli, reflecting the flexibility of the Pauli flow rules $[\lambda.X]$, $[\lambda.Z]$, and $[\lambda.Y]$.

The key idea driving the method of circuit extraction presented here is to apply the Product Rotation Lemma to alter these rotations and move them onto the output qubits one at a time. This requires identifying an appropriate stabilizer of the resource state to use.

To remove the rotation from qubit v, the stabilizer must contain $P_v^{p(v)\to v}$, so let's just look at the correcting stabilizers from the Pauli flow. If the stabilizer only uses v and the outputs (i.e. we can immediately use it to extract the rotation) then v must be at measurement depth 1, so we need something a little more general for this to work for all vertices. Fortunately, if we assume all future vertices have already had their measurement angles set to zero (making the corresponding projectors act in a Pauli basis) we can use them alongside any Pauli measurements to absorb additional Paulis.

Definition 4.3.10: Extraction string

Let (Γ,α) describe a measurement pattern with Pauli flow $(p,\prec)$ and choose a measured vertex $v \in \overline{O}$. A Pauli string Q is a P*-extraction string* ($P \in \{X,Y,Z\}$) for v if $P_v Q_O$ is a stabilizer of the following linear map:

$$\left(\prod_{\substack{u\in\overline{O}\\u>v\\\lambda(u)\in\{XY,XZ,YZ\}}} \langle +_{\lambda(u),0}|_u\right)\left(\prod_{\substack{u\in\overline{O}\setminus\{v\}\\\lambda(u)\in\{X,Y,Z\}}} \langle +_{\lambda(u),\alpha(u)}|_u\right)\left(\prod_{u\sim w} CZ_{u,w}\right)\left(\prod_{u\in\overline{I}} |+\rangle_u\right) \tag{4.69}$$

A *primary extraction string* for v is a P-extraction string for any Pauli P orthogonal to $\lambda(v)$.

A focussed flow guarantees that the components of the stabilizer on other measured qubits can always be absorbed by the projectors to leave something in the form of an extraction string.

Lemma 4.3.11

Given a Pauli flow $(p,<)$ for a labelled open graph (G,I,O,λ), for any vertex $v\in\overline{O}$ the size of $p(v)\cap\mathrm{Odd}(p(v))$ is even.

Proof. Let $G=(V,E)$ and define the subgraph $G'=(p(v),E\cap(p(v)\times p(v)))$. For any vertex $u\in p(v)$, $u\in\mathrm{Odd}(p(v))$ if and only if u has odd degree in G'. Since the sum of the vertex degrees must equal 2 times the number of edges in G', there must be an even number of vertices with odd degree. □

Lemma 4.3.12

Let $\Gamma=(G,I,O,\lambda)$ be a labelled open graph with some measurement angles $\alpha:\overline{O}\to[0,2\pi)$ and a focussed Pauli flow $(p,<)$. Then for any vertex $v\in\overline{O}$,

$$\mathrm{ES}_O^{p;v} := (-1)^{a+b+c}\prod_{u\in O} P_u^{p(v)\to u} \tag{4.70}$$

with

$$a = |E\cap(p(v)\times p(v))| \tag{4.71}$$
$$b = |p(v)\cap\mathrm{Odd}(p(v))|/2 \tag{4.72}$$
$$c = \left|(p(v)\cup\mathrm{Odd}(p(v)))\cap\left\{u\in\overline{O}\;\middle|\;\lambda(u)\in\{X,Y,Z\}\wedge\alpha(u)=\pi\right\}\right| \tag{4.73}$$

is a primary extraction string for v.

Proof. Consider an arbitrary measured vertex $v \in \overline{O}$.

Since each correction set $p(v)$ is a subset of the non-input vertices, we can combine their corresponding resource state stabilizers. We may reorder the Z and X terms with the possible introduction of a (-1).

$$\left(\prod_{u\sim w} CZ_{u,w}\right)\left(\prod_{u\in\overline{I}} |+\rangle_u\right) = (-1)^a \left(\prod_{u\in p(v)} X_u\right)\left(\prod_{u\in \mathrm{Odd}(p(v))} Z_u\right)\left(\prod_{u\sim w} CZ_{u,w}\right)\left(\prod_{u\in\overline{I}} |+\rangle_u\right) \tag{4.74}$$

where a is the number of edges in the subgraph of $p(v)$, since for each edge here we have to reorder the Z and X terms on exactly one of the two vertices.

There are an even number of vertices with both an X and a Z in this stabilizer (Lemma 4.3.11), so we can apply $Y = iXZ$ on all such instances, again introducing a possible (-1) term.

$$\begin{aligned}\left(\prod_{u\sim w} CZ_{u,w}\right)\left(\prod_{u\in\overline{I}} |+\rangle_u\right) &= (-1)^{a+b} \left(\prod_{u\in p(v)\setminus \mathrm{Odd}(p(v))} X_u\right)\left(\prod_{u\in \mathrm{Odd}(p(v))\setminus p(v)} Z_u\right) \\ &\qquad \left(\prod_{u\in p(v)\cap \mathrm{Odd}(p(v))} Y_u\right)\left(\prod_{u\sim w} CZ_{u,w}\right)\left(\prod_{u\in\overline{I}} |+\rangle_u\right) \\ &= (-1)^{a+b}\left(\prod_{u\in V} P_u^{p(v)\to u}\right)\left(\prod_{u\sim w} CZ_{u,w}\right)\left(\prod_{u\in\overline{I}} |+\rangle_u\right)\end{aligned} \tag{4.75}$$

Since $(p, \prec)$ is a Pauli flow, every vertex in $p(v)$ or $\mathrm{Odd}(p(v))$ is either v itself, a Pauli measurement, or greater than v in $\prec$ (from conditions $[\prec .X]$ and $[\prec .Z]$). To fit the form an extraction string, we consider adding the corresponding projectors. Since $(p, \prec)$ is focussed, each vertex (besides outputs and v) in $p(v) \setminus \mathrm{Odd}(p(v))$ is projected into an X basis eigenvector, and similarly $\mathrm{Odd}(p(v)) \setminus p(v)$ into Z and $p(v) \cap \mathrm{Odd}(p(v))$ into Y. This means we can absorb these Pauli operators into the projectors, again with the possible introduction of a (-1).

$$\begin{aligned} &\left(\prod_{\substack{u \in \overline{O} \\ u \succ v \\ \lambda(u) \in \{XY, XZ, YZ\}}} \langle +_{\lambda(u), 0}|_u \right) \left(\prod_{\substack{u \in \overline{O} \setminus \{v\} \\ \lambda(u) \in \{X, Y, Z\}}} \langle +_{\lambda(u), \alpha(u)}|_u \right) \left(\prod_{u \sim w} CZ_{u,w} \right) \left(\prod_{u \in \overline{I}} |+\rangle_u \right) \\ = \; &(-1)^{a+b+c} \left(\prod_{u \in O \cup \{v\}} P_u^{p(v) \to u} \right) \\ &\left(\prod_{\substack{u \in \overline{O} \\ u \succ v \\ \lambda(u) \in \{XY, XZ, YZ\}}} \langle +_{\lambda(u), 0}|_u \right) \left(\prod_{\substack{u \in \overline{O} \setminus \{v\} \\ \lambda(u) \in \{X, Y, Z\}}} \langle +_{\lambda(u), \alpha(u)}|_u \right) \left(\prod_{u \sim w} CZ_{u,w} \right) \left(\prod_{u \in \overline{I}} |+\rangle_u \right) \end{aligned} \tag{4.76}$$

This stabilizer is exactly $\mathrm{ES}_O^{p;v}$ over the outputs along with $P_v^{p(v) \to v}$ which is guaranteed to be orthogonal to $\lambda(v)$ by the Pauli flow conditions. □

Viewing these extracted rotations as the sequence of rotations in a Pauli graph, the following result shows that the commutativity of the rotations is captured precisely by the corrections used in the focussed Pauli flow; therefore, the partial order over the planar measurements in the measurement pattern precisely coincides with the partial order over the rotations in the extracted Pauli graph.

Lemma 4.3.13

Let (Γ, α) describe a measurement pattern with a focussed Pauli flow $(p, \prec)$. Then for any vertices $u, v \in \overline{O}$ the primary extraction strings satisfy

$$\mathrm{ES}_O^{p;u} \mathrm{ES}_O^{p;v} = (-1)^{F^{p(u) \to v} + F^{p(v) \to u}} \mathrm{ES}_O^{p;v} \mathrm{ES}_O^{p;u} \tag{4.77}$$

where $F^{p(x) \to y} := |\{y\} \cap (p(x) \cup \mathrm{Odd}(p(x)))|$ indicates whether y is used in the correction of x ($P^{p(x) \to y} \neq I$).

Proof. Since $\mathrm{ES}_O^{p;u}$ and $\mathrm{ES}_O^{p;v}$ are tensor products of Pauli matrices, they must either commute or anticommute. Consider the linear map:

$$\left(\prod_{\substack{w\in\overline{O}\setminus\{u,v\}\\ \lambda(w)\notin\{X,Y,Z\}\\ u<w\vee v<w}} \langle +_{\lambda(w),0}|_w\right)\left(\prod_{\substack{w\in\overline{O}\setminus\{u,v\}\\ \lambda(w)\in\{X,Y,Z\}}} \langle +_{\lambda(w),\alpha(w)}|_w\right)\left(\prod_{u\sim w} CZ_{u,w}\right)\left(\prod_{u\in\overline{I}} |+\rangle_u\right) \tag{4.78}$$

This may not be in the exact form needed to introduce extraction strings for u and v if they use each other for corrections.

We can still follow the construction in the proof of Lemma 4.3.12 to deduce that $(-1)^{A^{p(u)\to v}} P_u^{p(u)\to u} P_v^{p(u)\to v} \mathrm{ES}_O^{p;u}$ and $(-1)^{A^{p(v)\to u}} P_u^{p(v)\to u} P_v^{p(v)\to v} \mathrm{ES}_O^{p;v}$ are stabilizers where

$$A^{p(x)\to y} := \begin{cases} F^{p(x)\to y} & \lambda(y) \in \{X,Y,Z\} \wedge \alpha(y) = \pi \\ 0 & \text{otherwise} \end{cases} \tag{4.79}$$

Comparing conditions $[\lambda.XY]$-$[\lambda.Y]$ and the focussed conditions, we see that, for any vertex $x \in \overline{O}$, $P^{p(x)\to x}$ is orthogonal to $\lambda(x)$ and hence anticommutes with any $P^{p(y)\to x} \neq I$. Equationally, this gives $P^{p(x)\to x} P^{p(y)\to x} = (-1)^{F^{p(y)\to x}} P^{p(y)\to x} P^{p(x)\to x}$.

Finally, we use the fact that the stabilizer group is abelian:

$$\begin{aligned}
&(-1)^{A^{p(u)\to v}+A^{p(v)\to u}} P_u^{p(u)\to u} P_v^{p(u)\to v} \mathrm{ES}_O^{p;u} P_u^{p(v)\to u} P_v^{p(v)\to v} \mathrm{ES}_O^{p;v} \\
=&(-1)^{A^{p(v)\to u}+A^{p(v)\to u}} P_u^{p(v)\to u} P_v^{p(v)\to v} \mathrm{ES}_O^{p;v} P_u^{p(u)\to u} P_v^{p(u)\to v} \mathrm{ES}_O^{p;u} \\
=&(-1)^{F^{p(u)\to v}+F^{p(v)\to u}+A^{p(u)\to v}+A^{p(v)\to u}} P_u^{p(u)\to u} P_v^{p(u)\to v} \mathrm{ES}_O^{p;v} P_u^{p(v)\to u} P_v^{p(v)\to v} \mathrm{ES}_O^{p;u}
\end{aligned} \tag{4.80}$$

i.e. $\mathrm{ES}_O^{p;u}\mathrm{ES}_O^{p;v} = (-1)^{F^{p(u)\to v}+F^{p(v)\to u}} \mathrm{ES}_O^{p;v}\mathrm{ES}_O^{p;u}$. □

We can extract the rotations one at a time until we are left with the following Clifford process:

$$\left(\prod_{\substack{u\in\overline{O}\\ \lambda(u)\notin\{X,Y,Z\}}} \langle +_{\lambda(u),0}|_u\right)\left(\prod_{\substack{u\in\overline{O}\\ \lambda(u)\in\{X,Y,Z\}}} \langle +_{\lambda(u),\alpha(u)}|_u\right)\left(\prod_{u\sim w} CZ_{u,w}\right)\left(\prod_{u\in\overline{I}} |+\rangle_u\right) \tag{4.81}$$

If we can identify the stabilizer tableau of its Choi operator, we will complete the Pauli graph description which we can then synthesise into a circuit.

4.3.3 Characterising Clifford Patterns

The Choi operator of a Clifford process C with $|I|$ inputs and $|O|$ outputs is a $(|I| + |O|)$-qubit state. To identify the stabilizer group, we need to find $|I| + |O|$ independent generators. In general this process will be an isometry, giving $|I| \leq |O|$. In order to span the full Pauli group on the input space, we can always find $|I|$ generators of the form (X_u, P) where $X_u = C^\dagger P C$ for each $u \in I$, and the same for each Z_u. The remaining $|O| - |I|$ generators can be chosen to act as the identity over the input space. This form is used in the examples of Pauli graphs shown in Section 4.3.1 and will help guide our search for a characterisation of the process in Equation 4.81.

Since Z on an input will commute through the CZ gates, we can get the corresponding stabilizer row from the primary extraction string assuming $P^{p(u) \to u} = Z$ for this input. This is guaranteed since the correction set $p(u)$ cannot contain any inputs, including u itself.

We can use the same trick to obtain the X rows from primary extraction strings. In this case, we find an altered pattern (Γ', α') implementing the same linear map up to a Hadamard gate on the input of our chosen vertex - the Hadamard maps X to a Z, meaning we can use the extraction string of the corresponding input to (Γ', α'). One way to achieve this is by *input extension*.

Definition 4.3.14: Input extension

Given an open graph (G, I, O), *input extension* about $u \in I$ adds a new vertex u' that is only connected to u and replaces u in the input set.

$$V' = V \cup \{u'\} \tag{4.82}$$
$$E' = E \cup \{u' \sim u\} \tag{4.83}$$
$$I' = (I \cup \{u'\}) \setminus \{u\} \tag{4.84}$$

Proposition 4.3.15

Let (Γ, α) describe a measurement pattern with some chosen input $u \in I$. Then let (Γ', α') be the measurement pattern formed by taking an input extension about u and setting $\lambda' := \lambda \cup \{u' \mapsto XY\}$ and $\alpha' := \alpha \cup \{u' \mapsto 0\}$. The linear map implemented by (Γ', α') is identical to that of (Γ, α) with a Hadamard gate applied to the input u'.

Proof. It is enough to verify the following equation, where the right-hand side matches the difference between the measurement patterns:

$$H = \sqrt{2} \langle +_{XY,0}|_{u'} CZ_{u,u'} |+_{X,0}\rangle_u \tag{4.85}$$

□

Lemma 4.3.16: Generalisation of [7, Lemma 3.8]

Let Γ and Γ' be labelled open graphs related by an input extension on vertex $u \in I$ (generating $u' \in I'$). If Γ has a Pauli flow, then so does Γ'.

Proof. Suppose Γ has a Pauli flow $(p, \prec)$. Let $p' = p \cup \{u' \mapsto \{u\}\}$ and $\prec'$ be the transitive closure of $\prec \cup \{(u', w) \mid w \in N_G(u) \cup \{u\}\}$.

For any $v \in V \setminus O$, $u' \notin p'(v) = p(v)$ and $u' \notin \mathrm{Odd}_{G'}(p'(v)) = \mathrm{Odd}_G(p(v))$ since its only neighbour u is an input in Γ which could not appear in any correction sets from p. This allows us to inherit all the Pauli flow properties from $(p, \prec)$ for v as they have remained unchanged.

For u', the definition of $\prec'$ guarantees the $[\prec.P]$ conditions since $u' \prec' w$ for every $w \in N_G(u) \cup \{u\} = (\mathrm{Odd}_{G'}(p'(u')) \setminus \{u'\}) \cup p'(u')$. $p'(u')$ also satisfies $[\lambda.XY]$ by construction. □

The remaining stabilizer generators (those that act just over the outputs) can be obtained from the focussed sets of the measurement pattern.

Lemma 4.3.17

Given a measurement pattern (Γ, α), for any focussed set $\hat{p}$ of Γ,

$$\mathrm{ES}_O^{\hat{p}} \coloneqq (-1)^{a+b+c} \prod_{u \in O} P_u^{\hat{p} \to u} \tag{4.86}$$

is a stabilizer of the linear map in Equation 4.81 where a, b, c are defined as in Lemma 4.3.12.

Proof. The proof is similar to Lemma 4.3.12. After applying the measurement projectors, the only Pauli terms that are not absorbed into the projectors are over the output qubits. □

Lemma 4.2.14 implies that we can get $|O| - |I|$ generators for the group of focussed sets, and we are looking for $|O| - |I|$ generators for the free stabilizers. For this to work, we need the generators for the focussed sets to not degenerate when mapped into stabilizers, i.e. the map $\mathrm{ES}_O^{(-)}$ must be injective. We show this using the following relationship with the extraction strings to ensure that only the empty set is mapped to the identity.

Lemma 4.3.18

Let (Γ, α) describe a measurement pattern with a focussed Pauli flow $(p, \prec)$. Then for any vertex $u \in \overline{O}$ and focussed set $\hat{p}$, the respective primary extraction string and stabilizer satisfy

$$\mathrm{ES}_O^{p;u} \mathrm{ES}_O^{\hat{p}} = (-1)^{F^{\hat{p}\to u}} \mathrm{ES}_O^{\hat{p}} \mathrm{ES}_O^{p;u} \tag{4.87}$$

where $F^{\hat{p}\to u} := |\{u\} \cap (\hat{p} \cup \mathrm{Odd}(\hat{p}))|$ indicates whether u is used in the generation of the stabilizer.

Proof. The proof is virtually the same as that of Lemma 4.3.13, though we only need care about anticommuting Paulis on u. □

Lemma 4.3.19

Given a labelled open graph $\Gamma = (G, I, O, \lambda)$ with a focussed Pauli flow and any focussed set $\hat{p}$, then $(\hat{p} \cup \mathrm{Odd}(\hat{p})) \cap O = \varnothing \iff \hat{p} = \varnothing$.

Proof. $\hat{p} = \varnothing \implies (\hat{p} \cup \mathrm{Odd}(\hat{p})) \cap O = \varnothing$ is trivial. For the other direction, suppose for a contradiction that we have a non-empty $\hat{p}$ for which $(\hat{p} \cup \mathrm{Odd}(\hat{p})) \cap O = \varnothing$, so we have some $v \in \hat{p} \cap \overline{O}$ and the corresponding stabilizer $\mathrm{ES}_O^{\hat{p}}$ is the identity. However, Lemma 4.3.18 implies that $\mathrm{ES}_O^{\hat{p}}$ must anticommute with $\mathrm{ES}_O^{p;v}$, contradicting the identity assumption. □

We go one step further by showing that the map from focussed sets to stabilizers also preserves the group action.

Lemma 4.3.20

Let $\Gamma = (G, I, O, \lambda)$ be a labelled open graph. For any pair of focussed sets $\hat{p}$, $\hat{q}$, $\mathrm{ES}_O^{\hat{p}} \mathrm{ES}_O^{\hat{q}} \approx \mathrm{ES}_O^{\hat{p}\Delta\hat{q}}$.

Proof. This follows from the anticommutativity of X and Z and the decomposition $\mathrm{Odd}(\hat{p}\Delta\hat{q}) = \mathrm{Odd}(\hat{p}) \Delta \mathrm{Odd}(\hat{q})$.

$$\begin{aligned}
\mathrm{ES}_O^{\hat{p}}\mathrm{ES}_O^{\hat{q}} &\approx \left(\prod_{w\in O\cap\hat{p}} X_w\right)\left(\prod_{w\in O\cap\mathrm{Odd}(\hat{p})} Z_w\right)\left(\prod_{w\in O\cap\hat{q}} X_w\right)\left(\prod_{w\in O\cap\mathrm{Odd}(\hat{q})} Z_w\right) \\
&\approx \left(\prod_{w\in O\cap\hat{p}} X_w\right)\left(\prod_{w\in O\cap\hat{q}} X_w\right)\left(\prod_{w\in O\cap\mathrm{Odd}(\hat{p})} Z_w\right)\left(\prod_{w\in O\cap\mathrm{Odd}(\hat{q})} Z_w\right) \qquad (4.88) \\
&= \left(\prod_{w\in O\cap(\hat{p}\Delta\hat{q})} X_w\right)\left(\prod_{w\in O\cap\mathrm{Odd}(\hat{p}\Delta\hat{q})} Z_w\right) \\
&\approx \mathrm{ES}_O^{\hat{p}\Delta\hat{q}}
\end{aligned}$$

□

Lemma 4.3.21

For any non-zero linear map C with stabilizers A and B, if $A \approx B$ then $A = B$.

Proof. Let $B = e^{i\theta}A$. Since stabilizers form a group, $AB = A\left(e^{i\theta}A\right) = e^{i\theta}I$ is also a stabilizer, i.e. $C = e^{i\theta}C$. $e^{i\theta} \neq 1$ implies C is a zero map, so we must have $A = B$. □

Proposition 4.3.22

$\mathrm{ES}_O^{(-)}$ is a group monomorphism from the group of focussed sets on a labelled open graph to the stabilizer group of the corresponding linear map.

Proof. The group action is preserved due to Lemmas 4.3.20 and 4.3.21. By Lemma 4.3.19, the map is injective (if we had two distinct focussed sets with the same stabilizer, combining them with Δ would give a non-empty focussed set that maps to the identity stabilizer). □

Now that we definitely obtain $|O| - |I|$ independent stabilizers over the output system from the focussed sets, combining these with the $2|I|$ rows from the focussed Pauli flow for the inputs and their extensions, we now have the full tableau for the Choi operator of a Clifford measurement pattern. All that remains is to put all the steps together to present a formal procedure for circuit extraction.

4.3.4 A Complete Algorithm

Combining the extraction of planar measurement angles and characterisation of the remaining Clifford process, we can rewrite the measurement pattern's linear map into the form of a Pauli graph. Algorithm 3 summarises the method following these steps:

1. Extend the graph with all input extensions. Identify a focussed Pauli flow $(p, <)$ (using the Pauli Flow Identification Algorithm and Focussing Lemma) along with generators for the focussed sets (using the Focussed Set Identification Algorithm).

2. Starting from the outputs and working backwards, use the Pauli flow and Product Rotation Lemma to move the rotation from each planar measurement to the outputs.

3. Complete the data for a Pauli graph representation by interpreting the corrections for the input measurements (and the input extensions) and the focussed sets as rows of a stabilizer tableau:

   - For each $u \in I$, the row $(Z_u, \mathrm{ES}_O^{p;u})$.
   - For each $u \in I$, the row $\left(X_u, \mathrm{ES}_O^{p;u'}\right)$ where u' is the new vertex from an input extension at u.
   - For each $\hat{p}$, the stabilizer $\mathrm{ES}_O^{\hat{p}}$.

4. Synthesise the resulting Pauli graph component-wise.

Theorem 4.3.23: Circuit Extraction Theorem

Let (Γ, α) describe a measurement pattern where Γ has a Pauli flow. Then Algorithm 3 identifies an equivalent circuit (via its Pauli graph) requiring no ancillas, and it terminates in time polynomial in the number of vertices in Γ.

Proof. Correctness of the output is guaranteed by the Product Rotation Lemma applied to the extraction strings from Lemma 4.3.12 for the rotations, and correctness of the tableau rows follows from Lemmas 4.3.12 and 4.3.17.

To examine the complexity, we suppose an explicit Pauli flow is not given up front. The Pauli Flow Identification Algorithm runs in $O\left(|V|^4\right)$ time. We can extend the flow to consider all of the input extensions simultaneously using Lemma 4.3.16 in $O(|I|)$ time (ignoring the update to $<$ since it won't affect the outcome of the extraction procedure). The Focussing Lemma focusses this Pauli flow in $O\left(|V|^3\right)$ time (we apply at most $O\left(|V|^2\right)$ updates, each of which takes $O(|V|)$ time to identify and apply). The Focussed Set Identification Algorithm also takes $O\left(|V|^3\right)$ time. Synthesising this into a circuit using the naive CX ladder construction takes $O(|V| \cdot |O|)$ time for the rotations from non-Pauli measurements, and $O\left(|O|^3\right)$ to synthesise a stabilizer tableau using

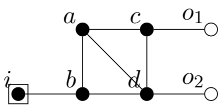
```
ExtractCircuit $(V, \Gamma, I, O, \lambda, \alpha)$ = begin
    $[V', \Gamma', I', O', \lambda'] := $ ExtendInputs $(V, \Gamma, I, O, \lambda]$;
    $[\_, p, d] := $ PauliFlow $(V', \Gamma', I', O', \lambda')$;
    $[p, d] := $ FocusFlow $(p, d)$;
    FSets := FocussedSets $(V, \Gamma, I, O, \lambda)$;
    Rotations := [];
    foreach $v \in V$.sort_by $(d)$ do
        if $\lambda(v) == YZ$ then Rotations.push_front $(\mathrm{ES}_O^{p;v}, \alpha(v))$;
        else Rotations.push_front $(\mathrm{ES}_O^{p;v}, -\alpha(v))$;
    end
    ChoiTab := CliffordTableau$(I \uplus O)$;
    foreach $i \in I$ do
        ChoiTab.add_row$\left(Z_i \cdot \mathrm{ES}_O^{p;i}\right)$;
        $i'$ := [Corresponding input in $I'$];
        ChoiTab.add_row$\left(X_i \cdot \mathrm{ES}_O^{p;i'}\right)$;
    end
    foreach $\hat{p} \in$ FSets do ChoiTab.add_row$\left(\mathrm{ES}_O^{\hat{p}}\right)$;
    return PauliGraph (ChoiTab, Rotations);
end
```

Algorithm 3: An algorithm for extracting a circuit in the form of a Pauli graph from a measurement pattern.

Aaronson and Gottesman's method [1] (when we have an isometry, its circuit will be some qubit initialisations followed by a unitary, and we can find such a unitary by adding more rows to the tableau until they span the full Pauli group over the outputs, which can be identified by Gaussian elimination in $O\left(|O|^3\right)$ time). The overall complexity is therefore dominated by the $O\left(|V|^4\right)$ Pauli flow identification. □

Example 4.3.24

Suppose we start with the following measurement pattern and focussed Pauli flow.



v	$\lambda(v)$	$p(v)$	$\mathrm{Odd}(p(v))$	$\{u \mid v \prec u\}$
i	XY	b, o_2	i, a	a, b, c, o_1, o_2
a	YZ	a, c, d, o_2	d, o_1, o_2	c, o_1, o_2
b	XY	c, d, o_1	b, d, o_1, o_2	c, o_1, o_2
c	XY	o_1	c	o_1
d	Y	o_2	d	o_2

(4.89)

Because we have two outputs and only one input, there is also an extra focussed set $\hat{p} = \{c, o_2\}$. We start by constructing the primary extraction strings for each vertex from Lemma 4.3.12. Let $a_d \in \{0,1\}$ be such that $\alpha(d) = a_d\pi$. Recall that we get one phase flip per edge between adjacent vertices in the correction/focussed set $(|E \cap (p(v) \times p(v))|)$, one phase flip for every two Ys that appear in the stabilizer $(|p(v) \cap \mathrm{Odd}(p(v))| / 2)$, and one for each term in the stabilizer that is absorbed from a Pauli measurement with angle π $(|(p(v) \cup \mathrm{Odd}(p(v))) \cap \{w \mid \lambda(w) \in \{X, Y, Z\} \wedge \alpha(w) = \pi\}|)$.

S	$S \cap O$	$\mathrm{Odd}(S) \cap O$	Edges	Y's	Paulis	ES_O^S
$p(i)$	o_2		0	0		I_1X_2
$p(a)$	o_2	o_1, o_2	4	2	d	$(-1)^{a_d+1} Z_1Y_2$
$p(b)$	o_1	o_1, o_2	2	2	d	$(-1)^{a_d+1} Y_1Z_2$
$p(c)$	o_1		0	0		X_1I_2
$p(d)$	o_2		0	0		I_1X_2
$\hat{p}$	o_2	o_1	0	0		Z_1X_2

(4.90)

We start by extracting the planar angles as rotations in $\succ$-order. We must start with c, extracting it as $e^{i\frac{\alpha(c)}{2}X_1I_2}$ over the outputs.

Having extracted this, the projector on c becomes $\langle +_{X,0}|$ so it can absorb the corresponding Paulis on the stabilizers from $p(a)$ and $p(b)$. Either a or b can be extracted next, giving $e^{(-1)^{a_d}\frac{\alpha(a)}{2}Z_1Y_2}$ and $e^{(-1)^{a_d+1}\frac{\alpha(b)}{2}Y_1Z_2}$ respectively. Note that the phase of the rotation from a got an extra -1 since $D_a = 1$ $(\lambda(a) = YZ)$.

The final rotation to be extracted is $e^{i\frac{\alpha(i)}{2}I_1X_2}$ from i. We don't extract any rotation from d since the measurement is already in a Pauli basis, and hence we can treat it as part of the leftover Clifford process.

We now turn to tableau for this Clifford process which we can read from the primary extraction strings. Γ can already be viewed as an input extension of $\Gamma \setminus \{i\}$ with input b, so we don't need to perform an input extension here to get that X on the input is mapped to $(-1)^{a_d+1} Y_1Z_2$ $(\mathrm{ES}_O^{p;b})$ over the outputs. Z on the input is mapped to I_1X_2 $(\mathrm{ES}_O^{p;i})$, and we have an extra row Z_1X_2 obtained from $\hat{p}$.

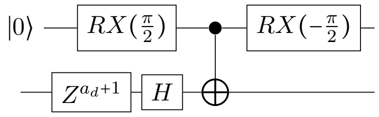
Ins	Outs		Sign
X	Y	Z	$(-)^{a_d+1}$
Z		X	$+$
	Z	X	$+$

$$\begin{tikzcd}
(I_1X_2, -\alpha(i)) \arrow[r] \arrow[dr] & (Z_1Y_2, -(-1)^{a_d}\alpha(a)) \arrow[r] & (X_1I_2, -\alpha(c)) \\
 & (Y_1Z_2, (-1)^{a_d}\alpha(b)) \arrow[ur] &
\end{tikzcd}$$

(4.91)

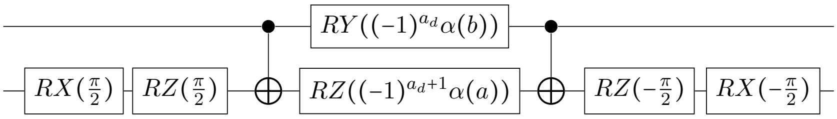
See how the dependency relation matches the $\prec$ ordering of the Pauli flow.

We can then synthesise a circuit from the Pauli graph component-wise. We start with the tableau, for which an example circuit is given below.



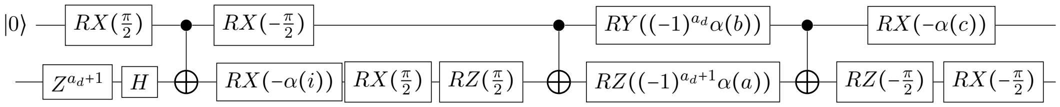
(4.92)

For the rotations, the $(I_1X_2, -\alpha(i))$ and $(X_1I_2, -\alpha(c))$ are simply RX gates. The other two rotations commute, and so can be efficiently diagonalised simultaneously with the following circuit:



(4.93)

Putting it all together, we obtain the final circuit:



(4.94)

Remark 4.3.25

It is relatively easy to see what structure/redundancy is being exploited when we move Clifford rotations around a Pauli graph or modifying rotations with the Product Rotation Lemma to optimise a circuit. On the other hand, graph-theoretic rewrites used to simplify measurement patterns have less obvious interpretations in how they relate to optimisations in the circuit model, especially through the lens of prior extraction algorithms [7] which would be very sensitive to changes in the pattern. To compare the two, we can consider extracting a Pauli graph from a measurement pattern before and after a rewrite to observe the changes in the order, Pauli strings, and phases of the rotations from each measurement or tableau row, and look for a sequence of simple rewrites on the Pauli graph that produces the same effect, i.e. makes the following diagram commute:

$$\begin{tikzcd}
\text{MBQC} \arrow[r, "\text{extract}"] \arrow[d, "\text{rewrite}"] & \text{Pauli graph} \arrow[d, "?"] \\
\text{MBQC} \arrow[r, "\text{extract}"] & \text{Pauli graph}
\end{tikzcd} \tag{4.95}$$

In the paper presenting the extraction algorithm [105], we proved that the following rewrites on measurement patterns can be simulated in the Pauli graph framework:

- Vertex relabelling: if a planar measurement is fixed to some angle $\alpha(u) \in \left\{0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}\right\}$, then it is equivalent to some Pauli measurement so we can change its label.

- Z vertex elimination: any vertex u measured in the Z basis can be removed from the pattern, up to some modifications of the angles on its neighbours and additional gates at any neighbour outputs.

- Local complementation and pivoting: similar to Z vertex elimination, this modifies the local connectivity of the graph (complementing the connectivity between neighbours of the chosen vertex) along with the measurements themselves and gates on outputs.

- Switching one focussed Pauli flow for another.

Any additional gates at the end are Clifford, which we can spawn along with their inverses and pull the inverses back through the Pauli graph into the Clifford circuit at the start. Any changes to the measurement angles give additional Clifford rotations throughout the Pauli graph, which can also be simulated by pulling their inverses back to the start. These account for all the changes to the Pauli strings of the rotations in the first three cases. As for switching to a different focussed Pauli flow, the differences always correspond to a combination of row updates to the tableau and applications of the Product Rotation Lemma between the Clifford process and the rotations in the Pauli graph.

The proofs are rather tedious but straightforward, following the steps:

1. Prove that the existence of Pauli flows is preserved by giving explicit updates to the focussed Pauli flows (and focussed sets);

2. Give the explicit updates to the primary extraction strings (and free stabilizers) for each rewrite;
3. Justify that the same updates are performed by the Pauli graph rewrite.

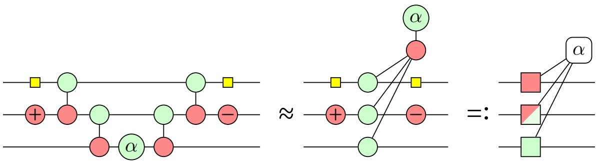
4.3.5 Interpretation in ZX-Calculus

This section will not contain any new results, instead serving as an alternative visual representation of the extraction algorithm to cement the reader's intuitions or support readers who are less comfortable with the linear algebra used so far.

The ZX-calculus [30] provides a completely diagrammatic way to reason about linear maps that arise in quantum computation with qubits. The generators can universally represent any such linear map and the associated rewrite rules give a sound and complete axiomatisation [91, 116] of equivalence of diagrams with respect to their maps. The intuitive nature of visualisations make it an attractive representation for teaching and exploring quantum computing. We will assume the reader is familiar with the fundamentals, and recommend van de Wetering's literature review [114] for a full introduction.

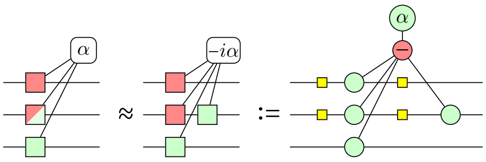
Throughout this section, we will ignore all global scalars with ≈ (both global phase and normalisation constants) and use the shorthand of ± on spiders to refer to $\pm\frac{\pi}{2}$.

The crux of the extraction algorithm is from the rules of Pauli rotations and their relationship with stabilizers through the Product Rotation Lemma. We will adopt the Pauli gadget notation from [36]. Starting with a standard circuit implementation of $e^{-i\frac{\alpha}{2}X_1Y_2Z_3}$, we can derive the form of the symmetric phase gadget [79] ($Z_1Z_2Z_3$ rotation) and define the new notation to just consider the changes of local basis.



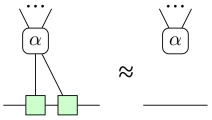
(4.96)

A red leg of a Pauli gadget indicates that it acts on that qubit in the X basis, and similarly a green leg for the Z basis or mixed colours for Y. We will also allow the phase to be imaginary, corresponding to a $\pm\frac{\pi}{2}$ phase on the internal phase gadget, allowing us to split Y legs according to $Y = -iZX$.

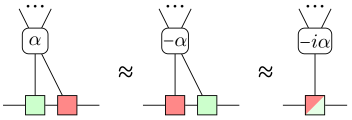


(4.97)

From these definitions, one can see that adjacent legs of a Pauli gadget behave in correspondence with the Pauli operators themselves in the sense that legs of the same type cancel out and legs of different types can be reordered up to a phase flip or combined to give the third kind.

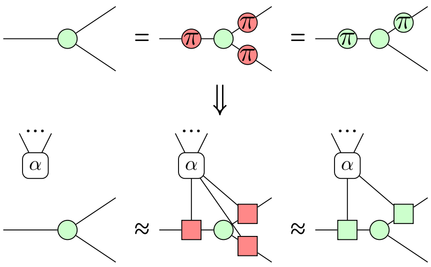


(4.98)

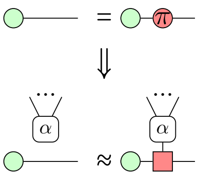


(4.99)

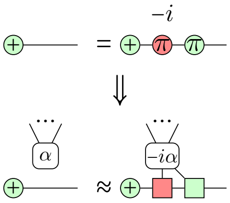
This goes hand-in-hand with the visual representation of the Product Rotation Lemma. Given each generator from the stabilizer fragment of the ZX-calculus (all phases are multiples of $\frac{\pi}{2}$), one can find a generating set of stabilizers and prove that, in each case, one can add/remove legs of a Pauli gadget exactly where π phases pop up.



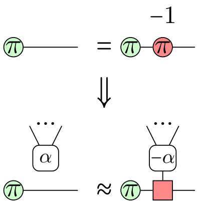
(4.100)



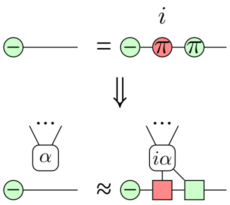
(4.101)



(4.102)

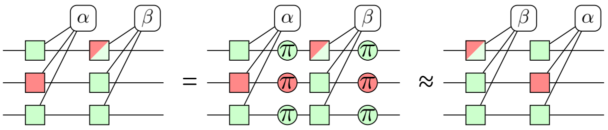


(4.103)

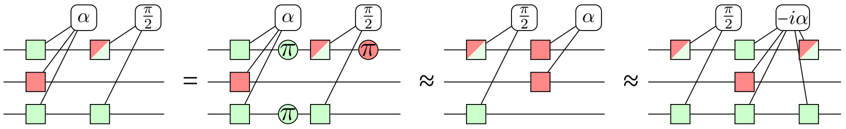


(4.104)

We can actually view the Commutation Rules themselves as special cases of the Product Rotation Lemma. The commutation of the Pauli strings just depends on how many of the qubit wires have different colours - an even number of differences means the Pauli strings commute and an odd number means they anticommute.

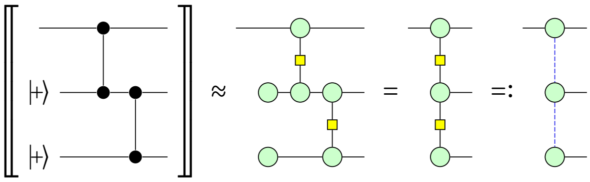


(4.105)



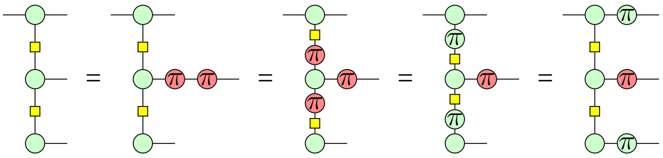
(4.106)

Now we turn to the measurement patterns themselves and interpret them in the ZX-calculus. The resource states are modelled by a graph-like diagram (a ZX-diagram with only green spiders connected by Hadamards), following from the fact that the $|+\rangle$ state is given by a 0 phase green spider, and the CZ gates are green spiders on each qubit with a Hadamard in between.



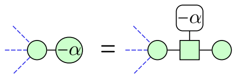
(4.107)

Since every spider here has 0 phase, the stabilizers of the resource state arise easily from the π-copy rule.

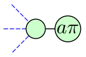


(4.108)

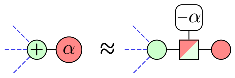
If we focus on the intended branch of the measurement pattern, it suffices to represent the measurements by an appropriate postselection. The form of this depends on the measurement label.

$\lambda(v) = XY \mapsto$ 

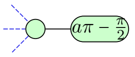
(4.109)

$\lambda(v) = X \mapsto$ 

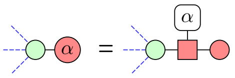
(4.110)

$\lambda(v) = XZ \mapsto$ 

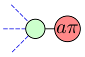
(4.111)

$\lambda(v) = Y \mapsto$ 

(4.112)

$\lambda(v) = YZ \mapsto$ 

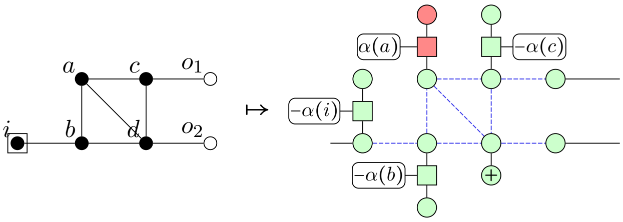
(4.113)

$\lambda(v) = Z \mapsto$ 

(4.114)

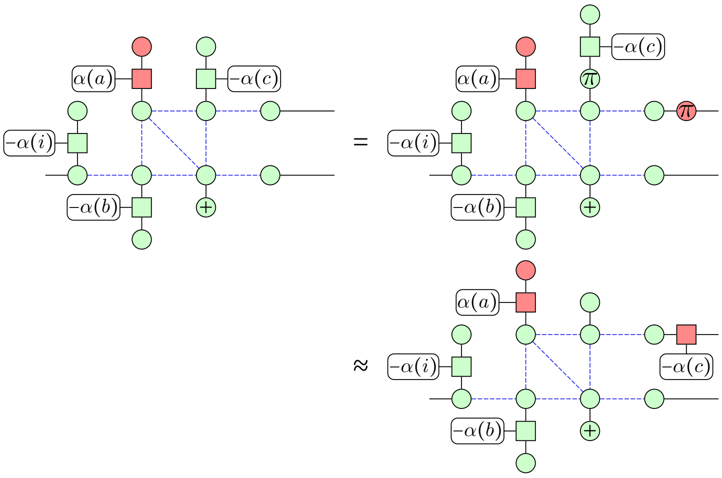
We can deterministically rewrite any ZX-diagram to a graph-like diagram [44], resembling such a measurement pattern with all measurements in the XY plane (including both X and Y Pauli measurements). Therefore, tools like flows and extraction methods for MBQC can be applied to arbitrary ZX-diagrams.

We can now piece it all together to start visualising the extraction procedure. We will use the same pattern as Example 4.3.24 for consistency, fixing $\alpha(d) = \pi$.



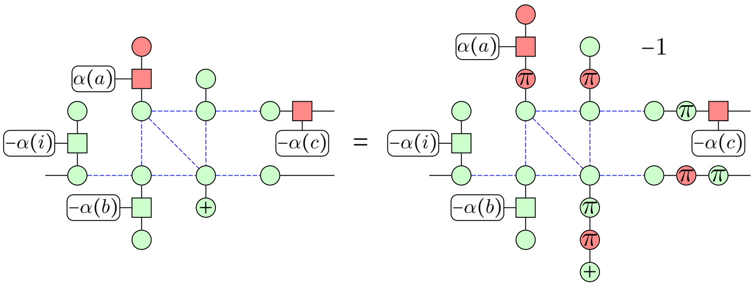
(4.115)

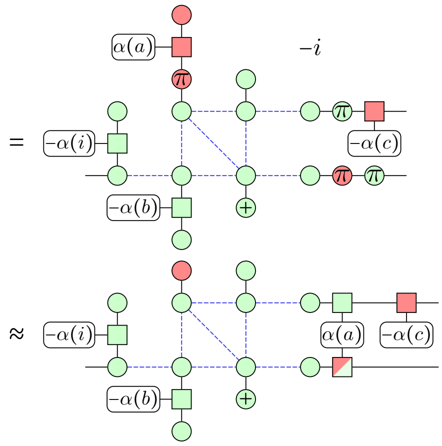
The Pauli flow told us that the rotation from vertex c should be extracted first. The stabilizer generated by o_1 introduces an X on that output and a Z at c, with which we can apply the Product Rotation Lemma. This adds a Z leg to the gadget at c, cancelling the existing one, and a new X leg at o_1, effectively moving the Pauli gadget to the output.



(4.116)

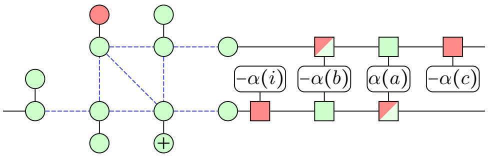
Extracting the rotation at a is a little more involved. Taking the stabilizer straight from the flow gives some non-trivial effects at c and d, so if we were to just apply the Product Rotation Lemma immediately we wouldn't be removing it from the graph completely, just moving it onto different qubits. However, we can remove these additional Paulis because the flow is focussed and we have already extracted the rotation from c.





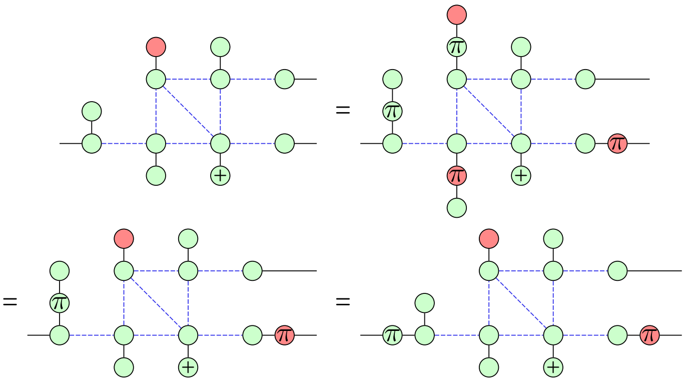
(4.117)

Continuing in this fashion, we can extract the other rotations until we have the form of a Clifford process followed by a sequence of rotations.



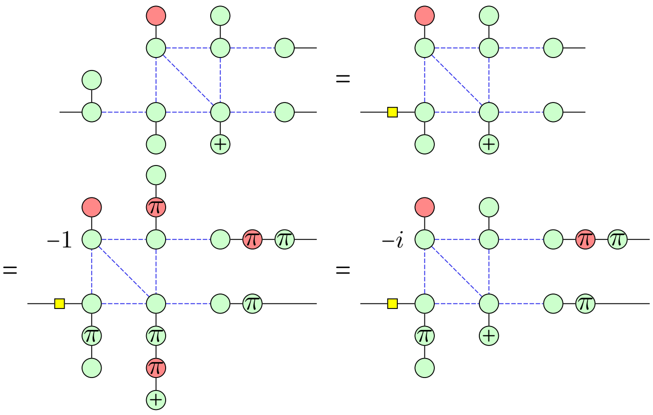
(4.118)

Finally, we focus on the Clifford process to obtain the rows of the tableau: the Z row,

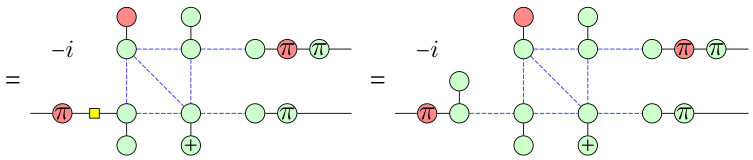


(4.119)

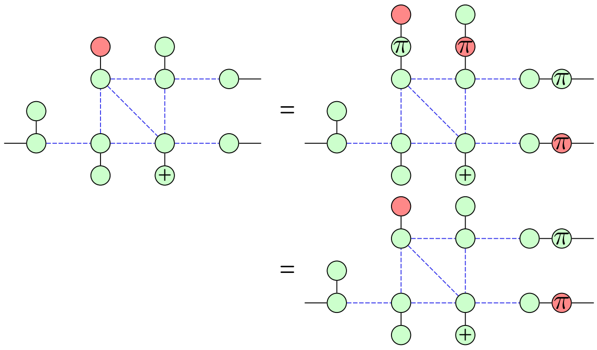
the X row,



(4.120)



and the stabilizer over the outputs.



(4.121)

4.4 Causal Structure via Parameterisation

Measurement patterns and Pauli graphs share notions of temporal dependencies between the measurements/rotations, captured by the partial orders in the flow or graph. This would, at least superficially, fall in line with viewing these as causal structures. In this final section, we will reframe measurement patterns and Pauli graphs within the framework of causal categories to more rigorously lay out this correspondence with causal structures and information signalling.

4.4.1 Signalling Orders in Flow

To model something in a causal category, we need to be able to express it as an open process, where there is a notion of interaction between the process and its environment. To describe a measurement pattern, it is clear that the interface of this open process should at least include the input and output qubits. The choices of input state and observations on the outputs are made by the programmer, who also chooses the measurement angles - let's also include those in the interface. To keep the picture simple, we will ignore the potential for measurement errors and any desire to guarantee a deterministic pure quantum channel, and allow the programmer to choose from a range of postselection operations. This reduces the "process and environment" dynamic to the constant resource state interacting with the programmer. We will still restrict the postselections available to the programmer according to the labels on the qubits by defining the following objects in Caus $[\mathrm{CP}^*]$.

Definition 4.4.1: Measurement objects

For each label $L \in \{XY, XZ, YZ, X, Y, Z\}$, we define the following object in $\mathrm{Caus}[\mathrm{CP}^*]$, where $\mathrm{CP}(-) : \mathrm{FHilb} \to \mathrm{CP}^*$ is the standard "doubling" functor:

$$\mathbf{Q}_L := \left(\mathrm{CP}\left(\mathbb{C}^2\right), \begin{cases} \{\mathrm{CP}(|+_{L,\alpha}\rangle) \mid \alpha \in \mathbb{R}\}^{**} & L \in \{XY, XZ, YZ\} \\ \{\mathrm{CP}(|+_{L,0}\rangle), \mathrm{CP}(|+_{L,\pi}\rangle)\}^{**} & L \in \{X, Y, Z\} \end{cases} \right) \tag{4.122}$$

To each labelled open graph (G, I, O, λ), we associate a local interpretation $\Gamma : V \to \mathrm{Ob}(\mathrm{Caus}[\mathcal{C}])$:

$$\Gamma(v) := \begin{cases} \mathbf{Q^{1*}} \parr \mathbf{Q^1} & v \in I \cap O \\ \mathbf{Q^{1*}} \parr \mathbf{Q}^*_{\lambda(v)} & v \in I \cap \overline{O} \\ \mathbf{Q^1} & v \in \overline{I} \cap O \\ \mathbf{Q}^*_{\lambda(v)} & v \in \overline{I} \cap \overline{O} \end{cases} \tag{4.123}$$

where $\mathbf{Q^1} := \left(\mathrm{CP}(\mathbb{C}^2), \{\overset{=}{\top}\}^*\right)$ is the first-order object containing density matrices for a single qubit.

The pure postselections are extremal points on the Bloch sphere, so the closure just gives probabilistic mixtures to fully span a plane/axis of the sphere. If we take $L = XY$ as an example, the states in $c^*_{\mathbf{Q}_{XY}}$ are those which, when measured at any angle in the XY plane, give each outcome with equal likelihood - this is precisely those that lie on the Z axis of the Bloch sphere, including mixtures. In general, $\mathbf{Q}^*_L \cong \mathbf{Q}_{L'}$ where L' is the orthogonal label to L.

It is important to note that these objects restrict postselections but not actual measurements - we can still apply any binary test $\mathbf{Q}_L \parr \mathbf{2}$ as per the Binary Test Lemma, as it only constrains the marginal effect to the appropriate subspace of the Bloch sphere. When we discuss signalling across the resource state, we can use this fact to assume the programmer has essentially complete access to the resource state for the purposes of observations.

If we consider just the local effects (i.e. choices of input state, postselections, and effect on the output), the resource state must at least be compatible with

$$\begin{aligned} \bigparr_{v \in V} \Gamma(v) &\cong \left(\bigparr_{i \in I} \mathbf{Q^{1*}} \right) \parr \left(\bigparr_{o \in O} \mathbf{Q^1} \right) \parr \left(\bigparr_{v \in \overline{O}} \mathbf{Q}^*_{\lambda(v)} \right) \\ &\cong \left(\bigotimes_{i \in I} \mathbf{Q^1} \multimap \bigotimes_{o \in O} \mathbf{Q^1} \right) \parr \left(\bigparr_{v \in \overline{O}} \mathbf{Q}^*_{\lambda(v)} \right) \end{aligned} \tag{4.124}$$

i.e. any choice of measurement angles yield some quantum channel from the inputs to the outputs. The bidirectional signalling of $\parr$ between the channel and measurements captures the influence of programmability, where the channel implemented may depend on the measurement angles chosen, and the intermediate state prior to measurements may depend on the chosen input state.

The only scope left for refining this causal type to something more specific is to look at signalling between the measured vertices. In the spirit of Equation 2.173, we would consider signalling in the context of some chosen input state and discarding the outputs - the constraint "$P \subseteq \overline{O}$ is non-signalling to $\overline{O} \setminus P$" still permits the choices of measurement angles over P to influence the outputs, just not any local measurements at $\overline{O} \setminus P$ (even measurements outside the planes/axes specified by λ). If flow really does give a causal structure, we expect this to hold for any P that is closed under the $\prec$ relation.

Let's take a simple example where λ assigns all vertices planar labels, so it is sufficient to consider gflow, and P is a single vertex v at measurement depth 1, so all corrections for the measurement at v must be over the outputs. For any choice of local postselection $\langle +_{\lambda(v),\alpha(v)}|$, we can extract the angle as a rotation over the outputs which is subsequently discarded, making the marginal state over $\overline{O} \setminus \{v\}$ the same as if $\langle +_{\lambda(v),0}|$ had been chosen. By linearity, this marginal must also be the same for any mixed effect applied at v. This is enough to show that the resource state is of the following type:

$$\left(\bigotimes_{i\in I} \mathbf{Q^1} \multimap \bigotimes_{o\in O} \mathbf{Q^1}\right) \parr \left(\left(\bigparr_{u\in\overline{O}\setminus\{v\}} \mathbf{Q}^*_{\lambda(u)}\right) < \mathbf{Q}^*_{\lambda(v)}\right) \tag{4.125}$$

Now suppose that correcting v involves some Pauli corrections on non-outputs, and let $P = \{u \mid v \prec u\} \cup \{v\}$. Trying the same trick with extracting the angle from v will give a rotation across both the outputs and some of these future measured qubits. However, by extracting the measurement angles from all qubits in P in measurement depth order, we can inductively guarantee a constant marginal. At each step, all future angles have already been extracted, so we can extract directly to the outputs using a focussed Pauli flow.

$$\left(\bigotimes_{i\in I} \mathbf{Q^1} \multimap \bigotimes_{o\in O} \mathbf{Q^1}\right) \parr \left(\left(\bigparr_{u\in\overline{O}\setminus P} \mathbf{Q}^*_{\lambda(u)}\right) < \left(\bigparr_{u\in P} \mathbf{Q}^*_{\lambda(u)}\right)\right) \tag{4.126}$$

When Pauli measurement labels are involved, the process is a little more complicated since now two vertices may be mutually involved in each other's corrections.

The extraction algorithm supposes we have some fixed postselections on all qubits with Pauli labels, so we instead only find that there is no signalling from $P \subseteq \overline{O}$ to $\overline{O} \setminus P$ when restricted to the subsets with planar measurements. A Pauli flow with order $\prec$ can be focussed without changing the ordering, enabling us to follow the extraction algorithm exactly (i.e. using the focussed property to eliminate the corrections on measured qubits to move the rotation to the outputs, where it can be discarded straightforwardly).

$$\left(\bigotimes_{i\in I}\mathbf{Q^1} \multimap \bigotimes_{o\in O}\mathbf{Q^1}\right) \parr \left(\bigparr_{\substack{u\in\overline{O}\\ \lambda(u)\in\{X,Y,Z\}}} \mathbf{Q}^*_{\lambda(u)}\right) \parr \left(\left(\bigparr_{\substack{u\in\overline{O}\setminus P\\ \lambda(u)\notin\{X,Y,Z\}}} \mathbf{Q}^*_{\lambda(u)}\right) < \left(\bigparr_{\substack{u\in P\\ \lambda(u)\notin\{X,Y,Z\}}} \mathbf{Q}^*_{\lambda(u)}\right)\right) \tag{4.127}$$

We could even still extract the angles from the Pauli vertices in P as if they were planar to get the slightly more specific type below.

$$\left(\bigotimes_{i\in I}\mathbf{Q^1} \multimap \bigotimes_{o\in O}\mathbf{Q^1}\right) \parr \left(\bigparr_{\substack{u\in\overline{O}\setminus P\\ \lambda(u)\in\{X,Y,Z\}}} \mathbf{Q}^*_{\lambda(u)}\right) \parr \left(\left(\bigparr_{\substack{u\in\overline{O}\setminus P\\ \lambda(u)\notin\{X,Y,Z\}}} \mathbf{Q}^*_{\lambda(u)}\right) < \left(\bigparr_{u\in P} \mathbf{Q}^*_{\lambda(u)}\right)\right) \tag{4.128}$$

For any choice of P that is closed under $\prec$, $\overline{O} \setminus P$ is down-closed in the sense of Definition 2.8.9, and all down-closed sets can be formed in this way. So by taking the intersections of 4.127 for every P, we obtain a graph type matching $\prec$ over the planar vertices.

Theorem 4.4.2: Flow Causality Theorem

Let (G, I, O, λ) be a labelled open graph with a Pauli flow $(p, \prec)$. Then the linear map

$$\mathrm{CP}\left(\left(\prod_{u\sim v} CZ_{u,v}\right)\left(\prod_{u\in\overline{I}} |+\rangle_u\right)\right) \tag{4.129}$$

is a causal morphism of type

$$\bigotimes_{i\in I}\mathbf{Q^1} \to \left(\bigotimes_{o\in O}\mathbf{Q^1}\right) \parr \left(\bigparr_{\substack{u\in\overline{O}\\ \lambda(u)\in\{X,Y,Z\}}} \mathbf{Q}^*_{\lambda(u)}\right) \parr \mathbf{Gr}_{G_\prec}^{\left\{u\mapsto\mathbf{Q}^*_{\lambda(u)} \,\middle|\, \lambda(u)\notin\{X,Y,Z\}\right\}} \tag{4.130}$$

where $G_\prec$ is the graph over $\left\{u \in \overline{O} \,\middle|\, \lambda(u) \notin \{X, Y, Z\}\right\}$ with edges given by $\prec$.

Proof. By linearity, we can consider pure input states and postselections, so we will just work in the category FHilb rather than CP* for clarity.

Wlog, assume that $(p, \prec)$ is focussed (otherwise, we apply the Focussing Lemma which retains the order $\prec$).

Fix an input $|\psi\rangle$ and Pauli postselections $\left\{\langle +_{\lambda(u), a_u \pi}|_u\right\}_{\lambda(u) \in \{X,Y,Z\}}$. Then consider any arbitrary down-closed set $Q \subseteq \left\{u \in \overline{O} \;\middle|\; \lambda(u) \notin \{X,Y,Z\}\right\}$, so the remaining planar vertices $P := \left\{u \in \overline{O} \;\middle|\; \lambda(u) \notin \{X,Y,Z\}\right\} \setminus Q$ are closed under $\prec$. For any choices of postselection angles α over P, look at the linear map:

$$\left(\prod_{u \in P} \langle +_{\lambda(u), \alpha(u)}|_u\right)\left(\prod_{\substack{u \in \overline{O} \\ \lambda(u) \in \{X,Y,Z\}}} \langle +_{\lambda(u), a_u \pi}|_u\right)\left(\prod_{u \sim v} CZ_{u,v}\right)\left(\prod_{u \in \overline{I}} |+\rangle_u\right)|\psi\rangle_I \quad (4.131)$$

Guided by the focussed Pauli flow, we can extract each of the measurement angles to leave the following linear map followed by some Pauli rotations over the outputs (up to global phase):

$$\left(\prod_{u \in P} \langle +_{\lambda(u), 0}|_u\right)\left(\prod_{\substack{u \in \overline{O} \\ \lambda(u) \in \{X,Y,Z\}}} \langle +_{\lambda(u), a_u \pi}|_u\right)\left(\prod_{u \sim v} CZ_{u,v}\right)\left(\prod_{u \in \overline{I}} |+\rangle_u\right)|\psi\rangle_I \quad (4.132)$$

The Pauli rotations are trace-preserving, so we can ignore them when tracing over the outputs. The above linear map is independent of the choices of angles α, so we have the non-signalling condition required for the graph type. □

As for the converse of inferring the existence of a flow from causal types, the basic type of $\bigotimes_{v \in V} \Gamma(v)$ (Equation 4.124) implies that every combination of measurement outcomes occurs with the same probability. Using results by Mhalla et al. [87], this implies the existence of gflow in the case where $|I| = |O|$ and $\forall v \in \overline{O}.\ \lambda(v) = XY$, but they also gave an example open graph with $|I| \neq |O|$ which has equal probability of all measurement outcomes but no gflow.

We have shown above how to infer non-signalling across partitions of the planar vertices of the graph, but given partial orders can be completely defined by pairs of vertices, can we narrow this down to determine from flow exactly when one vertex can signal to another? Whilst the corrections for vertex v show that it can clearly signal to the set $p(v) \cup \mathrm{Odd}(p(v))$, it may not be possible to actually detect the difference when we only have access to a subset of this, similar to the one-time pad example discussed in Section 2.8.7. We conjecture that the ability to signal between any two vertices is actually determined by the common orderings amongst all Pauli flows.

Conjecture 4.4.3

Let (G, I, O, λ) be a labelled open graph that admits a Pauli flow. Then, for any two $u, v \in \overline{O}$, the linear map

$$\mathrm{CP}\left(\left(\prod_{u \sim v} CZ_{u,v}\right)\left(\prod_{u \in \overline{I}} |+\rangle_u\right)\right) \tag{4.133}$$

is a causal morphism of type

$$\bigotimes_{i \in I} \mathbf{Q}^{\mathbf{1}} \to \left(\bigotimes_{o \in O} \mathbf{Q}^{\mathbf{1}}\right) \parr \left(\bigparr_{w \in \overline{O} \setminus \{u,v\}} \mathbf{Q}^{*}_{\lambda(w)}\right) \parr \left(\mathbf{Q}^{*}_{\lambda(v)} < \mathbf{Q}^{*}_{\lambda(u)}\right) \tag{4.134}$$

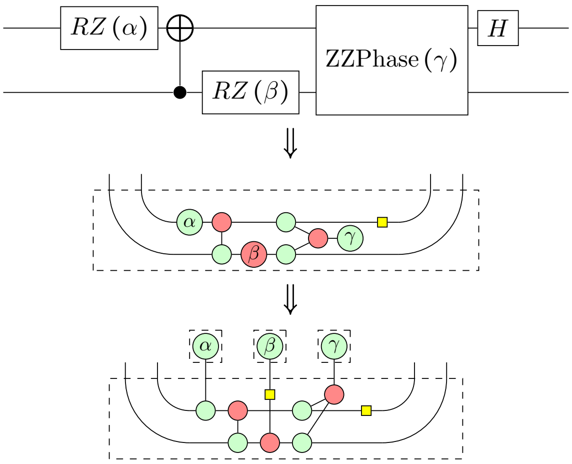
(i.e. there is no signalling from u to v) iff there exists some Pauli flow $(p, \prec)$ such that $v \preceq u$ and if $\lambda(v) \in \{X, Y, Z\}$ then $\forall w \succ u.\ v \notin p(w) \cup \mathrm{Odd}(p(w))$.

The "if" direction of this conjecture is straightforward to prove, as such a Pauli flow can be focussed without removing this property and then we can extract the any angle from u to the outputs without using v. The "only if" direction remains for future investigation. If this conjecture were to hold, this would be especially important for the unitary ($|I| = |O|$) case where there is a unique focussed Pauli flow, and hence a unique partial order $\prec$ that precisely describes the signalling between any pair of vertices.

4.4.2 Causal Structure in Parameterised Quantum Circuits

We can apply an analogous construction to view parameterised quantum circuits as a multi-party protocol with some potential signalling between the parties who decide the parameter values. For this discussion we will consider *linear parameters* - rotations may be parameterised by a sum of real-valued parameters with some constant offset, and each parameter appears in the circuit once.

If we start with a representation of the quantum circuit in the ZX-calculus, each parameterised rotation gate is a spider with the parameter as part of the phase. We can unfuse these spiders to pull each parameter out as a unique one-legged spider matching one of the planar projector cases of Equations 4.109, 4.111, and 4.113. If we cut these off of the diagram, we obtain a presentation of the parameterised circuit as a linear map onto which we can apply postselections to initialise the parameters.



(4.135)

We can then immediately specify a local interpretation in terms of objects of $\mathrm{Ob}\left(\mathrm{Caus}\left[\mathrm{CP}^{*}\right]\right)$ in the same way as Equation 4.124.

The possible circuit implementations of this parameterised linear map give information about the causal structure, as parameter α cannot signal to parameter β if there exists some circuit implementation in which the gate parameterised by α appears after the one for β. All gates are trace-preserving, so in the context where we discard the outputs of the circuit we can remove gates, including the α rotation, to leave a marginal process at β that is independent of α.

The Pauli graph representation of the parameterised circuit would allow us to abstract away commutativity between some parameterised gates, indicating that there is no signalling in either direction between them. Given there are semantics-preserving rewrites that can alter the dependency relation in the Pauli graph, it cannot capture the entire causal structure of the parameterised linear map; though if Conjecture 4.4.3 holds, the correspondence between the focussed Pauli flows of a measurement pattern and the Pauli graph of the extracted circuit would allow us to obtain an exact characterisation of signalling between parameters as the necessary orderings across every Pauli graph for the given circuit.

Such a result could be significant for the compilation of parameterised quantum circuits. Any causal structure between the parameters implies some necessary orderings between their corresponding parameterised gates in any implementation of the same linear map, which reduces the search space of possible circuit implementations.

Additionally, knowledge of the causal structure can help reduce the complexity of parameter optimisation in quantum machine learning applications [13]. Relating causal structure to structures like Pauli flow or Pauli graphs that are easy to construct in advance may be fruitful in highlighting more ways to speed up the training procedure for a given parameterised circuit.

Chapter 5

Outlook

This thesis contributes to the literature of causal structures in higher-order processes across abstract process theories, formal logic, and practical quantum computing, all using the common framework of causal categories. Between the standard examples of $\text{Caus}\left[\text{CP}^*\right]$ for finite-dimensional quantum theory, $\text{Caus}\left[\text{Mat}[\mathbb{R}^+]\right]$ for finite-outcome classical probability theory, and $\text{Caus}\left[\text{Mat}[\mathbb{R}]\right]$ for its pseudo-probabilistic equivalent, they share identical constructions for a hierarchy of types characterising non-signalling constraints, and in every case the processes compatible with a given causal structure by signal-consistency (i.e. obeying the appropriate restrictions on information signalling between parties) are exactly the same as those that are causally realisable (determined by factorising into the shape of a graph, which we can restrict further to passing a single classical bit between each local process) up to affine combination.

The example categories additionally admit a common logic that exactly captures composition of higher-order causal structures. The proof of this relied entirely on the definite causal structures, showing that the existence of indefinite causal structures does not introduce any new rules for composing black boxes. Similarly, the logic is entirely determined by the fragment inductively generated by first-order systems and transformations, meaning there is no new compositional behaviour satisfied by the extension to arbitrary affine-closed spaces. The relation to pomset logic situates this framework amongst the landscape of logics for mixing spacial and temporal connectives, showing it has more structure than BV, GV, and MAV and giving physical intuition for all theorems of these logics.

Specific to quantum computing, we gave a new construction relating MBQC to parameterised quantum circuits through which it is clear that causal structure in the form of ordering restrictions between the measurements or parameterised gates is preserved faithfully. These relate to traditional causal structure by coinciding with non-signalling constraints between the parameters when they are viewed as local systems in a multi-party protocol deciding the overall channel. The extraction algorithm itself is beneficial for automated circuit compilation where it is common to convert the circuit to different representations to identify new optimisations, such as the ZX-calculus which is closely related to MBQC [30, 45]. To this end, it improves on the previous algorithm [7] by weakening the requirements on the input pattern/ZX-diagram, and produces circuits in a form compatible with other existing circuit optimisation procedures [120, 36, 37, 115]. The connection to causal structure also has potential to enable future circuit optimisations or algorithms guided by causal inference where knowing the causal structure of the target linear map can reduce the size of the search space for a circuit implementation.

As a research project, the majority of these results were straightforward by taking the obvious generalisations of existing definitions and results known for first-order quantum processes or by generating and studying example processes (where the code in Appendix B was of great use), with a few exceptions. The Seq Equivalence Theorem required imposing additional assumptions on the base category (see Remark 2.6.16) without which it may have only been possible to show a de Morgan duality between one-way signalling and semi-localisability - still an interesting result, but not quite as satisfying as equivalence. The proof of the Graph Equivalence Theorem was also challenging to find the right decomposition of an arbitrary process into a network of local tests. Even the definitions of preferred bases and graph types were proposed late in the project to simplify earlier proofs of the Non-signalling Theorem and Causal Characterisation Theorem. The development of Section 2.9 was also surprising, where I expected more of the results of effectus theory to carry over, but the failures gave insight into why the concept of partiality in theories of higher-order processes is not as trivial as in the first-order case.

Each avenue of research within this thesis could be pushed further by future projects, with the following suggestions:

- In Section 2.8.1, we remarked that there was some aesthetic similarity between graph states for local graph types and Bayesian networks, but this is not an exact match. True Bayesian networks broadcast the observable outcome to the dependents. In order to define another alternative graph type definition that actually matches with this, there are two big hurdles to overcome. The first is the lack of a broadcasting map in CP^*. One can define a linear map that performs the right action but it will fail to be completely positive; that just means that our definition would have to write the network expression in $\mathrm{Sub}\left(\mathcal{C}\right)$ and just assert that the overall composition exists in $\mathcal{C}$. The second issue is whether broadcasting would work with higher-order structure - when effects are no longer unique, would we have to ensure that any causal effect works as a counit for the broadcasting map (dually, restricting the ability to only broadcast the causal states), or is the uniform effect sufficient knowing that this constrains the local processes of the network to not only have a constant marginal but *this* specific marginal? Would either of these give the right higher-order generalisation or are Bayesian networks too inherently tied to a first-order picture?

- The investigation into options for building a theory of partial maps in Section 2.9 eventually leaned towards combining the input and output spaces into a single object and looking at the binary tests on it, possibly modulo an equivalence relation. Relating this back to the inspiration of effectus theory, this unified partial maps with predicates. Knowing that predicates in an effectus always form an effect algebra, we could ask whether the same is true here and many more similar questions about what structure and results move over, even as special cases for partial first-order causal processes. Some definitions would clearly need adapting to work, like image objects requiring us to explicitly take affine closures again.

- A full characterisation of causal consistency would extend causal logic to richer grammars to incorporate all operators present in the $\mathrm{Caus}\left[-\right]$ construction. Our brief discussion of this in Section 3.5 showed that the additives, in particular, would likely be a challenging task to integrate into the logic as we would need to break the existing time-symmetry to capture the necessary distributive laws and preservation of first-order systems under coproduct. There exist multiple methods of extending proof-nets for MLL to MALL [55, 70] that differ on whether they equate sequent proofs that differ by reordering rules for $\otimes$ and $\times$. Knowing that our additives are categorical (co)products and we don't particularly care for the size of proofs, we will want to equate such proofs; it remains to attempt to integrate the corresponding techniques of [70] into causal proof-nets.

- The assumptions of an additive precausal category can be quite restrictive in practice, eliminating several example categories of key interest such as Set, Rel, Hilb, unitaries, and real quantum theory. Relaxing the assumptions on the base category can help us to see more easily when the key results of this thesis can be generalised to other contexts. To this end, we advocate for adopting and extending the profunctorial framework of Hefford and Wilson [62] which is fully generic and can apply to any symmetric monoidal category. Their initial results connecting logical content of the strong endoprofunctor category to decomposition theorems are worth pushing further to give logical accounts of physical structure and vice versa - especially if it can give a better physical motivation for the distinction between theories modelling BV and those modelling pomset/causal logic. A starting point could be to generalise their framework to suitably-enriched categories to reuse some of the linear algebraic proofs of this thesis.

- To help further the study of categorical models of causation and relating them to logic, it would help to have a true characterisation of pomset/causal logic through some kind of free category construction. A good candidate could be a weakening of a dependence category [104] to a pseudo-algebra of an operad that just considers series-parallel graphs, on top of monoidal closure and a duality.
- Since the failure of causal consistency for some scenario amounts to the existence of a cycle of information flow (such as in Example 3.3.12), one can analogise this as checking for deadlock-freedom in a concurrent programming model. This raises some curiosity as to whether there is any correspondence between deadlock-free π-calculus programs and pomset or a related logic that can be proved using similar techniques. BV, GV, and MAV are already studied as logics for session types [27, 3]. By imagining ideal send and receive operations as physical processes between the channels and the local threads, we could argue about the information signalling present to obtain candidate causal types. Connecting these in a fixed manner (to build a diagram we can check for causal consistency) would impose a deterministic matching of the sends and receives across the entire execution of the protocol as done with session types [65, 66]. More work is required to formalise this and verify the conjecture of capturing deadlock-freedom.
- Conjecture 4.4.3 follows the evidence for strong links between signalling-based causal structure in parameterised linear maps and temporal dependencies in their circuit implementations or flows in MBQC. Beyond just answering the conjecture, there is space for more work here to see if causal discovery algorithms (existing ones or new ones based on characterisations from flow or Pauli

graphs) can be applied to improve the time complexity of parameterised circuit synthesis. Though in the case where the parameterisations capture all non-Clifford behaviour, it is likely that more efficient causal discovery is possible by directly studying the tableau of the remaining Clifford process.

Bibliography

[1] S. Aaronson and D. Gottesman. *Improved Simulation of Stabilizer Circuits.* Physical Review A, 70(5):052328, jun 2004. `doi:10.1103/PhysRevA.70.052328`.

[2] S. Abramsky and R. Jagadeesan. *Games and full completeness for multiplicative linear logic.* Journal of Symbolic Logic, 59(2):543–574, jun 1994. `doi:10.2307/2275407`.

[3] M. Acclavio, R. Horne, S. Mauw, and L. Straßburger. *A Graphical Proof Theory of Logical Time.* In 7th International Conference on Formal Structures for Computation and Deduction (FCSD 2022), Leibniz International Proceedings in Informatics (LIPIcs), Haifa, Israel, 2022. `doi:10.4230/LIPIcs.FSCD.2022.22`.

[4] M. Aguiar and S. Mahajan. *Monoidal Functors, Species and Hopf Algebras*, volume 29 of *CRM Monograph Series.* American Mathematical Society, Providence, Rhode Island, oct 2010. `doi:10.1090/crmm/029`.

[5] M. Amy. *Formal Methods in Quantum Circuit Design.* PhD thesis, University of Waterloo, 2019. URL: `http://hdl.handle.net/10012/14480`.

[6] R. Atkey and W. Kokke. *A Semantic Proof of Generalised Cut Elimination for Deep Inference.* In 40th Conference on Mathematical Foundations of Programming Semantics (MFPS XL), Electronic Notes in Theoretical Informatics and Computer Science `doi:10.46298/entics.14870`.

[7] M. Backens, H. Miller-Bakewell, G. de Felice, L. Lobski, and J. van de Wetering. *There and back again: A circuit extraction tale.* Quantum, 5:421, mar 2021. `doi:10.22331/q-2021-03-25-421`.

[8] H. Barendregt, M. Coppo, and M. Dezani-Ciancaglini. *A filter lambda model and the completeness of type assignment.* Journal of Symbolic Logic, 48(4):931–940, 1983. `doi:10.2307/2273659`.

[9] P. K. Barkoutsos, J. F. Gonthier, I. Sokolov, N. Moll, G. Salis, A. Fuhrer, M. Ganzhorn, D. J. Egger, M. Troyer, A. Mezzacapo, S. Filipp, and I. Tavernelli. *Quantum algorithms for electronic structure calculations: Particle-hole Hamiltonian and optimized wave-function expansions.* Physical Review A, 98(2):022322, 2018. `doi:10.1103/PhysRevA.98.022322`.

[10] J. Barrett. *Information processing in generalized probabilistic theories.* Physical Review A - Atomic, Molecular, and Optical Physics, 75(3):032304, mar 2007. `doi:10.1103/PhysRevA.75.032304`.

[11] Ä. Baumeler and E. Tselentis. *Equivalence of Grandfather and Information Antinomy Under Intervention.* In 17th International Conference on Quantum Physics and Logic (QPL 2020), Electronic Proceedings in Theoretical Computer Science, 340:1-12, sep 2021. `doi:10.4204/EPTCS.340.1`.

[12] D. Beckman, D. Gottesman, M. A. Nielsen, and J. Preskill. *Causal and localizable quantum operations.* Physical Review A - Atomic, Molecular, and Optical Physics, 64(5):21, oct 2001. `doi:10.1103/PhysRevA.64.052309`.

[13] M. Benedetti, M. Fiorentini, and M. Lubasch. *Hardware-efficient variational quantum algorithms for time evolution.* Physical Review Research, 3(3):33083, sep 2021. `doi:10.1103/PhysRevResearch.3.033083`.

[14] C. H. Bennett, G. Brassard, C. Crépeau, R. Jozsa, A. Peres, and W. K. Wootters. *Teleporting an unknown quantum state via dual classical and Einstein-Podolsky-Rosen channels.* Physical Review Letters, 70(13):1895–1899, 1993. `doi:10.1103/PhysRevLett.70.1895`.

[15] A. Bisio and P. Perinotti. *Theoretical framework for higher-order quantum theory.* Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences, 475(2225), may 2019. `doi:10.1098/rspa.2018.0706`.

[16] R. Blute. *Linear logic, coherence and dinaturality.* Theoretical Computer Science, 115(1):3–41, 1993. `doi:10.1016/0304-3975(93)90053-V`.

[17] R. Blute, P. Panangaden, and S. Slavnov. *Deep inference and probabilistic coherence spaces.* Applied Categorical Structures, 20:209–228, 2012. `doi:10.1007/s10485-010-9241-0`.

[18] S. Bravyi, J. M. Gambetta, A. Mezzacapo, and K. Temme. *Tapering off qubits to simulate fermionic Hamiltonians.* jan 2017. `arXiv:1701.08213`.

[19] D. E. Browne, E. Kashefi, M. Mhalla, and S. Perdrix. *Generalized flow and determinism in measurement-based quantum computation.* New Journal of Physics, 9(8):250–250, aug 2007. `doi:10.1088/1367-2630/9/8/250`.

[20] K. Brünnler and A. F. Tiu. *A local system for classical logic.* In Logic for Programming Artificial Intelligence and Reasoning (LPAR 2001), Lecture Notes in Computer Science, 2250:347–361, 2001. `doi:10.1007/3-540-45653-8_24`.

[21] P. J. Cavalcanti, J. H. Selby, J. Sikora, and A. B. Sainz. *Simulating all multipartite non-signalling channels via quasiprobabilistic mixtures of local channels in generalised probabilistic theories.* Journal of Physics A: Mathematical and Theoretical, 55(40):404001, apr 2022. `doi:10.1088/1751-8121/ac8ea4`.

[22] G. Chiribella, G. M. D'Ariano, and P. Perinotti. *Theoretical framework for quantum networks.* Physical Review A, 80(2):022339, aug 2009. `doi:10.1103/PhysRevA.80.022339`.

[23] G. Chiribella, G. M. D'Ariano, and P. Perinotti. *Probabilistic theories with purification.* Physical Review A, 81(6):062348, jun 2010. `doi:10.1103/PhysRevA.81.062348`.

[24] G. Chiribella, G. M. D'Ariano, P. Perinotti, and B. Valiron. *Quantum computations without definite causal structure.* Physical Review A, 88(2):022318, aug 2013. `doi:10.1103/PhysRevA.88.022318`.

[25] K. Cho, B. Jacobs, B. Westerbaan, and A. Westerbaan. *An Introduction to Effectus Theory.* dec 2015. `arXiv:1512.05813`.

[26] M. D. Choi. *Completely positive linear maps on complex matrices.* Linear Algebra and Its Applications, 10(3):285–290, 1975. `doi:10.1016/0024-3795(75)90075-0`.

[27] G. Ciobanu and R. Horne. *Behavioural analysis of sessions using the calculus of structures.* In Perspectives of System Informatics (PSI 2015), Lecture Notes in Computer Science, 9609:91–106, 2016. `doi:10.1007/978-3-319-41579-6_8`.

[28] J. R. Cockett and R. A. Seely. *Proof theory for full intuitionistic linear logic, bilinear logic, and MIX categories.* Theory and Applications of Categories, 3(5):85–131, 1997.

[29] B. Coecke. *Terminality implies non-signalling.* In 11th International Workshop on Quantum Physics and Logic (QPL 2014), Electronic Proceedings in Theoretical Computer Science, EPTCS, 172:27–35, 2014. `doi:10.4204/EPTCS.172.3`.

[30] B. Coecke and R. Duncan. *Interacting quantum observables.* In Automata, Languages and Programming (ICALP 2008), Lecture Notes in Computer Science, 5126:298–310, 2008. `doi:10.1007/978-3-540-70583-3_25`.

[31] B. Coecke and A. Kissinger. *Picturing Quantum Processes.* Cambridge University Press, mar 2017. `doi:10.1017/9781316219317`.

[32] B. Coecke and R. Lal. *Causal Categories: Relativistically Interacting Processes.* Foundations of Physics, 43(4):458–501, apr 2013. `doi:10.1007/s10701-012-9646-8`.

[33] O. Cole. *Quantum Circuit Optimisation Through Stabiliser Reduction of Pauli Exponentials.* MSc thesis, University of Oxford, 2022. URL: `http://www.cs.ox.ac.uk/people/aleks.kissinger/theses/cole-thesis.pdf`.

[34] C. Comfort, A. Delpeuch, and J. Hedges. *Sheet diagrams for bimonoidal categories.* oct 2020. `arXiv:2010.13361`.

[35] M. Coppo and M. Dezani-Ciancaglini. *An extension of the basic functionality theory for the λ-calculus.* Notre Dame Journal of Formal Logic, 21(4):685–693, 1980. `doi:10.1305/ndjfl/1093883253`.

[36] A. Cowtan, S. Dilkes, R. Duncan, W. Simmons, and S. Sivarajah. *Phase gadget synthesis for shallow circuits.* In 16th International Conference on Quantum Physics and Logic (QPL 2019), Electronic Proceedings in Theoretical Computer Science, EPTCS, 2020. `doi:10.4204/EPTCS.318.13`.

[37] A. Cowtan, W. Simmons, and R. Duncan. *A Generic Compilation Strategy for the Unitary Coupled Cluster Ansatz.* jul 2020. `arXiv:2007.10515`.

[38] V. Danos and E. Kashefi. *Determinism in the one-way model.* Physical Review A, 74(5), 2006. `doi:10.1103/PhysRevA.74.052310`.

[39] V. Danos, E. Kashefi, and P. Panangaden. *The measurement calculus.* Journal of the ACM, 54(2):8, apr 2007. `doi:10.1145/1219092.1219096`.

[40] V. Danos and L. Regnier. *The structure of multiplicatives.* Archive for Mathematical Logic, 28(3):181–203, oct 1989. `doi:10.1007/BF01622878`.

[41] N. De Beaudrap. *Finding flows in the one-way measurement model.* Physical Review A - Atomic, Molecular, and Optical Physics, 77(2):022328, feb 2008. `doi:10.1103/PhysRevA.77.022328`.

[42] N. De Beaudrap. *Unitary-circuit semantics for measurement-based computations.* International Journal of Quantum Information, 08(01n02):1–91, 2010. `doi:10.1142/S0219749910006113`.

[43] E. Dubuc and R. Street. *Dinatural transformations.* Reports of the Midwest Category Seminar IV, Lecture Notes in Mathematics, 137:126–137, 1970. `doi:10.1007/bfb0060443`.

[44] R. Duncan, A. Kissinger, S. Perdrix, and J. Van De Wetering. *Graph-theoretic Simplification of Quantum Circuits with the ZX-calculus.* Quantum, 4:279, 2020. `doi:10.22331/q-2020-06-04-279`.

[45] R. Duncan and S. Perdrix. *Rewriting Measurement-Based Quantum Computations with Generalised Flow.* In Automata, Languages, and Programming (ICALP 2010), Lecture Notes in Computer Science, 6199:285–296. 2010. `doi:10.1007/978-3-642-14162-1\_24`.

[46] M. Earnshaw, J. Hefford, and M. Román. *The Produoidal Algebra of Process Decomposition.* In 32nd EACSL Annual Conference on Computer Science Logic (CSL 2024), Leibniz International Proceedings in Informatics (LIPIcs), 288:25:1–25:19, 2024. `doi:10.4230/LIPIcs.CSL.2024.25`.

[47] D. Ebler, S. Salek, and G. Chiribella. *Enhanced Communication with the Assistance of Indefinite Causal Order.* Physical Review Letters, 120(12), mar 2018. `doi:10.1103/PhysRevLett.120.120502`.

[48] A. Eddins, M. Motta, T. P. Gujarati, S. Bravyi, A. Mezzacapo, C. Hadfield, and S. Sheldon. *Doubling the Size of Quantum Simulators by Entanglement Forging.* PRX Quantum, 3(1):10309, mar 2022. `doi:10.1103/PRXQuantum.3.010309`.

[49] T. Eggeling, D. Schlingemann, and R. F. Werner. *Semicausal operations are semilocalizable*. EPL (Europhysics Letters), 57(6):782, mar 2002. doi:10.1209/EPL/I2002-00579-4.

[50] S. Eilenberg and G. M. Kelly. *A generalization of the functorial calculus*. Journal of Algebra, 3(3):366–375, 1966. doi:10.1016/0021-8693(66)90006-8.

[51] A. Fleury and C. Retoré. *The Mix Rule*. Mathematical Structures in Computer Science, 4(2):273–285, 1994. doi:10.1017/S0960129500000451.

[52] B. Fong and D. I. Spivak. *An Invitation to Applied Category Theory: Seven Sketches in Compositionality*. Cambridge University Press, 2019. doi:10.1017/9781108668804.

[53] T. Fritz. *A synthetic approach to Markov kernels, conditional independence and theorems on sufficient statistics*. Advances in Mathematics, 370:107239, 2020. doi:10.1016/j.aim.2020.107239.

[54] T. Fritz and A. Lorenzin. *Involutive Markov categories and the quantum de Finetti theorem*. 2023. arXiv:2312.09666.

[55] J.-Y. Girard. *Linear logic*. Theoretical Computer Science, 50(1):1–101, jan 1987. doi:10.1016/0304-3975(87)90045-4.

[56] J.-Y. Girard. *Between Logic and Quantic: a Tract*. In Linear Logic in Computer Science, pages 346–381. 2010. doi:10.1017/cbo9780511550850.011.

[57] D. Gosset, V. Kliuchnikov, M. Mosca, and V. Russo. *An algorithm for the T-count*. Quantum Information and Computation, 14(15&16):1261–1276, nov 2014. doi:10.26421/QIC14.15-16-1.

[58] A. Guglielmi. *A system of interaction and structure*. ACM Transactions on Computational Logic, 2007. doi:10.1145/1182613.1182614.

[59] A. Guglielmi, T. Gundersen, and M. Parigot. *A proof calculus which reduces syntactic bureaucracy*. In Proceedings of the 21st International Conference on Rewriting Techniques and Applications, Leibniz International Proceedings in Informatics (LIPIcs), volume 6, pages 135–150, 2010. doi:10.4230/LIPIcs.RTA.2010.135.

[60] G. Gutoski. *Properties of Local Quantum Operations with Shared Entanglement.* Quantum Information and Computation, 9(9-10):0739–0764, may 2008. arXiv:0805.2209.

[61] J. Hefford and C. Comfort. *Coend Optics for Quantum Combs.* In Fifth International Conference on Applied Category Theory (ACT 2022), Electronic Proceedings in Theoretical Computer Science, EPTCS, 380:63–76, aug 2023. doi:10.4204/EPTCS.380.4.

[62] J. Hefford and M. Wilson. *A Profunctorial Semantics for Quantum Supermaps.* In Proceedings of the 39th Annual ACM/IEEE Symposium on Logic in Computer Science, pages 1–15, New York, NY, USA, jul 2024. ACM. doi:10.1145/3661814.3662123.

[63] C. Heunen and J. Vicary. *Categories for Quantum Theory.* Oxford Graduate Texts in Mathematics. Oxford University Press, Oxford, 2019. doi:10.1093/oso/9780198739623.001.0001.

[64] T. Hoffreumon and O. Oreshkov. *Projective characterization of higher-order quantum transformations.* 2022. arXiv:2206.06206.

[65] K. Honda. *Types for dyadic interaction.* In CONCUR 1993, Lecture Notes in Computer Science, 715:509–523, 1993. doi:10.1007/3-540-57208-2_35.

[66] K. Honda, N. Yoshida, and M. Carbone. *Multiparty asynchronous session types.* In Conference Record of the Annual ACM Symposium on Principles of Programming Languages, 273–284, 2008. doi:10.1145/1328438.1328472.

[67] R. Horne. *The consistency and complexity of multiplicative additive system virtual.* Scientific Annals of Computer Science, 25(2):245–316, 2015. doi:10.7561/SACS.2015.2.245.

[68] R. Horne. *The sub-additives: A proof theory for probabilistic choice extending linear logic.* In 4th International Conference on Formal Structures for Computation and Deduction (FSCD 2019), Leibniz International Proceedings in Informatics (LIPIcs), 131, 2019. doi:10.4230/LIPIcs.FSCD.2019.23.

[69] R. Houston. *Finite products are biproducts in a compact closed category.* Journal of Pure and Applied Algebra, 212(2):394–400, feb 2008. doi:10.1016/j.jpaa.2007.05.021.

[70] D. J. Hughes and R. J. Van Glabbeek. *Proof nets for unit-free multiplicative-additive linear logic.* In ACM Transactions on Computational Logic, 6:784–842, oct 2005. `doi:10.1145/1094622.1094629`.

[71] M. Huth and M. Ryan. *Logic in Computer Science.* Cambridge University Press, aug 2004. `doi:10.1017/CBO9780511810275`.

[72] M. Hyland and A. Schalk. *Glueing and orthogonality for models of linear logic.* In Theoretical Computer Science, 294:183–231, 2003. `doi:10.1016/S0304-3975(01)00241-9`.

[73] B. Jacobs. *New directions in categorical logic, for classical, probabilistic and quantum logic.* Logical Methods in Computer Science, 11(3):1–76, 2015. `doi:10.2168/LMCS-11(3:24)2015`.

[74] A. Jamiołkowski. *Linear transformations which preserve trace and positive semidefiniteness of operators.* Reports on Mathematical Physics, 3(4):275–278, dec 1972. `doi:10.1016/0034-4877(72)90011-0`.

[75] D. Jia and N. Sakharwade. *Tensor products of process matrices with indefinite causal structure.* Physical Review A, 97(3):032110, mar 2018. `doi:10.1103/PhysRevA.97.032110`.

[76] O. Kahramanoğullari. *System BV is NP-complete.* Annals of Pure and Applied Logic, 152(1-3):107–121, 2008. `doi:10.1016/j.apal.2007.11.005`.

[77] A. Kissinger, M. Hoban, and B. Coecke. *Equivalence of relativistic causal structure and process terminality.* aug 2017. `arXiv:1708.04118`.

[78] A. Kissinger and S. Uijlen. *A categorical semantics for causal structure.* Logical Methods in Computer Science, 15(3):4426, 2019. `doi:10.23638/LMCS-15(3:15)2019`.

[79] A. Kissinger and J. van de Wetering. *Reducing the number of non-Clifford gates in quantum circuits.* Physical Review A, 102(2):022406, aug 2020. `doi:10.1103/PhysRevA.102.022406`.

[80] E. H. Kronheimer and R. Penrose. *On the structure of causal spaces.* Mathematical Proceedings of the Cambridge Philosophical Society, 63(2):481–501, 1967. `doi:10.1017/S030500410004144X`.

[81] F. Lamarche and L. Strassburger. *From proof nets to the free *-autonomous category.* Logical Methods in Computer Science, 2(4):1–44, 2006. doi:10.2168/LMCS-2(4:3)2006.

[82] L. Lamport. *Time, Clocks, and the Ordering of Events in a Distributed System.* Communications of the ACM, 21(7):558–565, jul 1978. doi:10.1145/359545.359563.

[83] D. Litinski. *A Game of Surface Codes: Large-Scale Quantum Computing with Lattice Surgery.* Quantum, 3:128, mar 2019. doi:10.22331/q-2019-03-05-128.

[84] D. Maslov and M. Roetteler. *Shorter Stabilizer Circuits via Bruhat Decomposition and Quantum Circuit Transformations.* IEEE Transactions on Information Theory, 64(7):4729–4738, jul 2018. doi:10.1109/TIT.2018.2825602.

[85] P.-A. Melliès. *Categorical Semantics of Linear Logic.* Panoramas et Synthèses,, 27:15–215, 2009.

[86] P.-A. Melliès. *A Topological Correctness Criterion for Multiplicative Non-Commutative Logic.* In Linear Logic in Computer Science, 316:283–322. London Mathematical Society, 2010. doi:10.1017/cbo9780511550850.009.

[87] M. Mhalla, M. Murao, S. Perdrix, M. Someya, and P. S. Turner. *Which Graph States are Useful for Quantum Information Processing?* In Theory of Quantum Computation, Communication, and Cryptography, 174–187, 2014. doi:10.1007/978-3-642-54429-3_12.

[88] M. Mhalla and S. Perdrix. *Finding Optimal Flows Efficiently.* In Automata, Languages, and Programming (ICALP 2008), Lecture Notes in Computer Science, 5125:857–868, 2008. doi:10.1007/978-3-540-70575-8_70.

[89] M. Mhalla, S. Perdrix, and L. Sanselme. *Characterising Determinism in MBQCs involving Pauli Measurements.* 2022. arXiv:2207.09368.

[90] A. S. Murawski and C. H. Ong. *Fast verification of MLL proof nets via IMLL.* ACM Transactions on Computational Logic, 7(3):473–498, jul 2006. doi:10.1145/1149114.1149116.

[91] K. F. Ng and Q. Wang. *A universal completion of the ZX-calculus.* jun 2017. arXiv:1706.09877.

[92] L. T. D. Nguyên. *Unique perfect matchings, forbidden transitions and proof nets for linear logic with mix.* Logical Methods in Computer Science, 16(1):27:1–27:31, 2020. doi:10.23638/LMCS-16(1:27)2020.

[93] L. T. D. Nguyên and L. Straßburger. *BV and Pomset Logic Are Not the Same.* In 30th EACSL Annual Conference on Computer Science Logic (CSL 2022), Leibniz International Proceedings in Informatics (LIPIcs), 216:1–32, 2022. doi:10.4230/LIPIcs.CSL.2022.32.

[94] L. T. D. Nguyên and L. Straßburger. *A System of Interaction and Structure III: The Complexity of BV and Pomset Logic.* Logical Methods in Computer Science, 19(4):25:1–25:60, dec 2023. doi:10.46298/lmcs-19(4:25)2023.

[95] M. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information.* 2010. doi:10.1017/cbo9780511976667.

[96] O. Oreshkov, F. Costa, and Č. Brukner. *Quantum correlations with no causal order.* Nature Communications, 3(1):1–8, oct 2012. doi:10.1038/ncomms2076.

[97] A. J. Parzygnat. *Inverses, disintegrations, and Bayesian inversion in quantum Markov categories.* jan 2020. arXiv:2001.08375.

[98] S. Popescu and D. Rohrlich. *Quantum nonlocality as an axiom.* Foundations of Physics, 24(3):379–385, mar 1994. doi:10.1007/BF02058098.

[99] R. Raussendorf and H. J. Briegel. *A one-way quantum computer.* Physical Review Letters, 2001. doi:10.1103/PhysRevLett.86.5188.

[100] C. Retoré. *Réseaux et séquents ordonnés.* PhD thesis, Université Paris-Diderot - Paris VII, feb 1993. URL: https://theses.hal.science/tel-00585634.

[101] C. Retoré. *On the relation between coherence semantics and multiplicative proof nets* . Technical Report RR-2430, INRIA, dec 1994. URL: https://inria.hal.science/inria-00074245.

[102] C. Retoré. *Pomset logic: A non-commutative extension of classical linear logic.* In Typed Lambda Calculi and Applications (TLCA 1997), Lecture Notes in Computer Science, 1210:300–318, 1997. doi:10.1007/3-540-62688-3_43.

[103] C. Retoré. *Handsome proof-nets: Perfect matchings and cographs.* In Theoretical Computer Science, 294:473–488, 2003. doi:10.1016/S0304-3975(01)00175-X.

[104] B. T. Shapiro and D. I. Spivak. *Duoidal Structures for Compositional Dependence.* oct 2022. arXiv:2210.01962.

[105] W. Simmons. *Relating Measurement Patterns to Circuits via Pauli Flow.* In 18th International Conference on Quantum Physics and Logic (QPL 2021), Electronic Proceedings in Theoretical Computer Science, EPTCS, 343:50–101, sep 2021. doi:10.4204/EPTCS.343.4.

[106] W. Simmons and A. Kissinger. *Higher-Order Causal Theories Are Models of BV-Logic.* In 47th International Symposium on Mathematical Foundations of Computer Science (MFCS 2022), Leibniz International Proceedings in Informatics (LIPIcs), 241:80:1–80:14, 2022. doi:10.4230/LIPIcs.MFCS.2022.80.

[107] W. Simmons and A. Kissinger. *A complete logic for causal consistency.* mar 2024. arXiv:2403.09297.

[108] S. Slavnov. *On noncommutative extensions of linear logic.* Logical Methods in Computer Science, 15(3):25, 2019. doi:10.23638/LMCS-15(3:30)2019.

[109] L. Straßburger. *A local system for linear logic.* In Logic for Programming, Artificial Intelligence, and Reasoning (LPAR 2002), Lecture Notes in Computer Science, 2514:388–402, 2002. doi:10.1007/3-540-36078-6_26.

[110] L. Straßburger. *Linear Logic and Noncommutativity in the Calculus of Structures.* PhD thesis, Technische Universität Dresden, 2003. URL: https://www.lix.polytechnique.fr/Labo/Lutz.Strassburger/papers/dissvonlutz.pdf.

[111] W. Tang, T. Tomesh, M. Suchara, J. Larson, and M. Martonosi. *CutQC: Using small Quantum computers for large Quantum circuit evaluations.* In International Conference on Architectural Support for Programming Languages and Operating Systems - ASPLOS, 473–486, apr 2021. doi:10.1145/3445814.3446758.

[112] A. Tiu. *A system of interaction and structure II: The need for deep inference.* Logical Methods in Computer Science, 2(2):1–24, 2006. doi:10.2168/LMCS-2(2:4)2006.

[113] J. Valdes, R. E. Tarjan, and E. L. Lawler. *The recognition of series parallel digraphs.* In Proceedings of the Annual ACM Symposium on Theory of Computing, 1–12, apr 1979. doi:10.1145/800135.804393.

[114] J. van de Wetering. *ZX-calculus for the working quantum computer scientist.* dec 2020. arXiv:2012.13966.

[115] E. van den Berg and K. Temme. *Circuit optimization of Hamiltonian simulation by simultaneous diagonalization of Pauli clusters.* Quantum, 4:322, sep 2020. doi:10.22331/q-2020-09-12-322.

[116] R. Vilmart. *A Near-Minimal Axiomatisation of ZX-Calculus for Pure Qubit Quantum Mechanics.* In 2019 34th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS), 1–10, jun 2019. doi:10.1109/LICS.2019.8785765.

[117] M. Wilson and G. Chiribella. *Causality in Higher Order Process Theories.* 18th International Conference on Quantum Physics and Logic (QPL 2021), Electronic Proceedings in Theoretical Computer Science, 343:265–300, jul 2021. doi:10.4204/eptcs.343.12.

[118] M. Wilson and G. Chiribella. *Free Polycategories for Unitary Supermaps of Arbitrary Dimension.* 2022. arXiv:2207.09180.

[119] M. Wilson, G. Chiribella, and A. Kissinger. *Quantum Supermaps are Characterized by Locality.* may 2022. arXiv:2205.09844.

[120] F. Zhang and J. Chen. *Optimizing T gates in Clifford+T circuit as $\pi/4$ rotations around Paulis.* mar 2019. arXiv:1903.12456.

Appendix A

Notational Conventions

The following tables provide a key for symbols, fonts, and notational standards used in this thesis. Some of these meanings will be dependent on the context, due to conflicting conventions from different disciplines.

Notation for category theory content

$\mathcal{C}, \mathcal{D}$	Categories
$\mathrm{Mat}[\mathbb{R}^{+}]$	The category of matrices over positive reals
$\mathrm{Mat}[\mathbb{R}]$	The category of matrices over real numbers
CP	The category of completely positive (linear) maps between finite-dimensional Hilbert spaces
CP^{*}	The category of completely positive maps between finite-dimensional C^{*}-algebras
$\mathrm{Caus}[-]$	The $\mathrm{Caus}[-]$ construction on categories (Definition 2.3.9)
$\mathrm{FO}(-)$	The first-order subcategory of $\mathrm{Caus}[-]$ (Definition 2.3.15)
$\mathrm{Sub}(-)$	The subtractive closure of a category (Definition 2.4.6)
$[-]$	The embedding functor $\mathcal{C} \to \mathrm{Sub}(\mathcal{C})$ (Proposition 2.4.8)
$\mathcal{U}$	The forgetful functor $\mathrm{Caus}[\mathcal{C}] \to \mathcal{C}$ (Definition 2.3.9)
$=$	Exact equality of morphisms in $\mathcal{C}$
$\sim$	Equivalence of morphisms in $\mathrm{Sub}(\mathcal{C})$
$A, B, \ldots$	Objects of some category $\mathcal{C}$
$\mathbf{A}, \mathbf{B}, \ldots$	Objects of some $\mathrm{Caus}[\mathcal{C}]$
A^{*}	Dual object in $\mathcal{C}$ (from compact closure)
$\mathbf{A}^{*}$	Dual object in $\mathrm{Caus}[\mathcal{C}]$ (Definition 2.3.13)
$\otimes$	Monoidal product in both $\mathcal{C}$ and $\mathrm{Caus}[\mathcal{C}]$ (Definition 2.3.16)
$\parr$	De Morgan dual of $\otimes$ in $\mathrm{Caus}[\mathcal{C}]$ (Definition 2.3.18)
$<$	Seq monoidal product in $\mathrm{Caus}[\mathcal{C}]$ (Definition 2.6.17)
I	Monoidal unit of $\otimes$ in $\mathcal{C}$
$\mathbf{I}$	Monoidal unit of $\{\otimes, \parr, <\}$ in $\mathrm{Caus}[\mathcal{C}]$ (Definition 2.3.12)
$\mathbf{A^{1}}$	First-order object in $\mathrm{Caus}[\mathcal{C}]$ (Definition 2.3.14)
$\oplus$	Biproduct in $\mathcal{C}$ and coproduct in $\mathrm{Caus}[\mathcal{C}]$ (Definition 2.5.1)

$\times$	Product in Caus $[\mathcal{C}]$ (Definition 2.5.1)
0	Zero object in $\mathcal{C}$
$\mathbf{0}$	Initial object in Caus $[\mathcal{C}]$ (Proposition 2.9.5)
$\mathbf{1}$	Terminal object in Caus $[\mathcal{C}]$ (Proposition 2.9.5)
$f, g, \ldots$	Morphisms
ρ, υ	States in $\mathcal{C}(I, A)$
h, k	Bipartite states in $\mathcal{C}(I, A \otimes B)$
π	Effects in $\mathcal{C}(A, I)$
$\alpha, \beta, \delta, \lambda, \mu, \theta$	Scalars in $\mathcal{C}(I, I)$
e	Effects in Sub $(\mathcal{C})$ (A, I)
s	Scalars in Sub $(\mathcal{C})$ (I, I)
$\fatsemi$	*Left-to-right* sequential composition of morphisms (e.g. $f : A \to B$, $g : B \to C$ compose to $f \fatsemi g : A \to C$)
$\prod$	Indexed sequential product
$\cdot$	Scalar multiplication (e.g. $\lambda : I \to I$, $f : A \to B$ compose to $\lambda \cdot f : A \to B$)
$[-, =]$	Copairings (e.g. $f : A \to C$, $g : B \to C$ combine to $[f, g] : A \oplus B \to C$)
$\langle -, = \rangle$	Pairings (e.g. $f : A \to B$, $g : A \to C$ combine to $\langle f, g \rangle : A \to B \times C$)
$(-, =)$	Representative pairs in Sub $(\mathcal{C})$ (Definition 2.4.6)
ι_A	Injections (e.g. $\iota_A : A \to A \oplus B$)
p_A	Projections (e.g. $p_A : A \times B \to A$)
$+$	Summation via additive enrichment (e.g. $f, g : A \to B$ sum to $f + g : A \to B$)
$0_{A,B}$	Zero morphisms
η_A	Cup state $I \to A^* \otimes A$
ϵ_A	Cap effect $A \otimes A^* \to I$
f^*	Transpose of f wrt the compact structure (Equation 2.19)
$\bar{\bar{\top}}$	The discarding effect in $\mathcal{C}(A, I)$
$\underline{\underline{\bot}}$	The maximally mixed state ($\bar{\bar{\top}}^*$) in $\mathcal{C}(I, A)$ (Equation 2.19)
$c_{\mathbf{A}}$	The set of states of some object $\mathbf{A} \in \mathrm{Ob}\,(\mathrm{Caus}\,[\mathcal{C}])$
$:$	Type membership, $\rho : \mathbf{A}$ is equivalent to $\rho \in c_{\mathbf{A}}$
$\Rightarrow$	Natural transformation
$\theta_{\mathbf{A}}$	Scalar for the uniform effect in $c^*_{\mathbf{A}}$ (Definition 2.3.9)
$\mu_{\mathbf{A}}$	Scalar for the uniform state in $c_{\mathbf{A}}$ (Definition 2.3.9)
⫯$_{\mathbf{A}}$	The scaled uniform effect in $c^*_{\mathbf{A}}$ (Definition 2.3.9)
⫰$_{\mathbf{A}}$	The scaled uniform state in $c_{\mathbf{A}}$ (Definition 2.3.9)
W	Process matrices (Definition 2.1.14)
$i, j, \ldots$	Indices in some collection
Γ	Local interpretation (Definition 2.8.1)
Δ	Edge interpretation (Definition 2.8.2)

Notation for logic content

MLL	Multiplicative Linear Logic (Section 3.1.1)
MLL+Mix	Multiplicative Linear Logic with Mix rules (Section 3.1.1)
MALL	Multiplicative Additive Linear Logic
BV	Basic system Virtual (Section 3.1.3)
MAV	Multiplicative Additive system Virtual (Section 3.1.3)
GV	Extension of BV with graphs [3]
pomset	pomset logic (Section 3.1.3)
Var	The set of all variable names
$A, B, \dots$	Variables/atoms
$A^1, B^1, \dots$	First order (FO) atoms
$F, G, \dots$	Formulae
Γ, Δ	Multisets of formulae
$\otimes$	Multiplicative conjunction with unit 1 or I
$⅋$	Multiplicative disjunction with unit $\bot$ or I
$\times$	Additive conjunction with unit 0
$\oplus$	Additive disjunction with unit $\top$
$<$	Seq operator
$(-)^*$	Duality/Negation
$\multimap$	Linear implication
π	Proof
P	Proof-structure (Definitions 3.1.3, 3.1.9 3.3.1)
P_F	The unique cut-free proof-structure of a balanced formula (Definition 3.3.1)
F_P	The formula associated with a proof-structure (Definition 3.3.1)
$\mathcal{S}_P$	The set of switchings of a proof-structure (Definition 3.3.2)
s	Switching (Definition 3.3.2)
G_s	Switching graph (Definition 3.3.2)
Φ	Interpretation (Definition 3.2.2)
ϵ_F^Φ	Contraction morphism (Definition 3.2.3)
$\mathcal{F}_F$	Causal functor of a formula (Definition 3.2.4)
$\vdash$	Logical entailment in a given logic
$\Vdash$	Semantic entailment in a given model
$\Vdash_{\mathcal{C}}^{\Phi}$	Causal consistency under the interpretation Φ (Definition 3.2.3)
$\Vdash_{\mathcal{C}}$	Causal consistency by extranatural transformation (Definition 3.2.6)
$\mathrm{pom}\,(-)$	Reduction to the pomset fragment (Definition 3.4.1)
$\mathrm{fo}\,(-)$	Reduction to the first-order inductive fragment (Definition 3.4.5)

Notation for quantum computing content

Γ	Labelled open graph (Definition 4.1.2)
u, v, w	Vertices/qubits in a labelled open graph/measurement pattern
I	Inputs to an open graph/measurement pattern
$\overline{I}$	Non-inputs $V \setminus I$, a.k.a. prepared qubits
O	Outputs from an open graph/measurement pattern
$\overline{O}$	Non-outputs $V \setminus O$, a.k.a. measured qubits
λ	A labelling of measurements $\lambda : \overline{O} \to \{XY, XZ, YZ, X, Y, Z\}$
L	An individual label in $\{XY, XZ, YZ, X, Y, Z\}$
α	An assignment of measurement angles $\alpha : \overline{O} \to [0, 2\pi)$, or an individual measurement angle
f_v	Application of a linear map f to the qubit v alongside the identity on all other qubits
$u \sim_G v$	Adjacent vertices in a graph G
$N_G(u)$	Neighbours of u in the graph G
$\mathrm{Odd}(A)$	Odd neighbourhood of $A \subseteq V$ in G (Definition 4.1.2)
$M^{L,\alpha}$	Destructive measurement at angle α in plane/Pauli L
f	Correction function for a causal flow (Definition 4.1.5)
g	Correction function for a generalised flow (Definition 4.1.7)
p	Correction function for a Pauli flow (Definition 4.1.9)
$\prec$	Ordering of vertices in a flow
$\preceq$	$u \preceq v := \neg(v \prec u)$
$A \Delta B$	Symmetric difference of sets
$V_k^{\prec}$	Vertices at measurement depth k under $\prec$ (Definition 4.2.1)
$V_{\cup k}^{\prec}$	Cumulative vertices up to measurement depth k under $\prec$ (Definition 4.2.1)
$V_{\cup k}^{\prec, L}$	Vertices at measurement depth k with label L (Lemma 4.2.2)
Λ^L	Vertices with measurement label L (Lemma 4.2.2)
$\hat{p}$, $\hat{q}$	Focussed set (Definition 4.2.4)
P, Q	Pauli strings in $\{I, X, Y, Z\}^{\otimes n}$ (with some associated phase)
$P^{p(v) \to u}$	The Pauli correction at u from the measurement of v (Equation 4.22)
$\mathrm{ES}_O^{p;v}$	Primary extraction string (Definition 4.3.10)
$\mathrm{ES}_O^{\hat{p}}$	Stabilizer related to a focussed set (Lemma 4.3.17)

Appendix B

Exploring $\mathrm{Caus}\left[\mathbf{Mat}[\mathbb{R}]\right]$ in Python

The code samples in this section will help to provide additional examples of many of the key operators of the $\mathrm{Caus}\left[-\right]$ construction in a way that is easy to customise and generate. To simplify the representation and not have to worry about any positivity constraints, these examples work with $\mathrm{Caus}\left[\mathrm{Mat}[\mathbb{R}]\right]$.

Each object of $\mathrm{Caus}\left[\mathrm{Mat}[\mathbb{R}]\right]$ is a pair (N, c) of a natural number N and an affine-closed set c of N-dimensional real vectors. It is sufficient to store a maximal linearly-independent subset of these vectors (i.e. the causal portion of a preferred basis for the object), which we arrange as the columns of a matrix. Floating-point errors can cause issues around keeping within the affine space generated by these vectors, so it is best to use a symbolic representation such as `sympy.Matrix`.

All code here is compatible with Python 3.11.9 and `sympy` 1.12.1.

```
# PREAMBLE
from sympy import (
    Matrix,
    init_printing,
    StrPrinter,
    ones,
    Rational,
    BlockMatrix,
    Array,
    Symbol,
    tensorcontraction,
    permutedims,
    sqrt,
)
from sympy.physics.quantum import TensorProduct
from typing import Tuple

init_printing(use_unicode=True)
printer = StrPrinter()


# Pretty printing
def pprint(mat: Matrix) -> None:
    print(mat.table(printer, colsep="␣"), "\n")
```

In addition to representing an affine-closed space, we need to restrict ourselves to looking at flat sets. This is harder to achieve by construction, so we instead provide a method that can verify flatness during testing.

```
# FLATNESS VERIFICATION


# Verifies a Matrix represents an object, i.e. the state set is flat
def check_flatness(c: Matrix) -> bool:
    # Strict flatness requires at least one state and one effect
    if c.cols == 0:
        return False

    # Check that the uniform effect is in the dual state
    # i.e. all columns have the same norm
    norms = c.T @ ones(c.rows, 1)
    for i in range(c.cols):
        if norms[i] != norms[0]:
            return False

    # Check that the uniform state is generated by affine combinations of states
    # i.e. adding the uniform state to the matrix does not change its rank
    added = c.col_insert(0, ones(c.rows, 1))
    return added.rank() == c.rank()
```

To start with, let's just define some methods for generating elementary spaces. Most importantly, the matrix of the first-order object for a given dimension is just the identity - the columns enumerate an orthonormal basis. If we just take a subset of the basis vectors (along with the uniform state to ensure flatness), we can construct objects that cannot be inductively defined from first-order objects and the operators like those seen in Remark 2.7.8.

```
# ELEMENTARY SPACES


# The first-order space of a given dimension can be generated by the identity
    Matrix
def fo(dim: int) -> Matrix:
    return Matrix.eye(dim)


# Verify that a space is first-order (up to isomorphism), i.e. it has a unique
    effect
def is_fo(c: Matrix) -> bool:
    # Rank of the dual set is dim (c.rows) + 1 - c.rank(), so is 1 iff c.rank() =
        dim
    # Checking that the unique state is uniform is done by checking flatness
    return check_flatness(c) & (c.rank() == c.rows)


# Creates a flat space of a given carrier dimension and affine dimension by picking
     the uniform state and the first (n-1) basis vectors
def simple_space(rows: int, cols: int) -> Matrix:
    assert cols <= rows
    assert cols >= 1
    return Matrix(
        BlockMatrix(
            [
                [Matrix.eye(cols - 1, cols - 1), Rational(1, rows) * ones(cols - 1,
                     1)],
                [
                    Matrix.zeros(rows + 1 - cols, cols - 1),
                    Rational(1, rows) * ones(rows + 1 - cols, 1),
                ],
            ]
        )
    )
```

```
# TEST ELEMENTARY SPACES

for m in [fo(1), fo(3), simple_space(3, 1), simple_space(3, 2), simple_space(3, 3)
    ]:
    print(check_flatness(m))
    print(is_fo(m))
    pprint(m)
```

```
True
True
[1]

True
True
[1 0 0]
[0 1 0]
[0 0 1]

True
False
[1/3]
[1/3]
[1/3]

True
False
[1 1/3]
[0 1/3]
[0 1/3]

True
True
[1 0 1/3]
[0 1 1/3]
[0 0 1/3]
```

An alternative way to view an affine space is to use a single parameterised vector. This can provide ways to quickly see at a glance which coefficients remain constant (or near constant) across a space and which ones can vary significantly. This is also the form provided by `sympy` when solving linear equation systems, so we provide the following functions for mapping between the forms. Using a single vector to represent the space also makes it easier to reshape into tensors to handle tensor product spaces.

```
# CONVERSIONS FOR SYMBOLIC FORM


# Maps between a Matrix describing a linearly-independent set, and a single Array (
    tensor) using symbols to represent degrees of freedom
def mat_to_sym_tensor(c: Matrix, tensor_shape: Tuple[int]) -> Array:
    vec = c.col(0)
    for i in range(1, c.cols):
        tau = Symbol("tau_" + str(i - 1))
        vec += tau * (c.col(i) - c.col(0))
    return Array(vec, tensor_shape)


def sym_tensor_to_mat(t: Array) -> Matrix:
    # Pick out one solution, then vary each parameter to give independent solutions
         that span the entire space
    t_vec = Matrix(t.reshape(len(t), 1))
    params = t_vec.free_symbols
    set_zero = {tau: 0 for tau in params}
    base_sol = t_vec.xreplace(set_zero)
    all_sols = list()
    all_sols.append(base_sol)
    for tau in params:
        tau_sol = t_vec.xreplace({tau: 1}).xreplace(set_zero)
        all_sols.append(tau_sol)

    # Reformat solution vectors into a single matrix
    return Matrix.hstack(*all_sols)
```

```
# TEST SYMBOLIC FORM

space = simple_space(5, 3)
sym_space = mat_to_sym_tensor(space, (5))
print(sym_space)
pprint(sym_tensor_to_mat(sym_space))

composite = simple_space(6, 4)
sym_space = mat_to_sym_tensor(composite, (3, 2))
print(sym_space)
pprint(sym_tensor_to_mat(sym_space))
```

```
[-tau_0 - 4*tau_1/5 + 1, tau_0 + tau_1/5, tau_1/5, tau_1/5, tau_1/5]
[1 1/5 0]
[0 1/5 1]
[0 1/5 0]
[0 1/5 0]
[0 1/5 0]

[[-tau_0 - tau_1 - 5*tau_2/6 + 1, tau_0 + tau_2/6],
    [tau_1 + tau_2/6, tau_2/6], [tau_2/6, tau_2/6]]
[1 0 0 1/6]
[0 1 0 1/6]
[0 0 1 1/6]
[0 0 0 1/6]
[0 0 0 1/6]
[0 0 0 1/6]
```

This finally brings us to our first way to generate new objects from old ones: the duality functor $(-)^*$. We can express the constraints on the inner products as a linear equation system and solve it for (a generating subset of) the dual set. Solving this may fail if the dual set is empty, but success is guaranteed when the input represents a flat set.

```
# DUALITY


# Obtain the dual object of a given Matrix
# Given state x as a column vector, M.T * x gives the inner product with each state
    in the Matrix M. x is a state of the dual object iff M.T * x = (1 ... 1).T.
    Treat this as a system of linear equations and identify the space of solutions
def dual_set(c: Matrix) -> Matrix:
    c_size = c.cols
    dim = c.rows

    # Use sympy Gauss Jordan to obtain parameterisation of dual space. The
        solutions are given as a symbolic vector, parameterising the solution space
    try:
        sol, _ = c.T.gauss_jordan_solve(ones(c_size, 1))
    except ValueError:
        return Matrix.zeros(dim, 0)

    return sym_tensor_to_mat(sol)
```

```
# TEST DUALITY

ss_5_2 = simple_space(5, 2)
pprint(ss_5_2)
pprint(dual_set(ss_5_2))
```

```
[1 1/5]
[0 1/5]
[0 1/5]
[0 1/5]
[0 1/5]

[1 1 1 1]
[4 3 3 3]
[0 1 0 0]
[0 0 0 1]
[0 0 1 0]
```

The simplest mathematical definitions are for the setwise constructions for union and intersection. Union of objects combines their preferred basis by union (Proposition 2.7.5), so we can just append the matrices side-by-side and find some maximal linearly-independent subset to maintain a simple form. We could handle intersection using the de Morgan duality, but we can also express it directly as the solution space of a linear equation system.

These methods also give rise to easy ways to compare objects for containment or equality since union will degenerate in these cases. When two objects are not equal, we can also use them to generate counterexamples to containment.

```
# SETWISE


def union(cA: Matrix, cB: Matrix) -> Matrix:
    # Assert that the dimensions of systems match
    assert cA.rows == cB.rows

    # Combine and reduce to a maximal linearly-independent subset
    combined = Matrix(BlockMatrix([[cA, cB]]))
    columns = combined.columnspace()
    return Matrix.hstack(*columns)


def intersection(cA: Matrix, cB: Matrix) -> Matrix:
    # Assert that the dimensions of systems match
    assert cA.rows == cB.rows

    # Every vector in the intersection is both an affine combination of columns of
        cA and an affine combination of columns of cB, and hence can be uniquely
        identified by the vector of coefficients in one of these sums. We
        characterise this space by finding all possible affine coefficients that
        represent identical states.
    # Phrase this as a linear problem Ax = b where x has dimension cA.cols + cB.
        cols. We have two equations to ensure the coefficients sum to one. The rest
         regenerates the affine combinations as vectors in the original space and
        asserts they are equal (their difference is zero).
    lhs = Matrix(
        BlockMatrix(
            [
                [cA, -cB],
                [ones(1, cA.cols), Matrix.zeros(1, cB.cols)],
                [Matrix.zeros(1, cA.cols), ones(1, cB.cols)],
            ]
        )
    )
    rhs = Matrix.zeros(cA.rows + 2, 1)
    rhs[cA.rows, 0] = 1
    rhs[cA.rows + 1, 0] = 1

    # Use sympy Gauss Jordan to obtain parameterisation of coefficient space. The
        solutions are given as a symbolic vector, parameterising the solution space
    try:
        sol, _ = lhs.gauss_jordan_solve(rhs)
    except ValueError:
        return Matrix.zeros(cA.rows, 0)

    sol_mat = sym_tensor_to_mat(sol)

    # Compute affine combinations to yield a subspace of cA
    return cA @ sol_mat[0 : cA.cols, :]
```

```
# COMPARISON FUNCTIONS


def equal_objects(cA: Matrix, cB: Matrix) -> bool:
    return (cA.cols == cB.cols) & (cA.cols == union(cA, cB).cols)


# Returns whether A is completely contained within B
def check_containment(cA: Matrix, cB: Matrix) -> bool:
    return cB.cols == union(cA, cB).cols
```

```
# Returns basis states of cA that are not in cB
def difference(cA: Matrix, cB: Matrix) -> Matrix:
    un = union(cB, cA)
    # The first cB.cols columns of un will match cB, so just take the rest
    return un[:, cB.cols :]
```

```
# TEST COMPARISON FUNCTIONS

assert equal_objects(fo(2), fo(2))
assert not equal_objects(simple_space(4, 2), simple_space(4, 3))
assert check_containment(simple_space(4, 2), simple_space(4, 3))
assert not check_containment(simple_space(4, 2), 0.25 * dual_set(simple_space(4, 3)
    ))
assert check_containment(Matrix([[0.5], [0.5], [0], [0]]), simple_space(4, 3))
assert not check_containment(Matrix([[0], [0], [0], [1]]), simple_space(4, 3))
pprint(difference(simple_space(4, 3), simple_space(4, 2)))
```

```
[0]
[1]
[0]
[0]
```

```
# TEST SETWISE OPERATIONS

assert equal_objects(
    union(simple_space(5, 3), Rational(1, 5) * dual_set(simple_space(5, 3))), fo(5)
)
assert equal_objects(
    intersection(simple_space(5, 3), Rational(1, 5) * dual_set(simple_space(5, 3)))
        ,
    0.2 * dual_set(fo(5)),
)
```

However, when looking in terms of preferred bases, the additives are even simpler to construct. For coproducts, we just list the bases side by side (injected into the larger space), and for products we take all pairs.

```
# ADDITIVES


def coproduct(cA: Matrix, cB: Matrix) -> Matrix:
    # Block matrix of cA, zero, zero, cB
    return Matrix.diag(cA, cB)


def product(cA: Matrix, cB: Matrix) -> Matrix:
    # Iterate through pairs of states
    # Splitting the matrix into the A and B components, we can represent each as a
        tensor product of cA/cB with a row vector of all 1s
    top = TensorProduct(cA, ones(1, cB.cols))
    bottom = TensorProduct(ones(1, cA.cols), cB)
    return Matrix(BlockMatrix([[top], [bottom]]))
```

```
# TEST ADDITIVES

pprint(coproduct(simple_space(4, 2), simple_space(5, 3)))
pprint(product(fo(2), simple_space(3, 2)))
```

```
[1 1/4 0 0   0]
[0 1/4 0 0   0]
[0 1/4 0 0   0]
[0 1/4 0 0   0]
[0   0 1 0 1/5]
[0   0 0 1 1/5]
[0   0 0 0 1/5]
[0   0 0 0 1/5]
[0   0 0 0 1/5]

[1   1 0   0]
[0   0 1   1]
[1 1/3 1 1/3]
[0 1/3 0 1/3]
[0 1/3 0 1/3]
```

Finally, we have the multiplicatives. Tensor is again very straightforward, combining the preferred bases by tensor product. We could define par and the linear implication/internal hom $\multimap$ via a single linear equation system following Proposition 2.4.26, but for simplicity we will just dualise the tensor.

```
# MULTIPLICATIVES


def tensor(cA: Matrix, cB: Matrix) -> Matrix:
    # Simply take the tensor product of the basis vectors
    return TensorProduct(cA, cB)


def parr(cA: Matrix, cB: Matrix) -> Matrix:
    # Define in terms of tensor and dual
    return dual_set(tensor(dual_set(cA), dual_set(cB)))


def lin_impl(cA: Matrix, cB: Matrix) -> Matrix:
    # Define in terms of tensor and dual
    return dual_set(tensor(cA, dual_set(cB)))
```

```
# TEST MULTIPLICATIVES

assert equal_objects(tensor(fo(2), fo(3)), fo(6))
assert equal_objects(parr(fo(2), fo(3)), fo(6))
assert equal_objects(tensor(dual_set(fo(2)), dual_set(fo(3))), dual_set(fo(6)))
assert equal_objects(parr(dual_set(fo(2)), dual_set(fo(3))), dual_set(fo(6)))

fo2d_x_fo2 = tensor(dual_set(fo(2)), fo(2))
fo2d_p_fo2 = parr(dual_set(fo(2)), fo(2))
pprint(fo2d_x_fo2)
pprint(fo2d_p_fo2)
assert check_containment(fo2d_x_fo2, fo2d_p_fo2)
assert not check_containment(fo2d_p_fo2, fo2d_x_fo2)
```

```
[1 0]
[0 1]
[1 0]
[0 1]

[1 1 0]
[0 0 1]
[1 0 1]
[0 1 0]
```

Seq is the most complicated operator to code up. We will give two example constructions based on the one-way signalling and (affine) semi-localisability definitions (restricting the intermediate system to the binary object by the Affine-Bit Sufficiency Theorem). In the case of the latter, we naively take each pair of local processes and perform a tensor contraction, then take a linearly-independent subset of the results. This is obviously an expensive procedure, but it guarantees that every column vector in the resulting matrix is actually semi-localisable rather than some affine combination of them, which is helpful for finding circuit implementations.

To swap the direction of seq from < to >, we can make use of a permutation matrix based on permuting the indices of a tensor.

```
# ONE—WAY SIGNALLING


def seq_sig(cA: Matrix, cB: Matrix) —> Matrix:
    # Constructs the seq type using the one—way signalling definition.
    # We prepare a set of linear equations Ax = b where x has dimension (cA.rows *
        cB.rows) + cA.cols. The first section of this gives the true vector, and
        the second gives the coefficients of the marginal when expressed as an
        affine combination of columns of cA.
    # For each column of dual_set(cB), we add a constraint that the marginal is
        identical to the same as the uniform (i.e. their difference is zero).
    # We add a constraint that the affine combination regenerates the marginal of
        the actual vector and a final constraint that it is affine.
    normB = (ones(1, cB.rows) @ cB.col(0))[0, 0]
    dB = dual_set(cB)
    dB_diff_uniform = dB − (1 / normB) * ones(dB.rows, dB.cols)
    projectors = TensorProduct(Matrix.eye(cA.rows, cA.rows), dB_diff_uniform.T)
    lhs = Matrix(
        BlockMatrix(
            [
                [projectors, Matrix.zeros(projectors.rows, cA.cols)],
                [
                    TensorProduct(
                        Matrix.eye(cA.rows, cA.rows), −(1 / normB) * ones(1, cB.
                            rows)
                    ),
                    cA,
                ],
                [Matrix.zeros(1, projectors.cols), ones(1, cA.cols)],
            ]
        )
    )
    rhs = Matrix.zeros(lhs.rows, 1)
    rhs[lhs.rows - 1, 0] = 1

    # Use sympy Gauss Jordan to obtain the solution space. The solutions are given
        as a symbolic vector, parameterising the solution space
    try:
        sol, _ = lhs.gauss_jordan_solve(rhs)
    except ValueError:
        return Matrix.zeros(cA.rows * cB.rows, 0)

    sol_mat = sym_tensor_to_mat(sol)

    # Cut off coefficients to just leave raw vectors
    return sol_mat[0 : (cA.rows * cB.rows), :]
```

```
# SEMI-LOCALISABILITY


def contracted_basis(
    basis: Matrix, tensor_shape: Tuple[int], indices: Tuple[int]
) -> Matrix:
    # Take each basis element, interpret as a tensor, then contract.
    # Collect these and reduce to a minimal basis.
    # basis is a basis whose elements can be interpreted as a tensor of shape
        tensor_shape.
    # We contract along all of indices
    full_tensor_shape = (basis.cols, *tensor_shape)
    full_indices = tuple((ind + 1 for ind in indices))
    ar = Array(basis.T, full_tensor_shape)
    contracted = tensorcontraction(ar, full_indices)
    contracted_mat = Matrix(
        contracted.reshape(basis.cols, len(contracted) // basis.cols)
    ).T
    columns = contracted_mat.columnspace()
    return Matrix.hstack(*columns)


def seq_lo2(cA: Matrix, cB: Matrix) -> Matrix:
    # Constructs the seq type as the 2-local graph type over two vertices and one
        edge
    fo2 = fo(2)
    lhs = parr(cA, fo2)
    rhs = lin_impl(fo2, cB)
    # A possible future improvement for performance might be to convert these to
        symbolic form to cut down on the dimension of the tensor that is built, and
         hopefully leveraging the symbolic simplification to optimise the space as
        we compute
    return contracted_basis(tensor(lhs, rhs), (cA.rows, 2, 2, cB.rows), (1, 2))
```

```
# PERMUTATIONS

import numpy as np


def tensor_permutation(
    tensor_shape: Tuple[int], index_permutation: Tuple[int]
) -> Matrix:
    assert len(tensor_shape) == len(index_permutation)
    total_dim = int(np.prod(tensor_shape))
    id = Matrix.eye(total_dim, total_dim)
    full_tensor_shape = (total_dim, *tensor_shape)
    full_perm = tuple((0, *(ind + 1 for ind in index_permutation)))
    ar = Array(id, full_tensor_shape)
    permuted = permutedims(ar, full_perm)
    return Matrix(permuted.reshape(total_dim, total_dim))
```

```
# TEST SEQ

chan2x2 = lin_impl(fo(2), fo(2))
chan3x3 = lin_impl(fo(3), fo(3))
oneway_lr = seq_sig(chan2x2, chan3x3)
print(oneway_lr.shape)

semilocal_lr = seq_lo2(chan2x2, chan3x3)
assert equal_objects(oneway_lr, semilocal_lr)

oneway_rl = tensor_permutation((9, 4), (1, 0)) @ seq_sig(chan3x3, chan2x2)
assert not equal_objects(oneway_lr, oneway_rl)

assert equal_objects(seq_sig(fo(2), chan2x2), tensor(fo(2), chan2x2))
assert equal_objects(seq_sig(dual_set(fo(2)), chan2x2), parr(dual_set(fo(2)),
    chan2x2))
assert equal_objects(seq_sig(chan2x2, fo(2)), parr(chan2x2, fo(2)))
assert equal_objects(
    seq_sig(chan2x2, dual_set(fo(2))), tensor(chan2x2, dual_set(fo(2)))
)
```

```
(36, 27)
```

Beyond $\mathrm{Caus}\left[\mathrm{Mat}[\mathbb{R}]\right]$, we can reuse most of this code to explore $\mathrm{Caus}\left[\mathrm{CP}^*\right]$ by taking the real vectors to indicate coefficients of some basis of completely positive matrices. For example, qubits can be encoded through the Pauli basis $\{I, X, Y, Z\}$ or some choice of four independent density matrices, e.g. $\{|0\rangle\langle 0|, |1\rangle\langle 1|, |+\rangle\langle +|, |i\rangle\langle i|\}$. The code for the operators should still work, but the verification of flatness will need changing to adapt the uniform state and effect for the chosen basis.

Let's go through a quantum example. We will aim to verify that 1-qubit process matrices are spanned by the definite causal orderings, and find the expansion of the OCB process as in Example 2.7.14.

Firstly, we pick out a basis for qubit states, effects, preparations (conditioned on some binary input), and binary tests. Using the basis $\{|0\rangle\langle 0|, |1\rangle\langle 1|, |+\rangle\langle +|, |i\rangle\langle i|\}$, qubit states are still spanned by `fo(4)` and the uniform effect is still `[1, 1, 1, 1]`, but the uniform state is now `[1/2, 1/2, 0, 0]`. We choose to create the bases explicitly to provide easy interpretability later.

```
h = Rational(1, 2)

qubit = fo(4)
ket0 = qubit[:, 0]
ket1 = qubit[:, 1]
mix = h * (ket0 + ket1)
ketp = qubit[:, 2]
ketm = ket0 + ket1 - ketp
keti = qubit[:, 3]
ketmi = ket0 + ket1 - ketm

qdiscard = dual_set(qubit)

bit = fo(2)
b0 = bit[:, 0]
b1 = bit[:, 1]
bmix = h * (b0 + b1)

constant_ket0 = Matrix.vstack(ket0, ket0)
constant_ket1 = Matrix.vstack(ket1, ket1)
constant_ketp = Matrix.vstack(ketp, ketp)
constant_keti = Matrix.vstack(keti, keti)
z_prep = Matrix.vstack(ket0, ket1)
x_prep = Matrix.vstack(ketp, ketm)
y_prep = Matrix.vstack(keti, ketmi)
prep_basis = Matrix.hstack(
    constant_ket0, constant_ket1, constant_ketp, constant_keti, z_prep, x_prep,
        y_prep
)
assert equal_objects(prep_basis, lin_impl(bit, qubit))

constant_0 = TensorProduct(ones(4, 1), b0)
constant_1 = TensorProduct(ones(4, 1), b1)
meas_z = Matrix.vstack(b0, b1, bmix, bmix)
meas_x = Matrix.vstack(bmix, bmix, b0, bmix)
meas_y = Matrix.vstack(bmix, bmix, bmix, b0)
test_basis = Matrix.hstack(constant_0, constant_1, meas_z, meas_x, meas_y)
assert equal_objects(test_basis, lin_impl(qubit, bit))

qubit_channel = contracted_basis(tensor(test_basis, prep_basis), (4, 2, 2, 4), (1,
    2))
assert equal_objects(qubit_channel, lin_impl(qubit, qubit))
```

To construct the definite causal orderings, we can take separable combinations of states, channels, and effects. By symmetry of the systems, we can reverse the order by permuting the tensors. Then we can verify that their union coincides with the space of density matrices.

```
l_to_r = tensor(tensor(qubit, qubit_channel), qdiscard)
r_to_l = tensor_permutation((4, 4, 4, 4), (2, 3, 0, 1)) @ l_to_r

process_matrix = union(l_to_r, r_to_l)
assert equal_objects(process_matrix, dual_set(tensor(qubit_channel, qubit_channel))
    )
```

We then build the OCB process using the encodings of the Pauli matrices and identify the coefficients that express it as an affine combination of the basis processes.

```
i_state = 2 * mix
z_state = Matrix([1, -1, 0, 0])
x_state = Matrix([-1, -1, 2, 0])
i_effect = qdiscard
z_effect = Matrix([1, -1, 0, 0])

iiii = tensor(tensor(i_state, i_effect), tensor(i_state, i_effect))
izzi = tensor(tensor(i_state, z_effect), tensor(z_state, i_effect))
zixz = tensor(tensor(z_state, i_effect), tensor(x_state, z_effect))
w_ocb = Rational(1, 4) * (iiii + sqrt(h) * izzi + sqrt(h) * zixz)

assert check_containment(w_ocb, process_matrix)

coeffs = process_matrix.solve(w_ocb)

for i, c in enumerate(coeffs):
    if c != 0:
        print(i, c)
```

```
0 1/4
1 1/4
2 -sqrt(2)/4
4 sqrt(2)/4
13 1/4
14 1/4
15 -sqrt(2)/4
17 sqrt(2)/4
52 -sqrt(2)/4
61 -sqrt(2)/4
70 sqrt(2)/2
```

Being a combination of 11 terms is fine, but may be too many terms to realistically use in practical contexts. Choosing a different basis set would get a simpler expression. For example, promoting the uniform state and the noisy channel to basis elements reveals the exact decomposition given in Example 2.7.14.

```
noisy = Matrix.vstack(mix, mix, mix, mix)

qubit_alt = union(mix, qubit)
qubit_channel_alt = union(noisy, qubit_channel)
l_to_r_alt = tensor(tensor(qubit_alt, qubit_channel_alt), qdiscard)
r_to_l_alt = tensor_permutation((4, 4, 4, 4), (2, 3, 0, 1)) @ l_to_r_alt
process_matrix_alt = union(l_to_r_alt, r_to_l_alt)

coeffs_alt = process_matrix_alt.solve(w_ocb)

for (i, c) in enumerate(coeffs_alt):
    if c != 0:
        print(i, c)
```

```
0 1
2 -sqrt(2)/2
4 sqrt(2)/2
52 -sqrt(2)/2
70 sqrt(2)/2
```