

The d -Separation Criterion in Categorical Probability

Tobias Fritz

*Department of Mathematics
University of Innsbruck
6020 Innsbruck, Austria*

TOBIAS.FRITZ@UIBK.AC.AT

Andreas Klingler

*Institute for Theoretical Physics
University of Innsbruck
6020 Innsbruck, Austria*

ANDREAS.KLINGLER@UIBK.AC.AT

Editor: Jin Tian

Abstract

The d -separation criterion detects the compatibility of a joint probability distribution with a directed acyclic graph through certain conditional independences. In this work, we study this problem in the context of categorical probability theory by introducing a categorical definition of causal models, a categorical notion of d -separation, and proving an abstract version of the d -separation criterion. This approach has two main benefits. First, categorical d -separation is a very intuitive criterion based on topological connectedness. Second, our results apply both to measure-theoretic probability (with standard Borel spaces) and beyond probability theory, including to deterministic and possibilistic networks. It therefore provides a clean proof of the equivalence of local and global Markov properties with causal compatibility for continuous and mixed random variables as well as deterministic and possibilistic variables.

Keywords: Categorical Probability, Markov categories, Bayesian networks, causal models, causal compatibility

1. Introduction

The d -separation criterion (Pearl, 2009) is a necessary and sufficient condition for the compatibility of a probability distribution with a causal structure in the form of a directed acyclic graph (DAG). It states that a joint probability distribution of a collection of random variables is compatible with the DAG—in the sense that each of its nodes is one of the given variables, and each arrow denotes the possibility of causal influence—if and only if the distribution satisfies a list of conditional independence relations encoded in the structure of the DAG.

In this paper, we study this causal compatibility problem in the framework of *categorical probability theory*. We elaborate on the framework of generalized causal models recently proposed by Fritz and Liang (2022), introduce a categorical notion of d -separation, and prove a categorical generalization of the d -separation criterion.

The framework of generalized causal models involves freely generated categories called *free Markov categories* (Fritz and Liang, 2022). Starting from a set of morphisms as building blocks (representing the basic causal mechanisms), we construct a morphism in this category

by assembling these blocks consistently. More precisely, a morphism in such a free category is a *string diagram* consisting of wires and boxes. In our context, these diagrams represent the causal models. Each wire corresponds to a local random variable and each box to a causal mechanism generating one or several new output variables from its input ones. A morphism in any Markov category is *compatible* with such a causal model if it can be decomposed in the form specified by the causal model; this generalizes the standard factorization definition of Bayesian networks (Fong, 2013).

Further, we define a categorical notion of d -separation in terms of the string diagrams. A causal structure displays d -separation of a collection of outputs \mathcal{Z} if it becomes disconnected upon removing the corresponding wires in \mathcal{Z} . We show that this is equivalent to classical d -separation for the class of causal models for which the latter is defined, namely those corresponding to DAGs.

Finally, we prove an abstract version of the d -separation criterion for our categorical notions of causal model and d -separation. We show that a given distribution, possibly depending on additional input variables, is compatible with a causal structure if and only if it displays conditional independence between sets of variables whenever the causal model displays d -separation for the corresponding sets of wires. This result goes beyond the classical result for “distributions”; it holds for arbitrary morphisms in any suitable Markov category \mathbf{C} . A central structural ingredient in the proof is the assumption that \mathbf{C} has conditionals (Fritz, 2020). Intuitively, this property states that every Markov kernel with multiple output variables arises by generating one output variable after the other, namely as a function of the input variables and previously generated outputs. For example, Markov kernels between finite sets have conditionals since every conditional probability distribution $p(x, y | a)$ factorizes into

$$p(x, y | a) = p(x | y, a) \cdot p(y | a).$$

The conditional distribution $p(x | y, a)$ arises by solving this equation for every x, y , and a and taking it to be arbitrary when $p(y | a) = 0$. Conditionals also exist for continuous or mixed random variables (technically for Markov kernels between standard Borel spaces, but not arbitrary measurable spaces) as well as in the Markov category of Gaussian kernels. Moreover, we will show that Markov categories with conditionals also exist in settings that do not involve numerical probabilities. This implies that our results apply equally easily to all of these cases.

This approach leads to new insights into the structure of d -separation and generalizes the classical result, to the best of our knowledge, in four notable directions:

- It generalizes the d -separation criterion in the pure DAG setting to a larger class of causal models. First, it gives a criterion for the compatibility of *Markov kernels* with a causal structure rather than merely for probability distributions. This is because Markov categories formalizing probabilities have Markov kernels as their morphisms, which therefore become the basic primitives of our formalism. Second, the local mechanisms in the causal structures are allowed to have multiple outputs and to appear multiple times. While all those generalizations can also be included in the DAG setting via suitable workarounds, as considered for example by Forré (2021), string diagrams contain them already intrinsically.

- Categorical d -separation on string diagrams is more general and conceptually simpler than “classical” d -separation on DAGs since it only relies on three intuitive operations on string diagrams: *marginalization*, *removing wires*, and *checking disconnectedness*.
- It provides a uniform proof of the equivalence between d -separation and causal compatibility for discrete variables, continuous variables (or rather arbitrary variables taking values in standard Borel spaces), and Gaussian variables. This follows from the fact that we reason abstractly and even more generally in any Markov category with conditionals. Together with the first point, this approach, therefore, generalizes the d -separation criterion on the pure DAG setting for continuous variables shown for distribution with a density (Lauritzen et al., 1990; Lauritzen, 1996), which was later extended to arbitrary variables with values in standard Borel spaces in Forré and Mooij (2017).
- The abstract reasoning using Markov categories implies that the d -separation criterion also applies to non-probabilistic settings, for example to networks modeling deterministic or possibilistic variables and causal mechanisms.

This paper is organized as follows. In Section 2, we give a more detailed non-technical overview of the main concepts of this paper, including causal models as string diagrams and categorical d -separation. In Section 3, we recall and explain the definitions of Markov categories and gs-monoidal categories. In Section 4, we review the construction of free gs-monoidal and free Markov categories, leading to generalized causal models and causal compatibility for morphisms in Markov categories. In Section 5, we present the notions of conditionals and conditional independence. In Section 6, we introduce a categorical notion of d -separation and prove the main results of this paper, namely the equivalence with classical d -separation (Proposition 22) and our abstract version of the d -separation criterion for causal compatibility (Theorem 28).

2. Causal Models and Markov Categories

In this paper, we study causal models, in the sense of Bayesian networks, from a categorical perspective. In order to make this accessible to readers without a formal background in category theory, we outline this paper’s main concepts and results in this section.

Section 2.1 gives a non-technical introduction to the string diagrammatic formalism representing causal models and its relation to the classical DAG formalism. We then present the concept of causal compatibility in Section 2.2 as a functorial property. In Section 2.3, we present categorical d -separation as a statement about the connectedness of the string diagram and explain our result on the d -separation criterion in Section 2.4. Therefore, the string diagrammatic approach opens the door for a new perspective on d -separation and its connection to causal compatibility.

2.1 Extending Bayesian Networks with String Diagrams

Traditionally, the definition of Bayesian networks relies on the concept of directed acyclic graphs (DAG) since such a graph encodes the underlying causal structure. Each node $v \in V(G)$ of a DAG G is associated with a random variable X_v , and each directed arrow $w \rightarrow v$

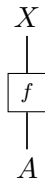
is associated with a direct possible causal dependence of the variable X_v on X_w . Formally, if we index the nodes by $1, \dots, n$, then this means that a joint probability distribution P is compatible with G if P factorizes into

$$P(X_1, \dots, X_n) = \prod_{i=1}^n P(X_i | \text{Pa}(X_i))$$

where $\text{Pa}(X_i) = \{X_j : G \text{ contains the arrow } j \rightarrow i\}$ is the set of parents of X_i relative to G .

In the categorical framework, we can refine the notion of Bayesian networks via string diagrams, an idea that extends the categorical approach developed by Fong (2013) and others (Rischel, 2020; Rischel and Weichwald, 2021; Gao, 2022; Jacobs et al., 2019). These diagrams arise naturally in *categorical probability*, which refers more generally to recent work on axiomatizing probability theory with simple, algebraic rules which avoid the low-level machinery of measure theory by hiding it in the proofs of the relevant axioms (Fritz, 2020).

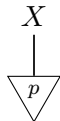
In categorical probability, the basic primitives are conditional probability distributions, also known as Markov kernels. A Markov kernel is represented as a morphism $f : A \rightarrow X$, depicted as



Here, A represents the input space and X the output space. The morphism f can be understood as a map that assigns a probability distribution on X to every point $a \in A$ in a measurable way. Depending on the particular context, f can be a stochastic matrix representing a Markov kernel between finite sets, a linear map with added Gaussian noise, or an arbitrary measurable Markov kernel. For example, a finite stochastic matrix is defined as a map

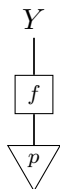
$$\begin{aligned} f : X \times A &\rightarrow [0, 1] \\ (x, a) &\mapsto f(x|a) \end{aligned}$$

such that $f(-|a)$ is a probability distribution on X for every $a \in A$. When choosing A to be a singleton set I , a morphism $p : I \rightarrow X$ represents a probability distribution, depicted as



Here, I is not drawn in the diagram, the upper wire represents the probability space X , and the box p the distribution, considered now as a function with no input and one (random) output in X . Therefore, probability distributions are considered as special cases of the basic primitives in categorical probability by setting the input space A to be a trivial object I .

These morphisms can be composed into new ones. For example, composing $f : X \rightarrow Y$ with $p : I \rightarrow X$, pictorially



gives rise to a new distribution of a random variable with values in Y . In the concrete setting, the composition is given by the Chapman–Kolmogorov equation, which reads for discrete distributions as

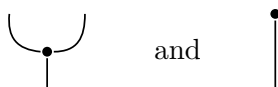
$$(f \circ p)(y) := \sum_{x \in X} f(y|x) \cdot p(x)$$

and for Markov kernels on arbitrary measurable spaces as

$$(f \circ p)(A) := \int_X f(A|x) p(dx).$$

where $A \in \Sigma_Y$, with Σ_Y the σ -algebra of the measurable space Y .

Each random variable can be copied or marginalized over, for which there exists special string diagrammatic notation:



These maps satisfy manipulation rules formalizing basic properties of information flow, for example:

- (i) When marginalizing the output of a box, the action of the box becomes irrelevant and therefore the box can be removed from the diagram, pictorially

(1)

This also applies in the case of multiple outputs since the formalism allows viewing multiple wires as a single one (which corresponds to the operation of taking products of measurable spaces).

- (ii) When marginalizing an output of the copy map, the resulting map acts like an identity, pictorially

(2)

We formally axiomatize the semantics of the resulting string diagram calculus in Definition 1 as the definition of a *Markov category*.

Each flavour of probability has its own Markov category. There is a Markov category for discrete probabilities (called **FinStoch**), one for Gaussian probability (**Gauss**), one for

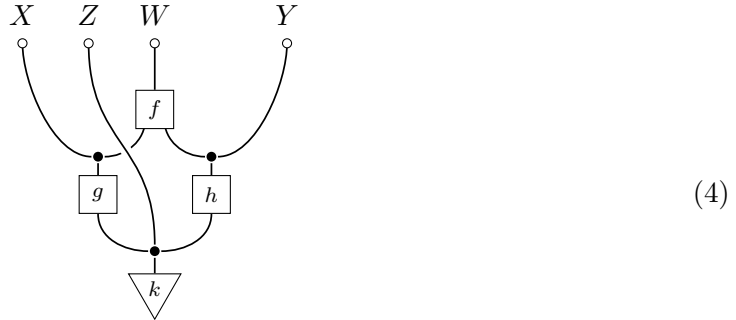
probability theory on standard Borel spaces (**BorelStoch**), and one for probability theory on arbitrary measurable spaces (**Stoch**).

But there are also non-probabilistic Markov categories. These include Markov categories in which every morphism is deterministic (for example **FinSet**) and possibilistic Markov categories (for example **FinSetMulti**), in which the non-determinism comes merely in the form a distinction between which outcomes are possible and which ones are not: for a morphism $f : X \rightarrow Y$, we have $f(y|x) = 1$ or $f(y|x) = 0$ according to whether a certain outcome $y \in Y$ is possible on an input $x \in X$ or not.

The causal structures themselves are also morphisms in yet another class of Markov categories, the so-called *free Markov categories*. These form our framework for causal structures which generalize DAGs. Rather than Markov kernels, the morphisms in a free Markov category are the string diagrams themselves, i.e. all “networks” that can be built by wiring together a set of boxes, similar to how an electrical circuit is obtained by wiring together electrical components. In this way, string diagrams constitute generalized causal models. In particular, we will see that string diagrams can represent arbitrary DAG causal models. Consider for example this DAG:



As a string diagram, this causal structure looks like this:



By definition, a probability distribution $P(X, Y, Z, W)$ is compatible with this structure if

$$P(X, Y, Z, W) = P(W|X, Y) \cdot P(X|Z) \cdot P(Y|Z) \cdot P(Z).$$

Each box in the string diagram represents a placeholder for a conditional distribution. For example, the box f stands for the conditional distribution $P(W|XY)$. Each loose wire represents a random variable where the wire’s name indicates the variable’s name.¹ Further, each variable has a corresponding type, a placeholder for the measurable space in which

1. Note that we use the term “wire” as referring to an entire connected piece of circuitry, i.e. traversing a black dot in the diagram does not leave the wire.

the variable takes values. Unless necessary, we will not explicitly mention the type of each variable.

In our setting, some wires are connected to an “output” representing a variable that is “observed” rather than marginalized over, as indicated by the white dots. Note that every variable then becomes an output in at most one way. While the above string diagram connects every wire to an output, the following string diagram also contains “unobserved” variables:



In the setting of discrete probability, a distribution $P(X, Y)$ is therefore compatible with this structure, if

$$P(X, Y) = \sum_Z P(X | Z) \cdot P(Y | Z) \cdot P(Z),$$

where the marginalized variable Z is represented by the wire not connected to the output.

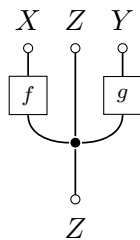
In certain situations, we only consider causal models where every wire is connected to an output. Throughout the paper, we call such diagrams *pure blooms* (see Definition 7). Unless stated otherwise, we will denote the output by the name of the wire connected to the output. We will define certain operations, like d -separations, only with respect to output wires, highlighting that we cannot address latent variables (i.e. wires that are not connected to an output) in general.

Causal models given by a DAG correspond to pure bloom string diagrams where each box has precisely one output wire, and the input wires represent the parents of the node in the DAG. The following table explains the relation between nodes in a DAG, boxes in the string diagram and the corresponding conditional probability distribution.

	DAG	string diagram	conditional distribution
one output			$P(X ABC)$
ident. outputs			$P(XX ABC)$
diff. outputs	X		$P(XY ABC)$

Using string diagrams themselves as generalized causal models allows us to go beyond the DAG approach in several directions:

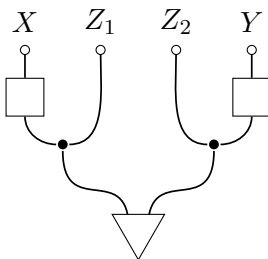
- String diagrams in Markov categories describe Markov kernels instead of just probability distributions. Therefore, the string diagram language allows for modeling causal structures with inputs, such as



This describes a causal structure in which the input variable at the bottom does not have any particular distribution itself. For example, a conditional probability distribution is compatible with the above string diagram if

$$P(X, A, Y | A) = P(X | A) \cdot P(Y | A)$$

- As indicated in the table, boxes in the string diagram can have more than a single output wire. This allows for causal structures like

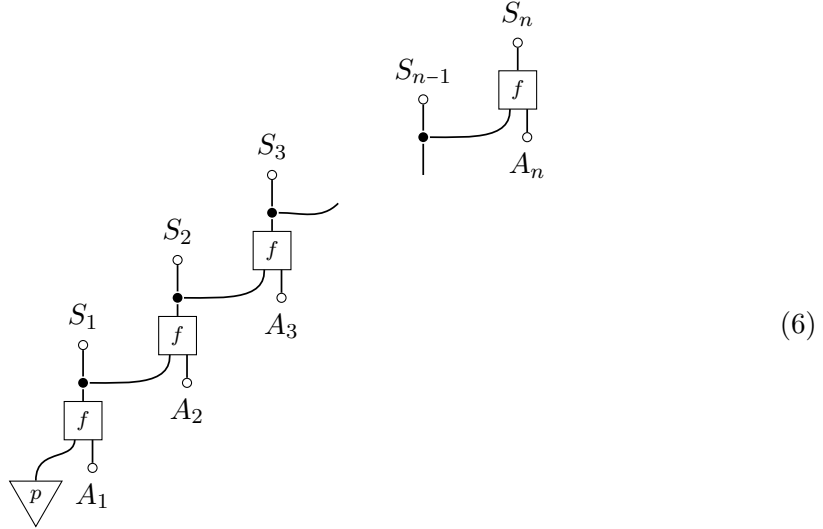


which are not native to the DAG framework (see Example 7(i) for a detailed discussion of this structure). In this situation a probability distribution $P(X, Y, Z_1, Z_2)$ is compatible with the given structure if

$$P(X, Y, Z_1, Z_2) = P(X | Z_1) \cdot P(Y | Z_2) \cdot P(Z_1, Z_2).$$

- String diagrams allow for the use of identical boxes multiple times. This allows to represent for example the transition probabilities of a *Markov decision processes* (Put-

erman, 1994, Section 2.1) as the causal structure



Here, S_i denotes the state variable and A_i the action variables at time i and the boxes are all identical since such a decision process is given by a fixed conditional probability distribution

$$P(S_{i+1} | S_i, A_i),$$

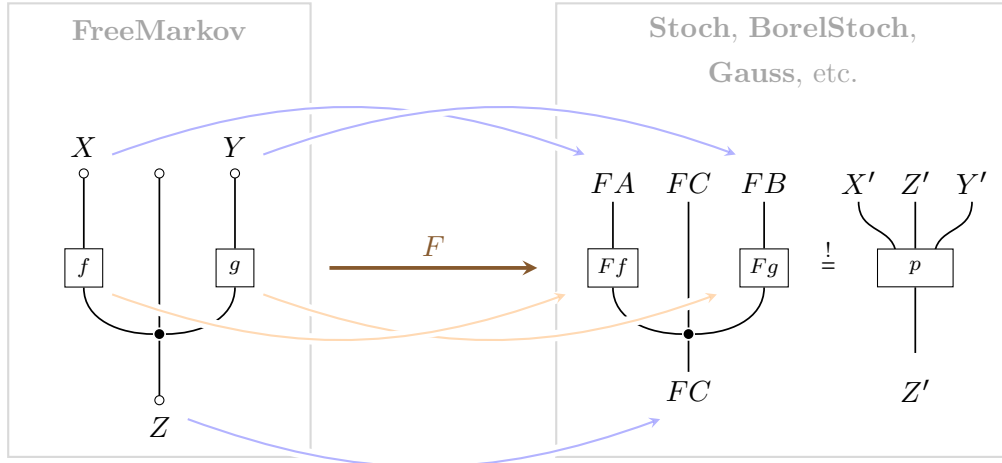
which highlights that a Markov decision process has no additional internal memory besides the previous state. Moreover, in this situation the types of the variables A_1, \dots, A_n must be the same, namely a particular action space A , and similarly the S_1, \dots, S_n all represent the same state space S . Therefore, the box f can be seen as a morphism $S \otimes A \rightarrow S$.

Throughout the paper, we will usually follow the convention of labelling wires by their types together with an index that lets us address them individually, and at the same time identify these wires with the variables they carry. We do not index the boxes as there is no need to reference the distinct instances of the same type of box separately.

2.2 Causal Compatibility for Markov Categories

A distribution is compatible with a causal model if it can be written as a composition in precisely the way specified by the corresponding string diagram. In other words, every type W in the string diagram must be mapped to a concrete measurable space FW and every box f to a concrete Markov kernel Ff .

In category-theoretic language, this is captured in the following way (Fong, 2013). A morphism p in a concrete Markov category is compatible with a causal structure φ if and only if there exists a Markov functor F such that $p = F\varphi$. Intuitively, this functor acts in the following way:



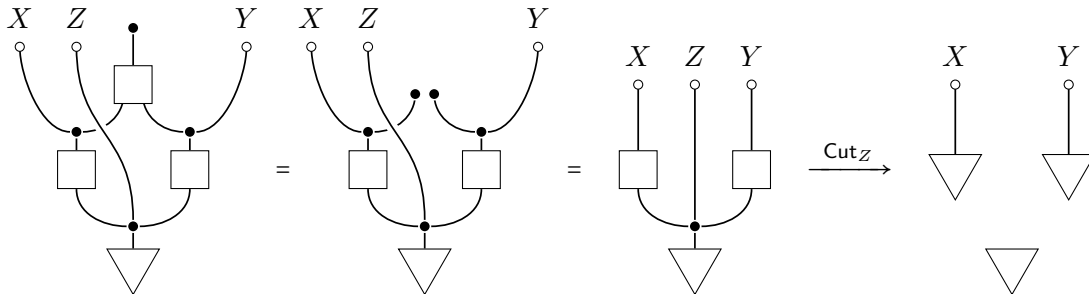
where p is the given morphism in a concrete Markov category and **FreeMarkov** the free Markov category whose morphisms capture the causal models. Further, X has type A , Y has type B , and Z has type C . So if the original p takes input from a measurable space Z' and outputs values in measurable spaces X' , Z' and Y' , then the types must match in the sense that $FA = X'$ etc.

2.3 Categorical d -Separation

The notion of d -separation for DAGs is a criterion relating conditional independence in causal models to the causal compatibility with a DAG. In Section 6.1, we introduce a categorical notion of d -separation from a different perspective. Although this notion looks different and *conceptually* much simpler than classical d -separation, we prove that it coincides with the classical one when considering causal models on DAGs.

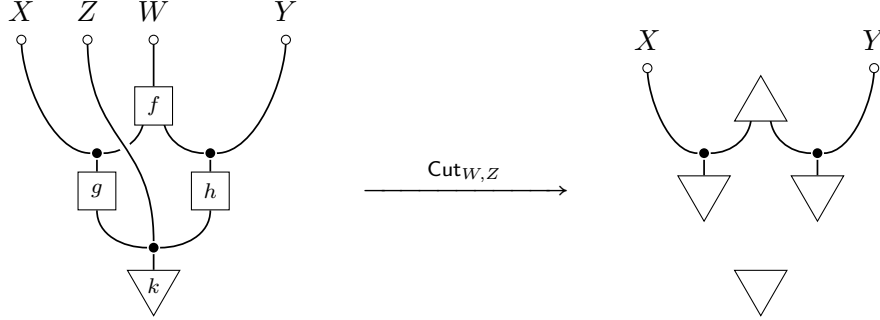
An output wire Z *categorically d -separates* the output X from output Y if X and Y become disconnected upon *marginalizing* over all wires not in X, Y and Z and then *removing* the wire Z . We also express this by saying that X is categorically d -separated from Y by Z .

Consider, for example, the DAG in Equation (3). Z classically d -separates X from Y , as one can see based on the fact that the only paths between X and Y are the collider $X \rightarrow W \leftarrow Y$ and the fork $X \leftarrow Z \rightarrow Y$. In the corresponding string diagram, Equation (4), we witness categorical d -separation by first marginalizing over W , then removing the Z wire, and finally observing that X and Y are disconnected, pictorially:



Here we have used the rules of marginalization, namely Equation (1) and Equation (2).

On the other hand, X is not d -separated from Y by W and Z due to the collider $X \rightarrow W \leftarrow Y$. In the string diagram, this is apparent since upon removing the wires Z and W ,



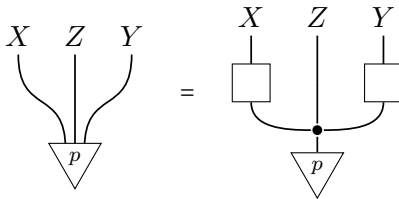
X and Y are still connected.

We note that there also are approaches to classical d -separation based on topological disconnectedness, such as using the moralized graph of a DAG (Lauritzen, 1996, Proposition 3.25). An application of this criterion consists of four steps: first, *marginalizing* a certain subset of variables not present in the d -separation; second, *moralizing* the DAG to an undirected graph; third, removing every vertex associated to Z ; and finally, checking *disconnectedness*. Categorical d -separation is conceptually simpler than the DAG approach since it consists only of three steps, namely marginalizing *all* variables not involved in the putative d -separation, removing the Z wires, and then checking disconnectedness. The moralization step is omitted since its function is already encoded in the structure of string diagrams. Therefore, categorical d -separation is a more intuitive procedure compared to the standard definitions of d -separation on DAGs.

2.4 d -Separation and Causal Compatibility

For Bayesian networks, d -separation detects conditional independences for any compatible probability distribution. In particular, if two output wires X and Y are d -separated by the output wire Z , we have that X is independent of Y given Z (denoted as $X \perp Y \mid Z$).

Conditional independence in Markov categories is defined string diagrammatically. A probability distribution p on a product space $X \times Y \times Z$ displays the conditional independence $X \perp Y \mid Z$ if it can be written in the form



This reflects the classical notion of conditional independence with the two unnamed boxes corresponding to the conditionals (see also Fritz (2020, Remark 12.4) for discussion); in the context of discrete probability, it encodes the equation

$$P(X = x, Y = y, Z = z) = P(X = x \mid Z = z) \cdot P(Y = y \mid Z = z) \cdot P(Z = z).$$

But also for Gaussian random variables as well as measures on standard Borel spaces, one recovers the intuitive notions of conditional independence (Fritz, 2020, Section 12). Moreover, the diagrammatic definition is even sufficient to derive the semigraphoid properties (Fritz, 2020, Lemma 12.5). It also generalizes to a notion of conditional independence for morphisms with inputs (see Definition 16).

In Section 6.2, we prove that the categorical d -separation criterion applies to generalized causal models in Markov categories. For this reason, we define a notion of conditional independence which applies to arbitrary Markov kernels (Definition 16). We first prove the soundness of the d -separation, namely, if Z categorically d -separates X from Y , then $X \perp Y \mid Z$ (Corollary 26). Second, we prove the completeness of d -separation for causal compatibility: if a Markov kernel satisfies the global Markov property for a causal structure (i.e. every d -separated triple shows conditional independence), then the Markov kernel is compatible with the structure (see Theorem 28 for the precise assumptions and also the equivalence with the local Markov property).

A central assumption for the proof is the existence of conditionals (see Definition 12). Intuitively it says that the outputs of any morphism f can be generated successively while having access to all prior information. Conditionals exist in discrete probability, measure-theoretic probability on standard Borel spaces and in Gaussian probability, and this facilitates the application of our results to all of these cases, where the second case includes continuous random variables.

3. gs-Monoidal and Markov Categories

In the following, we recall the definitions of Markov categories and gs-monoidal categories. Markov categories are the basic structure modeling different flavors of probability, including discrete random variables, Gaussian random variables, continuous random variables on standard Borel spaces, or random variables on arbitrary measurable spaces. We assume some familiarity with symmetric monoidal categories and string diagrams. We refer to Leinster (2014) and Perrone (2019) for a general introduction to category theory and to Heunen and Vicary (2019), Baez and Stay (2010) and Coecke (2009) for a more detailed account of monoidal categories and their string diagrammatic calculus.

Intuitively, a monoidal category is a category with a product structure \otimes . Specifically, two objects A, B in the category have an associated product object $A \otimes B$. Moreover, if $f : A \rightarrow X$ and $g : B \rightarrow Y$ are morphisms, then \otimes gives rise to a new morphism $f \otimes g : A \otimes B \rightarrow X \otimes Y$. This monoidal product can have different realizations. For example, the category **Set** is monoidal with the Cartesian product of sets and functions. A monoidal category is *symmetric* if the objects $A \otimes B$ and $B \otimes A$ are naturally isomorphic in a nice way. All of the monoidal categories in this paper are symmetric.

The notion of gs-monoidal categories goes back to Gadducci (1996, Definition 3.9) and Corradini and Gadducci (1999). There, it was considered with a different motivation in the context of term graphs and term graph rewriting.

Definition 1 (gs-monoidal category and Markov category).

- (i) A gs-monoidal category \mathbf{C} is a symmetric monoidal category with monoidal unit I equipped with a commutative comonoid structure for every object $X \in \mathbf{C}$ given by a

counit $\text{del}_X : X \rightarrow I$ and a comultiplication $\text{copy}_X : X \rightarrow X \otimes X$. In the string diagrammatic notation, these operations are depicted as

$$\text{del}_X := \begin{array}{c} \bullet \\ | \end{array} \qquad \text{copy}_X := \begin{array}{c} \cup \\ | \end{array}$$

They are required to satisfy the commutative comonoid equations, diagrammatically given by

$$\begin{array}{c} \cup \\ | \end{array} = \begin{array}{c} \cup \\ | \end{array} \qquad \begin{array}{c} \bullet \\ | \end{array} = \begin{array}{c} \bullet \\ | \end{array} = \begin{array}{c} | \end{array} \qquad \begin{array}{c} \cup \\ | \end{array} = \begin{array}{c} \cup \\ | \end{array} \quad (7)$$

and to be compatible with the monoidal structure, i.e.

$$\begin{array}{c} \bullet \\ | \\ A \otimes B \end{array} = \begin{array}{c} \bullet \\ | \\ A \end{array} \begin{array}{c} \bullet \\ | \\ B \end{array} \qquad \begin{array}{c} A \quad B \quad A \quad B \\ \cup \quad \cup \\ | \quad | \\ A \quad B \end{array} = \begin{array}{c} A \otimes B \quad A \otimes B \\ \cup \\ | \\ A \otimes B \end{array} \quad (8)$$

as well as

$$\begin{array}{c} \bullet \\ | \\ I \end{array} = \begin{array}{c} \square \end{array} \qquad \begin{array}{c} I \quad I \\ \cup \\ | \\ I \end{array} = \begin{array}{c} \square \end{array} \quad (9)$$

(ii) A gs -monoidal category is called a Markov category if del is in addition natural, i.e. if for all morphisms f ,

$$\begin{array}{c} \bullet \\ \square \\ | \\ f \end{array} = \begin{array}{c} \bullet \\ | \end{array} \quad (10)$$

We refer to Fritz and Liang (2022, Remark 2.2) for more details on the multifarious history of these notions. When considering morphisms with multiple inputs or outputs, we often denote the collective inputs and outputs as (multi)sets instead of tensor products. For example, we write

$$\begin{array}{c} X \quad Y \\ \cup \\ \square \\ | \\ A \quad B \quad C \end{array} = \begin{array}{c} \mathcal{W} \\ | \\ \square \\ | \\ \mathcal{V} \end{array}$$

where $\mathcal{V} = \{A, B, C\}$ and $\mathcal{W} = \{X, Y\}$. Modulo some abuse of notation, the order of the inputs and outputs is irrelevant since we can always permute the wires, and therefore it is enough to consider \mathcal{V} and \mathcal{W} as multisets rather than lists.

Example 1. The most important examples of Markov categories are the following:

- (a) **FinStoch** is the Markov category of finite stochastic maps. The objects are finite sets X , morphisms $p : I \rightarrow X$ are probability distributions and general morphisms $f : X \rightarrow Y$ are stochastic matrices with entries $f(y|x)$. The symmetric monoidal structure is given by the Cartesian product of sets.
- (b) **Stoch** is the Markov category of arbitrary Markov kernels on measurable spaces. The objects are measurable spaces, morphisms $p : I \rightarrow X$ are probability measures, and general morphisms $f : X \rightarrow Y$ are measurable Markov kernels.
- (c) **BorelStoch** is given by **Stoch** restricted to standard Borel spaces as objects.
- (d) **Gauss** is the Markov category of Gaussian probability distributions. The objects are the spaces \mathbb{R}^n , morphisms $p : I \rightarrow \mathbb{R}^n$ are Gaussian probability measures and general morphisms $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$ can be understood as stochastic maps of the form

$$x \mapsto Ax + \xi$$

where A is any real $m \times n$ matrix and ξ is an independent Gaussian variable with given mean and variance. **Gauss** is a (non-full) subcategory of **BorelStoch**.

- (e) **FinSet** is the Markov category of finite sets and (deterministic) functions. Therefore, a morphism $f : X \rightarrow Y$ is a function from X to Y and $p : I \rightarrow X$ is a single element of X . The copy maps are given by

$$\begin{aligned} \text{copy}_X : X &\rightarrow X \times X \\ x &\mapsto (x, x) \end{aligned}$$

and the deletion morphism $\text{del}_X : X \rightarrow I$ is the only map to the one-element set I .

Equivalently, a morphism $f : X \rightarrow Y$ can be thought of a matrix $(f(y|x))_{y \in Y, x \in X}$ with entries in $\{0, 1\}$ by defining $f(y|x) = 1$ if and only if $f(x) = y$. This shows that there is a functor **FinSet** \rightarrow **FinStoch** through which the Markov category structure of **FinSet** is inherited from **FinStoch**. **FinSet** also generalizes to the category **Set** by considering arbitrary sets.

- (f) **FinSetMulti** is the Markov category of finite sets and multivalued maps. A morphism $f : X \rightarrow Y$ is given by a matrix $(f(y|x))_{y \in Y, x \in X}$ with entries in $\{0, 1\}$ and subject to the condition that for every x , we have $f(y|x) = 1$ for some y . If $f(y|x) = 1$, then y is possible when given x , while $f(y|x) = 0$ implies that y is impossible given x . We define composition of two morphisms $f : X \rightarrow Y$ and $g : Y \rightarrow Z$ via

$$(g \circ f)(z|x) := \sum_{y \in Y} g(z|y) \cdot f(y|x)$$

where we use the convention that $1 + 1 = 1$. This suggests the same notation as for morphisms as in **FinStoch**.

For further details on the composition, the symmetric monoidal structure and the Markov category structure of these examples, we refer to Fritz (2020).

In our context, gs-monoidal categories that are not Markov categories play more of an auxiliary role which we will detail below.

4. Free Markov Categories and Generalized Causal Models

Causal models are a framework for studying and modeling dependencies between random variables. In this section, we introduce such a framework in the language of Markov categories. We therefore investigate causal relationships independently of the particular notion of probability behind it (discrete, measure-theoretic, Gaussian, etc).

Free Markov categories are the tailored notion for these purposes. These categories contain precisely all blueprints for causal networks that can be built from a bunch of given causal mechanisms. A morphism in this category is then a *generalized causal model*.

The remainder of this section explains this in technical detail based on the formalism of free gs-monoidal categories and free Markov categories from Fritz and Liang (2022).² This part is structured as follows: In Section 4.1, we introduce the category of hypergraphs. In Section 4.2 we use hypergraphs to define gs-monoidal string diagrams, free gs-monoidal categories, and subsequently free Markov categories. In Section 4.3 we introduce generalized causal models and causal compatibility of morphisms in arbitrary Markov categories.

4.1 The Category of Hypergraphs

A gs-monoidal string diagram, and therefore also a generalized causal model is defined as a hypergraph with extra structure. To define the relevant notion of hypergraph following Bonchi et al. (2016), let first \mathbf{I} be the category defined in the following way:

- (i) The set of objects is given by $\{(k, \ell) \mid k, \ell \in \mathbb{N}\} \cup \{\star\}$.
- (ii) Besides the identity morphisms, for every (k, ℓ) there are $k + \ell$ different morphisms

$$\text{in}_1, \dots, \text{in}_k, \text{out}_1, \dots, \text{out}_\ell : (k, \ell) \rightarrow \star,$$

and no other morphisms.

It is not necessary to specify a composition rule in \mathbf{I} since no compositions except the trivial ones can be formed.

Definition 2. A functor $G : \mathbf{I} \rightarrow \mathbf{Set}$ is called a hypergraph. Accordingly, we define the functor category

$$\mathbf{Hyp} := \mathbf{Set}^{\mathbf{I}}$$

to be the category of hypergraphs.

Intuitively, the functor G characterizes our common interpretation of (directed) hypergraphs in the following way:

- (i) $W(G) := G(\star)$ is the set of vertices, which we will call *wires*.
- (ii) $B_{k,\ell}(G) := G((k, \ell))$ is the set of hyperedges, which we will call *boxes*, with k inputs and ℓ outputs.
- (iii) $G(\text{in}_i)$ specifies the i th input wire of every box.

2. Another very similar construction of free gs-monoidal categories has been given independently in Milosavljevic and Zanasi (2022).

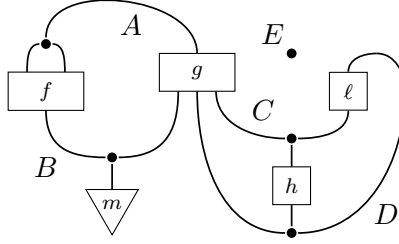


Figure 1: Pictorial representation of a hypergraph with wire set $\{A, B, C, D, E\}$ and box set $\{f, g, h, m, n\}$. For example, the box f has one input incident to the wire B and two outputs both incident to the wire A . The wire E is not incident to any box.

(iv) $G(\text{out}_j)$ specifies the j th output wire of every box.

While the set of wires and boxes of a hypergraph may be infinite, the number of inputs and outputs of a box is always finite. We present a pictorial representation of a hypergraph in Figure 1.

Given a box $b \in B_{k,\ell}(G)$ and a wire $A \in W(G)$, we define

$$\begin{aligned} \text{in}(b, A) &:= |\{j \in \{1, \dots, \ell\} : \text{in}_j(b) = A\}|, \\ \text{out}(b, A) &:= |\{i \in \{1, \dots, k\} : \text{out}_i(b) = A\}|. \end{aligned}$$

Thus $\text{in}(b, A)$ and $\text{out}(b, A)$ counts the number of incoming or outgoing wires of type A in the box b . We also define the sets of inputs and outputs as

$$\begin{aligned} \text{in}(b) &:= \{\text{in}_i(b) : i \in \{1, \dots, \ell\}\}, \\ \text{out}(b) &:= \{\text{out}_i(b) : i \in \{1, \dots, k\}\}, \end{aligned}$$

where repeated wires are counted only once.

Next, we analyze the morphisms in **Hyp**. Since **Hyp** is a functor category, a morphism $\alpha : F \rightarrow G$ is precisely a natural transformation $\alpha : F \Rightarrow G$. Such a natural transformation is fully determined by its components

$$\alpha_\star : W(F) \rightarrow W(G) \quad \text{and} \quad \alpha_{(k,\ell)} : B_{(k,\ell)}(F) \rightarrow B_{(k,\ell)}(G) \quad \text{for all } k, \ell \in \mathbb{N}$$

satisfying naturality, i.e. commutativity of all diagrams of the form

$$\begin{array}{ccc} B_{k,\ell}(F) & \xrightarrow{\text{in}_i} & W(F) \\ \downarrow \alpha_{k,\ell} & & \downarrow \alpha_{k,\ell} \\ B_{k,\ell}(G) & \xrightarrow{\text{in}_i} & W(G) \end{array} \qquad \begin{array}{ccc} B_{k,\ell}(F) & \xrightarrow{\text{out}_j} & W(F) \\ \downarrow \alpha_{k,\ell} & & \downarrow \alpha_{k,\ell} \\ B_{k,\ell}(G) & \xrightarrow{\text{out}_j} & W(G). \end{array}$$

In other words, every natural transformation is a structure-preserving map between hypergraphs, i.e. if box f is incident to wire A in its i th input in the hypergraph F , then the same applies to their images with respect to α in the hypergraph G .

A hypergraph can contain an infinite set of wires and boxes. In the following we mainly restrict to *finite hypergraphs*, i.e. functors $G : \mathbf{I} \rightarrow \mathbf{Set}$ where $W(G)$ and

$$B(G) := \coprod_{k, \ell \in \mathbb{N}} B_{k, \ell}(G).$$

are finite sets. We denote the corresponding full subcategory of **Hyp** by **FinHyp**.

4.2 gs-Monoidal String Diagrams and Free Markov Categories

The pictorial representation of hypergraphs indicates their use for modeling causal structures in a categorical framework. In this subsection, we construct the free Markov category associated to a fixed hypergraph Σ . This is a Markov category in which the morphisms are string diagrams formed out of the boxes in Σ , and we argue that these morphisms can be used as generalized causal models.

However, three apparent problems make hypergraphs not directly applicable to represent string diagrams:

1. General hypergraphs might contain loops, for example in the sense that an output wire of a box may directly feed back as an input.
2. While the splitting of a wire into two represents the copying of values and makes sense in any Markov category, the merging of wires as in Figure 1 does not make sense.
3. A hypergraph in itself does not include any information about which wires are inputs or outputs of the overall diagram.

We resolve these issues by restricting to acyclic and left monogamous hypergraphs and by representing gs-monoidal string diagrams by cospans thereof:

Definition 3. *Let Σ be a hypergraph. A gs-monoidal string diagram for Σ is a cospan in the slice category \mathbf{FinHyp}/Σ of the form*

$$\begin{array}{ccc} & G & \\ p \nearrow & & \nwarrow q \\ \underline{n} & & \underline{m} \end{array}$$

satisfying that:

- (i) G is acyclic, i.e. there is no family of wires $A_0, \dots, A_{k-1} \in W(G)$ and boxes $f_0, \dots, f_{k-1} \in B(G)$ such that

$$\text{in}(f_i, A_i) \geq 1 \quad \text{and} \quad \text{out}(f_i, A_{i+1}) \geq 1,$$

where the subscripts are modulo k .

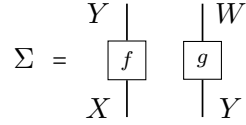
- (ii) G is left monogamous, i.e. for every wire $W \in W(G)$ we have

$$|p^{-1}(W)| + \sum_{f \in B(G)} \text{out}(f, W) \leq 1.$$

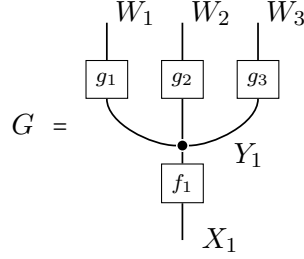
By abuse of notation, we also write G for the underlying hypergraph of the object G in \mathbf{FinHyp}/Σ , and we write $\text{type} : G \rightarrow \Sigma$ for the morphism that makes it into an object of \mathbf{FinHyp}/Σ .

In this definition, the *discrete hypergraph* \underline{n} is defined to have $W(\underline{n}) = \{1, \dots, n\}$ and contains no boxes. Thus the morphism $p : \underline{n} \rightarrow G$ simply equips some wires with labels from $1, \dots, n$, thereby telling us which wires of G are input wires of the overall diagram and in which order. The other cospan leg $q : \underline{m} \rightarrow G$ similarly encodes the m output wires.

Example 2. Consider the two hypergraphs



and



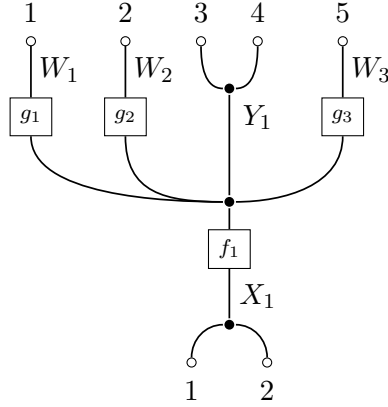
Note that we represent Σ by two distinct boxes instead of connecting them via the Y wire in order to highlight the purpose of Σ being the collection of elementary building blocks. The hypergraph G is an object of the slice category \mathbf{FinHyp}/Σ through the hypergraph morphism $\text{type} : G \rightarrow \Sigma$ that maps the wires X_1 to X , Y_1 to Y , and W_1, W_2, W_3 to W , and that maps the box f_1 to f , and the boxes g_1, g_2, g_3 to g .

We define a *gs-monoidal string diagram* as a cospan by specifying two discrete hypergraphs \underline{n} and \underline{m} together with mappings $p : \underline{n} \rightarrow W(G)$, $q : \underline{m} \rightarrow W(G)$ that specify which wires are connected to the input and to the output. Let us consider two different choices:

(i) If $\underline{n} = \{1, 2\}$ and $\underline{m} = \{1, 2, 3, 4, 5\}$ as well as

$$\begin{aligned} p : 1 &\mapsto X_1, 2 \mapsto X_1, \\ q : 1 &\mapsto W_1, 2 \mapsto W_2, 3 \mapsto Y_1, 4 \mapsto Y_1, 5 \mapsto W_3, \end{aligned}$$

then we get a cospan that corresponds to a diagram that could be drawn as

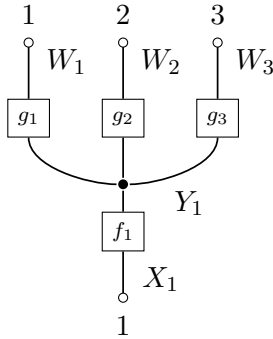


(ii) If $\underline{n} = \{1\}$ and $\underline{m} = \{1, 2, 3\}$ as well as

$$p : 1 \rightarrow X_1,$$

$$q : i \mapsto W_i,$$

then we get a cospan that corresponds to the gs -monoidal string diagram drawn as



Note that we have secretly used the convention to draw the discrete hypergraph \underline{n} (representing the input) on the bottom and \underline{m} (representing the output) on the top. Moreover, we used the convention that wires at the bottom of a box are input wires and wires on the top are output-wires.

Remark 4. Throughout the rest of the paper, we will omit the numbering of input and output based on the convention that inputs and outputs are numbered from left to right. Moreover, we will only label the boxes by their type, highlighting that the causal mechanisms are identical. For example, we will draw the string diagram Example 2 (ii) in simplified

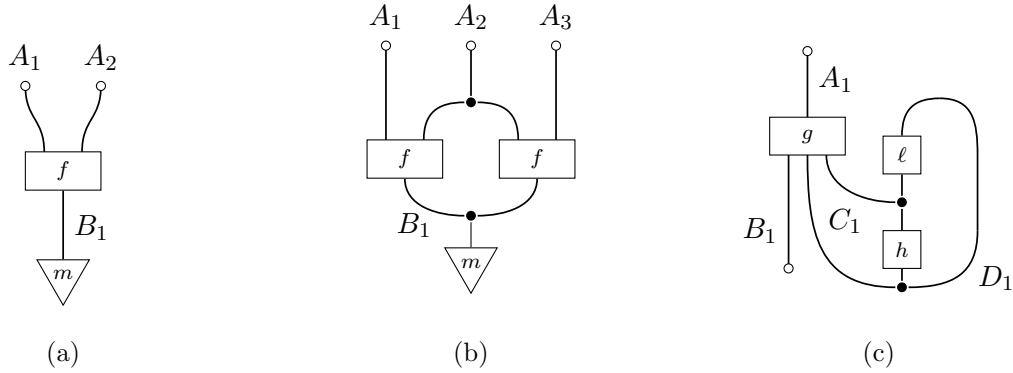
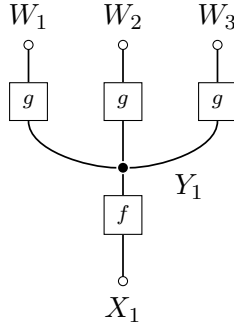


Figure 2: One example (a) and two non-examples (b), (c) of gs-monoidal string diagrams, where Σ is the hypergraph from Figure 1. The hypergraph in (b) is not left monogamous and (c) is not acyclic.

form as follows:



Moreover, this string diagram is also gs-monoidal since it is acyclic and left monogamous. In contrast, the string diagram in Example 2 (i) is not left monogamous since the wire X_1 does not arise from a unique global input.

For further examples and non-examples of gs-monoidal string diagrams, we refer to Figure 2.

Pictorially, an acyclic hypergraph does not contain a family of wires which form a loop. Further, left monogamy requires that every wire in the hypergraph arises as either a global input or as an output of a box in precisely one way, ensuring that no “merging” of wires occurs. See Figure 2 for an illustration of all of this.

The notion of gs-monoidal string diagram is the main ingredient for constructing a gs-monoidal category whose morphisms are freely generated by the wires and boxes in a fixed hypergraph Σ . Indeed we can now define the category \mathbf{FreeGS}_Σ as follows:

- (i) Objects are all hypergraph morphisms $\sigma : \underline{n} \rightarrow \Sigma$ for $n \in \mathbb{N}$, or equivalently finite sequences of wires in Σ .
- (ii) Morphisms are the isomorphism classes of gs-monoidal string diagrams.

Composition in \mathbf{FreeGS}_Σ is defined by a pushout which coincides with the way of stacking and connecting up drawings of string diagrams. The gs-monoidal structure likewise corresponds to the obvious operations in terms of string diagrams. We refer to Fritz and Liang (2022) for details. In the following, we will not distinguish between a gs-monoidal string diagram and its isomorphism class.

Although \mathbf{FreeGS}_Σ is a gs-monoidal category, it is typically not a Markov category. For example, the first step in the following simplification does not hold since the (cospans of) hypergraphs are not isomorphic, while the second equation does hold:

$$\begin{array}{c} \text{Diagram 1} \end{array} \neq \begin{array}{c} \text{Diagram 2} \end{array} = \begin{array}{c} \text{Diagram 3} \end{array} \quad (11)$$

In the following, we define the free Markov category $\mathbf{FreeMarkov}_\Sigma$ by taking a quotient of \mathbf{FreeGS}_Σ which enforces Equation (10), so that also the first equation above becomes true.

Definition 5. *Let*

$$\varphi := \begin{array}{ccc} & G & \\ p \nearrow & & \nwarrow q \\ \underline{n} & & \underline{m} \end{array} \quad (12)$$

be a gs-monoidal string diagram.

(i) A box $b \in B(G)$ is called *eliminable* if each output of b gets discarded, i.e. if for every $W \in W(G)$ such that $\text{out}(b, W) > 0$ we have

- (a) $q^{-1}(W) = \emptyset$.
- (b) $\text{in}(b', W) = 0$ for every box $b' \in B(G)$.

(ii) φ is called *normalized* if it contains no eliminable boxes.

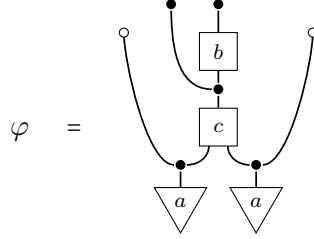
Every gs-monoidal string diagram has a normalized version obtained by iteratively applying the rule of Equation (10) to any eliminable box. Since every diagram is finite, this procedure terminates after finitely many steps, and we reach the normalized version. In addition, this diagram is unique since the order of elimination does not matter.

The free Markov category $\mathbf{FreeMarkov}_\Sigma$ is now defined just as \mathbf{FreeGS}_Σ , but with morphisms restricted to the normalized gs-monoidal string diagrams. The composition of morphisms is then defined as composition in \mathbf{FreeGS}_Σ followed by normalization since the composition of two normalized diagrams need not be normalized. See Fritz and Liang (2022) for details. In particular, any normalized gs-monoidal string diagram φ as in Equation (12) is a morphism of the form

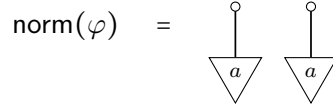
$$\varphi : \bigotimes_{i=1}^n \text{type}(p(i)) \longrightarrow \bigotimes_{j=1}^m \text{type}(q(j)) \quad (13)$$

in $\mathbf{FreeMarkov}_\Sigma$.

Example 3. *The morphism*



is not normalized since the output of b gets discarded. Applying Equation (10), also the output of c get discarded. Therefore the normalization of φ is



In general, normalizing a gs-monoidal string diagram defines a strict gs-monoidal functor

$$\text{norm} : \mathbf{FreeGS}_\Sigma \rightarrow \mathbf{FreeMarkov}_\Sigma$$

that is identity-on-objects.

4.3 Causal Models and Causal Compatibility

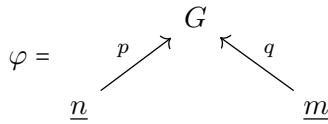
We now introduce the notion of a generalized causal model and define when a morphism in a Markov category is considered compatible with a generalized causal model.

Definition 6 (Fritz and Liang (2022, Definition 7.1)). *Given a hypergraph Σ , a generalized causal model is a normalized gs-monoidal string diagram (represented by Equation 12) over Σ such that q is injective.*

Intuitively, a generalized causal model is a morphism in $\mathbf{FreeMarkov}_\Sigma$ where the injectivity of q ensures that each wire is connected to at most one output. This lets us identify the global inputs and outputs with the wires in $W(G)$ (see Notation 10). In the traditional terminology of random variables, the injectivity of q guarantees that different outputs correspond to different variables. Figure 3 shows examples of generalized causal models.

One relevant subclass of generalized causal models are pure blooms. These morphisms represent causal models in which all variables are observed, i.e. every wire is an output in exactly one way, such as in Figure 3(c).

Definition 7 (Fritz and Liang (2022)). *Let φ be a generalized causal model represented by a gs-monoidal string diagram*



Then φ is called pure bloom if q is a bijection on wires.

Remark 8. *Bayesian networks are a strict subclass of generalized causal models. More precisely, a generalized causal model φ arises from a DAG if and only if*

- (i) φ is pure bloom (no latent variables),
- (ii) $\text{in}(\varphi) = \emptyset$ (no global inputs), and
- (iii) $|\text{out}(b)| = 1$ for every box $b \in B(G)$.

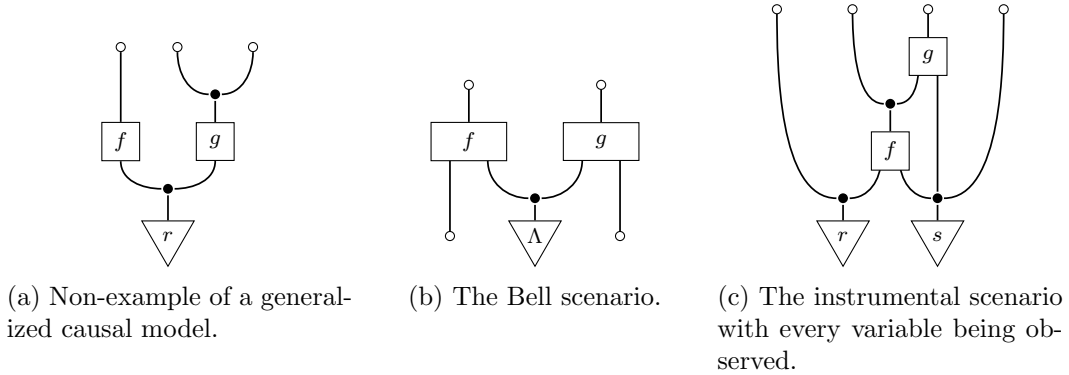


Figure 3: (Non-)Examples of generalized causal models. While (b), (c) are generalized causal models, the string diagram (a) is not since the output wire of g is connected to two global outputs. Concerning Definition 7, the generalized causal model (b) is not a pure bloom since the output of Λ is not connected to a global output. The generalized causal model (c) is a pure bloom without inputs. For a further analysis of (c) regarding d -separation, see Example 7(i). In all three examples, we have $\Sigma = G$ and assume type to be the identity map for simplicity.

For an example of a pure bloom morphism, we refer to Figure 3. We will show that the soundness of the d -separation criterion holds for arbitrary generalized causal models (see Corollary 26) while the completeness holds for pure bloom morphisms (see Theorem 28).

To define causal compatibility, we make the following assumption for the rest of the paper for convenience:

Assumption 9. *Throughout, \mathbf{C} is a strict Markov category.*

A monoidal category is strict if monoidal products like $A \otimes (B \otimes C)$ and $(A \otimes B) \otimes C$, or $B \otimes A$ and $A \otimes B$ are not only isomorphic but identical. We refer to Heunen and Vicary (2019, Section 1.3) for an introduction to strict monoidal categories.

Although most examples like **FinStoch**, **BorelStoch** or **Stoch** already fail strictness, this does not exclude these examples since we can always work with a strictification instead (Fritz, 2020, Theorem 10.17), which satisfies Assumption 9. On the other hand, our free Markov categories **FreeMarkov** $_{\Sigma}$ already satisfy this condition “on the nose”. In any case, Assumption 9 is a useful convenience that holds without loss of generality.

Notation 10. For the rest of the paper, we will assume that φ is a generalized causal model with

$$\varphi := \begin{array}{ccc} & G & \\ p \nearrow & & \nwarrow q \\ \underline{n} & & \underline{m} \end{array}$$

which becomes a cospan in \mathbf{FinHyp}/Σ through $\text{type} : G \rightarrow \Sigma$.

We identify inputs and outputs with the wires they map to under p and q and refer to them as such. In particular, we define

$$\text{in}(\varphi) := p(\underline{n}) \subseteq W(G) \quad (14)$$

$$\text{out}(\varphi) := q(\underline{m}) \subseteq W(G) \quad (15)$$

for the set of all input/output wires. If φ is a pure bloom morphism, then $\text{out}(\varphi) = W(G)$.

In the following, we present the notion of causal compatibility for a generalized causal model φ . Intuitively, a morphism f in any Markov category \mathbf{C} is compatible with φ if we can plug in a morphism from \mathbf{C} into every box in $B(\Sigma)$ in such a way that the composite is exactly f , and such that the global input and output wires of φ correspond to a given tensor factorization of the domain and codomain of f :

Definition 11 (Compatibility). For Σ a hypergraph, let φ be a generalized causal model as in (13). Let further

$$f : \bigotimes_{i=1}^n W'_i \rightarrow \bigotimes_{j=1}^m V'_j$$

be a morphism in any Markov category \mathbf{C} satisfying Assumption 9, equipped with a fixed tensor decomposition of its domain and codomain as indicated.

We call f compatible with φ if there exists a strict Markov functor³ $F : \mathbf{FreeMarkov}_\Sigma \rightarrow \mathbf{C}$ such that:

(i) We have

$$W'_i = F(\text{type}(p(i))), \quad V'_j = F(\text{type}(q(j))) \quad (16)$$

for all input indices $i = 1, \dots, n$ and output indices $j = 1, \dots, k$.

(ii) $f = F(\varphi)$.

This generalizes the functorial definition of causal models as first studied in Fong (2013).

Note that the functor F must assign to every wire type (i.e. wire in Σ) a corresponding object in the category \mathbf{C} . This implies that wires in $W(G)$ with identical types must map to the same object in \mathbf{C} . For example, one may consider a situation where f is a probability distribution with no inputs and all output variables are real-valued. In this case, we have $V'_j = \mathbb{R}$ for all j , and one may want to consider a causal model φ in which all wires are likewise of the same type. For the string diagrammatic picture of this definition, we refer to Section 2.2.

3. A strict Markov functor is a strict symmetric monoidal functor which preserves the comonoid structure.

Similarly, the hypergraph morphism $\text{type} : G \rightarrow \Sigma$ assigns to each box in G a specific “type” box in Σ . This means that under F , any two boxes with the same type must map to the same morphism in \mathbf{C} . This is why generalized causal models in our sense can naturally incorporate the condition that several causal mechanisms must be the same, namely when choosing the types in a way enforcing this. This has already been highlighted in our discussion of gs-monoidal string diagrams (see Example 2).

In the following, we denote for every wire $X \in W(G)$ in φ the corresponding object $F(\text{type}(X))$ in \mathbf{C} by X' . Similarly, for every set of wires $\mathcal{W} \subseteq W(G)$ in φ , we denote the corresponding multiset of wires in \mathbf{C} by \mathcal{W}' . For the rest of the paper, we will associate this multiset with the corresponding tensor product in \mathbf{C} obtained by tensoring its elements, where we ignore the question of how to order the factors.

5. Conditionals and Conditional Independence

In this section, we review a central assumption to prove the d -separation criterion, namely the existence of conditionals. Moreover, we introduce a string diagrammatic definition of conditional independence.

5.1 Existence of Conditionals

To use the d -separation criterion to detect causal compatibility, we need in addition the existence of conditionals. This notion has been studied in categorical terms in Cho and Jacobs (2019) in a special case, where the authors call it *admitting disintegration*, and subsequently in Fritz (2020) in general. In the following, we briefly review the definition following Fritz (2020, Section 11).

Definition 12. *Let \mathbf{C} be a Markov category. We say that \mathbf{C} has conditionals if for every morphism $f : \mathcal{A} \rightarrow \mathcal{X} \otimes \mathcal{Y}$, there is a morphism $f_{|\mathcal{X}}$ such that*

$$\begin{array}{c} \mathcal{X} \quad \mathcal{Y} \\ \diagdown \quad \diagup \\ \boxed{f} \\ | \\ \mathcal{A} \end{array} = \begin{array}{c} \mathcal{X} \quad \mathcal{Y} \\ \diagdown \quad \diagup \\ \boxed{f_{|\mathcal{X}}} \\ | \\ \mathcal{Y} \\ \diagdown \quad \diagup \\ \boxed{f} \\ | \\ \mathcal{A} \end{array}$$

Note that \mathcal{X}, \mathcal{Y} are single output spaces. However, since these can arise as products of multiple output spaces each, we use notation that suggests interpreting \mathcal{X} and \mathcal{Y} as multisets of spaces. Since Markov categories are symmetric monoidal, we can use the multisets or tensor product pictures interchangeably.

Examples of categories having conditionals are **FinStoch**, **Gauss** as well as **BorelStoch**. In contrast, **Stoch** does not have conditionals (see Fritz, 2020, Examples 11.6–11.8 and references therein).

In the following, we prove that the deterministic Markov category **FinSet** and the possibilistic Markov category **FinSetMulti** have conditionals.

Proposition 13. *FinSet and FinSetMulti have conditionals.*

Proof We show the statements separately, doing the case of single objects A , X and Y without loss of generality.

- (i) In **FinSet**, any map $f : A \rightarrow X \times Y$ is uniquely determined by its components $f_X : A \rightarrow X$ and $f_Y : A \rightarrow Y$, which are equivalently its marginals. Defining $f_{|X} : X \times A \rightarrow Y$ as $f_{|X}(x, a) := f_X(a)$ trivially shows the statement.⁴
- (ii) Let $f : A \rightarrow X \times Y$ be a morphism in **FinSetMulti** with $f(x, y|a) \in \{0, 1\}$ for $x \in X, y \in Y, a \in A$. Marginalizing over Y , we get

$$f_Y(x|a) = \sum_{y \in Y} f(x, y|a) = \begin{cases} 1 & \text{if } \exists y \in Y \text{ s.t. } f(x, y|a) = 1, \\ 0 & \text{else.} \end{cases}$$

where we have used the equation $1 + 1 = 1$. Defining $f(y|a, x) := f(x, y|a)$, we have

$$f(x, y|a) = f(y|a, x) \cdot f(x|a),$$

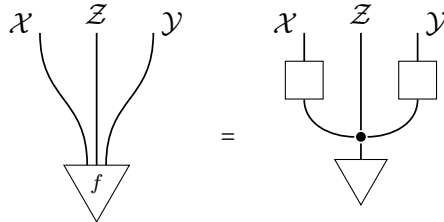
which shows the statement. ■

Note that the proof also generalizes trivially to **Set** and **SetMulti**.

5.2 Conditional Independence

A second ingredient of the d -separation criterion is the definition of conditional independence. The following definition has been introduced in several works (see for example Cho and Jacobs, 2019; Coecke and Spekkens, 2011) and shown to satisfy the well-known semigraphoid properties. In addition, Fritz (2020) shows that it is still meaningful to define conditional independence in the absence of conditionals.

Definition 14. *A morphism $f : I \rightarrow \mathcal{X} \otimes \mathcal{Z} \otimes \mathcal{Y}$ in \mathbf{C} displays the conditional independence $\mathcal{X} \perp \mathcal{Y} \mid \mathcal{Z}$ if it can be written as*



In other words, $\mathcal{X} \perp \mathcal{Y} \mid \mathcal{Z}$ holds if f is compatible with the generalized causal model which corresponds to the string diagram on the right-hand side (where all boxes are of distinct type, and we leave labels off for simplicity).

Remark 15. *The conditional independence $\mathcal{X} \perp \emptyset \mid \mathcal{Z}$ is equivalent to the existence of the conditional $r_{|\mathcal{Z}}$. Therefore, if \mathbf{C} has conditionals, then every state r satisfies $\mathcal{X} \perp \emptyset \mid \mathcal{Z}$ for every tensor factorization of its codomain $\mathcal{X} \otimes \mathcal{Z}$.*

⁴ The same argument shows that every cartesian monoidal category has conditionals.

While conditional independence coincides with the standard definition in probabilistic Markov categories like **FinStoch**, **Gauss** or **BorelStoch**, we show in the following an example of conditional independence for **FinSetMulti**.

Example 4. (i) Let $X = Y = Z$ be finite sets and $f(x, y, z) = \delta_{x,y,z}$, where $\delta_{x,y,z} = 1$ if and only if $x = y = z$. In other words, a triple of outcomes (x, y, z) is possible if and only if $x = y = z$. Then f displays the conditional independence $X \perp Y \mid Z$ since

$$f(x, y, z) = f(x \mid z) \cdot f(y \mid z) \cdot f(z),$$

with $f(z) = 1$ for every $z \in Z$ and $f(x \mid z) = \delta_{x,z}$ and $f(y \mid z) = \delta_{y,z}$. Note that in this representation $f(x \mid z)$ and $f(y \mid z)$ correspond to the left and the right boxes in the decomposition of Definition 14.

(ii) Let $X = Y = Z$ be finite sets and similarly $f(x, y, z) = \delta_{x,y} \cdot \delta_{z,z_0}$ for fixed $z_0 \in Z$, which has marginal $f_Z(z) = \delta_{z,z_0}$. Then f does not display the conditional independence $X \perp Y \mid Z$. Assuming the contrary, we would have

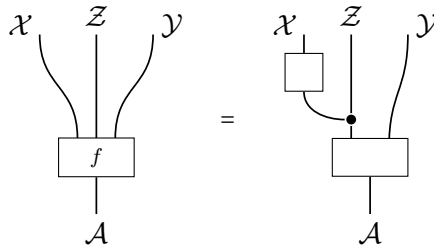
$$\delta_{x,y} = f(x \mid z) \cdot f(y \mid z)$$

which cannot be the case as soon as our sets have at least two elements.

As we have already done in the previous definition, we write $\mathcal{X}, \mathcal{Y}, \dots$ for arbitrary lists of objects in \mathbf{C} . We also allow for implicit reordering of these lists—effectively identifying these lists with multisets—and omit mention of the relevant compositions of f by swap morphisms.⁵ This allows us to talk about conditional independence with respect to any tripartition of the tensor factors in the codomain of any state f .

With this in mind, we now introduce a notion of conditional independence for morphisms with inputs. This notion is the key ingredient of the d -separation criterion presented in Section 6.2, and it is the categorical generalization of the *transitional conditional independence* introduced recently in Forré (2021, Definition 3.1).

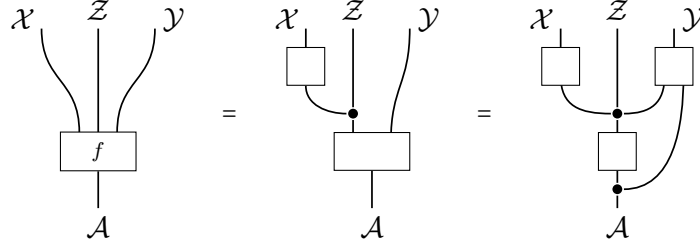
Definition 16. A morphism $f : \mathcal{A} \rightarrow \mathcal{X} \otimes \mathcal{Y} \otimes \mathcal{Z}$ in \mathbf{C} displays the conditional independence $\mathcal{X} \perp \mathcal{Y} \mid \mathcal{Z}$ if there exists a factorization of the form



Remark 17. Note that the above definition of conditional independence is not symmetric, i.e. $\mathcal{X} \perp \mathcal{Y} \mid \mathcal{Z}$ does not necessarily imply $\mathcal{Y} \perp \mathcal{X} \mid \mathcal{Z}$. If \mathbf{C} has conditionals then $\mathcal{X} \perp \mathcal{Y} \mid \mathcal{Z}$

5. Of course, these kinds of bookkeeping mechanisms are also present in the traditional notation of probability distributions and measures, though rarely made explicit in that context.

rewrites to



which highlights the asymmetry. Moreover, if \mathcal{A} is trivial, then the conditional independence coincides with Definition 14.

Due to the asymmetry, the outputs in \mathcal{Y} might contain information about the inputs \mathcal{A} which cannot be retrieved just from the outputs in \mathcal{Z} . On the other hand, the outputs in \mathcal{X} are generated using only the information from the outputs in \mathcal{Z} . The local Markov property that we will use in Definition 27 explicitly highlights this asymmetry: the output of a box (corresponding to \mathcal{X}) is independent of its non-descendants (\mathcal{Y}) given its inputs (\mathcal{Z}). Every global input is non-descendant of any box; however, not every global input wire is directly an input to the box itself.

Example 5. In the Markov category **Set**, a function $f : A \rightarrow X \times Z \times Y$ is always such that each one of its components f_X , f_Y and f_Z is a deterministic function of $a \in A$. Such f displays the conditional independence $X \perp Y \mid Z$ if and only if the X output can be rewritten as a function of the Z output only, i.e. $x = g(z)$, or equivalently if and only if f_X factors through f_Z .

6. Deciding Causal Compatibility with d -Separation

The main goal of this section is to prove that the d -separation criterion (Pearl, 2009, Section 1.2.3) correctly detects causal compatibility not just in discrete probability but in all Markov categories with conditionals. To this end, we introduce a categorical notion of d -separation phrased in terms of connectedness of the gs-monoidal string diagram representing the causal model. We then show that this notion coincides with the classical notion of d -separation whenever the latter applies.

This part is structured as follows. In Section 6.1, we introduce the categorical notion of d -separation on generalized causal models. Moreover, we show in Proposition 22 that this notion coincides with the classical notion of d -separation for all those generalized causal models that correspond to DAGs. In Section 6.2, we first show that d -separation implies conditional independences for compatible morphisms in Markov categories with conditionals. We then show in Theorem 28 that d -separation fully characterizes causal compatibility.

6.1 Categorical d -Separation

For a gs-monoidal string diagram

$$\varphi = \begin{array}{ccc} & G & \\ p \nearrow & & \nwarrow q \\ \underline{n} & & \underline{m} \end{array}$$

and a set of output wires $\mathcal{Z} \subseteq \text{out}(\varphi)$, we define a new gs-monoidal string diagram $\text{Cut}_{\mathcal{Z}}(\varphi)$ obtained by removing the wires in \mathcal{Z} in the following sense. Its underlying hypergraph H is such that the set of boxes is the same, $B(H) := B(G)$, while the set of wires is $W(H) := W(G) \setminus \mathcal{Z}$. Each box has the same input and output wires as before, except in that those in \mathcal{Z} are simply removed, which lowers the arities of the boxes correspondingly. We also remove all occurrences of wires in \mathcal{Z} from the global inputs and outputs, and this results in a gs-monoidal string diagram

$$\text{Cut}_{\mathcal{Z}}(\varphi) := \begin{array}{ccc} & H & \\ p' \nearrow & & \nwarrow q' \\ \underline{n}' & & \underline{m}' \end{array}$$

where $\underline{n}' \subseteq \underline{n}$, $\underline{m}' \subseteq \underline{m}$ and p' , q' are the restrictions of p and q , respectively. Note that $\text{Cut}_{\mathcal{Z}}(\varphi)$ is generally not a morphism in $\mathbf{FreeMarkov}_{\Sigma}$ anymore since it is not normalized. However, it can be understood as a morphism in \mathbf{FreeGS}_H . Example 6 will present a few examples.

We next introduce some notation for paths of wires.

Definition 18. *Let φ be a gs-monoidal string diagram in \mathbf{FreeGS}_{Σ} .*

(i) *An undirected path between two wires $X, Y \in W(G)$ is a sequence of wires*

$$X = W_1, W_2, \dots, W_n, W_{n+1} = Y$$

together with a sequence of boxes $b_1, \dots, b_n \in B(G)$ such that

$$\text{in}(b_i, W_i) + \text{out}(b_i, W_i) \geq 1 \quad \text{and} \quad \text{in}(b_i, W_{i+1}) + \text{out}(b_i, W_{i+1}) \geq 1.$$

If there exists an undirected path between X and Y , then we write $X - Y$.

(ii) *For two wires $A, B \in W(G)$, we write $A \rightarrow B$ if there exists a box $b \in B(G)$ such that*

$$\text{in}(b, A) = 1 \quad \text{and} \quad \text{out}(b, B) = 1. \quad (17)$$

(iii) *For two wires $A, B \in W(G)$, we write $A \twoheadrightarrow B$ if there exists a sequence of wires $W_1, \dots, W_n \in W(G)$ such that*

$$A \rightarrow W_1 \rightarrow \dots \rightarrow W_n \rightarrow B. \quad (18)$$

Thus, an undirected path in φ may traverse a box not just from input or output or vice versa, but also from input to input or output to output.

The intuitive idea behind the following definition of d -separation, as already briefly discussed in Fritz and Liang (2022, Remark 7.2), was communicated to us by Rob Spekkens.

Definition 19 (Categorical d -separation). *Let φ be a generalized causal model. For three disjoint sets of output wires $\mathcal{X}, \mathcal{Y}, \mathcal{Z} \subseteq \text{out}(\varphi)$, we say that \mathcal{Z} d -separates \mathcal{X} and \mathcal{Y} if*

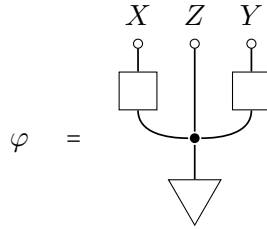
$$\text{Cut}_{\mathcal{Z}}(\varphi_{\mathcal{X}, \mathcal{Y}, \mathcal{Z}})$$

has no undirected path between any output in \mathcal{X} and any output in \mathcal{Y} .

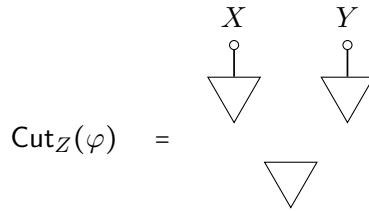
Here, $\varphi_{\mathcal{W}} := \text{norm}(\text{del}_{\mathcal{W}^c} \circ \varphi)$ with $\mathcal{W}^c = \text{out}(\varphi) \setminus \mathcal{W}$ denotes the marginal on \mathcal{W} in $\mathbf{FreeMarkov}_{\Sigma}$. The absence of an undirected path as in the definition manifests itself in the string diagrams simply as topological disconnectedness.

Example 6. *The following examples constitute the basic components of “classical” d-separation and illustrate the simplicity of categorical d-separation.⁶ In all cases, the unlabeled boxes denote distinct generators, i.e. distinct boxes in the generating hypergraph Σ .*

(i) *Fork: consider the morphism*

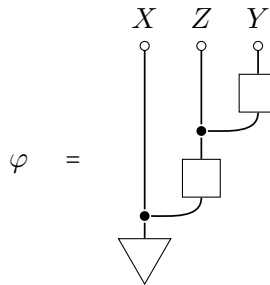


Then Z d-separates X from Y since

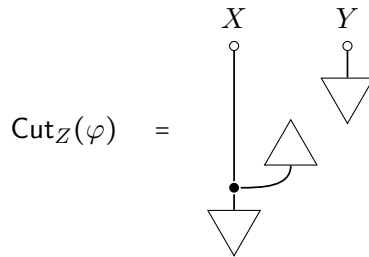


has disconnected X and Y.

(ii) *Chain: consider the morphism*



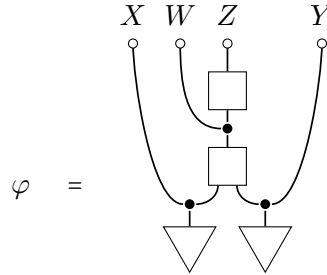
Then Z d-separates X from Y since



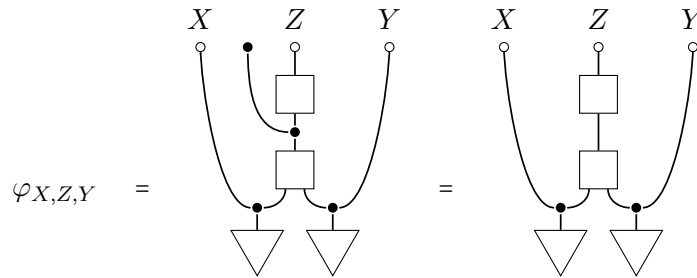
6. See Jacobs and Zanasi (2017) for an account of chains, forks and colliders featuring both numerical examples and a categorical formalism.

has disconnected X and Y .

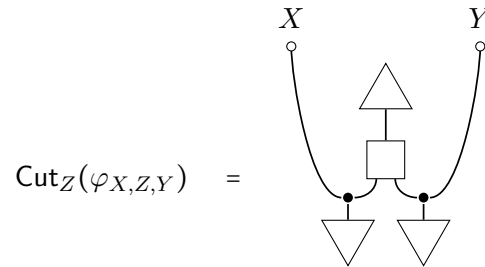
(iii) Collider: consider the morphism



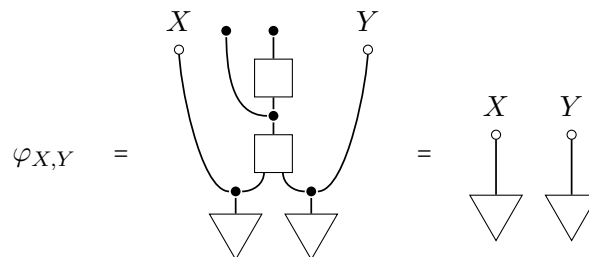
Then Z does not d -separate X from Y since we have



and therefore

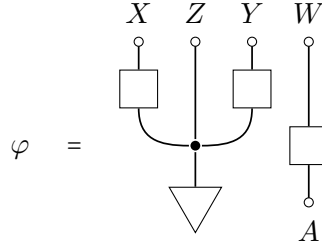


which still contains an undirected path $X - Y$. The same reasoning applies when $\mathcal{Z} = \{W\}$ or $\mathcal{Z} = \{W, Z\}$. However, if $\mathcal{Z} = \emptyset$, then

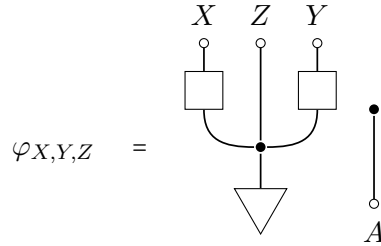


which has disconnected X and Y . Therefore \emptyset d -separates X and Y .

(iv) Consider the morphism

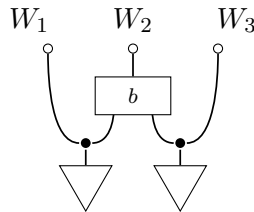


This morphism differs from (i) by adding an independent process $A \rightarrow W$. The normalized marginal $\varphi_{X,Y,Z}$ is given by



which again shows that Z d -separates X from Y since cutting Z makes X and Y disconnected.

In order to define the classical notion of d -separation, we note that certain gs-monoidal string diagrams have an underlying DAG, given by using wires as nodes and taking the edges to be \rightarrow as in Definition 18. We use the term *DAG path* to refer to an undirected path in this DAG, i.e. to a sequence of wires connected by boxes from input to output or vice versa (but *not* from input to input or output to output). For example, in the string diagram



there are two undirected paths witnessing $W_1 - W_3$, namely on the one hand the direct path W_1, W_3 and on the other hand the indirect W_1, W_2, W_3 . Only the second is a DAG path since its two steps are input to output and output to input, while going from W_1 to W_3 directly would be input of b to input of b . We also define the ancestor wires of a given set of wires as

$$\text{An}(\mathcal{X}) = \{U \in W(G) : \exists X \in \mathcal{X} \text{ such that } U \twoheadrightarrow X\}$$

and the set of descendant wires as

$$\text{Dec}(\mathcal{X}) = \{U \in W(G) : \exists X \in \mathcal{X} \text{ such that } X \twoheadrightarrow U\},$$

where $A \twoheadrightarrow B$ is defined in Definition 18(iii). Note that $\mathcal{X} \subseteq \text{An}(\mathcal{X}), \text{Dec}(\mathcal{X})$. To state the following classical definition (Pearl, 2009, Definition 1.2.3) in our language, we restrict further to those generalized causal models that are determined by their underlying DAGs.

As noted in Remark 8, a causal structure φ is represented by a DAG, if the string diagram is pure bloom, every box has exactly one output, and it has no global inputs, i.e. $\text{in}(\varphi) = \emptyset$.

Definition 20 (Classical d -separation). *Let φ be a pure bloom causal model with $\text{in}(\varphi) = \emptyset$ and such that every box has exactly one output. Then:*

- (a) *A DAG path p in φ is called d -separated by a set of wires $\mathcal{Z} \subseteq \text{out}(\varphi)$ if at least one of the following two conditions holds:

 - (i) p contains a chain $W \rightarrow Z \rightarrow U$ or a fork $W \leftarrow Z \rightarrow U$ for some $Z \in \mathcal{Z}$.
 - (ii) p contains a collider $W \rightarrow M \leftarrow U$ where $M \notin \text{An}(\mathcal{Z})$.*
- (b) \mathcal{X} is d -separated from \mathcal{Y} by \mathcal{Z} if every DAG path between every $X \in \mathcal{X}$ and $Y \in \mathcal{Y}$ is d -separated by \mathcal{Z} .

We will now prove the equivalence of categorical d -separation with classical d -separation for the class of causal models on which the latter is defined. This requires some preparation.

Lemma 21. *Let φ be a pure bloom causal model, $b \in B(G)$ a box in φ and $\mathcal{W} \subseteq \text{out}(\varphi)$ a subset of its wires. The following statements are equivalent:*

- (i) $\text{out}(b) \cap \text{An}(\mathcal{W}) = \emptyset$.
- (ii) b gets discarded in $\varphi_{\mathcal{W}} = \text{norm}(\text{del}_{\mathcal{W}^c} \circ \varphi)$.

Proof (ii) \implies (i): To prove the contrapositive, assume $\exists A \in \text{out}(b)$ such that $A \in \text{An}(\mathcal{W})$. Then there is a path $A \rightarrow W$ with $W \in \mathcal{W}$. Since W is still an overall output that does not get discarded, this path is still valid in $\text{del}_{\mathcal{W}^c} \circ \varphi$. Therefore b remains in $\text{norm}(\text{del}_{\mathcal{W}^c} \circ \varphi)$.

(i) \implies (ii): Consider the set $\text{Dec}(\text{out}(b))$. By assumption, we have that $\text{Dec}(\text{out}(b)) \cap \text{An}(\mathcal{W}) = \emptyset$. We show that the box b gets discarded in $\text{norm}(\text{del}_{\text{Dec}(\text{out}(b))} \circ \varphi)$, which is enough because of $\mathcal{W}^c \supseteq \text{Dec}(\text{out}(b))$. By definition of $\text{Dec}(\text{out}(b))$, there exists a final box \hat{b} such that $\text{out}(\hat{b}) \subseteq \text{Dec}(\text{out}(b))$.⁷ This shows that \hat{b} gets discarded in $\text{norm}(\text{del}_{\text{Dec}(\text{out}(\hat{b}))} \circ \varphi)$.

Define $\tilde{\varphi} := \text{norm}(\text{del}_{\text{Dec}(\text{out}(\hat{b}))} \circ \varphi)$. Repeating the above procedure with $\tilde{\varphi}$, we arrive after finite number of steps at \hat{b} itself being a final box. Since it is then eliminable after composing with $\text{del}_{\text{Dec}(\text{out}(b))}$, it indeed no longer appears in the normalization. \blacksquare

We now show the promised equivalence result between categorical d -separation and classical d -separation in the cases where φ represents a causal structure given by a DAG, i.e. $\text{in}(\varphi) = \emptyset$ and every box has a single output.

Proposition 22. *Both concepts of d -separation coincide on pure bloom causal models φ with $\text{in}(\varphi) = \emptyset$ and in which every box has exactly one output.*

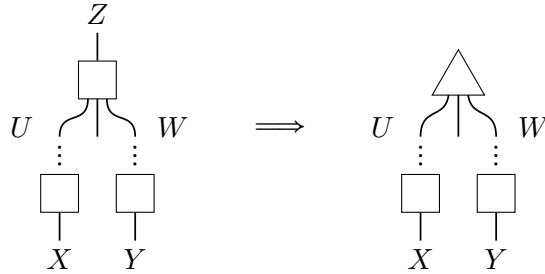
7. As defined in Fritz and Liang (2022), a final box is one whose outputs are global outputs of φ without further copy or discard. Such a box always exists since φ is pure bloom and normalized (compare Fritz and Liang, 2022, Lemma 4.6).

Proof To make the proof more intuitive, we introduce the term *d-connected* as the negation of *d-separated* (in either version).

We start by showing that classical *d-connectedness* implies categorical *d-connectedness*. Let p be a DAG path between some $X \in \mathcal{X}$ and some $Y \in \mathcal{Y}$ which witnesses that \mathcal{Z} makes \mathcal{X} and \mathcal{Y} be *d-connected* in the classical sense, which means that the following hold:

- (i) For every chain $W \rightarrow M \rightarrow U$ or fork $W \leftarrow M \rightarrow U$ in p , we have $M \notin \mathcal{Z}$.
- (ii) For every collider $W \rightarrow M \leftarrow U$ in p , we have $M \in \text{An}(\mathcal{Z})$.

For simplicity, we also assume without loss of generality that p contains only one wire from \mathcal{X} and \mathcal{Y} each, say X and Y respectively. Then this p can also be interpreted as an undirected path in φ , but generally not in $\varphi_{\text{cut}} := \text{Cut}_{\mathcal{Z}}(\varphi_{\mathcal{X}, \mathcal{Y}, \mathcal{Z}})$ since it may traverse wires that are not in φ_{cut} . However, we now show that there still is an undirected path p' between X and Y in φ_{cut} . Since p is *d-connected*, if p contains a wire $Z \in \mathcal{Z}$, then it has to arise from a collider $U \rightarrow Z \leftarrow W$ in p . Removing wire Z from p still defines a valid *undirected* path between X and Y , pictorially:



We prove that the path p' obtained by removing all wires in \mathcal{Z} from p like this is an undirected path in φ_{cut} , which implies categorical *d-connectedness*. To this end, it only remains to show that each wire in p' is an existing wire in φ_{cut} , which we do as follows:

- (i) X and Y themselves are still in φ_{cut} .
- (ii) Every $Z \in \mathcal{Z}$ in p is part of a collider $U \rightarrow Z \leftarrow W$ as above, so that $U, W \in \text{An}(\mathcal{Z})$. This implies that U and W survive in $\varphi_{\mathcal{X}, \mathcal{Y}, \mathcal{Z}}$ by Lemma 21.
- (iii) Since U and W are themselves either the middle node in a chain or fork or the start or end of p , we have $U, W \notin \mathcal{Z}$. This implies $U, W \in \text{An}(\mathcal{Z}) \setminus \mathcal{Z}$, and therefore U and W survive also in φ_{cut} .
- (iv) For every chain $W \rightarrow M \rightarrow U$ in p , if U survives in φ_{cut} , then so does M (since it survives in $\varphi_{\mathcal{X}, \mathcal{Y}, \mathcal{Z}}$ and $M \notin \mathcal{Z}$).
- (v) For every fork $W \leftarrow M \rightarrow U$ in p , if U or W survives in φ_{cut} , then so does M (since it survives in $\varphi_{\mathcal{X}, \mathcal{Y}, \mathcal{Z}}$ and $M \notin \mathcal{Z}$).

Since the wires in p' are exactly those of p minus some of the colliders, we can start with the first two observations and then apply the latter two repeatedly on any segment bounded by colliders or the starting node X or the final node Y in order to conclude that all wires in p' are present in φ_{cut} . This concludes one direction of the proof.

The converse direction of showing that categorical d -connectedness implies classical d -connectedness works similarly. Let p be an undirected path between $X \in \mathcal{X}$ and $Y \in \mathcal{Y}$ in φ_{cut} . We assume without loss of generality that all wires in p are distinct. Furthermore, we also assume without loss of generality that p is of the form, in terms of the notation from Definition 18(iii),

$$X \leftarrow A - B \rightarrow Y, \quad (19)$$

where every wire that is in between A and B is not contained in $\text{An}(\mathcal{X})$ or $\text{An}(\mathcal{Y})$, or equivalently that every wire in p that is also in $\text{An}(\mathcal{X})$ is directly reached from X by output-to-input traversals in p , and similarly for all wires in $\text{An}(\mathcal{Y})$. This property can be achieved by taking every wire in p which is additionally in $\text{An}(\mathcal{X})$ and replace it with the path from X to it by a sequence of output-to-input traversals, and similarly for every wire in $\text{An}(\mathcal{Y})$. Note that this replacement may involve changing the starting and ending wires X and Y as well.

In order to turn p into a DAG path p' that witnesses classical d -separation, we need to remove all direct input-to-input traversals of a box in p ; direct output-to-output traversals cannot occur due to the assumption that every box has exactly one output. We can hence simply add to p the unique output wire of every box that has an input-to-input traversal in p , and we obtain a valid DAG path p' .

It remains to verify the conditions on chains, forks and colliders. Clearly, p' does not contain any chain $W \rightarrow Z \rightarrow U$ or fork $W \leftarrow Z \rightarrow U$ with $Z \in \mathcal{Z}$ since such a configuration cannot occur in p to begin with. For a collider $W \rightarrow M \leftarrow U$, the unique box which outputs M must be contained in $\varphi_{\mathcal{X}, \mathcal{Y}, \mathcal{Z}}$, and therefore be in $\text{An}(\mathcal{X} \cup \mathcal{Y} \cup \mathcal{Z})$ by Lemma 21. However, M being in $\text{An}(\mathcal{X})$ or $\text{An}(\mathcal{Y})$ violates the assumption that p is of the form (19). Therefore M has to be in $\text{An}(\mathcal{Z})$, showing the collider condition (ii). \blacksquare

We record one more observation on categorical d -separation for further use below.

Lemma 23. *Let φ be a pure bloom causal model and $\mathcal{X}, \mathcal{Y}, \mathcal{Z} \subseteq \text{out}(\varphi)$ a partition of all output wires such that \mathcal{Z} categorically d -separates \mathcal{X} and \mathcal{Y} . Then every box $b \in B(G)$ in φ satisfies at least one of the cases:*

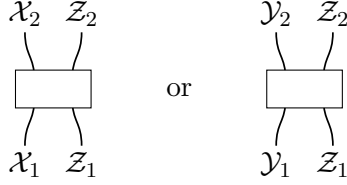
$$(i) \text{in}(b), \text{out}(b) \subseteq \mathcal{X} \cup \mathcal{Z}.$$

$$(ii) \text{in}(b), \text{out}(b) \subseteq \mathcal{Y} \cup \mathcal{Z}.$$

Proof If there exist $Y \in \mathcal{Y} \cap \text{out}(b)$ and $X \in \mathcal{X} \cap \text{out}(b)$, then these wires are still in the output of b in φ_{cut} , and this contradicts the assumed disconnectedness of φ_{cut} with respect to \mathcal{X} and \mathcal{Y} . Since $\mathcal{X}, \mathcal{Y}, \mathcal{Z}$ form a partition, this shows that either $\text{out}(b) \subseteq \mathcal{X} \cup \mathcal{Z}$ or $\text{out}(b) \subseteq \mathcal{Y} \cup \mathcal{Z}$.

Since φ is a pure bloom we have that $\text{in}(b) \subseteq \text{out}(\varphi)$ which guarantees that $\text{in}(b) \subseteq \mathcal{X} \cup \mathcal{Y} \cup \mathcal{Z}$. Proving $\text{out}(b) \subseteq \mathcal{X} \cup \mathcal{Z} \Rightarrow \text{in}(b) \subseteq \mathcal{X} \cup \mathcal{Z}$ and $\text{out}(b) \subseteq \mathcal{Y} \cup \mathcal{Z} \Rightarrow \text{in}(b) \subseteq \mathcal{Y} \cup \mathcal{Z}$ works similarly to the first part of the proof, and this then proves the statement. \blacksquare

Pictorially, Lemma 23 shows that if \mathcal{Z} d -separates \mathcal{X} and \mathcal{Y} , then every box b in φ is of the form



where $\mathcal{X}_i \subseteq \mathcal{X}$, $\mathcal{Y}_i \subseteq \mathcal{Y}$, $\mathcal{Z}_i \subseteq \mathcal{Z}$.

6.2 Causal Compatibility

In the following, we show that d -separation implies conditional independence for any generalized causal model. We first prove this result for a partition of wires in a pure bloom causal model in Lemma 24. We then refine it to any disjoint collection of wires in Corollary 26 in any generalized causal model. Finally, we show in Theorem 28 that d -separation fully characterizes causal compatibility for pure bloom causal models in all Markov categories with conditionals.

Throughout, we also use the following convenient notation: If a morphism f in \mathbf{C} is compatible with a causal model φ in the sense of Definition 11, then we refer to the wires of φ to indicate conditional independence instead of the objects in the tensor factorization of f . In other words, instead of writing $\mathcal{X}' \perp \mathcal{Y}' \mid \mathcal{Z}'$, we simply write $\mathcal{X} \perp \mathcal{Y} \mid \mathcal{Z}$. Here, each $W' = F(\text{type}(W))$ is the object in \mathbf{C} associated with the wire W by the causal model functor F (see Definition 11).

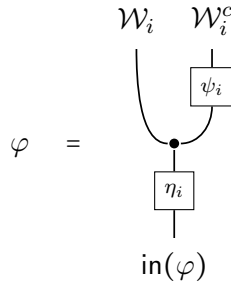
Lemma 24. *Let \mathbf{C} be a strict Markov category with conditionals, and let φ be a pure bloom causal model. Further, let $\mathcal{X}, \mathcal{Y}, \mathcal{Z} \subseteq \text{out}(\varphi)$ be a partition of wires in φ such that $\text{in}(\varphi) \subseteq \mathcal{Y} \cup \mathcal{Z}$ and \mathcal{X} and \mathcal{Y} are d -separated by \mathcal{Z} .*

If a morphism f in \mathbf{C} is compatible with φ , then $\mathcal{X} \perp \mathcal{Y} \mid \mathcal{Z}$ (as in Definition 16).

Proof Choose a total ordering of all boxes $b_1, \dots, b_{k-1} \in B(G)$ and a chain of sets of wires in $\text{out}(\varphi)$,

$$\text{in}(\varphi) = \mathcal{W}_1 \subseteq \dots \subseteq \mathcal{W}_k = \text{out}(\varphi),$$

such that $\mathcal{W}_{i+1} = \text{out}(b_i) \cup \mathcal{W}_i$ and $\text{An}(\mathcal{W}_i) = \mathcal{W}_i$. Note that there is a factorization



in $\mathbf{FreeMarkov}_\Sigma$, where η_i is again a pure bloom and η_1 is an identity morphism. The existence of such a chain of sets follows easily by induction on the number of boxes based on the existence of a final box (Fritz and Liang, 2022, Lemma 4.6).⁸

8. It is also worth noting that for causal models which correspond to DAGs, this statement amounts to the standard fact that every DAG can be refined to a total order.

Then for every $i \in \{1, \dots, k\}$, we show the existence of a decomposition⁹

$$f = \begin{array}{c} \text{---} (\mathcal{W}_i^c)' \text{---} \\ | \\ \boxed{F(\psi_i)} \\ | \\ \text{---} (\mathcal{X} \cap \mathcal{W}_i)' \text{---} \quad \text{---} (\mathcal{Z} \cap \mathcal{W}_i)' \text{---} \quad \text{---} (\mathcal{Y} \cap \mathcal{W}_i)' \text{---} \\ | \quad | \quad | \\ \boxed{k_i} \quad \bullet \quad \bullet \\ | \quad | \quad | \\ \text{---} \bullet \text{---} \quad \text{---} \bullet \text{---} \quad \text{---} \bullet \text{---} \\ | \\ \boxed{h_i} \\ | \\ \text{in}(\varphi)' \end{array} \tag{20}$$

Since $\mathcal{W}_k^c = \emptyset$, setting $i = k$ proves the desired statement.

We prove this stronger claim by induction on i . The start of the induction at $i = 1$ is trivial since η_1 is the identity and therefore

$$f = \begin{array}{c} \text{---} (\mathcal{W}_1^c)' \text{---} \\ | \\ \boxed{F(\psi_1)} \\ | \\ \text{---} (\mathcal{Z} \cap \mathcal{W}_1)' \text{---} \quad \text{---} (\mathcal{Y} \cap \mathcal{W}_1)' \text{---} \\ | \quad | \\ \bullet \quad \bullet \\ | \quad | \\ \text{---} \bullet \text{---} \\ | \\ \text{in}(\varphi)' \end{array}$$

since $\mathcal{X} \cap \mathcal{W}_1 = \mathcal{X} \cap \text{in}(\varphi) = \emptyset$. For the induction step, we prove the statement at $i + 1$. Since φ is pure bloom and since $\text{An}(\mathcal{W}_{i+1}) = \mathcal{W}_{i+1}$, we can peel off the box b_i with $\mathcal{W}_{i+1} = \text{out}(b_i) \cup \mathcal{W}_i$

9. Note that we assume, without loss of generality, that every box in the string diagram depends on every variable in the collection of wires. For example, k_i depends on every variable in $\mathcal{Z} \cap \mathcal{W}_i$. If this is not the case, say, k_i does not depend on $W \in \mathcal{Z} \cap \mathcal{W}_i$, we replace it by $k'_i = k_i \circ (\text{id} \otimes \text{del}_W)$.

from ψ_i , so as to achieve the decomposition

$$f = \begin{array}{c} \begin{array}{c} (\mathcal{W}_{i+1}^c)' \\ \downarrow \\ F(\psi_{i+1}) \\ \downarrow \\ \text{out}(b_i)' \\ \downarrow \\ F(b_i) \\ \downarrow \\ (\mathcal{X} \cap \mathcal{W}_i)' \quad (\mathcal{Z} \cap \mathcal{W}_i)' \quad (\mathcal{Y} \cap \mathcal{W}_i)' \\ \downarrow \\ k_i \\ \downarrow \\ h_i \\ \downarrow \\ \text{in}(\varphi)' \end{array} \end{array} \quad (21)$$

where we have used the induction assumption to obtain a decomposition as in the lower half, and the dashed wires indicate that only some of them may be present since the inputs of b_i are an unspecified subset of \mathcal{W}_i .

By Lemma 23 we have to distinguish two cases:

- (i) $\text{in}(b_i), \text{out}(b_i) \subseteq \mathcal{X} \cup \mathcal{Z}$. Then, the third dashed wire in the above decomposition of f is not needed, and we consider the morphism

$$g := \begin{array}{c} \begin{array}{c} (\text{out}(b_i) \cap \mathcal{X})' \quad (\text{out}(b_i) \cap \mathcal{Z})' \quad (\mathcal{X} \cap \mathcal{W}_i)' \quad (\mathcal{Z} \cap \mathcal{W}_i)' \\ \downarrow \\ F(b_i) \\ \downarrow \\ k_i \\ \downarrow \\ (\mathcal{Z} \cap \mathcal{W}_i)' \end{array} \end{array}$$

which is part of that decomposition. By the existence of conditionals, we can rewrite g in the form

$$g = \begin{array}{c} \begin{array}{c} (\mathcal{X} \cap \mathcal{W}_{i+1})' \quad (\mathcal{Z} \cap \mathcal{W}_{i+1})' \\ \downarrow \\ r_i \\ \downarrow \\ \ell_i \\ \downarrow \\ (\mathcal{Z} \cap \mathcal{W}_i)' \end{array} \end{array} = \begin{array}{c} \begin{array}{c} (\mathcal{X} \cap \mathcal{W}_{i+1})' \quad (\mathcal{Z} \cap \mathcal{W}_{i+1})' \\ \downarrow \\ r_i \\ \downarrow \\ \ell_i \\ \downarrow \\ (\mathcal{Z} \cap \mathcal{W}_i)' \end{array} \end{array}$$

where both lower boxes can be refined with internal structure consisting of carrying $(\mathcal{Z} \cap \mathcal{W}_i)'$ forward on a separate wire, but this internal structure is not relevant for the remainder of the proof. Substituting this form of g into Equation (21), i.e. replacing the morphism k_i there with the box r_i here and merging the box ℓ_i here with h_i there, proves the induction step.

- (ii) $\text{in}(b_i), \text{out}(b_i) \subseteq \mathcal{Y} \cup \mathcal{Z}$. Then, the first dashed wire in the above decomposition of f is not needed, and we can merge $F(b_i)$ with h_i , which shows the statement. ■

We will now generalize Lemma 24 to all generalized causal models and to arbitrary disjoint sets \mathcal{X}, \mathcal{Y} and \mathcal{Z} which do not necessarily partition the set of all wires.

Lemma 25. *Let φ be a generalized causal model and $\mathcal{X}, \mathcal{Y}, \mathcal{Z} \subseteq \text{out}(\varphi)$ a tripartition of output wires in φ such that $\text{in}(\varphi) \subseteq \mathcal{Y} \cup \mathcal{Z}$ and such that \mathcal{Z} categorically d -separates \mathcal{X} and \mathcal{Y} . Then there exists a tripartition of wires $\tilde{\mathcal{X}} \supseteq \mathcal{X}, \tilde{\mathcal{Y}} \supseteq \mathcal{Y}, \mathcal{Z}$ in the pure bloom version $\varphi_{\text{pure-bloom}}$ of φ such that¹⁰*

$$\mathcal{Z} \text{ } d\text{-separates } \tilde{\mathcal{X}} \text{ and } \tilde{\mathcal{Y}} \quad \text{in } \varphi_{\text{pure-bloom}}.$$

Proof With $\varphi_{\text{cut}} := \text{Cut}_{\mathcal{Z}}(\varphi_{\text{pure-bloom}})$, define

$$\tilde{\mathcal{X}} := \{U \in \text{out}(\varphi_{\text{cut}}) : \exists X \in \mathcal{X} : X - U \text{ in } \varphi_{\text{cut}}\} \supseteq \mathcal{X}$$

to be the connected component of \mathcal{X} in φ_{cut} , and

$$\tilde{\mathcal{Y}} := \text{out}(\varphi_{\text{pure-bloom}}) \setminus (\tilde{\mathcal{X}} \cup \mathcal{Z}) \supseteq \mathcal{Y}.$$

By definition, $\tilde{\mathcal{X}}, \tilde{\mathcal{Y}}, \mathcal{Z}$ form a tripartition of wires in $\varphi_{\text{pure-bloom}}$. Moreover, $\tilde{\mathcal{X}}$ and $\tilde{\mathcal{Y}}$ are categorically d -separated by \mathcal{Z} since any path in $\text{Cut}_{\mathcal{Z}}(\varphi_{\text{pure-bloom}})$ is a valid path in $\text{Cut}_{\mathcal{Z}}(\varphi)$ and vice versa. ■

Corollary 26. *Let \mathbf{C} be a strict Markov category with conditionals, and let φ be a generalized causal model.¹¹ Further, let $\mathcal{X}, \mathcal{Y}, \mathcal{Z} \subseteq \text{out}(\varphi)$ be disjoint sets of output wires in φ such that $\text{in}(\varphi) \subseteq \mathcal{Y} \cup \mathcal{Z}$ and \mathcal{X} and \mathcal{Y} are d -separated by \mathcal{Z} . If f is compatible with φ , then $\mathcal{X} \perp \mathcal{Y} \mid \mathcal{Z}$.*

In this statement, we use another standard convention: when the disjoint sets $\mathcal{X}, \mathcal{Y}, \mathcal{Z}$ do not partition the set of wires of φ , then the conditional independence $\mathcal{X} \perp \mathcal{Y} \mid \mathcal{Z}$ is to be understood as Definition 16 applied to the corresponding marginal

$$f_{\mathcal{X}', \mathcal{Y}', \mathcal{Z}'} = \text{del}_{(\mathcal{X}' \cup \mathcal{Y}' \cup \mathcal{Z}')^c} \circ f$$

10. The pure-bloom version $\varphi_{\text{pure-bloom}}$ is obtained by copying each wire so to make it into an output. It is part of the bloom-circuitry factorization of Fritz and Liang (2022).

11. In this situation φ does not need to be pure bloom.

rather than to f itself.

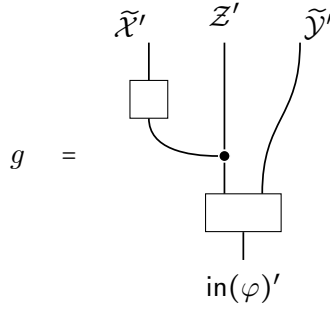
Proof We prove this statement by reducing it to the case of pure bloom causal models treated in Lemma 24.

Consider the restricted causal model $\psi := \varphi_{\mathcal{X}, \mathcal{Y}, \mathcal{Z}}$. We have that $f_{\mathcal{X}', \mathcal{Y}', \mathcal{Z}'} = F(\psi)$ since F is a Markov functor. By the definition of categorical d -separation, \mathcal{Z} d -separates \mathcal{X} and \mathcal{Y} also in ψ . Let $\psi_{\text{pure-bloom}}$ be the pure bloom version of ψ . Since $f_{\mathcal{X}', \mathcal{Y}', \mathcal{Z}'}$ is compatible with ψ , we can extend it to a pure bloom version

$$g := F(\psi_{\text{pure-bloom}})$$

of which $f_{\mathcal{X}', \mathcal{Y}', \mathcal{Z}'}$ is a marginal.

By Lemma 25, for $\psi_{\text{pure-bloom}}$ there is a tripartition of output wires $\tilde{\mathcal{X}} \supseteq \mathcal{X}, \tilde{\mathcal{Y}} \supseteq \mathcal{Y}, \mathcal{Z}$ such that \mathcal{Z} d -separates $\tilde{\mathcal{X}}$ and $\tilde{\mathcal{Y}}$. Since \mathbf{C} has conditionals, Lemma 24 provides us with a decomposition of the form



By marginalizing over $\tilde{\mathcal{X}} \setminus \mathcal{X}', \tilde{\mathcal{Y}} \setminus \mathcal{Y}'$ in g , we obtain the desired conditional independence for the marginal $f_{\mathcal{X}', \mathcal{Y}', \mathcal{Z}'}$. ■

Note that this result includes the soundness of the classical d -separation criterion in the classical case of discrete random variables in Bayesian networks.¹² Here, it is obtained upon restricting to pure bloom causal models with $\text{in}(\varphi) = \emptyset$, the Markov category **FinStoch**, and every box having precisely one output since then conditional independence reduces to Definition 14 by Remark 17.

Definition 27. Let φ be a generalized causal model and f a morphism in a strict Markov category \mathbf{C} (not necessarily having conditionals). Then we say that f satisfies:

- (i) the global Markov property with respect to φ if for every three disjoint sets of outputs $\mathcal{X}, \mathcal{Y}, \mathcal{Z} \subseteq \text{out}(\varphi)$ with $\text{in}(\varphi) \subseteq \mathcal{Y} \cup \mathcal{Z}$:

$$\mathcal{X} \text{ and } \mathcal{Y} \text{ are categorically } d\text{-separated by } \mathcal{Z} \text{ in } \varphi \implies \mathcal{X} \perp \mathcal{Y} \mid \mathcal{Z} \text{ in } f.$$

- (ii) the local Markov property with respect to φ if for every box b in φ , we have

$$\text{out}(b) \perp \text{Dec}(\text{out}(b))^c \setminus \text{in}(b) \mid \text{in}(b) \text{ in } f.$$

¹² See Verma and Pearl (1990) for the original proof and Pearl (2009, Theorem 1.2.5(i)) for a textbook account.

Note that in the special case of φ arising from a DAG, the local and global Markov property agree with the classical definitions of Markov properties (Pearl, 2009, Theorem 1.2.6 and Theorem 1.2.7).

Theorem 28. *Suppose that we are given the following:*

- \mathbf{C} is a strict Markov category with conditionals.
- φ is a pure bloom causal model over a hypergraph Σ such that the boxes in φ have distinct types in Σ .
- $f : \otimes_{i=1}^n W_i' \rightarrow \otimes_{j=1}^m V_j'$ is a morphism in \mathbf{C} .

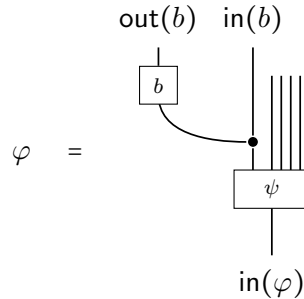
Then the following statements are equivalent:

- (i) f is compatible with the causal model φ .
- (ii) f satisfies the global Markov property.
- (iii) f satisfies the local Markov property.

Proof (i) \implies (ii): The global Markov property is precisely the statement of Corollary 26.

(ii) \implies (iii): This follows from the fact that $\text{Dec}(\text{out}(b))^c$ and $\text{out}(b)$ are d -separated by $\text{in}(b)$, and $\text{in}(\varphi) \subseteq \text{Dec}(\text{out}(b))^c$, which makes the global Markov property specialize to the local one.

(iii) \implies (i): We prove this statement by induction over the number of boxes $k := |B(G)|$. The case $k = 1$ is trivial. For the step from k to $k + 1$, let b be a final box in φ , which means that $\text{Dec}(\text{out}(b)) = \text{out}(b)$. Then, φ factorizes as



where ψ is another causal model satisfying all of our assumptions, and no box in ψ has the same type in Σ as b does.

In order to construct a functor F as in Definition 11, note first that it must satisfy (16), which already lets us write the domain of f as $\text{in}(\varphi)'$, and similarly for the codomain. Since f satisfies the local Markov property with respect to b , we can decompose f by Definition 16

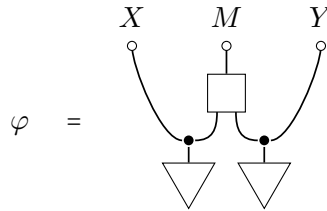
as

(22)

By induction hypothesis, we have that g is compatible with ψ since g satisfies the local Markov properties specified by ψ . Since the box b appears only once in φ , we can freely define the action of the functor F on b as $F(b) := h$. Then, we obtain

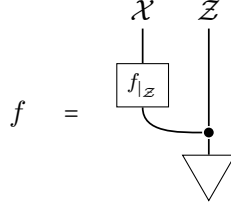
where we use in the first step Equation (22) and in the last the fact that F is a Markov functor. ■

Remark 29. (i) Note that we have used the assumption that \mathbf{C} has conditionals only for the implication (i) \implies (ii). Therefore, for an arbitrary strict Markov category, the global as well as the local Markov property is a sufficient condition for the compatibility of a morphism with a generalized causal model (satisfying our assumptions). However, these Markov properties implicitly require the existence of certain conditionals. Consider, for example, the generalized causal model



where all boxes are of distinct types. Choosing $\mathcal{X} = \{M\}$, $\mathcal{Y} = \emptyset$ and $\mathcal{Z} = \{X, Y\}$, a morphism f satisfying the global Markov property displays in particular the conditional

independence $\{M\} \perp \emptyset \mid Z$, pictorially:

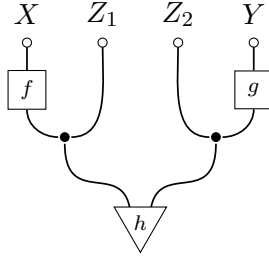


This shows that the conditional $f_{|Z}$ exists, and this recovers the box that outputs M (up to almost sure equality).

- (ii) Theorem 28 shows that d -separation correctly detects causal compatibility for the Markov categories **FinStoch**, **Gauss** or **BorelStoch**. For the Markov category **Stoch**, which does not have conditionals, the global and local Markov properties are at least sufficient for the compatibility since our proof of these implications has not used conditionals.
- (iii) Note that Theorem 28 only applies to causal models where each box appears at most once in the model (which in particular implies that φ has no nontrivial symmetries). However, the implication (i) \implies (ii) applies to arbitrary generalized causal models as proven in Corollary 26.

Example 7. We now present three examples that go beyond the classical d -separation criterion. In (i) we study a causal structure that does not arise from a DAG, in (ii) we study a DAG causal structure with continuous and possibilistic variables, and in (iii) we study a causal structure with inputs with deterministic variables.

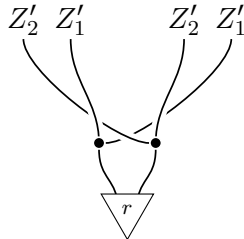
- (i) Let φ be the causal structure



and let \mathbf{C} be a strict Markov category with conditionals. By Theorem 28, a morphism $t: I \rightarrow X' \otimes Z'_1 \otimes Z'_2 \otimes Y'$ in \mathbf{C} is compatible with this structure if and only if it satisfies

$$X \perp \{Y, Z_2\} \mid Z_1 \quad \text{and} \quad Y \perp \{X, Z_1\} \mid Z_2$$

For a general class of examples, consider $X' = Z'_2$ and $Y' = Z'_1$ and any morphism in \mathbf{C} of the form



We claim that such a distribution is compatible with φ if and only if there exist morphisms d and d' such that

$$\text{Diagram 1} = \text{Diagram 2} = \text{Diagram 3} \quad (23)$$

where s is the first marginal of r , and similarly $d' : Z'_2 \rightarrow Z'_1$ satisfies the same equations the other way around. Here, the second equation states that the morphism d is *s*-a.s. deterministic (Fritz, 2020, Definition 13.11), and similarly for d' .

Indeed, assuming compatibility we have that

$$\text{Diagram 1} = \text{Diagram 2} \quad (24)$$

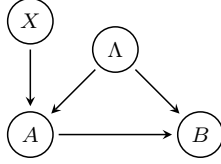
which shows the first equality in Equation (23). For the second equality we have that

where we have used the conditional independence $Z_2 \perp X \mid Z_1$ in the first step and Equation (24) in the second step. Since the morphism is symmetric with respect to permutations of the output wires X and Z_2 , we have $a = F(f)$ *s*-a.s. which shows the second equality in Equation (23). Proving the existence of d' works analogously by interchanging the roles of X and Y as well as Z_1 and Z_2 .

Conversely, we have

where we have used the assumption that d is *s*-a.s. deterministic in the second equation. Repeating this calculation interchanging the roles of Z_1 and Z_2 as well as X and Y shows the statement.

(ii) Consider the instrumental scenario, given by the DAG



This has been previously studied mainly in the context of DAGs with latent variables (Pearl, 1995; Bonet, 2001). For our analysis, we assume each variable to be observed, which means that the causal structure reads string diagrammatically as

$$\varphi = \tag{25}$$

There are two non-trivial d -separations:

- (a) Between X and B by $\{A, \Lambda\}$,
- (b) Between X and Λ .

Therefore, Theorem 28 implies that a distribution P on a four-fold tensor product object in a Markov category with conditionals is compatible with φ if and only if $X \perp B \mid A, \Lambda$ and $X \perp \Lambda$.

In **BorelStoch**, this means that P is compatible with φ if and only if

$$\begin{aligned} &P(X \in E_1, A \in E_2, B \in E_3, \Lambda \in E_4) \\ &= \int_{E_2} \int_{E_4} P_{X|A,\Lambda}(X \in E_1 | a, \lambda) P_{B|A,\Lambda}(B \in E_3 | a, \lambda) P_{A,\Lambda}(da, d\lambda) \end{aligned}$$

and

$$P(X \in E_1, \Lambda \in E_4) = P(X \in E_1) \cdot P(\Lambda \in E_4)$$

where E_i are measurable sets in the Borel σ -algebras of the spaces X', A', B' and Λ' .

For simplicity, assume that all random variables take values in \mathbb{R} and are absolutely continuous, i.e. there exists a density $f : X' \times A' \times B' \times \Lambda' \rightarrow [0, \infty)$ such that

$$P(X \in E_1, A \in E_2, B \in E_3, \Lambda \in E_4) = \int_{E_1 \times E_2 \times E_3 \times E_4} f(x, a, b, \lambda) dx da db d\lambda$$

The causal compatibility now amounts to the following two conditions:

(a) $X \perp \Lambda$, i.e.

$$f_{X,\Lambda}(x, \lambda) = f_X(x) \cdot f_\Lambda(\lambda) \quad a.e. \quad (26)$$

where a.e. means almost everywhere with respect to the Lebesgue measure on \mathbb{R} .

(b) $X \perp B \mid A, \Lambda$, i.e.

$$f(x, a, b, \lambda) = f_{X|A,\Lambda}(x, a, \lambda) \cdot f_{A,\Lambda}(a, \lambda) \cdot f_{B|A,\Lambda}(b, a, \lambda) \quad a.e., \quad (27)$$

where the conditional densities are defined implicitly by

$$f_{A|X,\Lambda}(a|x, \lambda) \cdot f_{X,\Lambda}(x, \lambda) = f_{X,A,\Lambda}(x, a, \lambda) \quad a.e.$$

Combining Eq. (26) and Eq. (27) results in

$$f(x, a, b, \lambda) = f_\Lambda(\lambda) \cdot f_X(x) \cdot f_{A|X,\Lambda}(a, x, \lambda) \cdot f_{B|A,\Lambda}(b, a, \lambda) \quad a.e.$$

which is the usual factorization condition for compatibility with the causal structure in (25).

In **FinSetMulti**, the same observation implies that a morphism

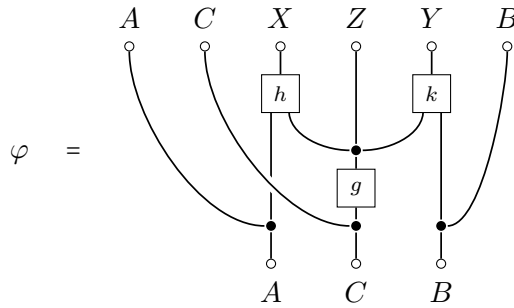
$$f : I \rightarrow X' \otimes A' \otimes B' \otimes \Lambda'$$

is compatible with the causal structure φ if and only if the possibility of X and Y can be independently determined, knowing the outcomes of A and Λ , i.e.

$$f(x, y|a, \lambda) = f(x|a, \lambda) \cdot f(y|a, \lambda)$$

and the possibilities of X and Λ are independent, i.e. $f(x, \lambda) = f(x) \cdot f(\lambda)$.

(iii) Consider the following string diagram



A morphism f in **FinSet** is compatible with φ if the variables $z \in Z, x \in X$ and $y \in Y$ and arise via functions $g : C \rightarrow Z$, $h : A \times Z \rightarrow X$, and $k : B \times Z \rightarrow Y$, i.e.

$$z = g(c), \quad x = h(a, z) \quad \text{and} \quad y = k(b, z).$$

By Theorem 28, f is compatible with φ if the following conditions are satisfied (due to the local Markov property):

- (a) $Z \perp \{A, B\} \mid C$
- (b) $X \perp \{B, C\} \mid \{A, Z\}$
- (c) $Y \perp \{A, C\} \mid \{B, Z\}$

By Example 5, Condition (a) implies the existence of a function g such that $z = g(c)$, Condition (b) implies $x = h(a, z)$, and Condition (c) implies $y = k(a, z)$. But this directly shows the compatibility of f with the causal model φ . The local Markov property immediately implies causal compatibility in **FinSet** since the construction of conditionals (see Proposition 13) is trivial.

Question 30. Can Theorem 28 be extended to more general causal models? In particular, what about allowing the same box to appear several times in φ ?

Acknowledgments

We thank Rob Spekkens for the original idea behind the definition of categorical d -separation and Patrick Forré for discussion and pointers to the literature. AK acknowledges support from the Stand Alone project P33122-N of the Austrian Science Fund (FWF) and funding of the Austrian Academy of Sciences (ÖAW) through a DOC scholarship.

References

- John C. Baez and Mike Stay. Physics, topology, logic and computation: A rosetta stone. *New Structures for Physics*, pages 95–172, 2010. doi: 10.1007/978-3-642-12821-9_2.
- Filippo Bonchi, Fabio Gadducci, Aleks Kissinger, Paweł Sobociński, and Fabio Zanasi. Rewriting modulo symmetric monoidal structure. In *Symposium on Logic in Computer Science*, pages 710–719, 2016. doi: 10.1145/2933575.2935316.
- Blai Bonet. Instrumentality tests revisited. In *Conference on Uncertainty in Artificial Intelligence*, pages 48–55. Morgan Kaufmann Publishers Inc., 2001. ISBN 1558608001. doi: 10.5555/2074022.2074029.
- Kenta Cho and Bart Jacobs. Disintegration and Bayesian inversion via string diagrams. *Mathematical Structures in Computer Science*, 29:938–971, 2019. ISSN 09601295. doi: 10.1017/S0960129518000488.
- Bob Coecke. Quantum picturalism. *Contemporary Physics*, 51(1):59–83, 2009. doi: 10.1080/00107510903257624.
- Bob Coecke and Robert W. Spekkens. Picturing classical and quantum Bayesian inference. *Synthese*, 186(3):651–696, 2011. doi: 10.1007/s11229-011-9917-5.
- Andrea Corradini and Fabio Gadducci. An algebraic presentation of term graphs, via gsmoidal categories. *Applied Categorical Structures*, 7(4):299–331, 1999. doi: 10.1023/A:100864741750.

- Brendan Fong. *Causal Theories: A Categorical Perspective on Bayesian Networks*. Master's thesis, University of Oxford, 2013. Available electronically via <http://arxiv.org/abs/1301.6201>.
- Patrick Forré. Transitional conditional independence. *arXiv:2104.11547*, 2021.
- Patrick Forré and Joris M. Mooij. Markov properties for graphical models with cycles and latent variables. *arXiv:1710.08775*, 2017.
- Tobias Fritz. A synthetic approach to Markov kernels, conditional independence and theorems on sufficient statistics. *Advances in Mathematics*, 370:107239, 2020. ISSN 10902082. doi: 10.1016/j.aim.2020.107239.
- Tobias Fritz and Wendong Liang. Free gs-monoidal categories and free Markov categories. *arXiv:2204.02284*, 2022.
- Fabio Gadducci. *On the Algebraic Approach to Concurrent Term Rewriting*. PhD thesis, University of Pisa, 1996. Available electronically via <https://citeseerx.ist.psu.edu/pdf/1277313dd2c4fce640780b7472221156579e392f>.
- Dichuan Gao. *Functorial Causal Models: Towards an Algebraic Approach to Causal Inference*. Bachelor's thesis, Brown University, 2022. Available electronically via https://www.brown.edu/academics/math/sites/math/files/Gao%20Dichuan_0.pdf.
- Chris Heunen and Jamie Vicary. *Categories for Quantum Theory : An Introduction*. Oxford University Press, 2019. ISBN 9780198739616.
- Bart Jacobs and Fabio Zanasi. A formal semantics of influence in Bayesian reasoning. In *International Symposium on Mathematical Foundations of Computer Science*, volume 83, pages 1–14. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2017. doi: 10.4230/LIPIcs.MFCS.2017.21.
- Bart Jacobs, Aleks Kissinger, and Fabio Zanasi. Causal inference by string diagram surgery. In *International Conference on Foundations of Software Science and Computation Structures*, pages 313–329. Springer, 2019. ISBN 9783030171261. doi: 10.1007/978-3-030-17127-8_18.
- Steffen L. Lauritzen. *Graphical Models*. Oxford University Press, 1996. ISBN 978-0198522195.
- Steffen L. Lauritzen, Philip Dawid, Birgitte N. Larsen, and Hanns-Georg Leimer. Independence properties of directed Markov fields. *Networks*, 20(5):491–505, 1990. doi: 10.1002/net.3230200503.
- Tom Leinster. *Basic Category Theory*. Cambridge University Press, 2014. ISBN 9781107360068. doi: 10.1017/CBO9781107360068.
- Aleksandar Milosavljevic and Fabio Zanasi. String diagram rewriting modulo commutative monoid structure. *arXiv:2204.04274*, 2022.

- Judea Pearl. On the testability of causal models with latent and instrumental variables. In *Conference on Uncertainty in Artificial Intelligence*, pages 435–443. Morgan Kaufmann Publishers Inc., 1995. ISBN 1558603859. doi: 10.5555/2074158.2074208.
- Judea Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, 2009. ISBN 9780511803161. doi: 10.1017/CBO9780511803161. Second edition.
- Paolo Perrone. Notes on category theory with examples from basic mathematics. *Lecture Notes*, 2019. Available electronically via <http://arxiv.org/abs/1912.10642>.
- Martin L. Puterman. *Markov Decision Processes*. Wiley, 1994. ISBN 9780471619772. doi: 10.1002/9780470316887.
- Eigil F. Rischel. The category theory of causal models. Master’s thesis, University of Copenhagen, 2020. Available electronically via <https://www.erischel.com/documents/mscthesis.pdf>.
- Eigil F. Rischel and Sebastian Weichwald. Compositional abstraction error and a category of causal models. In *Conference on Uncertainty in Artificial Intelligence*, volume 161, pages 1013–1023. MLResearchPress, 2021. Available electronically via <https://proceedings.mlr.press/v161/rischel21a.html>.
- Thomas Verma and Judea Pearl. Causal networks: Semantics and expressiveness. In *Conference on Uncertainty in Artificial Intelligence*, volume 9, pages 69–76. Morgan Kaufmann Publishers, 1990. doi: 10.1016/B978-0-444-88650-7.50011-1.