A Categorical Approach to DIBI Models

Tao Gu
University College London, UK

Jialu Bao
Cornell University, Ithaca, NY, USA

Justin Hsu
Cornell University, Ithaca, NY, USA

Alexandra Silva
Cornell University, Ithaca, NY, USA

Fabio Zanasi
University College London, UK
University of Bologna, OLAS team (INRIA), Italy

Abstract

The logic of Dependence and Independence Bunched Implications (DIBI) is a logic to reason about conditional independence (CI); for instance, DIBI formulas can characterise CI in discrete probability distributions and in relational databases, using a probabilistic DIBI model and a similarly-constructed relational model. Despite the similarity of the two models, there lacks a uniform account. As a result, the laborious case-by-case verification of the frame conditions required for constructing new models hinders them from generalising the results to CI in other useful models such that continuous distribution. In this paper, we develop an abstract framework for systematically constructing DIBI models, using category theory as the unifying mathematical language. We show that DIBI models arise from arbitrary symmetric monoidal categories with copy-discard structure. In particular, we use string diagrams – a graphical presentation of monoidal categories – to give a uniform definition of the parallel composition and subkernel relation in DIBI models. Our approach not only generalises known models, but also yields new models of interest and reduces properties of DIBI models to structures in the underlying categories. Furthermore, our categorical framework enables a comparison between string diagrammatic approaches to CI in the literature and a logical notion of CI, defined in terms of the satisfaction of specific DIBI formulas. We show that the logical notion is an extension of string diagrammatic CI under reasonable conditions.

2012 ACM Subject Classification Theory of computation → Logic; Theory of computation → Semantics and reasoning; Theory of computation → Models of computation

Keywords and phrases Conditional Independence, Dependence Independence Bunched Implications, String Diagrams, Markov Categories

Digital Object Identifier 10.4230/LIPIcs.FSCD.2024.17

Funding *Tao Gu*: Tao Gu is supported by the UK EPSRC grant EP/S013008/1.
Justin Hsu: Justin Hsu is partially supported by NSF grant #194313.
Alexandra Silva: Silva was partially supported by ERC grant Autoprobe (no. 101002697).
Fabio Zanasi: Fabio Zanasi acknowledges support from EPSRC EP/V002376/1 and MIUR P2022HXNSC (PRIN 2022 PNRR – Next Generation EU).

Acknowledgements We thank the anonymous reviewers for their close reading and detailed feedback.

1 Introduction

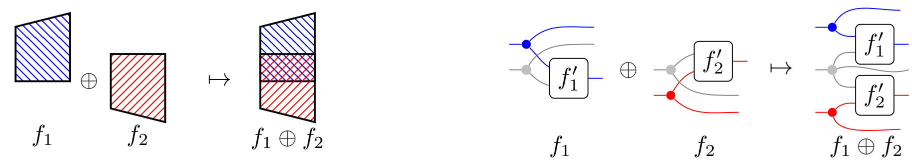
Conditional independence (CI) is a fundamental concept that can be traced back to the pioneer work on probabilities in Bayes [6] and Laplace [25]. In modern days, this notion is formalised and applied across various fields of science. For instance, CI is a central concept in the first formal definition of secrecy by Shannon [37], and remains crucial in various subsequent works in cryptography [26, 30]; graphical models leverage CI relations to have efficient representations of probabilistic phenomenons [32, 21, 13]. The core idea of CI is straightforward: events A and B are "independent" when information about one does not convey information about the other; events A and B are "conditionally independent" given event C if, with knowledge of event C, events A and B become independent. Albeit intuitive, reasoning about conditional independence is intricate, leading to extensive research aimed at formalising such reasoning [32, 14].

For probabilistic programs, an extension of standard programs with constructs to sample from distributions, formal methods for (conditional) independence have emerged as powerful tools for program verification. For instance, Barthe et al. [5] introduced Probabilistic Separation Logic (PSL) to formalise several cryptography protocols, where the independence of variables guarantees no leakage of information and thus security of the algorithms. A follow-up work from Bao et al. [4] proposed the logic of *Dependence and Independence Bunched Implications* (DIBI), which enhances PSL with the ability to reason about *conditional* independence. Syntactically, DIBI extends the logic of Bunched Implications (BI) [28, 34], which is the assertion logic underpinning Separation Logic (SL) [35] and PSL, with a non-commutative conjunction $\fatsemi$ and its adjoints. Semantically, as in BI, the separating conjunction $*$ is interpreted through a partial operation $\oplus$ on states, regarded as the parallel composition. In addition, they define a sequential composition $\odot$ to interpret $P \fatsemi Q$. Informally, $P * Q$ says that P and Q hold in states that can be separated, and $P \fatsemi Q$ expresses a possible dependency of Q on P. Section 3 will review the logic in more details.

Bao et al. [4] introduced two kinds of semantic models for DIBI logic: the probabilistic DIBI models for reasoning about CI of variables in discrete probabilistic computation, and the relational DIBI models for expressing the CI notion in relational databases called join dependency. These two models are defined analogously, yielding similar conditions for one to laboriously check to ensure that they are models. Such similarity led the authors to conjecture a family of categorical DIBI models that induce these concrete models as instances.

We believe that such categorical models would facilitate the construction of new models and set out to solve the conjecture with a simple observation: in both the probabilistic and relational DIBI models, the states resemble *Markov kernels* – maps from input elements to distributions/powersets over output elements. Such DIBI states can be identified categorically as morphisms in the Kleisli categories associated to the discrete distribution monad $\mathcal{D}$ (Definition 36) or the nonempty powerset monad $\mathcal{P}_i$ (Definition 37). However, giving a categorical definition for the parallel compositions $\oplus$ is difficult. The previous work [4] gives Figure 1a as a pictorial intuition for the parallel composition. The states are drawn as trapezoids, with the short and long vertical sides representing the input and output domains, respectively. There, given a blue map f_1 and a red map f_2, their parallel composition $f_1 \oplus f_2$ takes as input the union of their inputs. Then, each f_i takes its counterpart in the combined input domain and generates an output. Finally these two outputs are combined to be the output of $f_1 \oplus f_2$. This parallel composition *is partial* because the combination of their outputs is allowed only when the variables overlap in particular ways. This creates a challenge to capture DIBI models categorically because, in a categorical setting, the domains and codomains of DIBI states are objects, and it is not obvious how to define the overlap of objects.

Our solution stems from a formalisation of this graphical intuition through *string diagrams*, a pictorial formalism for monoidal categories. String diagrams are widely adopted as intuitive yet mathematically rigorous reasoning tools across different areas of science, see [33] for



(a) Parallel composition depicted in [4]. **(b)** A diagrammatic representation.

Figure 1 Intuition for parallel composition.

an overview. We formalise the trapezoids intuition in Figure 1a into string diagrams in Figure 1b. The maps previously embodied as trapezoids now have a fork shape, with some branches being straight lines and some other branches going through boxes. The boxes represent arbitrary morphisms in the underlying category, and the straight lines represent the identity morphisms. Whereas composition of two DIBI states were hand-waved as two trapezoids tiled together in Figure 1a, the string diagram defines it precisely: the overlap of the two trapezoids is witnessed by the grey wires, and the composition joins two diagrams side-by-side with the grey wires shared. We show in Section 4 that this string diagram representation yields DIBI models in any category with enough structure to interpret , namely, Markov categories [9, 12]. We then derive existing and new concrete DIBI models as instances in Section 5.

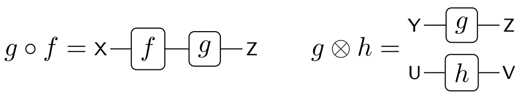
This framework also enables a comparison between different characterisations of conditional independence (CI). While Bao et al. [4] show that probabilistic or relational CI are both captured by some DIBI formulas, it is unclear if these formulas generalise to CI in other models and how they compare to other abstract notions of CI. Since we can construct categorical DIBI models based on any Markov categories, we define a logical notion of CI for morphisms in Markov categories as satisfaction of those DIBI formulas. In Section 6, we investigate the relationship between our "logical" CI and various CI notions based on categorical structures from literature in synthetic statistics [9, 12] and identify the conditions that make them equivalent.

Throughout the paper we fix a countably infinite set of variables Var, use $x, y, z, \ldots$ for elements of Var, and use $W, X, Y, \ldots$ for finite subsets of Var.

2 Category Theory Preliminaries

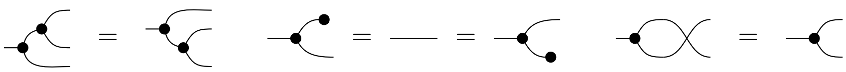
Unless specified, all monoidal categories we consider are strict and we write $dom(f)$ and $cod(f)$ for the domain and codomain of any morphism f. We write $\langle \mathbb{C}, \otimes, \mathsf{I} \rangle$ for a (strict) monoidal category, where $\otimes$ is the monoidal product and I the unit object of $\mathbb{C}$. If it is also symmetric, we write $\sigma_{\mathsf{A},\mathsf{B}} \colon \mathsf{A} \otimes \mathsf{B} \to \mathsf{B} \otimes \mathsf{A}$ for the symmetry natural transformation indexed by objects A and B.

As detailed for instance in [36, 33, 11], morphisms of symmetric monoidal categories have a graphical presentation as string diagrams, where sequential composition and monoidal product are depicted as concatenation and juxtaposition of diagrams, respectively: given morphisms $f \colon \mathsf{X} \to \mathsf{Y}$, $g \colon \mathsf{Y} \to \mathsf{Z}$, $h \colon \mathsf{U} \to \mathsf{V}$,

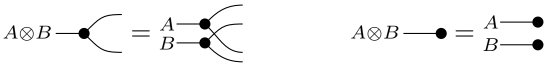


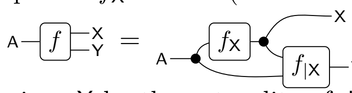
We read string diagrams from left to right, and tensor products from top to bottom. Object labels in the diagrams are omitted when they are evident or irrelevant to the context. Symmetries are indicated with the string diagram . We call string diagrams consisting solely of combinations of s *rewirings*: intuitively, they permute the order of the objects.

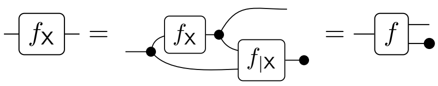
We use the notion of a Markov category, which suitably generalises categories of probabilistic processes [12]. First, a *copy-delete category* (*CD category*) is a symmetric monoidal category (SMC) $\langle \mathbb{C}, \otimes, \mathsf{I} \rangle$ with "copy" $\mathsf{copy}_{\mathsf{C}}$ and "delete" $\mathsf{del}_{\mathsf{C}}$ morphisms for each object C, drawn diagrammatically as and respectively, that form a commutative comonoid:



Because of the leftmost equation above, we sometimes write a "trident" for either side of it. Moreover, both copy and del need to be compatible with the monoidal structure:



We say del *is natural* if $= $ for every morphism f. A *Markov category* is a CD category in which del is natural. A CD category $\mathbb{C}$ *has conditionals* if for each morphism $f\colon \mathsf{A} \to \mathsf{X} \otimes \mathsf{Y}$, there exist (not necessarily unique) morphisms $f_{\mathsf{X}}\colon \mathsf{A} \to \mathsf{X}$ (called the *marginal*) and $f_{|\mathsf{X}}\colon \mathsf{X} \to \mathsf{Y}$ (called the *conditional*) such that . When $\mathbb{C}$ is a Markov category, such marginal f_{X} is unique given X by the naturality of del:



3 DIBI Logic and its Probabilistic Model

In this section we review the logic of *Dependence and Independence Bunched Implications* (DIBI). For space reasons, we focus on the discrete probabilistic model for DIBI. Interested readers may refer to [4] for the relational model, whose construction follows similar steps.

DIBI formulas (based on a set $\mathcal{AP}$ of atomic formulas) are defined inductively as follows:

$$P, Q ::= p \in \mathcal{AP} \mid \top \mid I \mid P \wedge Q \mid P \to Q \mid P * Q \mid P \mathrel{-\!\!*} Q \mid P \fatsemi Q \mid P \multimap Q \mid P \multimapinv Q$$

The additive conjunction $\wedge$ is the standard Boolean conjunction. The multiplicative conjunction $*$ states that P and Q are independent. Both are already present in BI. DIBI extends BI with the non-commutative conjunction $\fatsemi$[1], where $P \fatsemi Q$ states that Q may depend on P. The operation $\mathrel{-\!\!*}$ is adjoint to $*$, $\to$ is adjoint to $\wedge$, and $\multimap$, $\multimapinv$ are adjoints to $\fatsemi$. DIBI formulas are interpreted on DIBI *models*, each consisting of a *DIBI frame* on a set of states A and a *valuation* function $\mathcal{V}\colon \mathcal{AP} \to \mathcal{P}(A)$ that maps an atomic proposition to the set of states on which it is true. While a BI frame is based on a partial commutative monoid [10], a DIBI frame consists of two monoids (one commutative and one not) on the same underlying set, taking care of the two non-additive conjunctions $*$ and $\fatsemi$, respectively.

---

[1] Not to be confused with the additive context constructor which is also denoted as $\fatsemi$ in the standard BI literature such as [28, 34].

$$
\begin{array}{llll}
a \oplus b \doteq b \oplus a & (\oplus\text{-}\textsc{Com}) & \exists e \in E\colon a = e \odot a & (\odot\text{-}\textsc{UnitExist}_{\mathrm{L}}) \\
\exists e \in E\colon a = e \oplus a & (\oplus\text{-}\textsc{UnitExist}) & \exists e \in E\colon a = a \odot e & (\odot\text{-}\textsc{UnitExist}_{\mathrm{R}}) \\
(a \oplus b) \oplus c \doteq a \oplus (b \oplus c) & (\oplus\text{-}\textsc{Assoc}) & (a \odot b) \odot c \doteq a \odot (b \odot c) & (\odot\text{-}\textsc{Assoc})
\end{array}
$$

$$
\begin{array}{ll}
e \in E \,\&\, (a \oplus e){\Downarrow} \Longrightarrow (a \oplus e) \sqsupseteq a & (\oplus\text{-}\textsc{UnitCoh}) \\
e \in E \,\&\, (a \odot e){\Downarrow} \Longrightarrow (a \odot e) \sqsupseteq a & (\odot\text{-}\textsc{UnitCoh}_{\mathrm{R}}) \\
e \in E \,\&\, e' \sqsupseteq e \Longrightarrow e' \in E & (\textsc{UnitClosure}) \\
(a \oplus b){\Downarrow} \,\&\; a \sqsupseteq a' \,\&\, b \sqsupseteq b' \Longrightarrow (a' \oplus b'){\Downarrow} \,\&\, (a \oplus b) \sqsupseteq (a' \oplus b') & (\oplus\text{-}\textsc{DownClosed}) \\
(a \odot b){\Downarrow} \,\&\, (a \odot b) \sqsubseteq c' \Longrightarrow \exists a', b'\colon a' \sqsupseteq a \,\&\, b' \sqsupseteq b \,\&\, c' = (a' \odot b') & (\odot\text{-}\textsc{UpClosed}) \\
(a_1 \odot a_2) \oplus (b_1 \odot b_2) \doteq (a_1 \oplus b_1) \odot (a_2 \oplus b_2) & (\textsc{RevExchange})
\end{array}
$$

Figure 2 DIBI frame conditions (with implicit outermost universal quantifiers), where $\Downarrow$ stands for "is defined", $\doteq$ means "equal when either side is defined".

▶ **Definition 1** ([4]). *A* DIBI frame *is a tuple $\mathcal{A} = \langle A, \sqsubseteq, \oplus, \odot, E\rangle$, where A is a set of states, $\sqsubseteq$ is a preorder on A, $E \subseteq A$ are units, and $\oplus, \odot\colon A \times A \rightharpoonup A$ are partial binary operations*[2]*, satisfying the frame conditions in Figure 2.*

The operations $\odot$ and $\oplus$ are referred to as the sequential and parallel compositions of states. Intuitively, $a \sqsubseteq b$ says that a can be extended to b, and E is the set of states that act as units for these operations. For capturing conditional independence, atomic propositions $\mathcal{AP}$ have the form $S \triangleright [T]$, for finite sets of variables S, T. Roughly, $S \triangleright [T]$ means the values of variables in T only depend on that of S. We now present the semantics of DIBI formulas, restricting to the fragment needed for the current work.

▶ **Definition 2.** *Given a DIBI model $\langle \mathcal{A}, \mathcal{V}\rangle$,* satisfaction $\models_{\mathcal{V}}$ *of DIBI$_{\{\wedge,*,\fatsemi\}}$-formulas at $\mathcal{A}$-states is inductively defined as follows:*

$$
\begin{array}{lll}
a \models_{\mathcal{V}} I \;\; \textit{iff} \;\; a \in E & & a \models_{\mathcal{V}} \top \;\; \textit{always} \\
a \models_{\mathcal{V}} (A \triangleright [B]) & \textit{iff} & a \in \mathcal{V}(A \triangleright [B]) \\
a \models_{\mathcal{V}} P \wedge Q & \textit{iff} & a \models_{\mathcal{V}} P \textit{ and } a \models_{\mathcal{V}} Q \\
a \models_{\mathcal{V}} P * Q & \textit{iff} & \exists b_1, b_2 \in A \textit{ such that } b_1 \oplus b_2 \sqsubseteq a,\ b_1 \models_{\mathcal{V}} P,\ b_2 \models_{\mathcal{V}} Q \\
a \models_{\mathcal{V}} P \fatsemi Q & \textit{iff} & \exists b_1, b_2 \in A \textit{ such that } b_1 \odot b_2 = a,\ b_1 \models_{\mathcal{V}} P,\ b_2 \models_{\mathcal{V}} Q
\end{array}
$$

For a concrete example of DIBI models, we review the probabilistic models on program memories. Let Val be a set of values, to which variables in Var are assigned. A *memory over* a finite set of variables X is a function $\mathbf{m}\colon X \to \mathrm{Val}$, and the *memory space* over X is the set of all memories over X, denoted as $\mathbf{M}[X; \mathrm{Val}]$, or $\mathbf{M}[X]$ when Val is clear. Given a memory $\mathbf{m} \in \mathbf{M}[X]$ and a subset $U \subseteq X$, the memory $\mathbf{m}^U : U \to \mathrm{Val}$ is the restriction of $\mathbf{m}$ to the domain U. To express probabilistic features, we use $\mathcal{D}S$ to denote the set of discrete distributions over S; that is, the set of all $\mu\colon S \to [0, 1]$ such that the *support* $\mathrm{supp}(\mu) = \{s \in S \mid \mu(s) > 0\}$ is finite, and $\sum_{s\in S} \mu(s) = 1$. A dirac distribution δ_s on an outcome s is the distribution such that $\delta_s(s) = 1$, and $\delta_s(s') = 0$ for any $s' \neq s$. Given a distribution μ in $\mathcal{D}\mathbf{M}[X]$, if $Y \subseteq X$, we define the marginalisation of μ to $\mathcal{D}\mathbf{M}[Y]$, written as $\pi_Y \mu$, by letting $(\pi_Y \mu)(\mathbf{m}') = \sum_{\mathbf{m}\in\mathbf{M}[X] \mid \mathbf{m}^Y = \mathbf{m}'} \mu(\mathbf{m})$.

[2] Note that, even though $\odot, \oplus$ are also partial in the models considered in [4], they have type $A \times A \to \mathcal{P}(A)$ in that work. This is because the authors obtain completeness of DIBI logic using a method developed by Docherty [10], which only works for the more general type. Because the operations are actually partial rather than non-deterministic, and we are not interested in completeness here, we stick to the more accurate type.

We are now ready to introduce the notion of *probabilistic input-preserving kernels*. In words, a probabilistic kernel f maps a memory $\mathbf{m}$ on X to a distribution of memories on $Y \supseteq X$ whose support contains only memories $\mathbf{m}'$ that faithfully extend $\mathbf{m}$ (thus the name "input-preserving"). Alternatively, f can be seen as a conditional distribution $\Pr(Y \mid X)$ where $Y \supseteq X$, such that $\Pr(Y = B \mid X = A)$ is nonzero only if B restricted to X equals A.

▶ **Definition 3** ([4]). *A* probabilistic input-preserving kernel *(or* probabilistic kernel *for short) is a function $f \colon \mathbf{M}[X] \to \mathcal{D}\mathbf{M}[Y]$ satisfying:*

(i) $X \subseteq Y$,

(ii) $\pi_X \circ f = \eta^{\mathcal{D}}_{\mathbf{M}[X]}$*, where* $\eta^{\mathcal{D}}_{\mathbf{M}[X]}(\mathbf{m})$ *returns the dirac distribution over* $\mathbf{m}$.

The set of all probabilistic kernels is denoted ProbKer.

The probabilistic model is a structure based on the carrier set ProbKer.

▶ **Definition 4** (Probabilistic model, [4]). *The* probabilistic frame based on Val $\mathbf{PrFr}[\mathrm{Val}]$ *(or simply* $\mathbf{PrFr}$ *when* Val *is evident) is a tuple* $\langle \mathrm{ProbKer}, \sqsubseteq, \oplus, \odot, \mathrm{ProbKer}\rangle$ *where* $\odot, \oplus, \sqsubseteq$ *are defined for arbitrary* $f \colon \mathbf{M}[X] \to \mathcal{D}\mathbf{M}[Y]$ *and* $g \colon \mathbf{M}[Z] \to \mathcal{D}\mathbf{M}[W]$ *as:*

- *the sequential composition* $f \odot g$ *is defined iff* $Y = Z$*. In this case,* $f \odot g$ *is of the form* $\mathbf{M}[X] \to \mathcal{D}\mathbf{M}[W]$*, and given* $\mathbf{m} \in \mathbf{M}[X]$*,* $(f \odot g)(\mathbf{m})$ *maps* $\mathbf{n} \in \mathbf{M}[W]$ *to* $\sum_{\boldsymbol{\ell} \in \mathrm{supp}(f(\mathbf{m}))} (f(\mathbf{m})(\boldsymbol{\ell}) \cdot g(\boldsymbol{\ell})(\mathbf{n}))$*;*
- *the parallel composition* $f \oplus g$ *is defined iff* $X \cap Z = Y \cap W$*. In this case,* $f \oplus g$ *is of the form* $\mathbf{M}[X \cup Z] \to \mathcal{D}\mathbf{M}[Y \cup W]$ *such that given* $\boldsymbol{\ell} \in \mathbf{M}[X \cup Z]$ *and* $\mathbf{m} \in \mathbf{M}[Y \cup W]$*, we have* $(f \oplus g)(\boldsymbol{\ell})(\mathbf{m}) = f(\boldsymbol{\ell}^X)(\mathbf{m}^Y) \cdot g(\boldsymbol{\ell}^Z)(\mathbf{m}^W)$*;*
- *the subkernel relation* $f \sqsubseteq g$ *holds if there exist a finite set of variables* S *and* $h \in \mathrm{ProbKer}$ *such that* $g = \left(f \oplus \eta^{\mathcal{D}}_{\mathbf{M}[S]}\right) \odot h$*.*

The probabilistic model based on Val *consists of the probabilistic frame* $\mathbf{PrFr}[\mathrm{Val}]$ *and the following* natural valuation $\mathcal{V}_{\mathrm{nat}} \colon \mathcal{AP} \to \mathcal{P}(\mathrm{ProbKer})$*: given* $(S \rhd [T])$ *and* $f \colon \mathbf{M}[X] \to \mathcal{D}\mathbf{M}[Y]$*,* $f \in \mathcal{V}_{\mathrm{nat}}(S \rhd [T])$ *iff there exists a probabilistic kernel* $f' \colon \mathbf{M}[X'] \to \mathcal{D}\mathbf{M}[Y']$ *such that* $f' \sqsubseteq f$*,* $X' = S$ *and* $T \subseteq Y'$*.*

Next we give examples of probabilistic kernels and how they compose. We write a map from a variable x to a value c as c_x and use the ket notation $a|\omega\rangle$ to denote a probabilistic outcome ω of probability a.

▶ **Example 5.** Consider variables x, y, z that take values in $\mathbf{Val} = \{0, 1\}$. We define a map $f \colon \mathbf{M}[\{z\}] \to \mathcal{D}\mathbf{M}[\{x, y, z\}]$ by:

$$f(0_z) = \frac{1}{4}|0_x, 0_y, 0_z\rangle + \frac{1}{4}|0_x, 1_y, 0_z\rangle + \frac{1}{4}|1_y, 0_y, 0_z\rangle + \frac{1}{4}|1_y, 1_y, 0_z\rangle$$
$$f(1_z) = \frac{1}{16}|0_x, 0_y, 1_z\rangle + \frac{3}{16}|0_x, 1_y, 1_z\rangle + \frac{3}{16}|1_y, 0_y, 1_z\rangle + \frac{9}{16}|1_y, 1_y, 1_z\rangle$$

Each input memory (coloured) is preserved by f so it is a probabilistic kernel. Then define $g_1 \colon \mathbf{M}[\{z\}] \to \mathcal{D}\mathbf{M}[\{x, z\}]$ and $g_2 \colon \mathbf{M}[\{z\}] \to \mathcal{D}\mathbf{M}[\{y, z\}]$ as:

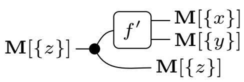
$$g_1(0_z) = \frac{1}{2}|0_x, 0_z\rangle + \frac{1}{2}|1_y, 0_z\rangle \qquad g_1(1_z) = \frac{1}{4}|0_x, 1_z\rangle + \frac{3}{4}|1_y, 1_z\rangle$$
$$g_2(0_z) = \frac{1}{2}|0_y, 0_z\rangle + \frac{1}{2}|1_y, 0_z\rangle \qquad g_2(1_z) = \frac{1}{4}|0_y, 1_z\rangle + \frac{3}{4}|1_y, 1_z\rangle$$

Both g_1 and g_2 are probabilistic kernels as well. The parallel composition $g_1 \oplus g_2$ is defined since $\{z\} \cap \{z\} = \{x, z\} \cap \{y, z\}$; in fact, it is easy to verify that $g_1 \oplus g_2 = f$. Moreover, g_1 and g_2 can be obtained by projecting the output of f on $\{x, z\}$ and $\{y, z\}$, respectively, and we can show $g_1 \sqsubseteq f$ and $g_2 \sqsubseteq f$.

4 DIBI models in Markov categories

In this section we construct more abstract DIBI models based on categorical structures. The starting point of our approach is a categorical characterisation of the concrete probabilistic models given above. In the following, we begin by showing examples of how elements in that model can be reformulated in categorical terms and then formally present our categorical construction of DIBI models.

As we noted in Section 1, the probabilistic DIBI kernels can be identified as morphisms in the Kleisli category for the distribution monad $\mathcal{K}\ell(\mathcal{D})$ (Definition 36); however, not all morphisms in $\mathcal{K}\ell(\mathcal{D})$ are probabilistic DIBI kernels, so we need to define the extra conditions categorically. First, we identify the $\mathcal{K}\ell(\mathcal{D})$ morphisms operating on memories. Let $\mathbb{MemPr}$ be the subcategory of $\mathcal{K}\ell(\mathcal{D})$ where objects are restricted to memory spaces over Val. That is, the objects are memory spaces $\mathbf{m}\colon X \to \mathrm{Val}$, and the morphisms are maps $f : \mathbf{M}[X] \to \mathcal{D}\mathbf{M}[Y]$ (or $f\colon \mathbf{M}[X] \rightharpoondown \mathbf{M}[Y]$ using the Kleisli category notation). Then, probabilistic kernels are exactly those morphisms in the $\mathbb{MemPr}$ that satisfy the input-preserving condition in Definition 3. So next, we need to express the input-preserving condition categorically. To do that, we depict $\mathbb{MemPr}$ morphisms using string diagrams, which is possible because $\mathbb{MemPr}$ is a subcategory of the monoidal category $\mathcal{K}\ell(\mathcal{D})$ We also observe that the codomain of an input-preserving kernel $f\colon \mathbf{M}[X] \rightharpoondown \mathbf{M}[Y]$ can be decomposed as $\mathbf{M}[X] \times \mathbf{M}[Y \setminus X]$. Recall the probabilistic kernel f from Example 5. Since its codomain $\mathbf{M}[\{x, y, z\}]$ can be decomposed as $\mathbf{M}[\{x\}] \times \mathbf{M}[\{y\}] \times \mathbf{M}[\{z\}]$, we can draw it as follows:



Intuitively, $\mathbf{M}[\{z\}]$ –●⊂ produces two copies of the value of z, and the values of x and y are computed from that of z via $\mathbf{M}[\{z\}]$ –$\boxed{f'}$– $\mathbf{M}[\{x\}]$, $\mathbf{M}[\{y\}]$, while the value of z gets preserved through a straight wire in the bottom. As in this example, such copy structure of $\mathcal{K}\ell(\mathcal{D})$ enables us to capture the "input-preserving" condition of probabilistic kernels generally.

Next we want to express the sequential ($\odot$) and parallel ($\oplus$) compositions of probabilistic kernels categorically. The former is exactly the sequential composition in $\mathcal{K}\ell(\mathcal{D})$. The parallel composition, however, is *not* the monoidal product $\otimes$ in $\mathcal{K}\ell(\mathcal{D})$. By definition, the monoidal product is total, while the parallel composition is partial. Even when the parallel composition is defined, the types of the resulting morphisms do not match. Suppose that the parallel composition of $f\colon \mathbf{M}[X] \rightharpoondown \mathbf{M}[Y]$ and $g\colon \mathbf{M}[U] \rightharpoondown \mathbf{M}[V]$ is defined, we have

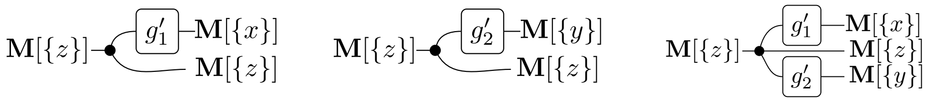
$$f \oplus g\colon \mathbf{M}[X \cup U] \rightharpoondown \mathbf{M}[Y \cup V] \qquad f \otimes g\colon \mathbf{M}[X] \times \mathbf{M}[U] \rightharpoondown \mathbf{M}[Y] \times \mathbf{M}[V]$$

The key difference is that parallel composition considers a single memory that can be projected into two pieces, while the monoidal product considers the cartesian product of two pieces of memory, no matter if they agree or not on overlapped variables. To define the parallel composition, we need to combine $\mathbf{M}[X]$ and $\mathbf{M}[U]$ into $\mathbf{M}[X \cup U]$ categorically. Thus, we use the fact that for disjoint Z_1, Z_2, $\mathbf{M}[Z_1 \cup Z_2] \cong \mathbf{M}[Z_1] \times \mathbf{M}[Z_2]$, which implies that $\mathbf{M}[X \cup U] \cong \mathbf{M}[X \setminus U] \times \mathbf{M}[X \cap U] \times \mathbf{M}[U \setminus X]$. We illustrate the parallel composition of two probabilistic kernels from Example 5 in the following example.

▶ **Example 6.** A first way of describing parallel composition of probabilistic kernels g_1 and g_2 from Example 5 category-theoretically is by seeing them as $\mathcal{K}\ell(\mathcal{D})$-morphisms. In this setting, we may define $g_1 \oplus g_2$ as the composite

$$\begin{array}{rl} \mathbf{M}[\{z\}] \xrightarrow{\langle\langle \eta, g_1' \rangle, g_2' \rangle} & (\mathcal{D}\mathbf{M}[\{z\}] \times \mathcal{D}\mathbf{M}[\{x\}]) \times \mathcal{D}\mathbf{M}[\{y\}] \\ & \Big\downarrow{\scriptstyle \mathsf{dst} \circ \langle \mathsf{dst}, id \rangle} \\ & \mathcal{D}((\mathbf{M}[\{z\}] \times \mathbf{M}[\{x\}]) \times \mathbf{M}[\{y\}]) \xrightarrow{\mathcal{D}\cong} \mathcal{D}\mathbf{M}[\{x, y, z\}] \end{array} \tag{1}$$

where dst is the double strength of the monad $\mathcal{D}$, and $g_1' \colon \mathbf{M}[\{z\}] \to \mathbf{M}[\{x\}]$, $g_2' \colon \mathbf{M}[\{z\}] \to \mathbf{M}[\{y\}]$ represent the conditional distributions obtained by suitable projections of g_1 and g_2 respectively. Now consider an alternative presentation: we draw kernels g_1 and g_2 respectively as the first and second string diagrams below. The parallel composition $g_1 \oplus g_2$ is then given by the rightmost string diagram below.



(2)

The formulation (2), which we adopt in our work, has two advantages over (1). First, string diagrams make for a cleaner presentation, abstracting away most "bureaucratic" steps in (1). Second, for kernels of larger sizes, the use of diagrams drastically simplifies calcu lations, see, e.g., the verification of frame conditions in proving Theorem 12 below. e Therefore, we will define categorical DIBI models and their compositions using string diagrams, though (1) exists as an alternative formulation.

We give the formal string diagrammatic definitions of the compositions later in Definition 10, as part of the generic construction of DIBI models.

While we simply use the concept of memory spaces $\mathbf{M}[X]$ to define the subcategory $\mathbb{M}\mathbb{e}\mathbb{m}\mathbb{P}\mathbb{r}$, that concept of memory spaces is customised for reasoning about probabilistic programs and relational databases and has potential to be parameterised. We observe that the side conditions of the parallel and sequential compositions are all based on comparing the set of variables in the (co)domains, so they only depend on the variable part (i.e., X) in $\mathbf{M}[X]$. This motivates us to define DIBI states as morphisms in a category whose objects are made of variables (see Definition 7) and abstracts the map between variables and corresponding memory spaces through an assignment $\theta \colon \mathrm{Var} \to \mathbf{ob}(\mathbb{C})$, for some Markov category $\langle \mathbb{C}, \otimes, \mathsf{I} \rangle$.

Finally, we need to express finite *sets* of variables and the union of *disjoint* such sets in a monoidal category, where the monoidal products of objects do not take care of deduplication. To address that, we impose a linear order $\preceq$ on Var such that indexed variables inherit the order of their indices, e.g., $x_1 \preceq x_2 \preceq x_3$. Let $x \prec y$ abbreviate for $x \preceq y$ and $x \neq y$. Then, finite sets of variables can be represented as finite lists of variables ordered by $\prec$, via a translation that we write as $[\![\cdot]\!]$. For instance, $[\![\{x_3, x_1, x_3, x_4\}]\!] = [x_1, x_3, x_4]$. This will be realised in two steps: we first define a category whose objects are finite lists of variables (Definition 7), and then we restrict the objects to finite lists without duplicates that respect the linear order (Definition 9).

Now we are ready to define a symmetric monoidal category $\mathbb{C}[\theta]$ that has enough structure to support our categorical characterisation of DIBI models. The category $\mathbb{C}[\theta]$ is parameterised by $\mathbb{C}$, whose objects abstract the concept of memory spaces. For simplicity, we fix a Markov category $\mathbb{C}$ throughout the rest of the section.

▶ **Definition 7.** *Let $\mathbb{C}[\theta]$ be the symmetric monoidal category whose objects are finite lists of variables, and morphisms $[x_1, \ldots, x_m] \to [y_1, \ldots, y_n]$ are $\mathbb{C}$-morphisms $\theta(x_1) \otimes \cdots \otimes \theta(x_m) \to \theta(y_1) \otimes \cdots \otimes \theta(y_n)$. Sequential composition is defined as in $\mathbb{C}$. The identity on $[x_1, \ldots, x_m]$ is $id_{\theta x_1 \otimes \cdots \otimes \theta x_m}$. The monoidal product in $\mathbb{C}[\theta]$ – which we also write as $\otimes$ with abuse of notation – is list concatenation on objects, and monoidal product in $\mathbb{C}$ on morphisms.*

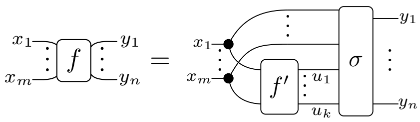
The SMC and Markov category structure of $\mathbb{C}[\theta]$ follow from those of $\mathbb{C}$. In particular, the symmetric map $[x_1,\ldots,x_m]\otimes[y_1,\ldots,y_n] \to [y_1,\ldots,y_n]\otimes[x_1,\ldots,x_m]$ is the symmetry morphism $(\theta(x_1)\otimes\cdots\otimes\theta(x_m))\otimes(\theta(y_1)\otimes\cdots\otimes\theta(y_n)) \to (\theta(y_1)\otimes\cdots\otimes\theta(y_n))\otimes(\theta(x_1)\otimes\cdots\otimes\theta(x_m))$ in $\mathbb{C}$. The copy map $[x_1,\ldots,x_m] \to [x_1,\ldots,x_m]\otimes[x_1,\ldots,x_m] = [x_1,\ldots,x_m,x_1,\ldots,x_m]$ is the copy map $\mathsf{copy}_{\theta(x_1)\otimes\cdots\otimes\theta(x_m)} \colon \theta(x_1)\otimes\cdots\otimes\theta(x_m) \to (\theta(x_1)\otimes\cdots\otimes\theta(x_m))\otimes(\theta(x_1)\otimes\cdots\otimes\theta(x_m))$ in $\mathbb{C}$. The tensor unit in $\mathbb{C}[\theta]$ is the empty list $[\ ]$, and θ maps it to the tensor unit I of the SMC $\mathbb{C}$. The delete map $[x_1,\ldots,x_m] \to [\ ]$ is then the delete map $\mathsf{del}_{\theta(x_1)\otimes\cdots\otimes\theta(x_m)} \colon \theta(x_1)\otimes\cdots\otimes\theta(x_m) \to \mathsf{I}$ in $\mathbb{C}$.

Sometimes we restrict ourselves to a uniform assignment θ; that is, for some fixed $\mathsf{C} \in \mathbf{ob}(\mathbb{C})$, $\theta(x) = \mathsf{C}$ for all $x \in \mathrm{Var}$. This is in line with the scenario where a fixed value space Val is used for all variables (see Definition 3). In this case, we write $\mathbb{C}[\theta]$ as $\mathbb{C}[\mathsf{C}]$ to emphasise the uniform value of the assignment. This category can be seen as the full subcategory of $\mathbb{C}$ freely generated by C, but with each occurrence of the generating object named by a variable. The next example shows how the construction in Definition 7 selects morphisms of $\mathcal{K}\ell(\mathcal{D})$ that act on memory spaces, among which we have all the probabilistic kernels.

▶ **Example 8.** Let $\mathbb{C}$ be $\mathcal{K}\ell(\mathcal{D})$, and $\theta\colon \mathrm{Var} \to \mathbf{ob}(\mathcal{K}\ell(\mathcal{D}))$ be the constant function $x \mapsto \mathrm{Val}$ for all $x \in \mathrm{Var}$. Then there is a full and faithful embedding functor $\iota\colon \mathbb{MemPr} \to \mathcal{K}\ell(\mathcal{D})[\theta]$: on objects, given a set X, $\iota(\mathbf{M}[X]) = [\![X]\!]$; on morphisms, given $f\colon \mathbf{M}[X] \to \mathcal{D}\mathbf{M}[Y]$ with $X = \{x_1,\ldots,x_m\}$ and $Y = \{y_1,\ldots,y_n\}$, its image $\iota(f)\colon [\![X]\!] \to [\![Y]\!]$ is the composed map $\mathrm{Val}^m \xrightarrow{\cong} \mathbf{M}[X] \xrightarrow{f} \mathcal{D}\mathbf{M}[Y] \xrightarrow{\mathcal{D}\cong} \mathcal{D}\mathrm{Val}^n$, where the isomorphisms are, e.g., $\mathbf{M}[Y] \xrightarrow{\cong} \mathbf{M}[\{y_1\}] \times\cdots\times \mathbf{M}[\{y_n\}] \xrightarrow{\cong^\theta} \mathrm{Val}^n$, using the valuation $\theta(y_j) = \mathrm{Val}$.

Just as the states of the probabilistic models are exactly input-preserving morphisms in $\mathbb{MemPr}$, we define the notion of *input-preserving kernels* in $\mathbb{C}[\theta]$, written $\mathrm{Ker}(\mathbb{C}[\theta])$ and use them as the states of our categorical DIBI models.

▶ **Definition 9.** *A $\mathbb{C}[\theta]$-morphism $f\colon [x_1,\ldots,x_m] \to [y_1,\ldots,y_n]$ is a $\mathbb{C}[\theta]$* input-preserving kernel *(or $\mathbb{C}[\theta]$*-kernel *for short) if $x_1 \prec\cdots\prec x_m$, $y_1 \prec\cdots\prec y_n$, and f can be decomposed as follows, where σ is a rewiring:*

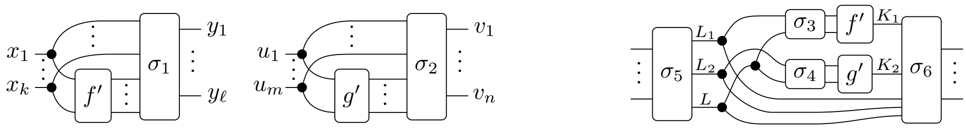


(3)

In words, a $\mathbb{C}[\theta]$-kernel is a morphism whose interfaces are essentially finite sets of variables, such that the input is preserved as part of the output (through the upper leg of those s). The map f' in (3) is referred to as the nontrivial part of the input-preserving kernel. It follows from Definition 9 that, for a $\mathbb{C}[\theta]$-kernel, its codomain $[y_1,\ldots,y_n]$ always subsumes its domain $[x_1,\ldots,x_m]$; also, $u_1,\ldots,u_k$ are precisely those y_js that are not among these x_is. Since the (co)domains of $\mathbb{C}[\theta]$-kernel are list presentation of sets, we also write the types of $\mathbb{C}[\theta]$-kernels using the corresponding sets, e.g., in (3), $f\colon \{x_1,\ldots,x_m\} \to \{y_1,\ldots,y_n\}$.

Next we define compositions on input-preserving kernels, generalising what we have seen in Example 6 for the probabilistic models.

▶ **Definition 10** (Compositions). *Given arbitrary $\mathbb{C}[\theta]$-kernels $f\colon X \to Y$ and $g\colon U \to V$ as in Figure 3a, their* sequential composition *$f \odot g$ is defined iff $cod(f) = dom(g)$, in which case $f \odot g = g \circ f$. Their* parallel composition *$f \oplus g$ is defined iff $X \cap U = Y \cap V$. Assume $L = [\![X \cap U]\!]$, $L_1 = [\![X \setminus (X \cap U)]\!]$, $L_2 = [\![U \setminus (X \cap U)]\!]$, $K_1 = [\![Y \setminus (Y \cap V)]\!]$, and $K_2 = [\![V \setminus (Y \cap V)]\!]$, then $f \oplus g\colon X \cup U \to Y \cup V$ is defined as in Figure 3b, where all the σ_is are rewiring morphisms for making the input and output variables $\prec$-ordered.*



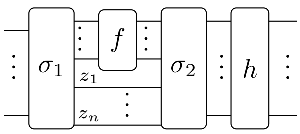
(a) $\mathbb{C}[\theta]$-kernels f and g.

(b) $f \oplus g$.

Figure 3 Parallel composition of $\mathbb{C}[\theta]$.

Note here a benefit of the diagrammatic representation: we can easily identify the memory overlap $\mathbf{M}[X \cap Y]$, as it is depicted a separate wire; with traditional syntax, we would need to apply associativity and commutativity to extract it from $\mathbf{M}[X \cup Y]$. It is easy to see that kernels are closed under compositions. Also, for curious readers, $\mathbb{C}[\theta]$-kernels with their parallel compositions form a *partially monoidal category* [2]. Next we define the subkernel relation.

▶ **Definition 11** (Subkernel). *Given two $\mathbb{C}[\theta]$-kernels f and g, we say f is a* subkernel *of g – denoted as $f \sqsubseteq g$ – if there exist $z_1, \ldots, z_n \in \mathrm{Var}$, a $\mathbb{C}[\theta]$-kernel h, and rewiring morphisms σ_1, σ_2 such that g can be expressed as follows:*

$g =$ 

The subkernel relation is transitive and reflexive, which can be shown simply by manipulations of the string diagram. We are finally able to state the main result of this section: $\mathbb{C}[\theta]$-kernels and their compositions form a DIBI frame.

▶ **Theorem 12.** $\mathbf{Fr}(\mathbb{C}[\theta]) = \langle \mathrm{Ker}(\mathbb{C}[\theta]), \sqsubseteq, \oplus, \odot, \mathrm{Ker}(\mathbb{C}[\theta]) \rangle$ *is a DIBI frame.*

Also, under the natural valuation $\mathcal{V}_{\mathrm{nat}}$, a $\mathbb{C}[\theta]$-kernel $f \colon X \to Y$ satisfies $S \triangleright [T]$ iff there is a subkernel $(f' \colon X' \to Y') \sqsubseteq f$ such that $X' = S$ and $Y' \supseteq T$. Thus:

▶ **Corollary 13.** $(\mathbf{Fr}(\mathbb{C}[\theta]), \mathcal{V}_{\mathrm{nat}})$ *is a DIBI model.*

We will see in Section 5 how to use this categorical construction to derive a wide range of DIBI models. Moreover, it also enables us to extract the conditions needed for a specific feature of a DIBI model as properties of the underlying category. Here is an example.

▶ **Proposition 14.** *If $\mathbb{C}$ further satisfies that for arbitrary morphisms f, g and object D, $f \otimes \mathsf{del}_{\mathsf{D}} = g \otimes \mathsf{del}_{\mathsf{D}}$ implies $f = g$, then a subkernel is unique given its type in the following sense: if $\mathbb{C}[\theta]$-kernels $f_1, f_2 \colon U \to V$ are both subkernels of g, then $f_1 = f_2$.*

Note that the uniqueness of subkernels has been observed already in the context of probabilistic and relational models, see [4, Sect. IV]. Proposition 14 reveals the general conditions under which this uniqueness holds for a wider class of DIBI models.

5 Examples

In this section we provide concrete instances of the categorical construction in Section 4. The first example recovers the probabilistic DIBI models. The remaining examples are new DIBI models. Some of them have been suggested in the DIBI paper [4], yet not materialised due to the complexity involved in stating each component and verifying the frame conditions. Within our framework, these steps become much easier to perform.

5.1 Probabilistic (and Relational) DIBI Models

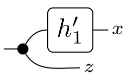
As we sketched in Example 6 and Example 8, the probabilistic DIBI kernels and $\langle \mathbf{Fr}(\mathcal{K}\ell(\mathcal{D})), \mathcal{V}_{\text{nat}} \rangle$ input-preserving kernels correspond to each other. We now formally show that the probabilistic DIBI model in Definition 4 can be recovered from the categorical DIBI model $\langle \mathbf{Fr}(\mathcal{K}\ell(\mathcal{D})), \mathcal{V}_{\text{nat}} \rangle$. Since both models are equipped with the natural valuation $\mathcal{V}_{\text{nat}}$, we focus on the frame part. To make the correspondence precise, we introduce the category of DIBI frames, as hinted in [4, Sect. III].

▶ **Definition 15.** *In* the category of DIBI frames $\mathbb{DibiFr}$, *objects are DIBI frames; morphisms* $f : \langle S, \sqsubseteq_S, \oplus_S, \odot_S, E_S \rangle \to \langle T, \sqsubseteq_T, \oplus_T, \odot_T, E_T \rangle$ *are functions* $f : S \to T$ *that respect all the relations and partial operations: for arbitrary* $s, s' \in S$*,*

- $s \sqsubseteq_S s'$ *implies* $f(s) \sqsubseteq_T f(s')$*;*
- *if* $s \star_S s'$ *is defined, then* $f(s) \star_S f(s')$ *is defined, and* $f(s) \star_T f(s') = f(s \star_S s')$*, for* $\star \in \{\oplus, \odot\}$*;*
- $s \in E_S$ *implies* $f(s) \in E_T$*.*

It turns out that the function ι introduced in Example 8 extends to an isomorphism of DIBI frames from $\mathbf{PrFr}[\mathrm{Val}]$ to $\mathbf{Fr}(\mathcal{K}\ell(\mathcal{D})[\mathrm{Val}])$.

▶ **Proposition 16.** $\mathbf{PrFr}[\mathrm{Val}] \cong \mathbf{Fr}(\mathcal{K}\ell(\mathcal{D})[\mathrm{Val}])$.

▶ **Example 17.** The probabilistic kernel $g_1 : \mathbf{M}[\{z\}] \to \mathcal{D}\mathbf{M}[\{x, z\}]$ from Example 5 corresponds to the following $\mathcal{K}\ell(\mathcal{D})[\{0,1\}]$-kernel $h_1 : [z] \to [x, z]$ – i.e., a $\mathcal{K}\ell(\mathcal{D})$-morphism $\{0,1\} \twoheadrightarrow \{0,1\}^2$ – where: $0 \mapsto \frac{1}{2}|0, 0\rangle + \frac{1}{2}|1, 0\rangle$, $1 \mapsto \frac{1}{4}|0, 1\rangle + \frac{3}{4}|1, 1\rangle$. Diagrammatically, h_1 is of the form 

, where $h_1' : [z] \to [x]$ is the map such that $0 \mapsto \frac{1}{2}|0\rangle + \frac{1}{2}|1\rangle$ and $1 \mapsto \frac{1}{4}|0\rangle + \frac{3}{4}|1\rangle$

Similarly, the relational DIBI model from [4] with the value space Val can be shown to be isomorphic to $\mathbf{Fr}(\mathcal{K}\ell(\mathcal{P}_i)[\mathrm{Val}])$, where $\mathcal{P}_i$ is the nonempty powerset monad A.

5.2 Stochastic DIBI Models

Using our categorical construction, we can derive a notion of DIBI model for continuous probabilistic (stochastic) processes, not previously considered. This is of interest because, as we show in Section 6, it allows to capture conditional independence for continuous probability using DIBI formulas. We take as underlying category $\mathbb{Stoch}$ of stochastic processes, defined as the Kleisli category $\mathcal{K}\ell(\mathcal{G})$ for the Giry monad on measurable spaces – see Appendix A for a full definition. Since $\mathcal{G}$ is an affine symmetric monoidal monad, $\mathbb{Stoch}$ is a Markov category. Applying Theorem 12 to $\mathbb{C} = \mathbb{Stoch}$, we get DIBI frames based on stochastic processes.

▶ **Proposition 18.** *Given an arbitrary map* $\theta : \mathrm{Var} \to \mathbf{ob}(\mathbb{Meas})$*,* $\mathbf{Fr}(\mathbb{Stoch}[\theta]) = \langle \mathrm{Ker}(\mathbb{Stoch}[\theta]), \sqsubseteq, \oplus, \odot, \mathrm{Ker}(\mathbb{Stoch}[\theta]) \rangle$ *is a DIBI frame.*

We call $\mathbf{Fr}(\mathbb{Stoch}[\theta])$ the *stochastic DIBI frame* based on θ and elements in $\mathrm{Ker}(\mathbb{Stoch}[\theta])$ stochastic kernels.

▶ **Example 19.** We show a representation of the *Box-Muller transformation* using stochastic kernels. Consider θ that maps all variable names to the Borel σ-algebra over reals $(\mathbb{R}, \mathcal{B}(\mathbb{R}))$. Define stochastic kernels $g_1 : \varnothing \to \{u\}$ and $g_2 : \varnothing \to \{w\}$ – both standing for $\mathbb{Stoch}$-morphisms $(\mathbf{1}, \{\varnothing, \mathbf{1}\}) \to (\mathbb{R}, \mathcal{B}(\mathbb{R}))$, or equivalently, a probabilistic measure on $(\mathbb{R}, \mathcal{B}(\mathbb{R}))$ – by $g_i(\bullet) = \textsc{Unif}(0, 1)$ for $i = 1, 2$, where $\textsc{Unif}(0, 1)$ is the uniform measure over the interval $(0, 1)$. Such a uniform measure over infinite outcomes is not possible in the discrete probabilistic DIBI model. Define another stochastic kernel $f : \{u, w\} \to \{u, w, x, y\}$ where the value of x, y are determined by the value of u, w:

$$f(u \mapsto v_u, w \mapsto v_w) = \delta_{v_u, v_w, \left(\sqrt{-2\ln u}\cdot\cos(2\pi w)\right)_x, \left(\sqrt{-2\ln u}\cdot\sin(2\pi w)\right)_y}.$$

Then $h = (g_1 \oplus g_2) \odot f$ gives a stochastic kernel $\varnothing \to \{u, w, x, y\}$. Box-Muller transformation says that x and y are independent in $h(\langle\rangle)$ despite their seemingly dependence on u and w.

Comparison with Lilac [24]. Our stochastic DIBI models can be used to reason about independence and conditional probabilities in continuous distributions. A recent work Lilac by Li et al. [24] proposed a BI model for the same goal, yet with some crucial differences in the set-up.

First, the states in Lilac's BI model are probabilistic space fragments of a fixed sample space, and their variables are mathematical random variables that deterministically map elements in the sample space to values. In comparison, we treat variables as names that can be associated to values or distributions. Our stochastic kernels – though not using an ambient sample space – can encode their set-up: we can devise a special variable Ω for "the sample space", and deterministic kernels from Ω to other variables encode random variables.

Second, to reason about conditional probabilities, Lilac want probability spaces to be disintegrable with respect to well-behaved random variables. To achieve that, they require probability spaces in their model to be extensible to Borel spaces, since disintegration works nicer in Borel spaces. By working with kernels, which already represent conditional probability spaces, we do not need to impose disintegratability on our DIBI states to reason about conditional probabilities. For instance, while disintegration is not always possible in the category $\mathbb{Stoch}$, we can still construct a DIBI model based on $\mathbb{Stoch}$.

Other measure-theoretic probabilistic DIBI models. The category $\mathbb{Stoch}$ is not the only Markov category for measure-theoretic probability. Another choice is $\mathbb{BorelStoch}$, a subcategory of $\mathbb{Stoch}$ obtained by restricting to standard Borel spaces as objects. It has some nice properties that $\mathbb{Stoch}$ does not satisfy, such as having conditionals as mentioned above. $\mathbb{BorelStoch}$ is also a Markov category and we can easily instantiate a DIBI model.

▶ **Proposition 20.** *Given any map $\theta\colon \mathrm{Var} \to \mathbf{ob}(\mathbb{BorelStoch})$, $\mathbf{Fr}(\mathbb{BorelStoch}[\theta])$ defined as $\langle \mathrm{Ker}(\mathbb{BorelStoch}[\theta]), \sqsubseteq, \oplus, \odot, \mathrm{Ker}(\mathbb{BorelStoch}[\theta])\rangle$ is a DIBI frame.*

The study of measure theory is also intertwined with topology, and another category for measure-theoretic probability is the Kleisli category of the *Radon monad* $\mathcal{R}$ based on the category of compact Hausdorff spaces $\mathbb{CHaus}$ and continuous maps, which we denote as $\mathcal{K}\ell_{\mathbb{CHaus}}(\mathcal{R})$. $\mathcal{K}\ell_{\mathbb{CHaus}}(\mathcal{R})$ is also a Markov category [12], so Theorem 12 applies.

▶ **Proposition 21.** *Given any map $\theta : \mathrm{Var} \to \mathbf{ob}(\mathcal{K}\ell_{\mathbb{CHaus}}(\mathcal{R}))$, $\mathbf{Fr}(\mathcal{K}\ell_{\mathbb{CHaus}}(\mathcal{R})[\theta])$ defined as $\langle \mathrm{Ker}(\mathcal{K}\ell_{\mathbb{CHaus}}(\mathcal{R})[\theta]), \sqsubseteq, \oplus, \odot, \mathrm{Ker}(\mathcal{K}\ell_{\mathbb{CHaus}}(\mathcal{R})[\theta])\rangle$ is a DIBI frame.*

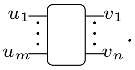
A measure-theoretic Markov category not formed as Kleisli categories is the Gaussian probability category $\mathbb{Gauss}$ [12]. Its objects are natural numbers, and a morphism $n \to m$ is a tuple (M, σ^2, μ) representing the function $f : \mathbb{R}^n \to \mathbb{R}^m$ with $f(v) = M \cdot v + \xi$, where ξ is the Gaussian noise with mean μ and covariance matrix σ^2. Its monoidal product is addition $+$ on the objects and vector concatenation on morphisms. $\mathbb{Gauss}$ differs from $\mathbb{Stoch}$, $\mathbb{BorelStoch}$ and $\mathcal{K}\ell_{\mathbb{CHaus}}(\mathcal{R})$ in that it does not arise as the Kleisli category associated to some monad. But since it is a Markov category, we can again instantiate DIBI models based on $\mathbb{Gauss}$.

▶ **Proposition 22.** *Given any map* θ : Var $\to$ $\mathbf{ob}(\mathbb{G}\mathrm{auss})$*,* $\mathbf{Fr}(\mathbb{G}\mathrm{auss}[\theta])$ *defined as* $\langle \mathrm{Ker}(\mathbb{G}\mathrm{auss}[\theta]), \sqsubseteq, \oplus, \odot, \mathrm{Ker}(\mathbb{G}\mathrm{auss}[\theta])\rangle$ *is a DIBI frame.*

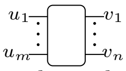
5.3 Syntactic DIBI Models

The DIBI models defined so far all have kernels defined by some processes over memory spaces. It is worth considering a different flavour: purely formal, syntactically generated DIBI models. We start by defining the underlying category.

▶ **Definition 23.** $\mathbb{S}\mathrm{yn}\mathbb{V}\mathrm{ar}$ *is the Markov category freely generated as follows:*

- *the generating objects are variables in* Var*;*
- *there is exactly one generating morphism of type* $[u_1, \dots, u_m] \to [v_1, \dots, v_n]$ *for distinct variables* $u_1 \prec \cdots \prec u_m$ *and* $v_1 \prec \cdots \prec v_n$*, written as string diagrams of the form* 

.

In words, $\mathbb{S}\mathrm{yn}\mathbb{V}\mathrm{ar}$-objects are finite lists of variables (without the requirements of duplicate-free or $\preceq$-ordered); morphisms are diagrams freely concatenated using —, -●<, -●, ><, and 

(quotiented by the Markov category equations). The syntactic DIBI frame is based on the category $\mathbb{S}\mathrm{yn}\mathbb{V}\mathrm{ar}[id]$, where $id\colon \mathrm{Var} \to \mathbf{ob}(\mathbb{S}\mathrm{yn}\mathbb{V}\mathrm{ar})$ maps $x \in \mathrm{Var}$ to the singleton list $[x]$.

▶ **Proposition 24.** $\mathbf{SynFr} = \langle \mathrm{Ker}(\mathbb{S}\mathrm{yn}\mathbb{V}\mathrm{ar}[id]), \sqsubseteq, \oplus, \odot, \mathrm{Ker}(\mathbb{S}\mathrm{yn}\mathbb{V}\mathrm{ar}[id])\rangle$ *is a DIBI frame.*

Equipped with the natural valuation $\mathcal{V}_{\mathrm{nat}}$, one obtains a DIBI model $\langle \mathbf{SynFr}, \mathcal{V}_{\mathrm{nat}}\rangle$. We postpone an example of $\mathbb{S}\mathrm{yn}\mathbb{V}\mathrm{ar}[id]$-kernels till Section 6, Example 33, in which $\mathbb{S}\mathrm{yn}\mathbb{V}\mathrm{ar}[id]$-kernels are used to distinguish two notions of conditional independence in Markov categories.

An interesting question for future work is how to extend the syntactic DIBI model to a term model. Typically being initial objects in categories of models, term models can help proving completeness and defining categorical semantics for formal systems, including algebraic theories [23], logics [38] (e.g., Lindenbaum–Tarski algebras) and type theories [19, 18]. A term model for DIBI could lead to a sound and complete axiomatisation of the specific version of DIBI logic in this paper, whose atomic propositions take the form of $S \triangleright [T]$.

6 Conditional independence

DIBI logic is designed for reasoning about conditional independence (CI). The prior work [4] shows that, CI in the discrete probabilistic models and join dependency in the relational models can be characterised by the same class of DIBI formulas. Generalising this result, in this section we define a notion of CI on $\mathbb{C}[\theta]$-kernels based on formula satisfaction. Since $\mathbb{C}[\theta]$ is a Markov category, we can compare our logical notion of CI with existing categorical definitions of CI in Markov categories [9, 12].

Fix a Markov category $\mathbb{C}$ and a map $\theta\colon \mathrm{Var} \to \mathbf{ob}(\mathbb{C})$. We define CI in the DIBI model $\langle \mathbf{Fr}(\mathbb{C}[\theta]), \mathcal{V}_{\mathrm{nat}}\rangle$.

▶ **Definition 25** (Conditional Independence)**.** *For any mutually disjoint finite sets of variables* W, X, Y, U*,* X *and* Y *are* DIBI conditionally independent *given* W *in a* $\mathbb{C}[\theta]$*-kernel*[3] $f\colon \varnothing \to W \cup X \cup Y \cup U$ *(denoted as* $X \perp\!\!\!\perp_{\mathrm{L}} Y \mid W$*) if*

$$f \vDash_{\mathcal{V}_{\mathrm{nat}}} (\varnothing \triangleright [W]) \mathbin{⨟} ((W \triangleright [X]) * (W \triangleright [Y])). \tag{4}$$

[3] Note that $\mathbb{C}[\theta]$-kernels with domain $\varnothing$ are not to be thought of as maps with empty domains. For instance, $\mathcal{K}\ell(\mathcal{D})[\theta]$-kernels of the form $\varnothing \to \{x, y\}$ corresponds to $\mathcal{K}\ell(\mathcal{D})$-morphisms $\mathbf{1} \to \theta(x) \times \theta(y)$, which denote distributions over x, y.

Let us unfold what (4) means. Under the natural valuation $\mathcal{V}_{\mathrm{nat}}$, the atomic proposition $S \triangleright [T]$ encodes the dependence of T on S: formally, a $\mathbb{C}[\theta]$-kernel $f\colon X \to Y$ satisfies $S \triangleright [T]$ iff f contains some subkernel $f'\colon S \to Y'$ such that $T \subseteq Y'$. So the formula in (4) requires that the kernel f has empty domain and can be decomposed as $f \sqsupseteq f_0 \odot (f_1 \oplus f_2)$, where f_0 determines the value on W, f_1 and f_2 determine the value on X and Y given the value on W, respectively, and f_1 and f_2 do so independently of each other. We illustrate the formula with examples in the discrete probabilistic DIBI model and the stochastic DIBI model.

▶ **Example 26.** In the setting of Example 5, consider the probabilistic kernel $h\colon \mathbf{M}[\varnothing] \to \mathcal{D}\mathbf{M}[\{x, y, z\}]$ such that :

$$\begin{aligned} h(\emptyset) =& \frac{1}{8}|0_x, 0_y, 0_z\rangle + \frac{1}{8}|0_x, 1_y, 0_z\rangle + \frac{1}{8}|1_y, 0_y, 0_z\rangle + \frac{1}{8}|1_y, 1_y, 0_z\rangle \\ &+ \frac{1}{32}|0_x, 0_y, 1_z\rangle + \frac{3}{32}|0_x, 1_y, 1_z\rangle + \frac{3}{32}|1_y, 0_y, 1_z\rangle + \frac{9}{32}|1_y, 1_y, 1_z\rangle \end{aligned}$$

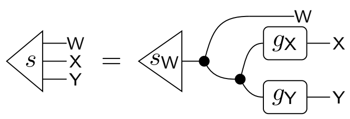
Then $h \vDash_{\mathcal{V}_{\mathrm{nat}}} (\varnothing \triangleright [\{z\}]) \fatsemi ((\{z\} \triangleright [\{z, x\}]) * (\{z\} \triangleright [\{z, y\}]))$, because $h = h_0 \odot f = h_0 \odot (g_1 \oplus g_2)$, where h_0 denotes the uniform distribution $\frac{1}{2}|0_z\rangle + \frac{1}{2}|1_z\rangle$.

▶ **Example 27.** Define g_1, g_2, f, h as in Example 19. We want to assert that variables x and y are independent in the distribution constructed by Box-Muller Transform. Independence is a special case of conditional independence in which the set of conditioned variables is empty. Thus, the goal is to assert $(\varnothing \triangleright [\varnothing]) \fatsemi ((\varnothing \triangleright [\{x\}]) * (\varnothing \triangleright [\{y\}]))$ – equivalently, $(\varnothing \triangleright [\{x\}]) * (\varnothing \triangleright [\{y\}])$.

Define $h_1 : \varnothing \to \{x\}$ and $h_2 : \varnothing \to \{y\}$ both as the standard normal distribution $\mathcal{N}(0, 1)$. Clearly $h_1 \vDash_{\mathcal{V}_{\mathrm{nat}}} \varnothing \triangleright [\{x\}]$ and $h_2 \vDash_{\mathcal{V}_{\mathrm{nat}}} \varnothing \triangleright [\{y\}]$. Moreover, some non-trivial calculations show that $(h_1 \oplus h_2) \sqsubseteq h$, and consequently $h \vDash_{\mathcal{V}_{\mathrm{nat}}} (\varnothing \triangleright [\{x\}]) * (\varnothing \triangleright [\{y\}])$ by definition.

Since the categorical DIBI models are based on Markov categories, we compare our logical notion of CI on kernels with the canonical notion of CI in Markov categories, which defines CI as decomposability of morphisms. Fix a Markov category $\mathbb{X}$ in Definitions 28, 31, and 34.

▶ **Definition 28.** *An $\mathbb{X}$-morphism $s\colon \mathsf{I} \to \mathsf{W} \otimes \mathsf{X} \otimes \mathsf{Y}$ displays the conditional independence of X and Y given W if there exist $\mathbb{X}$-morphisms $s_{\mathsf{W}}\colon \mathsf{I} \to \mathsf{W}$, $g_{\mathsf{X}}\colon \mathsf{W} \to \mathsf{X}$, $g_{\mathsf{Y}}\colon \mathsf{W} \to \mathsf{Y}$ such that the following equation holds. We write this as $\mathsf{X} \perp \mathsf{Y} \mid \mathsf{W}$.*



In the context of DIBI models, Definition 28 restricts to stating the conditional independence of X and Y given W in $\mathbb{C}[\theta]$-kernels of the form $\varnothing \to W \cup X \cup Y$. In particular, no extra variable (as that U in Definition 25) in the kernel's codomain is allowed.

▶ **Example 29.** We show an example of this notion of CI in the Markov category $\mathbb{G}\mathsf{auss}$. Consider a morphism $s : \varnothing \to \{w, x, y\}$ specified by the tuple $\left(!, \sigma^2 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 2 \end{bmatrix}, \mu = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}\right)$, where ! denotes the trivial map from empty domain. That is, s takes a length 0 vector and generates a length 3 vector, holding the values of w, x and y, with the normal distribution $\mathcal{N}(\mu, \sigma^2)$. This s can be decomposed as in Definition 28 with $s_w = (!, 0, 1)$, $g_x = (1, 0, 1)$, and $g_y = (1, 0, 1)$: composing s_w, g_x and g_y, we get $\mathbb{E}(w) = \mathbb{E}(\xi_w) = 0$, $\mathbb{E}(x) = \mathbb{E}(w + \xi_x) = 0 + 0 = 0$, and $\mathbb{E}(y) = \mathbb{E}(w + \xi_y) = 0$, justifying the noise's mean μ being a zero vector. For the covariance matrix, let $v = (w, x, y) - (\mathbb{E}(w), \mathbb{E}(x), \mathbb{E}(y))$. Then $\sigma^2 = \mathbb{E}(v \cdot v^T) = \mathbb{E}((w, x, y) \cdot (w, x, y)^T)$, and one may show that σ^2 is equal to the matrix above.

(a) Markov CI. **(b)** superset CI.

Figure 4 Two possible extension of plain CI.

▶ **Proposition 30.** *For any $\mathbb{C}[\theta]$-kernel $s\colon \varnothing \to W \cup X \cup Y$ where W, X, Y are mutually disjoint, $X \perp Y \,|\, W$ iff $X \perp\!\!\!\perp_{\mathrm{L}} Y \,|\, W$.*

In order to extend Proposition 30 to the scenario in Definition 25 where a kernel f might contain some U that does not appear in the CI statement in its codomain, we need to modify the notion of CI from Definition 28 – referred to as plain CI – to allow objects that do not appear in the CI statement to occur in the codomain of s. We suggest two sensible extensions.

▶ **Definition 31.** *Given an $\mathbb{X}$-morphism $s\colon \mathsf{I} \to \mathsf{W} \otimes \mathsf{X} \otimes \mathsf{Y} \otimes \mathsf{U}$,*

- *s displays* Markov CI*, denoted $\mathsf{X} \perp_{\mathrm{M}} \mathsf{Y} \,|\, \mathsf{W}$, if there exist $s_{\mathsf{W}}, g_{\mathsf{X}}, g_{\mathsf{Y}}$ satisfying 4a.*
- *s displays* superset CI*, denoted $\mathsf{X} \perp_{\mathrm{S}} \mathsf{Y} \,|\, \mathsf{W}$, if there exist s_0, g_1, g_2 satisfying 4b.*

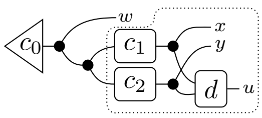
These two notions differ regarding to the treatment of the extra object U. In Figure 4a, we project out the extra object U and reduce the situation to that of Definition 28. In Figure 4b, U is kept and passed along through s_0, g_1, g_2. Clearly, both reduce to Definition 28 when no such U appears. We can now state that DIBI CI coincides with Markov CI, but is weaker than superset CI.

▶ **Theorem 32.** *Given the $\mathbb{C}[\theta]$-kernel $f\colon \varnothing \to W \cup X \cup Y \cup U$ from Definition 25,*

1. *f satisfies $X \perp_{\mathrm{M}} Y \,|\, W$ if and only if it satisfies $X \perp\!\!\!\perp_{\mathrm{L}} Y \,|\, W$;*
2. *if f satisfies $X \perp_{\mathrm{S}} Y \,|\, Z$, then it also satisfies $X \perp\!\!\!\perp_{\mathrm{L}} Y \,|\, Z$.*

Item 1 follows straightforwardly by unpacking the definitions. Item 2 follows from Item 1 and that $\mathsf{X} \perp_{\mathrm{S}} \mathsf{Y} \,|\, \mathsf{W}$ implies $\mathsf{X} \perp_{\mathrm{M}} \mathsf{Y} \,|\, \mathsf{W}$: simply apply $-\!\!\bullet_{\mathsf{U}}$ on both sides of Figure 4b, and Figure 4a follows via naturality of $-\!\!\bullet$. The converse of Item 2 does not hold in general, as demonstrated below.

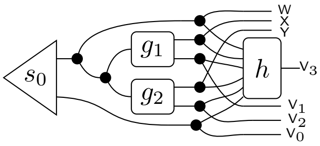
▶ **Example 33.** Consider the syntactic DIBI model $\langle \mathbf{SynFr}, \mathcal{V}_{\mathrm{nat}} \rangle$ from Section 5.3. Define the $\mathbb{SynVar}[id]$-kernel f as follows, where c_0, c_1, c_2, d are all generating morphisms, i.e., not further decomposable:



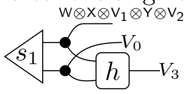
Then f satisfies the DIBI CI $x \perp\!\!\!\perp_{\mathrm{L}} y \,|\, w$, but not the superset CI $x \perp_{\mathrm{S}} y \,|\, w$: one cannot rewrite the diagram in the dotted box into a juxtaposition of two diagrams with output wires containing x and y, respectively; in other words, it cannot be rewritten as the style in Figure 4b.

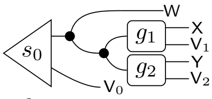
Example 33 gives some hint at how to weaken the superset CI to match DIBI CI: one needs to allow some morphism d following the morphism witnessesing $x \perp_{\mathrm{S}} y \,|\, z$. We formalise this idea and show the resulting notion is indeed equivalent to both Markov and DIBI CI.

▶ **Definition 34.** *An* $\mathbb{X}$*-morphism* $s\colon \mathsf{I} \to \mathsf{W} \otimes \mathsf{X} \otimes \mathsf{Y} \otimes \mathsf{U}$ displays the extended superset conditional independence *– denoted as* $\mathsf{X} \perp_{\mathsf{S}+} \mathsf{Y} \,|\, \mathsf{W}$ *– if there exist* $\mathbb{X}$*-morphisms* s_0, g_1, g_2, h *such that* s *can be decomposed as follows:*



(5)

Compared with Figure 4b, here one allows an extra morphism h to appear after the original superset CI diagrams in Figure 4b; in fact, modulo rewiring, (5) is exactly 

, where $s_1 =$ 

. One intuitive way to think of the extended superset CI is to view the morphisms as certain computational processes [31]: X and Y are independent given W in s if s could be obtained via a computation in which X and Y are computed independently from W (using g_1 and g_2 in (5) respectively), after which some further computation may apply (for which stands the h part in (5)).

▶ **Proposition 35.** *In Markov categories with conditionals, extended superset CI and Markov CI are equivalent. Therefore, in the context of Theorem 32, if* $\mathbb{C}$ *has conditionals, then the three notions of CI – DIBI CI, Markov CI, and extended superset CI – coincide.*

7 Conclusion

In this paper we provide a general recipe to construct models for DIBI logic, generalising the previously studied probabilistic and relational models. We adopt string diagrams to best visualise the "input-preserving" property that characterises the states in the models, as well as the compositions and subkernel relations, whose definition would be quite convoluted in non-diagrammatic syntax. Then, we derive various new classes of DIBI models of interest. In addition, we define an abstract notion of conditional independence in terms of DIBI formulas. Since our approach can construct DIBI models based on any Markov categories, we are then able to compare the logical CI notion with other definitions of CI proposed in the literature.

There are many promising directions for future work. On the logic side, DIBI logic – interpreted in the probabilistic models – was designed to be the assertion logic of Conditional Probabilistic Separation Logic (CPSL). Our categorical construction of a wide class of DIBI models suggests a generalisation of CPSL to obtain program logics in various scenarios beyond probabilistic programs, in the spirit of Moggi [27].

The notion of CI we propose can be straightforwardly generalised from Markov categories to copy-delete categories (see Section 2). This would allow us to encompass models such as relations with bag semantics in databases [8, 16], sub-probability measures [20]. However, to the best of the authors' knowledge, Proposition 30 fails for generic CD categories. Hence, finding appropriate notions of CI in this more general setting remains an open question.

From a categorical perspective, the definition of the category $\mathbb{C}[\theta]$ deserves further exploration, from at least three angles. First, the $\mathbb{C}[\theta]$-morphisms may be seen as a "bundle" of the images of some syntactic categories of variables and renaming (similar to $\mathbb{SynVar}$ from Section 5.3) under suitable functors – usually referred to as "models" in functorial semantics. We would like to make the connection with functorial semantics rigorous in terms of the categorical structures involved [23, 7]. Second, while the current work represents finite sets of variables using deduplicated finite $\preceq$-ordered lists, towards a more principled treatment, it is worth exploring using nominal string diagrams, a diagrammatic calculus for variables and renaming [2, 3, 1], to represent sets of variables. Third, our categorical treatment of variables seems related to prior work on internalising variables in categories; this problem has been studied since the early days of categorical logic, which led to the construction of polynomial categories [22], later extended to the monoidal setting [29, 17]. It is worth exploring potential connection with this line of work.

References

1 Samuel Balco. *Display calculi and nominal string diagrams.* PhD thesis, University of Leicester, 2020.

2 Samuel Balco and Alexander Kurz. Partially monoidal categories and the algebra of simultaneous substitutions, 2018. Available at `https://gdlyrttnap.pl/resources/papers/syco1.pdf`.

3 Samuel Balco and Alexander Kurz. Nominal String Diagrams. In Markus Roggenbach and Ana Sokolova, editors, *8th Conference on Algebra and Coalgebra in Computer Science (CALCO 2019)*, volume 139 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 18:1–18:20, Dagstuhl, Germany, 2019. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. `doi:10.4230/LIPIcs.CALCO.2019.18`.

4 Jialu Bao, Simon Docherty, Justin Hsu, and Alexandra Silva. A bunched logic for conditional independence. In *2021 36th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 1–14. IEEE, 2021. `doi:10.1109/LICS52264.2021.9470712`.

5 Gilles Barthe, Justin Hsu, and Kevin Liao. A probabilistic separation logic. *Proceedings of the ACM on Programming Languages*, 4(POPL):1–30, 2019. `doi:10.1145/3371123`.

6 Thomas Bayes. Lii. an essay towards solving a problem in the doctrine of chances. by the late rev. mr. bayes, f. r. s. communicated by mr. price, in a letter to john canton, a. m. f. r. s. *Philosophical Transactions of the Royal Society of London*, 53:370–418, 1763. `doi:10.1098/rstl.1763.0053`.

7 Filippo Bonchi, Paweł Sobociński, and Fabio Zanasi. Deconstructing lawvere with distributive laws. *Journal of logical and algebraic methods in programming*, 95:128–146, 2018. `doi:10.1016/j.jlamp.2017.12.002`.

8 Surajit Chaudhuri and Moshe Y Vardi. Optimization of real conjunctive queries. In *Proceedings of the twelfth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems*, pages 59–70, 1993. `doi:10.1145/153850.153856`.

9 Kenta Cho and Bart Jacobs. Disintegration and Bayesian inversion via string diagrams. *Mathematical Structures in Computer Science*, 29(7):938–971, 2019. `doi:10.1017/S0960129518000488`.

10 Simon Docherty. *Bunched logics: a uniform approach.* PhD thesis, UCL (University College London), 2019.

11 Brendan Fong and David I Spivak. *An invitation to applied category theory: seven sketches in compositionality.* Cambridge University Press, 2019. `doi:10.1017/9781108668804`.

12 Tobias Fritz. A synthetic approach to markov kernels, conditional independence and theorems on sufficient statistics. *Advances in Mathematics*, 370:107239, August 2020. `doi:10.1016/j.aim.2020.107239`.

13 Dan Geiger and Judea Pearl. Logical and algorithmic properties of conditional independence and graphical models. *The annals of statistics*, 21(4):2001–2021, 1993.

14 Clark Glymour, Kun Zhang, and Peter Spirtes. Review of causal discovery methods based on graphical models. *Frontiers in genetics*, 10:524, 2019. `doi:10.3389/fgene.2019.00524`.

15 Peter V Golubtsov. Monoidal kleisli category as a background for information transformers theory. *Inf. Process.*, 2(1):62–84, 2002.

16 Todd J. Green. *Bag Semantics*, pages 201–206. Springer US, Boston, MA, 2009. doi:10.1007/978-0-387-39940-9_979.

17 C. Hermida and R.D. Tennent. Monoidal indeterminates and categories of possible worlds. *Theoretical Computer Science*, 430:3–22, 2012. Mathematical Foundations of Programming Semantics (MFPS XXV). doi:10.1016/j.tcs.2012.01.001.

18 Martin Hofmann. On the interpretation of type theory in locally cartesian closed categories. In Leszek Pacholski and Jerzy Tiuryn, editors, *Computer Science Logic*, pages 427–441, Berlin, Heidelberg, 1995. Springer Berlin Heidelberg. doi:10.1007/BFb0022273.

19 Bart Jacobs. Semantics of weakening and contraction. *Annals of pure and applied logic*, 69(1):73–106, 1994. doi:10.1016/0168-0072(94)90020-5.

20 Achim Klenke. *Probability theory: a comprehensive course*. Springer Science & Business Media, 2013. doi:10.1007/978-1-84800-048-3.

21 Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

22 Joachim Lambek and Philip J Scott. *Introduction to higher-order categorical logic*, volume 7. Cambridge University Press, 1988. doi:10.5555/7517.

23 F William Lawvere. Functorial semantics of algebraic theories. *Proceedings of the National Academy of Sciences*, 50(5):869–872, 1963. doi:10.1073/pnas.50.5.869.

24 John M Li, Amal Ahmed, and Steven Holtzen. Lilac: a modal separation logic for conditional probability. *Proceedings of the ACM on Programming Languages*, 7(PLDI):148–171, 2023. doi:10.1145/3591226.

25 Pierre Simon marquis de Laplace. *Théorie analytique des probabilités*, volume 7. Courcier, 1820.

26 Ueli Maurer. Abstract models of computation in cryptography. In *Cryptography and Coding: 10th IMA International Conference, Cirencester, UK, December 19-21, 2005. Proceedings 10*, pages 1–12. Springer, 2005.

27 Eugenio Moggi. Notions of computation and monads. *Information and computation*, 93(1):55–92, 1991. doi:10.1016/0890-5401(91)90052-4.

28 Peter W O'Hearn and David J Pym. The logic of bunched implications. *Bulletin of Symbolic Logic*, 5(2):215–244, 1999. doi:10.2307/421090.

29 Duško Pavlović. Categorical logic of names and abstraction in action calculi. *Mathematical Structures in Computer Science*, 7(6):619–637, 1997. doi:10.1017/S0960129597002296.

30 Dusko Pavlovic. Chasing diagrams in cryptography. In *Categories and Types in Logic, Language, and Physics: Essays Dedicated to Jim Lambek on the Occasion of His 90th Birthday*, pages 353–367. Springer, 2014.

31 Dusko Pavlovic. *Programs as diagrams: From categorical computability to computable categories*. Springer Nature, 2023. doi:10.1007/978-3-031-34827-3.

32 Judea Pearl. *Causality*. Cambridge university press, 2009. doi:10.1017/CBO9780511803161.

33 Robin Piedeleu and Fabio Zanasi. An introduction to string diagrams for computer scientists. *CoRR*, abs/2305.08768, 2023.

34 David J Pym. *The semantics and proof theory of the logic of bunched implications*, volume 26. Springer Science & Business Media, 2013. doi:10.1007/978-94-017-0091-7.

35 John C Reynolds. Separation logic: A logic for shared mutable data structures. In *Proceedings 17th Annual IEEE Symposium on Logic in Computer Science*, pages 55–74. IEEE, 2002. URL: https://dl.acm.org/doi/10.5555/645683.664578.

36 Peter Selinger. A survey of graphical languages for monoidal categories. In *New structures for physics*, pages 289–355. Springer, 2010. doi:10.1007/978-3-642-12821-9_4.

37 Claude E Shannon. Communication theory of secrecy systems. *The Bell system technical journal*, 28(4):656–715, 1949.

38 Alfred Tarski. *Logic, semantics, metamathematics: papers from 1923 to 1938*. Hackett Publishing, 1983. doi:10.2307/2216869.

A Background on Monads

We first recall the basic definition of monads. We refer to [12, Sect. 3] for an overview of the material in this section. An endofunctor $\mathcal{T}\colon \mathbb{C} \to \mathbb{C}$ is a *monad* if it has a unit $\eta^{\mathcal{T}} : 1_{\mathbb{C}} \to \mathcal{T}$ and a multiplication $\mu^{\mathcal{T}} : \mathcal{T}^2 \to \mathcal{T}$ natural transformations satisfying certain compatibility conditions. Every monad $\mathcal{T}\colon \mathbb{C} \to \mathbb{C}$ induces a Kleisli category $\mathcal{K}\ell(\mathcal{T})$, whose objects are exactly $\mathbb{C}$-objects, and morphisms $\mathsf{X} \to \mathsf{Y}$ are $\mathbb{C}$ morphisms of type $\mathsf{X} \to \mathcal{T}\mathsf{Y}$, with compositions of $f\colon \mathsf{X} \to \mathcal{T}\mathsf{Y}$ and $g\colon \mathsf{Y} \to \mathcal{T}\mathsf{Z}$ given by $\mathsf{X} \xrightarrow{f} \mathcal{T}\mathsf{Y} \xrightarrow{\mathcal{T}g} \mathcal{T}\mathcal{T}\mathsf{Z} \xrightarrow{\mu^{\mathcal{T}}_{\mathsf{Z}}} \mathcal{T}\mathsf{Z}$. We will write the morphisms in $\mathcal{K}\ell(\mathcal{T})$ as $\mathsf{X} \to\!\!\!\!\!\circ\; \mathsf{Y}$ to distinguish them from their counterpart $\mathsf{X} \to \mathcal{T}\mathsf{Y}$ in $\mathbb{C}$. Importantly, if $\mathbb{C}$ is a SMC and $\mathcal{T}$ is a commutative monad, then $\mathcal{K}\ell(\mathcal{T})$ is also an SMC [19]. If $\mathcal{T}$ is affine symmetric monoidal, then $\mathcal{K}\ell(\mathcal{T})$ is a Markov category [15, 9].

In the remainder of this section, we recall the monads used in this paper: the distribution monad $\mathcal{D}$, the powerset monad $\mathcal{P}$ (and $\mathcal{P}_i$), the Giry monad $\mathcal{G}$, and the Radon monad $\mathcal{R}$.

▶ **Definition 36** (Discrete Distribution Monad). *The* discrete distribution *monad $\mathcal{D}$ is an endofunctor on* Set. *It maps a countable set X to the set of distributions over X, i.e., the set containing all functions μ over X is satisfying $\sum_{x\in X} \mu(x) = 1$, and maps a function $f : X \to Y$ to $\mathcal{D}(f) : \mathcal{D}(X) \to \mathcal{D}(Y)$, given by $\mathcal{D}(f)(\mu)(y) := \sum_{f(x)=y} \mu(x)$.*

For the monadic structure, define the unit η by $\eta_X(x) := \delta_x$, where δ_x denotes the Dirac distribution on x: for any $y \in X$, we have $\delta_x(y) = 1$ if $y = x$, otherwise $\delta_x(y) = 0$. Further, define bind*: $\mathcal{D}(X) \to (X \to \mathcal{D}(Y)) \to \mathcal{D}(Y)$ by $\mathsf{bind}(\mu)(f)(y) := \sum_{p\in\mathcal{D}(Y)} \mathcal{D}(f)(\mu)(p) \cdot p(y)$.*

▶ **Definition 37** (Powerset monad). *The* powerset monad *$\mathcal{P}$ is an endofunctor on* Set. *It maps every set to the set of its subsets $\mathcal{P}(X) = \{U \mid U \subseteq X\}$. We define $\eta_X\colon X \to \mathcal{P}(X)$ mapping each $x \in X$ to the singleton $\{x\}$, and* bind*: $\mathcal{P}(X) \to (X \to \mathcal{P}(Y)) \to \mathcal{P}(Y)$ by $\mathsf{bind}(U)(f) := \cup\{y \mid \exists x \in U. f(x) = y\}$. When restricted to nonempty powersets, the resulting functor $\mathcal{P}_i$ is still a monad, called the* nonempty powerset monad*.*

The next monad is defined on the category $\mathbb{Meas}$ of measurable spaces, which consists of measurable spaces (A, Σ_A) as objects, and measurable functions as morphisms. $\mathbb{Meas}$ is a monoidal category, where the monoidal product on objects and morphisms are given by the product of measurable spaces and measurable functions, respectively. In particular, the monoidal unit consists of the singleton measurable space $(\mathbf{1} = \{\bullet\}, \{\varnothing, \mathbf{1}\})$.

▶ **Definition 38** (Giry Monad). *The* giry monad *$\mathcal{G}$ maps a measurable space (X, Σ_X) to another measurable space $(\mathcal{G}(X), \Sigma_{\mathcal{G}(X)})$, where $\mathcal{G}(X)$ is the set of probability measures over X, and the σ-algebra $\Sigma_{\mathcal{G}(X)}$ is the coarsest σ-algebra over $\mathcal{G}(X)$ making the evaluation function $ev_A : \mathcal{G}(X) \to [0, 1]$, defined by $ev_A(\nu) = \nu(A)$, measurable for any $A \in \Sigma_X$. For each measurable function $f : X \mapsto Y$, $\mathcal{G}f : \mathcal{G}X \to \mathcal{G}Y$ is defined by $(\mathcal{G}f)(\nu)(B) = \nu(f^{-1}(B))$ for $B \in \Sigma_Y$. For the monadic structure, define the unit η by $\eta_X(x) = \delta_x$; define the bind operator $\mathsf{bind}_{X,Y} : \mathcal{G}X \to ((X \to \mathcal{G}Y) \to \mathcal{G}Y)$ by $\mathsf{bind}(\nu)(f)(B) = \int_X f(X)(B)d\nu$ for $B \in \Sigma_{\mathcal{G}Y}$.*

▶ **Definition 39** (Radon Monad). *The Radon monad $\mathcal{R}$ is a measure monad on the category of compact Hausdorff spaces. If maps a compact Hausdorff space X to the set of Radom measures μ on X such that $\mu(X) \leq 1$. It maps a continuous map between compact Hausdorff spaces $f : X \to Y$ to the push-forward measure $\mathcal{R}(f) : \mathcal{R}X \to \mathcal{R}Y$ given by $\mathcal{D}(f)(\mu)(y) := \mu(f^{-1}(y))$.*

For the monadic structure: we define the unit η to take a point $x \in X$ to the diract distribution δ_x solely supported at x. We also define the bind operator $\mathsf{bind}_{X,Y} : \mathcal{R}X \to ((X \to \mathcal{R}Y) \to \mathcal{R}Y)$ by $\mathsf{bind}(\nu)(f)(B) = \int_X f(X)(B)d\nu$.

The category of stochastic processses $\mathbb{Stoch}$ is the Kleisli category of the Giry monad $\mathcal{G}$. It is helpful to explicate its structure.

▶ **Definition 40.** *The symmetric monoidal category of stochastic processses $\mathbb{Stoch}$ has the following components:*

- *objects are measurable spaces (A, Σ_A);*
- *morphisms $(A, \Sigma_A) \to (B, \Sigma_B)$ are maps $f\colon \Sigma_B \times A \to [0,1]$ satisfying: for arbitrary $T \in \Sigma_B$, $f(T,-)\colon A \to [0,1]$ is measurable, and for arbitrary $a \in A$, $f(-,a)\colon \Sigma_B \to [0,1]$ is a probability measure;*
- *compositions of $f\colon (A, \Sigma_A) \to (B, \Sigma_B)$ and $g\colon (B, \Sigma_B) \to (C, \Sigma_C)$ is the map $g \circ f\colon \Sigma_C \times A \to [0,1]$ mapping (U, a) to $\int_{b\in B} g(U, b) \cdot f(db, a)$;*
- *$id_{(A,\Sigma_A)}$ maps $(S, a) \in \Sigma_A \times A$ to 1 if $a \in S$, and to 0 if $a \notin S$;*
- *the monoidal product $\otimes$ acts on objects as $(A, \Sigma_A) \otimes (B, \Sigma_B) = (A \times B, \Sigma_A \otimes \Sigma_B)$, where $\Sigma_A \otimes \Sigma_B$ is the smallest sigma-algebra containing $\Sigma_A \times \Sigma_B)$;*
- *the monoidal product $\otimes$ acts on morphisms to obtain product measures. That is, $(U, V) \in \Sigma_B \times \Sigma_D$ as follows: given $f\colon (A, \Sigma_A) \to (B, \Sigma_B)$ and $g\colon (C, \Sigma_C) \to (D, \Sigma_D)$, $f \otimes g\colon \Sigma_B \otimes \Sigma_D \times A \times C \to [0,1]$ maps (U, V, a, c) to $f(U, a) \cdot g(V, b)$.*