# Unifilar Machines and the Adjoint Structure of Bayesian Models

Nathaniel Virgo

We apply recent work on category theoretical probability to the idea of *Bayesian filtering*, making use of the concept of a strongly representable Markov category. We show that there is an adjunction between 'dynamical' and 'epistemic' models of a hidden Markov process. Concepts such as Bayesian filtering and conjugate priors arise as natural consequences of this adjunction. Along the way we define a notion of *unifilar machine*, which is a kind of stochastic Moore machine in which the output is chosen stochastically, but the update function is deterministic given the output. Unifilar machines are useful as models of the behaviour of stochastic systems; we show that in the Kleisli category of the distribution monad there is a terminal unifilar machine, and its elements are controlled stochastic processes, mapping sequences of the input alphabet probabilistically to sequences of the output alphabet.

## 1 Introduction

This paper is concerned with the mathematical structure of *Bayesian filtering*, which is a common problem in applications of Bayesian inference. The idea is that there is some system with known dynamics (which in general are stochastic) but an unknown hidden state. The goal is to keep track of a Bayesian prior over the states of the system, updating it to a posterior whenever a new observation is made. This is useful if we want to be able to control the hidden state, as in solving a partially observable Markov decision process (POMDP), for example.

To reveal the underlying mathematical structure we make use of recent results in synthetic probability, which allows us to write proofs at the category theoretic level without using measure theory directly. We work in the framework of Markov categories [7], and in particular we make use of the concept of *strongly representable Markov category* as defined in [9]. Strongly representable Markov categories include **BorelStoch** (whose objects are standard Borel spaces and whose morphisms are Markov kernels) and the Kleisli category of the (real-valued) distribution monad, which we refer to as **Dist**. Therefore most of our results apply to both measure-theoretic probability and finitely supported probability.

We model a system with a hidden state as a certain kind of stochastic Moore machine (essentially a hidden Markov model); we call this a *dynamical model* of the system. There is then a functor $B$ that takes such a dynamical model and maps it to an *epistemic model*. This lives in a different category of machines that we call *unifilar machines*, whose outputs are stochastic but whose state updates are deterministic. Its state space consists of probability distributions over the hidden states of the system, and its dynamics are given by Bayesian updating.

Our main technical result is theorem 2.7, which states that this functor is right adjoint to a forgetful functor in the opposite direction. This has an interesting consequence. The functor $B$ maps a dynamical model $\kappa$ to what could be called its *universal epistemic model*, $B(\kappa)$. If we consider another unifilar machine $\alpha$ equipped with a morphism $\alpha \to B(\kappa)$, we can also consider this an epistemic model of $\kappa$, in a sense that we describe next.

In applications, one doesn't necessarily want to keep track of the Bayesian distribution directly. Instead, one uses a parametrised family of distributions, chosen such that the update step only needs

Submitted to:
ACT 2023

to update the parameters to produce the posterior distribution. For this to work, the Bayesian posterior must always be in the same family of distributions as the prior. An example with enormous practical importance is the Kalman filter, example 2.8. Here the prior is a multivariate Gaussian and the posterior is always also a multivariate Gaussian. The filter's state space consists of the parameters of such a Gaussian, and the update step simply maps them to their new values. In our framework this kind of structure arises from considering a morphism $\alpha \to B(\kappa)$. The state space of $B(\kappa)$ consists of probability distributions, and the state space of $\alpha$ consists of values that parametrise them in a consistent way.

This idea is closely related to the notion of conjugate prior, which was previously studied in a category-theoretic context in [12]. The definition in that paper is essentially our eq. (21), which arises from our framework in a very natural way. The connection between Bayesian filtering and Bayesian inference is explored in section 2.1, where we also briefly touch on connections to recent work on de Finetti's theorem within a category-theoretic context [14, 8].

A secondary contribution of our paper is an exploration of the possible generalisations of Moore machines to the stochastic case. Our result involves two different generalisations of Moore machine, which we term *comb machine* (definition 2.2) and *unifilar machine* (definition 2.4). Unifilar machines in particular are of independent interest. They are based on an idea from the literature on $\epsilon$-machines [2].They are defined such that their output map is stochastic but their update map is (almost surely) deterministic given their input and their output. This means that their states map more directly to 'behaviours' than the states of a more general stochastic machine. Indeed, we show in section 3 that in **Dist** the category of unifilar machines has a terminal object, which consists of the collection of 'controlled stochastic processes,' also known as 'stochastic streams' [6]. In general, if a category of unifilar machines has a terminal object then it can be seen as an "object of behaviours" of stochastic systems.

Our Bayesian filtering machines have a strong resemblance to Bayesian lenses [19, 4] (see also [16], which develops the idea in a way that makes all of the relevant maps measurable). However, Bayesian filters seem to lack the backwards component of lenses and don't appear to compose like lenses. Understanding the relationship is an open problem.

Our work also seems related to the notion of *determinisation* from automata theory. Here one takes a nondeterministic automaton (one that may have more than one transition from a given state, but with no notion of probability assigned to them) and turns it into a deterministic automaton whose state space is the power set of the original automaton. Determinisation has been studied and generalised in a coalgebraic context [18, 1]; understanding the exact relation to our work remains a task for future work.

Bayesian filtering and its connection to conjugate priors was previously considered in a Markov category context by the author and colleagues in [22]. The novel contribution of the present paper is to reveal more of the abstract categorical structure underlying the idea, including the definitions of comb machine, unifilar machine and the adjoint structure involving the functor $B$, as well as the brief discussion of terminal unifilar machines in section 3.

## 1.1 Background on Representable Markov categories

We will use the machinery of representable Markov categories and in particular, strongly representable Markov categories, both defined in [9].[1] For general background on Markov categories we refer to [7]. Definitions in this section are from the literature, except for definition 1.2 (generalised almost surely).

[1] The definitions of strongly representable Markov categories and 'deterministic given $X$' appear in version 2 of the arXiv preprint [9] but not in the published version of the same paper ([10]) or version 3 of the preprint. The author understands that the removal was for reasons of narrative structure rather than any technical defect.

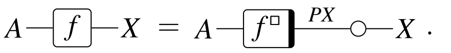
Recall from [9] that given an object $X$ in a Markov category $\mathcal{C}$, a *distribution object* is an object $PX$ equipped with a map $\mathrm{samp}_X\colon PX \to X$ such that for every morphism $f\colon A \to X$ there is a unique deterministic morphism $f^\square\colon A \to PX$ such that $f^\square \mathbin{⨾} \mathrm{samp}_\mathrm{X} = f$. A Markov category is then called *representable* if every object has a distribution object. Representable Markov categories often arise as the Kleisli categories of monads obeying conditions spelt out in [9].

The two examples we will use are **BorelStoch** (the Kleisli category of the Giry monad, restricted to standard Borel spaces) and the Kleisli category of the (real-valued) distribution monad, which we will call **Dist**. These are both shown to be representable in [9].

We recall also the following results about representable Markov categories: When every object has a distribution object, $P$ extends to a functor $P\colon \mathcal{C} \to \mathcal{C}_{\mathrm{det}}$. Restricting the domain of this functor we obtain a functor $P_{\mathrm{det}}\colon \mathcal{C}_{\mathrm{det}} \to \mathcal{C}_{\mathrm{det}}$, which will also be written as $P$, except when we wish to explicitly disambiguate. The functor $P_{\mathrm{det}}$ can be made into a monad on $\mathcal{C}_{\mathrm{det}}$, and the Kleisli category of this monad is $\mathcal{C}$. The unit has components $\delta_X = \mathrm{id}_X^\square\colon X \to PX$, and the multiplication has components $\mu_X = P(\mathrm{samp}_X)\colon PPX \to PX$.

This monad arises from an adjunction: the functor $P$ is right adjoint to the inclusion functor $\mathcal{C}_{\mathrm{det}} \hookrightarrow \mathcal{C}$. Its unit has components $\delta_X$ and its counit has components given by the sampling map $\mathrm{samp}_X\colon PX \to P$.

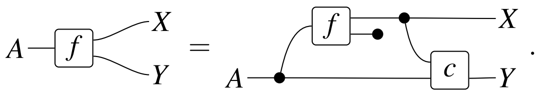
In string diagrams we will draw $\mathrm{samp}_X$ as a white dot. Additionally, if a morphism is known to be deterministic ([7], definition 2.2) we indicate this with a black bar at its right-hand edge, so we can write



(1)

We will need the definition of a strongly representable Markov category. For this we first recall some more definitions from [7] and [9].

**Definition 1.1** (conditionals; [7], definition 11.5)**.** Let $f\colon A \to X \otimes Y$ be a morphism in a Markov category $\mathcal{C}$. We say that a morphism $c\colon X \otimes A \to Y$ is a *conditional* of $f$ if



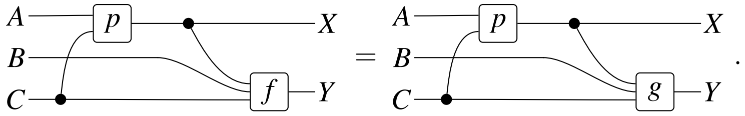
(2)

We say $\mathcal{C}$ *has conditionals* if every morphism of the appropriate type has a conditional.

The intuition is that for every value of the parameter $A$, the morphism $f$ defines a joint distribution between $X$ and $Y$, and eq. (2) represents a factorisation of this joint distribution into the marginal distribution over $X$ and a conditional distribution of $Y$ given $X$. Recall from [7] that **Dist** and **BorelStoch** both have conditionals, but **Stoch** does not. Conditionals are in general not unique when they exist; see [7], proposition 11.15 and the discussion surrounding it.

The following is essentially definition 13.1 of [7] ('almost surely'), but we generalise it slightly.

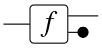
**Definition 1.2** (generalised almost surely)**.** Given a morphism $p\colon A \otimes C \to X$ in a Markov category $\mathcal{C}$, we say morphisms $f, g\colon X \otimes B \otimes C \to Y$ are *p-generalised-almost-surely equal* or *p-g.a.s. equal* if

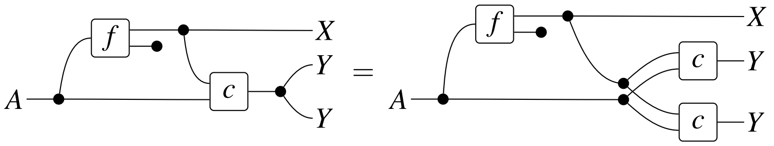


(3)

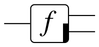
The idea is that in a measure-theoretic context such as **Stoch** or **BorelStoch**, for given values of $A$, $B$ and $C$, the values of $f$ and $g$ can differ only on subsets of $X$ that have measure zero according to $p$. In **Dist** it means that $f(y \mid x,b,c) = g(y \mid x,b,c)$ whenever $p(x \mid a,c) > 0$.

The definition of almost-surely in [7] allows $p$ but not $f$ or $g$ to depend on a parameter, so it is the same as ours but with $B$ and $C$ taken to be the unit. We avoid calling our version "almost surely" because several of the definitions in [9] rely on the original definition.

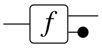
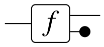
**Definition 1.3** (deterministic given $X$; [9], definition 6.4)**.** Let $f\colon A \to X\otimes Y$ be a morphism in a Markov category $\mathcal{C}$, and let $f$ be such that a conditional $c\colon X\otimes A \to Y$ exists. The morphism $f$ is said to be *deterministic given* $X$ if the conditional is  -generalised-almost-surely deterministic, in the sense that



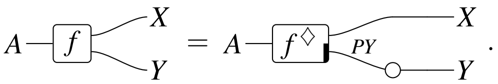
(4)

If $f\colon A\to X\otimes Y$ is known to be deterministic given $X$ we write it as  .

In [9] it is shown that if eq. (4) holds for one conditional of $f$ then it holds for all conditionals, so that this definition is independent of the choice of conditional $c$.

For both **BorelStoch** and **Dist**, if eq. (4) holds then $c$ is  -generalised-almost-surely equal to a deterministic morphism ([9], example 6.12), so for most purposes it will not hurt to think of such conditionals as genuinely deterministic, though only defined up to  -g.a.s. equivalence.

**Definition 1.4** (Strongly representable Markov category; [9], definition 6.7)**.** A strongly representable Markov category is a representable Markov category in which for every morphism $f\colon A\to X\otimes Y$ there is a unique morphism $f^{\diamondsuit}\colon A\to X\otimes PY$ such that (i) $f^{\diamondsuit}$ is deterministic given $X$, and (ii)



(5)

(This definition is less efficient than the one given in [9], which doesn't include an assumption that the category is representable, since this can be proven from weaker assumptions.) A strongly representable Markov category necessarily has conditionals, because $f^{\diamondsuit}$ has a conditional by the definition of deterministic given $X$, and if $c\colon X\times A\to PY$ is such a conditional then $c\,\mathbin{⨟}\,\mathrm{samp}_Y$ is a conditional of $f$.

**BorelStoch** is shown to be strongly representable in example 6.12 of [9]. For completeness we provide a proof that **Dist** is strongly representable in appendix A.1, where we also give an explicit construction for $f^{\diamondsuit}$ in **Dist**.

## 2 Machines and Bayesian Filtering

The following definitions are all relative to a Markov category $(\mathcal{C},\otimes,1)$ and a choice of objects called the *input space* $I$ and *output space* $O$, which we will assume to be fixed throughout this section.

For most of the following we will work with what we call "comb machines," which are a generalisation of Moore machines. However, many of the results also carry over to the case of Mealy machines, which we define first because they are simpler. The following definition is standard:

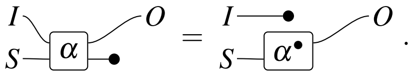
**Definition 2.1** (Stochastic Mealy machine)**.** A *stochastic Mealy machine* is an object $S$ of $\mathcal{C}$ called the *state space*, together with a morphism $\alpha\colon I\otimes S\to O\otimes S$ in $\mathcal{C}$. A morphism of Mealy machines $(S,\alpha)\to(T,\beta)$ is a morphism $f\colon S\to T$ in $\mathcal{C}$ such that $I\otimes S\xrightarrow{\alpha} O\otimes S\xrightarrow{\mathrm{id}_O\otimes f} O\otimes T = I\otimes S\xrightarrow{\mathrm{id}_I\otimes f} I\otimes T\xrightarrow{\beta} O\otimes T$. The category of Mealy machines will be written $\mathbf{Mealy}(I,O)$.

The idea is that a Mealy machine starts in some state in $S$, receives an input in $I$, and then produces an output in $O$ while simultaneously transitioning to a new state. The output may depend on the input and may be correlated with the new state. We don't require morphisms of Mealy machines to be deterministic.

We now briefly discuss Moore machines and their generalisation to the stochastic context. In a Cartesian category, a Moore machine consists of a state space $S$ and two maps: a *readout map* $S\to O$ and an *update map* $I \times S \to S$. An obvious way to generalise this to the stochastic case is to let both maps be stochastic, so that the update map has type $I \otimes S \to S$. However, machines with this definition tend not to be very well behaved, and in practise other definitions tend to be used.

One way to make stochastic Moore machines well behaved is to make the readout map deterministic. Machines of this kind can be expressed in terms of generalised lenses [20]; this is the approach taken in [17], for example. Intuitively, requiring a deterministic readout map allows the update map to "know" what the machine's last output was, since this can be inferred from the current value of $S$. However, for the present work we need the readout map to be stochastic, so we take a different approach:

**Definition 2.2** (Comb machine)**.** A *comb machine* in a Markov category $\mathcal{C}$ is an object $S$ of $\mathcal{C}$ (the state space), together with a morphism $\alpha \colon I \otimes S \to O \otimes S$ in $\mathcal{C}$ and a morphism $\alpha^{\bullet} \colon S \to O$ such that

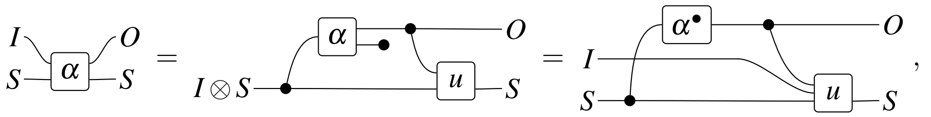


(6)

A morphism of comb machines $(S, \alpha) \to (T, \beta)$ is a morphism $f \colon S \to T$ in $\mathcal{C}$ that commutes with $\alpha$ and $\beta$ in the same way as for a Mealy machine. The category of comb machines will be written $\mathbf{CombMachine}(I, O)$.

Comb machines take their name from comb elements, as defined in a probability context in [13]. We avoid calling them Moore machines in order to avoid confusion with the more usual definition.

Equation (6) expresses the idea that the output of a comb machine cannot directly depend on the input. Consequently a comb machine $\alpha$ could be seen as a Mealy machine that obeys an extra condition, namely the existence of $\alpha^{\bullet}$ such that eq. (6) holds. However, we will often think of them differently. If $\mathcal{C}$ has conditionals then a comb machine $\alpha$ can always be factored as

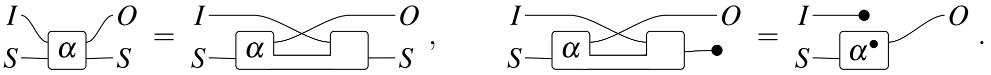


(7)

where $u$ is a conditional of $\alpha$ as shown. We refer to $\alpha^{\bullet}$ as the *readout map* and $u$ as an *update map* of the comb machine $(S, \alpha)$, analogously to the maps that define a Moore machine. If $f \colon (S, \alpha) \to (T, \beta)$ is a morphism of comb machines, then we have $f \mathbin{⨾} \alpha^{\bullet} = \beta^{\bullet}$, which we show in appendix A.2.

The readout map $\alpha^{\bullet}$ has the same type as in a Moore machine, $S \to O$, but the update map has type $O \otimes I \otimes S \to S$ and is only defined up to $\alpha^{\bullet}$-g.a.s. equality. This allows the next state and the output to be correlated for a given previous state and input, while still requiring the output to be independent of the input. Although update maps are not uniquely defined, their behaviour can only differ on measure zero subsets of the output space. In **Dist** this means their behaviour can differ only on outputs $o \in O$ that cannot occur at all in a given state, i.e. for which $\alpha^{\bullet}(o \mid s) = 0$.

We think of comb machines as giving their output first and then receiving their input, in contrast to Mealy machines, which first receive an input and then give an output.[2] The picture to have in mind for a comb machine is this:



(8)

[2] This raises the question of whether we can interpose some other morphism in between $\alpha^{\bullet}$ and $u$, so that the machine receives an input that can depend on its output, and perhaps also on the outputs of other machines. Answering this in the most general case is rather involved and we will not address it in this paper. However, in the case where $\mathcal{C}$ is **FinStoch**, [13] provides a way to compose 2-combs, of which comb machines are a special case.

We now introduce the concept of a *unifilar* machine. A unifilar machine has a stochastic readout map but a deterministic update map. (Or at least, a generalised-almost-surely deterministic one.) The term "unifilar" comes from the literature on computational mechanics, where it can be used to define $\epsilon$-machines [21]. In particular it appears in a machine-like context in [2], proposition 5. The formal context is different and we don't assume stationarity or irreducibility, but our definition achieves the same idea. We define unifilar machines in Mealy machine and comb machine flavours:

**Definition 2.3** (unifilar Mealy machine)**.** A *unifilar Mealy machine* in a Markov category is a Mealy machine $(S,\alpha)$ with the condition that $\alpha$ is deterministic given $O$. Additionally, we require morphisms of unifilar mealy machines to be deterministic. The category of unifilar Mealy machines will be written as $\mathbf{UnifilarMealy}(I,O)$.

**Definition 2.4** (unifilar comb machine)**.** A *unifilar comb machine* in a Markov category is a comb machine $(S,\alpha,\alpha^{\bullet})$ with the condition that $\alpha$ is deterministic given $O$. As with unifilar Mealy machines, we require morphisms of unifilar comb machines to be deterministic. The category of unifilar comb machines will be written as $\mathbf{UnifilarComb}(I,O)$.

When we say "unifilar machine" without qualification we mean a unifilar comb machine. Note that $\alpha$ must admit a conditional $O\otimes I\otimes S\to S$ in order to satisfy either definition.

The idea of a unifilar machine (of either type) is that all of the randomness comes from the choice of output. A unifilar comb machine factors according to eq. (7), with the additional feature that the conditional $u$ is $\alpha^{\bullet}$-generalised-almost-surely deterministic. We interpret this as follows: first the output $O$ is chosen stochastically (via $\alpha^{\bullet}\colon S\to O$), and then the state updates $\alpha^{\bullet}$-g.a.s. deterministically as a function of the output and the input. As for comb machines in general, the behaviour of an update map is uniquely specified on all but $\alpha^{\bullet}$-measure-zero subsets of the output space.

If $\mathcal{C}$ is Cartesian then Mealy machines and unifilar Mealy machines coincide, as do comb machines and unifilar comb machines, both of which coincide with Moore machines. So both comb machines and unifilar comb machines can claim to be a generalisation of Moore machines to the stochastic case.

It is worth saying something about the meaning of morphisms in these categories. The following can be made formal using the machinery we introduce in section 3, but for now we state it informally. We can think of a non-unifilar machine (of either flavour) as providing a stochastic map from infinite sequences of inputs to infinite sequences of outputs, subject to a *causality condition* that each output can only depend on inputs that were received at earlier points in time. (Recall that for Mealy machines we consider the input to be received before the output, and vice versa for comb machines.) We refer to this map as the machine's behaviour. For Mealy machines and comb machines, a morphism $(S,\alpha)\to(T,\beta)$ witnesses that $\beta$ is capable of exhibiting all of the externally observable behaviours that $\alpha$ can exhibit. Using a stochastic map makes sense because the states are unobserved and change randomly; we consider distributions over states to exhibit behaviours, as well as states themselves.

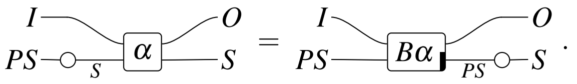
The interpretation of morphisms between unifilar machines is similar, but we require the morphisms to be deterministic. A morphism of unifilar machines witnesses not only that their externally observable behaviour is the same, but also that there is a mapping between their internal states that preserves this behaviour. This makes sense conceptually because we will generally consider the state of a unifilar machine to be observable.

Our first result concerns the existence of an adjunction between the categories $\mathbf{CombMachine}(I,O)$ and $\mathbf{UnifilarComb}(I,O)$, from which Bayesian filtering arises. A similar result holds for $\mathbf{Mealy}(I,O)$ and $\mathbf{UnifilarMealy}(I,O)$, which we will state at the end. Its proof is largely the same.

We first note that there is a forgetful functor $F\colon \mathbf{UnifilarComb}(I,O)\to\mathbf{CombMachine}(I,O)$ that embeds unifilar comb machines into comb machines. On objects it forgets that the machine obeys the deterministic-given-$O$ condition, and it also forgets that morphisms are deterministic.

If $\mathcal{C}$ is strongly representable we can construct a functor in the opposite direction. We first define it and then prove that it lands in $\mathbf{UnifilarComb}(I,O)$ and is a functor.

**Definition 2.5.** Suppose that $\mathcal{C}$ is a strongly representable Markov category. Then we define a putative functor $B\colon \mathbf{CombMachine}(I,O) \to \mathbf{UnifilarComb}(I,O)$. On objects it maps $(S,\alpha) \mapsto (PS, B\alpha)$, where $B\alpha = (\mathrm{id}_I \otimes \mathrm{samp}_S \mathbin{⨾} \alpha)^{\diamondsuit}$ is the unique morphism such that $B\alpha$ is deterministic given $O$ and

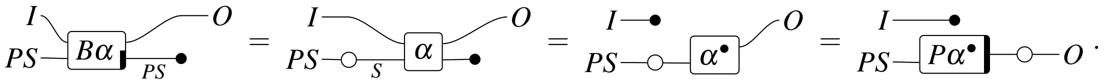


(9)

On morphisms, $B$ maps a morphism of comb machines with underlying map $f\colon S \to T$ to a morphism of unifilar machines with underlying deterministic map $Pf\colon PS \to PT$.

**Proposition 2.6.** *Let $\mathcal{C}$ be a strongly representable Markov category. Then definition 2.5 yields a functor $B\colon \mathbf{CombMachine}(I,O) \to \mathbf{UnifilarComb}(I,O)$.*

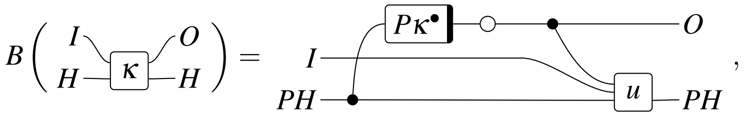
*Proof.* The mapping respects composition and identities by functoriality of $P$, but to prove $B$ is a functor we have to show $(i)$ that $B\alpha$ is indeed a unifilar comb machine, and $(ii)$ that that $Pf$ is indeed a morphism of unifilar comb machines. For $(i)$ we show that if eq. (6) holds for $\alpha$ then it holds for $B\alpha$:



(10)

We prove $(ii)$ in appendix A.3. The proof uses the strong representability of $\mathcal{C}$. □

We think of the functor $B$ as taking a dynamical model (in the form of a comb machine) and converting it into an epistemic model in the form of a unifilar machine. To see this, consider a comb machine $(H,\kappa)$, where we think of $H$ as a set of hidden states and $\kappa$ as a dynamical process that emits outputs and stochastically changes the hidden state as a function of the input. Then $B((H,\kappa))$ is a unifilar machine, which can be written (using eq. (10)) as



(11)

where the conditional $u$ is $(P\kappa^\bullet \mathbin{⨾} \mathrm{samp})$-g.a.s. deterministic as well as $(P\kappa^\bullet \mathbin{⨾} \mathrm{samp})$-g.a.s. unique.

The state space of this unifilar machine consists of probability measures over $H$. We will see that we can think of these as "beliefs" about the hidden state of $\kappa$, held by an idealised Bayesian reasoner, whose prior at any given time is an element of $PH$. This Bayesian reasoner does not interact with the machine $\kappa$, it only observes the inputs that $\kappa$ receives and the outputs it emits in response, updating its prior to a posterior at each time step.

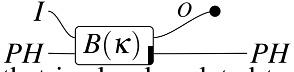
The output map $PH \xrightarrow{\mathrm{samp}_H} H \xrightarrow{\kappa^\bullet} O = PH \xrightarrow{P\kappa^\bullet} PO \xrightarrow{\mathrm{samp}_O} O$ "simulates" the output of $\kappa$. The map $P\kappa^\bullet$ maps the reasoner's prior beliefs about $H$ to its beliefs about the next output it will observe.

The update map $u$ performs Bayesian filtering. It takes as input a probability measure over the hidden states along with an input and an output, and it returns a new probability measure over hidden states. We think of it as taking a prior over the *current* value hidden state and returning a posterior distribution over the *next* value of the hidden state, conditioned on the observed output. It thus combines Bayesian updating with "simulating" the stochastic change in $H$.

The posterior distribution output from $u$ is only defined up to almost sure equivalence. In the case where $O$ is finite this is because for a given output $o \in O$ and a given belief $b \in PH$ we might have

$(b \mathbin{⨾} P\kappa^\bullet)(o) = 0$, i.e. the output $o$ is "subjectively impossible" according to the agent's current epistemic state. In this case calculating the Bayesian posterior in the usual way would lead to a division by zero, so there is no consistent value that the posterior distribution could take. Since the update map $u$ is only defined up to $(P\kappa^\bullet \mathbin{⨾} \mathrm{samp})$-g.a.s. equality its output only matters in those cases where this doesn't happen.

As one would expect from a Bayesian filter, instances of the map $u$ can be chained together in such a way that, given an initial distribution over $H$ and a sequence of inputs, we can recover the posterior over $H$ for a given observed sequence of outputs. We give a precise statement and proof of this in appendix A.4. The proof uses the fact that $u$ is $(P\kappa^\bullet \mathbin{⨾} \mathrm{samp})$-g.a.s. deterministic.

We can thus regard the functor $B$ as taking a dynamical model as input and turning it into an epistemic model. We remark that a similar operation is performed in the process of solving a partially observable Markov decision process (POMDP). A POMDP consists of some kind of machine — for simplicity let us say a comb machine $(H,\kappa)$ — together with a reward function. This machine is a dynamical model of some environment, and the goal is to find a "policy" that maximises the expected amount of reward that is accumulated over time, usually with an exponential discounting factor. (We will not consider reward functions in the present work.) A common solution technique involves converting the POMDP into a Markov decision process (MDP), which is a simpler class of problem. In an MDP the state space is assumed to be fully observed, so that there is no need to consider outputs. In an MDP the machine only takes inputs, and changes state stochastically as a function of its input, so it can be seen as an object of $\mathbf{CombMachine}(I,1)$. Again there is an associated reward function, which we will not consider in detail. To turn a POMDP into an MDP one forms the so-called "belief MDP", whose state space is given by probability distributions over $H$. In our framework it is given by 

. Note that this is a stochastic map in general. For an approach to POMDPs that is closely related to the present work, see [3].

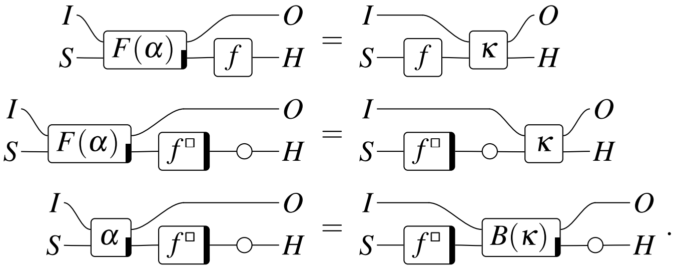
The following is our main technical result.

**Theorem 2.7.** *When $\mathcal{C}$ is strongly representable, the functor $B$ is right adjoint to the forgetful functor $F$,*

$$\begin{tikzcd} \mathbf{CombMachine}(I,O) \arrow[r, shift right, "B"'] \arrow[r, phantom, "\perp"] & \mathbf{UnifilarComb}(I,O). \arrow[l, shift right, "F"'] \end{tikzcd}$$

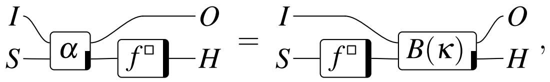
*Proof.* We show that if $f\colon S \to H$ is the map in $\mathcal{C}$ underlying a morphism $F((S,\alpha)) \to (H,\kappa)$ in $\mathbf{CombMachine}(I,O)$ then $f^\square\colon S \to PH$ is the deterministic map underlying a morphism $(S,\alpha) \to B((H,\kappa))$ in $\mathbf{UnifilarComb}(I,O)$, and vice versa. This will form the natural isomorphism of hom-sets needed for an adjunction.

Suppose $f\colon S \to H$ is the map underlying a morphism $F((S,\alpha)) \to (H,\kappa)$ in $\mathbf{CombMachine}(I,O)$. Then we have the following (where, as always, all diagrams are in $\mathcal{C}$):



(12)

Each side of the last equation consists of a morphism $I \otimes S \to O \otimes PH$ that is deterministic given $O$, composed with $\mathrm{id}_I \otimes \mathrm{samp}_H$. Using the defining property of a strongly representable Markov category we can conclude that



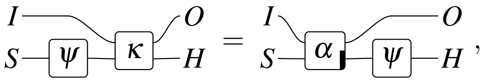
(13)

so that $f^\square$ underlies a morphism $(S,\alpha)\to B((H,\kappa))$ in $\mathbf{UnifilarComb}(I,O)$. Each of these steps can be reversed, so this gives a bijection $\mathbf{CombMachine}(I,O)(F(-),=)\cong\mathbf{UnifilarComb}(I,O)(-,B(=))$. Naturality follows from functoriality of $f$ and the naturality of the sampling map. □

This adjunction is related to the one between $P\colon \mathcal{C}_{\text{det}}\to\mathcal{C}$ and $\mathcal{C}_{\text{det}}\hookrightarrow\mathcal{C}$ in a representable Markov category, and it shares the same unit and counit. The unit has components $\delta_X\colon X\to PX$ and the counit has components $\text{samp}_X\colon PX\to X$, where $PX=BX$ on objects.

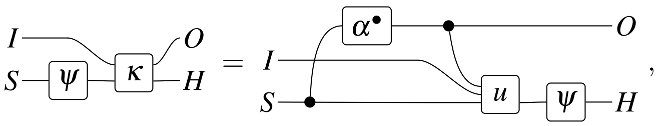
The existence of this adjunction has some interesting consequences. We have already established that the unifilar machine $B((H,\kappa))$ can be seen as an epistemic model of the comb machine $(H,\kappa)$, seen as a dynamical model. But now consider a morphism $(S,\alpha)\to B((H,\kappa))$ in $\mathbf{UnifilarComb}(I,O)$ from some other unifilar machine into $B((H,\kappa))$. We argue that when equipped with such a morphism, $(S,\alpha)$ *also* deserves to be seen as modelling $(H,\kappa)$.

To see this we consider its adjoint map $F((S,\alpha))\to(H,\kappa)$, which is given by an underlying map $\psi\colon S\to H$ in $\mathcal{C}$ such that



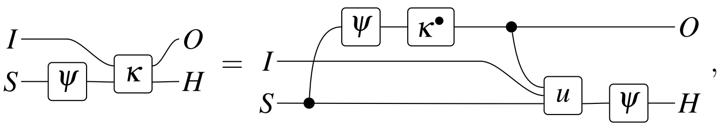
(14)

or



(15)

where $u$ is an update map for $\alpha$. By marginalising both sides (i.e. post-composing with $\text{id}_O\otimes\text{del}_H$) we have $\alpha^\bullet=\psi\,\mathring{,}\,\kappa^\bullet$, so this equation becomes



(16)

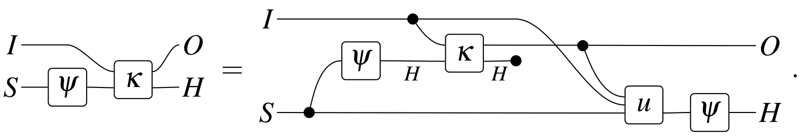
where $u$ is $\psi\,\mathring{,}\,\kappa^\bullet$-g.a.s. deterministic. This is a Bayesian filtering version of Jacobs' [12] definition of conjugate priors. It is not quite the same as the one in [22] because in that paper $u$ is not assumed to be almost-surely deterministic, so a stronger equation is needed. However, it is conceptually the same.

The morphism $\psi$ can be regarded as what the author and colleagues called an *interpretation map* in [22]. This means we think of the update map $u$ as a physical machine whose job is to keep track of an epistemic model of $\kappa$. At each step it receives both the input that was given to $\kappa$ and the output that $\kappa$ emitted in response. The machine's physical state ($S$) then updates in a ($\psi\,\mathring{,}\,\kappa^\bullet$-g.a.s.) deterministic way.

Equation (16) expresses the idea that when the machine receives a new piece of information in the form of an $(i,o)$ pair it should update its beliefs in a consistent way. The left-hand side can be seen as the agent's current beliefs about the *next* output and the *next* value of the hidden state, as a function of the next input. The equation says that after receiving an input and output pair, its new beliefs about the *current* hidden state should equal a conditional of its prior beliefs, conditioned on $i$ and $o$.

The adjoint map $\psi^\square\colon S\to PH$ can then be seen as mapping the unifilar machine's physical state to a probability measure over $H$ that we think of as "the machine's beliefs about $H$," i.e. its current Bayesian prior. Since $\psi^\square$ underlies a morphism $(S,\alpha)\to B((H,\kappa))$ it means that $\alpha$'s updates have to be able to 'simulate' the idealised Bayesian filtering that $B((H,\kappa))$ performs.

We now state the corresponding result for Mealy machines: as for comb machines there are functors $\mathbf{Mealy}(I,O) \underset{F}{\overset{B}{\rightleftarrows}} \mathbf{UnifilarMealy}(I,O)$ such that $F$ is left adjoint to $B$. The definitions and proofs are the same as for comb machines and unifilar comb machines, except that we don't need to care about the comb condition. These functors can be thought of in the same terms, with $B$ mapping a dynamical model to a corresponding epistemic model. The Mealy machine version of eq. (16) is



(17)

An example with enormous practical importance in control theory is the Kalman filter. Although Kalman [15] originally derived it in terms of error minimisation it is well known that it can also be constructed as a Bayesian filter. (See [11] for a somewhat informal exposition, for example.) We consider a version with measurement noise but no input signal. Unlike most treatments we allow the measurement noise and the process noise to be correlated.

**Example 2.8** (Kalman filter)**.** Let $\mathcal{C} = \mathbf{BorelStoch}$ and let $I = 1, H = \mathbb{R}^n, O = \mathbb{R}^m$. Consider a comb machine $(H,\kappa)$ where $\kappa(- \mid h)$ is normally distributed according to $\kappa(- \mid h) \sim \mathcal{N}(Ah,\Sigma)$, where $A$ is an $(m+n) \times n$ matrix and the $(m+n) \times (m+n)$ covariance matrix $\Sigma$ doesn't depend on $h$.

If $p\colon 1 \to H$ is a normal distribution with mean $\bar{h}$ and covariance matrix $\Sigma_p$, then $\boxed{p}\!-\!\boxed{\kappa}\!=$ is also Gaussian, with mean $\bar{s}$ and covariance $\Sigma' := A\Sigma_p A^T + \Sigma$. (See section 6 of [7], for example.) Writing $\Sigma'$ in block form as $\Sigma' = \left(\begin{smallmatrix}\Sigma'_{OO} & \Sigma'_{OH} \\ \Sigma'_{HO} & \Sigma'_{HH}\end{smallmatrix}\right)$, this distribution $p \mathbin{⨟} \kappa$ has a conditional $c\colon O \to H$ given by

$$c(- \mid o) \sim \mathcal{N}(\Sigma'_{HO}\Sigma'^{-}_{OO}o,\, \Sigma'_{HH} - \Sigma'_{HO}\Sigma'^{-}_{OO}\Sigma'_{OH}), \tag{18}$$

where $\Sigma'^{-}_{OO}$ is the Moore-Penrose pseudoinverse of $\Sigma'_{OO}$. (See example 11.8 of [7].)

Let us therefore define a unifilar machine $(S,\alpha)$ where $S$ is the set of pairs $(\bar{h},\Sigma_p)$, where $\bar{h} \in \mathbb{R}^n$ and $\Sigma_p$ is an $n \times n$ positive definite matrix. To define $\alpha\colon S \to O \times S$ we first define the map $\psi\colon S \to H$, which maps $(\bar{h},\Sigma_p)$ to a Gaussian with mean $\bar{h}$ and covariance $\Sigma_p$. We can define $\alpha\colon S \to O \times S$ by the readout function $S\!-\!\boxed{\alpha^\bullet}\!-\!O \;=\; -\!\boxed{\psi}\!-\!\boxed{\kappa}\!\bullet$ , and a deterministic update map $u\colon O \times S \to S$ that maps $((\bar{h},\Sigma_p),o)$ to $(\Sigma'_{HO}\Sigma'^{-}_{OO}o,\, \Sigma'_{HH} - \Sigma'_{HO}\Sigma'^{-}_{OO}\Sigma'_{OH})$, as in eq. (18). By construction, the maps $u$, $\psi$, $\alpha^\bullet$ and $\kappa$ obey eq. (16), and we can conclude that $\psi^\square$ is a map of unfilar machines from $(S,\alpha)$ to $B((H,\kappa))$.

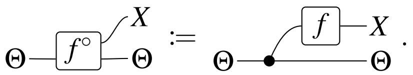
The update map $u$ is a version of the Kalman filter. Its state space $H$ parametrises Gaussian distributions via the map $\psi^\square$. The machine $(H,\kappa)$ is such that for a Gaussian prior the posterior will also be a Gaussian, and therefore the deterministic map $u$ only has to update the parameters upon receiving new data. We note a similarity between Kalman filtering and the category **Gauss** defined in [7], which we referred to in deriving it.

## 2.1 Bayesian Inference and Conjugate Priors

Up to now we have considered a version of Bayesian filtering in which the systems being modelled have the form of a comb machine. In this section we consider an important special case of this, in which the system being modelled simply emits independent and identically distributed outputs. This corresponds to the standard setting of Bayesian inference, where we receive independent samples from a known distribution with an unknown (but fixed) value for its parameters, and wish to use this data to make inferences about the parameters.

In this section we primarily consider machines whose input space is the terminal object in $\mathcal{C}$. In this case the distinction between comb machines and Mealy machines isn't relevant, and we refer to such machines as generators, defining $\mathbf{Generator}(X) = \mathbf{Mealy}(1,X) \cong \mathbf{UnifilarMachine}(1,X)$ and $\mathbf{UnifilarGenerator}(X) = \mathbf{UnifilarMealy}(1,X) \cong \mathbf{UnifilarComb}(1,X)$.

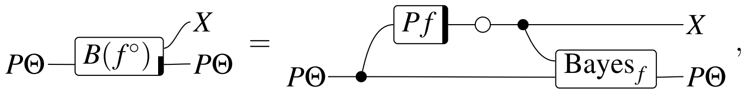
To model Bayesian inference in our setup we consider objects of $\mathbf{Generator}(X)$ represented by morphisms in $\mathcal{C}$ of the following special form:



(19)

In this setting we call $X$ the *sample space* and $\Theta$ the *parameter space*, and we think of $f$ as a statistical model, that is, a family of distributions over $X$ parametrised by $\Theta$.

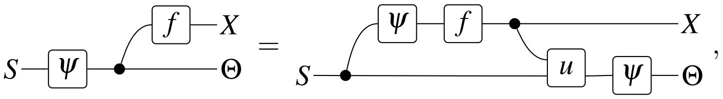
Applying the functor $B$ we get



(20)

where we have called the conditional $\text{Bayes}_f$ because that is what it does: it takes in a prior over the parameters together with some data $x \in X$, and returns the Bayesian posterior over the parameters, according to the model $f$. We give a precise statement and proof for this claim in appendix A.5.

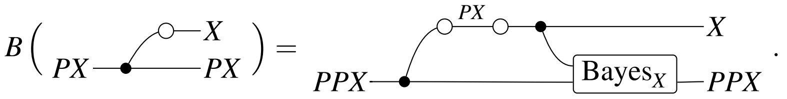
If we consider a map $\psi^{\square}$ into this machine from some other unifilar machine $(S,\alpha)$, we obtain exactly the notion of a conjugate prior. Its adjoint map of comb machines, $\psi\colon F((S,\alpha)) \to f^{\circ}$, obeys



(21)

which is the equation given in [12] as a definition of conjugate prior. The only minor difference is that here the update map is only defined up to $(\psi\,{}^{\circ}_{9}\,f)$-g.a.s. equality, instead of being a specified deterministic function. We think of $\psi\colon S \to \Theta$ as a statistical model and say that it is a conjugate prior for $f$. Its parameter space $S$ is referred to as the space of hyperparameters. Obtaining this more abstract perspective on the definition from [12] was one of the main motivations of this work.

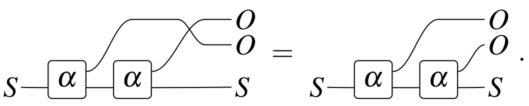
It is worth briefly mentioning the further special case in which $f$ is the sampling map, although we will not make use of it.



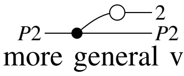
(22)

Here $\text{Bayes}_X$ also performs Bayesian updating, corresponding to inference about an unknown distribution. It takes a distribution over distributions over $X$, representing a prior, along with a sample from the unknown distribution. Its output is the Bayesian posterior over distributions, conditioned on the sample.

We note that all the generators in this section obey the property of *exchangeability*, specifically the version of that concept defined in [14] in the context of de Finetti's theorem. That is, they are all machines $(S,\alpha)$ such that



(23)

In our context, one of the results of [14] is that in **Stoch** (and hence also in **BorelStoch**) the category $\mathbf{Generator}(\{0,1\})$ has a terminal object, given by , which is part of their category-theoretic treatment of de Finetti's theorem. (A much more general version of de Finetti's theorem is proved for **BorelStoch** in [8], though in a less machine-like context.)

In the context of the machines in eq. (20) and eq. (22), exchangeability amounts to the idea that a Bayesian reasoner should reach the same posterior from the same data, regardless of the order in which the data are presented. (Except that here this is subject to the usual generalised-almost-surely condition.)

There is much more that can be said about exchangeability and its relationship to Bayesian inference within the framework of unifilar machines, but we will leave the topic here and return to the more general case of non-exchangeable machines in the next section.

## 3 Terminal objects as "objects of behaviours"

If **UnifilarComb**$(I,O)$ has a terminal object then it can be seen as an "object of behaviours," in much the same manner as a final coalgebra. If such a terminal object exists we call it an *object of transducers*. The intuition is that if we can meaningfully talk about its elements then they can be thought of as stochastic maps from infinite sequences of inputs to infinite sequences of outputs, subject to the causality condition described above, that each output can only depend on inputs that were received prior to it. To illustrate this idea we prove that transducer objects always exist in **Dist**, and their elements indeed have the form of stochastic maps between sequences.

**Proposition 3.1.** *In* **Dist***, for every set I, O, the categories* **UnifilarComb**$(I,O)$ *and* **UnifilarMealy**$(I,O)$ *both have terminal objects.*

*Proof.* One way to prove this is to note that unifilar machines in **Dist** can be expressed as coalgebras of polynomial functors in **Set**, and hence the transducer objects are final coalgebras, which are guaranteed to exist. However, in appendix A.6 we prove it a different way, by explicitly constructing the terminal objects as sets of so-called *controlled stochastic processes*. □

One advantage of formulating transducers internally in this way is that we can consider probability distributions over them. In particular, since $(T,\omega)$ is a terminal object it is equipped with an algebra of the monad $F \mathbin{⨟} B$ arising from the adjunction in theorem 2.7. This means that we can think of the unique map $B(F((T,\omega))) \to (T,\omega)$ as taking a probability distribution over transducers and returning a new transducer that represents is 'average' or 'expected' behaviour. This will work in any suitable Markov category, whenever the terminal object of **UnifilarComb**$(I,O)$ exists.

On the other hand, in **BorelStoch** the category **UnifilarMealy**$(\mathbb{R},\{0,1\})$ does not have a terminal object. Consider those machines with trivial state spaces, whose output depends only on the current input. Specifying the behaviour of such a machine amounts to specifying a measurable map $\mathbb{R} \to [0,1]$. But there is no measurable space of such functions, so there is no measurable space that includes the behaviours of all such machines. However, we conjecture that **BorelStoch** has terminal objects for **UnifilarComb**$(I,O)$ and **UnifilarMealy**$(I,O)$ when $I$ is a countable or finite set.

## Acknowledgements

The author thanks Martin Biehl for insightful comments on the manuscript, and Martin Biehl, Simon McGregor, Timorl, Matteo Capucci and Toby Smithe for discussions that stimulated the work. This paper was made possible through the support of Grant 62229 from the John Templeton Foundation. The opinions expressed in this publication are those of the author(s) and do not necessarily reflect the views of the John Templeton Foundation.

## References

[1] Jiří Adámek, Filippo Bonchi, Mathias Hülsbusch, Barbara König, Stefan Milius & Alexandra Silva (2012): *A coalgebraic perspective on minimization and determinization.* In: *Foundations of Software Science and Computational Structures: 15th International Conference, FOSSACS 2012, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2012, Tallinn, Estonia, March 24–April 1, 2012. Proceedings 15*, Springer, pp. 58–73.

[2] Nix Barnett & James P. Crutchfield (2015): *Computational Mechanics of Input-Output Processes: Structured Transformations and the $\epsilon$-Transducer.* *Journal of Statistical Physics* 161(2), pp. 404–451, doi:10.1007/s10955-015-1327-5. Available at `https://doi.org/10.1007/s10955-015-1327-5`.

[3] Martin Biehl & Nathaniel Virgo (2023): *Interpreting Systems as Solving POMDPs: A Step Towards a Formal Understanding of Agency.* In: *Active Inference. IWAI 2022. Communications in Computer and Information Science*, Springer, pp. 16–31, doi:10.1007/978-3-031-28719-0_2. Available at `https://doi.org/10.1007/978-3-031-28719-0_2`.

[4] Dylan Braithwaite, Jules Hedges & Toby St Clere Smithe (2023): *The Compositional Structure of Bayesian Inference*, doi:10.48550/ARXIV.2305.06112. Available at `https://arxiv.org/abs/2305.06112`.

[5] Kenta Cho & Bart Jacobs (2017): *Disintegration and Bayesian Inversion via String Diagrams*, doi:10.48550/ARXIV.1709.00322. Available at `https://arxiv.org/abs/1709.00322`.

[6] Elena Di Lavore, Giovanni de Felice & Mario Román (2022): *Coinductive Streams in Monoidal Categories*, doi:10.48550/ARXIV.2212.14494. Available at `https://arxiv.org/abs/2212.14494`.

[7] Tobias Fritz (2020): *A synthetic approach to Markov kernels, conditional independence and theorems on sufficient statistics.* *Advances in Mathematics* 370, p. 107239, doi:10.1016/j.aim.2020.107239. Available at `https://doi.org/10.1016/j.aim.2020.107239`.

[8] Tobias Fritz, Tomáš Gonda & Paolo Perrone (2021): *De Finetti's Theorem in Categorical Probability.* *Journal of Stochastic Analysis* 2(4), doi:10.31390/josa.2.4.06. Available at `https://doi.org/10.31390/josa.2.4.06`.

[9] Tobias Fritz, Tomáš Gonda, Paolo Perrone & Eigil Fjeldgren Rischel (2020): *Representable Markov Categories and Comparison of Statistical Experiments in Categorical Probability (version 2)*, doi:10.48550/ARXIV.2010.07416. Available at `https://arxiv.org/abs/2010.07416v2`.

[10] Tobias Fritz, Tomáš Gonda, Paolo Perrone & Eigil Fjeldgren Rischel (2023): *Representable Markov categories and comparison of statistical experiments in categorical probability.* *Theoretical Computer Science* 961, p. 113896, doi:10.1016/j.tcs.2023.113896. Available at `https://doi.org/10.1016/j.tcs.2023.113896`.

[11] Ramakrishna Gurajala, Praveen B. Choppala, James Stephen Meka & Paul D. Teal (2021): *Derivation of the Kalman filter in a Bayesian filtering perspective.* In: *2nd International Conference on Range Technology (ICORT)*, pp. 1–5, doi:10.1109/ICORT52730.2021.9581918.

[12] B. Jacobs (2020): *A channel-based perspective on conjugate priors.* *Mathematical Structures in Computer Science* 30(1), pp. 44–61, doi:10.1017/s0960129519000082. Available at `https://doi.org/10.1017/s0960129519000082`.

[13] Bart Jacobs, Aleks Kissinger & Fabio Zanasi (2019): *Causal Inference by String Diagram Surgery.* In Mikołaj Bojańczyk & Alex Simpson, editors: *Foundations of Software Science and Computation Structures*, Springer International Publishing, Cham, pp. 313–329.

[14] Bart Jacobs & Sam Staton (2020): *De Finetti's Construction as a Categorical Limit.* In D. Petrişan & J. Rot, editors: *Coalgebraic Methods in Computer Science*, Springer International Publishing, pp. 90–111, doi:10.1007/978-3-030-57201-3_6. Available at `https://doi.org/10.1007/978-3-030-57201-3_6`.

[15] Rudolph Emil Kalman (1960): *A New Approach to Linear Filtering and Prediction Problems.* *Transactions of the ASME–Journal of Basic Engineering* 82(Series D), pp. 35–45.

[16] Kotaro Kamiya & John Welliaveetil (2021): *A category theory framework for Bayesian learning*, doi:10.48550/ARXIV.2111.14293. Available at `https://arxiv.org/abs/2111.14293`.

[17] David Jaz Myers (2022): *Categorical Systems Theory*. Unpublished book draft. Available at `http://davidjaz.com/Papers/DynamicalBook.pdf`.

[18] Alexandra Silva, Filippo Bonchi, Marcello Bonsangue & Jan Rutten (2013): *Generalizing determinization from automata to coalgebras*. Logical Methods in Computer Science Volume 9, Issue 1, doi:10.2168/lmcs-9(1:9)2013. Available at `https://doi.org/10.2168/lmcs-9(1:9)2013`.

[19] Toby St. Clere Smithe (2020): *Bayesian Updates Compose Optically*, doi:10.48550/ARXIV.2006.01631. Available at `https://arxiv.org/abs/2006.01631`.

[20] David I. Spivak (2019): *Generalized Lens Categories via functors* $\mathscr{C}^{\mathrm{op}} \to \mathsf{Cat}$, doi:10.48550/ARXIV.1908.02202. Available at `https://arxiv.org/abs/1908.02202`.

[21] Nicholas F. Travers & James P. Crutchfield (2011): *Equivalence of History and Generator Epsilon-Machines*, doi:10.48550/ARXIV.1111.4500. Available at `https://arxiv.org/abs/1111.4500`.

[22] Nathaniel Virgo, Martin Biehl & Simon McGregor (2021): *Interpreting Dynamical Systems as Bayesian Reasoners*. In: International Workshops of ECML PKDD 2021, Springer, pp. 726–762, doi:10.1007/978-3-030-93736-2_52. Available at `https://doi.org/10.1007/978-3-030-93736-2_52`.

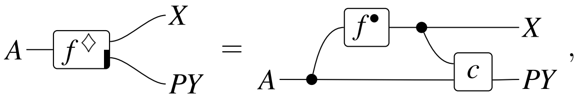
# A Proof Details

## A.1 Dist is strongly representable

For a morphism $f\colon A \to B$ in **Dist** we write $f(b \mid a)$ for the probability assigned to $b$ by the morphism $f$ when given $a$ as an input. We have that for a given $a$ there only finitely many elements of $b$ for which $f(b \mid a) > 0$, and we have $\sum_{b\in B} f(b \mid a) = 1$.

For a set $A$ the distribution object $PA$ is the set of all finitely supported probability distributions over $A$. In other words it is the set of functions $A \to [0,1]$ that satisfy the properties above, i.e. functions that have a finite support and sum to 1. The sampling map $\mathrm{samp}_A\colon PA \to A$ is given by $\mathrm{samp}_A(a \mid p) = p(a)$.

We now consider an arbitrary morphism $f\colon A \to X \otimes Y$ and a morphism $f^{\diamondsuit}\colon A \to X \otimes PY$ such that $f^{\diamondsuit}$ is deterministic given $X$ and such that eq. (5) holds. We can factor $f^{\diamondsuit}(x, p \mid a)$ as



(24)

or $f^{\bullet}(x \mid a)c(p \mid x, a)$, for some conditional $c$.

For $f^{\diamondsuit}$ to be deterministic given $X$ means that the conditional $c$ must be $f^{\bullet}$-generalised-almost-surely deterministic (eq. (4)), which in **Dist** means that $c(- \mid x, a)$ is a delta distribution whenever $f^{\bullet}(x \mid a) > 0$. For this to be true we must have that that whenever $a$ and $x$ are such that $f^{\bullet}(x \mid a) > 0$ there exists a distribution $p_{x,y} \in PY$ such that

$$f^{\diamondsuit}(x, p \mid a) = \begin{cases} f^{\bullet}(x \mid a) & \text{if } p = p_{a,x} \\ 0 & \text{otherwise.} \end{cases} \tag{25}$$

In **Dist**, eq. (5) (the definition of $f^{\diamondsuit}$) amounts to

$$\begin{aligned} f(x,y \mid a) &= \sum_{p\in PY} f^{\diamondsuit}(x,p \mid a)\,\mathrm{samp}_Y(y \mid p) \\ &= \sum_{p\in PY} f^{\diamondsuit}(x,p \mid a)p(y) \\ &= f^{\bullet}(x \mid a)p_{x,a}(y). \end{aligned} \tag{26}$$

To show that **Dist** is strongly representable we have to show that $p_{x,a}$ is uniquely defined whenever $f^{\bullet}(x \mid a) > 0$. But this follows immediately because we have, from eq. (26),

$$p_{x,a}(y) = \frac{f(x,y \mid a)}{f^{\bullet}(x \mid a)}, \tag{27}$$

which completes the proof.

Explicitly, the only choices for $f^{\diamondsuit}$ are those of the form

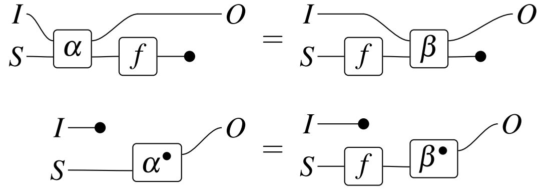
$$f^{\diamondsuit}(x,p \mid a) = \begin{cases} \left\{\begin{matrix} f^{\bullet}(x \mid a) & \text{if } p = \frac{f(x,-|a)}{f^{\bullet}(x|a)} \\ 0 & \text{otherwise} \end{matrix}\right\} & \text{if } f^{\bullet}(x \mid a) > 0, \\ \\ \text{arbitrary} & \text{otherwise.} \end{cases} \tag{28}$$

The arbitrary values are subject to the constraint that $\sum_{x,p} f(x,p \mid a) = 1$, as always. They occur only outside the support of $f^{\bullet}(- \mid a)$, which in **Dist** means that all possible choices for $f^{\diamondsuit}$ are $f^{\bullet}$-g.a.s. equal, as required.

We note that the last step in the proof, eq. (27), would not be valid in example 3.23 of [9], in which the probabilities are not real-valued.

## A.2 Comb machines morphisms commute with readout maps

We want to show that if $f\colon (S,\alpha) \to (T,\beta)$ is a morphism of comb machines, then $\alpha^{\bullet} = f \fatsemi \beta^{\bullet}$. By marginalising the definition of a morphism of comb machines and then substituing the definition of $\alpha^{\bullet}$ and $\beta^{\bullet}$ we have

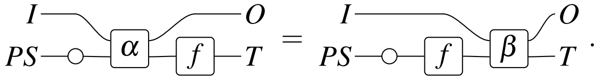


(29)

and the result follows.

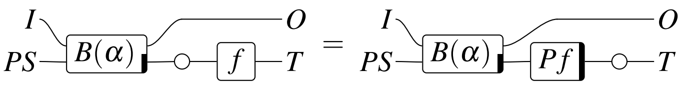
## A.3 *B* is a functor

Let $f$ be a morphism of comb machines $(S,\alpha) \to (T,\beta)$. To finish proving proposition 2.6 we have to show that $Pf$ is a morphism of unifilar machines $B(\alpha) \to B(\beta)$. Since $f$ is a morphism of comb machines we have



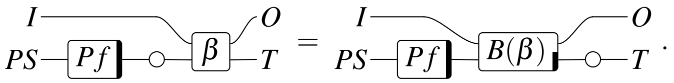
(30)

Since $\mathcal{C}$ is strongly representable there must be a unique morphism $\gamma\colon I \otimes PS \to O \times PT$ such that $\gamma$ is deterministic given $O$ and such that both sides are equal to $I \otimes PS \xrightarrow{\gamma} O \otimes PS \xrightarrow{\mathrm{id}_O \otimes \mathrm{samp}_T} O \otimes T$. But the left-hand side is equal to



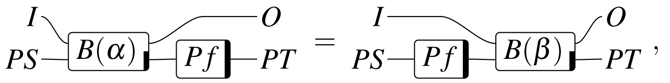
(31)

and the right-hand side is equal to



(32)

Hence (by uniqueness of $\gamma$) we must have



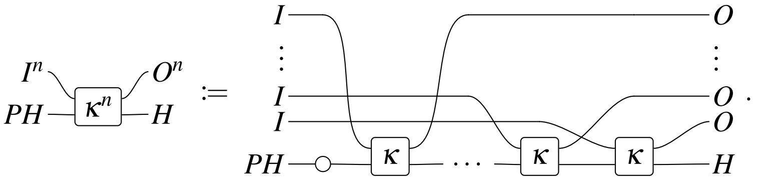
(33)

which completes the proof.

## A.4 Filtering on sequences

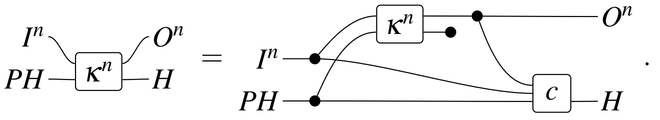
Our claim is that given a dynamical model in the form of a comb machine $(H, \kappa)$, an initial distribution over $H$, a sequence of inputs (an element of $I^n$), and an observed sequence of outputs (an element of $O^n$), the Bayesian filter (i.e. an update map for $B((H, \kappa))$) allows us to recover the posterior distribution over $H$, given the observations.

To make this formal let us define



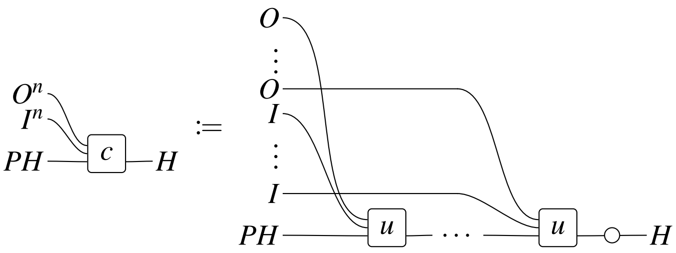
(34)

What we seek is a conditional of $\kappa^n$, that is $c\colon O^n \otimes I^n \otimes PH \to H$ such that



(35)

Then $c$, or rather $c^\square\colon O^n \times I^n \times PH \to PH$, is the map that takes the observed output sequence, the input sequence and returns the prior over $H$ to the posterior over $H$. Our claim is that such a conditional $c$ is given by composing $n$ instances of $u$ as follows,



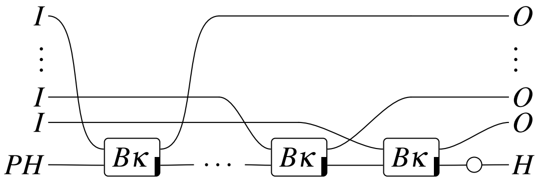
(36)

where $u$ is an update map for $B((H,\kappa))$.

To prove this we carry out the following calculation, which has several steps and involves some large diagrams but is otherwise straightforward. A similar proof was given by the author and colleagues in [22] (proposition 2 in appendix B.2), although with somewhat different definitions.

In following the proof it will be helpful to note that the last instances of $B\kappa$, $u$, etc. are treated somewhat differently than the rest. First we use the definition of $B$ (eq. (9)) $n$ times to obtain

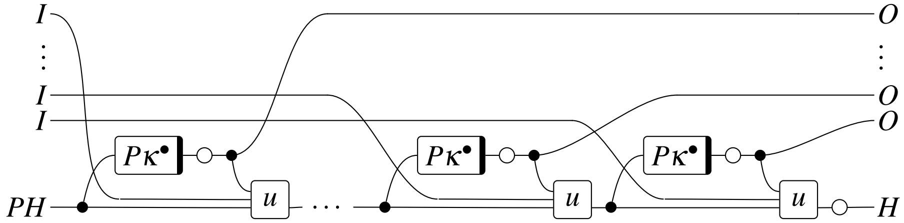
$\kappa^n =$



. (37)

We then split each instance of $B\kappa$ into a readout map and an unpdate map according to eq. (11) to obtain

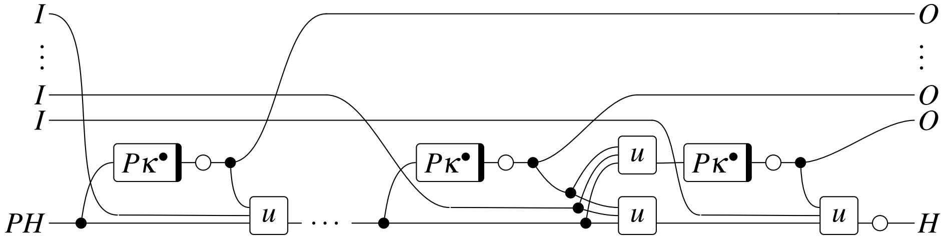
$\kappa^n =$



. (38)

Then using the fact that $u$ is $(P\kappa^\bullet \fatsemi \mathrm{samp})$-g.a.s. deterministic (an instance of eq. (4)),

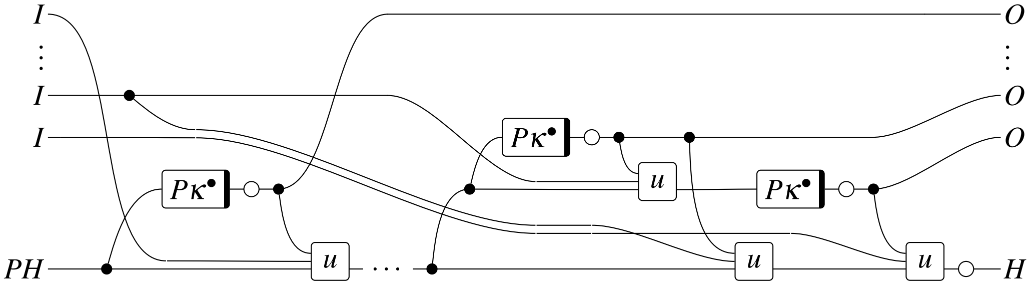
$\kappa^n =$



. (39)

Next we rearrange, using the comonoid axioms (see [7], definition 2.1).

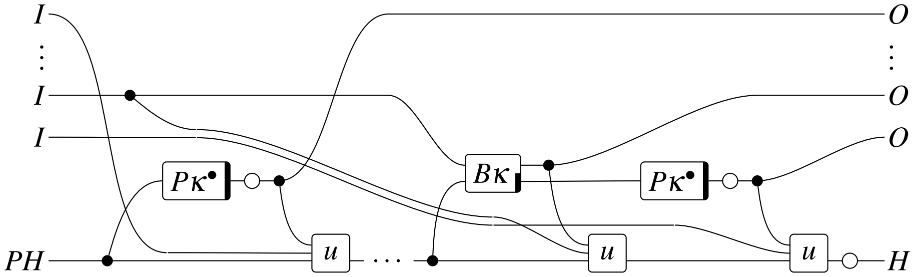
$\kappa^n =$



(40)

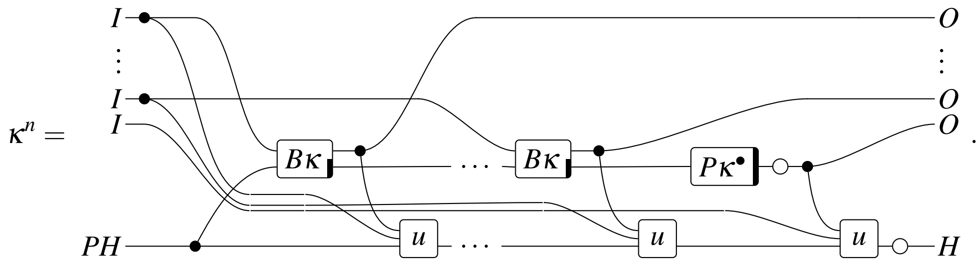
Applying eq. (9) again in reverse we obtain

$\kappa^n =$

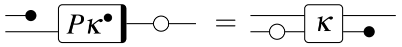


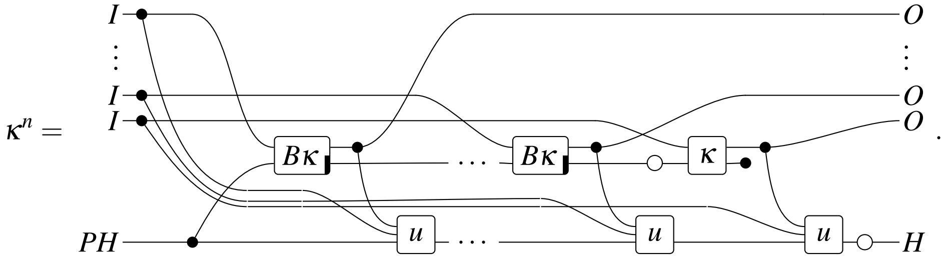
(41)

The next step is to repeat the steps in eqs. (39) to (41) for all the remaining instances of $u$ and $P\kappa^\bullet$ except for the last one, working backwards recursively along the chain, which ultimately results in



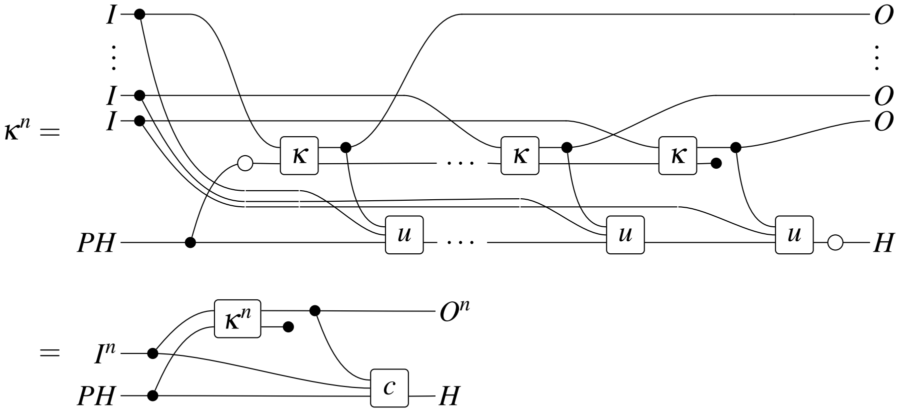
(42)

From naturality of samp and the definition of $\kappa^\bullet$ we have  , and so



(43)

Finally we apply eq. (9) again $n-1$ times to obtain



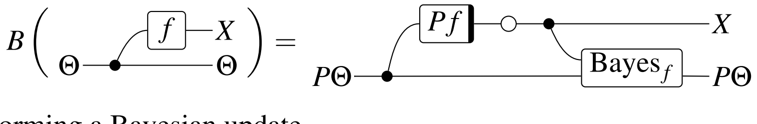
(44)

as desired.

A similar result holds for the Mealy machine case. The proof is similar but slightly simpler, because we don't need to use the comb condition, and as a result we don't need to treat the last instance of $\kappa$ differently from the others.

## A.5 Bayes$_f$ does Bayes

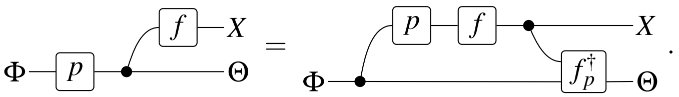
We want to show that the morphism Bayes$_f$ in the expression



(45)

can be seen as performing a Bayesian update.

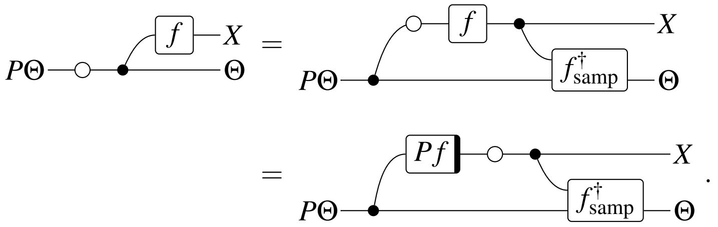
For this we recall the definition of a *Bayesian inverse* from [5], which we generalise slightly by allowing the prior to depend on a parameter. In a Markov category $\mathcal{C}$, given a morphism $f\colon \Theta \to X$ and a morphism $p\colon \Phi \to \Theta$ called the prior, we say that $f^\dagger_p\colon X \to \Theta$ is a *Bayesian inverse of* $f\colon \Theta \to X$ *with respect to* $p$ if



(46)

If $\mathcal{C}$ has conditionals then such Bayesian inverse exists for any $f$ and $p$. It is not necessarily unique, but like any conditional it is unique up to $(p \mathbin{⨾} f)$-g.a.s. equivalence.

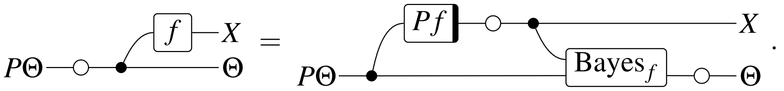
If $\mathcal{C}$ is representable we can take $\Phi = P\Theta$ and $p = \mathrm{samp}_\Theta$, yielding



(47)

The morphism $f^\dagger_{\mathrm{samp}}$ can be thought of as performing a Bayesian inversion of $f$ with respect to *any* prior, since it takes an element of $P\Theta$ as an input.

If $\mathcal{C}$ is strongly representable, then from the definition of $\mathrm{Bayes}_f$ we have



(48)

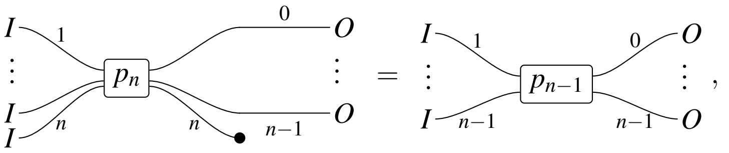
We conclude that up to $Pf \mathbin{⨾} \mathrm{samp}$-g.a.s. equivalence, $\mathrm{Bayes}_f$ is the same as $(f^\dagger_{\mathrm{samp}})^\square$, the deterministic version of $f^\dagger_{\mathrm{samp}}$. It takes as input a prior in $P\Theta$ and a data point from $X$, and gives as output the Bayesian posterior as an element of $P\Theta$.

## A.6 Dist has transducers

We want to show that in Dist, for any sets $I$ and $O$, there are a terminal objects of $\mathbf{UnifilarComb}(I,O)$ and $\mathbf{UnifilarMealy}(I,O)$. We will show this explicitly for $\mathbf{UnifilarComb}(I,O)$. For $\mathbf{UnifilarMealy}(I,O)$ the proof is similar.

We will need the definition of a controlled stochastic process. This is a classical idea, but the category theoretic definition we give is similar to definition 9.12 of [6]. For further generalisations with a slightly different flavour, see section 7 of [7].

**Definition A.1** (controlled stochastic process)**.** In a Markov category $\mathcal{C}$, we define an *output-first controlled stochastic process* with input space $I$ and output space $O$ as a family of morphisms $p_n\colon I^{n-1} \to O^n$ for $n \geq 1$, subject to the condition that



(49)

where the labels on the wires represent the indexes of the inputs and outputs. (Note that the indices for the inputs start from 1 while the indices for the outputs start from 0, so that $p_n$ has $n-1$ inputs and $n$

outputs. We use this convention because we consider the first output to occur “at time 0,” before the first input.) An *input-first controlled stochastic process* is defined similarly, but with the outputs indexed starting from 1 instead of 0, so that $p_n$ has type $I^{n-1} \to O^{n-1}$.

When we say “controlled stochastic process” without qualification, we mean an output-first controlled stochastic process. The condition says both that the family of distributions has to be consistent with each other, and that each output can only depend on inputs that were received prior to it.

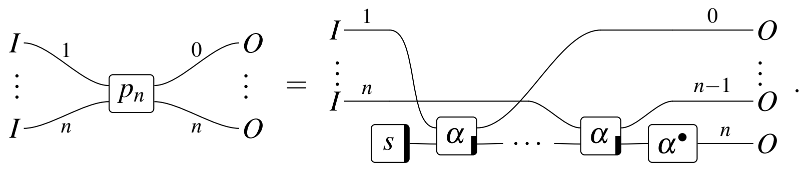
We now give a more precise version of proposition 3.1.

**Proposition A.2** (**Dist** has transducer objects)**.** *In* **Dist***, the terminal object* $(\omega, T)$ *of* **UnifilarComb**$(I,O)$ *exists and is as follows.* $T$ *is the set of all output-first controlled stochastic processes (in* **Dist***).* $\omega$ *is composed of the following readout and update maps: the readout map sends a controlled stochastic process* $p$ *to the distribution* $p_1$*, which is a distribution over* $O$ *with no input. Given* $i \in I$*,* $o \in O$ *and a controlled stochastic process* $p$*, the update map sends* $(i,o,p)$ *to a delta distribution concentrated on a new controlled stochastic process* $p^{i,o}$ *given by*

$$p^{i,o}_n(o_0,\ldots,o_n \mid i_1,\ldots,i_n) = \frac{1}{p_1(o)} p_{n+1}(o,o_0,\ldots,o_n \mid i,i_1,\ldots,i_n) \tag{50}$$

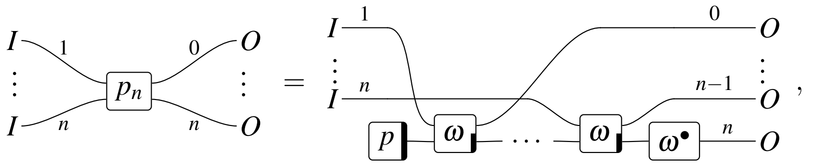
*if* $p_1(o) > 0$*, and to some arbitrary distribution over controlled stochastic processes otherwise. (As such it is defined up to the appropriate generalised-almost-surely condition.)*

*Proof.* We will show that given a unifilar machine $(S,\alpha)$ and a state $s \in S$, under any morphism of unifilar machines $(S,\alpha) \to (T,\omega)$, the state $s$ must map to the controlled stochastic process given by



(51)

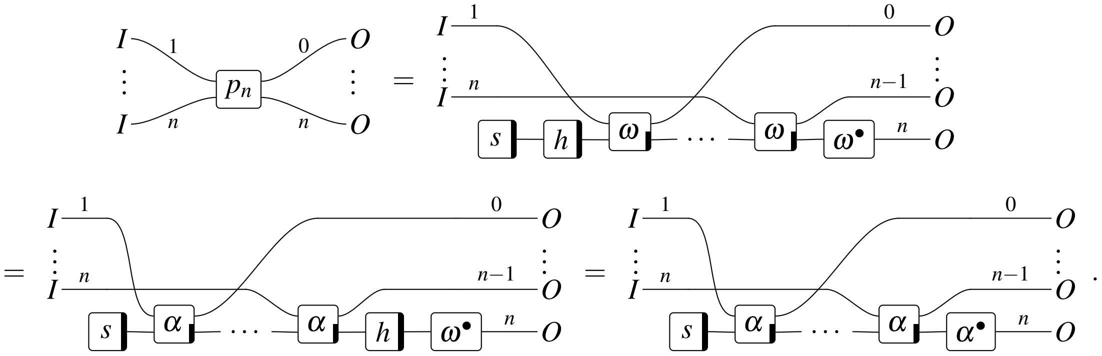
It is straightforward to show inductively that for $p \in T$ we have



(52)

where $\omega^\bullet$ is the readout map of $(T,\omega)$.

Now suppose we are given a unifilar machine $(S,\alpha)$, and write $\alpha^\bullet \colon S \to O$ for its readout map. Consider a map of unifilar machines $h\colon (S,\alpha) \to (T,\omega)$. Our goal is to show that such a map always exists and is uniquely defined. Let $s \in S$ be state of $\alpha$ and let $p = h(s)$ be the transducer that it maps to under $h$. We then calculate



(53)

The second equality is by induction, moving $h$ to the right across the chain of $n$ morphisms.

This leaves us with exactly one choice for $p = h(s)$ for each $s \in S$, and we conclude that the map $h\colon (S,\alpha) \to (T,\omega)$ is unique.

□

The update map of $(T,\omega)$ performs Bayesian conditioning: it returns a new map from input sequences to output sequences, formed by fixing the first input and conditioning on the first output.

A similar result holds for **UnifilarMealy**$(I,O)$. The main differences are that we use input-first controlled stochastic processes instead of output-first ones, and the comb condition is not needed in the proof.