

Lawvere Categories as Composed PROPs

Filippo Bonchi¹(✉), Pawel Sobocinski², and Fabio Zanasi³

¹ CNRS, École Normale Supérieure de Lyon, Lyon, France

`filippo.bonchi@ens-lyon.fr`

² University of Southampton, Southampton, UK

³ Radboud University Nijmegen, Nijmegen, The Netherlands

Abstract. PROPs and Lawvere categories are related notions adapted to the study of algebraic structures borne by an object in a category, but whereas PROPs are symmetric monoidal, Lawvere categories are cartesian. This paper formulates the connection between the two notions using Lack’s technique for composing PROPs via distributive laws. We show Lawvere categories can be seen as resulting from a distributive law of two PROPs — one expressing the algebraic structure in linear form and the other expressing the ability of copying and discarding variables.

1 Introduction

PROPs [28] are symmetric monoidal categories with objects the natural numbers. In the last two decades, they have become increasingly popular as an environment for the study of diagrammatic formalisms from diverse branches of science in a compositional, resource sensitive fashion. Focussing on computer science, they have recently featured in algebraic approaches to Petri nets [11, 35], bigraphs [12], quantum circuits [15], and signal flow graphs [1, 5, 7, 20].

PROPs describe both the *syntax* and the *semantics* of diagrams, with the interpretation expressed as a PROP morphism $[\cdot]: \mathbf{Syntax} \rightarrow \mathbf{Semantics}$. Typically, \mathbf{Syntax} is freely generated by a signature Σ of operations with arbitrary arity/coarity and can be composed sequentially and in parallel. Thus diagram syntax—which we refer to as Σ -terms—is inherently 2-dimensional: the term structure is that of directed acyclic graphs, rather than trees, as in the familiar case of operations with coarity 1. A crucial aspect is *linearity*: variables in Σ -terms cannot be copied nor discarded.

It is often useful to axiomatise the equivalence induced by $[\cdot]$ by means of a set of equations E , and then study the theory (Σ, E) . For PROPs, completeness proofs typically provide a serious challenge, involving the retrieval of a normal form for Σ -terms modulo E . The difficulty can be drastically reduced by exploiting certain operations on PROPs: an example of this modular methodology is provided by [5, 7], where the PROP operations of sum and composition are crucial for giving a sound and complete axiomatization of signal flow diagrams.

Sum is just the coproduct in the category of PROPs. Whenever two PROPs \mathcal{T}_1 and \mathcal{T}_2 can be presented by the theories (Σ_1, E_1) and (Σ_2, E_2) , then their sum $\mathcal{T}_1 + \mathcal{T}_2$ is presented by the disjoint union $(\Sigma_1 \uplus \Sigma_2, E_1 \uplus E_2)$.

Composition of PROPs is more subtle, as it requires certain compatibility conditions between the structure of \mathcal{T}_1 and \mathcal{T}_2 . Lack [25] describes this operation formally by means of distributive laws, seeing PROPs as monads in the 2-categorical sense of Street [36]. In a nutshell, a distributive law $\lambda: \mathcal{T}_1; \mathcal{T}_2 \rightarrow \mathcal{T}_2; \mathcal{T}_1$ of PROPs is a recipe for moving arrows of \mathcal{T}_1 past those of \mathcal{T}_2 . The resulting PROP $\mathcal{T}_2; \mathcal{T}_1$ enjoys a factorisation property: every arrow in $\mathcal{T}_2; \mathcal{T}_1$ decomposes as one of \mathcal{T}_2 followed by one of \mathcal{T}_1 . The graph of λ can be seen as a set of directed equations $E_\lambda := (\xrightarrow{\in \mathcal{T}_1, \mathcal{T}_2} \xrightarrow{\mathcal{T}_2, \mathcal{T}_1}) \approx (\xrightarrow{\mathcal{T}_2, \mathcal{T}_1} \xrightarrow{\in \mathcal{T}_1, \mathcal{T}_2})$ and $\mathcal{T}_2; \mathcal{T}_1$ is presented by the theory $(\Sigma_1 \uplus \Sigma_2, E_1 \uplus E_2 \uplus E_\lambda)$.

This work uses distributive laws of PROPs to characterise Lawvere categories¹, a well known class of structures adapted to the study of categorical universal algebra. The essential difference with PROPs is that Lawvere categories express *cartesian* theories (Σ, E) , i.e. where Σ only features operations with coarity 1 and E may include non-linear equations. Our starting observation is that the Lawvere category \mathcal{L}_Σ on a cartesian signature Σ exhibits a factorisation property analogous to the one of composed PROPs: arrows can always be decomposed as $\xrightarrow{\in \mathbf{Cm}} \xrightarrow{\in \mathcal{T}_\Sigma}$, where \mathbf{Cm} is the PROP of commutative comonoids, generated by a copy $1 \rightarrow 2$ and a discard $1 \rightarrow 0$ operation, and \mathcal{T}_Σ is the PROP freely generated by Σ . This factorisation represents cartesian Σ -terms by their syntactic tree — the \mathcal{T}_Σ -part — with the possibility of explicitly indicating variable-sharing among sub-terms — the \mathbf{Cm} -part. This simple observation leads us to the main result of the paper: for any cartesian signature Σ , there is a distributive law of PROPs $\lambda: \mathcal{T}_\Sigma; \mathbf{Cm} \rightarrow \mathbf{Cm}; \mathcal{T}_\Sigma$ which is presented by equations that express the naturality of copier and discarder; the resulting composed PROP $\mathbf{Cm}; \mathcal{T}_\Sigma$ is the Lawvere category \mathcal{L}_Σ .

By a quotient construction on distributive laws, it follows immediately that the above theorem holds more generally for any cartesian theory (Σ, E) where the set of axioms E only contains linear equations. For instance the Lawvere category of commutative monoids $\mathcal{L}_{\mathbf{Mn}}$ can be obtained by means of PROP composition, while the one $\mathcal{L}_{\mathbf{Gr}}$ of abelian groups cannot, because of the non-linear axiom $x \times x^{-1} = 1$. Obviously, one can still formulate $\mathcal{L}_{\mathbf{Gr}}$ as the quotient of the composite $\mathcal{L}_{\mathbf{Mn}}$ by adding this equation, see Example 4.6 below.

As a side remark, we observe that, by taking the sum $\mathbf{Cm} + \mathcal{T}_\Sigma$, rather than the composition, we are able to capture a different, well-known representation for cartesian Σ -terms, namely *term graphs*, which are acyclic graphs labeled over Σ . With respect to the standard tree representation, the benefit of term graphs is that the sharing of any common sub-term can be represented explicitly, making them particularly appealing for efficient rewriting algorithms, see e.g. [34] for a

¹ Usually called Lawvere *theories* in the literature: i.e. finite product categories with set of objects the natural numbers, where product on objects is addition [23, 27]. In order to keep the exposition uniform, we reserve the word *theory* for presentations and refer to the presentation (Σ, E) of a Lawvere theory as a *cartesian theory*.

survey on the subject. As shown in [16], Σ -term graphs are in 1-1 correspondence with the arrows of the free *gs-monoidal category* generated by Σ , a concept that actually amounts to forming the sum of PROPs $\mathbf{Cm} + \mathcal{T}_\Sigma$. Thus the only difference between term graphs and the representation of terms given by $\mathbf{Cm}; \mathcal{T}_\Sigma$ is in the validity of naturality of copier and discarder. Intuitively, a term where a resource is explicitly copied is not identified with the term where two copies appear separately: in short, copying is not natural.

Related Works. The motto “*cartesian terms = linear terms + copying and discarding*” inspired several papers exploiting the role of \mathbf{Cm} in Lawvere categories, see e.g. [14, 16, 26]. In our work, Lawvere categories feature as a distinguished example of a construction, PROP composition, that is increasingly important in many recent research threads [5, 6, 20, 32]. The significance of this exercise is two-fold. First, it gives a deeper understanding of the nature of Lawvere categories and how they formally relate to PROPs, by showing the provenance of the natural copy-discard structure. Second, our result provides a canonical means of defining a distributive law for freely generated PROPs, showing that the result of composition is a familiar algebraic notion and enjoys a *finite* axiomatisation.

The following construction, reported by Baez in [2], is close in spirit to our work. There is pseudo-adjunction between symmetric monoidal and categories with finite products

$$\mathbf{SMCat} \begin{array}{c} \xrightarrow{L} \\ \perp \\ \xleftarrow{R} \end{array} \mathbf{FPCat}$$

where R is the evident forgetful functor and L adds to any object of $\mathbb{C} \in \mathbf{SMCat}$ a natural copy-discard structure: natural diagonals and projections. Baez [2] states an equivalence between $RL(\mathbb{C})$ and $\mathbb{C} \otimes \mathbf{Cm}$, with the tensor \otimes defined by $\mathbf{SMCat}[\mathbb{C}_1 \otimes \mathbb{C}_2, \mathbb{C}_3] \simeq \mathbf{SMCat}[\mathbb{C}_1, \mathbf{SMCat}[\mathbb{C}_2, \mathbb{C}_3]]$. Indeed, our main construction, as well as being a distributive law, is also an instance of a tensor or Kronecker product of symmetric monoidal theories; a concept that has been explored in some detail in the cartesian setting of Lawvere categories, see e.g. [22].

Our work restricts attention to PROPs $\mathcal{T}_\Sigma \in \mathbf{SMCat}$ freely generated by a cartesian signature Σ : in this case, it is enough to add a copy-discard structure for the object 1 and $RL(\mathcal{T}_\Sigma)$ coincides with PROP composition $\mathbf{Cm}; \mathcal{T}_\Sigma$. Our perspective exploiting distributive laws has the advantage of providing a *finite* presentation in terms of the naturality axioms.

It is also worth mentioning that the relationship between symmetric monoidal and cartesian structures is central in the categorical semantics of linear logic; in this perspective, the presence of \mathbf{Cm} allows to interpret the structural rules of contraction and weakening — see e.g. [24, 30].

Prerequisites and Notation. We assume familiarity with the basics of category theory (see e.g. [10, 29]), the definition of symmetric strict monoidal category [29, 33] (often abbreviated as SMC) and of bicategory [3, 10]. We write $f; g: a \rightarrow c$ for composition of $f: a \rightarrow b$ and $g: b \rightarrow c$ in a category \mathbb{C} , and $\mathbb{C}[a, b]$ for the

$$\begin{aligned}
& (t_1 ; t_3) \oplus (t_2 ; t_4) = (t_1 \oplus t_2) ; (t_3 \oplus t_4) \\
& (t_1 ; t_2) ; t_3 = t_1 ; (t_2 ; t_3) \quad id_n ; c = c = c ; id_m \\
& (t_1 \oplus t_2) \oplus t_3 = t_1 \oplus (t_2 \oplus t_3) \quad id_0 \oplus t = t = t \oplus id_0 \\
& \sigma_{1,1} ; \sigma_{1,1} = id_2 \quad (t \oplus id_z) ; \sigma_{m,z} = \sigma_{n,z} ; (id_z \oplus t)
\end{aligned}$$

Fig. 1. Axioms of symmetric strict monoidal categories for a PROP \mathcal{T} .

hom-set of arrows $a \rightarrow b$. It will be sometimes convenient to indicate an arrow $f: a \rightarrow b$ of \mathbb{C} as $x \xrightarrow{f \in \mathbb{C}} y$ or also $\xrightarrow{\in \mathbb{C}}$, if names are immaterial. For \mathbb{C} an SMC, \oplus is its monoidal product, with unit object I , and $\sigma_{a,b}: a \oplus b \rightarrow b \oplus a$ is the symmetry associated with $a, b \in \mathbb{C}$.

2 PROPS

Our exposition is founded on *symmetric monoidal theories*: specifications for algebraic structures borne by objects in a symmetric monoidal category.

Definition 2.1. A (*one-sorted*) symmetric monoidal theory (*SMT*) is a pair (Σ, E) consisting of a signature Σ and a set of equations E . The signature Σ is a set of generators $o: n \rightarrow m$ with arity n and coarity m . The set of Σ -terms is obtained by composing generators in Σ , the unit $id: 1 \rightarrow 1$ and the symmetry $\sigma_{1,1}: 2 \rightarrow 2$ with $;$ and \oplus . This is a purely formal process: given Σ -terms $t: k \rightarrow l$, $u: l \rightarrow m$, $v: m \rightarrow n$, one constructs new Σ -terms $t ; u: k \rightarrow m$ and $t \oplus v: k + n \rightarrow l + n$. The set E of equations contains pairs $(t, t': n \rightarrow m)$ of Σ -terms with the same arity and coarity.

The categorical concept associated with symmetric monoidal theories is the notion of PROP (**product and permutation category** [28]).

Definition 2.2. A PROP is a symmetric strict monoidal category with objects the natural numbers, where \oplus on objects is addition. Morphisms between PROPs are strict symmetric identity-on-objects monoidal functors: PROPs and their morphisms form the category **PROP**. We call a sub-PROP a sub-category of a PROP which is also a PROP; i.e. the inclusion functor is a PROP morphism.

The PROP \mathcal{T} freely generated by an SMT (Σ, E) has as its set of arrows $n \rightarrow m$ the set of Σ -terms $n \rightarrow m$ taken modulo the laws of symmetric strict monoidal categories — Fig. 1 — and the smallest congruence (with respect to $;$ and \oplus) containing the equations $t = t'$ for any $(t, t') \in E$.

There is a natural graphical representation for arrows of a PROP as string diagrams, which we now sketch, referring to [33] for the details. A Σ -term $n \rightarrow m$ is pictured as a box with n ports on the left and m ports on the right. Composition via $;$ and \oplus are rendered graphically by horizontal and vertical juxtaposition of boxes, respectively.

$$t; s \text{ is drawn } \boxed{\begin{array}{|c|} \hline t \\ \hline s \\ \hline \end{array}} \quad t \oplus s \text{ is drawn } \boxed{\begin{array}{|c|} \hline t \\ \hline s \\ \hline \end{array}}. \quad (1)$$

In any SMT there are specific Σ -terms generating the underlying symmetric monoidal structure: these are $id_1: 1 \rightarrow 1$, represented as $\boxed{}$, the symmetry $\sigma_{1,1}: 1 + 1 \rightarrow 1 + 1$, represented as $\boxed{\text{X}}$, and the unit object for \oplus , that is, $id_0: 0 \rightarrow 0$, whose representation is an empty diagram $\boxed{}$. Graphical representation for arbitrary identities id_n and symmetries $\sigma_{n,m}$ are generated according to the pasting rules in (1).

Example 2.3.

- (a) We write (Σ_M, E_M) for the SMT of *commutative monoids*. The signature Σ_M contains a *multiplication* $\boxed{\text{D}}$: $2 \rightarrow 1$ and a *unit* $\boxed{\text{U}}$: $0 \rightarrow 1$. Equations E_M assert associativity (A1), commutativity (A2) and unitality (A3).

$$\boxed{\text{D}} = \boxed{\text{D}} \quad (\text{A1}) \quad \boxed{\text{D}} = \boxed{\text{D}} \quad (\text{A2}) \quad \boxed{\text{U}} = \boxed{} \quad (\text{A3})$$

We call Mn the PROP freely generated by the SMT (Σ_M, E_M) .

- (b) We also introduce the SMT (Σ_C, E_C) of *cocommutative comonoids*. The signature Σ_C consists of a *comultiplication* $\boxed{\text{C}}$: $1 \rightarrow 2$ and a *counit* $\boxed{\text{CU}}$: $1 \rightarrow 0$. E_C is the following set of equations.

$$\boxed{\text{C}} = \boxed{\text{C}} \quad (\text{A4}) \quad \boxed{\text{C}} = \boxed{\text{C}} \quad (\text{A5}) \quad \boxed{\text{CU}} = \boxed{} \quad (\text{A6})$$

We call Cm the PROP freely generated by (Σ_C, E_C) . Modulo the white vs. black colouring, string diagrams of Cm can be seen as those of Mn “reflected about the y -axis”. This observation yields $\text{Cm} \cong \text{Mn}^{op}$.

- (c) The PROP \mathbf{B} of (commutative/cocommutative) *bialgebras* is generated by the theory $(\Sigma_M \uplus \Sigma_C, E_M \uplus E_C \uplus B)$, where B is the following set of equations.

$$\boxed{\text{D}} = \boxed{\text{C}} \quad (\text{A7}) \quad \boxed{\text{D}} = \boxed{\text{C}} \quad (\text{A8})$$

$$\boxed{\text{U}} = \boxed{\text{CU}} \quad (\text{A9}) \quad \boxed{\text{U}} = \boxed{} \quad (\text{A10})$$

Remark 2.4 (Models of a PROP). The assertion that (Σ_M, E_M) is the SMT of *commutative monoids*—and similarly for other SMTs appearing in our exposition—can be made precise using the notion of *model* (sometimes also called *algebra*) of a PROP. Given a strict symmetric monoidal category \mathbb{C} , a model of a PROP \mathcal{T} in \mathbb{C} is a symmetric strict monoidal functor $\mathcal{F}: \mathcal{T} \rightarrow \mathbb{C}$. Then $\text{LinMod}(\mathcal{T}, \mathbb{C})$ is the category of models of \mathcal{T} in \mathbb{C} and natural transformations between them.

Turning to commutative monoids, there is a category $\text{Monoid}(\mathbb{C})$ whose objects are the commutative monoids in \mathbb{C} , i.e., objects $x \in \mathbb{C}$ equipped with

arrows $x \oplus x \rightarrow x$ and $I \rightarrow x$, satisfying the usual equations. Given any model $\mathcal{F}: \mathbf{Mn} \rightarrow \mathbb{C}$, it follows that $\mathcal{F}(1)$ is a commutative monoid in \mathbb{C} : this yields a functor $\mathbf{LinMod}(\mathbf{Mn}, \mathbb{C}) \rightarrow \mathbf{Monoid}(\mathbb{C})$. Saying that (Σ_M, E_M) is the SMT of commutative monoids means that this functor is an equivalence natural in \mathbb{C} . We shall not focus on models as they are not central in our developments and refer the reader to [19, 25] for more information.

Example 2.3 only shows PROPs freely generated from an algebraic specification. However, one can also define PROPs in a more direct manner, without relying on SMTs. Two basic examples will be useful for our exposition:

- the PROP \mathbf{F} whose arrows $n \rightarrow m$ are functions from \bar{n} to \bar{m} , where $\bar{n} = \{0, 1, \dots, n-1\}$.
- the PROP \mathbf{P} whose arrows $n \rightarrow m$ exist only if $n = m$, in which case they are the permutations on \bar{n} .

This kind of definition is often useful as a different, more concrete perspective on PROPs that arise from symmetric monoidal theories. For instance, \mathbf{F} is *presented* by the theory of commutative monoids, in the sense that \mathbf{F} and \mathbf{Mn} are isomorphic PROPs: one can consider a string diagram $t \in \mathbf{Mn}[n, m]$ as the graph of a function of type $\{1, \dots, n\} \rightarrow \{1, \dots, m\}$. For instance, $\square \oplus \square : 2 \rightarrow 2$ describes the function $f: \{1, 2\} \rightarrow \{1, 2\}$ mapping both elements to 1. By duality, $\mathbf{Cm} \cong \mathbf{F}^{op}$, that is, \mathbf{F}^{op} is presented by the theory of commutative comonoids.

Similarly, \mathbf{P} provides a concrete description of the theory (\emptyset, \emptyset) with empty signature and no equations. It is the initial object in the category **PROP**.

3 PROP Composition

A basic operation on SMTs (Σ, E) and (Σ', E') is to take their *sum* $(\Sigma \uplus \Sigma', E \uplus E')$. In **PROP**, the PROP generated by $(\Sigma \uplus \Sigma', E \uplus E')$ is the *coproduct* $\mathcal{T} + \mathcal{S}$ of the PROP \mathcal{T} generated by (Σ, E) and \mathcal{S} , generated by (Σ', E') .

The sum $\mathcal{T} + \mathcal{S}$ is the least interesting way of combining theories, because there are no equations that express compatibility between the algebraic structures in \mathcal{T} and \mathcal{S} . This is a standard pattern in algebra: e.g. a ring is given by a monoid and an abelian group, subject to equations that ensure that the former distributes over the latter. Similarly, the equations of bialgebras (Example 2.3) describe the interplay of a monoid and a comonoid. Ordinary functions, which can always be decomposed as a surjection followed by an injection, are another example.

In [25] Lack shows how these phenomena can be uniformly described as the operation of composing PROPs. The conceptual switch is to understand PROPs as monads, and their composition as a distributive law. These monads live in a certain bicategory [3], as in the classical work by Street [36]².

² Actually, Street worked in a 2-category, but the same theory can be developed in any bicategory with the obvious, minor modifications [25, §3.1].

Definition 3.1. A monad on an object x of a bicategory \mathfrak{B} is a 1-cell $\mathcal{F} : x \rightarrow x$ with 2-cells $\eta^{\mathcal{F}} : id_x \rightarrow \mathcal{F}$ and $\mu^{\mathcal{F}} : \mathcal{F}; \mathcal{F} \rightarrow \mathcal{F}$ (called the unit and the multiplication respectively) making the following diagrams—in which we suppress the associativity isomorphisms—commute.

$$\begin{array}{ccc}
 \mathcal{F} & \xrightarrow{\mathcal{F}\eta^{\mathcal{F}}} & \mathcal{F}; \mathcal{F} & \xleftarrow{\eta^{\mathcal{F}}\mathcal{F}} & \mathcal{F} \\
 & \searrow & \mu^{\mathcal{F}} \downarrow & \swarrow & \\
 & id & \mathcal{F} & \leftarrow id &
 \end{array}
 \quad (2)$$

$$\begin{array}{ccc}
 \mathcal{F}; \mathcal{F}; \mathcal{F} & \xrightarrow{\mathcal{F}\mu^{\mathcal{F}}} & \mathcal{F}; \mathcal{F} \\
 \mu^{\mathcal{F}}\mathcal{F} \downarrow & & \downarrow \mu^{\mathcal{F}} \\
 \mathcal{F}; \mathcal{F} & \xrightarrow{\mu^{\mathcal{F}}} & \mathcal{F}
 \end{array}
 \quad (3)$$

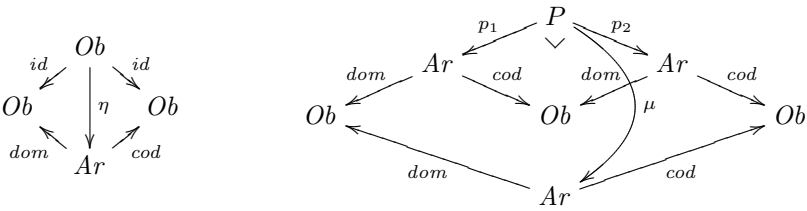
A morphism between monads $x \xrightarrow{\mathcal{F}} x$ and $x \xrightarrow{\mathcal{G}} x$ is a 2-cell $\theta : \mathcal{F} \rightarrow \mathcal{G}$ making the following diagrams commute³.

$$\begin{array}{ccc}
 id_x & \searrow \eta^{\mathcal{G}} & \\
 \eta^{\mathcal{F}} \downarrow & & \theta \\
 \mathcal{F} & \xrightarrow{\theta} & \mathcal{G}
 \end{array}
 \quad (4)$$

$$\begin{array}{ccc}
 \mathcal{F}; \mathcal{F} & \xrightarrow{\theta\theta} & \mathcal{G}; \mathcal{G} \\
 \mu^{\mathcal{F}} \downarrow & & \downarrow \mu^{\mathcal{G}} \\
 \mathcal{F} & \xrightarrow{\theta} & \mathcal{G}
 \end{array}
 \quad (5)$$

An epimorphic monad morphism is called a monad quotient.

For $\mathfrak{B} = \mathbf{Cat}$, the above definition yields the standard notion of monad as an endofunctor with a pair of natural transformations. Something interesting happens for the case of the bicategory $\mathfrak{B} = \mathbf{Span}(\mathbf{Set})$ whose objects are sets, 1-cells are spans of functions (with composition defined by pullback) and 2-cells are span morphisms: monads in $\mathbf{Span}(\mathbf{Set})$ are precisely the small categories. Indeed, a monad (\mathcal{F}, η, μ) there consists of a span $Ob \xleftarrow{dom} Ar \xrightarrow{cod} Ob$, which yields a set Ob of objects, one Ar of arrows and domain/codomain maps $Ar \rightrightarrows Ob$. The unit $\eta : id \rightarrow \mathcal{F}$ is a span morphism associating an identity arrow to each object (below left). The multiplication $\mu : \mathcal{F}; \mathcal{F} \rightarrow \mathcal{F}$ is a span morphism defining composition for any two arrows $a \xrightarrow{f} b \xrightarrow{g} c$ in Ar (below right).



By thinking of categories as monads, one can define the composition of categories with the same set of objects as monad composition by a *distributive law* in $\mathbf{Span}(\mathbf{Set})$. This phenomenon is studied in [31].

Definition 3.2. Let $(\mathcal{F}, \eta^{\mathcal{F}}, \mu^{\mathcal{F}})$, $(\mathcal{G}, \eta^{\mathcal{G}}, \mu^{\mathcal{G}})$ be monads in a bicategory \mathfrak{B} on the same object. A distributive law of \mathcal{F} over \mathcal{G} is a 2-cell $\lambda : \mathcal{F}; \mathcal{G} \rightarrow \mathcal{G}; \mathcal{F}$ in \mathfrak{B} making the following diagrams—in which we again omit associativity—commute.

³ A notion of morphism can be defined also between monads on different objects, like in [36]. We will not need that level of generality here.

$$\begin{array}{ccc}
 \begin{array}{ccc}
 \mathcal{F} & & \\
 \mathcal{F}\eta^{\mathcal{G}} \downarrow & \searrow \eta^{\mathcal{G}\mathcal{F}} & \\
 \mathcal{F}; \mathcal{G} & \xrightarrow{\lambda} & \mathcal{G}; \mathcal{F} \\
 \eta^{\mathcal{F}\mathcal{G}} \uparrow & \nearrow \mathcal{G}\eta^{\mathcal{F}} & \\
 \mathcal{G} & &
 \end{array} & &
 \begin{array}{ccc}
 \mathcal{F}; \mathcal{G}; \mathcal{G} & \xrightarrow{\lambda^{\mathcal{G}}} & \mathcal{G}; \mathcal{F}; \mathcal{G} & \xrightarrow{\mathcal{G}\lambda} & \mathcal{G}; \mathcal{G}; \mathcal{F} \\
 \mathcal{F}\mu^{\mathcal{G}} \downarrow & & & & \downarrow \mu^{\mathcal{G}\mathcal{F}} \\
 \mathcal{F}; \mathcal{G} & \xrightarrow{\lambda} & \mathcal{G}; \mathcal{F} & & \\
 \mu^{\mathcal{F}\mathcal{G}} \uparrow & & & & \uparrow \mathcal{G}\mu^{\mathcal{F}} \\
 \mathcal{F}; \mathcal{F}; \mathcal{G} & \xrightarrow{\mathcal{F}\lambda} & \mathcal{F}; \mathcal{G}; \mathcal{F} & \xrightarrow{\lambda^{\mathcal{F}}} & \mathcal{G}; \mathcal{F}; \mathcal{F}
 \end{array}
 \end{array} \quad (6)$$

A distributive law $\lambda: \mathcal{F}; \mathcal{G} \rightarrow \mathcal{G}; \mathcal{F}$ yields a monad $\mathcal{G}; \mathcal{F}$ with the following unit and multiplication:

$$\begin{aligned}
 \eta^{\mathcal{G}; \mathcal{F}} &: id \xrightarrow{\eta^{\mathcal{F}}} \mathcal{F} \xrightarrow{\eta^{\mathcal{G}\mathcal{F}}} \mathcal{G}; \mathcal{F} \\
 \mu^{\mathcal{G}; \mathcal{F}} &: \mathcal{G}; \mathcal{F}; \mathcal{G}; \mathcal{F} \xrightarrow{\mathcal{G}\lambda^{\mathcal{F}}} \mathcal{G}; \mathcal{G}; \mathcal{F}; \mathcal{F} \xrightarrow{\mu^{\mathcal{G}\mathcal{F}\mathcal{F}}} \mathcal{G}; \mathcal{F}; \mathcal{F} \xrightarrow{\mathcal{G}\mu^{\mathcal{F}}} \mathcal{G}; \mathcal{F}
 \end{aligned} \quad (7)$$

Let us verify how the abstract definition works for the case of categories. Pick categories \mathbb{C} and \mathbb{D} with the same set Ob of objects, seen as monads $Ob \xleftarrow{dom_{\mathbb{C}}} Ar_{\mathbb{C}} \xrightarrow{cod_{\mathbb{C}}} Ob$ and $Ob \xleftarrow{dom_{\mathbb{D}}} Ar_{\mathbb{D}} \xrightarrow{cod_{\mathbb{D}}} Ob$ in $\mathbf{Span}(\mathbf{Set})$. A distributive law $\lambda: \mathbb{C}; \mathbb{D} \rightarrow \mathbb{D}; \mathbb{C}$ is a span morphism

mapping composable pairs $a \xrightarrow{\in \mathbb{C}} \xrightarrow{\in \mathbb{D}} b$ to composable pairs $a \xrightarrow{\in \mathbb{D}} \xrightarrow{\in \mathbb{C}} b$. As described in (7), λ allows to define a monad structure on $\mathbb{D}; \mathbb{C}$. That means, λ yields a category $\mathbb{D}; \mathbb{C}$ whose arrows $a \rightarrow b$ are composable pairs $a \xrightarrow{\in \mathbb{D}} \xrightarrow{\in \mathbb{C}} b$ of arrows of \mathbb{D} , \mathbb{C} and composition is defined as

$$\left(a \xrightarrow{f \in \mathbb{D}} \xrightarrow{g \in \mathbb{C}} b \right); \left(b \xrightarrow{f' \in \mathbb{D}} \xrightarrow{g' \in \mathbb{C}} c \right) := \left(a \xrightarrow{f \in \mathbb{D}} \lambda \left(\xrightarrow{g \in \mathbb{C}} \xrightarrow{f' \in \mathbb{D}} \right) \xrightarrow{g' \in \mathbb{C}} c \right).$$

PROPs are understood as monads in the same sense that small categories are. The difference is that one needs to refine the bicategory of interest, in order for the composition of PROPs-monads to yield another PROP-monad and not an arbitrary small category. These refinements are in two steps. First, one takes the bicategory $\mathbf{Span}(\mathbf{Mon})$ whose 1-cells are spans in the category of *monoids*, instead of sets. This accounts for the monoidal structure — in fact, monads in $\mathbf{Span}(\mathbf{Mon})$ are small strict monoidal categories. The second refinement has the purpose of correctly account for the symmetry structure of PROPs: one

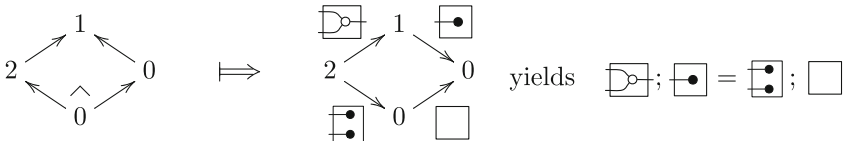
takes the bicategory $\mathbf{Bimod}(\mathbf{Span}(\mathbf{Mon}))$ whose objects are small strict monoidal categories and 1-cells are the *bimodules* in $\mathbf{Span}(\mathbf{Mon})$. PROPs are then monads on the object \mathbf{P} of $\mathbf{Bimod}(\mathbf{Span}(\mathbf{Mon}))$.

We shall gloss over further details about the exact formalisation of this observation, as it is out of the scope of this paper — we refer to [25] and [37, § 2.4] for the detailed definitions. The simpler setting of composition of mere categories should provide enough guidance to follow the rest of our exposition.

It is important for our purposes to remark how composition works for PROPs $\mathcal{T}_1, \mathcal{T}_2$ generated by SMTs, say (Σ_1, E_1) and (Σ_2, E_2) . The PROP $\mathcal{T}_1 ; \mathcal{T}_2$ induced by a distributive law $\lambda: \mathcal{T}_2 ; \mathcal{T}_1 \rightarrow \mathcal{T}_1 ; \mathcal{T}_2$ will also enjoy a presentation by generators and equations, consisting of the sum $(\Sigma_1 \uplus \Sigma_2, E_1 \uplus E_2)$ plus the equations E_λ arising from the the distributive law. The set E_λ is simply the graph of λ , now seen as a set of directed equations $(\xrightarrow{\in \mathcal{T}_2}, \xrightarrow{\in \mathcal{T}_1}) \approx (\xrightarrow{\in \mathcal{T}_1}, \xrightarrow{\in \mathcal{T}_2})$ telling how Σ_2 -terms modulo E_2 distribute over Σ_1 -terms modulo E_1 . In fortunate cases, it is possible to present E_λ by a simpler, or even finite, set of equations, thus giving a sensible axiomatisation of the compatibility conditions expressed by λ . This is the case for both examples considered below.

Example 3.3.

- (a) The PROP \mathbf{F} of functions can be described as the composite of PROPs for surjections and injections. Let \mathbf{In} be the PROP whose arrows $n \rightarrow m$ are injective functions from \bar{n} to \bar{m} . The PROP \mathbf{Su} of surjective functions is defined analogously. There is a distributive law $\lambda: \mathbf{In}; \mathbf{Su} \rightarrow \mathbf{Su}; \mathbf{In}$ defined by epi-mono factorisation: it maps a composable pair $\xrightarrow{\in \mathbf{In}} \xrightarrow{\in \mathbf{Su}}$ to a composable pair $\xrightarrow{\in \mathbf{Su}} \xrightarrow{\in \mathbf{In}}$ [25]. The resulting PROP $\mathbf{Su}; \mathbf{In}$ is isomorphic to \mathbf{F} because any function in \mathbf{F} can be uniquely factorised (up-to permutation) as a surjection followed by an injection. In more syntactic terms, using the isomorphism $\mathbf{F} \cong \mathbf{Mn}$, this result says that \mathbf{Mn} is the composite $\mathbf{Mu}; \mathbf{Un}$, where $\mathbf{Mu} \cong \mathbf{Su}$ is the PROP freely generated by the SMT $(\{\square\}, \{(A1), (A2)\})$ and $\mathbf{Un} \cong \mathbf{In}$ by the SMT $(\{\square\}, \emptyset)$. The distributive law $\lambda: \mathbf{In}; \mathbf{Su} \rightarrow \mathbf{Su}; \mathbf{In}$ is then presented by the remaining equation (A3) of \mathbf{Mn} , which indeed describes how the generator \square of \mathbf{Un} can be moved past the one \square of \mathbf{Mu} .
- (b) The composition of \mathbf{Cm} and \mathbf{Mn} yields the PROP \mathbf{B} of commutative bialgebras. First, because $\mathbf{Mn} \cong \mathbf{F}$ and $\mathbf{Cm} \cong \mathbf{F}^{op}$, we can express a distributive law $\lambda: \mathbf{Mn}; \mathbf{Cm} \rightarrow \mathbf{Cm}; \mathbf{Mn}$ as having the type $\mathbf{F}; \mathbf{F}^{op} \rightarrow \mathbf{F}^{op}; \mathbf{F}$. This amounts to saying that λ maps *cospans* $n \xrightarrow{f \in \mathbf{F}} \xleftarrow{g \in \mathbf{F}} m$ to *spans* $n \xleftarrow{p \in \mathbf{F}} \xrightarrow{q \in \mathbf{F}} m$. Defining this mapping via (chosen) pullback in \mathbf{F} satisfies the conditions of distributive laws [25]. One can now read the equations arising by the distributive law from pullback squares in \mathbf{F} . For instance:



where the second diagram is obtained from the pullback by applying the isomorphisms $\mathbf{F} \cong \mathbf{Mn}$ and $\mathbf{F}^{op} \cong \mathbf{Cm}$. In fact, Lack [25] shows that the equations presenting $\mathbf{Cm}; \mathbf{Mn}$ arise from (those of $\mathbf{Cm} + \mathbf{Mn}$ and) just four pullback squares, yielding equations (A7)–(A10). Therefore, $\mathbf{Cm}; \mathbf{Mn}$ is isomorphic to the PROP \mathbf{B} of bialgebras encountered in Example 2.3. Furthermore, these PROPs have a “concrete” descriptions as $\mathbf{F}^{op}; \mathbf{F}$. In the terminology of [3], one can see $\mathbf{F}^{op}; \mathbf{F}$ as the *classifying category* of the bicategory $\mathbf{Span}(\mathbf{F})$, obtained by identifying the isomorphic 1-cells and forgetting the 2-cells.

There is a tight relationship between distributive laws and factorisation systems. Distributive laws of small categories are in 1-1 correspondence with so-called *strict* factorisation systems [31], in which factorisations must be specified uniquely on the nose, rather than merely up-to isomorphism. Distributive laws of PROPs correspond instead to a more liberal kind of factorisation system, for which decompositions are up-to permutation. As this perspective will be useful later, we recall the following result from Lack [25].

Proposition 3.4 ([25], **Theorem 4.6**). *Let \mathbf{S} be a PROP and $\mathcal{T}_1, \mathcal{T}_2$ be sub-PROPs of \mathbf{S} . Suppose that each arrow $n \xrightarrow{f \in \mathbf{S}} m$ can be factorised as $n \xrightarrow{g_1 \in \mathcal{T}_1} \xrightarrow{g_2 \in \mathcal{T}_2} m$ uniquely up-to permutation, that is, for any other decomposition $n \xrightarrow{h_1 \in \mathcal{T}_1} \xrightarrow{h_2 \in \mathcal{T}_2} m$ of f , there exists permutation $\xrightarrow{p \in \mathbf{P}}$ such that the following diagram commutes.*

$$\begin{array}{ccc} & \nearrow^{g_1} & \\ & \uparrow^p & \searrow^{g_2} \\ & \xrightarrow{h_1} & \xrightarrow{h_2} \end{array} .$$

Then there exists a distributive law $\lambda: \mathcal{T}_2; \mathcal{T}_1 \rightarrow \mathcal{T}_1; \mathcal{T}_2$, defined by associating to a composable pair $\xrightarrow{f \in \mathcal{T}_2} \xrightarrow{g \in \mathcal{T}_1}$ the factorisation of $f; g$, yielding $\mathbf{S} \cong \mathcal{T}_1; \mathcal{T}_2$.

Quotient of a Distributive Law. Definition 3.1 introduced the notion of *quotient* $\theta: \mathcal{F} \rightarrow \mathcal{G}$ of a monad \mathcal{F} : the idea is that the monad \mathcal{G} is obtained by imposing additional equations on the algebraic theory described by \mathcal{F} . As one may expect, distributive laws are compatible with monad quotients, provided that the law preserves the newly added equations. This folklore result appears in various forms in the literature: [9] gives it for distributive laws of endofunctors over monads and [4, 8] for distributive laws of monads. All these references concern distributive laws in \mathbf{Cat} . For our purposes, it is useful to state the result for arbitrary bicategories.

Proposition 3.5. *Suppose that $\lambda: \mathcal{F}; \mathcal{H} \rightarrow \mathcal{H}; \mathcal{F}$ is a distributive law in a bicategory \mathfrak{B} , $\theta: \mathcal{F} \rightarrow \mathcal{G}$ a monad quotient and $\lambda': \mathcal{G}; \mathcal{H} \rightarrow \mathcal{H}; \mathcal{G}$ another 2-cell of \mathfrak{B} making the following diagram commute.*

$$\begin{array}{ccc} \mathcal{F}; \mathcal{H} & \xrightarrow{\theta\mathcal{H}} & \mathcal{G}; \mathcal{H} \\ \lambda \downarrow & & \downarrow \lambda' \\ \mathcal{H}; \mathcal{F} & \xrightarrow{\mathcal{H}\theta} & \mathcal{H}; \mathcal{G} \end{array} \tag{8}$$

Then λ' is a distributive law of monads.

Proof. The diagrams for compatibility of λ' with unit and multiplication of \mathcal{G} commute because θ is a monad morphism and (8) commutes. For compatibility of λ' with unit and multiplication of \mathcal{H} , one needs to use commutativity of (8) and the fact that θ is epi.

We remark that Proposition 3.5 holds also in the version in which one quotients the monad \mathcal{H} instead of \mathcal{F} . It is now useful to instantiate the result to the case of distributive laws of PROPs.

Proposition 3.6. *Let \mathcal{T} be the PROP freely generated by (Σ, E) and $E' \supseteq E$ be another set of equations on Σ -terms. Suppose that there exists a distributive law $\lambda: \mathcal{T}; \mathbf{S} \rightarrow \mathbf{S}; \mathcal{T}$ such that, if E' implies $c = d$, then $\lambda(\frac{c \in \mathcal{T}}{c \in \mathbf{S}} \rightarrow \frac{e \in \mathbf{S}}{e \in \mathbf{S}}) = \lambda(\frac{d \in \mathcal{T}}{d \in \mathbf{S}} \rightarrow \frac{e \in \mathbf{S}}{e \in \mathbf{S}})$. Then there exists a distributive law $\lambda': \mathcal{T}'; \mathbf{S} \rightarrow \mathbf{S}; \mathcal{T}'$ presented by the same equations as λ , i.e., $E_{\lambda'} = E_{\lambda}$.*

Proof. There is a PROP morphism $\theta: \mathcal{T} \rightarrow \mathcal{T}'$ defined by quotienting string diagrams in \mathcal{T} by E' . This is a monad quotient in the bicategory $\mathbf{Prof}(\mathbf{Mon})$ where PROPs are monads. We now define another 2-cell $\lambda': \mathcal{T}'; \mathbf{Cm} \rightarrow \mathbf{Cm}; \mathcal{T}'$ as follows: given $\frac{e \in \mathcal{T}'}{c \in \mathbf{Cm}} \rightarrow$, pick any $\frac{d \in \mathcal{T}}{d \in \mathbf{Cm}}$ such that $\theta(d) = e$ and let $\frac{c' \in \mathbf{Cm}}{d' \in \mathcal{T}} \rightarrow$ be $\lambda(\frac{d \in \mathcal{T}}{c \in \mathbf{Cm}})$. Define $\lambda'(\frac{e \in \mathcal{T}'}{c \in \mathbf{Cm}} \rightarrow)$ as $\frac{c' \in \mathbf{Cm}}{\theta(d') \in \mathcal{T}'}$. λ' is well-defined because, by assumption, if $\theta(d_1) = \theta(d_2)$ then E' implies that $d_1 = d_2$ and thus $\lambda(\frac{d_1}{c} \rightarrow) = \lambda(\frac{d_2}{c} \rightarrow)$.

Now, λ , λ' and θ satisfy the assumptions of Proposition 3.5. In particular, (8) commutes by definition of λ' in terms of λ and θ . The conclusion of Proposition 3.5 guarantees that λ' is a distributive law. By construction, λ' is presented by the same equations as λ .

We will see an application of Proposition 3.6 in the next section (Lemma 4.8).

4 Lawvere Categories as Composed PROPs

This section introduces and characterises Lawvere categories via a certain class of distributive laws of PROPs. As mentioned in the introduction, Lawvere categories are closely related to PROPs: the essential difference is that, whereas a Lawvere category is required to be a category with finite products — henceforth called a *cartesian* category, a PROP may carry any symmetric monoidal structure, not necessarily cartesian.

Just as PROPs, Lawvere categories can be also freely obtained by generators and equations. By analogy with symmetric monoidal theories introduced in Sect. 2, we organise this data as a *cartesian theory*: it simply amounts to the notion of equational theory that one typically finds in universal algebra, see e.g. [13].

Definition 4.1. A (one-sorted) cartesian theory (Σ, E) consists of a signature $\Sigma = \{o_1: n_1 \rightarrow 1, \dots, o_k: n_k \rightarrow 1\}$ and a set E of equations between cartesian Σ -terms, which are defined as follows:

- for each $i \in \mathbb{N}$, the variable x_i is a cartesian term;
- suppose $o: n \rightarrow 1$ is a generator in Σ and t_1, \dots, t_n are cartesian terms. Then $o(t_1, \dots, t_n)$ is a cartesian term.

The Lawvere category $\mathcal{L}_{(\Sigma, E)}$ freely generated by (Σ, E) is the category whose objects are the natural numbers and arrows $n \rightarrow m$ are lists $\langle t_1, \dots, t_m \rangle$ of cartesian Σ -terms quotiented by E , such that, for each t_i , only variables among x_1, \dots, x_n appear in t_i . Composition is by substitution:

$$\left(n \xrightarrow{\langle t_1, \dots, t_m \rangle} m \right); \left(m \xrightarrow{\langle s_1, \dots, s_z \rangle} z \right) = n \xrightarrow{\langle s_1[t_i/x_i | 1 \leq i \leq m], \dots, s_z[t_i/x_i | 1 \leq i \leq m] \rangle} z$$

where $t[t'/x]$ denotes the cartesian term t in which all occurrences of the variable x have been replaced with t' .

$\mathcal{L}_{(\Sigma, E)}$ is equipped with a product \times which is defined on objects by addition and on arrows by list concatenation and suitable renaming of variables:

$$\begin{aligned} \left(n \xrightarrow{\langle t_1, \dots, t_m \rangle} m \right) \times \left(z \xrightarrow{\langle s_1, \dots, s_l \rangle} l \right) \\ = n + z \xrightarrow{\langle t_1, \dots, t_m, s_1[x_{i+m}/x_i | 1 \leq i \leq l], \dots, s_l[x_{i+m}/x_i | 1 \leq i \leq l] \rangle} m + l. \end{aligned}$$

We use notation $ovar(t)$ for the list of occurrences of variables appearing (from left to right) in t and, more generally, $ovar(t_1, \dots, t_m)$ for the list $ovar(t_1) :: \dots :: ovar(t_m)$. Also, $|l| \in \mathbb{N}$ denotes the length of a list l . We say that a list $\langle t_1, \dots, t_m \rangle: n \rightarrow m$ is *linear* if each variable among x_1, \dots, x_n appears exactly once in $ovar(t_1, \dots, t_m)$.

Our first observation is that Lawvere categories are PROPs.

Proposition 4.2. $\mathcal{L}_{(\Sigma, E)}$ is a PROP.

Proof. Let \times act as the monoidal product, 0 as its unit and define the symmetry $n + m \rightarrow m + n$ as the list $\langle x_{n+1}, \dots, x_{n+m}, x_1, \dots, x_n \rangle$. It follows that $\mathcal{L}_{(\Sigma, E)}$ equipped with this structure satisfies the laws of symmetric strict monoidal categories, thus it is a PROP.

As a side observation, note that the unique PROP morphism $\mathbf{P} \rightarrow \mathcal{L}_{(\Sigma, E)}$ given by initiality of \mathbf{P} in **PROP** sends $p: n \rightarrow n$ to $\langle x_{p-1(1)}, \dots, x_{p-1(n)} \rangle$.

Remark 4.3. In spite of Proposition 4.2, cartesian theories are *not* a subclass of symmetric monoidal theories: in fact, the two concepts are incomparable. On the one hand, a symmetric monoidal theory (Σ, E) is cartesian if and only if all generators in Σ have coarity 1 and, for all equations $t = s$ in E , t and s are Σ -terms with coarity 1. Under these conditions, there is a canonical way to interpret any

Σ -term $n \rightarrow m$ as a list of m cartesian Σ -terms on variables x_1, \dots, x_n . Below, o ranges over Σ :

$$\begin{aligned} \boxed{\text{---}} : 1 \rightarrow 1 &\mapsto \langle x_1 \rangle & \boxed{\text{---}} : 2 \rightarrow 2 &\mapsto \langle x_2, x_1 \rangle \\ \boxed{\text{---}} : n \rightarrow 1 &\mapsto \langle o(x_1, \dots, x_n) \rangle. \end{aligned}$$

The inductive cases are defined using the operations $;$ and \oplus on lists given in Definition 4.1. Note that Σ -terms always denote (lists of) *linear* cartesian terms. This explains why, conversely, not all the cartesian theories are symmetric monoidal: their equations possibly involve non-linear Σ -terms, which are not expressible with (symmetric monoidal) Σ -terms. The subtlety here is that, in a sense, we can still *simulate* a cartesian theory on signature Σ with a symmetric monoidal theory, which however will be based on a larger signature Σ' , recovering the possibility of copying and discarding variables by the use of additional generators. This point will become more clear below, where we will see how copier and discharger, i.e., the cartesian structure, can be mimicked with the use of the PROP Cm .

Example 4.4. The SMT (Σ_M, E_M) of commutative monoids is cartesian. It generates the Lawvere category $\mathcal{L}_{\Sigma_M, E_M}$ whose arrows $n \rightarrow m$ are lists $\langle t_1, \dots, t_m \rangle$ of elements of the free commutative monoid on $\{x_1, \dots, x_n\}$.

The example of commutative monoids is particularly instructive for sketching our approach to Lawvere categories as composed PROPs. First, note that the Lawvere category $\mathcal{L}_{\Sigma_M, E_M}$ *includes* the PROP Mn freely generated by (Σ_M, E_M) . Indeed, any string diagram of Mn can be interpreted as a list of terms following the recipe of Remark 4.3. For instance,

$$\boxed{\text{---}} : 4 \rightarrow 3 \text{ is interpreted as } \langle \boxed{\text{---}}(x_2, \boxed{\text{---}}(x_1, x_3)), x_4, \boxed{\text{---}} \rangle : 4 \rightarrow 3$$

As we observed above, string diagrams of Mn can only express *linear* terms. What makes $\mathcal{L}_{\Sigma_M, E_M}$ more general than Mn is the ability to *copy* and *discard* variables. Indeed, just as any monoidal category in which \oplus is the cartesian product, $\mathcal{L}_{\Sigma_M, E_M}$ comes equipped with canonical choices of a “copy” and “discard” operation

$$\text{cpy}(n) := \langle x_1, \dots, x_n, x_1, \dots, x_n \rangle : n \rightarrow 2n \quad \text{dsc}(n) := \langle \rangle : n \rightarrow 0 \quad n \in \mathbb{N}$$

natural in n , which satisfy some expected equations: copying is commutative and associative; copying and then discarding is the same as not doing anything — see e.g. [14, 18].

How can we mimic the copy and discard structure in the language of PROPs? First, for each $n > 1$ one can define $\text{cpy}(n)$ and $\text{dsc}(n)$ in terms of $\text{cpy}(1)$ and $\text{dsc}(1)$, which can therefore be regarded as the only fundamental operations⁴.

⁴ For $n = 0$, both operations are equal to the identity on 0.

Also, the equations that they satisfy can be synthesised as saying that $cpy(1)$ acts as the comultiplication and $dsc(1)$ as the counit of a commutative comonoid on 1. Therefore, they are none other than the generators of the PROP \mathbf{Cm} :

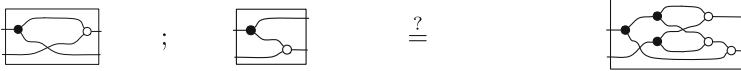
$$\begin{array}{c} \square \\ \bullet \\ \hline \end{array} : 1 \rightarrow 2 \qquad \begin{array}{c} \square \\ \bullet \\ \hline \end{array} : 1 \rightarrow 0.$$

Our approach suggests that a copy of \mathbf{Mn} and of \mathbf{Cm} “live” inside $\mathcal{L}_{\Sigma_M, E_M}$. We claim that these two PROPs provide a *complete* description of $\mathcal{L}_{\Sigma_M, E_M}$, that means, any arrow of $\mathcal{L}_{\Sigma_M, E_M}$ can be presented diagrammatically by using \mathbf{Mn} and \mathbf{Cm} . For instance,

$$\langle \begin{array}{c} \square \\ \circ \\ \square \end{array} (x_2, \begin{array}{c} \square \\ \circ \\ \square \end{array} (x_1, x_4)), x_1, \begin{array}{c} \square \\ \circ \\ \square \end{array} \rangle : 4 \rightarrow 3 \text{ corresponds to } \begin{array}{c} \xrightarrow{\in \mathbf{Cm}} \quad \xrightarrow{\in \mathbf{Mn}} \\ \begin{array}{c} \square \\ \bullet \\ \square \end{array} \\ \hline \end{array} : 4 \rightarrow 3$$

Observe that the diagram is of the factorised form $\xrightarrow{\in \mathbf{Cm}} \xrightarrow{\in \mathbf{Mn}}$. Intuitively, \mathbf{Cm} is deputed to model the interplay of variables — in this case, the fact that x_1 is copied and x_3 is deleted — and \mathbf{Mn} describes the syntactic tree of the terms. Of course, to claim that this factorisation is always possible, we need additional equations to model composition of factorised diagrams. For instance:

$$\langle \begin{array}{c} \square \\ \circ \\ \square \end{array} (x_1, x_2), x_1 \rangle ; \langle x_1, \begin{array}{c} \square \\ \circ \\ \square \end{array} (x_1, x_2) \rangle = \langle \begin{array}{c} \square \\ \circ \\ \square \end{array} (x_1, x_2), \begin{array}{c} \square \\ \circ \\ \square \end{array} (\begin{array}{c} \square \\ \circ \\ \square \end{array} (x_1, x_2), x_1) \rangle.$$



The second equality holds if we assume the equation (A8) of the SMT of *bialgebras*. Thus the example suggests that composition by substitution in $\mathcal{L}_{\Sigma_M, E_M}$ can be mimicked at the diagrammatic level by allowing the use of bialgebra equations, which as we know from Example 3.3(b) present the composite PROP $\mathbf{Cm}; \mathbf{Mn}$. Therefore, the conclusive conjecture of our analysis is that $\mathcal{L}_{\Sigma_M, E_M}$ must be isomorphic to $\mathbf{Cm}; \mathbf{Mn}$ and can be presented by equations (A1)–(A10).

We now generalise and make formal the above approach. Our main result is the following.

Theorem 4.5. *Suppose that (Σ, E) is an SMT which is also cartesian and let $\mathcal{T}_{(\Sigma, E)}$ be its freely generated PROP. Then $\mathcal{L}_{(\Sigma, E)} \cong \mathbf{Cm}; \mathcal{T}_{(\Sigma, E)}$, where distributive law $\mathcal{T}_{(\Sigma, E)}; \mathbf{Cm} \rightarrow \mathbf{Cm}; \mathcal{T}_{(\Sigma, E)}$ yielding $\mathcal{L}_{(\Sigma, E)}$ is presented by equations*

$$\begin{array}{c} \equiv \\ \square \\ \circ \\ \square \\ \vdots \end{array} \bullet = \begin{array}{c} \vdots \\ \bullet \end{array} \qquad (\text{Lw1}) \qquad \begin{array}{c} \equiv \\ \square \\ \circ \\ \square \\ \vdots \end{array} \begin{array}{c} \bullet \\ \vdots \end{array} = \begin{array}{c} \begin{array}{c} \bullet \\ \vdots \end{array} \\ \vdots \\ \begin{array}{c} \bullet \\ \vdots \end{array} \end{array} \begin{array}{c} \square \\ \circ \\ \square \\ \vdots \end{array} \qquad (\text{Lw2})$$

for each $o \in \Sigma$.

Before moving to the proof of Theorem 4.5, we show its significance by revisiting some well-known theories in terms of our result.

Example 4.6.

- (a) If we instantiate Theorem 4.5 to the theory (Σ_M, E_M) of commutative monoids (Example 2.3), then (Lw1)–(Lw2) are the bialgebra equations (A7)–(A10). The result that $\mathbf{B} \cong \mathbf{Cm}; \mathbf{Mn}$ (Example 3.3(b)) is now an immediate consequence of Theorem 4.5 and tells us that the Lawvere category of commutative monoids can be considered as the PROP of bialgebras.
- (b) In the case of monoids, the resulting Lawvere category is precisely a composite PROP, because all the equations only affect the linear part of the theory, that means, the generating cartesian theory is also an SMT. As observed in Remark 4.3, this is not true in general: for instance, the cartesian theory (Σ_G, E_G) of abelian groups extends the one (Σ_M, E_M) of commutative monoids with an inverse operation $\boxed{\blacksquare} : 1 \rightarrow 1$ and a non-linear equation $\boxed{\circlearrowleft}(x, \boxed{\blacksquare}(x)) = \boxed{\square}$. In such cases, Theorem 4.5 still yields useful information about the structure of the resulting Lawvere category. For instance, it means that $\mathcal{L}_{(\Sigma_G, E_G)}$ is isomorphic to the PROP $\mathbf{Cm}; \mathcal{T}_{\Sigma_G, E_M}$ quotiented by the above non-linear equation, which is rendered in string diagrams as:

$$\text{---} \bullet \text{---} \boxed{\circlearrowleft} \text{---} \circ \text{---} = \text{---} \bullet \text{---} \circ \text{---}$$

Interestingly, the result of this quotient is isomorphic to the PROP of integer matrices, see e.g. [37, §3.5] and its models in a symmetric monoidal category are the *Hopf algebras* [17], with $\boxed{\blacksquare}$ playing the role of the antipode.

- (c) In [21] Fritz presents the category of finite sets and probabilistic maps using generators and equations. The resulting Lawvere category $\mathcal{L}_{\text{Prob}}$ can be decomposed following the scheme of Theorem 4.5: there is a linear part (Σ_P, E_P) of the theory — given by binary convex combinations $\Sigma_P = \{\boxed{\lambda} : 2 \rightarrow 1 \mid \lambda \in [0, 1]\}$ and suitable associativity and commutativity laws in E_P , a commutative comonoid structure, and the two interact according to (Lw1)–(Lw2). This interaction yields a composite PROP $\mathbf{Cm}; \mathcal{T}_{(\Sigma_P, E_P)}$ which, quotiented by non-linear equations $\boxed{0} = \text{---} \bullet$ and $\bullet \boxed{\lambda} = \text{---}$, yields $\mathcal{L}_{\text{Prob}}$.⁵

Remark 4.7. It is instructive to observe how Theorem 4.5 translates to models of theories. We recalled what is a model for a PROP in Remark 2.4; there is an analogous notion for Lawvere categories. A model for a Lawvere category $\mathcal{L}_{(\Sigma, E)}$ is a cartesian category \mathbb{C} together with a cartesian (i.e., finite-products preserving) functor $\mathcal{L}_{(\Sigma, E)} \rightarrow \mathbb{C}$: models of $\mathcal{L}_{(\Sigma, E)}$ in \mathbb{C} and natural transformations between them form a category $\text{CartMod}(\mathcal{L}_{(\Sigma, E)}, \mathbb{C})$. Now, for (Σ, E) and $\mathcal{T}_{(\Sigma, E)}$ as in Theorem 4.5, we have that models of $\mathcal{L}_{(\Sigma, E)}$ in \mathbb{C} cartesian are the same as models of $\mathcal{T}_{(\Sigma, E)}$ in \mathbb{C} , now seen more abstractly as a symmetric monoidal category. That means, there is an equivalence $\text{LinMod}(\mathcal{T}_{(\Sigma, E)}, \mathbb{C}) \simeq \text{CartMod}(\mathcal{L}_{(\Sigma, E)}, \mathbb{C})$.

⁵ In fact, the Lawvere category in [21] has finite coproducts, while our $\mathcal{L}_{\text{Prob}}$ is based on finite products. This is just a matter of co-/contra-variant presentation of the same data: one can switch between the two by “vertical rotation” of diagrams.

The rest of the section is devoted to proving Theorem 4.5. First we observe that, by the following lemma, it actually suffices to check our statement for SMTs with no equations. This reduction has just the purpose of making computations in $\mathcal{L}_{(\Sigma, E)}$ easier, by working with terms instead of equivalence classes.

Lemma 4.8. *If the statement of Theorem 4.5 holds in the case $E = \emptyset$, then it holds for any cartesian SMT (Σ, E) .*

Proof. Let (Σ, E) be a cartesian SMT and $\mathcal{T}_\Sigma, \mathcal{T}_{(\Sigma, E)}$ be the PROPs freely generated, respectively, by (Σ, \emptyset) and (Σ, E) . By assumption, Theorem 4.5 holds for (Σ, \emptyset) , yielding a distributive law $\lambda: \mathcal{T}_\Sigma; \mathbf{Cm} \rightarrow \mathbf{Cm}; \mathcal{T}_\Sigma$. A routine check shows that λ preserves the equations of E , whence Proposition 3.6 gives a distributive law $\lambda': \mathcal{T}_{(\Sigma, E)}; \mathbf{Cm} \rightarrow \mathbf{Cm}; \mathcal{T}_{(\Sigma, E)}$ with the required properties.

In the sequel, let us abbreviate $\mathcal{L}_{\Sigma, \emptyset}$ as \mathcal{L}_Σ . By virtue of Lemma 4.8, we shall prove Theorem 4.5 for \mathcal{L}_Σ and by letting \mathcal{T}_Σ be the PROP freely generated by (Σ, \emptyset) . We shall obtain the distributive law $\mathcal{T}_\Sigma; \mathbf{Cm} \rightarrow \mathbf{Cm}; \mathcal{T}_\Sigma$ from the recipe of Proposition 3.4, by showing that any arrow of \mathcal{L}_Σ decomposes as $\xrightarrow{\in \mathbf{Cm}} \xrightarrow{\in \mathcal{T}_\Sigma}$.

We now give some preliminary lemmas that are instrumental for the definition of the factorisation and the proof of the main result. We begin by showing how string diagrams of \mathbf{Cm} and \mathcal{T}_Σ are formally interpreted as arrows of \mathcal{L}_Σ .

Lemma 4.9.

– \mathbf{Cm} is the sub-PROP of \mathcal{L}_Σ whose arrows are lists of variables. The inclusion of \mathbf{Cm} in \mathcal{L}_Σ is the morphism $\Phi: \mathbf{Cm} \rightarrow \mathcal{L}_\Sigma$ defined on generators of \mathbf{Cm} by

$$\boxed{\bullet} \mapsto \langle x_1, x_1 \rangle: 1 \rightarrow 2 \qquad \boxed{\bullet} \mapsto \langle \rangle: 0 \rightarrow 1.$$

– \mathcal{T}_Σ is the sub-PROP of \mathcal{L}_Σ whose arrows are linear terms. The inclusion of \mathcal{T}_Σ in \mathcal{L}_Σ is the morphism $\Psi: \mathcal{T}_\Sigma \rightarrow \mathcal{L}_\Sigma$ defined on generators of \mathcal{T}_Σ by

$$\boxed{\text{O}} \mapsto \langle o(x_1, \dots, x_n) \rangle: n \rightarrow 1 \qquad (o: n \rightarrow 1) \in \Sigma.$$

Proof. First, it is immediate to verify that lists of variables are closed under composition, monoidal product and include all the symmetries of \mathcal{L}_Σ : therefore, they form a sub-PROP. The same holds for linear terms.

We now consider the first statement of the lemma. There is a 1-1 correspondence between arrows $n \xrightarrow{f \in \mathcal{L}_\Sigma} m$ that are lists of variables and functions $\bar{m} \rightarrow \bar{n}$: the function for f maps k , for $1 \leq k \leq m$, to the index l of the variable x_l appearing in position k in f . This correspondence yields an isomorphism between the sub-PROP of \mathcal{L}_Σ whose arrows are lists of variables and \mathbf{F}^{op} . Composing this isomorphism with $\mathbf{Cm} \xrightarrow{\cong} \mathbf{F}^{op}$ yields Φ as in the statement of the lemma.

For the second statement, faithfulness is immediate by the fact that arrows of \mathcal{T}_Σ are Σ -terms modulo the laws of SMCs, with no additional equations. One can easily verify that $\Psi: \mathcal{T}_\Sigma \rightarrow \mathcal{L}_\Sigma$ identifies the linear terms in \mathcal{L}_Σ following the observations in Remark 4.3.

Henceforth, for the sake of readability we shall not distinguish between \mathbf{Cm} and the isomorphic sub-PROP of \mathcal{L}_Σ identified by the image of Φ , and similarly for \mathcal{T}_Σ and Ψ . Lemma 4.9 allows us to use \mathcal{L}_Σ as an environment where \mathbf{Cm} and \mathcal{T}_Σ interact. The following statement guarantees the soundness of the interaction described by (LW1)–(LW2).

Lemma 4.10. *Equations (LW1) and (LW2) are sound in \mathcal{L}_Σ .*

Proof. We first focus on (LW1). Following the isomorphisms of Lemma 4.9, $\boxed{\mathcal{O}} \in \mathcal{T}_\Sigma[n, 1]$ is interpreted as the arrow $\langle o(x_1, \dots, x_n) \rangle \in \mathcal{L}_\Sigma[n, 1]$ and $\boxed{\bullet} \in \mathbf{Cm}[0, 1]$ as $\langle \rangle \in \mathcal{L}_\Sigma[1, 0]$. The left-hand side of (LW1) is then the composite $\langle o(x_1, \dots, x_n) \rangle; \langle \rangle \in \mathcal{L}_\Sigma[n, 0]$, which is equal by definition to $\langle \rangle \in \mathcal{L}_\Sigma[n, 0]$. Therefore, the left- and right-hand side of (LW1) are the same arrow of \mathcal{L}_Σ .

It remains to show soundness of (LW2). Following Lemma 4.9, the left-hand side $\boxed{\mathcal{O}}; \boxed{\bullet}$ is interpreted in \mathcal{L}_Σ as $\langle o(x_1, \dots, x_n) \rangle; \langle x_1, x_1 \rangle$ and the right-hand side as $\langle x_1, \dots, x_n, x_1, \dots, x_n \rangle; \langle o(x_1, \dots, x_n), o(x_{n+1}, \dots, x_{n+n}) \rangle$. By definition, both composites are equal to $\langle o(x_1, \dots, x_n), o(x_1, \dots, x_n) \rangle$ in \mathcal{L}_Σ . Therefore, (LW2) is also sound in \mathcal{L}_Σ .

It is useful to observe that (LW1)–(LW2) allows us to copy and discard not only the generators but arbitrary string diagrams of \mathcal{T}_Σ .

Lemma 4.11. *Suppose \boxed{d} is a string diagram of \mathcal{T}_Σ . Then the following holds in $\mathcal{T}_\Sigma + \mathbf{Cm}$ quotiented by (LW1)–(LW2).*

$$\begin{array}{c} \boxed{d} \\ \text{---} \\ \text{---} \\ \text{---} \\ \text{---} \end{array} = \begin{array}{c} \boxed{d} \\ \text{---} \\ \text{---} \\ \text{---} \\ \text{---} \end{array} \quad (\text{Lw3}) \qquad \boxed{d} \text{---} \bullet \text{---} \bullet \text{---} \bullet = \text{---} \bullet \text{---} \bullet \text{---} \bullet \quad (\text{Lw4})$$

Proof. The proof is by induction on \boxed{d} . For (Lw3), the base cases of $\boxed{}$ and $\boxed{\times}$ follow by the laws of SMCs (Fig. 1). The base case of $\boxed{\mathcal{O}}$, for o a generator in Σ , is given by (LW2). The inductive cases of composition by $;$ and \oplus immediately follow by induction hypothesis. The proof of (Lw4) is analogous.

We can now show the factorisation lemma.

Lemma 4.12. *Any arrow $n \xrightarrow{f \in \mathcal{L}_\Sigma} m$ has a factorisation $n \xrightarrow{\hat{c} \in \mathbf{Cm}} \xrightarrow{\hat{d} \in \mathcal{T}_\Sigma} m$ which is unique up-to permutation.*

Proof. Since the cartesian theory generating \mathcal{L}_Σ has no equations, $n \xrightarrow{f} m$ is just a list of cartesian Σ -terms $\langle t_1, \dots, t_m \rangle$. The factorisation consists in replacing all variables appearing in $\langle t_1, \dots, t_m \rangle$ with fresh ones x_1, \dots, x_z , so that no repetition occurs: this gives us the second component of the decomposition as a list of linear terms $z \xrightarrow{\hat{d} \in \mathcal{T}_\Sigma} m$. The first component \hat{c} will be the list $n \xrightarrow{\text{ovar}(t_1, \dots, t_m) \in \mathbf{Cm}} z$ of variables originally occurring in f , so that post-composition with \hat{d} yields $\langle t_1, \dots, t_m \rangle$. It is simple to verify uniqueness up-to permutation of this factorisation.

We now have all the ingredients to conclude the proof of our main statement.

Proof (Theorem 4.5). Using the conclusion of Lemma 4.12, Proposition 3.4 gives us a distributive law $\lambda: \mathcal{T}_\Sigma; \mathbf{Cm} \rightarrow \mathbf{Cm}; \mathcal{T}_\Sigma$ such that $\mathcal{L}_\Sigma \cong \mathbf{Cm}; \mathcal{T}_\Sigma$. It remains to show that (Lw1)–(Lw2) allow to prove all the equations arising from λ . By Proposition 3.4, λ maps a composable pair $n \xrightarrow{d \in \mathcal{T}_\Sigma} \xrightarrow{c \in \mathbf{Cm}} m$ to the factorisation $n \xrightarrow{c' \in \mathbf{Cm}} \xrightarrow{d' \in \mathcal{T}_\Sigma} m$ of d ; c in \mathcal{L}_Σ , calculated according to Lemma 4.12. The corresponding equation generated by λ is $d; c = c'; d'$, with d, c, c', d' now seen as string diagrams of $\mathcal{T}_\Sigma + \mathbf{Cm}$. The equational theory of $\mathcal{L}_\Sigma \cong \mathbf{Cm}; \mathcal{T}_\Sigma$ consists of all the equations arising in this way plus those of $\mathcal{T}_\Sigma + \mathbf{Cm}$. What we need to show is that

$$\text{the string diagrams } d; c \text{ and } c'; d' \text{ are equal modulo the} \quad (\dagger) \\ \text{equations of } \mathcal{T}_\Sigma + \mathbf{Cm} \text{ and (Lw1)–(Lw2).}$$

Since our factorisation is unique up-to permutation, it actually suffices to show a weaker statement, namely that

$$\text{there exists a factorisation } n \xrightarrow{c'' \in \mathbf{Cm}} \xrightarrow{d'' \in \mathcal{T}_\Sigma} m \text{ of } d; c \quad (\ddagger) \\ \text{in } \mathcal{L}_\Sigma \text{ such that the string diagrams } d; c \text{ and } c''; d'' \text{ are} \\ \text{equal modulo the equations of } \mathcal{T}_\Sigma + \mathbf{Cm} \text{ and (Lw1)–(Lw2).}$$

Statement (\ddagger) implies (\dagger) because, by uniqueness of the factorisation $c'; d'$ up-to permutation, there exists $\xrightarrow{p \in \mathbf{P}}$ such that $d' = p; d''$ and $c' = c'; p$ in \mathcal{L}_Σ . Since p is an arrow of both sub-PROPs \mathcal{T}_Σ and \mathbf{Cm} , the first equality also holds in \mathcal{T}_Σ and the second in \mathbf{Cm} . So $c'; d' = c'; p; d'' = c''; d''$ in $\mathcal{T}_\Sigma + \mathbf{Cm}$.

Therefore, we turn to a proof of (\ddagger) . We describe a procedure to transform the string diagram $\xrightarrow{d \in \mathcal{T}_\Sigma} \xrightarrow{c \in \mathbf{Cm}}$ into the form $\xrightarrow{c'' \in \mathbf{Cm}} \xrightarrow{d'' \in \mathcal{T}_\Sigma}$, by only using the equations in $\mathcal{T}_\Sigma + \mathbf{Cm}$ plus (Lw1)–(Lw2). Lemmas 4.9 and 4.10 guarantee that $d; c = c''; d''$ as arrows of \mathcal{L}_Σ .

1. First, there is a preparatory step in which we move all symmetries to the outmost part of the string diagram $d; c$, to ease the application of (Lw1)–(Lw2). By definition, d only contains components of the kind $\boxed{\text{O}}: k \rightarrow 1, \circ \in \Sigma, \boxed{\text{X}}: 2 \rightarrow 2$ and $\boxed{\text{I}}: 1 \rightarrow 1$. Using naturality (Fig. 1), we can move all symmetries $\boxed{\text{X}}$ to the left of components $\boxed{\text{O}}$.



The result is a string diagram $p; \bar{d}; c'$, where p only contains components $\boxed{\text{X}}$ and $\boxed{\text{I}}$ — i.e., it is a string diagram of \mathbf{P} — and \bar{d} is a string diagram

of \mathcal{T}_Σ where $\boxed{\text{X}}$ does not appear.

$$\boxed{d'} \boxed{c'} \implies \boxed{p} \boxed{\bar{d}} \boxed{c'}$$

We then perform a symmetric transformation on the string diagram c . By definition, c contains components $\boxed{\bullet} : 1 \rightarrow 2$, $\boxed{\bullet} : 0 \rightarrow 1$, $\boxed{\bowtie} : 2 \rightarrow 2$ and $\boxed{\square} : 1 \rightarrow 1$. By naturality, we can move all symmetries $\boxed{\bowtie}$ to the right of any component $\boxed{\bullet}$ and $\boxed{\bullet}$.

$$\begin{array}{ccc} \text{Diagram 1} \implies \text{Diagram 2} & & \text{Diagram 3} \implies \text{Diagram 4} \\ \text{Diagram 5} \implies \text{Diagram 6} & & \text{Diagram 7} \implies \text{Diagram 8} \end{array}$$

The result is a string diagram $p; \bar{d}; \bar{c}; p'$, where \bar{c} is a string diagram of Cm in which $\boxed{\bowtie}$ does not appear and p' is a string diagram of P .

$$\boxed{p} \boxed{\bar{d}} \boxed{c'} \implies \boxed{p} \boxed{\bar{d}} \boxed{\bar{c}} \boxed{p'}$$

- We now make \bar{d} and \bar{c} interact. First note that, since \bar{d} does not contain $\boxed{\bowtie}$ and all generators $o \in \Sigma$ have coarity 1, \bar{d} must be the \oplus -product $\bar{d}_1 \oplus \dots \oplus \bar{d}_z$ of string diagrams $\bar{d}_i : k_i \rightarrow 1$ of \mathcal{T}_Σ .

$$\boxed{p} \boxed{\bar{d}} \boxed{\bar{c}} \boxed{p'} = \begin{array}{c} \boxed{p} \quad \begin{array}{c} \boxed{\bar{d}_1} \\ \boxed{\bar{d}_2} \\ \vdots \\ \boxed{\bar{d}_z} \end{array} \quad \boxed{\bar{c}} \quad \boxed{p'} \end{array}$$

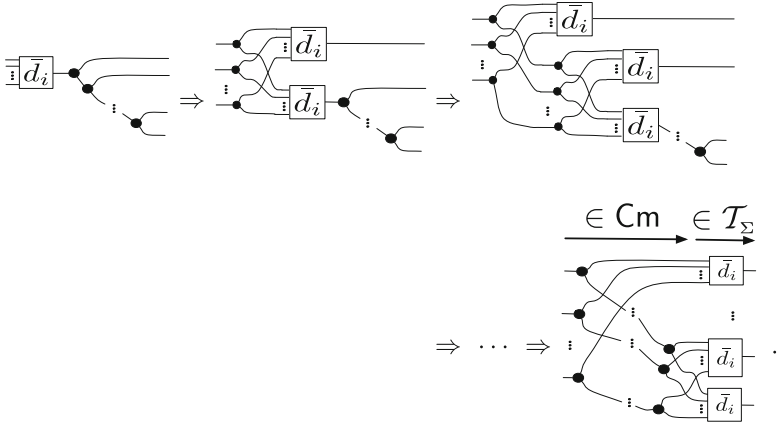
For analogous reasons, \bar{c} is also a \oplus -product $\bar{c}_1 \oplus \dots \oplus \bar{c}_z$ where, for $1 \leq i \leq z$,

$$\boxed{\bar{c}_i} = \text{Diagram} \quad \text{or} \quad \boxed{\bar{c}_i} = \bullet \quad (9)$$

We thus can present \bar{c} as follows:

$$\begin{array}{c} \boxed{p} \quad \begin{array}{c} \boxed{\bar{d}_1} \\ \boxed{\bar{d}_2} \\ \vdots \\ \boxed{\bar{d}_z} \end{array} \quad \boxed{\bar{c}} \quad \boxed{p'} \\ = \\ \boxed{p} \quad \begin{array}{c} \boxed{\bar{d}_1} \quad \boxed{\bar{c}_1} \\ \boxed{\bar{d}_2} \quad \boxed{\bar{c}_2} \\ \vdots \\ \boxed{\bar{d}_z} \quad \boxed{\bar{c}_z} \end{array} \quad \boxed{p'} \end{array}$$

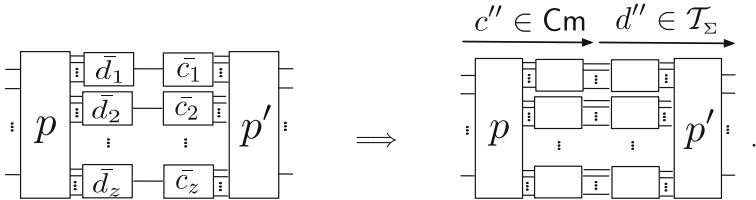
We are now in position to distribute each \bar{d}_i over the corresponding \bar{c}_i . Suppose first \bar{c}_i satisfies the left-hand equality in (9). By assumption, all the equations of $\mathcal{T}_\Sigma + \text{Cm}$, Lw1 and (Lw2) hold. Thus, by Lemma 4.11, also (Lw3) holds. Starting from $\bar{d}_i; \bar{c}_i$, we can iteratively apply (Lw3) to obtain a string diagram of shape $\xrightarrow{\in \text{Cm}} \xrightarrow{\in \mathcal{T}_\Sigma}$:



In the remaining case, \bar{c}_i satisfies the right-hand equality in (9). Then, one application of (Lw1) also gives us a string diagram of shape $\xrightarrow{\in \text{Cm}} \xrightarrow{\in \mathcal{T}_\Sigma}$.



Applying the above transformations for each \bar{d}_i ; \bar{c}_i yields a string diagram of the desired shape $\xrightarrow{c'' \in \text{Cm}} \xrightarrow{d'' \in \mathcal{T}_\Sigma}$.



Observe that all the transformations that we described only used equations in $\mathcal{T}_\Sigma + \text{Cm}$, (Lw1) and (Lw2). This concludes the proof of (‡) and thus of the main theorem.

Acknowledgements. The first author acknowledge support by project ANR 12IS02001 PACE. The third author acknowledges support from the European Research Council under the European Union’s Seventh Framework Programme (FP7/2007-2013) / ERC grant agreement n° 320571.

References

1. Baez, J.C., Erbele, J.: Categories in control (2014). CoRR abs/1405.6881

2. Baez, J.C.: Universal algebra and diagrammatic reasoning. Lecture (2006). http://math.ucr.edu/home/baez/universal/universal_hyper.pdf
3. Studer, C.: Introduction. In: Studer, C. (ed.) Numerics of Unilateral Contacts and Friction. LNACM, vol. 47, pp. 1–8. Springer, Heidelberg (2009)
4. Bonchi, F., Milius, S., Silva, A., Zanasi, F.: Killing epsilons with a dagger – a coalgebraic study of systems with algebraic label structure. *Theoret. Comput. Sci.* **604**, 102–126 (2015)
5. Bonchi, F., Sobociński, P., Zanasi, F.: A categorical semantics of signal flow graphs. In: Baldan, P., Gorla, D. (eds.) CONCUR 2014. LNCS, vol. 8704, pp. 435–450. Springer, Heidelberg (2014)
6. Bonchi, F., Sobociński, P., Zanasi, F.: Interacting bialgebras are Frobenius. In: Muscholl, A. (ed.) FOSSACS 2014 (ETAPS). LNCS, vol. 8412, pp. 351–365. Springer, Heidelberg (2014)
7. Bonchi, F., Sobocinski, P., Zanasi, F.: Full abstraction for signal flow graphs. In: Proceedings of the 42nd Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL 2015, pp. 515–526 (2015)
8. Bonchi, F., Zanasi, F.: Bialgebraic semantics for logic programming. *Log. Methods Comput. Sci.* **11**(1:14), 1–47 (2015)
9. Bonsangue, M.M., Hansen, H.H., Kurz, A., Rot, J.: Presenting distributive laws. In: Heckel, R., Milius, S. (eds.) CALCO 2013. LNCS, vol. 8089, pp. 95–109. Springer, Heidelberg (2013)
10. Borceux, F.: Handbook of Categorical Algebra 1 - Basic Category Theory. Cambridge Univ. Press, Cambridge (1994)
11. Bruni, R., Melgratti, H., Montanari, U.: A connector algebra for P/T nets interactions. In: Katoen, J.-P., König, B. (eds.) CONCUR 2011. LNCS, vol. 6901, pp. 312–326. Springer, Heidelberg (2011)
12. Bruni, R., Montanari, U., Plotkin, G.D., Terreni, D.: On hierarchical graphs: reconciling bigraphs, gs-monoidal theories and gs-graphs. *Fundam. Inform.* **134**(3–4), 287–317 (2014)
13. Burris, S., Sankappanavar, H.P.: A Course in Universal Algebra. Springer, Heidelberg (1981)
14. Burroni, A.: Higher dimensional word problems with applications to equational logic. *Theor. Comput. Sci.* **115**, 43–62 (1993)
15. Coecke, B., Duncan, R.: Interacting quantum observables: categorical algebra and diagrammatics. *New J. Phys.* **13**(4), 043016 (2011)
16. Corradini, A., Gadducci, F.: An algebraic presentation of term graphs, via gs-monoidal categories. *Appl. Categorical Struct.* **7**(4), 299–331 (1999)
17. Dascalescu, S., Nastasescu, C., Raianu, S.: Hopf Algebras, Pure and Applied Mathematics, vol. 235. Marcel Dekker Inc., New York (2001). An Introduction
18. Eilenberg, S., Kelly, G.M.: Closed categories. In: Eilenberg, S., Harrison, D.K., MacLane, S., Röhri, H. (eds.) Proceedings of the Conference on Categorical Algebra, pp. 421–562. Springer, Heidelberg (1966)
19. Fiore, M., Devesas Campos, M.: The algebra of directed acyclic graphs. In: Coecke, B., Ong, L., Panangaden, P. (eds.) Computation, Logic, Games and Quantum Foundations. LNCS, vol. 7860, pp. 37–51. Springer, Heidelberg (2013)
20. Fong, B., Rapisarda, P., Sobociński, P.: A categorical approach to open and interconnected dynamical systems (2015). CoRR/1510.05076
21. Fritz, T.: A presentation of the category of stochastic matrices (2009). CoRR abs/0902.2554. <http://arxiv.org/abs/0902.2554>
22. Hyland, M., Plotkin, G., Power, J.: Combining effects: sum and tensor. *Theor. Comput. Sci.* **357**(1–3), 70–99 (2006)

23. Hyland, M., Power, J.: The category theoretic understanding of universal algebra: Lawvere theories and monads. *Electron. Notes Theor. Comput. Sci.* **172**, 437–458 (2007)
24. Jacobs, B.: Semantics of weakening and contraction. *Ann. Pure Appl. Logic* **69**(1), 73–106 (1994)
25. Lack, S.: Composing PROPs. *Theor. App. Categories* **13**(9), 147–163 (2004)
26. Lafont, Y.: Equational reasoning with 2-dimensional diagrams. *Term Rewriting. LNCS*, vol. 909, pp. 170–195. Springer, Heidelberg (1995)
27. Lawvere, W.F.: Functorial semantics of algebraic theories. Ph.D. thesis (2004)
28. Mac Lane, S.: Categorical algebra. *Bull. Am. Math. Soc.* **71**, 40–106 (1965)
29. Mac Lane, S.: *Categories for the Working Mathematician*. Springer, Heidelberg (1998)
30. Mellies, P.A.: Categorical semantics of linear logic. In: *Interactive Models of Computation and Program Behaviour, Panoramas et Synthèses 27*, Société Mathématique de France 1196 (2009)
31. Rosebrugh, R., Wood, R.: Distributive laws and factorization. *J. Pure Appl. Algebra* **175**(13), 327–353 (2002). Special Volume celebrating the 70th birthday of Professor Max Kelly
32. Ross Duncan, K.D.: Interacting Frobenius algebras are Hopf. *CoRR* abs/1601.04964 (2016). <http://arxiv.org/abs/1601.04964>
33. Selinger, P.: A survey of graphical languages for monoidal categories. *Springer Lect. Notes Phys.* **13**(813), 289–355 (2011)
34. Sleep, M.R., Plasmeijer, M.J., van Eekelen, M.C.J.D. (eds.): *Term Graph Rewriting: Theory and Practice*. Wiley, Chichester (1993)
35. Sobociński, P., Stephens, O.: A programming language for spatial distribution of net systems. In: *Petri Nets 2014* (2014)
36. Street, R.: The formal theory of monads. *J. Pure Appl. Algebra* **2**(1), 243–265 (2002)
37. Zanasi, F.: *Interacting Hopf algebras: the theory of linear systems*. Ph.D. thesis, Ecole Normale Supérieure de Lyon (2015)