The Microcosm Principle and Concurrency in Coalgebra

Ichiro Hasuo[1,3,4], Bart Jacobs[1,⋆], and Ana Sokolova[2,⋆⋆]

[1] Radboud University Nijmegen, the Netherlands
[2] University of Salzburg, Austria
[3] RIMS, Kyoto University, Japan
[4] PRESTO Research Promotion Program, Japan Science and Technology Agency

Abstract. Coalgebras are categorical presentations of state-based systems. In investigating parallel composition of coalgebras (realizing *concurrency*), we observe that the same algebraic theory is interpreted in two different domains in a nested manner, namely: in the category of coalgebras, and in the final coalgebra as an object in it. This phenomenon is what Baez and Dolan have called the *microcosm principle*, a prototypical example of which is "a monoid in a monoidal category." In this paper we obtain a formalization of the microcosm principle in which such a nested model is expressed categorically as a suitable lax natural transformation. An application of this account is a general compositionality result which supports modular verification of complex systems.

1 Introduction

Design of systems with *concurrency* is nowadays one of the mainstream challenges in computer science [19]. Concurrency is everywhere: with the Internet being the biggest example and multi-core processors the smallest; also in a modular, component-based architecture of a complex system its components collaborate in a concurrent manner. However, numerous difficulties have been identified in getting concurrency right. For example, a system's exponentially growing complexity is one of the main obstacles. One way to cope with it is a *modular* verification method in which correctness of the whole system $\mathcal{C}_1 \parallel \cdots \parallel \mathcal{C}_n$ is established using correctness of each component $\mathcal{C}_i$. *Compositionality*—meaning that the behavior of $\mathcal{C} \parallel \mathcal{D}$ is determined by the behavior of $\mathcal{C}$ and that of $\mathcal{D}$—is an essential property for such a modular method to work.

Coalgebras as systems This paper is a starting point of our research program aimed at better understanding of the mathematical nature of concurrency. In its course we shall use *coalgebras* as presentations of systems to be run in parallel. The use of coalgebras as an appropriate abstract model of state-based systems is increasingly established [11,26]; the notion's mathematical simplicity and clarity provide us with a sound foundation

⋆ Also part-time at Technical University Eindhoven, the Netherlands.
⋆⋆ Supported by the Austrian Science Fund (FWF) project no. P18913-N15.

for our exploration. The following table summarizes how ingredients of the theory of systems are presented as coalgebraic constructs.

$$
\begin{array}{l||c|c|c}
 & \text{system} & \text{behavior-preserving map} & \text{behavior} \\
\hline
\text{coalgebraically} & \text{coalgebra} \quad \begin{array}{c} FX \\ \uparrow \\ X \end{array} & \begin{array}{c}\text{morphism of coalgebras} \\ \begin{array}{ccc} FX & \xrightarrow{Ff} & FY \\ \uparrow & & \uparrow \\ X & \xrightarrow[f]{} & Y \end{array}\end{array} & \begin{array}{c}\text{by coinduction} \\ \begin{array}{ccc} FX & \dashrightarrow & FZ \\ {\scriptstyle c}\uparrow & & {\scriptstyle \mathsf{final}}\uparrow\cong \\ X & \underset{\mathsf{beh}(c)}{\dashrightarrow} & Z \end{array}\end{array}
\end{array}
\qquad (1)
$$

This view of "coalgebras as systems" has been successfully applied in the category $\mathbf{Sets}$ of sets and functions, in which case the word "behavior" in (1) refers (roughly) to bisimilarity. Our recent work [5, 6] has shown that "behavior" can also refer to trace semantics by moving from $\mathbf{Sets}$ to a suitable Kleisli category.

Compositionality in coalgebras We start with the following question: what is "compositionality" in this coalgebraic setting? Conventionally compositionality is expressed as: $\mathcal{C} \sim \mathcal{C}'$ and $\mathcal{D} \sim \mathcal{D}'$ implies $\mathcal{C} \parallel \mathcal{D} \sim \mathcal{C}' \parallel \mathcal{D}'$, where the relation $\sim$ denotes the behavioral equivalence of interest. If this is the case the relation $\sim$ is said to be a *congruence*, with its oft-heard instance being "bisimilarity is a congruence."

When we interpret "behavior" in compositionality as the coalgebraic behavior induced by coinduction (see (1)), the following equation comes natural as a coalgebraic presentation of compositionality.

$$
\mathsf{beh}\left(\begin{array}{c} FX \\ c\uparrow \\ X \end{array} \middle\| \begin{array}{c} FY \\ d\uparrow \\ Y \end{array}\right) = \mathsf{beh}\left(\begin{array}{c} FX \\ c\uparrow \\ X \end{array}\right) \Bigg\| \; \mathsf{beh}\left(\begin{array}{c} FY \\ d\uparrow \\ Y \end{array}\right) \qquad (2)
$$

But a closer look reveals that the two "parallel composition operators" $\parallel$ in the equation have in fact different types: the first one $\mathbf{Coalg}_F \times \mathbf{Coalg}_F \to \mathbf{Coalg}_F$ combines systems (as coalgebras) and the second one $Z \times Z \to Z$ combines behavior (as states of the final coalgebra).[5] Moreover, the two domains are actually nested: the latter one $Z \xrightarrow{\cong} FZ$ is an object of the former one $\mathbf{Coalg}_F$.

The microcosm principle What we have just observed is one instance—probably the first one explicitly claimed in computer science—of the *microcosm principle* as it is called by Baez and Dolan [1]. It refers to a phenomenon that the same algebraic theory (or algebraic "specification," consisting of operations and equations) is interpreted twice in a nested manner, once in a category $\mathbb{C}$ and the other time in its object $X \in \mathbb{C}$. This is not something very unusual, because "a monoid in a monoidal category" constitutes

[5] At this stage the presentation remains sloppy for the sake of simplicity. Later in technical sections the first composition operator will be denoted by $\otimes$; and the second composition operator will have the type $Z \otimes Z \to Z$ instead of $Z \times Z \to Z$.

a prototypical example.

monoidal category $\mathbb{C}$		monoid $X \in \mathbb{C}$
$\otimes : \mathbb{C} \times \mathbb{C} \to \mathbb{C}$	multiplication	$X \otimes X \xrightarrow{\mu} X$
$I \in \mathbb{C}$	unit	$I \xrightarrow{\eta} X$
$I \otimes X \cong X \cong X \otimes I$	unit law	$\begin{tikzcd} X \arrow[r] \arrow[dr, equal] & X \otimes X \arrow[d] & X \arrow[l] \arrow[dl, equal] \\ & X & \end{tikzcd}$
$X \otimes (Y \otimes Z) \cong (X \otimes Y) \otimes Z$	associativity law	$\begin{tikzcd} X \otimes X \otimes X \arrow[r] \arrow[d] & X \otimes X \arrow[d] \\ X \otimes X \arrow[r] & X \end{tikzcd}$

(3)

Notice here that the outer operation $\otimes$ appears in the formulation of the inner operation μ. Moreover, to be precise, in the inner "equations" the outer isomorphisms should be present in suitable places. Hence this monoid example demonstrates that, in such nested algebraic structures, the inner structure depends on the outer. What is a mathematically precise formalization of such nested models? Answering this question is a main goal of this paper.

Such a formalization has been done in [1] when algebraic structures are specified in the form of *opetopes*. Here instead we shall formalize the microcosm principle for *Lawvere theories* [18], whose role as categorical representation of algebraic theories has been recognized in theoretical computer science.

As it turns out, our formalization looks like the situation on the right. Here $\mathbb{L}$ is a category (a Lawvere theory) representing an algebraic theory; an outer model $\mathbb{C}$ is a product-preserving functor; and an inner model X is a lax natural transformation. The whole setting is 2-categorical: 2-categories (categories in categories) serve as an appropriate basis for the microcosm principle (algebras in algebras).

$$\begin{tikzcd} \mathbb{L} \arrow[r, bend left=60, "\mathbf{1}"{name=U}] \arrow[r, "\mathbb{C}"'{name=D}] & \mathbf{CAT} \arrow[from=U, to=D, phantom, "\Downarrow X"] \end{tikzcd}$$

Applications to coalgebras: parallel composition via sync The categorical account we have sketched above shall be applied to our original question about parallel composition of coalgebras. As a main application we prove a *generic compositionality theorem*. For an arbitrary algebraic theory $\mathbb{L}$, compositionality like (2) is formulated as follows: the "behavior" functor $\mathsf{beh} : \mathbf{Coalg}_F \to \mathbb{C}/Z$ via coinduction preserves an $\mathbb{L}$-structure. This general form of compositionality holds if: $\mathbb{C}$ has an $\mathbb{L}$-structure and $F : \mathbb{C} \to \mathbb{C}$ *lax*-preserves the $\mathbb{L}$-structure.

Turning back to the original setting of (2), these general assumptions read roughly as follows: the base category $\mathbb{C}$ has a binary operation $\|$; and the endofunctor F comes with a natural transformation $\mathsf{sync} : FX \parallel FY \to F(X \parallel Y)$. Essentially, this sync is what lifts $\|$ on $\mathbb{C}$ to $\|$ on $\mathbf{Coalg}_F$, hence "parallel composition via sync." It is called a *synchronization* because it specifies the way two systems synchronize with each other. In fact, for a fixed functor F there can be different choices of sync (such as CSP-style vs. CCS-style), which in turn yield different "parallel composition" operators on the category $\mathbf{Coalg}_F$.

Related work Our interest is pretty similar to that of studies of *bialgebraic structures* in computer science (such as [3, 12, 14–16, 27]), in the sense that we are also concerned about algebraic structures on coalgebras as systems. Our current framework is distinguished in the following aspects.

First, we handle *equations* in an algebraic theory as an integral part of our approach. Equations such as associativity and commutativity appear explicitly as commutative diagrams in a Lawvere theory $\mathbb{L}$. We benefit from this explicitness in e.g. spelling out a condition for the generic associativity result (Theorem 2.4). In contrast, in the bialgebraic studies an algebraic theory is presented either by an endofunctor $X \mapsto \coprod_{\sigma \in \Sigma} X^{|\sigma|}$ or by a monad T. In the former case equations are simply not present; in the latter case equations are there but only implicitly.

Secondly and more importantly, by considering higher-dimensional, nested algebraic structures, we can now compose different coalgebras as well as different states of the same coalgebra. In this way the current work can be seen as a higher-dimensional extension of the existing bialgebraic studies (which focus on "inner" algebraic structures).

Organization of the paper We shall not dive into our 2-categorical exploration from the beginning. In Section 2, we instead focus on one specific algebraic theory, namely the one for parallel composition of systems. Our emphasis there is on the fact that the sync natural transformation essentially gives rise to parallel composition $\|$, and the fact that equational properties of $\|$ (such as associativity) can be reduced to the corresponding equational properties of sync.

These concrete observations will provide us with intuition for abstract categorical constructs in Section 3, where we formalize the microcosm principle for an arbitrary Lawvere theory $\mathbb{L}$. Results on coalgebras such as compositionality are proved here in their full generality and abstraction.

In this paper we shall focus on *strict* algebraic structures on categories in order to avoid complicated coherence issues. This means for example that we only consider *strict* monoidal categories for which the isomorphisms in (3) are in fact equalities. However, we have also obtained some preliminary observations on relaxed ("pseudo" or "strong") algebraic structures: see Section 3.3.

2 Parallel composition of coalgebras

2.1 Parallel composition via sync natural transformation

Let us start with the equation (2), a coalgebraic representation of compositionality. The operator $\|$ on the left is of type $\mathbf{Coalg}_F \times \mathbf{Coalg}_F \to \mathbf{Coalg}_F$. It is natural to require functoriality of this operation, making it a *bifunctor*. A bifunctor—especially an associative one which we investigate in Section 2.3—plays an important role in various applications of category theory. Usually such an (associative) bifunctor is called a *tensor* and denoted by $\otimes$, a convention that we also follow. Therefore the "compositionality" statement now looks as follows.

$$\mathsf{beh}\begin{pmatrix} FX & & FY \\ c\uparrow & \otimes & d\uparrow \\ X & & Y \end{pmatrix} = \mathsf{beh}\begin{pmatrix} FX \\ c\uparrow \\ X \end{pmatrix} \Big\| \ \mathsf{beh}\begin{pmatrix} FY \\ d\uparrow \\ Y \end{pmatrix} \qquad (4)$$

The first question is: when do we have such a tensor $\otimes$ on $\mathbf{Coalg}_F$? In many applications of coalgebras, it is obtained by lifting a tensor $\otimes$ on the base category $\mathbb{C}$ to $\mathbf{Coalg}_F$.[6] Such a lifting is possible in presence of a natural transformation

$$FX \otimes FY \overset{\mathsf{sync}_{X,Y}}{\longrightarrow} F(X \otimes Y), \quad \text{used in} \quad \begin{array}{c} FX \\ c\uparrow \\ X \end{array} \;\otimes\; \begin{array}{c} FY \\ d\uparrow \\ Y \end{array} \;:=\; \begin{array}{c} F(X \otimes Y) \\ \uparrow \mathsf{sync}_{X,Y} \\ FX \otimes FY \\ \uparrow c \otimes d \\ X \otimes Y \end{array} \;. \tag{5}$$

We shall call this sync a *synchronization* because its computational meaning is indeed a specification of the way two systems synchronize. This will be illustrated in the coming examples.

Once we have an outer parallel composition $\otimes$, an inner operator $\|$ which composes behavior (i.e. states of the final coalgebra) is also obtained immediately by coinduction as on the right. Compositionality (4) is also straightforward by finality: both sides of the equation are the unique coalgebra morphism from $c \otimes d$ to the final ζ. The following theorem summarizes the observations so far.

$$\begin{tikzcd} F(Z \otimes Z) \arrow[r, dashed] & FZ \\ Z \otimes Z \arrow[u, "\zeta \otimes \zeta"] \arrow[r, dashed, "\|"'] & Z \arrow[u, "\text{final}"', "\zeta"] \end{tikzcd}$$

Theorem 2.1 (Coalgebraic compositionality) *Assume that a category $\mathbb{C}$ has a tensor $\otimes : \mathbb{C} \times \mathbb{C} \to \mathbb{C}$ and an endofunctor $F : \mathbb{C} \to \mathbb{C}$ has a natural transformation $\mathsf{sync}_{X,Y} : FX \otimes FY \to F(X \otimes Y)$. If moreover there exists a final F-coalgebra, then:*

1. *The tensor $\otimes$ on $\mathbb{C}$ lifts to an "outer" composition operator $\otimes : \mathbf{Coalg}_F \times \mathbf{Coalg}_F \to \mathbf{Coalg}_F$.*
2. *We obtain an "inner" composition operator $\|: Z \otimes Z \to Z$ by coinduction.*
3. *Between the two composition operators the compositionality property (4) holds.*

□

We can put the compositionality property (4) in more abstract terms as "the functor $\mathsf{beh} : \mathbf{Coalg}_F \to \mathbb{C}/Z$ preserves a tensor," meaning that the diagram below left commutes. Here a tensor $\underline{\otimes}$ on the slice category $\mathbb{C}/Z$ is given as on the right, using the inner composition $\|$.

$$\begin{tikzcd} \mathbf{Coalg}_F \times \mathbf{Coalg}_F \arrow[r, "\mathsf{beh} \times \mathsf{beh}"] \arrow[d, "\otimes"'] & \mathbb{C}/Z \times \mathbb{C}/Z \arrow[d, "\underline{\otimes}"] \\ \mathbf{Coalg}_F \arrow[r, "\mathsf{beh}"'] & \mathbb{C}/Z \end{tikzcd} \qquad \left(\begin{array}{c} X \\ \downarrow f \\ Z \end{array}, \begin{array}{c} Y \\ \downarrow g \\ Z \end{array} \right) \overset{\underline{\otimes}}{\longmapsto} \begin{array}{c} X \otimes Y \\ \downarrow f \otimes g \\ Z \otimes Z \\ \downarrow \| \\ Z \end{array} \tag{6}$$

The point of Theorem 2.1 is as follows. Those parallel composition operators which are induced by sync are well-behaved ones: good properties like compositionality come for free. We shall present some examples in Section 2.2.

Remark 2.2 The view of parallel composition of systems as a tensor structure on $\mathbf{Coalg}_F$ has been previously presented in [13]. The interest there is on categorical

[6] Note that we use boldface $\otimes$ for a tensor on $\mathbf{Coalg}_F$ to distinguish it from $\otimes$ on $\mathbb{C}$.

structures on $\mathbf{Coalg}_F$ rather than on properties of parallel composition such as compositionality. In [13] and other literature an endofunctor F with sync (equipped with some additional compatibility) is called a *monoidal endofunctor*.[7]

2.2 Examples

In $\mathbf{Sets}$: bisimilarity is a congruence We shall focus on LTSs and bisimilarity as their process semantics. For this purpose it is appropriate to take $\mathbf{Sets}$ as our base category $\mathbb{C}$ and $\mathcal{P}_\omega(\Sigma \times \_)$ as the functor F. We use Cartesian products as a tensor on $\mathbf{Sets}$. This means that a composition of two coalgebras has the product of the two state spaces as its state space, which matches our intuition. The functor $\mathcal{P}_\omega$ in F is the finite powerset functor; the finiteness assumption is needed for existence of a final F-coalgebra. It is standard (see e.g. [26]) that a final F-coalgebra captures bisimilarity via coinduction.

In considering parallel composition of LTSs, the following two examples are well-known ones.[8]

- *CSP-style* [7]: $a.P \parallel a.Q \xrightarrow{a} P \parallel Q$. For the whole system to make an a-action, each component has to make an a-action.
- *CCS-style* [21]: $a.P \parallel \overline{a}.Q \xrightarrow{\tau} P \parallel Q$, assuming $\Sigma = \{a, b, \dots\} \cup \{\overline{a}, \overline{b}, \dots\} \cup \{\tau\}$. When one component outputs on a channel a and another inputs from a, then the whole system makes an internal τ move.

In fact, each of these different ways of synchronization can be represented by a suitable sync natural transformation.

$$\begin{array}{ccc}
\mathcal{P}_\omega(\Sigma \times X) \times \mathcal{P}_\omega(\Sigma \times Y) & \longrightarrow & \mathcal{P}_\omega\big(\Sigma \times (X \times Y)\big) \\
(u, v) & \overset{\mathsf{sync}^{\mathrm{CSP}}_{X,Y}}{\longmapsto} & \big\{\, (a, (x, y)) \mid (a, x) \in u \wedge (a, y) \in v \,\big\} \\
(u, v) & \overset{\mathsf{sync}^{\mathrm{CCS}}_{X,Y}}{\longmapsto} & \big\{\, (\tau, (x, y)) \mid (a, x) \in u \wedge (\overline{a}, y) \in v \,\big\}
\end{array}$$

By Theorem 2.1, each of these gives (different) $\otimes$ on $\mathbf{Coalg}_F$, and $\parallel$ on Z; moreover the behavior functor beh satisfies compositionality. In other words: bisimilarity is a congruence with respect to both CSP-style and CCS-style parallel composition.

Remark 2.3 As mentioned in the introduction, in some ways this paper can be seen as an extension of the bialgebraic studies started in [27]. However there is also a drawback, namely the limited expressive power of $\mathsf{sync} : FX \otimes FY \to F(X \otimes Y)$.

Our sync specifies the way an algebraic structure interacts with a coalgebraic one. In this sense it is a counterpart of a distributive law $\Sigma F \Rightarrow F\Sigma$ in [27] representing operational rules, where Σ is a functor induced by an algebraic signature. However there are many common operational rules which do not allow representation of the

[7] Later in Section 3 we will observe that a functor F with sync is a special case of a *lax* $\mathbb{L}$-functor, by choosing a suitable algebraic theory $\mathbb{L}$. Such a functor F with sync is usually called a monoidal functor (as opposed to a *lax* monoidal functor), probably because it preserves (inner) monoid objects; see Proposition 3.8.1.

[8] Here we focus on synchronous interaction. Both CSP and CCS have an additional kind of interaction, namely an "interleaving" one; see Remark 2.3.

form $\Sigma F \Rightarrow F\Sigma$; therefore in [27] the type of such a distributive law is eventually extended to $\Sigma(F \times \mathrm{id}) \Rightarrow F\Sigma^*$. The class of rules representable in this form coincides with the class of so-called *GSOS-rules*.

At present it is not clear how we can make a similar extension for our sync; consequently there are some operational rules which we cannot model by sync. One important example is an *interleaving* kind of interaction—such as $a.P \parallel Q \xrightarrow{a} P \parallel Q$ which leaves the second component unchanged. This is taken care of in [27] by the identity functor (id) appearing on the left-hand side of $\Sigma(F \times \mathrm{id}) \Rightarrow F\Sigma^*$. For our sync to be able to model such interleaving, we can replace F by the cofree comonad on it, as is done in [13, Example 3.11]. This extension should be straightforward but detailed treatment is left as future work.

In $\mathcal{K}\ell(T)$: trace equivalence is a congruence In our recent work [6] we extend earlier observations in [10, 25] and show that trace semantics—including trace *set* semantics for non-deterministic systems and trace *distribution* semantics for probabilistic systems—is also captured by coinduction when it is employed in a Kleisli category $\mathcal{K}\ell(T)$. Applying the present composition framework, we can conclude that trace semantics is compositional with respect to well-behaved parallel composition. The details are omitted here due to lack of space.

2.3 Equational properties of parallel composition

Now we shall investigate equational properties—associativity, commutativity, and so on—of parallel composition $\circledast$, which we have ignored deliberately for simplicity of argument. We present our result in terms of associativity; it is straightforward to transfer the result to other properties like commutativity. The main point of the following theorem is as follows: if $\otimes$ is associative and sync is "associative," then the lifting $\circledast$ is associative. The proof is straightforward.

Theorem 2.4 *Let $\mathbb{C}$ be a category with a strictly associative tensor $\otimes$,*[9] *and $F : \mathbb{C} \to \mathbb{C}$ be a functor with $\mathsf{sync} : FX \otimes FY \to F(X \otimes Y)$. If the diagram*

$$\begin{array}{ccccc} FX \otimes (FY \otimes FZ) & \xrightarrow{FX \otimes \mathsf{sync}} & FX \otimes F(Y \otimes Z) & \xrightarrow{\mathsf{sync}} & F(X \otimes (Y \otimes Z)) \\ \downarrow \mathrm{id} & & & & \downarrow \mathrm{id} \\ (FX \otimes FY) \otimes FZ & \xrightarrow[\mathsf{sync} \otimes FZ]{} & F(X \otimes Y) \otimes FZ & \xrightarrow[\mathsf{sync}]{} & F((X \otimes Y) \otimes Z) \end{array} \tag{7}$$

commutes, then the lifted tensor $\circledast$ on $\mathbf{Coalg}_F$ is strictly associative. □

The two identity arrows in (7) are available due to strict associativity of $\otimes$. In the next section we shall reveal the generic principle behind the commutativity condition of (7), namely a coherence condition on a lax natural transformation.

As an example, $\mathsf{sync}^{\mathrm{CSP}}$ and $\mathsf{sync}^{\mathrm{CCS}}$ in Section 2.2 are easily seen to be "associative" in the sense of the diagram (7). Therefore the resulting tensors $\circledast$ are strictly associative.

[9] As mentioned already, in this paper we stick to *strict* algebraic structures.

3 Formalizing the microcosm principle

In this section we shall formalize the microcosm principle for an arbitrary algebraic theory presented as a Lawvere theory $\mathbb{L}$. This and the subsequent results generalize the results in the previous section. In particular, we will obtain a general compositionality result which works for an arbitrary algebraic theory.

As we sketched in the introduction, an outer model will be a product-preserving functor $\mathbb{C} : \mathbb{L} \to \mathbf{CAT}$; an inner model inside will be a lax natural transformation $X : \mathbf{1} \Rightarrow \mathbb{C}$. Here $\mathbf{1} : \mathbb{L} \to \mathbf{CAT}$ is the constant functor which maps everything to the category $\mathbf{1}$ with one object and one arrow (which is a special case of an outer model). Mediating 2-cells for the *lax* natural transformation X play a crucial role as inner interpretation of algebraic operations. In this section we heavily rely on 2-categorical notions, about which detailed accounts can be found in [4].

$$\begin{tikzcd} \mathbb{L} \arrow[r, bend left=50, "\mathbf{1}"{name=U}] \arrow[r, "\mathbb{C}"'{name=D}] & \mathbf{CAT} \arrow[from=U, to=D, phantom, "\Downarrow X"] \end{tikzcd}$$

3.1 Lawvere theories

Lawvere theories are categorical presentations of algebraic theories. The notion is introduced in [18] (not under this name, though) aiming at a categorical formulation of "theories" and "semantics." An accessible introduction to the notion can be found in [17]. Lawvere theories are known to be equivalent to *finitary monads*. These two ways of presenting algebraic theories have been widely used in theoretical computer science, e.g. for modeling computation with effect [8, 22]. Recent developments (such as [24]) utilize the increased expressive power of *enriched* Lawvere theories.

In the sequel, by an *FP-category* we refer to a category with (a choice of) finite products. An *FP-functor* is a functor between FP-categories which preserves finite products "on-the-nose," that is, up-to-equality instead of up-to-isomorphism.

Definition 3.1 (Lawvere theory) By $\mathbf{Nat}$ we denote the category of natural numbers (as sets) and functions between them. Therefore every arrow in $\mathbf{Nat}$ is a (cotuple of) coprojection; an arrow in $\mathbf{Nat}^{\mathrm{op}}$ is a (tuple of) projection.[10]

A *Lawvere theory* is a small FP-category $\mathbb{L}$ equipped with an FP-functor $H : \mathbf{Nat}^{\mathrm{op}} \to \mathbb{L}$ which is bijective on objects. We shall denote an object of $\mathbb{L}$ by a natural number k, identifying $k \in \mathbf{Nat}^{\mathrm{op}}$ and $Hk \in \mathbb{L}$.

The category $\mathbf{Nat}^{\mathrm{op}}$—which is a free FP-category on the trivial category $\mathbf{1}$—is there in order to specify the choice of finite products in $\mathbb{L}$. For illustration, we make some remarks on $\mathbb{L}$'s objects and arrows.

- An object $k \in \mathbb{L}$ is a k-fold product $1 \times \cdots \times 1$ of 1.
- An arrow in $\mathbb{L}$ is intuitively understood as an algebraic operation. That is, $k \to 1$ as a k-ary operation; and $k \to n$ as an n-tuple $\langle \mathsf{f}_1, \dots, \mathsf{f}_n \rangle$ of k-ary operations. To be precise, arrows in $\mathbb{L}$ also include projections (such as $\pi_1 : 2 \to 1$) and *terms* made up of operations and projections (such as $\mathsf{m} \circ \langle \pi_1, \pi_2 \rangle : 3 \to 1$).

[10] An arrow $f : n \to k$ in $\mathbf{Nat}$ can be written as a cotuple $[\kappa_{f(1)}, \dots, \kappa_{f(n)}]$ where $\kappa_i : 1 \to k$ is the coprojection into the i-th summand of $1 + \cdots + 1$ (k times).

Conventionally in universal algebra, an algebraic theory is presented by an *algebraic specification* (Σ, E)—a pair of a set Σ of operations and a set E of equations. A Lawvere theory $\mathbb{L}$ arises from such (Σ, E) as its so-called *classifying category* (see e.g. [9, 18]). An arrow $k \to n$ in the resulting Lawvere theory $\mathbb{L}$ is an n-tuple $([t_1(\overrightarrow{x})], \ldots, [t_n(\overrightarrow{x})])$ of Σ-terms with k variables $\overrightarrow{x}$, where $[\_]$ denotes taking an equivalence class modulo equations in E. An equivalent way to describe this construction is via *sketches*: (Σ, E) is identified with an FP-sketch, which in turn induces $\mathbb{L}$ as a free FP-category. See [2] for details.

Our leading example is the Lawvere theory $\mathbf{Mon}$ for monoids.[11] It arises as a classifying category from the well-known algebraic specification of monoids. This specification has a nullary operation e and a binary one m; subject to the equations $\mathsf{m}(x, \mathsf{e}) = x$, $\mathsf{m}(\mathsf{e}, x) = x$, and $\mathsf{m}(x, \mathsf{m}(y, z)) = \mathsf{m}(\mathsf{m}(x, y), z)$.

Equivalently, $\mathbf{Mon}$ is the freely generated FP-category by arrows $0 \stackrel{\mathsf{e}}{\to} 1$ and $2 \stackrel{\mathsf{m}}{\to} 1$ subject to the commutativity on the right. These data (arrows and commutative diagrams) form an FP-sketch (see [2]).

$$\begin{tikzcd} 1 \arrow[r, "\langle \mathrm{id}, \mathsf{e}\rangle"] \arrow[dr, "\mathrm{id}"'] & 2 \arrow[d, "\mathsf{m}"] & 1 \arrow[l, "\langle \mathsf{e}, \mathrm{id}\rangle"'] \arrow[dl, "\mathrm{id}"] \\ & 1 & \end{tikzcd} \qquad \begin{tikzcd} 3 \arrow[r, "\mathsf{m}\times\mathrm{id}"] \arrow[d, "\mathrm{id}\times\mathsf{m}"'] & 2 \arrow[d, "\mathsf{m}"] \\ 2 \arrow[r, "\mathsf{m}"'] & 1 \end{tikzcd}$$

3.2 Outer models: $\mathbb{L}$-categories

We start by formalizing an outer model. It is a category with an $\mathbb{L}$-structure, hence called an $\mathbb{L}$-*category*. It is standard that a (set-theoretic) model of $\mathbb{L}$—a *set* with an $\mathbb{L}$-structure—is identified with an FP-functor $\mathbb{L} \stackrel{X}{\to} \mathbf{Sets}$. Concretely, let $X = X1$ be the image of $1 \in \mathbb{L}$. Then $k \in \mathbb{L}$ must be sent to X^k due to preservation of finite products. Now the functor's action on arrows is what interprets $\mathbb{L}$'s operations in X, as illustrated above right. Equations (expressed as commutative diagrams in $\mathbb{L}$) are satisfied because a functor preserves commutativity.

$$\begin{array}{ccc} \mathbb{L} & \stackrel{X}{\longrightarrow} & \mathbf{Sets} \\ 2 & & X^2 \\ \downarrow\mathsf{m} & \longmapsto & \downarrow[\![\mathsf{m}]\!] \\ 1 & & X \end{array}$$

Turning back to $\mathbb{L}$-*categories*, what we have to do here is to just replace $\mathbf{Sets}$ by the category $\mathbf{CAT}$ of (possibly large and locally small) categories.

Definition 3.2 ($\mathbb{L}$-categories, $\mathbb{L}$-functors) A (strict) $\mathbb{L}$-*category* is an FP-functor $\mathbb{L} \stackrel{\mathbb{C}}{\to} \mathbf{CAT}$. In the sequel we denote the image $\mathbb{C}1$ of $1 \in \mathbb{L}$ by $\mathbb{C}$; and the image $\mathbb{C}(\mathsf{f})$ of an arrow f by $[\![\mathsf{f}]\!]$.

An $\mathbb{L}$-*functor* $F : \mathbb{C} \to \mathbb{D}$—a functor preserving an $\mathbb{L}$-structure—is a natural transformation $\begin{tikzcd} \mathbb{L} \arrow[r, bend left, "\mathbb{C}"{name=U}] \arrow[r, bend right, "\mathbb{D}"'{name=D}] & \mathbf{CAT} \arrow[from=U, to=D, Rightarrow, "F"] \end{tikzcd}$.

Another way to look at the previous definition is to view an $\mathbb{L}$-structure as "factorization through $\mathbf{Nat}^{\mathrm{op}} \to \mathbb{L}$." We can identify a category $\mathbb{C} \in \mathbf{CAT}$ with a functor $\mathbf{1} \to \mathbf{CAT}$, which is in turn identified with an FP-functor $\mathbf{Nat}^{\mathrm{op}} \to \mathbf{CAT}$, because $\mathbf{Nat}^{\mathrm{op}}$ is the free FP-category on $\mathbf{1}$. We say that $\mathbb{C}$ has an $\mathbb{L}$-structure, if this FP-functor factors through $H : \mathbf{Nat}^{\mathrm{op}} \to \mathbb{L}$ (as below left). Note that the factorization is

[11] The Lawvere theory $\mathbf{Mon}$ for the theory of monoids should not be confused with the category of (set-theoretic) monoids and monoid homomorphisms (which is often denoted by $\mathbf{Mon}$ as well).

not necessarily unique, because there can be different ways of interpreting the algebraic theory $\mathbb{L}$ in $\mathbb{C}$. Similarly, a functor $\mathbb{C} \xrightarrow{F} \mathbb{D}$ is identified with a natural transformation $\mathbf{1} \underset{}{\overset{}{\Downarrow F}} \mathbf{CAT}$; and then with $\mathbf{Nat}^{\mathrm{op}} \Downarrow F \; \mathbf{CAT}$ due to the 2-universality of $\mathbf{Nat}^{\mathrm{op}}$ as a free object. We say that this F preserves an $\mathbb{L}$-structure, if the last natural transformation factors through $H : \mathbf{Nat}^{\mathrm{op}} \to \mathbb{L}$ (as below right).

$$
\begin{tikzcd}
\mathbf{Nat}^{\mathrm{op}} \arrow[r, "H"] \arrow[dr, "\mathbb{C}"'] & \mathbb{L} \arrow[d] \\
& \mathbf{CAT}
\end{tikzcd}
\qquad
\begin{tikzcd}
\mathbf{Nat}^{\mathrm{op}} \arrow[r, "H"] \arrow[dr, bend right] \arrow[dr] \arrow[dr, phantom, "F\Downarrow"'] & \mathbb{L} \arrow[d, bend right] \arrow[d, bend left] \arrow[d, phantom, "\Leftarrow"] \\
& \mathbf{CAT}
\end{tikzcd}
$$

Example 3.3 The usual notion of strictly monoidal categories coincides with $\mathbb{L}$-categories for $\mathbb{L} = \mathbf{Mon}$. A tensor $\otimes$ and a unit I on a category arise as interpretation of the operations $2 \xrightarrow{\mathsf{m}} 1$ and $0 \xrightarrow{\mathsf{e}} 1$; commuting diagrams in $\mathbf{Mon}$ such as $\mathsf{m} \circ \langle \mathrm{id}, \mathsf{e} \rangle = \mathrm{id}$ yield equational properties of $\otimes$ and I.

3.3 Remarks on "pseudo" algebraic structures

As we mentioned in the introduction, in this paper we focus on *strict* algebraic structures. This means that monoidal categories (in which associativity holds only up-to-isomorphism, for example) fall out of our consideration. Extending our current framework to such "pseudo" algebraic structures is one important direction of our future work. Such an extension is not entirely obvious; we shall sketch some preliminary observations in this direction.

The starting point is to relax the definition of $\mathbb{L}$-categories from (strict) functors $\mathbb{L} \to \mathbf{CAT}$ to *pseudo* functors, meaning that composition and identities are preserved only up-to-isomorphism. Then it is not hard to see that a pseudo functor $\mathbf{Mon} \xrightarrow{\mathbb{C}} \mathbf{CAT}$ (which preserves finite products in a suitable sense) gives rise to a monoidal category. Indeed, let us denote a mediating iso-2-cell for composition by $\mathbb{C}_{\mathsf{g},\mathsf{f}} : [\![\mathsf{g}]\!] \circ [\![\mathsf{f}]\!] \overset{\cong}{\Rightarrow} [\![\mathsf{g} \circ \mathsf{f}]\!]$. The associativity diagram (below left) gives rise to the two iso-2-cells on the right.

$$
\underline{\text{in } \mathbf{Mon}} \quad
\begin{tikzcd}
3 \arrow[r, "\mathsf{m}\times\mathrm{id}"] \arrow[d, "\mathrm{id}\times\mathsf{m}"'] & 2 \arrow[d, "\mathsf{m}"] \\
2 \arrow[r, "\mathsf{m}"'] & 1
\end{tikzcd}
\qquad
\underline{\text{in } \mathbf{CAT}} \quad
\begin{tikzcd}[column sep=huge]
\mathbb{C}^3 \arrow[r, "{[\![\mathsf{m}\times\mathrm{id}]\!]}"] \arrow[d, "{[\![\mathrm{id}\times\mathsf{m}]\!]}"'] \arrow[dr, "{[\![\mathsf{m}\circ(\mathsf{m}\times\mathrm{id})]\!]=[\![\mathsf{m}\circ(\mathrm{id}\times\mathsf{m})]\!]}" description] \arrow[dr, phantom, shift left=3ex, "\mathbb{C}_{\mathsf{m},\mathsf{m}\times\mathrm{id}}\Swarrow\cong"] \arrow[dr, phantom, shift right=3ex, "\cong\Nearrow\mathbb{C}_{\mathsf{m},\mathrm{id}\times\mathsf{m}}"'] & \mathbb{C}^2 \arrow[d, "{[\![\mathsf{m}]\!]}"] \\
\mathbb{C}^2 \arrow[r, "{[\![\mathsf{m}]\!]}"'] & \mathbb{C}
\end{tikzcd}
\qquad (8)
$$

The composition $\mathbb{C}^{-1}_{\mathsf{m},\mathrm{id}\times\mathsf{m}} \bullet \mathbb{C}_{\mathsf{m},\mathsf{m}\times\mathrm{id}}$ is what gives us a natural isomorphism $\alpha : X \otimes (Y \otimes Z) \overset{\cong}{\Rightarrow} (X \otimes Y) \otimes Z$. Moreover, the coherence condition on such isomorphisms in a monoidal category (like the famous pentagon diagram; see [20]) follows from the coherence condition on mediating 2-cells of a pseudo functor (see [4]).

So far so good. However, at this moment it is not clear what is a canonical construction the other way round, i.e. from a monoidal category to a pseudo functor.[12] In the present paper we side-step these 2-categorical subtleties by restricting ourselves to strict, non-pseudo functors.

[12] For example, given a monoidal category $\mathbb{C}$, we need to define a functor $[\![\mathsf{m} \circ (\mathsf{m} \times \mathrm{id})]\!] = [\![\mathsf{m} \circ (\mathrm{id} \times \mathsf{m})]\!]$ in (8). It's not clear whether it should carry (X, Y, Z) to $X \otimes (Y \otimes Z)$, or to $(X \otimes Y) \otimes Z$.

3.4 Inner models: $\mathbb{L}$-objects

We proceed to formalize an inner model. It is an object in an $\mathbb{L}$-category which itself carries an (inner) $\mathbb{L}$-structure, hence is called an $\mathbb{L}$-*object*. A monoid object in a monoidal category is a prototypical example. We first present an abstract definition; some illustration follows afterwards.

Definition 3.4 ($\mathbb{L}$-objects) An $\mathbb{L}$-*object* X in an $\mathbb{L}$-category $\mathbb{C}$ is a lax natural transformation $X : \mathbf{1} \Rightarrow \mathbb{C}$ (below left) which is "product-preserving": this means that the composition $X \circ H$ (below right) is strictly, non-lax natural. Here $\mathbf{1} : \mathbb{L} \to \mathbf{CAT}$ denotes the constant functor to the trivial one-object category $\mathbf{1}$.

$$\begin{tikzcd} \mathbb{L} \arrow[r, bend left, "\mathbf{1}"] \arrow[r, "\mathbb{C}"'] \arrow[r, phantom, "\Downarrow X"] & \mathbf{CAT} \end{tikzcd} \qquad \begin{tikzcd} \mathbf{Nat}^{\mathrm{op}} \arrow[r, "H"] & \mathbb{L} \arrow[r, bend left, "\mathbf{1}"] \arrow[r, "\mathbb{C}"'] \arrow[r, phantom, "\Downarrow X"] & \mathbf{CAT} \end{tikzcd}$$

Such a nested algebraic structure—formalized as an $\mathbb{L}$-object in an $\mathbb{L}$-category—shall be called a *microcosm model* for $\mathbb{L}$.

Let us now illustrate the definition. First, X's component at $1 \in \mathbb{L}$ is a functor $\mathbf{1} \xrightarrow{X_1} \mathbb{C}$ which is identified with an object $X \in \mathbb{C}$. This is the "carrier" object of this inner algebra. Moreover, any other component $\mathbf{1} \xrightarrow{X_k} \mathbb{C}^k$ must be the k-tuple $(X, \dots, X) \in \mathbb{C}^k$ of X's. This is because of (strict) naturality of $X \circ H$ (see above right): for any $i \in [1, k]$ the composite $\pi_i \circ X_k$ is required to be X_1.

$$\begin{array}{cc} \underline{\text{in } \mathbf{Nat}^{\mathrm{op}}} & \underline{\text{in } \mathbf{CAT}} \\ \begin{tikzcd} k \arrow[d, "\pi_i"] \\ 1 \end{tikzcd} & \begin{tikzcd} \mathbf{1} \arrow[r, "X_k=(X,\dots,X)"] \arrow[d, equal] \arrow[dr, phantom, "\swarrow\!\!\!/"] & \mathbb{C}^k \arrow[d, "{[\![H\pi_i]\!] = \pi_i}"] \\ \mathbf{1} \arrow[r, "X_1=X"'] & \mathbb{C} \end{tikzcd} \end{array}$$

The (inner) algebraic structure on X arises in the form of mediating 2-cells of the *lax* natural transformation. For each arrow $k \xrightarrow{\mathsf{f}} n$ in $\mathbb{L}$, lax naturality of X requires existence of a mediating 2-cell $X_{\mathsf{f}} : [\![\mathsf{f}]\!] \circ X_k \Rightarrow X_n$. The diagram (above right) shows the situation when we set $\mathsf{f} = \mathsf{m}$, a binary operation. The natural transformation X_{m} can be identified with an arrow $X \otimes X \xrightarrow{\mu} X$ in $\mathbb{C}$, which gives an inner binary operation on X.

$$\begin{array}{cc} \underline{\text{in } \mathbb{L}} & \underline{\text{in } \mathbf{CAT}} \\ \begin{tikzcd} 2 \arrow[d, "\mathsf{m}"] \\ 1 \end{tikzcd} & \begin{tikzcd} \mathbf{1} \arrow[r, "X_2=(X,X)"] \arrow[d, equal] \arrow[dr, phantom, "\swarrow X_{\mathsf{m}}"] & \mathbb{C}^2 \arrow[d, "{[\![\mathsf{m}]\!]=\otimes}"] \\ \mathbf{1} \arrow[r, "X"'] & \mathbb{C} \end{tikzcd} \end{array}$$

How do such inner operations on X satisfy equations as specified in $\mathbb{L}$? The key is the coherence condition[13] on mediating 2-cells: it requires $X_{\mathrm{id}} = \mathrm{id}$ concerning identities; and $X_{\mathsf{g} \circ \mathsf{f}} = X_{\mathsf{g}} \bullet ([\![\mathsf{g}]\!] \circ X_{\mathsf{f}})$ concerning composition (as on the right). The following example illustrates how such coherence induces equational properties.

$$X_{\mathsf{g} \circ \mathsf{f}} = \begin{tikzcd} \mathbf{1} \arrow[r] \arrow[d, equal] \arrow[dr, phantom, "\swarrow X_{\mathsf{f}}"] & \mathbb{C}^l \arrow[d, "{[\![\mathsf{f}]\!]}"] \\ \mathbf{1} \arrow[r] \arrow[d, equal] \arrow[dr, phantom, "\swarrow X_{\mathsf{g}}"] & \mathbb{C}^k \arrow[d, "{[\![\mathsf{g}]\!]}"] \\ \mathbf{1} \arrow[r] & \mathbb{C}^n \end{tikzcd}$$

Example 3.5 A monoid object in a strictly monoidal category is an example of an $\mathbb{L}$-object in an $\mathbb{L}$-category. Here we take $\mathbb{L} = \mathbf{Mon}$, the theory of monoids.

For illustration, let us here derive associativity of multiplication $X \otimes X \xrightarrow{\mu} X$. In the current setting the multiplication μ is identified with a mediating 2-cell X_{m} as above.

[13] This is part of the notion of lax natural transformations; see [4].

The coherence condition yields the two equalities $(*)$ below.

$$
\underline{\text{in } \mathbb{L}}
\quad
\begin{tikzcd}[column sep=small]
 & 3 \arrow[dl, "\mathsf{id}\times\mathsf{m}"'] \arrow[dr, "\mathsf{m}\times\mathsf{id}"] & \\
2 \arrow[dr, "\mathsf{m}"'] & & 2 \arrow[dl, "\mathsf{m}"] \\
 & 1 &
\end{tikzcd}
\qquad
\underline{\text{in } \mathbf{CAT}}
\quad
\begin{tikzcd}
\mathbf{1} \arrow[r] \arrow[d, equal] \arrow[dr, phantom, "\Swarrow X_{\mathsf{id}\times\mathsf{m}}"] & \mathbb{C}^3 \arrow[d, "{[\![\mathsf{id}\times\mathsf{m}]\!]}"] \\
\mathbf{1} \arrow[r] \arrow[d, equal] \arrow[dr, phantom, "\Swarrow X_{\mathsf{m}}"] & \mathbb{C}^2 \arrow[d, "{[\![\mathsf{m}]\!]}"] \\
\mathbf{1} \arrow[r] & \mathbb{C}
\end{tikzcd}
\overset{(*)}{=}
\begin{tikzcd}
\mathbf{1} \arrow[r] \arrow[d, equal] \arrow[dr, phantom, "{\Swarrow \begin{array}{l} X_{\mathsf{m}\circ(\mathsf{id}\times\mathsf{m})} \\ = X_{\mathsf{m}\circ(\mathsf{m}\times\mathsf{id})} \end{array}}"] & \mathbb{C}^3 \arrow[d] \\
\mathbf{1} \arrow[r] & \mathbb{C}^1
\end{tikzcd}
\overset{(*)}{=}
\begin{tikzcd}
\mathbf{1} \arrow[r] \arrow[d, equal] \arrow[dr, phantom, "\Swarrow X_{\mathsf{m}\times\mathsf{id}}"] & \mathbb{C}^3 \arrow[d, "{[\![\mathsf{m}\times\mathsf{id}]\!]}"] \\
\mathbf{1} \arrow[r] \arrow[d, equal] \arrow[dr, phantom, "\Swarrow X_{\mathsf{m}}"] & \mathbb{C}^2 \arrow[d, "{[\![\mathsf{m}]\!]}"] \\
\mathbf{1} \arrow[r] & \mathbb{C}
\end{tikzcd}
$$

Now it is not hard to see that: the composed 2-cell on the left corresponds to $X^3 \xrightarrow{X\times\mu} X^2 \xrightarrow{\mu} X$; and the one on the right corresponds to $X^3 \xrightarrow{\mu\times X} X^2 \xrightarrow{\mu} X$. The equalities $(*)$ above prove that these two arrows $X^3 \rightrightarrows X$ are identical.

3.5 Microcosm structures in coalgebras

In this section we return to our original question and apply the framework we just introduced to coalgebraic settings. First we present some basic results, which are used later in our main result of general compositionality. The constructs in Section 2 (such as sync) will appear again, now in their generalized form. Some details and proofs are omitted here due to lack of space. They will appear in the forthcoming extended version of this paper, although the diligent reader will readily work them out.

Let $\mathbb{C}$ be an $\mathbb{L}$-category, and $F : \mathbb{C} \to \mathbb{C}$ be a functor. We can imagine that, for the category $\mathbf{Coalg}_F$ to carry an $\mathbb{L}$-structure, F needs to be somehow compatible with $\mathbb{L}$; it turns out that the following condition is sufficient. It is weaker than F's being an $\mathbb{L}$-functor (see Definition 3.2).

Definition 3.6 (Lax $\mathbb{L}$-functor) A functor $F : \mathbb{C} \to \mathbb{D}$ between $\mathbb{L}$-categories is said to be a *lax* $\mathbb{L}$*-functor* if it is identified with[14] some lax natural transformation $\begin{tikzcd} \mathbb{L} \arrow[r, bend left, "\mathbb{C}"{name=U}] \arrow[r, bend right, "\mathbb{D}"'{name=D}] \arrow[from=U, to=D, phantom, "\Downarrow F"] & \mathbf{CAT} \end{tikzcd}$ which is product-preserving (i.e. $F \circ H$ is strictly natural; see Definition 3.4).

$$
\underline{\text{in } \mathbb{L}}
\quad
\begin{tikzcd}
2 \arrow[d, "\mathsf{m}"] \\
1
\end{tikzcd}
\qquad
\underline{\text{in } \mathbf{CAT}}
\quad
\begin{tikzcd}
\mathbb{C}^2 \arrow[r, "{(F,F)}"] \arrow[d, "\otimes"'] \arrow[dr, phantom, "\Swarrow F_{\mathsf{m}}"] & \mathbb{C}^2 \arrow[d, "\otimes"] \\
\mathbb{C} \arrow[r, "F"'] & \mathbb{C}
\end{tikzcd}
$$

Lax $\mathbb{L}$-endofunctors are natural generalization of functors with sync as in Section 2. To illustrate this, look at the lax naturality diagram on the right for a binary operation m. Here we denote the outer interpretation $[\![\mathsf{m}]\!]$ by $\otimes$. The 2-component is $F_2 = (F, F)$ because the lax natural transformation F is product-preserving. The mediating 2-cell F_{m} can be identified with a natural transformation $FX \otimes FY \to F(X \otimes Y)$; this is what we previously called sync. Moreover, F_{m} (as generalized sync) is automatically compatible with equational properties (as in Theorem 2.4); this is because of the coherence condition on mediating 2-cells like "$F_{\mathsf{g}\circ\mathsf{f}}$ is a suitable composition of F_{g} after F_{f}."

The following results follow from a more general result concerning the notion of *inserters*, namely: when G is an oplax $\mathbb{L}$-functor and F is a lax $\mathbb{L}$-functor, then the inserter $\mathit{Ins}(G, F)$ is an $\mathbb{L}$-category.

[14] Meaning: $F : \mathbb{C} \to \mathbb{D}$ is the 1-component of such a lax natural transformation $\mathbb{C} \Rightarrow \mathbb{D}$.

Proposition 3.7 *1. Let $\mathbb{C}$ be an $\mathbb{L}$-category and $F : \mathbb{C} \to \mathbb{C}$ be a lax $\mathbb{L}$-functor. Then $\mathbf{Coalg}_F$ is an $\mathbb{L}$-category; moreover the forgetful functor $\mathbf{Coalg}_F \xrightarrow{U} \mathbb{C}$ is a (strict, non-lax) $\mathbb{L}$-functor.*

2. Given a microcosm model $X \in \mathbb{C}$ for $\mathbb{L}$, the slice category $\mathbb{C}/X$ is an $\mathbb{L}$-category; moreover the functor $\mathbb{C}/X \xrightarrow{\mathrm{dom}} \mathbb{C}$ is an $\mathbb{L}$-functor. □

Note that $\mathbf{Coalg}_F$ being an $\mathbb{L}$-category means not only that operations are interpreted in $\mathbf{Coalg}_F$ but also that all the equational properties specified in $\mathbb{L}$ are satisfied in $\mathbf{Coalg}_F$. Therefore this result generalizes Theorem 2.4.

Concretely, an operation $\mathsf{f} : k \to 1$ in $\mathbb{L}$ is interpreted in $\mathbf{Coalg}_F$ and $\mathbb{C}/X$ as follows, respectively.

$$\left(\begin{array}{c} FX_1 \\ \uparrow c_1 \\ X_1 \end{array}, \dots, \begin{array}{c} FX_k \\ \uparrow c_k \\ X_k \end{array}\right) \mapsto \begin{array}{c} F[\![\mathsf{f}]\!](\overrightarrow{X}) \\ \uparrow (F_\mathsf{f})_{\overrightarrow{X}} \\ [\![\mathsf{f}]\!](\overrightarrow{FX}) \\ \uparrow [\![\mathsf{f}]\!](\overrightarrow{c}) \\ [\![\mathsf{f}]\!](\overrightarrow{X}) \end{array} \qquad \left(\begin{array}{c} Y_1 \\ \downarrow y_1 \\ X \end{array}, \dots, \begin{array}{c} Y_k \\ \downarrow y_k \\ X \end{array}\right) \mapsto \begin{array}{c} [\![\mathsf{f}]\!](\overrightarrow{Y}) \\ \downarrow [\![\mathsf{f}]\!](\overrightarrow{y}) \\ [\![\mathsf{f}]\!](\overrightarrow{X}) \\ \downarrow X_\mathsf{f} \\ X \end{array}$$

Compare these with (5) and (6); these make an essential use of F_f and X_f which generalize sync and $\|$ in Section 2, respectively.

Proposition 3.8 *1. A lax $\mathbb{L}$-functor preserves $\mathbb{L}$-objects. Hence so does an $\mathbb{L}$-functor.*

2. A final object of an $\mathbb{L}$-category $\mathbb{C}$, if it exists, is an $\mathbb{L}$-object. The inner $\mathbb{L}$-structure is induced by finality. □

We can now present our main result. It generalizes Theorem 2.1, hence is a generalized version of the "coalgebraic compositionality" equation (4).

Theorem 3.9 (General compositionality) *Let $\mathbb{C}$ be an $\mathbb{L}$-category and $F : \mathbb{C} \to \mathbb{C}$ be a lax $\mathbb{L}$-functor. Assume further that $\zeta : Z \xrightarrow{\cong} FZ$ is the final coalgebra. Then the functor $\mathsf{beh} : \mathbf{Coalg}_F \to \mathbb{C}/Z$ is a (non-lax) $\mathbb{L}$-functor. It makes the following diagram of $\mathbb{L}$-functors commute.*

$$\begin{tikzcd} \mathbf{Coalg}_F \arrow[rr, "\mathsf{beh}"] \arrow[dr, "U"'] & & \mathbb{C}/Z \arrow[dl, "\mathrm{dom}"] \\ & \mathbb{C} & \end{tikzcd}$$

□

The proof is straightforward by finality. Here $\mathbf{Coalg}_F$ is an $\mathbb{L}$-category (Proposition 3.7.1). So is $\mathbb{C}/Z$ because: $\zeta \in \mathbf{Coalg}_F$ is an $\mathbb{L}$-object (Proposition 3.8.2); $Z = U\zeta$ is an $\mathbb{L}$-object (Propositions 3.8.1 and 3.7.1); hence $\mathbb{C}/Z$ is an $\mathbb{L}$-category (Proposition 3.7.2).

We have also observed some facts which look interesting but are not directly needed for our main result (Theorem 3.9). They include: the category $\mathbb{L}\text{-}\mathbf{obj}_\mathbb{C}$ of $\mathbb{L}$-objects in $\mathbb{C}$ and morphisms between them forms the lax limit of a diagram $\mathbb{C} : \mathbb{L} \to \mathbf{CAT}$; the simplicial category Δ is the "universal" microcosm model for $\mathbf{Mon}$ (cf. [20, Proposition VII.5.1]). The details will appear in the forthcoming extended version.

4 Conclusions and future work

In this paper we have observed that the microcosm principle (as called by Baez and Dolan) brings new mathematical insights into computer science. Specifically, we have looked into parallel composition of coalgebras, which would serve as a mathematical basis for the study of concurrency. As a purely mathematical expedition, we have presented a 2-categorical formalization of the microcosm principle, where an algebraic theory is presented by a Lawvere theory. Turning back to our original motivation, the formalization was applied to coalgebras and yielded some general results which ensure compositionality and equational properties such as associativity.

There are many questions yet to be answered. Some of them have been already mentioned, namely: extending the expressive power of sync (Remark 2.3), and a proper treatment of "pseudo" algebraic structures (Section 3.3).

On the application side, one direction of future work is to establish a relationship between sync and *(syntactic) formats* for process algebras. Our sync represents a certain class of operational rules; formats are a more syntactic way to do the same. Formats which guarantee certain good properties (such as commutativity, see [23]) have been actively studied. Such a format should be obtained by translating e.g. a "commutative" sync into a format.

On the mathematical side, one direction is to identify more instances of the microcosm principle. Mathematics abounds with the (often implicit) idea of nested algebraic structures. To name a few: a topological space in a topos which is itself a "generalized topological space"; a category of domains which itself carries a "structure as a domain." We wish to turn such an informal statement into a mathematically rigorous one, by generalizing the current formalization of the microcosm principle. As a possible first step towards this direction, we are working on formalizing the microcosm principle for finitary monads which are known to be roughly the same thing as Lawvere theories.

Another direction is a search for n-folded nested algebraic structures. In the current paper we have concentrated on two levels of interpretation; an example with more levels might be found e.g. in an internal category in an internal category.

Acknowledgments Thanks are due to Kazuyuki Asada, John Baez, Masahito Hasegawa, Bill Lawvere, Duško Pavlović, John Power and the participants of CALCO-jnr workshop 2007 including Alexander Kurz for helpful discussions and comments.

References

1. J.C. Baez and J. Dolan. Higher dimensional algebra III: n-categories and the algebra of opetopes. *Adv. Math*, 135:145–206, 1998.
2. M. Barr and C. Wells. *Toposes, Triples and Theories*. Springer, Berlin, 1985. Available online.
3. F. Bartels. *On generalised coinduction and probabilistic specification formats. Distributive laws in coalgebraic modelling*. PhD thesis, Free Univ. Amsterdam, 2004.
4. F. Borceux. *Handbook of Categorical Algebra*, vol. 50, 51 and 52 of *Encyclopedia of Mathematics*. Cambridge Univ. Press, 1994.

5. I. Hasuo. Generic forward and backward simulations. In C. Baier and H. Hermanns, editors, *International Conference on Concurrency Theory (CONCUR 2006)*, vol. 4137 of *Lect. Notes Comp. Sci.*, pp. 406–420. Springer, Berlin, 2006.
6. I. Hasuo, B. Jacobs and A. Sokolova. Generic trace semantics via coinduction. *Logical Methods in Comp. Sci.*, 3(4:11), 2007.
7. C.A.R. Hoare. *Communicating Sequential Processes*. Prentice Hall, 1985.
8. M. Hyland and A.J. Power. Discrete Lawvere theories and computational effects. *Theor. Comp. Sci.*, 366(1–2):144–162, 2006.
9. B. Jacobs. *Categorical Logic and Type Theory*. North Holland, Amsterdam, 1999.
10. B. Jacobs. Trace semantics for coalgebras. In J. Adámek and S. Milius, editors, *Coalgebraic Methods in Computer Science*, vol. 106 of *Elect. Notes in Theor. Comp. Sci.* Elsevier, Amsterdam, 2004.
11. B. Jacobs. Introduction to coalgebra. Towards mathematics of states and observations, 2005. Draft of a book, `www.cs.ru.nl/B.Jacobs/PAPERS/index.html`.
12. B. Jacobs. A bialgebraic review of deterministic automata, regular expressions and languages. In K. Futatsugi, J.P. Jouannaud and J. Meseguer, editors, *Essays Dedicated to Joseph A. Goguen*, vol. 4060 of *Lect. Notes Comp. Sci.*, pp. 375–404. Springer, 2006.
13. P.T. Johnstone, A.J. Power, T. Tsujishita, H. Watanabe and J. Worrell. An axiomatics for categories of transition systems as coalgebras. In *Logic in Computer Science*. IEEE, Computer Science Press, 1998.
14. M. Kick, A.J. Power and A. Simpson. Coalgebraic semantics for timed processes. *Inf. & Comp.*, 204(4):588–609, 2006.
15. B. Klin. From bialgebraic semantics to congruence formats. In *Workshop on Structural Operational Semantics (SOS 2004)*, vol. 128 of *Elect. Notes in Theor. Comp. Sci.*, pp. 3–37. 2005.
16. B. Klin. Bialgebraic operational semantics and modal logic. In *Logic in Computer Science*, pp. 336–345. IEEE Computer Society, 2007.
17. A. Kock and G.E. Reyes. Doctrines in categorical logic. In J. Barwise, editor, *Handbook of Mathematical Logic*, pp. 283–313. North-Holland, Amsterdam, 1977.
18. F.W. Lawvere. *Functorial Semantics of Algebraic Theories and Some Algebraic Problems in the Context of Functorial Semantics of Algebraic Theories*. PhD thesis, Columbia University, 1963. Reprints in Theory and Applications of Categories, 5 (2004) 1–121.
19. E.A. Lee. Making concurrency mainstream. Invited talk at CONCUR 2006, 2006.
20. S. Mac Lane. *Categories for the Working Mathematician*. Springer, Berlin, 2nd edn., 1998.
21. R. Milner. *Communication and Concurrency*. Prentice-Hall, 1989.
22. E. Moggi. Notions of computation and monads. *Inf. & Comp.*, 93(1):55–92, 1991.
23. M.R. Mousavi, M.A. Reniers and J.F. Groote. A syntactic commutativity format for SOS. *Inform. Process. Lett.*, 93(5):217–223, 2005.
24. K. Nishizawa and A.J. Power. Lawvere theories enriched over a general base. *Journ. of Pure & Appl. Algebra*, 2006. To appear.
25. J. Power and D. Turi. A coalgebraic foundation for linear time semantics. In *Category Theory and Computer Science*, vol. 29 of *Elect. Notes in Theor. Comp. Sci.* Elsevier, 1999.
26. J.J.M.M. Rutten. Universal coalgebra: a theory of systems. *Theor. Comp. Sci.*, 249:3–80, 2000.
27. D. Turi and G. Plotkin. Towards a mathematical operational semantics. In *Logic in Computer Science*, pp. 280–291. IEEE, Computer Science Press, 1997.