Logical Methods in Computer Science
Vol. 11(1:3)2015, pp. 1–40
www.lmcs-online.org

Submitted May 4, 2011
Published Mar. 6, 2015

MONADS NEED NOT BE ENDOFUNCTORS *

THORSTEN ALTENKIRCH [a], JAMES CHAPMAN [b], AND TARMO UUSTALU [c]

[a] School of Computer Science, University of Nottingham, Jubilee Campus, Wollaton Road, Nottingham NG8 1BB, United Kingdom
e-mail address: txa@cs.nott.ac.uk

[b,c] Institute of Cybernetics, Tallinn University of Technology, Akadeemia tee 21, 12618 Tallinn, Estonia
e-mail address: {james,tarmo}@cs.ioc.ee

ABSTRACT. We introduce a generalization of monads, called relative monads, allowing for underlying functors between different categories. Examples include finite-dimensional vector spaces, untyped and typed λ-calculus syntax and indexed containers. We show that the Kleisli and Eilenberg-Moore constructions carry over to relative monads and are related to relative adjunctions. Under reasonable assumptions, relative monads are monoids in the functor category concerned and extend to monads, giving rise to a coreflection between relative monads and monads. Arrows are also an instance of relative monads.

1. Introduction

Monads are the most successful programming pattern arising in functional programming. Apart from their use to model a generic notion of effect they also serve as a convenient interface to generalized notions of substitution. Research in the area on the border between category theory and functional programming focusses on unveiling new programming and reasoning constructions similar to monads, such as comonads [36], arrows [20] and idioms (closed functors) [25]. Indeed, especially when working in an expressive and total language with dependent types, such as Agda [27], we can exploit monads as a way to structure not only our programs but also their verification.

The present paper is concerned with a generalization of monads which arises naturally in dependently typed programming, namely monad-like entities that are not endofunctors.

2012 ACM CCS: [**Software and its engineering**]: Software notations and tools—General programming languages—Language features—Data types and structures; [**Theory of computation**]: Semantics and reasoning—Program semantics—Categorical semantics.

Key words and phrases: monads, adjunctions, monoids, skew-monoidal categories, Hughes's arrows.

* This article is a revised and expanded version of the FoSSaCS 2010 conference paper [5].

[a] T. Altenkirch was supported by the Engineering and Physical Sciences Research Council (EPSRC) grant no. EP/G034109/1.

[b,c] J. Chapman and T. Uustalu were supported by the ERDF funded Estonian CoE project EXCS, the Estonian Ministry of Education and Research target-financed themes no. 0322709s06 and 0140007s12, and the Estonian Science Foundation grants no. 6940, 9219 and 9475.

DOI:10.2168/LMCS-11(1:3)2015

Consider the following example, which arose when implementing notions related to quantum programming, namely finite-dimensional vector spaces [37, 7]. (See also Piponi [29] for this and other interesting uses of vector spaces in functional programming.)

Example 1.1. In quantum computing, we consider complex vector spaces, but for the present development any semiring $(R, 0, +, 1, \times)$ is sufficient. Finite-dimensional vector spaces (more precisely, right modules) with a given basis can be given by:

$$\begin{array}{l}
\mathsf{Vec} \in |\mathbf{Fin}| \to |\mathbf{Set}| \\
\mathsf{Vec}\, m =_{\mathrm{df}} J_{\mathrm{f}}\, m \to R \\
\eta \in \Pi_{m \in |\mathbf{Fin}|} J_{\mathrm{f}}\, m \to \mathsf{Vec}\, m \\
\eta_m\, (i \in \underline{m}) =_{\mathrm{df}} \lambda j \in \underline{m}.\ \mathrm{if}\ i = j\ \mathrm{then}\ 1\ \mathrm{else}\ 0 \\
(-)^* \in \Pi_{m,n \in |\mathbf{Fin}|} (J_{\mathrm{f}}\, m \to \mathsf{Vec}\, n) \to (\mathsf{Vec}\, m \to \mathsf{Vec}\, n) \\
A^*\, x =_{\mathrm{df}} \lambda j \in \underline{n}.\ \sum_{i \in \underline{m}} A\, i\, j \times x\, i
\end{array}$$

Here **Fin** is the category of finite cardinals (the skeletal version of finite sets). The objects are natural numbers $m \in \mathbb{N}$ and the maps between m and n are functions between $\underline{m}$ and $\underline{n}$ where $\underline{m} =_{\mathrm{df}} \{0, 1, \ldots, m-1\}$. By $J_{\mathrm{f}} \in \mathbf{Fin} \to \mathbf{Set}$ we mean the natural embedding $J_{\mathrm{f}}\ m =_{\mathrm{df}} \underline{m}$. The finite summation $\sum$ is just the finite iteration of $+$ over 0. Indeed η_m is just the unit $m \times m$-matrix (alternatively, a function assigning to every dimension $i \in \underline{m}$ the corresponding basis vector) and $A^*\, x$ corresponds to the product of the matrix A with the vector x, where both matrices and vectors are described as functions.

By the types of its data, the structure $(\mathsf{Vec}, \eta, (-)^*)$ looks suspiciously like a monad, except that **Fin** is not **Set** and in the types for η and $(-)^*$ we have used the embedding J_{f} to repair the mismatch. It is easy to verify that the structure also satisfies the standard monad laws, modulo the same discrepancy.

The category of finite-dimensional vector spaces with a given basis (coordinate spaces) arises as a kind of Kleisli category. Its objects are $m \in \mathbb{N}$, understood as finite sets of dimensions and its morphisms are functions $J_{\mathrm{f}}\, m \to \mathsf{Vec}\, n$, i.e., matrices (describing linear transformations).

The structure cannot generally be pushed to a monad on **Set**. $(-)^*$ requires that we can sum over a set. Summation over general index sets is not available, if R is just a semiring. Also, in a constructive setting, η requires that the set has a decidable equality, which is not the case for general sets.

Since we only require a semiring, the restrictions of the multiset and powerset functors to **Fin** are instances of this construction by using $(\mathbb{N}, 0, +, 1, \times)$ and $(\mathbb{B}, \bot, \vee, \top, \wedge)$ respectively.

We shall view Vec as a *relative monad* on the embedding $J_{\mathrm{f}} \in \mathbf{Fin} \to \mathbf{Set}$. Other examples of relative monads include untyped and simply typed λ-terms, the notions of indexed functors and indexed containers as developed in [26], and arrows.

Overview of the paper. In Section 2 we develop the notion of relative monads on a functor $J \in \mathbb{J} \to \mathbb{C}$, showing that they arise from relative adjunctions, and generalize the Kleisli and Eilenberg-Moore constructions to relative monads.

Since monads on $\mathbb{C}$ correspond to monoids in the endofunctor category $[\mathbb{C}, \mathbb{C}]$, a natural question is whether a relative monad on J gives rise to a monoid in the category $[\mathbb{J}, \mathbb{C}]$. If $\mathbb{J}$ is small and $\mathbb{C}$ is cocomplete (e.g., **Set**), the left Kan extension along J exists and gives rise to a left skew-monoidal structure where the unit is J and the tensor is given by $F \cdot^J G =_{\mathrm{df}} \mathrm{Lan}_J\, F \cdot G$.[1] Relative monads give rise to skew-monoids in this setting (Section 3).

Going further, we identify conditions on the functor $J \in \mathbb{J} \to \mathbb{C}$, under which the skew-monoidal structure induced by Lan_J is properly monoidal. Under these well-behavedness conditions, relative monads on J are proper monoids in $[\mathbb{J}, \mathbb{C}]$. Moreover, relative monads extend to monads via Lan_J and we get a coreflection between the categories of relative monads on the functor J and monads on the category $\mathbb{C}$ (Section 4). In the example of vector spaces, $\mathrm{Lan}_J\,\mathsf{Vec}$ is the monad whose Kleisli category is that of vector spaces over general sets of dimensions where a vector over an infinite set of dimensions may only have finitely many non-zero coordinates. However, it is worthwhile not to ignore the non-endofunctor case, because frequently this is the structure we actually want to use. E.g., in quantum computing we are interested in dagger compact closed categories [2] which model finite-dimensional vector spaces.

Finally, we show that arrows are relative monads (Section 5) on the Yoneda embedding. This leads to the, maybe surprising, outcome that while arrows generalize ordinary monads, they are actually a special case of relative monads.

What is new? This paper completes the conference paper [5] with proofs, but also adds new material. Throughout the technical part of the paper (spanning Sections 2—5), we systematically speak of not just relative monads but also relative monad morphisms, i.e., relative monads as a category, so that restriction of monads and extension of relative monads become functors. In Sections 4, 5 we also accordingly treat monoid morphisms and arrow morphisms. Discussing examples in Sections 2.1 and 2.2, we go into more depth than in the conference paper (in particular, we look at the EM-algebras of several examples of relative monads), but we also consider some additional examples. In the new Sections 2.4 and 4.4, we analyze the relationship of the Kleisli and Eilenberg-Moore constructions of a monad and its restriction and a relative monad and its extension. In the Section 3.4, we present an alternative definition of EM-algebras that is available as soon as Lan_J exists.

Related work. The untyped λ-calculus syntax as has been identified as a monoid in $[\mathbf{Fin}, \mathbf{Set}]$ by Fiore et al. [14]. Heunen and Jacobs [18] have shown that arrows on $\mathbb{C}$ are actually monoids in the category $[\mathbb{C}^{\mathrm{op}} \times \mathbb{C}, \mathbf{Set}]$ of endoprofunctors; Jacobs et al. have proved the Freyd construction of [30] is, in a good sense, the Kleisli construction for arrows. Spivey [31] has studied a generalization of monads, which differs from ours, but is similar in spirit and related (see Conclusion). Berger et al. [10, 16] have introduced a generalization of finitary monads, called monads with arities. Monads with arities constitute a special case of relative monads on well-behaved functors.

Monoidal-like categories where the unital and associativity laws are not isomorphisms have been considered by multiple authors. Skew-monoidal categories were introduced by Szlachányi [33] and caught then the interest of by Lack, Street, Buckley and Garner [22, 23, 12]. The axioms of skew-monoidality were also part of the axioms of the categories of contexts of Blute et al. [11].

[1] In the conference version we spoke of a 'lax monoidal' category, but then Szlachányi [33] discovered the same structure and termed it 'left skew-monoidal'.

Notation. We will be using a mixture of categorical and type-theoretic notation. In particular we will be using λ-calculus notation for defining functions (maps in **Set** or subcategories). Customarily for both category theory and type theory, we often hide some arguments of patterns and function applications (normally subscripted arguments, e.g., an object a natural transformation is applied to). We write _ for unique values that are easily inferable, e.g., the function on the empty domain.

We write $|\mathbb{C}|$ for the set of objects of $\mathbb{C}$ and $\mathbb{C}\,(X, Y)$ for the homsets. Given categories $\mathbb{C}$,$\mathbb{D}$ we write the functor category as $[\mathbb{C}, \mathbb{D}]$, we will also write $\mathbb{C} \to \mathbb{D}$ for the set of functors, i.e. the objects of the functor category $[\mathbb{C}, \mathbb{D}]$. We write id, $\circ$ for the identities and composition of maps and I, $\cdot$ for the identities and composition of functors.

2. Relative monads and relative adjunctions

We start by defining relative monads. Then we give some examples and show how the theory of ordinary monads carries over to the relative case.

2.1. Relative monads.

Rather than being defined for a category $\mathbb{C}$ like a monad, a relative monad is defined for a functor J between two categories $\mathbb{J}$ and $\mathbb{C}$.

Definition 2.1. A (Manes-style [24]) *relative monad* on a functor $J \in \mathbb{J} \to \mathbb{C}$ is given by
- an object mapping $T \in |\mathbb{J}| \to |\mathbb{C}|$,
- for any $X \in |\mathbb{J}|$, a map $\eta_X \in \mathbb{C}\,(J\,X, T\,X)$ (the *unit*),
- for any $X, Y \in |\mathbb{J}|$ and $k \in \mathbb{C}\,(J\,X, T\,Y)$, a map $k^* \in \mathbb{C}\,(T\,X, T\,Y)$ (the *Kleisli extension*)

satisfying the conditions
- for any $X, Y \in |\mathbb{J}|$, $k \in \mathbb{C}\,(J\,X, T\,Y)$, $k = k^* \circ \eta$ (the right unital law),
- for any $X \in |\mathbb{J}|$, $\eta_X^* = \mathrm{id}_{TX} \in \mathbb{C}\,(T\,X, T\,X)$ (the left unital law),
- for any $X, Y, Z \in |\mathbb{J}|$, $k \in \mathbb{C}\,(J\,X, T\,Y)$, $\ell \in \mathbb{C}\,(J\,Y, T\,Z)$, $(\ell^* \circ k)^* = \ell^* \circ k^*$ (the associativity law).

The data and laws of a relative monad are exactly as those of a monad, except that $\mathbb{C}$ has become $\mathbb{J}$ in some places and, to ensure type-compatibility, some occurrences of J have been inserted. Indeed, in the laws it is only the types that have changed.

The laws imply that T is functorial: $T \in \mathbb{J} \to \mathbb{C}$. Indeed, for $X, Y \in |\mathbb{J}|$, $f \in \mathbb{J}\,(X, Y)$, we can define a map $T\,f \in \mathbb{C}\,(TX, TY)$ by $T\;f =_{\mathrm{df}} (\eta \circ J f)^*$ and this satisfies the functor laws. Also, η and $(-)^*$ are natural.

A definition of relative monads based on a multiplication μ rather than a Kleisli extension $(-)^*$ is not immediately available: the simple functor composition $T \cdot T$ is not well-typed. In the next section, we will show that a suitable notion of functor composition is available under a condition.

Definition 2.2. A *relative monad morphism* between two relative monads $(T, \eta, (-)^*)$ and $(T', \eta', (-)^{*'})$ for a particular J is given by
- for any $X \in |\mathbb{J}|$ a map $\sigma_X \in \mathbb{C}\,(T\,X, T'\,X)$

satisfying the conditions
- for any $X \in |\mathbb{J}|$, $\sigma_X \circ \eta_X = \eta'_X$ (the unit preservation law),
- for any $X, Y \in |\mathbb{J}|, k \in \mathbb{C}\,(J\,X, T\,Y)$, $\sigma_Y \circ k^* = (\sigma_Y \circ k)^{*'} \circ \sigma_X$ (the multiplication preservation law).

The two conditions entail that σ is natural.

It is easy to see that relative monads on a particular J and morphisms between them form a category, which we denote by $\mathbf{RMon}(J)$. The identities and composition of this category are inherited from the functor category $[\mathbb{J}, \mathbb{C}]$.

Clearly, monads on $\mathbb{C}$ and monad morphisms between them are a special case of relative monads and their morphisms via $\mathbb{J} =_{\mathrm{df}} \mathbb{C}$, $J =_{\mathrm{df}} I_{\mathbb{C}}$.

For general $\mathbb{J}$, $\mathbb{C}$ and J, we always have that $T\,X =_{\mathrm{df}} J\,X$ is a relative monad with $\eta_X =_{\mathrm{df}} \mathrm{id}_{J\,X}$ and $k^* =_{\mathrm{df}} k$. In fact, a whole class of examples of relative monads on J is given by restricting monads on $\mathbb{C}$ (the relative monad J arising from restricting the monad $I_{\mathbb{C}}$).

Proposition 2.3. *Given a functor $J \in \mathbb{J} \to \mathbb{C}$.*

(1) *A monad $(T, \eta, (-)^*)$ on $\mathbb{C}$ restricts to a relative monad $(T^\flat, \eta^\flat, (-)^{(*^\flat)})$ on J defined by $T^\flat\,X =_{\mathrm{df}} T\,(J\,X)$, $\eta^\flat_X =_{\mathrm{df}} \eta_{J\,X}$, $k^{(*^\flat)} =_{\mathrm{df}} k^*$.*
(2) *A monad morphism σ between two monads T, T' on $\mathbb{C}$ restricts to a relative monad morphism $\sigma^\flat$ between the relative monads $T^\flat$, $T'^\flat$ on J defined by $\sigma^\flat_X =_{\mathrm{df}} \sigma_{J\,X}$.*
(3) *$(-)^\flat$ is a functor from $\mathbf{Mon}(\mathbb{C})$ to $\mathbf{RMon}(J)$.*

The three relative monad laws and two relative monad morphism laws follow immediately from the corresponding laws of monads and monad morphisms.

Later we will show that, under some reasonable conditions on J, it is also possible to extend relative monads to monads by a functor $(-)^\sharp : \mathbf{RMon}(J) \to \mathbf{Mon}(\mathbb{C})$. This functor is right adjoint to $(-)^\flat$; the adjunction is a coreflection.

As a first truly non-trivial example, we saw the relative monad of finite-dimensional vector spaces in the introduction. Here are some further examples.

Example 2.4. The syntax of untyped (but well-scoped) λ-calculus is a relative monad on $J_{\mathrm{f}} \in \mathbf{Fin} \to \mathbf{Set}$, as is the finite-dimensional vector spaces relative monad, i.e., we have $\mathbb{J} =_{\mathrm{df}} \mathbf{Fin}$, $\mathbb{C} =_{\mathrm{df}} \mathbf{Set}$, $J =_{\mathrm{df}} J_{\mathrm{f}}$. We view $\mathbf{Fin}$ as the category of nameless untyped contexts. The set of untyped λ-terms $\mathsf{Lam}\,\Gamma$ over a context Γ satisfies the isomorphism

$$\mathsf{Lam}\,\Gamma \cong J_{\mathrm{f}}\,\Gamma + \mathsf{Lam}\,\Gamma \times \mathsf{Lam}\,\Gamma + \mathsf{Lam}\,(1 + \Gamma)$$

The summands correspond to variables from the context (seen as terms), applications, and abstractions (their bodies are terms over an extended context). The functor $\mathsf{Lam} \in \mathbf{Fin} \to \mathbf{Set}$ is defined as the carrier of the initial algebra of the functor $F \in [\mathbf{Fin}, \mathbf{Set}] \to [\mathbf{Fin}, \mathbf{Set}]$ defined by

$$F\,G\,\Gamma =_{\mathrm{df}} J_{\mathrm{f}}\,\Gamma + G\,\Gamma \times G\,\Gamma + G\,(1 + \Gamma)$$

Lam is a relative monad. The unit $\eta \in J_{\mathrm{f}}\,\Gamma \to \mathsf{Lam}\,\Gamma$ is given by variables-as-terms and the Kleisli extension takes a finite substitution rule $k \in J_{\mathrm{f}}\,\Gamma \to \mathsf{Lam}\,\Delta$ to the corresponding substitution function $k^* \in \mathsf{Lam}\,\Gamma \to \mathsf{Lam}\,\Delta$.

We also introduce the relative monads Lam^β and $\mathsf{Lam}^{\beta\eta}$ over J_{f} by quotienting over β-equality (resp. $\beta\eta$-equality). We observe that the monad operations preserve the equalities, since β- and $\beta\eta$-equality are stable under substitution.

This example was described as a relative monad (under the name Kleisli structure) by Altenkirch and Reus [8]. Fiore et al. [14] described it as a monoid in a monoidal structure on $[\mathbf{Fin}, \mathbf{Set}]$. Their account of this example is an instance of our general description of relative monads as monoids from Section 4.

Example 2.5. Typed λ-terms form a relative monad in a similar fashion. Let Ty be the set of types (over some base types), which we see as a discrete category. We take $\mathbb{J}$ to be $\mathbf{Fin} \downarrow \mathsf{Ty}$, which is the category whose objects are pairs (Γ, ρ) where $\Gamma \in |\mathbf{Fin}|$ and $\rho \in \Gamma \to \mathsf{Ty}$ (typed contexts) and maps from (Γ, ρ) to (Γ', ρ') are maps $f \in \mathbf{Fin}\,(\Gamma, \Gamma')$ such that $\rho = \rho' \circ f$ (typed context maps).

We further take $\mathbb{C}$ to be the functor category $[\mathsf{Ty}, \mathbf{Set}]$ and let $J_{\mathrm{t}} \in \mathbf{Fin} \downarrow \mathsf{Ty} \to [\mathsf{Ty}, \mathbf{Set}]$ be the natural embedding defined by $J_{\mathrm{t}}\,(\Gamma, \rho)\,\sigma =_{\mathrm{df}} \{x \in \Gamma \mid \rho\, x = \sigma\}$.

Now, for $(\Gamma, \rho) \in |\mathbf{Fin} \downarrow \mathsf{Ty}|$ and $\sigma \in \mathsf{Ty}$, the set of typed λ-terms $\mathsf{TyLam}\,(\Gamma, \rho)\,\sigma$ has to satisfy the isomorphism

$$\begin{array}{rcl} \mathsf{TyLam}\,(\Gamma,\rho)\,\sigma & \cong & J_{\mathrm{t}}\,(\Gamma,\rho)\,\sigma \\ & + & \Sigma_{\tau \in \mathsf{Ty}} \mathsf{TyLam}\,(\Gamma,\rho)\,(\tau \Rightarrow \sigma) \times \mathsf{TyLam}\,(\Gamma,\rho)\,\tau \\ & + & \text{if } \sigma \text{ is of the form } \tau \Rightarrow \tau' \text{ then } \mathsf{TyLam}\,(1+\Gamma, \left[\begin{array}{ccc} \mathrm{inl}\,* & \mapsto & \tau \\ \mathrm{inr}\,x & \mapsto & \rho\,x \end{array}\right])\,\tau' \end{array}$$

Assuming Id-types the last summand can be more concisely written as:

$$+ \quad \Sigma_{\tau,\tau' \in \mathsf{Ty}, p \in \mathsf{Id}\,\sigma\,(\tau \Rightarrow \tau')} \mathsf{TyLam}\,(1+\Gamma, \left[\begin{array}{ccc} \mathrm{inl}\,* & \mapsto & \tau \\ \mathrm{inr}\,x & \mapsto & \rho\,x \end{array}\right])\,\tau'$$

The functor $\mathsf{TyLam} \in \mathbf{Fin} \downarrow \mathsf{Ty} \to [\mathsf{Ty}, \mathbf{Set}]$ is given by an initial algebra. It is a monad on J_{t}, with the unit and Kleisli extension given by variables-as-terms and substitution, like in the case of Lam. Fiore et al. [13] studied TyLam as a monoid in $[\mathbf{Fin} \downarrow \mathsf{Ty}, [\mathsf{Ty}, \mathbf{Set}]]$. As in Example 2.4 we can quotient by β- or $\beta\eta$-equality and as before we denote the corresponding relative monads as TyLam^{β} and $\mathsf{TyLam}^{\beta\eta}$.

Note that choosing $\mathbb{J}$ to be $[\mathsf{Ty}, \mathbf{Fin}]$ rather than $\mathbf{Fin} \downarrow \mathsf{Ty}$ would have given contexts possibly supported by infinitely many types: in every type there are finitely many variables, but the total number of variables can be infinite.

Example 2.6. Morris and Altenkirch [26] investigated a generalization of the notion of containers [1] to a dependently typed setting and used it to show that strictly positive families can be reduced to W-types. Relative monads played a central role in this development.

Let $\mathsf{U} \in |\mathbf{Set}|$ together with a family $\mathsf{El} \in \mathsf{U} \to |\mathbf{Set}|$ which we view as a type theoretic universe. As an example consider the universe of small sets which reflects all type theoretic constructions but U itself. E.g., there is $\pi \in \Pi A \in \mathsf{U}.\,(\mathsf{El}\,A \to \mathsf{U}) \to \mathsf{U}$ such that $\mathsf{El}\,(\pi\,A\,B)$ is isomorphic to $\Pi a \in \mathsf{El}\,A.\,\mathsf{El}\,(B\,a)$. And similarly for the other type formers.

Such a universe induces a category $\mathbf{U}$ with $|\mathbf{U}| =_{\mathrm{df}} \mathsf{U}$ and $\mathbf{U}\,(A, B) =_{\mathrm{df}} \mathsf{El}\,A \to \mathsf{El}\,B$. The functor $J_{\mathbf{U}} \in \mathbf{U} \to \mathbf{Set}$ is given by $J_{\mathbf{U}}\,A =_{\mathrm{df}} \mathsf{El}\,A$ on objects and the identity on maps.[2] We assume that $\mathbf{U}$ is locally cartesian closed, which corresponds to the assumption that U reflects Π, Σ and equality types.

While ordinary containers represent endofunctors on $\mathbf{U}$, indexed containers represent functors from the category of families $\mathbf{Fam}\,A$ to $\mathbf{U}$ for $A \in \mathsf{U}$. $\mathbf{Fam}\,A$ has as objects families $\mathsf{El}\,A \to \mathsf{U}$ and as morphisms between $F, G \in \mathsf{El}\,A \to \mathsf{U}$ families of functions $\Pi a \in \mathsf{El}\,A.\,\mathsf{El}\,(F\,a) \to \mathsf{El}\,(G\,a)$. Indeed, $\mathbf{Fam}\,A$ is equivalent to the slice category $\mathbf{U}/A$. For $A \in \mathsf{U}$, we define the set of indexed functors by $\mathsf{IF}\,A =_{\mathrm{df}} \mathbf{Fam}\,A \to \mathbf{U}$ and indeed IF gives rise to a relative monad on $J_{\mathbf{U}}$: The unit $\eta_A \in J_{\mathbf{U}}\,A \to \mathsf{IF}\,A$ is defined by $\eta_A\,x =_{\mathrm{df}} \lambda f.\,f\,x$ and the Kleisli extension $k^* \in \mathsf{IF}\,A \to \mathsf{IF}\,B$ of $k \in J_{\mathbf{U}}\,A \to \mathsf{IF}\,B$ is defined by $k^*\,G\,f =_{\mathrm{df}} G\,(\lambda x.k\,x\,f)$. The definitions clearly resemble the continuation monad apart from the size issue.

The main result of [26] was that strictly positive families (SPF) can be interpreted as indexed functors via indexed containers (IC). Just as IF, both SPF and IC are relative

[2] In [1] we actually used $\mathbf{Set}$ and $\mathbf{Set}_1$ instead of U and $\mathbf{Set}$, and El is usually implicit in type theory.

monads on $J_{\mathbf{U}}$ and the interpretations preserve this structure, i.e., are relative monad maps. The relative monads model the fact that all these notions are closed under substitution and that this is preserved by the constructions done in the paper.

Example 2.7. Sam Staton suggested to us this example of a "relative monad" that naturally arises at a different level—a relative pseudo-monad, in fact. Let $\mathbb{J} =_{\mathrm{df}} \mathbf{Cat}$ (the category of small categories), $\mathbb{C} =_{\mathrm{df}} \mathbf{CAT}$ (the category of locally small categories) and let $J \in \mathbf{Cat} \to \mathbf{CAT}$ be the inclusion.

Define $T \in \mathbf{Cat} \to \mathbf{CAT}$ by $T\,\mathbb{X} =_{\mathrm{df}} [\mathbb{X}^{\mathrm{op}}, \mathbf{Set}]$, i.e., T sends a given small category to the corresponding presheaf category, which is locally small.

We can define $\eta_{\mathbb{X}} \in \mathbb{X} \to [\mathbb{X}^{\mathrm{op}}, \mathbf{Set}]$ to be $\mathbf{Y}_{\mathbb{X}}$ (the Yoneda embedding for $\mathbb{X}$). And, for $K \in \mathbb{X} \to [\mathbb{Y}^{\mathrm{op}}, \mathbf{Set}]$, we can set $K^* \in [\mathbb{X}^{\mathrm{op}}, \mathbf{Set}] \to [\mathbb{Y}^{\mathrm{op}}, \mathbf{Set}]$ to be $\mathrm{Lan}_{\mathbf{Y}_{\mathbb{X}}} K$.

Note that the laws do not hold on the nose now, but only up to coherent isomorphism.

Example 2.8. We already know that, for any monad T on $\mathbb{C}$, $T^\flat = T \cdot J \in \mathbb{J} \to \mathbb{C}$ is a relative monad on J. Interesting special cases of this basic observation arise already when T is the identity functor on $\mathbb{C}$. E.g., we can take $\mathbb{C} =_{\mathrm{df}} \mathbf{Set}$, $\mathbb{J} =_{\mathrm{df}} \mathbf{Set}$, $J\,X =_{\mathrm{df}} X \times S$, $T^\flat\,X =_{\mathrm{df}} J\,X$ for some fixed $S \in |\mathbf{Set}|$. Or we can take $\mathbb{C} =_{\mathrm{df}} \mathbf{Set}$, $\mathbb{J} =_{\mathrm{df}} \mathbf{Set}^{\mathrm{op}}$, $JX =_{\mathrm{df}} R^X$ for some fixed $R \in |\mathbf{Set}|$. As we will see below, these constructions behave in some aspects like the state and continuations monads.

2.2. **Relative adjunctions.** As ordinary monads are intimately related to adjunctions, relative monads are related to a corresponding generalization of adjunctions. Similarly to the situation with relative monads, not every definition format of adjunctions is available for relative adjunctions, if we make no assumptions about J: definitions involving a counit are not possible. The following is one of the possible definitions.

Definition 2.9. A *relative adjunction* between $J \in \mathbb{J} \to \mathbb{C}$ and $\mathbb{D}$ is given by two functors $L \in \mathbb{J} \to \mathbb{D}$ and $R \in \mathbb{D} \to \mathbb{C}$ and a natural isomorphism $\phi_{X,Y} \in \mathbb{C}\,(J\,X, R\,Y) \cong \mathbb{D}\,(L\,X, Y)$.

As expected, ordinary adjunctions are a special case of relative adjunctions with $\mathbb{J} =_{\mathrm{df}} \mathbb{C}$, $J =_{\mathrm{df}} I$. Just like any adjunction defines a monad, relative adjunctions define relative monads.

Theorem 2.10. *Any relative adjunction (L, R, ϕ) between a functor $J \in \mathbb{J} \to \mathbb{C}$ and category $\mathbb{D}$ gives rise to a relative monad $(T, \eta, (-)^*)$ via $T\,X =_{\mathrm{df}} R\,(L\,X)$, $\eta_X =_{\mathrm{df}} \phi^{-1}\,(\mathrm{id}_{L\,X})$ and $k^* =_{\mathrm{df}} R\,(\phi\,k)$.*

$$
\begin{tikzcd}
 & \mathbb{D} \arrow[dr, "R"] & \\
\mathbb{J} \arrow[ur, "L"] \arrow[rr, "J"] \arrow[rr, bend left=80, "T"] & & \mathbb{C}
\end{tikzcd}
$$

Proof. We have to check the relative monad laws. The right unital law holds since $k = \phi^{-1}\,(\phi\,k) = \phi^{-1}\,(\phi\,k \circ \mathrm{id}_{L\,X}) = R\,(\phi\,k) \circ \phi^{-1}\mathrm{id}_{L\,X} = k^* \circ \eta_X$ by ϕ^{-1} being a left inverse of ϕ and ϕ^{-1} being natural.

The left unital law is verified by $(\eta_X)^* = R\,(\phi\,(\phi^{-1}\,\mathrm{id}_{L\,X})) = R\,\mathrm{id}_{L\,X} = \mathrm{id}_{R\,(L\,X)} = \mathrm{id}_{T\,X}$ by ϕ^{-1} being a right inverse of ϕ .

For the associative law we calculate $(\ell^* \circ k)^* = R\,(\phi\,(R(\phi\,\ell) \circ k)) = R\,(\phi\,\ell \circ \phi\,k) = R(\phi\,\ell) \circ R(\phi\,k) = \ell^* \circ k^*$ by ϕ being natural. $\square$

If a relative monad T on J is related to a relative adjunction (L, R, ϕ) between J and some category $\mathbb{D}$ in the above way, we call the relative adjunction a *splitting* of the relative monad via $\mathbb{D}$.

2.3. **Kleisli and Eilenberg-Moore constructions.** We know that a monad splits into an adjunction in two canonical ways: the Kleisli and Eilenberg-Moore constructions. Moreover, the splittings form a category where the Kleisli and EM splittings are the initial and terminal objects. We shall now establish that the same holds in the relative situation.

The Kleisli category $\mathbf{Kl}(T)$ of a relative monad T has as objects the objects of $\mathbb{J}$ and as maps between objects X and Y of $\mathbb{J}$ the maps between objects $J\,X, T\,Y$ of $\mathbb{C}$: $|\mathbf{Kl}(T)| =_{\mathrm{df}} |\mathbb{J}|$ and $\mathbf{Kl}(T)\,(X, Y) =_{\mathrm{df}} \mathbb{C}\,(J\,X, T\,Y)$. The identity and composition (we denote them by id^T, $\circ^T$) are defined by $\mathrm{id}^T_X =_{\mathrm{df}} \eta_X$ and $\ell \circ^T k =_{\mathrm{df}} \ell^* \circ k$.

The Kleisli relative adjunction between $\mathbb{J}$ and $\mathbf{Kl}(T)$ is defined by $L\,X =_{\mathrm{df}} X$, $L\,f =_{\mathrm{df}} \eta \circ J f$ (note that L is identity-on-objects), $R\,X =_{\mathrm{df}} T\,X$, $R\,k =_{\mathrm{df}} k^*$ and ϕ is identity. This relative adjunction is a splitting. Indeed, we have $R\,(L\,X) = T\,X$, $R\,(L\,f) = (\eta \circ J\,f)^* = T\,f$, $\eta_X = \mathrm{id}^T_X = \phi^{-1}\,(\mathrm{id}^T_X) = \phi^{-1}\,(\mathrm{id}^T_{L\,X})$ and $k^* = R\,k = R\,(\phi\,k)$.

The Eilenberg-Moore (EM) category $\mathbf{EM}(T)$ is given by EM-algebras and EM-algebra maps of the relative monad T. Since the usual definition of an EM-algebra refers to μ, which is not immediately available, we generalize a version based on $(-)^*$. For ordinary monads this is equivalent to the standard definition.

Definition 2.11. An EM-*algebra* of a relative monad T on $J \in \mathbb{J} \to \mathbb{C}$ is given by an object $X \in |\mathbb{C}|$ (the *carrier*) and, for any $Z \in |\mathbb{J}|$ and $f \in \mathbb{C}\,(J\,Z, X)$, a map $\chi\,f \in \mathbb{C}\,(T\,Z, X)$ (the *structure*), satisfying the conditions

- for any $Z \in |\mathbb{J}|$, $f \in \mathbb{C}\,(J\,Z, X)$, $f = \chi\,f \circ \eta$,
- for any $Z, W \in |\mathbb{J}|$, $k \in \mathbb{C}\,(J\,Z, T\,W)$, $f \in \mathbb{C}\,(J\,W, X)$, $\chi\,(\chi\,f \circ k) = \chi\,f \circ k^*$.

These conditions ensure, among other things, that χ is natural.

An EM-*algebra map* from (X, χ) to (Y, υ) is a map $h \in \mathbb{C}\,(X, Y)$ satisfying

- for any $Z \in |\mathbb{J}|$, $f \in \mathbb{C}\,(J\,Z, X)$, $h \circ \chi\,f = \upsilon\,(h \circ f)$.

The identity and composition maps of $\mathbf{EM}(T)$ are inherited from $\mathbb{C}$.

The Eilenberg-Moore relative adjunction between J and $\mathbf{EM}(T)$ is defined by $L\,X =_{\mathrm{df}} (T\,X, \lambda k.\,k^*)$, $L\,f =_{\mathrm{df}} T\,f$, $R\,(X, \chi) =_{\mathrm{df}} X$, $R\,h =_{\mathrm{df}} h$, $\phi_{X,(Y,\upsilon)}\,f =_{\mathrm{df}} \upsilon\,f$ and $\phi^{-1}_{X,(Y,\upsilon)}\,h =_{\mathrm{df}} h \circ \eta_X$. This is also a splitting.

Theorem 2.12. *The splittings of a relative monad T on $J \in \mathbb{J} \to \mathbb{C}$ form a category. An object is given by a category $\mathbb{D}$ and an adjunction (L, R, ϕ) splitting T via $\mathbb{D}$. A splitting morphism between $(\mathbb{D}, L, R, \phi)$ and $(\mathbb{D}', L', R', \phi')$ is a functor $V \in \mathbb{D} \to \mathbb{D}'$ such that $V \cdot L = L'$, $R = R' \cdot V$, and $V\,(\phi_{X,Y}\,k) = \phi'_{X,V\,Y}\,k$. The Kleisli construction is the initial and the Eilenberg-Moore construction the terminal splitting.*

Proof. To show that the Kleisli splitting is initial we show that the following is a unique morphism from the Kleisli splitting $(\mathbf{Kl}(T), L_T, R_T, \phi_T)$ to a given other splitting $(\mathbb{D}, L, R, \phi)$. We define:

$$\begin{array}{lcl} V & \in & \mathbf{Kl}(T) \to \mathbb{D} \\ V\,X & = & L\,X \\ V\,k & = & \phi_{X,LY}\,k \end{array}$$

The functoriality of V is verified by $V\,\eta = \phi\,\eta = \mathrm{id}$, $V\,(\ell^* \circ k) = \phi\,(R\,(\phi\,\ell) \circ k) = \phi\,\ell \circ \phi\,k$.

The splitting morphism conditions are verified by $V\,(L_T\,X) = L\,X$, $V\,(L_T\,f) = \phi\,(\eta \circ J\,f) = \phi\,(\phi^{-1}\,\mathrm{id} \circ J\,f) = \phi\,(\phi^{-1}\,(L\,f)) = L\,f$, $R_T\,X = T\,X = R\,(L\,X) = R\,(V\,X)$, $R_T\,k = k^* = R\,(\phi\,k) = R\,(V\,k)$, $V\,(\phi_T\,k) = \phi\,k$.

Uniqueness is established as follows. Any morphism V' between the two splittings must satisfy $V'\,X = V'\,(L_T\,X) = L\,X = V\,X$, $V'\,k = V'\,((\phi_T)_{X,Y}\,k) = V'\,\phi_{X,V\,Y}\,k = \phi_{X,L\,Y}\,k = V\,k$.

For finality of the EM splitting we prove that the following is a unique morphism between a given splitting $(\mathbb{D}, L, R, \phi)$ and the EM splitting $(\mathbf{EM}(T), L^T, R^T, \phi^T)$. We set

$$\begin{array}{lll} V & \in & \mathbb{D} \to \mathbf{EM}(T) \\ V\,X & =_{\mathrm{df}} & (R\,X, \lambda k.\, R\,(\phi\,k)) \\ V\,f & =_{\mathrm{df}} & R\,f \end{array}$$

That $V\,X$ is an EM-algebra is seen by checking that $R(\phi\,k) \circ \eta = k^* \circ \eta = k$, $R\,(\phi\,(R\,(\phi\,\ell) \circ k)) = (\ell^* \circ k)^* = \ell^* \circ k^* = R\,(\phi\,\ell) \circ k^*$ using that $(\mathbb{D}, L, R, \phi)$ is a splitting. The functoriality of V follows immediately from the functoriality of R.

The conditions of a splitting morphisms are verified by $V\,(L\,X) = (R\,(L\,X), \lambda k.\, R\,(\phi\,k)) = (T\,X, \lambda k.\, k^*) = L^T\,X$, $V\,(L\,f) = R\,(L\,f) = T\,f = L^T\,f$, $R\,X = R^T\,(R\,X, \lambda k.\ R\,(\phi\,k)) = R^T\,(V\,X)$, $R\,f = R^T\,(V\,f)$, $V\,(\phi\,k) = R\,(\phi\,k) = k^\star = \phi^T\,k$.

For uniqueness we observe that any splitting V' must satisfy $V'X = (V'_0\,X, V'_1\,X) = (V'_0\,X, \lambda k.\, \phi^T\,k) = (R^T\,(V'\,X), \lambda k.\, V'\,(\phi\,k)) = (R\,X, \lambda k.\, R\,(\phi\,k)) = V\,X$. $\square$

The Kleisli and Eilenberg-Moore categories of our examples correspond to well known concepts.

Example 2.13. The Kleisli category of Vec has as objects the objects of **Fin** understood as finite sets of dimensions. The maps are maps $J_{\mathrm{f}}\,m \to \mathsf{Vec}\,n$, i.e., $m \times n$-matrices (describing linear transformations). The identities are the unit $m \times m$-matrices, the composition is multiplication of matrices. It is the category of finite-dimensional coordinate spaces and linear transformations.

Example 2.14. The Kleisli category $\mathbf{Kl}(\mathsf{Lam})$ of the relative monad for untyped λ-terms (Example 2.4) has as objects the objects of **Fin** understood as untyped contexts. The maps are maps $J_{\mathrm{f}}\,m \to \mathsf{Lam}\,n$, i.e., substitution rules (assignments of terms over n to the variables in m). The identities are the trivial substitution rules. The composition is composition of substitution rules.

Example 2.15. The Kleisli category $\mathbf{Kl}(\mathsf{TyLam})$ of the relative monad for typed λ-terms (Example 2.5) has a very similar structure. Its objects are typed contexts, i.e., objects of $\mathbf{Fin} \downarrow \mathsf{Ty}$, and its morphisms are type-preserving substitution rules. Indeed, the Kleisli category of $\mathsf{TyLam}^{\beta\eta}$ is equivalent to the free cartesian closed category on the set of base types (if we also include finite products into the type language and amend the term language accordingly).

Example 2.16. The Kleisli categories of the two relative monads considered in Example 2.8 are isomorphic to those of the ordinary state and continuation monads.

For $JX =_{\mathrm{df}} X \times S$, $TX =_{\mathrm{df}} X \times S$, T is a relative monad on J and the maps of its Kleisli category are maps $X \times S \to Y \times S$. But the ordinary state monad T' given by $T'X =_{\mathrm{df}} (X \times S)^S$ has maps $X \to (Y \times S)^S$ as the maps of its Kleisli category. Clearly, the two categories are isomorphic.

We also get such an isomorphism for the Kleisli categories of the relative monad T given by $T X =_{\mathrm{df}} R^X$ on J given by $J X =_{\mathrm{df}} R^X$ and the ordinary continuations monad T' given by $T' X =_{\mathrm{df}} R^{R^X}$.

Example 2.17. A vector space (right module) over a semiring $(R, 0, +, 1, \times)$ is given by a commutative monoid $(M, \vec{0}, \oplus)$ and an operation $\cdot \in M \times R \to M$. It is isomorphic to a relative EM-algebra for the relative monad Vec over the semiring R. The carrier of the algebra is defined to be M and the structure map $m \in \Pi_{n \in |\mathbf{Fin}|}(J_{\mathrm{f}}\, n \to M) \to (J_{\mathrm{f}}\, n \to R) \to M$ is given by lifting the operation $\cdot$ straightforwardly to an operation on vectors: $m_n\, f\, g =_{\mathrm{df}} \bigoplus_{i \in \underline{n}} f\, i \cdot g\, i$. Going the other way, given an algebra with carrier M and structure map m, $\vec{0} =_{\mathrm{df}} m_0\, \_\, \_$, $a \oplus a' =_{\mathrm{df}} m_2\, (\lambda i.\, \mathrm{if}\ i = 0\ \mathrm{then}\ a\ \mathrm{else}\ a')(\lambda i.\, 1)$ and $a \cdot r =_{\mathrm{df}} m_1\, (\lambda i.\, a)\, (\lambda i.\, r)$.

Example 2.18. The objects of $\mathbf{EM}(\mathsf{Lam}^{\beta})$ correspond to λ-models, e.g., as given in definition 11.3 in [19, p. 112]. An EM-algebra is given by a set D and for any $n \in |\mathbf{Fin}| = \mathbb{N}$ a function

$$\delta \in (J_{\mathrm{f}}\, n \to D) \to (\mathsf{Lam}^{\beta}\, n \to D)$$

subject to the two conditions stated in definition 2.11. This gives rise to a λ-model with carrier D, the applicative structure can be obtained from δ and δ also gives rise to the evaluation function simply by $[\![t]\!]_{\rho} = \delta\, \rho\, t$. The conditions for a λ-model follow from the conditions of the EM-algebra. The evaluation function in [19] is not scoped, but it can be seen that the explicit indexing corresponds to the variable condition (e). On the other hand we can obtain an EM-algebra from a λ-model in the sense of [19]. We can also show that objects of $\mathbf{EM}(\mathsf{Lam}^{\beta\eta})$ correspond to extensional λ-models.

Example 2.19. In a similar way, the objects of $\mathbf{EM}(\mathsf{TyLam}^{\beta})$ correspond to type frames as given in [17, p. 53]. The carrier of an EM-algebra corresponds to the interpretation of types given by a preframe $\mathcal{A}^{\mathrm{type}}$, while the structure corresponds to the interpretation of terms $\mathcal{A}^{\mathrm{term}}$. The objects of $\mathbf{EM}(\mathsf{TyLam}^{\beta\eta})$ correspond to extensional type frames.

Example 2.20. An algebra of the first of the two relative monads T of Example 2.8 is a pair $(X, \chi \in \int_Z ((Z \times S \to X) \to (Z \times S \to X))$. As $\int_Z ((Z \times S \to X) \to (Z \times S \to X)) \cong (\int^Z (Z \times S \to X) \times Z \times S) \to X \cong (\int^Z (Z \to X^S) \times (Z \times S)) \to X \cong X^S \times S \to X$, this is the same as to give a pair $(X, x \in X^S \times S \to X)$.

The algebras of the state monad $T'\, X =_{\mathrm{df}} (X \times S)^S$ are pairs $(X, x \in (X \times S)^S \to X)$. We can see that the two EM categories are not equivalent.

2.4. **Kleisli and Eilenberg-Moore constructions and restriction.** What is the relationship between the Kleisli and Eilenberg-Moore constructions of some given monad T on $\mathbb{C}$ and the relative monad $T^{\flat}$ on J?

There is a functor $D \in \mathbf{Kl}(T^{\flat}) \to \mathbf{Kl}(T)$ defined as follows:

- for any $X \in |\mathbb{J}|$, $D\, X =_{\mathrm{df}} J\, X$,
- for any $X, Y \in |\mathbb{J}|$, $k \in \mathbb{C}(J\, X, T\, (J\, Y))$, $D\, k =_{\mathrm{df}} k$

No assumptions are needed to prove that D preserves the identities and composition of $\mathbf{Kl}(T^{\flat})$.

Let L, R be the Kleisli adjunction of T, which is given by $L\, X =_{\mathrm{df}} X, L\, f =_{\mathrm{df}} \eta \circ f, R\, X =_{\mathrm{df}} T\, X, R\, k =_{\mathrm{df}} k^*$.

The relative Kleisli adjunction of $T^\flat$ is given by $L'\,X =_{\mathrm{df}} X, L'\,f =_{\mathrm{df}} \eta^\flat \circ J\,f = \eta \circ J\,f, R'\,X =_{\mathrm{df}} T^\flat X = T\,(J\,X), R'\;k =_{\mathrm{df}} k^{(*^\flat)} = k^*$.

We have $D \cdot L' = L \cdot J$ and $R' = R \cdot D$. Moreover, the category $\mathbf{Kl}(T)$ together with the functors $L \cdot J$ and R gives a splitting of $T^\flat$: we have $R \cdot (L \cdot J) = T \cdot J$ and $L \cdot J$ is relative left adjoint to R.

In general we can define no functor in the opposite direction $\mathbf{Kl}(T) \to \mathbf{Kl}(T^\flat)$, for the simple reason that this would require some canonical functor $\mathbb{C} \to \mathbb{J}$ and we have none given.

There is also a functor $E \in \mathbf{EM}(T) \to \mathbf{EM}(T^\flat)$ defined by

- for any $(X, x) \in |\mathbf{EM}(T)|$, i.e., $X \in |\mathbb{C}|$, $x \in \mathbb{C}(TX, X)$ meeting the EM-algebra conditions, $E(X, x) =_{\mathrm{df}} (X, \chi)$ where for $Z \in |\mathbb{J}|$, $f \in \mathbb{C}(JZ, X)$, $\chi_Z\, f =_{\mathrm{df}} x \circ T\,f \in \mathbb{C}(T(JZ), X)$; $E(X, x)$ is a relative EM-algebra of $T^\flat$ under no assumptions;
- for any $h \in \mathbf{EM}(T)((X, x), (Y, y))$, i.e., $h \in \mathbb{C}(X, Y)$ meeting the EM-algebra morphism conditions, $E\,h =_{\mathrm{df}} h$;
  $E\,h$ satisfies the relative EM-algebra conditions.

It is trivial that E preserves the identities and composition of $\mathbf{EM}(T)$.

Let F, U be the EM adjunction of T, which is given by $F\,X =_{\mathrm{df}} (T\,X, \mu_X)$, $F\,f = T\,f$, $U\,(X, x) =_{\mathrm{df}} X$, $U\,h =_{\mathrm{df}} h$.

The relative EM adjunction of $T^\flat$ is given by $F'\,X =_{\mathrm{df}} (T\,(J\,X), (-)^*)$, $F'\,f =_{\mathrm{df}} T\,(J\,f)$, $U'\,(X, \chi) =_{\mathrm{df}} X$, $U'\,h =_{\mathrm{df}} h$.

We have $F' = E \cdot (F \cdot J)$ and $U' \cdot E = U$. Furthermore, the category $\mathbf{EM}(T)$ together with the functors $F \cdot J$ and U gives a splitting for $T^\flat$: we have $U \cdot (F \cdot J) = T \cdot J$ and $F \cdot J$ is relative left adjoint to U.

In general, we cannot construct a functor $\mathbf{EM}(T^\flat) \to \mathbf{EM}(T)$.

This situation is illustrated on the following diagram.

$$\begin{tikzcd}
 & \mathbf{Kl}(T) \arrow[r, dotted] \arrow[dr, bend left, "R"] & \mathbf{EM}(T) \arrow[dr, bend left, "E"] \arrow[d, bend left, "U"] & \\
\mathbf{Kl}(T^\flat) \arrow[ur, bend left, "D"] \arrow[drr, "R'"] & & & \mathbf{EM}(T^\flat) \arrow[dl, "U'"] \\
 & \mathbb{J} \arrow[ul, "L'"] \arrow[r, "J"'] \arrow[urr, "F'"] & \mathbb{C} \arrow[uul, bend left, "L"] \arrow[u, bend left, "F"] &
\end{tikzcd}$$

3. Relative monads as skew-monoids in a skew-monoidal category

A monad on $\mathbb{C}$ is the same as a monoid in the endofunctor category $[\mathbb{C}, \mathbb{C}]$. This category has a monoidal structure given by the identity functor I and composition of functors $\cdot$, which are strictly unital and associative. A monad can be specified by an object $T \in |[\mathbb{C}, \mathbb{C}]|$ and maps $\eta \in [\mathbb{C}, \mathbb{C}]\,(I, T)$ and $\mu \in [\mathbb{C}, \mathbb{C}]\,(T \cdot T, T)$ satisfying the laws of a monoid in the strict monoidal category $([\mathbb{C}, \mathbb{C}], I, \cdot)$.

Can we similarly define a relative monad on $J \in \mathbb{J} \to \mathbb{C}$ as a monoid in the functor category $[\mathbb{J}, \mathbb{C}]$? This requires a monoidal structure on $[\mathbb{J}, \mathbb{C}]$, ideally similar to that on $[\mathbb{C}, \mathbb{C}]$. The functor J is a good candidate for the unit, but the tensor is problematic, as functors $\mathbb{J} \to \mathbb{C}$ cannot be composed by simple functor composition. We shall use a left Kan extension to overcome the difficulty and obtain a skew-monoidal structure where relative monads are skew-monoids.

3.1. **Left Kan extensions.** Left Kan extensions are one of the two canonical constructions for extending functors. The left Kan extension along $J \in \mathbb{J} \to \mathbb{C}$ extends functors $\mathbb{J} \to \mathbb{D}$ to functors $\mathbb{C} \to \mathbb{D}$.

$$\begin{tikzcd} & \mathbb{D} & \\ \mathbb{J} \arrow[ur, bend left, "F"] \arrow[rr, "J"] & & \mathbb{C} \arrow[ul, bend right, "\mathrm{Lan}_J\, F"'] \end{tikzcd}$$

It is defined as the left adjoint (if it exists) of the restriction functor $- \cdot J \in [\mathbb{C}, \mathbb{D}] \to [\mathbb{J}, \mathbb{D}]$. i.e., it is given by a functor $\mathrm{Lan}_J \in [\mathbb{J}, \mathbb{D}] \to [\mathbb{C}, \mathbb{D}]$ and a natural isomorphism

$$[\mathbb{J}, \mathbb{D}]\,(F, G \cdot J) \cong [\mathbb{C}, \mathbb{D}]\,(\mathrm{Lan}_J\, F, G)$$

While it is possible to work directly with this definition of left Kan extension, we use an alternative definition, based on the coend formula

$$\mathrm{Lan}_J\, F\, X \cong \int^{Y \in |\mathbb{J}|} \mathbb{C}\,(J\, Y, X) \bullet F\, Y$$

Accordingly, we take a left Kan extension of a functor $F \in \mathbb{J} \to \mathbb{D}$ along $J \in \mathbb{J} \to \mathbb{C}$ to be given by

- an object function $\mathrm{Lan}_J\, F \in |\mathbb{C}| \to |\mathbb{D}|$,
- for any $X \in |\mathbb{C}|$, a natural transformation $\iota_{F,X} \in [\mathbb{J}^{\mathrm{op}}, \mathbf{Set}]\,(\mathbb{C}\,(J\,-, X), \mathbb{D}\,(F\,-, \mathrm{Lan}_J\, F\, X))$,
- for any $X \in |\mathbb{C}|$, $Y \in |\mathbb{D}|$ and $\theta \in [\mathbb{J}^{\mathrm{op}}, \mathbf{Set}]\,(\mathbb{C}\,(J\,-, X), \mathbb{D}\,(F\,-, Y))$, a map $[\theta] \in \mathbb{D}\,(\mathrm{Lan}_J\, F\, X, Y)$.

satisfying the conditions $[\theta] \circ \iota\, g = \theta\, g$, $[\iota] = \mathrm{id}$ and $f \circ [\theta] = [\lambda g. f \circ \theta\, g]$.

Left Kan extensions $\mathrm{Lan}_J\, F\, X$ are functorial in both arguments F and X, i.e., $\mathrm{Lan}_J \in [\mathbb{J}, \mathbb{D}] \to [\mathbb{C}, \mathbb{D}]$. For any $F \in |[\mathbb{J}, \mathbb{D}]|$, $X, Y \in |\mathbb{C}|$, $f \in \mathbb{C}\,(X, Y)$,

$$\begin{array}{l} \mathrm{Lan}_J\, F\, f \in \mathbb{D}\,(\mathrm{Lan}_J\, F\, X, \mathrm{Lan}_J\, F\, Y) \\ \mathrm{Lan}_J\, F\, f =_{\mathrm{df}} [\lambda g.\, \iota\,(f \circ g)] \end{array}$$

And for any $F, G \in |[\mathbb{J}, \mathbb{D}]|$, $\tau \in [\mathbb{J}, \mathbb{D}]\,(F, G)$, $X \in |\mathbb{C}|$, we have

$$\begin{array}{l} (\mathrm{Lan}_J\, \tau)_X \in \mathbb{D}\,(\mathrm{Lan}_J\, F\, X, \mathrm{Lan}_J\, G\, X) \\ (\mathrm{Lan}_J\, \tau)_X =_{\mathrm{df}} [\lambda g.\, \iota\, g \circ \tau] \end{array}$$

In general $\mathrm{Lan}_J \in [\mathbb{J}, \mathbb{D}] \to [\mathbb{C}, \mathbb{D}]$ exists, if $\mathbb{J}$ is small and $\mathbb{D}$ is cocomplete.

3.2. $[\mathbb{J}, \mathbb{C}]$ **is skew-monoidal.** If $\mathrm{Lan}_J \in [\mathbb{J}, \mathbb{C}] \to [\mathbb{C}, \mathbb{C}]$ exists, we can turn any functor $F \in |[\mathbb{J}, \mathbb{C}]|$ into one in $|[\mathbb{C}, \mathbb{C}]|$. Hence we can define a composition-like operation

$$\begin{array}{l} (\cdot^J) \in |[\mathbb{J}, \mathbb{C}]| \times |[\mathbb{J}, \mathbb{C}]| \to |[\mathbb{J}, \mathbb{C}]| \\ F \cdot^J G =_{\mathrm{df}} \mathrm{Lan}_J\, F \;\cdot\; G \end{array}$$

This is our candidate for the tensor on $[\mathbb{J}, \mathbb{C}]$. We also need the unital and associative laws. We define several families of maps indexed by $X \in |\mathbb{C}|$:

$$
\begin{aligned}
&\overline{\lambda}_X \in \mathbb{C}\,(\mathrm{Lan}_J\, J\, X, X)\\
&\overline{\lambda}_X =_{\mathrm{df}} [\lambda g.\, g]\\
&(\overline{\overline{\alpha}}_{F,G})_X \in \mathbb{C}\,(\mathrm{Lan}_J\,(F \cdot G)\, X, F\,(\mathrm{Lan}_J\, G\, X))\\
&(\overline{\overline{\alpha}}_{F,G})_X =_{\mathrm{df}} [\lambda g.\, F\,(\iota\, g)]\\
&(\overline{\alpha}_{F,G})_X \in \mathbb{C}\,(\mathrm{Lan}_J\,(\mathrm{Lan}_J\, F \cdot G)\, X, \mathrm{Lan}_J\, F\,(\mathrm{Lan}_J\, G\, X))\\
&(\overline{\alpha}_{F,G})_X =_{\mathrm{df}} (\overline{\overline{\alpha}}_{\mathrm{Lan}_J F,G})_X = [\lambda g.\, [\lambda g'.\, \iota\,(\iota\, g \circ g')]]
\end{aligned}
$$

All these families are natural in X, hence maps in $|[\mathbb{C}, \mathbb{C}]|$.

From these we further define our candidate unital and associative laws.

$$
\begin{aligned}
&\rho_F \in [\mathbb{J}, \mathbb{C}]\,(F, F \cdot^J J)\\
&\rho_F =_{\mathrm{df}} \iota\,\mathrm{id}\\
&\lambda_F \in [\mathbb{J}, \mathbb{C}]\,(J \cdot^J F, F)\\
&\lambda_F =_{\mathrm{df}} \overline{\lambda} \cdot F\\
&\alpha_{F,G,H} \in [\mathbb{J}, \mathbb{C}]\,((F \cdot^J G) \cdot^J H, F \cdot^J (G \cdot^J H))\\
&\alpha_{F,G,H} =_{\mathrm{df}} \overline{\alpha}_{F,G} \cdot H
\end{aligned}
$$

It turns out that the data so defined provide a structure that is almost monoidal, but not quite. It is skew-monoidal in the sense of Szlachányi [33]: while λ, ρ, α are generally not isomorphisms, they meet appropriate coherence conditions, namely the conditions (a)–(e) below. Importantly, in contrast to the properly monoidal case, all five conditions are necessary: the conditions (a), (c), (d) do not follow from (b) and (e).

In the next section we will identify conditions on J that enable us to construct the inverses, making the skew-monoidal structure properly monoidal.

Theorem 3.1. *If* $\mathrm{Lan}_J \in [\mathbb{J}, \mathbb{C}] \to [\mathbb{C}, \mathbb{C}]$ *exists, then* $([\mathbb{J}, \mathbb{C}], J, \cdot^J, \lambda, \rho, \alpha)$ *is a skew-monoidal category, i.e.,* $\cdot^J$ *is functorial,* λ*,* ρ*,* α *are natural and the following diagrams commute:*

(a)
$$
\begin{tikzcd}
 & J \cdot^J J \arrow[dr, "\lambda_J"] & \\
J \arrow[ur, "\rho_J"] \arrow[rr, equal] & & J
\end{tikzcd}
$$

(b)
$$
\begin{tikzcd}
(F \cdot^J J) \cdot^J G \arrow[r, "\alpha_{F,J,G}"] & F \cdot^J (J \cdot^J G) \arrow[d, "F \cdot^J \lambda_G"] \\
F \cdot^J G \arrow[u, "\rho_F \cdot^J G"] \arrow[r, equal] & F \cdot^J G
\end{tikzcd}
$$

(c)
$$
\begin{tikzcd}
(J \cdot^J F) \cdot^J G \arrow[rr, "\alpha_{J,F,G}"] \arrow[dr, "\lambda_F \cdot^J G"'] & & J \cdot^J (F \cdot^J G) \arrow[dl, "\lambda_{F \cdot^J G}"] \\
 & F \cdot^J G &
\end{tikzcd}
$$

(d)
$$
\begin{tikzcd}
(F \cdot^J G) \cdot^J J \arrow[rr, "\alpha_{F,G,J}"] & & F \cdot^J (G \cdot^J J) \\
 & F \cdot^J G \arrow[ul, "\rho_{F \cdot^J G}"] \arrow[ur, "F \cdot^J \rho_G"'] &
\end{tikzcd}
$$

(e)
$$
\begin{tikzcd}
(F \cdot^J (G \cdot^J H)) \cdot^J K \arrow[rr, "\alpha_{F,G \cdot^J H,K}"] & & F \cdot^J ((G \cdot^J H) \cdot^J K) \arrow[d, "F \cdot^J \alpha_{G,H,K}"] \\
((F \cdot^J G) \cdot^J H) \cdot^J K \arrow[u, "\alpha_{F,G,H} \cdot^J K"] \arrow[r, "\alpha_{F \cdot^J G,H,K}"] & (F \cdot^J G) \cdot^J (H \cdot^J K) \arrow[r, "\alpha_{F,G,H \cdot^J K}"] & F \cdot^J (G \cdot^J (H \cdot^J K))
\end{tikzcd}
$$

Proof. The required properties follow from the definitions of the functorial actions of Lan_J in both of its arguments, λ, ρ, α, and the laws of Lan_J by easy calculations.

We prove generalizations of properties (b), (c), and (e):

(b')
$$\begin{tikzcd}
\mathrm{Lan}_J(F \cdot^J J) \arrow[r, "\overline{\alpha}_{F,J}"] & F \cdot^J \mathrm{Lan}_J J \arrow[d, "\mathrm{Lan}_J F\,\overline{\lambda}"] \\
\mathrm{Lan}_J F \arrow[u, "\mathrm{Lan}_J \rho_F"] \arrow[r, equal] & \mathrm{Lan}_J F
\end{tikzcd}$$

(c')
$$\begin{tikzcd}
\mathrm{Lan}_J(J \cdot^J F) \arrow[rr, "\overline{\alpha}_{J,F}"] \arrow[dr, "\mathrm{Lan}_J \lambda_F"'] & & J \cdot^J \mathrm{Lan}_J F \arrow[dl, "\overline{\lambda} \cdot \mathrm{Lan}_J F"] \\
& \mathrm{Lan}_J F &
\end{tikzcd}$$

(e')
$$\begin{tikzcd}
\mathrm{Lan}_J(F \cdot^J (G \cdot^J H)) \arrow[rr, "\overline{\alpha}_{F,G\cdot^J H}"] & & F \cdot^J (\mathrm{Lan}_J(G \cdot^J H)) \arrow[d, "F \cdot^J \overline{\alpha}_{G,H}"] \\
\mathrm{Lan}_J((F \cdot^J G) \cdot^J H) \arrow[u, "\mathrm{Lan}_J \alpha_{F,G,H}"] \arrow[r, "\overline{\alpha}_{F\cdot^J G,H}"] & (F \cdot^J G) \cdot^J \mathrm{Lan}_J H \arrow[r, "\alpha_{F,G,\mathrm{Lan}_J H}"] & F \cdot^J (G \cdot^J \mathrm{Lan}_J H)
\end{tikzcd}$$

(b), (c) and (e) follow from (b'), (c') and (e') as simple instances.

We skip the proofs of functoriality of $\cdot^J$ and naturality of λ, ρ and α. The calculations for the other five laws are as follows:

(a)

$$\begin{aligned}
(\lambda_J)_X \circ (\rho_J)_X &= [\lambda g.\, g] \circ \iota\, \mathrm{id} \\
&= \mathrm{id}_{J\,X}
\end{aligned}$$

(b')

$$\begin{aligned}
&\mathrm{Lan}_J\, F\, \overline{\lambda}_X \circ (\overline{\alpha}_{F,J})_X \circ (\mathrm{Lan}_J\, \rho_F)_X \\
&= \mathrm{Lan}_J\, F\, \overline{\lambda}_X \circ (\overline{\alpha}_{F,G})_X \circ [\lambda g.\, \iota\, g \circ \iota\, \mathrm{id}] \\
&= \mathrm{Lan}_J\, F\, \overline{\lambda}_X \circ [\lambda g.\, (\overline{\alpha}_{F,J})_X \circ \iota\, g \circ \iota\, \mathrm{id}] \\
&= \mathrm{Lan}_J\, F\, \overline{\lambda}_X \circ [\lambda g.\, [\lambda g.\, [\lambda g'.\, \iota\, (\iota\, g \circ g')]] \circ \iota\, g \circ \iota\, \mathrm{id}] \\
&= \mathrm{Lan}_J\, F\, \overline{\lambda}_X \circ [\lambda g.\, [\lambda g'.\, \iota\, (\iota\, g \circ g')] \circ \iota\, \mathrm{id}] \\
&= \mathrm{Lan}_J\, F\, \overline{\lambda}_X \circ [\lambda g.\, \iota\, (\iota\, g)] \\
&= [\lambda g.\, \mathrm{Lan}_J\, F\, \overline{\lambda}_X \circ \iota\, (\iota\, g)] \\
&= [\lambda g.\, [\lambda g.\ \iota([\lambda g'.\, g'] \circ g)] \circ \iota\, (\iota\, g)] \\
&= [\lambda g.\, \iota\, ([\lambda g'.\, g'] \circ \iota\, g)] \\
&= [\lambda g.\, \iota\, g] \\
&= \mathrm{id}_{\mathrm{Lan}_J\, F\, X}
\end{aligned}$$

(c')

$$\begin{aligned}
\overline{\lambda}_{\mathrm{Lan}_J\, F\, X} \circ (\overline{\alpha}_{J,F})_X &= \overline{\lambda}_{\mathrm{Lan}_J\, F\, X} \circ [\lambda g.\, [\lambda g'.\, \iota\, (\iota\, g \circ g')]] \\
&= [\lambda g.\, \overline{\lambda}_{\mathrm{Lan}_J\, F\, X} \circ [\lambda g'.\, \iota\, (\iota\, g \circ g')]] \\
&= [\lambda g.\, [\lambda g'.\, \overline{\lambda}_{\mathrm{Lan}_J\, F\, X} \circ \iota\, (\iota\, g \circ g')]] \\
&= [\lambda g.\, [\lambda g'.\, [\lambda g''.\, g''] \circ \iota\, (\iota\, g \circ g')]] \\
&= [\lambda g.\, [\lambda g'.\, \iota\, g \circ g']] \\
&= [\lambda g.\, \iota\, g \circ [\lambda g'.\ g']] \\
&= (\mathrm{Lan}_J\, \lambda_F)_X
\end{aligned}$$

(d)

$$\begin{array}{rcl}(\alpha_{F,G,J})_X \circ (\rho_{\mathrm{Lan}_J F \cdot G})_X & = & [\lambda g.\, [\lambda g'.\, \iota\, (\iota\, g \circ g')]] \circ \iota\, \mathrm{id} \\ & = & [\lambda g'.\, \iota\, (\iota\, \mathrm{id} \circ g')] \\ & = & [\lambda g'.\, \iota\, ((\rho_G)_X \circ g')] \\ & = & (\mathrm{Lan}_J\, F\, \rho_G)_X\end{array}$$

(e')

$$\begin{array}{rl}\multicolumn{2}{l}{\mathrm{Lan}_J\, F\, (\overline{\alpha}_{G,H})_X \circ (\overline{\alpha}_{F,\mathrm{Lan}_J G \cdot H})_X \circ (\mathrm{Lan}_J\, \alpha_{F,G,H})_X} \\ = & \mathrm{Lan}_J\, F\, (\overline{\alpha}_{G,H})_X \circ (\overline{\alpha}_{F,\mathrm{Lan}_J G \cdot H})_X \circ [\lambda g.\, \iota\, g \circ [\lambda g'.\, [\lambda g''.\, \iota\, (\iota\, g' \circ g'')]]] \\ = & \mathrm{Lan}_J\, F\, (\overline{\alpha}_{G,H})_X \circ [\lambda g.\, (\overline{\alpha}_{F,\mathrm{Lan}_J G \cdot H})_X \circ \iota\, g \circ [\lambda g'.\, [\lambda g''.\, \iota\, (\iota\, g' \circ g'')]]] \\ = & \mathrm{Lan}_J\, F\, (\overline{\alpha}_{G,H})_X \circ [\lambda g.\, [\lambda g.\, [\lambda g'.\, \iota\, (\iota\, g \circ g')]] \circ \iota\, g \circ [\lambda g'.\, [\lambda g''.\, \iota\, (\iota\, g' \circ g'')]]] \\ = & \mathrm{Lan}_J\, F\, (\overline{\alpha}_{G,H})_X \circ [\lambda g.\, [\lambda g'.\, \iota\, (\iota\, g \circ g')] \circ [\lambda g'.\, [\lambda g''.\, \iota\, (\iota\, g' \circ g'')]]] \\ = & \mathrm{Lan}_J\, F\, (\overline{\alpha}_{G,H})_X \circ [\lambda g.\, [\lambda g'.\, [\lambda g''.\, [\lambda g'.\, \iota\, (\iota\, g \circ g')] \circ \iota\, (\iota\, g' \circ g'')]]] \\ = & \mathrm{Lan}_J\, F\, (\overline{\alpha}_{G,H})_X \circ [\lambda g.\, [\lambda g'.\, [\lambda g''.\, \iota\, (\iota\, g \circ \iota\, g' \circ g'')]]] \\ = & [\lambda g.\, [\lambda g'.\, [\lambda g''.\, \mathrm{Lan}_J\, F\, (\overline{\alpha}_{G,H})_X \circ \iota\, (\iota\, g \circ \iota\, g' \circ g'')]]] \\ = & [\lambda g.\, [\lambda g'.\, [\lambda g''.\, [\lambda g.\, \iota\, ((\overline{\alpha}_{G,H})_X \circ g)] \circ \iota\, (\iota\, g \circ \iota\, g' \circ g'')]]] \\ = & [\lambda g.\, [\lambda g'.\, [\lambda g''.\, \iota\, ((\overline{\alpha}_{G,H})_X \circ \iota\, g \circ \iota\, g' \circ g'')]]] \\ = & [\lambda g.\, [\lambda g'.\, [\lambda g''.\, \iota\, ([\lambda g.\, [\lambda g'.\, \iota\, (\iota\, g \circ g')]] \circ \iota\, g \circ \iota\, g' \circ g'')]]] \\ = & [\lambda g.\, [\lambda g'.\, [\lambda g''.\, \iota\, ([\lambda g'.\, \iota\, (\iota\, g \circ g')] \circ \iota\, g' \circ g'')]]] \\ = & [\lambda g.\, [\lambda g'.\, [\lambda g''.\, \iota\, (\iota\, (\iota\, g \circ g') \circ g'')]]] \\ = & [\lambda g.\, [\lambda g'.\, [\lambda g.\, [\lambda g''.\, \iota\, (\iota\, g \circ g'')]] \circ \iota\, (\iota\, g \circ g')]] \\ = & [\lambda g.\, [\lambda g'.\, (\overline{\alpha}_{F,G})_{\mathrm{Lan}_J H X} \circ \iota\, (\iota\, g \circ g')]] \\ = & (\overline{\alpha}_{F,G})_{\mathrm{Lan}_J H X} \circ [\lambda g.\, [\lambda g'.\, \iota\, (\iota\, g \circ g')]] \\ = & (\overline{\alpha}_{F,G})_{\mathrm{Lan}_J H X} \circ (\overline{\alpha}_{\mathrm{Lan}_J F \cdot G, H})_X\end{array}$$

Example 3.2. The functor category $[\mathbb{J}, \mathbb{C}]$ is skew-monoidal, but not monoidal, for $\mathbb{J} =_{\mathrm{df}} \mathbf{Set}$, $\mathbb{C} =_{\mathrm{df}} \mathbf{Set}$, $J\, X =_{\mathrm{df}} X \times S$.

In this case, we have $\mathrm{Lan}_J\, F\, X \cong \int^Y ((Y \times S \to X) \times F\, Y) \cong \int^Y ((Y \to X^S) \times F\, Y) \cong F\, (X^S)$.

Accordingly, $(\rho_F)_X \in F\, X \to \mathrm{Lan}_J\, F\, (J\, X)$ is given by $(\rho_X)_F =_{\mathrm{df}} F\, \mathrm{coeval}_X \in F\, X \to F((X \times S)^S)$, $\overline{\lambda}_X \in \mathrm{Lan}_J\, J\, X \to X$ is given by $\overline{\lambda}_X =_{\mathrm{df}} \mathrm{eval}_X \in X^S \times S \to X$. It is clear that ρ and $\overline{\lambda}$ are not isomorphisms in this case.

The map $(\overline{\overline{\alpha}}_{F,G})_X \in \mathrm{Lan}_J\, (F \cdot G)\, X \to F\, (\mathrm{Lan}_J\, G\, X)$ however is given by the identity on $F\, (G\, (X^S))$ and is therefore trivially an isomorphism.

Example 3.3. For $\mathbb{J} =_{\mathrm{df}} \mathbf{Set}$, $\mathbb{C} =_{\mathrm{df}} \mathbf{Set}$, $J\, X =_{\mathrm{df}} X + E$, the functor category $[\mathbb{J}, \mathbb{C}]$ is also skew-monoidal. But in this case, even the associativity law α fails to be an isomorphism.

We have $\mathrm{Lan}_J\, F\, X \cong \int^Y ((Y + E \to X) \times F\, Y) \cong \int^Y ((Y \to X) \times (E \to X) \times F\, Y) \cong F\, X \times X^E$.

Accordingly, ρ, λ, α are the canonical natural transformations with components $(\rho_F)_X \in F\, X \to F(X + E) \times (X + E)^E$, $\overline{\lambda}_X \in (X + E) \times X^E \to X$, $(\overline{\overline{\alpha}}_{F,G})_X \in F\, (G\, X) \times X^E \to F\, (G\, X \times X^E)$. None of these has an inverse.

3.3. **Relative monads are the same as skew-monoids in** $[\mathbb{J}, \mathbb{C}]$**.** With a skew-monoidal structure present on the functor category $[\mathbb{J}, \mathbb{C}]$, we should expect that relative monads on J are the same thing as skew-monoids in this structure, generalizing the case of ordinary monads on $\mathbb{C}$ and the strict monoidal structure on the endofunctor category $[\mathbb{C}, \mathbb{C}]$. This is indeed the case.

Theorem 3.4. *Assume that* $\mathrm{Lan}_J \in [\mathbb{J}, \mathbb{C}] \to [\mathbb{C}, \mathbb{C}]$ *exists.*

(1) *Given a relative monad* $(T, \eta, (-)^*)$ *on* J*, define, for any* $X \in |\mathbb{J}|$*, a map* $\mu_X \in \mathbb{C}\,(\mathrm{Lan}_J\, T\,(T\,X), T\,X)$ *by* $\mu_X =_{\mathrm{df}} [(-)^*]$*. This is well-defined, since* $(-)^*$ *is natural:* $(-)^* \in [\mathbb{J}^{\mathrm{op}}, \mathbf{Set}]\,(\mathbb{C}\,(J\,-, T\,X), \mathbb{C}\,(T\,-, T\,X))$*.*

   Then (T, η, μ) *is a skew-monoid in the skew-monoidal category* $([\mathbb{J}, \mathbb{C}], J, \cdot^J, \lambda, \rho, \alpha)$*: we have that* $T \in |[\mathbb{J}, \mathbb{C}]|$*,* $\eta \in [\mathbb{J}, \mathbb{C}]\,(J, T)$ *and* $\mu \in [\mathbb{J}, \mathbb{C}]\,(T \cdot^J T, T)$*, and the following diagrams commute in* $[\mathbb{J}, \mathbb{C}]$*:*

$$\begin{tikzcd} T \cdot^J J \arrow[r, "T \cdot^J \eta"] & T \cdot^J T \arrow[d, "\mu"] \\ T \arrow[u, "\rho_T"] \arrow[r, equal] & T \end{tikzcd} \qquad \begin{tikzcd} J \cdot^J T \arrow[r, "\lambda_T"] \arrow[d, "\eta \cdot^J T"'] & T \arrow[d, equal] \\ T \cdot^J T \arrow[r, "\mu"] & T \end{tikzcd} \qquad \begin{tikzcd} & T \cdot^J (T \cdot^J T) \arrow[r, "T \cdot^J \mu"] & T \cdot^J T \arrow[dd, "\mu"] \\ (T \cdot^J T) \cdot^J T \arrow[ur, "\alpha_{T,T,T}"] \arrow[d, "\mu \cdot^J T"'] & & \\ T \cdot^J T \arrow[rr, "\mu"] & & T \end{tikzcd}$$

(2) *Given a skew-monoid* (T, η, μ) *in* $([\mathbb{J}, \mathbb{C}], J, \cdot^J, \lambda, \rho, \alpha)$*, define, for any* $X, Y \in |\mathbb{J}|$*, a function* $(-)^* \in \mathbb{C}\,(J\,X, T\,Y) \to \mathbb{C}\,(T\,X, T\,Y)$ *by* $k^* =_{\mathrm{df}} \mu_Y \circ \iota\, k$*. Then* $(T, \eta, (-)^*)$ *is a relative monad on* J*.*

(3) *The above correspondence is bijective.*

Proof.

(1) The required properties follow from the definitions of μ and the functorial action of Lan_J and from T being a relative monad by the laws of Lan_J alone.

   For naturality of μ, we easily verify that, for any $f \in J\,(X, Y)$,

$$\begin{aligned} T\,f \circ \mu_X &= T\,f \circ [\lambda g.\, g^*] \\ &= [\lambda g.\, T\,f \circ g^*] \\ &= \quad \{ \text{ by naturality of } (-)^* \text{ } \} \\ &\quad\; [\lambda g.\, (T\,f \circ g)^*] \\ &= [\lambda g.\, [\lambda g.\, g^*] \circ \iota\,(T\,f \circ g)] \\ &= [\lambda g.\, g^*] \circ [\lambda g.\, \iota\,(T\,f \circ g)] \\ &= \mu_Y \circ \mathrm{Lan}_J\, T\,(T\,f) \end{aligned}$$

   The right unital law of T as a monoid is verified by

$$\begin{aligned} \mu_X \circ (\mathrm{Lan}_J\, \eta)_{T\,X} &= [\lambda g.\, g^*] \circ [\lambda g.\, \iota\, g \circ \eta] \\ &= [\lambda g.\, [\lambda g.\, g^*] \circ \iota\, g \circ \eta] \\ &= [\lambda g.\, g^* \circ \eta] \\ &= \quad \{ \text{ by right unital law of } T \text{ as a relative monad } \} \\ &\quad\; [\lambda g.\, g] \\ &= \lambda_{T,X} \end{aligned}$$

The left unital law of T as a monoid is checked by

$$\begin{array}{rcl}
\mu_X \circ \mathrm{Lan}_J\, T\, \eta_X \circ (\rho_T)_X & = & [\lambda g.\, g^*] \circ [\lambda g.\, \iota\, (\eta_X \circ g)] \circ (\rho_T)_X \\
& = & [\lambda g.\, [\lambda g.\, g^*] \circ \iota\, (\eta_X \circ g)] \circ (\rho_T)_X \\
& = & [\lambda g.\, (\eta_X \circ g)^*] \circ (\rho_T)_X \\
& = & [\lambda g.\, (\eta_X \circ g)^*] \circ \iota\, \mathrm{id}_{J\, X} \\
& = & (\eta_X \circ \mathrm{id}_{J\, X})^* \\
& = & \qquad \{ \text{ by left unital law of } T \text{ as a relative monad } \} \\
& & \mathrm{id}_{T\, X}
\end{array}$$

The associativity of T as a monoid is verified by

$$\begin{array}{rcl}
\mu_X \circ \mathrm{Lan}_J\, T\, \mu_X \circ (\alpha_{T,T,T})_X & = & [\lambda g.\, g^*] \circ [\lambda g.\, \iota\, (\mu_X \circ g)] \circ (\alpha_{T,T,T})_X \\
& = & [\lambda g.\, [\lambda g.\, g^*] \circ \iota\, (\mu_X \circ g)] \circ (\alpha_{T,T,T})_X \\
& = & [\lambda g.\, (\mu_X \circ g)^*] \circ (\alpha_{T,T,T})_X \\
& = & [\lambda g.\, (\mu_X \circ g)^*] \circ [\lambda g.\, [\lambda g'.\, \iota\, (\iota\, g \circ g')]] \\
& = & [\lambda g.\, [\lambda g.\, (\mu_X \circ g)^*] \circ [\lambda g'.\, \iota\, (\iota\, g \circ g')]] \\
& = & [\lambda g.\, [\lambda g'.\, [\lambda g.\, (\mu_X \circ g)^*] \circ \iota\, (\iota\, g \circ g')]] \\
& = & [\lambda g.\, [\lambda g'.\, (\mu_X \circ \iota\, g \circ g')^*]] \\
& = & [\lambda g.\, [\lambda g'.\, ([\lambda g.\, g^*] \circ \iota\, g \circ g')^*]] \\
& = & [\lambda g.\, [\lambda g'.\, (g^* \circ g')^*]] \\
& = & \qquad \{ \text{ by associative law of } T \text{ as a relative monad } \} \\
& & [\lambda g.\, [\lambda g'.\, g^* \circ g'^*]] \\
& = & [\lambda g.\, g^* \circ [\lambda g'.\, g'^*]] \\
& = & [\lambda g.\, g^* \circ \mu] \\
& = & [\lambda g.\, [\lambda g.\, g^*] \circ \iota\, g \circ \mu] \\
& = & [\lambda g.\, g^*] \circ [\lambda g.\, \iota\, g \circ \mu] \\
& = & \mu_X \circ (\mathrm{Lan}_J\, \mu)_{T\, X}
\end{array}$$

(2) The claim follows from the definitions of $(-)^*$ and the functorial action of Lan_J and from T being a skew-monoid by the laws of Lan_J.
(3) The claim follows from the definitions of μ and $(-)^*$ from each other and the laws of Lan_J. □

The bijective correspondence between relative monads on J and skew-monoids in $[\mathbb{J}, \mathbb{C}]$ extends to an isomorphism of categories.

Theorem 3.5. *Assume that* $\mathrm{Lan}_J \in [\mathbb{J}, \mathbb{C}] \to [\mathbb{C}, \mathbb{C}]$ *exists.*

(1) *A morphism σ between relative monads $(T, \eta, (-)^*)$ and $(T', \eta', (-)^{*'})$ is a morphism between the corresponding skew-monoids (T, η, μ) and (T, η', μ'): we have that* $\sigma \in$

$[\mathbb{J},\mathbb{C}](T,T')$ and the following diagrams commute in $[\mathbb{J},\mathbb{C}]$:

$$\begin{tikzcd} & T \arrow[dd, "\sigma"] \\ J \arrow[ur, "\eta"] \arrow[dr, "\eta'"'] & \\ & T' \end{tikzcd} \qquad \begin{tikzcd} T \cdot^J T \arrow[r, "\mu"] \arrow[d, "\sigma \cdot^J \sigma"'] & T \arrow[d, "\sigma"] \\ T' \cdot^J T' \arrow[r, "\mu'"] & T' \end{tikzcd}$$

(2) *A morphism σ between skew-monoids (T,η,μ) and (T',η',μ') is also a morphism between the corresponding relative monads $(T,\eta,(-)^*)$ and $(T,\eta',(-)^{*'})$.*
(3) *The above correspondence is an isomorphism of the categories of relative monads on J and skew-monoids in the skew-monoidal category $([\mathbb{J},\mathbb{C}],J,\cdot^J,\lambda,\rho,\alpha)$.*

Proof.
(1) We have already observed that σ is natural. The unit preservation law for σ as a skew-monoid morphism is the same as the unit preservation law of σ as a relative monad morphism.

The multiplication preservation law of σ as a skew-monoid morphism follows from the definitions of μ, μ' from $(-)^*$, $(-)^{*'}$ and Kleisli extension preservation of σ as a relative monad morphism by the laws of Lan_J:

$$\begin{array}{ll} \multicolumn{2}{l}{\mu'_X \circ (\mathrm{Lan}_J\,\sigma)_{T'\,X} \circ \mathrm{Lan}_J\,T\,\sigma_X} \\ = & \mu'_X \circ (\mathrm{Lan}_J\,\sigma)_{T'\,X} \circ [\lambda g.\,\iota\,(\sigma_X\ \circ g)] \\ = & \mu'_X \circ [\lambda g.\,(\mathrm{Lan}_J\,\sigma)_{T'\,X} \circ \iota\,(\sigma_X\ \circ g)] \\ = & \mu'_X \circ [\lambda g.\,[\lambda g.\,\iota\,g \circ \sigma] \circ \iota\,(\sigma_X\ \circ g)] \\ = & \mu'_X \circ [\lambda g.\,\iota\,(\sigma_X \circ g) \circ \sigma] \\ = & [\lambda g.\,\mu'_X \circ \iota\,(\sigma_X \circ g) \circ \sigma] \\ = & [\lambda g.\,[\lambda g.\,g^{*'}] \circ \iota\,(\sigma_X \circ g) \circ \sigma] \\ = & [\lambda g.\,(\sigma_X \circ g)^{*'} \circ \sigma] \\ = & \qquad \{ \text{ by Kl. ext. pres. law of } \sigma \text{ as rel. mon. morphism } \} \\ & [\lambda g.\,\sigma_X \circ g^*] \\ = & \sigma_X \circ [\lambda g.\,g^*] \\ = & \sigma_X \circ \mu_X \end{array}$$

(2) The claim follows from the definitions of $(-)^*$, $(-)^{*'}$ from μ, μ', the unit and multiplication preservation of σ as a skew-monoid morphism and the laws of Lan_J.
(3) The claim follows from the mutual definitions of μ, μ' from $(-)^*$, $(-)^{*'}$ by the laws of Lan_J. ☐

We have seen that, in the presence of Lan_J, relative monads can be defined equivalently in the Kleisli extension and multiplication based formats. Restriction $(-)^\flat$ is defined for multiplication as follows. Given a monad (T,η,μ), with $\mu \in [\mathbb{C},\mathbb{C}](T\cdot T,T)$, we set $\mu^\flat =_{\mathrm{df}}$

$$\mathrm{Lan}_J\,(T\cdot J)\cdot T\cdot J \xrightarrow{\overline{\overline{\alpha}}_{T,J}\cdot T\cdot J} T\cdot \mathrm{Lan}_J\,J\cdot T\cdot J \xrightarrow{T\cdot\overline{\lambda}\cdot T\cdot J} T\cdot T\cdot J \xrightarrow{\mu\cdot J} T\cdot J$$

3.4. **An equivalent version of EM-algebras.** Just as the availability of $\mathrm{Lan}_J \in [\mathbb{J},\mathbb{C}] \to [\mathbb{C},\mathbb{C}]$ allows us to define relative monads based on μ rather than $(-)^*$, it also facilitates a more traditional-style definition of EM-algebras.

Definition 3.6. If $\mathrm{Lan}_J \in [\mathbb{J},\mathbb{C}] \to [\mathbb{C},\mathbb{C}]$ exists, an EM-algebra$^{\mathrm{alt}}$ of a relative monad T on J is given by an object $X \in |\mathbb{C}|$ and a map $x \in \mathbb{C}\,(\mathrm{Lan}_J\,T\,X, X)$, making the following diagrams commute in $\mathbb{C}$:

$$\begin{tikzcd}
\mathrm{Lan}_J\,J\,X \arrow[r, "\overline{\lambda}_X"] \arrow[d, "(\mathrm{Lan}_J\,\eta)_X"'] & X \arrow[d, equal] \\
\mathrm{Lan}_J\,T\,X \arrow[r, "x"'] & X
\end{tikzcd}
\qquad
\begin{tikzcd}
 & \mathrm{Lan}_J\,T\,(\mathrm{Lan}_J\,T\,X) \arrow[r, "\mathrm{Lan}_J\,T\,x"] & \mathrm{Lan}_J\,T\,X \arrow[dd, "x"] \\
\mathrm{Lan}_J\,(\mathrm{Lan}_J\,T \cdot T)\,X \arrow[ur, "(\overline{\alpha}_{T,T})_X"] \arrow[d, "(\mathrm{Lan}_J\,\mu)_X"'] & & \\
\mathrm{Lan}_J\,T\,X \arrow[rr, "x"'] & & X
\end{tikzcd}$$

An EM-algebras$^{\mathrm{alt}}$ map between (X, x), (Y, y) is a map $h \in \mathbb{C}\,(X, Y)$, making the following diagram commute in $\mathbb{C}$:

$$\begin{tikzcd}
\mathrm{Lan}_J\,T\,X \arrow[r, "\mathrm{Lan}_J\,T\,h"] \arrow[d, "x"'] & \mathrm{Lan}_J\,T\,Y \arrow[d, "x"] \\
X \arrow[r, "h"'] & Y
\end{tikzcd}$$

EM-algebra $^{\mathrm{alt}}$ and EM-algebra$^{\mathrm{alt}}$ maps of T form a category $\mathbf{EM}^{\mathrm{alt}}(T)$ that inherits its identities and composition from $\mathbb{C}$.

Theorem 3.7. *Assume that* $\mathrm{Lan}_J \in [\mathbb{J},\mathbb{C}] \to [\mathbb{C},\mathbb{C}]$ *exists. Consider a relative monad T on J.*

(1) *An EM-algebra (X, χ) gives rise to an EM-algebra$^{\mathrm{alt}}$ $(X, [\chi])$.*
(2) *An EM-algebra$^{\mathrm{alt}}$ (X, x) gives rise to an EM-algebra $(X, \lambda g.\ x \circ \iota\, g)$.*
(3) *This correspondence is a bijection.*
(4) *An EM-algebra map h between (X, χ), (Y, ψ) is also an EM-algebra$^{\mathrm{alt}}$ map between $(X, [\chi])$, $(Y, [\psi])$.*
(5) *An EM-algebra$^{\mathrm{alt}}$ map h between (X, x), (Y, y) is also an EM-algebra map between $(X, \lambda g.\ x \circ \iota\, g)$, $(Y, \lambda g.\ y \circ \iota\, g)$.*
(6) *The categories* $\mathbf{EM}(T)$ *and* $\mathbf{EM}^{\mathrm{alt}}(T)$ *are isomorphic.*

Proof. We only prove (1) and (4).

(1) The two EM$^{\mathrm{alt}}$-algebra laws of $(X, [\chi])$ are obtained from the definitions of $\overline{\lambda}$, μ and the laws of Lan_J with the help of the corresponding EM-algebra laws of (X, χ) as follows:

$$\begin{array}{rcl}
[\chi] \circ (\mathrm{Lan}_J\,\eta)_X & = & [\chi] \circ [\lambda g.\, \iota\, g \circ \eta] \\
& = & [\lambda g.\, [\chi] \circ \iota\, g \circ \eta] \\
& = & [\lambda g.\, \chi\, g \circ \eta] \\
& = & \qquad \{\text{ by 1st EM-algebra law }\} \\
& & [\lambda g.\, g] \\
& = & \overline{\lambda}_X
\end{array}$$

$$
\begin{array}{rcl}
[\chi] \circ (\mathrm{Lan}_J\, \mu)_X & = & [\chi] \circ [\lambda g.\, \iota\, g \circ \mu] \\
& = & [\lambda g.\, [\chi] \circ \iota\, g \circ \mu] \\
& = & [\lambda g.\, \chi\, g \circ \mu] \\
& = & [\lambda g.\, \chi\, g \circ [\lambda g'.\, g'^*]] \\
& = & [\lambda g.\, [\lambda g'.\, \chi\, g \circ g'^*]] \\
& = & \qquad \{ \text{ by 2nd EM-algebra law } \} \\
& & [\lambda g.\, [\lambda g'.\, \chi\, (\chi\, g \circ g')]] \\
& = & [\lambda g.\, [\lambda g'.\, [\chi] \circ \iota\, (\chi\, g \circ g')]] \\
& = & [\lambda g.\, [\chi] \circ [\lambda g'.\, \iota\, (\chi\, g \circ g')]] \\
& = & [\chi] \circ [\lambda g.\, [\lambda g'.\, \iota\, (\chi\, g \circ g')]] \\
& = & [\chi] \circ [\lambda g.\, [\lambda g'.\, \iota\, ([\chi] \circ \iota\, g \circ g')]] \\
& = & [\chi] \circ [\lambda g.\, [\lambda g'.\, [\lambda g.\, \iota\, ([\chi] \circ g)] \circ \iota\, (\iota\, g \circ g')]] \\
& = & [\chi] \circ [\lambda g.\, [\lambda g'.\, \mathrm{Lan}_J\, T\, [\chi] \circ \iota\, (\iota\, g \circ g')]] \\
& = & [\chi] \circ [\lambda g.\, \mathrm{Lan}_J\, T\, [\chi] \circ [\lambda g'.\, \iota\, (\iota\, g \circ g')]] \\
& = & [\chi] \circ \mathrm{Lan}_J\, T\, [\chi] \circ [\lambda g.\, [\lambda g'.\, \iota\, (\iota\, g \circ g')]] \\
& = & [\chi] \circ \mathrm{Lan}_J\, T\, [\chi] \circ (\overline{\alpha}_{T,T})_X
\end{array}
$$

(4) The $\mathrm{EM}^{\mathrm{alt}}$-algebra map law of h is obtained from the laws of Lan_J with the help of the EM-algebra map law of of h as follows:

$$
\begin{array}{rcl}
h \circ [\chi] & = & [\lambda g.\, h \circ \chi\, g] \\
& = & \qquad \{ \text{ by EM-algebra morphism law } \} \\
& & [\lambda g.\, \upsilon\, (h \circ g)] \\
& = & [\lambda g.\, [\upsilon] \circ \iota\, (h \circ g)] \\
& = & [\upsilon] \circ [\lambda g.\, \iota\, (h \circ g)] \\
& = & [\upsilon] \circ \mathrm{Lan}_J\, T\, h
\end{array}
$$

□

4. Well-behaved relative monads

It is somewhat unsatisfactory to obtain that $[\mathbb{J}, \mathbb{C}]$ is just skew-monoidal, rather than properly monoidal. This begs the question: would some conditions on J ensure a properly monoidal structure? The answer is yes. Mild conditions turn the skew-monoidal structure of $[\mathbb{J}, \mathbb{C}]$ into properly monoidal. What is more, the same conditions also allow relative monads on J to extend to monads on $\mathbb{C}$.

4.1. Well-behavedness conditions.

We define three well-behavedness conditions on J. They are additional to the existence of $\mathrm{Lan}_J \in [\mathbb{J}, \mathbb{C}] \to [\mathbb{C}, \mathbb{C}]$ and require the constituent maps of three canonical families, which are actually natural, to be isomorphisms.

Definition 4.1. $J \in \mathbb{J} \to \mathbb{C}$ is *well-behaved*, if not only does $\mathrm{Lan}_J \in [\mathbb{J}, \mathbb{C}] \to [\mathbb{C}, \mathbb{C}]$ exist, but also the following three conditions hold:

(1) J is *fully faithful*, i.e., for any $X, Y \in |\mathbb{J}|$, there is an inverse to the map

$$\begin{array}{l} J_{X,Y} \in \mathbb{J}\,(X,Y) \to \mathbb{C}\,(J\,X, J\,Y) \\ J_{X,Y}\,f =_{\mathrm{df}} J\,f \end{array}$$

(2) J is *dense*, i.e., for any $X, Y \in |\mathbb{C}|$, there is an inverse to the map

$$\begin{array}{l} K_{X,Y} \in \mathbb{C}\,(X,Y) \to [\mathbb{J}^{\mathrm{op}}, \mathbf{Set}]\,(\mathbb{C}\,(J\,-, X), \mathbb{C}\,(J\,-, Y)) \\ K_{X,Y}\,f =_{\mathrm{df}} \lambda g.\, f \circ g \end{array}$$

This is the same as to say that the associated *nerve* functor $K \in \mathbb{C} \to [\mathbb{J}^{\mathrm{op}}, \mathbf{Set}]$, defined by $K\,X =_{\mathrm{df}} \mathbb{C}\,(J\,-, X)$, is fully faithful.

(3) For any $F \in \mathbb{J} \to \mathbb{C}$, $X \in |\mathbb{J}|$, $Y \in |\mathbb{C}|$, there is an inverse to the map

$$\begin{array}{l} L^F_{X,Y} \in \mathrm{Lan}_J\,(\mathbb{C}\,(J\,X, F-))\,Y \to \mathbb{C}\,(J\,X, \mathrm{Lan}_J\,F\,Y) \\ L^F_{X,Y} =_{\mathrm{df}} [\lambda g.\, \lambda g'.\, \iota\, g \circ g'] \end{array}$$

This condition says that the nerve functor K preserves left Kan extensions of functors $F \in \mathbb{J} \to \mathbb{C}$ along J.

The functors $J_{\mathrm{f}} \in \mathbf{Fin} \to \mathbf{Set}$ and $J_{\mathbf{U}} \in \mathbf{U} \to \mathbf{Set}$ we have considered in our examples 1.1, 2.4 resp. 2.6 turn out to be well-behaved. This is a consequence of a general construction.

Let $\mathsf{U} \in |\mathbf{Set}|$ and $\mathsf{El} \in \mathsf{U} \to |\mathbf{Set}|$ be a type-theoretic universe (as in Example 2.6). As we already explained above, U and El define a category $\mathbf{U}$ by $|\mathbf{U}| =_{\mathrm{df}} \mathsf{U}$ and $\mathbf{U}\,(A,B) =_{\mathrm{df}} \mathsf{El}\,A \to \mathsf{El}\,B$ and a functor $J_{\mathbf{U}} \in \mathbf{U} \to \mathbf{Set}$ by $J_{\mathbf{U}}\,A =_{\mathrm{df}} \mathsf{El}\,A$ on objects and $J_{\mathbf{U}}\,f =_{\mathrm{df}} f$ on maps. In order for $J_{\mathbf{U}}$ to be well-behaved, it suffices that the universe has dependent products, i.e., that we have

$$\begin{array}{l} \mathrm{one} \in \mathsf{U} \\ \sigma \in \Pi A \in \mathsf{U}.\, (\mathsf{El}\,A \to \mathsf{U}) \to \mathsf{U} \end{array}$$

such that

$$\begin{array}{rcl} \mathsf{El}\,\mathrm{one} & = & 1 \\ \mathsf{El}\,(\sigma\,A\,B) & = & \Sigma a \in \mathsf{El}\,A.\, \mathsf{El}\,(B\,a) \end{array}$$

To prove this, we exploit the fact that (small) coends in $\mathbf{Set}$ can be constructed explicitly. Given any small category $\mathbb{J}$ and functors $J, F \in \mathbb{J} \to \mathbf{Set}$, for any $X \in |\mathbf{Set}|$, we have

$$\begin{array}{rcl} \mathrm{Lan}_J\,F\,X & = & \int^{Z \in |\mathbb{J}|} \mathbf{Set}\,(J\,Z, X) \bullet F\,Z \\ & \cong & (\Sigma Z \in |\mathbb{J}|.\, (J\,Z \to X) \times F\,Z)/{\sim_X} \end{array}$$

where $\sim_X$ is the least equivalence relation containing $(Z, g \circ J\,h, x) \sim_X (W, g, F\,h\,x)$ for any $g \in J\,W \to X, x \in F\,Z, h \in \mathbb{J}\,(Z,W)$. We can derive a specialized definition of L in this case:

$$\begin{array}{l} L^F_{X,Y} \in \mathrm{Lan}_J\,(J\,X \to F\,-)\,Y \to (J\,X \to \mathrm{Lan}_J\,F\,Y) \\ L^F_{X,Y}\,(Z, g, k) =_{\mathrm{df}} \lambda x.\, (Z, g, k\,x) \end{array}$$

We omit the verification that the equivalence is preserved in the definition of $L^F_{X,Y}$.

We can now show that inverses to J, K, L exist and hence $J_{\mathbf{U}}$ is well-behaved.

Theorem 4.2. *For any type-theoretic universe closed under dependent products, the functor* $J =_{\mathrm{df}} J_{\mathbf{U}} \in \mathbf{U} \to \mathbf{Set}$ *is well-behaved:*

(1) *For any* $A, B \in |\mathbf{U}|$, $J_{A,B}$ *has an inverse*

$$J_{A,B}{}^{-1} \in (\mathsf{El}\, A \to \mathsf{El}\, B) \to (\mathsf{El}\, A \to \mathsf{El}\, B)$$

(2) *For any* $X, Y \in |\mathbf{Set}|$, $K_{X,Y}$ *has an inverse*

$$K_{X,Y}{}^{-1} \in (\int_{C \in |\mathbf{U}|} (\mathsf{El}\, C \to X) \to (\mathsf{El}\, C \to Y)) \to (X \to Y)$$

(3) *For any* $F \in \mathbf{U} \to \mathbf{Set}$, $A \in |\mathbf{U}|$, $Y \in |\mathbf{Set}|$, $L^F_{A,Y}$ *has an inverse*

$$L^F_{A,Y}{}^{-1} \in (\mathsf{El}\, A \to \mathrm{Lan}_J\, F\, Y) \to \mathrm{Lan}_J\, (\mathsf{El}\, A \to F\, -)\, Y$$

Proof.

(1) This is obvious, since $J_{A,B}\, f = J\, f = f$.

(2) Given $\tau \in \int_{C \in |\mathbf{U}|} (\mathsf{El}\, C \to X) \to (\mathsf{El}\, C \to Y)$, we can construct $K^{-1}_{X,Y}\, \tau \in X \to Y$ as

$$K_{X,Y}{}^{-1}\, \tau =_{\mathrm{df}} \lambda x.\, \tau\, (\lambda z.\, x) *$$

where $*$ is the unique element of $\mathsf{El}\, \mathrm{one} = 1$. We verify that this is indeed the inverse:

$$\begin{array}{rcl}
K_{X,Y}{}^{-1}\, (K_{X,Y}\, f) & = & K_{X,Y}{}^{-1}\, (\lambda g.\, f \circ g) \\
& = & \lambda x.\, (f \circ (\lambda z.\, x))\, * \\
& = & \lambda x.\, f\, ((\lambda z.\, x) *) \\
& = & \lambda x.\, f\, x \\
& = & f
\end{array}$$

$$\begin{array}{rcl}
K_{X,Y}\, (K^{-1}{}_{X,Y}\, \tau) & = & K_{X,Y}\, (\lambda x'.\, \tau\, (\lambda z.\, x') *) \\
& = & \lambda g.\, (\lambda x'.\, \tau\, (\lambda z.\, x') *) \circ g \\
& = & \lambda g.\, \lambda x.\, (\lambda x'.\, \tau\, (\lambda z.\, x') *)\, (g\, x) \\
& = & \lambda g.\, \lambda x.\, \tau\, (\lambda z.\, g\, x)\, * \\
& = & \lambda g.\, \lambda x.\, \tau\, (g \circ (\lambda z.\, x))\, * \\
& = & \qquad \{ \text{ by naturality of } \tau \ \} \\
& & \lambda g.\, \lambda x.\, (\tau\, g \circ (\lambda z.\, x))\, * \\
& = & \lambda g.\, \lambda x.\, \tau\, g\, ((\lambda z.\, x) *) \\
& = & \lambda g.\, \lambda x.\, \tau\, g\, x \\
& = & \tau
\end{array}$$

(3) Given $f \in \mathsf{El}\, A \to \mathrm{Lan}_J\, F\, Y = \mathsf{El}\, A \to \Sigma C : \mathsf{U}.\, (\mathsf{El}\, C \to Y) \times F\, C$, we define $L^F_{A,Y}{}^{-1}\, f \in \mathrm{Lan}_J\, (\mathsf{El}\, A \to F\, -)\, Y = \Sigma C : \mathsf{U}.\, (\mathsf{El}\, C \to Y) \times (\mathsf{El}\, A \to F\, Y)$ by

$$L^F_{A,Y}{}^{-1}\, f =_{\mathrm{df}} (\sigma\, A\, f_0, \lambda(a, c).\, f_{10}\, a\, c, \lambda a.\, F\, (\lambda c.\, (a, c))\, (f_{11}\, a))$$

where $f_0 \in \mathsf{El}\, A \to \mathsf{U}$, $f_{10} \in \Pi a \in \mathsf{El}\, A.\, \mathsf{El}\, (f_0\, a) \to Y$ and $f_{11} \in \Pi a \in \mathsf{El}\, A.\, F\, (f_0\, a)$ are defined by $f_0 =_{\mathrm{df}} \pi_0 \circ f$, $f_{10} =_{\mathrm{df}} \pi_0 \circ \pi_1 \circ f$, $f_{11} =_{\mathrm{df}} \pi_1 \circ \pi_1 \circ f$.

We omit the verification that the equivalence relations are preserved. We show that $L^F_{A,Y}{}^{-1}$ is indeed the inverse of $L^F_{A,Y}$. To prove that it is a left inverse:

$$\begin{array}{rcl} {L^F_{A,Y}}^{-1}\,(L^F_{A,Y}\,(C,g,k)) & = & {L^F_{A,Y}}^{-1}(\lambda a.\,(C,g,k\,a)) \\ & = & (\sigma\,A\,(\lambda a.\,C), \lambda(a,c).\,g\,c, \lambda a.\,F\,(\lambda c.\,(a,c))\,(k\,a)) \\ & = & \quad \{(*)\} \\ & & (C,g,k) \end{array}$$

To establish (*), we use $\pi_1 \in \mathsf{El}\,(\sigma\,A\,(\lambda a.\,C)) \to \mathsf{El}\,C$, noting that

$$(\lambda(a,c).f\,c) = f \circ \pi_1$$

and

$$\begin{array}{rl} & (\mathsf{El}\,A \to F\,\pi_1)\,(\lambda a.\,F\,(\lambda c.\,(a,c))\,(k\,a)) \\ = & \lambda a.\,(F\,\pi_1 \circ F\,(\lambda c.\,(a,c)))\,(k\,a) \\ = & \lambda a.\,F\,\mathrm{id}\,(k\,a) \\ = & \lambda a.\,k\,a \\ = & k \end{array}$$

For the other direction:

$$\begin{array}{rcl} L^F_{A,Y}\,({L^F_{A,Y}}^{-1}\,f) & = & L^F_{A,Y}\,(\sigma\,A\,f_0, \lambda(a,c).\,f_{10}\,a\,c, \lambda a.\,F\,(\lambda c.\,(a,c))\,(f_{11}\,a)) \\ & = & \lambda a.\,(\sigma\,A\,f_0, \lambda(a,c).\,f_{10}\,a\,c, F\,(\lambda c.\,(a,c))\,(f_{11}\,a)) \\ & = & \quad \{(*)\} \\ & & \lambda a.\,(f_0\,a, f_{10}\,a, f_{11}\,a) \\ & = & f \end{array}$$

To justify (*), we exploit, for any $a \in \mathsf{El}\,A$, the function $\lambda c.\,(a,c) \in \mathsf{El}\,(f_0\,a) \to \mathsf{El}\,(\sigma\,A\,f_0)$.

□

Corollary 4.3. *The functor $J_\mathrm{f} \in \mathbf{Fin} \to \mathbf{Set}$ is well-behaved.*

Proof. Choose $\mathsf{U} =_\mathrm{df} \mathbb{N}$ and $\mathsf{El}\,n =_\mathrm{df} \underline{n}$. Clearly this universe contains 1 and is closed under Σ. □

From the well-behavedness of J_f, it follows that $[\mathbf{Fin}, \mathbf{Set}]$ is monoidal and Lam is a monoid. These facts were proved by Fiore et al. [14].

Our theorem is not general enough to show that $J_\mathrm{t} \in \mathbf{Fin} \downarrow \mathsf{Ty} \to [\mathsf{Ty}, \mathbf{Set}]$ from Example 2.5 is well-behaved, but it ought to be possible to generalize the construction beyond the case of $\mathbb{C} = \mathbf{Set}$.

4.2. $[\mathbb{J}, \mathbb{C}]$ **is monoidal.** Our well-behavedness conditions suffice to ensure that the unital and associativity laws of the skew-monoidal structure on $[\mathbb{J}, \mathbb{C}]$ are isomorphisms. Specifically, the existence of inverses of J, K, L ensures that $\rho, \overline{\lambda}, \overline{\alpha}$ (and consequently also λ, α) have inverses too.

Theorem 4.4. *If $J \in \mathbb{J} \to \mathbb{C}$ is well-behaved, then the category $([\mathbb{J}, \mathbb{C}], J, \cdot^J, \lambda, \rho, \alpha)$ is monoidal.*

Proof. To show that this category is monoidal, it suffices to show that $\rho, \overline{\lambda}, \overline{\alpha}$ have inverses.

(1) We define, for any $F \in \mathbb{J} \to \mathbb{C}$, $X \in |\mathbb{J}|$,

$$(\rho_F^{-1})_X \in \mathbb{C}\,(\mathrm{Lan}_J\, F\,(J\,X), F\,X)$$
$$(\rho_F^{-1})_X =_{\mathrm{df}} [\lambda g.\, F\,(J^{-1}\,g)]$$

We get

$$\begin{array}{rcl}(\rho_F^{-1})_X \circ (\rho_F)_X & = & [\lambda g.\, F\,(J^{-1}\,g)] \circ \iota_{F,J\,X}\,\mathrm{id}_{J\,X} \\ & = & F\,(J^{-1}\,\mathrm{id}_{J\,X}) \\ & = & F\,(J^{-1}\,(J\,\mathrm{id}_X)) \\ & = & F\,\mathrm{id}_X \\ & = & \mathrm{id}_{F\,X}\end{array}$$

and

$$\begin{array}{rcl}(\rho_F)_X \circ (\rho_F^{-1})_X & = & \iota_{F,J\,X}\,\mathrm{id}_{J\,X} \circ [\lambda g.\, F\,(J^{-1}\,g)] \\ & = & [\lambda g.\, \iota_{F,J\,X}\,\mathrm{id}_{J\,X} \circ F\,(J^{-1}\,g)] \\ & = & \qquad \{ \text{ by naturality of } \iota_{F,J\,X} \ \} \\ & & [\lambda g.\, \iota_{F,J\,X}\,(J\,(J^{-1}\,g))] \\ & = & [\iota_{F,J\,X}] \\ & = & \mathrm{id}_{\mathrm{Lan}_J\,F\ (J\,X)}\end{array}$$

by the definitions of ρ_F, ρ_F^{-1}, the laws of Lan_J and J^{-1} being inverse to J.

(2) We define, for any $F \in \mathbb{J} \to \mathbb{C}$, $X \in |\mathbb{J}|$,

$$\bar{\lambda}_X^{-1} \in \mathbb{C}\,(X, \mathrm{Lan}_J\, J\, X)$$
$$\bar{\lambda}_X^{-1} =_{\mathrm{df}} K^{-1}\,\iota_{J,X}$$

This gives

$$\begin{array}{rcl}\bar{\lambda}_X^{-1} \circ \overline{\lambda}_X & = & K^{-1}\,\iota_{J,F\,X} \circ [\lambda g.\, g] \\ & = & [\lambda g.\, K^{-1}\,\iota_{J,X} \circ g] \\ & = & [K\,(K^{-1}\,\iota_{J,X})] \\ & = & [\iota_{J,X}] \\ & = & \mathrm{id}_{\mathrm{Lan}_J\,J\,X}\end{array}$$

and

$$\begin{array}{rcl}\overline{\lambda}_X \circ \bar{\lambda}_X^{-1} & = & [\lambda g.\, g] \circ K^{-1}\,\iota_{J,X} \\ & = & \qquad \{ \text{ by naturality of } K^{-1} \ \} \\ & & K^{-1}\,(\lambda g.\, [\lambda g.\, g] \circ \iota_{J,X}\, g) \\ & = & K^{-1}\,(\lambda g.\, g) \\ & = & K^{-1}\,(K\,\mathrm{id}_X) \\ & = & \mathrm{id}_X\end{array}$$

by the definitions of $\overline{\lambda}$, $\bar{\lambda}^{-1}$, K, the laws of Lan_J and K^{-1} being inverse to K.

(3) We define, for any $F, G \in \mathbb{J} \to \mathbb{C}$, $X \in |\mathbb{J}|$,

$$(\bar{\alpha}_{F,G}^{-1})_X \in \mathbb{C}\,(\mathrm{Lan}_J\, F\,(\mathrm{Lan}_J\, G\, X), \mathrm{Lan}_J\,(\mathrm{Lan}_J\, F \cdot G)\, X)$$
$$(\bar{\alpha}_{F,G}^{-1})_X =_{\mathrm{df}} [\lambda g.\, [\lambda g.\, \lambda g'.\, \iota\, g \circ \iota\, g']\,(L^{-1}\,g)]$$

We first observe that

$$\begin{aligned}
(\overline{\alpha}_{F,G})_X &= [\lambda g.\, [\lambda g'.\, \iota\, (\iota\, g \circ g')]] \\
&= [\lambda g.\, [\lambda g'.\, \iota\, ((\lambda g'.\, \iota\, g \circ g')\, g')]] \\
&= [\lambda g.\, [\lambda g'.\, \iota\, (([\lambda g.\, \lambda g'.\, \iota\, g \circ g'] \circ \iota\, g)\, g')]] \\
&= [\lambda g.\, [\lambda g'.\, \iota\, ([\lambda g.\, \lambda g'.\, \iota\, g \circ g']\, (\iota\, g\, g')]] \\
&= [\lambda g.\, [\lambda g'.\, \iota\, (L\, (\iota\, g\, g')]]
\end{aligned}$$

by the definitions of $\overline{\alpha}_{F,G}$, L and the laws of Lan_J. This observation, together with the definitions of $\bar{\alpha}^{-1}_{F,G}$, the laws of Lan_J and L^{-1} being inverse to L, allows us to verify

$$\begin{aligned}
(\bar{\alpha}^{-1}_{F,G})_X \circ (\overline{\alpha}_{F,G})_X &= (\bar{\alpha}^{-1}_{F,G})_X \circ [\lambda g.\, [\lambda g'.\, \iota\, (L\, (\iota\, g\, g')]] \\
&= [\lambda g.\, (\bar{\alpha}^{-1}_{F,G})_X \circ [\lambda g'.\, \iota\, (L\, (\iota\, g\, g')]] \\
&= [\lambda g.\, [\lambda g'.\, (\bar{\alpha}^{-1}_{F,G})_X \circ \iota\, (L\, (\iota\, g\, g')]] \\
&= [\lambda g.\, [\lambda g'.\, [\lambda g.\, [\lambda g.\, \lambda g'.\, \iota\, g \circ \iota\, g']\, (L^{-1}\, g)] \circ \iota\, (L\, (\iota\, g\, g')]] \\
&= [\lambda g.\, [\lambda g'.\, [\lambda g.\, \lambda g'.\, \iota\, g \circ \iota\, g']\, (L^{-1}\, (L\, (\iota\, g\, g')))]] \\
&= [\lambda g.\, [\lambda g'.\, [\lambda g.\, \lambda g'.\, \iota\, g \circ \iota\, g']\, (\iota\, g\, g')]] \\
&= [\lambda g.\, [\lambda g'.\, ([\lambda g.\, \lambda g'.\, \iota\, g \circ \iota\, g'] \circ \iota\, g) g']] \\
&= [\lambda g.\, [\lambda g'.\, (\lambda g'.\, \iota\, g \circ \iota\, g') g']] \\
&= [\lambda g.\, [\lambda g'.\, \iota\, g \circ \iota\, g']] \\
&= [\lambda g.\, \iota\, g \circ [\lambda g'.\, \iota\, g']] \\
&= [\lambda g.\, \iota\, g] \\
&= \mathrm{id}_{\mathrm{Lan}_J\, (\mathrm{Lan}_J\, F \cdot G)\, X}
\end{aligned}$$

and

$$\begin{aligned}
(\overline{\alpha}_{F,G})_X \circ (\bar{\alpha}^{-1}_{F,G})_X &= (\overline{\alpha}_{F,G})_X \circ [\lambda g.\, [\lambda g.\, \lambda g'.\, \iota\, g \circ \iota\, g']\, (L^{-1}\, g)] \\
&= [\lambda g.\, (\overline{\alpha}_{F,G})_X \circ [\lambda g.\, \lambda g'.\, \iota\, g \circ \iota\, g']\, (L^{-1}\, g)] \\
&= \qquad \{ \text{ by definition of } \mathbf{Y} \text{ (Yoneda embedding) } \} \\
&\quad\; [\mathbf{Y}\, (\overline{\alpha}_{F,G})_X \circ [\lambda g.\, \lambda g'.\, \iota\, g \circ \iota\, g'] \circ L^{-1}] \\
&= [[\lambda g.\, \mathbf{Y}\, (\overline{\alpha}_{F,G})_X \circ (\lambda g'.\, \iota\, g \circ \iota\, g')] \circ L^{-1}] \\
&= \qquad \{ \text{ by definition of } \mathbf{Y} \text{ } \} \\
&\quad\; [[\lambda g.\, \lambda g'.\, (\overline{\alpha}_{F,G})_X \circ \iota\, g \circ \iota\, g'] \circ L^{-1}] \\
&= [[\lambda g.\, \lambda g'.\, [\lambda g.\, [\lambda g'.\, \iota\, (L\, (\iota\, g\, g')]] \circ \iota\, g \circ \iota\, g'] \circ L^{-1}] \\
&= [[\lambda g.\, \lambda g'.\, [\lambda g'.\, \iota\, (L\, (\iota\, g\, g')] \circ \iota\, g'] \circ L^{-1}] \\
&= [[\lambda g.\, \lambda g'.\, \iota\, (L\, (\iota\, g\, g')] \circ L^{-1}] \\
&= [[\lambda g.\, \iota \circ L \circ \iota\, g] \circ L^{-1}] \\
&= [\iota \circ L \circ [\iota] \circ L^{-1}] \\
&= [\iota \circ L \circ L^{-1}] \\
&= [\iota] \\
&= \mathrm{id}_{\mathrm{Lan}_J\, F\, (\mathrm{Lan}_J\, G\, X)}
\end{aligned}$$

□

As an immediate corollary, we get that, in the well-behaved case, relative monads are proper monoids in a properly monoidal structure.

Corollary 4.5. *If $J \in \mathbb{J} \to \mathbb{C}$ is well-behaved, then the category* $\mathbf{RMon}(J)$ *of relative monads on J is isomorphic to the category of monoids in the monoidal category* $([\mathbb{J}, \mathbb{C}], J, \cdot^J, \lambda, \rho, \alpha)$.

4.3. **Relative monads extend to monads.** As a pleasant bonus, the well-behavedness conditions also ensure that a relative monad extends to a monad. Crucial here is that, if J is well-behaved, then $\overline{\lambda}$ and $\overline{\alpha}$ are isomorphisms.

Theorem 4.6. *Assume that $J \in \mathbb{J} \to \mathbb{C}$ is well-behaved.*
A monoid (T, η, μ) in $[\mathbb{J}, \mathbb{C}]$ (equivalently, a relative monad on J) extends to a monoid $(T^\sharp, \eta^\sharp, \mu^\sharp)$ in $[\mathbb{C}, \mathbb{C}]$ (equivalently, a monad on $\mathbb{C}$), defined by

$$\begin{array}{lll}
T^\sharp & =_{\mathrm{df}} & \mathrm{Lan}_J\, T \\
\eta^\sharp & =_{\mathrm{df}} & I \xrightarrow{\bar{\lambda}^{-1}} \mathrm{Lan}_J\, J \xrightarrow{\mathrm{Lan}_J\, \eta} \mathrm{Lan}_J\, T \\
\mu^\sharp & =_{\mathrm{df}} & \mathrm{Lan}_J\, T \cdot \mathrm{Lan}_J\, T \xrightarrow{\bar{\alpha}^{-1}_{T,T}} \mathrm{Lan}_J\, (\mathrm{Lan}_J\, T \cdot T) \xrightarrow{\mathrm{Lan}_J\, \mu} \mathrm{Lan}_J\, T
\end{array}$$

Proof. We verify the three monad laws of $T^\sharp$ by the following diagrams using the respective relative monad laws of T, the fact that $\bar{\alpha}^{-1}$ is natural, and one the conditions (b'), (c') and (e') in each case.

$$\begin{tikzcd}
\mathrm{Lan}_J\, T \arrow[r, "\mathrm{Lan}_J\, T \cdot \bar{\lambda}^{-1}"'] \arrow[rr, bend left, "T^\sharp \cdot \eta^\sharp"] \arrow[ddr, equal] & \mathrm{Lan}_J\, T \cdot \mathrm{Lan}_J\, \eta \arrow[r, "\mathrm{Lan}_J\, T \cdot \mathrm{Lan}_J\, \eta"'] \arrow[d, "\bar{\alpha}^{-1}_{T,J}"] & \mathrm{Lan}_J\, T \cdot \mathrm{Lan}_J\, T \arrow[d, "\bar{\alpha}^{-1}_{T,T}"'] \arrow[ddd, bend left, "\mu^\sharp"] \\
& \mathrm{Lan}_J\, (\mathrm{Lan}_J\, T \cdot J) \arrow[r, "\mathrm{Lan}_J\, (\mathrm{Lan}_J\, T \cdot \eta)"] & \mathrm{Lan}_J\, (\mathrm{Lan}_J\, T \cdot T) \arrow[dd, "\mathrm{Lan}_J\, \mu"'] \\
& \mathrm{Lan}_J\, T \arrow[u, "\mathrm{Lan}_J\, \rho_T"'] \arrow[dr, equal] & \\
& & \mathrm{Lan}_J\, T
\end{tikzcd}$$

$$\begin{tikzcd}
\mathrm{Lan}_J\,T \arrow[d, "\bar{\lambda}^{-1}\cdot\mathrm{Lan}_J\,T"] \arrow[drr, equal] \arrow[dd, bend right=60, "\eta^\sharp\cdot T^\sharp"'] & & & \\
\mathrm{Lan}_J\,T\cdot\mathrm{Lan}_J\,T \arrow[r, "\bar{\alpha}^{-1}_{J,T}"'] \arrow[d, "\mathrm{Lan}_J\,\eta\cdot\mathrm{Lan}_J\,T"] & \mathrm{Lan}_J\,(\mathrm{Lan}_J\,J\cdot T) \arrow[r, "\mathrm{Lan}_J\,\lambda_T"'] \arrow[d, "\mathrm{Lan}_J\,(\eta\cdot\mathrm{Lan}_J\,T)"] & \mathrm{Lan}_J\,T \arrow[dr, equal] & \\
\mathrm{Lan}_J\,T\cdot\mathrm{Lan}_J\,T \arrow[r, "\bar{\alpha}^{-1}_{T,T}"] \arrow[rrr, bend right=20, "\mu^\sharp"'] & \mathrm{Lan}_J\,(\mathrm{Lan}_J\,T\cdot T) \arrow[rr, "\mathrm{Lan}_J\,\mu"] & & \mathrm{Lan}_J\,T
\end{tikzcd}$$

$$\begin{tikzcd}
\mathrm{Lan}_J\,T\cdot\mathrm{Lan}_J\,T\cdot\mathrm{Lan}_J\,T \arrow[rr, "\mathrm{Lan}_J\,T\cdot\bar{\alpha}^{-1}_{T,T}"'] \arrow[rrr, bend left=15, "T^\sharp\cdot\mu^\sharp"] \arrow[dd, "\bar{\alpha}^{-1}_{T,T}\cdot\mathrm{Lan}_J\,T"] \arrow[ddd, bend right=60, "\mu^\sharp\cdot T^\sharp"'] & & \mathrm{Lan}_J\,T\cdot\mathrm{Lan}_J\,(\mathrm{Lan}_J\,T\cdot T) \arrow[r, "\mathrm{Lan}_J\,T\cdot\mathrm{Lan}_J\,\mu"'] \arrow[d, "\bar{\alpha}^{-1}_{T,\mathrm{Lan}_J\,T\cdot T}"'] & \mathrm{Lan}_J\,T\cdot\mathrm{Lan}_J\,T \arrow[d, "\bar{\alpha}^{-1}_{T,T}"'] \arrow[ddd, bend left=60, "\mu^\sharp"] \\
& & \mathrm{Lan}_J\,(\mathrm{Lan}_J\,T\cdot\mathrm{Lan}_J\,T\cdot T) \arrow[r, "\mathrm{Lan}_J\,(\mathrm{Lan}_J\,T\cdot\mu)"'] & \mathrm{Lan}_J\,(\mathrm{Lan}_J\,T\cdot T) \arrow[dd, "\mathrm{Lan}_J\,\mu"'] \\
\mathrm{Lan}_J\,(\mathrm{Lan}_J\,T\cdot T)\cdot\mathrm{Lan}_J\,T \arrow[r, "\bar{\alpha}^{-1}_{\mathrm{Lan}_J\,T\cdot T,T}"] \arrow[d, "\mathrm{Lan}_J\,\mu\cdot\mathrm{Lan}_J\,T"] & \mathrm{Lan}_J\,(\mathrm{Lan}_J\,(\mathrm{Lan}_J\,T\cdot T)\cdot T) \arrow[ur, "\mathrm{Lan}_J\,\alpha_{T,T,T}"] \arrow[d, "\mathrm{Lan}_J\,(\mathrm{Lan}_J\,\mu\cdot T)"] & & \\
\mathrm{Lan}_J\,T\cdot\mathrm{Lan}_J\,T \arrow[r, "\bar{\alpha}^{-1}_{T,T}"] \arrow[rrr, bend right=15, "\mu^\sharp"'] & \mathrm{Lan}_J\,(\mathrm{Lan}_J\,T\cdot T) \arrow[rr, "\mathrm{Lan}_J\,\mu"] & & \mathrm{Lan}_J\,T
\end{tikzcd}$$

$\square$

Similarly, relative monad morphisms extend to monad morphisms.

Theorem 4.7. *Assume that $J \in \mathbb{J} \to \mathbb{C}$ is well-behaved.*

(1) *A morphism σ between relative monads T and T' on J extends to a morphism $\sigma^\sharp$ between monads $T^\sharp$ and $T'^\sharp$ on $\mathbb{C}$ via $\sigma^\sharp =_{\mathrm{df}} \mathrm{Lan}_J\,\sigma$.*

(2) *$(-)^\sharp$ is functorial.*

Proof.

(1) The monad morphism laws of $\sigma^\sharp$ are verified by the following diagrams from the relative monad morphism laws of σ and naturality of $\bar{\alpha}^{-1}$.

$$
\begin{tikzcd}
 & & \mathrm{Lan}_J\, T \arrow[dd, "\mathrm{Lan}_J\, \sigma"'] \arrow[dd, bend left, "\sigma^\sharp"] \\
I \arrow[r, "\bar{\lambda}^{-1}"] \arrow[urr, bend left, "\eta^\sharp"] \arrow[drr, bend right, "\eta'^\sharp"'] & \mathrm{Lan}_J\, J \arrow[ur, "\mathrm{Lan}_J\, \eta"] \arrow[dr, "\mathrm{Lan}_J\, \eta'"'] & \\
 & & \mathrm{Lan}_J\, T'
\end{tikzcd}
$$

$$
\begin{tikzcd}
\mathrm{Lan}_J\, T \cdot \mathrm{Lan}_J\, T \arrow[r, "\bar{\alpha}^{-1}_{T,T}"] \arrow[d, "\mathrm{Lan}_J\, \sigma \cdot \mathrm{Lan}_J\, \sigma"] \arrow[d, bend right, "\sigma^\sharp \cdot \sigma^\sharp"'] \arrow[rr, bend left, "\mu^\sharp"] & \mathrm{Lan}_J\, (\mathrm{Lan}_J\, T \cdot T) \arrow[r, "\mathrm{Lan}_J\, \mu"] \arrow[d, "\mathrm{Lan}_J\, (\mathrm{Lan}_J\, \sigma \cdot \sigma)"] & \mathrm{Lan}_J\, T \arrow[d, "\mathrm{Lan}_J\, \sigma"'] \arrow[d, bend left, "\sigma^\sharp"] \\
\mathrm{Lan}_J\, T' \cdot \mathrm{Lan}_J\, T' \arrow[r, "\bar{\alpha}^{-1}_{T',T'}"'] \arrow[rr, bend right, "\mu'^\sharp"'] & \mathrm{Lan}_J\, (\mathrm{Lan}_J\, T' \cdot T') \arrow[r, "\mathrm{Lan}_J\, \mu'"'] & \mathrm{Lan}_J\, T'
\end{tikzcd}
$$

(2) Functoriality of $(-)^\sharp$ is immediate from functoriality of $\mathrm{Lan}_J \in [\mathbb{J}, \mathbb{C}] \to [\mathbb{C}, \mathbb{C}]$, as $\mathbf{RMon}(\mathbb{J})$ and $\mathbf{Mon}(\mathbb{C})$ inherit their identities and composition from the corresponding functor categories $[\mathbb{J}, \mathbb{C}]$ and $[\mathbb{C}, \mathbb{C}]$. ☐

We have learned that, in the well-behaved case, not only do monads restrict to relative monads (by $(-)^\flat$), but relative monads extend to monads (by $(-)^\sharp$). This relationship turns out to be an adjunction: $(-)^\sharp$ is left adjoint to $(-)^\flat$. Furthermore, the adjunction is a coreflection, i.e., the unit is an isomorphism.

Theorem 4.8. *Assume that $J \in \mathbb{J} \to \mathbb{C}$ is well-behaved. Then $(-)^\sharp$ and $(-)^\flat$ form an adjunction between $\mathbf{RMon}(J)$ and $\mathbf{Mon}(\mathbb{C})$. Moreover, this adjunction is a coreflection.*

Proof. $\mathrm{Lan}_J \in [\mathbb{J}, \mathbb{C}] \to [\mathbb{C}, \mathbb{C}]$ is left adjoint to $(-) \cdot J \in [\mathbb{C}, \mathbb{C}] \to [\mathbb{J}, \mathbb{C}]$ with $\rho_T \in T \to \mathrm{Lan}_J\, T \cdot J$ (which is an isomorphism) as the unit on T and

$$\mathrm{Lan}_J\, (T \cdot J) \xrightarrow{\overline{\overline{\alpha}}_{T,J}} T \cdot \mathrm{Lan}_J\, J \xrightarrow{T \cdot \overline{\lambda}} T$$

as the counit.

Since the identities and composition of $\mathbf{RMon}(J)$ and $\mathbf{Mon}(\mathbb{C})$ are those of the functor categories $[\mathbb{J}, \mathbb{C}]$ and $[\mathbb{C}, \mathbb{C}]$, we only need to verify the unit and counit are a relative monad morphism and a monad morphism, respectively.

The relative monad morphism laws of ρ_T for a relative monad T are verified by the following diagrams from naturality of ρ and the properties (a), (b), (d) from Theorem 3.1.

$$
\begin{tikzcd}
 & & T \arrow[dd, "\rho_T"] \\
J \arrow[urr, "\eta"] \arrow[r, "\bar{\lambda}^{-1}\cdot J"'] \arrow[r, bend left, "\rho_J"] \arrow[drr, bend right, "\eta^\sharp\cdot J"] \arrow[drr, bend right=60, "(\eta^\sharp)^\flat"'] & \mathrm{Lan}_J\, J\cdot J \arrow[dr, "\mathrm{Lan}_J\, \eta\cdot J"] & \\
 & & \mathrm{Lan}_J\, T\cdot J
\end{tikzcd}
$$

$$
\begin{tikzcd}
\mathrm{Lan}_J\, T\cdot T \arrow[rrrr, "\mu"] \arrow[d, "\mathrm{Lan}_J\, T\cdot\rho_T"] \arrow[dd, bend right=60, "\mathrm{Lan}_J\, \rho_T\cdot\rho_T"'] \arrow[ddrrr, "\rho_{\mathrm{Lan}_J\, T\cdot T}"] & & & & T \arrow[dd, "\rho_T"] \\
\mathrm{Lan}_J\, T\cdot\mathrm{Lan}_J\, T\cdot J \arrow[d, "\mathrm{Lan}_J\, \rho_T\cdot\mathrm{Lan}_J\, T\cdot J"] \arrow[drr, equal] & & & & \\
\mathrm{Lan}_J\,(\mathrm{Lan}_J\, T\cdot J)\cdot\mathrm{Lan}_J\, T\cdot J \arrow[r, "\overline{\overline{\alpha}}_{T,J}\cdot\mathrm{Lan}_J\, T\cdot J"'] \arrow[rrrr, bend right=30, "(\mu^\sharp)^\flat"'] & \mathrm{Lan}_J\, T\cdot\mathrm{Lan}_J\, J\cdot\mathrm{Lan}_J\, T\cdot J \arrow[r, "\mathrm{Lan}_J\, T\cdot\overline{\lambda}\cdot\mathrm{Lan}_J\, T\cdot J"'] & \mathrm{Lan}_J\, T\cdot\mathrm{Lan}_J\, T\cdot J \arrow[r, "\bar{\alpha}^{-1}_{T,T}\cdot J"] \arrow[rr, bend right, "\mu^\sharp\cdot J"'] & \mathrm{Lan}_J\,(\mathrm{Lan}_J\, T\cdot T)\cdot J \arrow[r, "\mathrm{Lan}_J\, \mu\cdot J"] & \mathrm{Lan}_J\, T\cdot J
\end{tikzcd}
$$

The monad morphism laws of $(T\cdot\overline{\lambda})\circ\overline{\overline{\alpha}}_{T,J}$ for a monad T are verified from naturality of η resp. μ, naturality of $\overline{\lambda}$ and $\overline{\overline{\alpha}}$, and from two elementary properties of $\overline{\overline{\alpha}}$, namely that $\overline{\overline{\alpha}}_{I,H} = \mathrm{Lan}_J\, H$ and $\overline{\overline{\alpha}}_{F\cdot G,H} = (\mathrm{Lan}_J\, F\cdot\overline{\overline{\alpha}}_{G,H})\circ\overline{\overline{\alpha}}_{F,G\cdot H}$.

$$
\begin{tikzcd}
 & & \mathrm{Lan}_J\, J \arrow[r, "\mathrm{Lan}_J\,(\eta\cdot J)"'] \arrow[r, bend left, "\mathrm{Lan}_J\, \eta^\flat"] \arrow[dl, "\overline{\overline{\alpha}}_{I,J}"] \arrow[dl, equal, bend right] & \mathrm{Lan}_J\,(T\cdot J) \arrow[d, "\overline{\overline{\alpha}}_{T,J}"] \\
 & \mathrm{Lan}_J\, J \arrow[rr, "\eta\cdot\mathrm{Lan}_J\, J"] \arrow[dl, "\overline{\lambda}"] & & T\cdot\mathrm{Lan}_J\, J \arrow[d, "T\cdot\overline{\lambda}"] \\
I \arrow[ur, bend left, "\bar{\lambda}^{-1}"] \arrow[rrr, "\eta"'] \arrow[uurrr, bend left=60, "(\eta^\flat)^\sharp"] & & & T
\end{tikzcd}
$$

$$
\begin{tikzcd}[column sep=small]
\mathrm{Lan}_J\,(T\cdot J)\cdot \mathrm{Lan}_J\,(T\cdot J) \arrow[r, "\bar{\alpha}^{-1}_{T\cdot J, T\cdot J}"] \arrow[d, "\overline{\overline{\alpha}}_{T,J}\cdot \mathrm{Lan}_J\,(T\cdot J)"] \arrow[dr, "\mathrm{Lan}_J\,(T\cdot J)\cdot\overline{\overline{\alpha}}_{T,J}"] \arrow[rrrr, bend left, "(\mu^\flat)^\sharp"] & \mathrm{Lan}_J\,(\mathrm{Lan}_J\,(T\cdot J)\cdot T\cdot J) \arrow[r, "\mathrm{Lan}_J\,(\overline{\overline{\alpha}}_{T,J}\cdot T\cdot J)"'] \arrow[d, "\overline{\overline{\alpha}}_{\mathrm{Lan}_J\,(T\cdot J)\cdot T, J}"] \arrow[rrr, bend left, "\mathrm{Lan}_J\,\mu^\flat"] & \mathrm{Lan}_J\,(T\cdot \mathrm{Lan}_J\cdot T\cdot J) \arrow[r, "\mathrm{Lan}_J\,(T\cdot\overline{\lambda}\cdot T\cdot J)"'] & \mathrm{Lan}_J\,(T\cdot T\cdot J) \arrow[r, "\mathrm{Lan}_J\,(\mu\cdot J)"'] & \mathrm{Lan}_J\,(T\cdot J) \arrow[ddd, "\overline{\overline{\alpha}}_{T,J}"] \\
T\cdot \mathrm{Lan}_J\,J\cdot \mathrm{Lan}_J\,(T\cdot J) \arrow[d, "T\cdot\overline{\lambda}\cdot \mathrm{Lan}_J\,(T\cdot J)"] & \mathrm{Lan}_J\,(T\cdot J)\cdot T\cdot \mathrm{Lan}_J\,J \arrow[d, "\overline{\overline{\alpha}}_{T,J}\cdot T\cdot \mathrm{Lan}_J\,J"] & & & \\
T\cdot \mathrm{Lan}_J\,(T\cdot J) \arrow[d, "T\cdot\overline{\overline{\alpha}}_{T,J}"] & T\cdot \mathrm{Lan}_J\,J\cdot T\cdot \mathrm{Lan}_J\,J \arrow[dl, "T\cdot\overline{\lambda}\cdot T\cdot \mathrm{Lan}_J\,J"] & & & \\
T\cdot T\cdot \mathrm{Lan}_J\,J \arrow[rrrr, "\mu\cdot \mathrm{Lan}_J\,J"] \arrow[d, "T\cdot T\cdot\overline{\lambda}"] & & & & T\cdot \mathrm{Lan}_J\,J \arrow[d, "T\cdot\overline{\lambda}"] \\
T\cdot T \arrow[rrrr, "\mu"] & & & & T
\end{tikzcd}
$$

□

We see that, once the extension of relative monads to monads is definable (which takes that J is well-behaved), it has very good properties and this happens because the adjunction $\mathrm{Lan}_J \dashv - \cdot J$ between $[\mathbb{J}, \mathbb{C}]$ and $[\mathbb{C}, \mathbb{C}]$—the defining adjunction of Lan_J— then lifts from functors to (relative) monads.

Unlike the unit, the counit of this adjunction is generally not an isomorphism, so the adjunction is not a reflection. For example, for $\mathbb{C} =_{\mathrm{df}} \mathbf{Set}$, $\mathbb{J} =_{\mathrm{df}} \mathbf{Fin}$, $J =_{\mathrm{df}} J_{\mathrm{f}}$, the T-component of the counit is an isomorphism if and only if the monad T is finitary. This is important for us: the categories of monads on $\mathbb{C}$ and relative monads on J are generally not equivalent.

Example 4.9. For the powerset monad $\mathcal{P}$ on **Set**, we have that $\mathcal{P}\,X$ is the powerset of a set X, $\mathcal{P}^\flat\,X =_{\mathrm{df}} \mathcal{P}\,(J_{\mathrm{f}}\,X)$ is the powerset of a finite set X, and $\mathcal{P}^{\flat\#}\,X =_{\mathrm{df}} \mathrm{Lan}_{J_{\mathrm{f}}}\,\mathcal{P}^{\#}\,X$ is the finitary powerset (the set of finite subsets) of a (possibly infinite) set X. The difference between $\mathcal{P}$ and $\mathcal{P}^{\flat\#}$ arises because $\mathcal{P}$ is not finitary.

Example 4.10. For the relative monad Vec on J_{f}, $\mathsf{Vec}^{\#}\,X$ is the space of vectors over a possibly infinite coordinate system X that may only have finitely many non-zero components.

Example 4.11. For the relative monad Lam on J_{f}, we have that $\mathsf{Lam}\,X$ is the set of λ-terms over a finite, nameless context X and $\mathsf{Lam}^{\#}\,X$ is given by the set of λ-terms over a possibly infinite, name-carrying context X. The functor $\mathsf{Lam}^{\#}$ is the carrier of the initial algebra of the functor $F \in [\mathbf{Set}, \mathbf{Set}] \to [\mathbf{Set}, \mathbf{Set}]$ defined by $F\,G\,X =_{\mathrm{df}} X + G\,X \times G\,X + G\,(1 + X)$.

For the relative monad Lam^∞ the picture is different. $\mathsf{Lam}^\infty\,X$ is the set of non-wellfounded λ-terms over a finite, nameless context, but $\mathsf{Lam}^{\infty\#}\,X$ is the set of non-wellfounded λ-terms using a finite number of variables from a possibly infinite, name-carrying context. This differs from the non-finitary carrier of the final coalgebra of F, capturing general non-wellfounded λ-terms that may use infinitely many variables.

The special case where the T-component of the counit of $(-)^{\#} \dashv (-)^\flat$ is an isomorphism (i.e., $(T^\flat)^\sharp \cong T$) corresponds to the notion of monad with arities of Berger et al. [10]. A monad on a category $\mathbb{C}$ with a dense subcategory $\mathbb{J}$ (included in $\mathbb{C}$ via $J \in \mathbb{J} \to \mathbb{C}$) is a monad with arities if $(T^\flat)^\sharp \cong T$ and if the nerve functor K corresponding to J preserves $\mathrm{Lan}_J\, T^\flat$ (see [16]). We can see that Berger et al. work under our well-formedness conditions, except that the third condition is only required of $T^\flat$. In this situation, the associativity law α of the skew-monoidal category $[\mathbb{J}, \mathbb{C}]$ need not be an isomorphism, but the component $\alpha_{T^\flat, T^\flat, T^\flat}$ is.

4.4. **Kleisli and Eilenberg-Moore constructions and extension.** We now explore the relationship between the Kleisli and Eilenberg-Moore constructions of a given relative monad T on J and the monad $T^\sharp$ on $\mathbb{C}$.

We assume that Lan_J exists, that J is dense and satisfies the 3rd well-behavedness condition (so that λ and α have inverses—only then is $T^\sharp$ defined) and optionally also that J is fully-faithful (so that ρ has also an inverse and $(T^\sharp)^\flat \cong T$).

There is a functor $D \in \mathbf{Kl}(T) \to \mathbf{Kl}(T^\sharp)$ defined by

- for any $X \in |\mathbb{J}|$, $D\, X =_{\mathrm{df}} J\, X$,
- for any $X, Y \in |\mathbb{J}|$, $k \in \mathbb{C}(J\, X, T\, Y)$, $D\, k =_{\mathrm{df}} \; J\, X \xrightarrow{k} T\, Y \xrightarrow{(\rho_T)_Y} \mathrm{Lan}_J\, T\, (J\, Y)$.

To prove that D preserves the identities and composition of $\mathbf{Kl}(T)$, the laws of the monoidal structure on $[\mathbb{J}, \mathbb{C}]$ must be invoked.

Let L, R be the Kleisli relative adjunction of T, which is given by $L\, X =_{\mathrm{df}} X$, $L\, f =_{\mathrm{df}} \eta \circ J\, f$, $R\, X =_{\mathrm{df}} T\, X$, $R\, k =_{\mathrm{df}} k^*$.

The Kleisli adjunction of $T^\sharp$ is given by $L' X =_{\mathrm{df}} X$, $L' f =_{\mathrm{df}} \eta^\sharp \circ f =_{\mathrm{df}} \mathrm{Lan}_J\, \eta \circ \bar{\lambda}^{-1} \circ f$, $R'\, X =_{\mathrm{df}} T^\sharp X = \mathrm{Lan}_J\, T\, X$, $R'\, k =_{\mathrm{df}} k^{(*^\sharp)} = \mu^\sharp \circ T^\sharp\, k = \mathrm{Lan}_J\, \mu \circ \bar{\alpha}^{-1}_{T,T} \circ \mathrm{Lan}_J\, T\, k$

We have $D \cdot L = L' \cdot J$ and $R = R' \cdot D$. As soon as $J \in \mathbb{J} \to \mathbb{C}$ is fully faithful (so that ρ also has an inverse), D (whose action on objects is J) is fully faithful too. Moreover, under the same condition, T splits through $\mathbf{Kl}(T^\sharp)$ via $L' \cdot J$ and R': we have $R' \cdot (L' \cdot J) = \mathrm{Lan}_J\, T \cdot J \cong T$ and $L' \cdot J$ is relative left adjoint to R'.

No functor is generally definable in the opposite direction $\mathbf{Kl}(T^\sharp) \to \mathbf{Kl}(T)$.

There is a functor $E \in \mathbf{EM}(T^\sharp) \to \mathbf{EM}(T)$, given by

- for any $(X, x) \in \mathbf{EM}(T^\sharp)$, i.e., $X \in |\mathbb{C}|$, $x \in \mathbb{C}(\mathrm{Lan}_J T X, X)$, subject to EM-algebra conditions, $E\,(X, x) =_{\mathrm{df}} (X, \chi)$ where, for $Z \in |\mathbb{J}|$, $g \in \mathbb{C}(J\, Z, X)$, $\chi_Z\, g =_{\mathrm{df}} x \circ \iota\, g \in \mathbb{C}(T\, Z, X)$; $E\,(X, x)$ is a relative EM-algebra for T.
- for any $h \in \mathbf{EM}(T^\sharp)((X, x), (Y, y))$, which is a map in $\mathbb{C}(X, Y)$ satisfying the EM-algebra map conditions, $E\, h =_{\mathrm{df}} h$, satisfying the relative EM-algebra map conditions.

There is also a functor $E^{-1} \in \mathbf{EM}(T) \to \mathbf{EM}(T^\sharp)$ in the opposite direction, given by

- for any $(X, \chi) \in \mathbf{EM}(T)$, i.e., $X \in |\mathbb{C}|$, for any $Z \in |\mathbb{J}|$, $\chi \in \mathbb{C}(J\, Z, X) \to \mathbb{C}(T\, Z, X)$, subject to the relative EM-algebra conditions, $E^{-1}\,(X, \chi) =_{\mathrm{df}} (X, x)$ where $x = [\chi] \in \mathbb{C}(\mathrm{Lan}_J\, T\, X, X)$; $E^{-1}\,(X, \chi)$ is an EM-algebra for $T^\sharp$;
- for any $h \in \mathbf{EM}(T)((X, x), (Y, y))$, which is a map in $\mathbb{C}(X, Y)$ satisfying the relative EM-algebra map conditions, $E^{-1}\, h =_{\mathrm{df}} h$, which satisfies the EM-algebra map conditions.

That the identities and composition are preserved is trivial for both E and E^{-1}.

E and E^{-1} are each other's inverses, i.e., the EM-algebras of $T^\sharp$ and T are the same thing: $E^{-1}\,(E\, x) = [\lambda g.\, x \circ \iota\, g] = x \circ [\iota] = x$ and $E\,(E^{-1}\, \chi) = \lambda g.\, [\chi] \circ \iota\, g = \lambda g.\, \chi\, g = \chi$.

We arrive at the following picture:

$$
\begin{tikzcd}
 & \mathbf{Kl}(T^\sharp) \arrow[r, dotted] \arrow[dr, bend left, "R'"] & \mathbf{EM}(T^\sharp) \arrow[dr, bend left, "E", "\text{iso}"'] \arrow[d, bend left, "U'"] & \\
\mathbf{Kl}(T) \arrow[ur, bend left, "D", "\text{f-f}"'] \arrow[drr, "R"] & & \mathbb{C} \arrow[ul, bend left, "L'"] \arrow[u, bend left, "F'"] & \mathbf{EM}(T) \arrow[dl, "U"] \\
 & \mathbb{J} \arrow[ul, "L"] \arrow[r, "J"'] \arrow[urr, "F"] & \mathbb{C} &
\end{tikzcd}
$$

5. Arrows as a special case of relative monads

We now turn to a whole class of examples, Hughes' arrows [20]. As we shall see, arrows are relative monads on the Yoneda embedding. Arrows are commonly perceived as a generalization of monads. With relative monads, this relationship is turned upside down![3]

The rigorous definition of arrows by Heunen and Jacobs [18] is as follows.[4]

Definition 5.1. A (**Set**-valued) *arrow* on a category $\mathbb{J}$ is given by

- a function $R \in |\mathbb{J}| \times |\mathbb{J}| \to |\mathbf{Set}|$,
- for any $X, Y \in |\mathbb{J}|$, a function $\mathsf{pure} \in \mathbb{J}\,(X,Y) \to R\,(X,Y)$,
- for any $X, Y, Z \in |\mathbb{J}|$, a function $\lll\, \in R\,(Y,Z) \times R\,(X,Y) \to R\,(X,Z)$,

satisfying the conditions

- $\mathsf{pure}\,(g \circ f) = \mathsf{pure}\,g \circ \mathsf{pure}\,f$,
- $s \lll \mathsf{pure}\,\mathrm{id} = s$,
- $\mathsf{pure}\,\mathrm{id} \lll r = r$,
- $t \lll (s \lll r) = (t \lll s) \lll r$.

It follows from the conditions that R is functorial (contravariantly in the first argument), i.e., $R \in \mathbb{J}^{\mathrm{op}} \times \mathbb{J} \to \mathbf{Set}$, which is the same as to say that R is an endoprofunctor on $\mathbb{J}$, and pure and $\lll$ are natural/dinatural.

A monad $(T, \eta, (-)^*)$ on $\mathbb{J}$ defines an arrow $(R, \mathsf{pure}, \lll)$ on $\mathbb{J}$ by $R\,(X,Y) =_{\mathrm{df}} \mathbf{Kl}(T)\,(X,Y)$, $\mathsf{pure}\,f =_{\mathrm{df}} L\,f$ and $\ell \lll k =_{\mathrm{df}} \ell \circ^T k$ where L is the left adjoint in the Kleisli adjunction and $\circ^T$ is the Kleisli composition.

We show now that an arrow on $\mathbb{J}$ is the same thing as a relative monad on the Yoneda embedding $\mathbf{Y} \in \mathbb{J} \to [\mathbb{J}^{\mathrm{op}}, \mathbf{Set}]$ defined by $\mathbf{Y}\,X\,Y =_{\mathrm{df}} \mathbb{J}\,(Y, X)$.

By definition, a relative monad on $\mathbf{Y}$ is given by

- a function $T \in |\mathbb{J}| \to |[\mathbb{J}^{\mathrm{op}}, \mathbf{Set}]|$,
- for any $X \in |\mathbb{J}|$, a map $\eta_X \in [\mathbb{J}^{\mathrm{op}}, \mathbf{Set}]\,(\mathbf{Y}\,X, T\,X)$,
- for any $X, Y \in |\mathbb{J}|$, a map function $(-)^* \in [\mathbb{J}^{\mathrm{op}}, \mathbf{Set}]\,(\mathbf{Y}\,X, T\,Y) \to [\mathbb{J}^{\mathrm{op}}, \mathbf{Set}]\,(T\,X, T\,Y)$

satisfying three coherence conditions.

[3] Since we compare arrows to monads, not strong monads, by arrows we mean arrows without strength in this paper. This said, our results scale also to strong arrows, but this remains outside the scope of this paper. We have proved this elsewhere [34]. Heunen and Jacobs considered strong arrows; their analysis of strength was elaborated by Asada [9].

[4] In agreement with the previous footnote, this definition does not require $\mathbb{J}$ to be symmetric monoidal and an arrow to come with a first operation (strength).

Theorem 5.2.

(1) *An arrow* $(R, \mathsf{pure}, \lll)$ *on* $\mathbb{J}$ *gives rise to a relative monad* $(T, \eta, (-)^*)$ *on* $\mathbf{Y}$ *defined by* $T\,X\,Y =_{\mathrm{df}} R\,(Y, X)$, $T\,\_\,f\,r =_{\mathrm{df}} r \lll \mathsf{pure}\,f$, $\eta\,f =_{\mathrm{df}} \mathsf{pure}\,f$, $k^*\,r =_{\mathrm{df}} k\,\mathrm{id} \lll r$.
(2) *A relative monad* $(T, \eta, (-)^*)$ *on* $\mathbf{Y}$ *gives rise to an arrow* $(R, \mathsf{pure}, \lll)$ *on* $\mathbb{J}$ *defined by* $R\,(X, Y) =_{\mathrm{df}} T\,Y\,X$, $\mathsf{pure}\,f =_{\mathrm{df}} \eta\,f$, $s \lll r =_{\mathrm{df}} (\lambda f.\, T\,\_\,f\;\;s)^* r$. *(The last item is well-defined, as* $\lambda f.\, T\,\_\,f\;\;s$ *is natural.)*
(3) *The above is a bijective correspondence.*

Proof.

(1) We have to verify functoriality of T and naturality of η, $(-)^*$ in their contravariant arguments and the three relative monad laws. The proofs are as follows.

Proofs of contravariant functoriality of T:

$$
\begin{array}{rcl}
T\,\_\,\mathrm{id}\,r & = & r \lll \mathsf{pure}\,\mathrm{id} \\
& = & \qquad \{ \text{ by 2nd arrow law } \} \\
& & r
\end{array}
$$

$$
\begin{array}{rcl}
T\,\_\,(g \circ f)\,r & = & r \lll \mathsf{pure}\,(g \circ f) \\
& = & \qquad \{ \text{ by 1st arrow law } \} \\
& & r \lll (\mathsf{pure}\,g \lll \mathsf{pure}\,f) \\
& = & \qquad \{ \text{ by 4th arrow law } \} \\
& & (r \lll \mathsf{pure}\,g) \lll \mathsf{pure}\,f \\
& = & T\,\_\,f\,(r \lll \mathsf{pure}\,g) \\
& = & T\,\_\,f\,(T\,\_\,g\,r) \\
& = & (T\,\_\,f \;\circ T\,\_\,g)\,r
\end{array}
$$

Proofs of contravariant naturality of η and $(-)^*$:

$$
\begin{array}{rcl}
\eta\,(g \circ f) & = & \mathsf{pure}\,(g \circ f) \\
& = & \qquad \{ \text{ by 1st arrow law } \} \\
& & \mathsf{pure}\,g \lll \mathsf{pure}\,f \\
& = & T\,\_\,f\,(\eta\,g)
\end{array}
$$

$$
\begin{array}{rcl}
k^*\,(T\,\_\,f\,r) & = & k\,\mathrm{id} \lll (r \lll \mathsf{pure}\,f) \\
& = & \qquad \{ \text{ by 4th arrow law } \} \\
& & (k\,\mathrm{id} \lll r) \lll \mathsf{pure}\,f \\
& = & k^*\,r \lll \mathsf{pure} f \\
& = & T\,\_\,f\,(k^*\,r)
\end{array}
$$

Proofs of relative monad laws:

$$\begin{array}{rcl}
(k^* \circ \eta)\, f & = & k^*(\eta\, f) \\
& = & k\, \mathrm{id} \lll \eta\, f \\
& = & k\, \mathrm{id} \lll \mathsf{pure}\, f \\
& = & T\, _\, f\, (k\, \mathrm{id}) \\
& = & \qquad \{ \text{ by contravar. naturality of } k \text{ } \} \\
& & k\, (\mathrm{id} \circ f) \\
& = & k\, f
\end{array}$$

$$\begin{array}{rcl}
\eta^*\, r & = & \mathsf{pure}^*\, r \\
& = & \mathsf{pure}\, \mathrm{id} \lll r \\
& = & \qquad \{ \text{ by 3rd arrow law } \} \\
& & r
\end{array}$$

$$\begin{array}{rcl}
(\ell^* \circ k)^*\, r & = & (\ell^* \circ k)\, \mathrm{id} \lll r \\
& = & (\ell^*\, (k\, \mathrm{id})) \lll r \\
& = & (\ell\, \mathrm{id} \lll k\, \mathrm{id}) \lll r \\
& = & \quad \{ \text{ by 4th arrow law } \} \\
& & \ell\, \mathrm{id} \lll (k\, \mathrm{id} \lll r) \\
& = & \ell^*\, (k\, \mathrm{id} \lll r) \\
& = & \ell^*\, (k^*\, r) \\
& = & (\ell^* \circ k^*)\, r
\end{array}$$

(2) To see that the definition of $\lll$ is wellformed, we must check that $\lambda f.\, T\, \_\, f\, s$ is natural in the contravariant argument, which it is.

We can verify all four arrow laws.

$$\begin{array}{rcl}
\mathsf{pure}\, g \lll \mathsf{pure}\, f & = & \eta\, g \lll \eta\, f \\
& = & (\lambda f'.\, T\, _\, f'\, (\eta\, g))^*(\eta\, f) \\
& & \qquad \{ \text{ by contravar. naturality of } \eta \text{ } \} \\
& = & (\lambda f'.\, \eta\, (g \circ f'))^*(\eta\, f) \\
& = & ((\lambda f'.\, \eta\, (g \circ f'))^* \circ \eta)\, f \\
& = & \qquad \{ \text{ by 1st relative monad law } \} \\
& & (\lambda f'.\, \eta\, (g \circ f'))\, f \\
& = & \eta\, (g \circ f) \\
& = & \mathsf{pure}\, (g \circ f)
\end{array}$$

$$
\begin{array}{rcl}
r \lll \mathsf{pure}\,\mathrm{id} & = & (\lambda f.\, T\,\_\,f\,r)^*(\eta\,\mathrm{id}) \\
& = & ((\lambda f.\, T\,\_\,f\,r)^* \circ \eta)\,\mathrm{id} \\
& = & \qquad \{ \text{ by 1st relative monad law } \} \\
& & T\,\_\,\mathrm{id}\,r \\
& = & \qquad \{ \text{ by contravar. functoriality of } T \text{ } \} \\
& & r
\end{array}
$$

$$
\begin{array}{rcl}
\mathsf{pure}\,\mathrm{id} \lll r & = & (\lambda f.\, T\,\_\,f\,(\eta\,\mathrm{id}))^*\,r \\
& = & \qquad \{ \text{ by contravar. naturality of } \eta \text{ } \} \\
& & (\lambda f.\, \eta\,(\mathrm{id} \circ f))^*\,r \\
& = & \eta^*\,r \\
& = & \qquad \{ \text{ by 2nd relative monad law } \} \\
& & r
\end{array}
$$

$$
\begin{array}{rcl}
(t \lll s) \lll r & = & (\lambda f.\, T\,\_\,f\,t)^*\,s) \lll r \\
& = & (\lambda f.\, T\,\_\,f\,((\lambda f.\, T\,\_\,f\,t)^*\,s))^*\,r \\
& = & \qquad \{ \text{ by contravar. naturality of } (-)^* \text{ } \} \\
& & (\lambda f.\, (\lambda f.\, T\,\_\,f\,t)^*\,(T\,\_\,f\,s))^*\,r \\
& = & (\lambda f.\, (\lambda f.\, T\,\_\,f\,t)^*\,((\lambda f.\, T\,\_\,f\,s)\,f))^*\,r \\
& = & ((\lambda f.\, T\,\_\,f\,t)^* \circ (\lambda f.\, T\,\_\,f\,s))^*\,r \\
& = & \qquad \{ \text{ by 3rd relative monad law } \} \\
& = & ((\lambda f.\, T\,\_\,f\,t)^* \circ (\lambda f.\, T\,\_\,f\,s)^*)\,r \\
& = & (\lambda f.\, T\,\_\,f\,t)^*((\lambda f.\, T\,\_\,f\,s)^*\,r) \\
& = & t \lll (\lambda f.\, T\,\_\,f\,s)^*\,r \\
& = & t \lll (s \lll r)
\end{array}
$$

The conditions for the bijection (3) just follow from the respective relative monad and arrow laws except in the case of $k^* r$ where we must use also invoke the naturality of k. □

The bijection extends to an isomorphism of the categories of arrows on $\mathbb{J}$ and relative monads on $\mathbf{Y}$.

Definition 5.3. A arrow morphism between arrows $(R, \mathsf{pure}, \lll)$ and $(R', \mathsf{pure}', \lll')$ is given by

- a function $\tau_{X,Y} \in R\,(X,Y) \to R'\,(X,Y)$

satisfying the conditions

- $\tau\,(\mathsf{pure}\,f) = \mathsf{pure}'\,f$,
- $\tau\,(f \lll g) = \tau\,f \lll' \tau\,g$.

Theorem 5.4.

(1) *An arrow morphism τ between arrows $(R, \mathsf{pure}, \lll)$ and $(R', \mathsf{pure}', \lll')$ on $\mathbb{J}$ gives rise to a relative monad morphism $\sigma_X \in [\mathbb{J}^{\mathrm{op}}, \mathbf{Set}]\,(T X, T' X)$ defined as $\sigma_{X,Y} =_{\mathrm{df}} \tau_{Y,X}$ where $T\,X\,Y =_{\mathrm{df}} R\,(Y,X)$ and $T'\,X\,Y =_{\mathrm{df}} R'\,(Y,X)$.*

(2) *A relative monad morphism σ between relative monads $(T, \eta, (-)^*)$ and $(T', \eta', (-)^{*'})$ gives rise to an arrow morphism τ whose components $\tau_{X,Y} \in R\,(X,Y) \to R'\,(X,Y)$ are defined as $\tau_{X,Y} =_{\mathrm{df}} \sigma_{Y,X}$ where $R\,(X,Y) =_{\mathrm{df}} T\,Y\,X$ and $R'\,(X,Y) =_{\mathrm{df}} T'\,Y\,X$.*

(3) *The categories of relative monads on $\mathbf{Y}$ and arrows on $\mathbb{J}$ are isomorphic.*

Proof.

(1) We need to check the relative monad morphism conditions:

$$\begin{array}{rcl} \sigma\,(\eta\,f) & = & \tau\,(\mathsf{pure}\,f) \\ & = & \qquad \{\text{ by pure pres. law of } \tau \text{ }\} \\ & & \mathsf{pure}'\,f \\ & = & \eta'\,f \end{array}$$

$$\begin{array}{rcl} \sigma\,(k^*\,f) & = & \tau\,(k\,\mathrm{id} \lll f) \\ & = & \qquad \{\text{ by compos. pres. law of } \tau \text{ }\} \\ & & \tau\,(k\,\mathrm{id}) \lll' \tau\,f \\ & = & (\sigma \circ k)^{*'}(\sigma\,f) \end{array}$$

(2) We check the arrow morphism conditions:

$$\begin{array}{rcl} \tau\,(\mathsf{pure}\,f) & = & \sigma\,(\eta\,f) \\ & = & \qquad \{\text{ by unit pres. law of } \sigma \text{ }\} \\ & & \eta'\,f \\ & = & \mathsf{pure}'\,f \end{array}$$

$$\begin{array}{rcl} \tau\,(f \lll g) & = & \sigma\,((\lambda h.\,T\,\_\,h\,f)^*\,g) \\ & = & \qquad \{\text{ by Kl. ext. pres. law of } \sigma \text{ }\} \\ & & (\sigma \circ (\lambda h.\,T\,\_\,h\,f))^{*'}(\sigma\,g) \\ & = & \qquad \{\text{ by naturality of } \sigma \text{ }\} \\ & & (\lambda h.\,T'\,\_\,h\,(\sigma f))^{*'}(\sigma\,g) \\ & = & \tau\,f \lll' \tau\,g \end{array}$$

(3) That the correspondence is an isomorphism is trivial. □

It is easy to verify that the Freyd category of an arrow is the Kleisli category of the corresponding relative monad. Jacobs et al. [21] have previously proved that "Freyd is Kleisli for arrows" taking "Kleisli for arrows" to mean a construction that is Kleisli-like under a 2-categorical view of the Kleisli construction for monads. We can take it to mean "Kleisli for arrows as relative monads".

Similarly to J_{f} and $J_{\mathbf{U}}$ considered above, the functor $\mathbf{Y}$ is well-behaved. The result of Heunen and Jacobs [18] about arrows being monoids follows as an instance of a generality about relative monads.

Theorem 5.5. *The Yoneda embedding $\mathbf{Y} \in \mathbb{J} \to [\mathbb{J}^{\mathrm{op}}, \mathbf{Set}]$ is well-behaved:*

(1) *for any $X, Y \in |\mathbb{J}|$, $J_{X,Y}$ has an inverse*

$$J_{X,Y}{}^{-1} \in [\mathbb{J}^{\mathrm{op}}, \mathbf{Set}]\,(\mathbf{Y}\,X, \mathbf{Y}\,Y) \to \mathbb{J}\,(X,Y)$$

(2) *for any* $G, H \in |[\mathbb{J}^{\mathrm{op}}, \mathbf{Set}]|$, $K_{G,H}$ *has an inverse*

$$K_{G,H}{}^{-1} \in [\mathbb{J}^{\mathrm{op}}, \mathbf{Set}]\,([\mathbb{J}^{\mathrm{op}}, \mathbf{Set}]\,(\mathbf{Y}\,-, G), [\mathbb{J}^{\mathrm{op}}, \mathbf{Set}]\,(\mathbf{Y}\,-, H)) \to [\mathbb{J}^{\mathrm{op}}, \mathbf{Set}]\,(G, H)$$

(3) *for any* $F \in \mathbb{J} \to [\mathbb{J}^{\mathrm{op}}, \mathbf{Set}]$, $X \in |\mathbb{J}|$, $H \in |[\mathbb{J}^{\mathrm{op}}, \mathbf{Set}]|$, $L^F_{X,H}$ *has an inverse*

$$L^F_{X,H}{}^{-1} \in [\mathbb{J}^{\mathrm{op}}, \mathbf{Set}]\,(\mathbf{Y}\,X, \mathrm{Lan}_{\mathbf{Y}}\,F\,H) \to \mathrm{Lan}_{\mathbf{Y}}\,([\mathbb{J}^{\mathrm{op}}, \mathbf{Set}]\,(\mathbf{Y}\,X, F\,-))\,H$$

To prove the 3rd item, we use that coends in presheaf categories are constructed pointwise. We have $\mathrm{Lan}_{\mathbf{Y}}\,F\,H\,Z \cong \mathrm{Lan}_{\mathbf{Y}}\,(F'\,Z)\,H$ where $F'\,Z\,X = F\,X\,Z$.

Proof.

(1) Recall that $J_{X,Y}\,f =_{\mathrm{df}} \mathbf{Y}\,f = \lambda g.\, f \circ g$. By the Yoneda lemma $J_{X,Y}$ is an isomorphism and the inverse of $J_{X,Y}$ is

$$J_{X,Y}{}^{-1}\,\tau =_{\mathrm{df}} \tau\,\mathrm{id}$$

(2) The inverse of $K_{G,H}$ is definable by

$$K_{G,H}{}^{-1}\alpha =_{\mathrm{df}} \lambda a.\, \alpha\,(\lambda f.\, G\,f\,a)\,\mathrm{id}$$

Proof:

$$\begin{array}{rcl}
K_{G,H}{}^{-1}\,(K_{G,H}\,\tau) & = & K_{G,H}{}^{-1}\,(\lambda\theta.\, \tau \circ \theta) \\
& = & \lambda a.\, (\tau \circ \lambda f.\, G\,f\,a)\,\mathrm{id} \\
& = & \lambda a.\, \tau\,((\lambda f.\, G\,f\,a)\,\mathrm{id}) \\
& = & \lambda a.\, \tau\,(G\,\mathrm{id}\,a) \\
& = & \lambda a.\, \tau\,a \\
& = & \tau
\end{array}$$

$$\begin{array}{rcl}
K_{G,H}\,(K_{G,H}{}^{-1}\,\alpha) & = & K_{G,H}\,(\lambda a.\, \alpha\,(\lambda f.\, G\,f\,a)\,\mathrm{id}) \\
& = & \lambda\theta.\, (\lambda a.\, \alpha\,(\lambda f.\, G\,f\,a)\,\mathrm{id}) \circ \theta \\
& = & \lambda\theta.\, \lambda g.\, (\lambda a.\, \alpha\,(\lambda f.\, G\,f\,a)\,\mathrm{id})\,(\theta\,g) \\
& = & \lambda\theta.\, \lambda g.\, \alpha\,(\lambda f.\, G\,f\,(\theta\,g))\,\mathrm{id} \\
& = & \quad \{ \text{ by naturality of } \theta \text{ } \} \\
& & \lambda\theta.\, \lambda g.\, \alpha\,(\lambda f.\, \theta\,(g \circ f))\,\mathrm{id} \\
& = & \lambda\theta.\, \lambda g.\, \alpha\,(\theta \circ (\lambda f.\, g \circ f))\,\mathrm{id} \\
& = & \lambda\theta.\, \lambda g.\, \alpha\,(\theta \circ \mathbf{Y}\,g)\,\mathrm{id} \\
& = & \quad \{ \text{ by naturality of } \alpha \text{ } \} \\
& & \lambda\theta.\, \lambda g.\, (\alpha\,\theta \circ \mathbf{Y}\,g)\,\mathrm{id} \\
& = & \lambda\theta.\, \lambda g.\, \alpha\,\theta\,(\mathbf{Y}\,g\,\mathrm{id}) \\
& = & \lambda\theta.\, \lambda g.\, \alpha\,\theta\,(g \circ \mathrm{id}) \\
& = & \lambda\theta.\, \lambda g.\, \alpha\,\theta\,g \\
& = & \alpha
\end{array}$$

In fact, it is immediate to conclude from the Yoneda lemma (applying it twice) that the sets $[\mathbb{J}^{\mathrm{op}}, \mathbf{Set}]\,(G, H)$ and $[\mathbb{J}^{\mathrm{op}}, \mathbf{Set}]\,([\mathbb{J}^{\mathrm{op}}, \mathbf{Set}]\,(\mathbf{Y}\,-, G), [\mathbb{J}^{\mathrm{op}}, \mathbf{Set}]\,(\mathbf{Y}\,-, H))$ are isomorphic, but we must also check that this isomorphism is $K_{G,H}$.

(3) That the sets $\mathrm{Lan}_{\mathbf{Y}}\left([\mathbb{J}^{\mathrm{op}}, \mathbf{Set}]\,(\mathbf{Y}\,X, F\,-)\right)H$ and $[\mathbb{J}^{\mathrm{op}}, \mathbf{Set}]\,(\mathbf{Y}\,X, \mathrm{Lan}_{\mathbf{Y}}\,F\,H)$ are isomorphic follows from the Yoneda lemma combined with the fact that coends in presheaf categories are constructed pointwise. Again it is important to verify that the isomorphism is $L^F_{X,H}$. □

Corollary 5.6. *If $\mathbb{J}$ is small, then, as $\mathbf{Y}$ is well-behaved, the category $[\mathbb{J}, [\mathbb{J}^{\mathrm{op}}, \mathbf{Set}]]$ is monoidal. An arrow on $\mathbb{J}$ is a monoid in this category.*

Heunen and Jacobs [18] considered the special case of arrows and showed an arrow to be a monoid in $[\mathbb{J}^{\mathrm{op}} \times \mathbb{J}, \mathbf{Set}]$ (the category of endoprofunctors on $\mathbb{J}$) as a monoidal category, which is, of course, an equivalent statement.

6. Conclusions and further work

We have introduced a generalization of monads, relative monads, which is motivated by examples and subsumes arrows, a well-known generalization of monads. Indeed, when moving to a more precise type discipline, the illusion that everything takes place in only one ambient category (say, **Set**) can no longer be maintained and as a consequence we have to revisit the categorically inspired concepts of functional programming. We believe that our examples demonstrate that monad-like entities which are not endofunctors are natural; fortunately, they are precisely monoids in the functor category. We also suggest that our presentation of relative monads given in Sect. 2.1 is accessible for functional programmers, indeed it does not differ substantially from ordinary monads.

We will elsewhere comment on the relation of our relative monads to the recent generalization of monads by Spivey [31] that was also motivated by programming examples: he fixes a functor $K \in \mathbb{C} \to \mathbb{J}$ (notice the direction) to then look for monad-like structures with an underlying functor $\mathbb{J} \to \mathbb{C}$. With Paul Levy we have checked that a fair amount of monad theory transfers to his generalized monads, but they are not monoids in $[\mathbb{J}, \mathbb{C}]$ unless K has a left adjoint, in which case they are equivalent to relative monads. Sam Staton has considered an enriched variant of relative monads [32].

It seems clear that many of the concepts known from ordinary monads carry over to the relative setting. We hope that this generalization of the monadic approach leads to new programming structures supporting a greater reusability of concepts and programs. Indeed, relative monads have already been used by Ahrens to model syntax with a reduction relation [3, 4]. Orchard [28] has generalized monads to relative monads in Haskell using constraint kinds and associated types. Gabbay and Nanevski [15] needed relative comonads in their work on contextual modal type theory.

We have formalized a large part of the development of the present paper in the dependently typed programming language Agda [6].

Skew-monoidal categories are interesting in their own right. We have recently [35] proved a coherence theorem for them—identified a sufficient condition for a unique "formal" map between two given "formal" objects. Lack and Street [23] proved a different one, which is a necessary and sufficient condition for equality of two given maps.

Acknowledgements. We are grateful to Paul Levy and Thomas Streicher for valuable comments and hints, and to the anonymous referees of both this paper and the conference version on which it is based.

References

[1] M. Abbott, T. Altenkirch, and N. Ghani. Containers—constructing strictly positive types. *Theor. Comput. Sci.*, 342(1):3–27, 2005.

[2] S. Abramsky and B. Coecke. A categorical semantics of quantum protocols. In *Proc. of 19th Ann. IEEE Symp. on Logic in Computer Science, LICS 2004 (Turku, July 2004)*, pp. 415–425. IEEE CS Press, 2004.

[3] B. Ahrens. Initiality for typed syntax and semantics. In L. Ong and R. de Queiroz, eds., *Proc. of 19th Wksh. on Logic, Language, Information and Computation, WoLLIC 2012 (Buenos Aires, Sept. 2012)*, vol. 7456 of *Lect. Notes in Comput. Sci.*, pp. 127–141. Springer, 2012.

[4] B. Ahrens. Modules over relative monads for syntax and semantics. *Log. Methods in Comput. Sci.*, 8(2:1), 2012.

[5] T. Altenkirch, J. Chapman, and T. Uustalu. Monads need not be endofunctors. In L. Ong, ed., *Proc. of 13th Int. Conf. on Foundations of Software Science and Computational Structures, FoSSaCS 2010 (Paphos, March 2010)*, vol. 6014 of *Lect. Notes in Comput. Sci.*, pp. 297–311. Springer, 2010.

[6] T. Altenkirch, J. Chapman, and T. Uustalu. Relative monads formalized. *J. of Formalized Reasoning*, 7(1):1–43, 2014.

[7] T. Altenkirch and A. Green. The quantum IO monad. In S. Gay and I. Mackie, eds., *Semantic Techniques in Quantum Computation*, pp. 173–205. Cambridge University Press, 2009.

[8] T. Altenkirch and B. Reus. Monadic presentations of lambda terms using generalized inductive types. In J. Flum and M. Rodríguez-Artalejo, eds., *Proc. of 13th Int. Wksh. on Computer Science Logic, CSL '99 (Madrid, Sept. 1999)*, vol. 683 of *Lect. Notes in Comput. Sci.*, pp. 453–468. Springer, 1999.

[9] K. Asada. Arrows are strong monads. In *Proc. of 3rd ACM SIGPLAN Wksh. on Mathematical Structured Functional Programming, MSFP '10 (Baltimore, MD, Sept. 2010)*, pp. 33–41. ACM Press, 2010.

[10] C. Berger, P.-A. Melliès, and M. Weber. Monads with arities and their associated theories. *J. of Pure and Appl. Alg.*, 216(8–9):2029–2048, 2012.

[11] R. F. Blute, R. J. B. Cockett, and R. A. G. Seely. Categories for computation in context and unified logic. *J. of Pure and Appl. Alg.*, 116(1–3):49–98, 1997.

[12] M. Buckley, R. Garner, S. Lack, and R. Street. The Catalan simplicial set. *Math. Proc. of Cambridge Philos. Soc.*, to appear.

[13] M. Fiore. Semantic analysis of normalisation by evaluation for typed lambda calculus. In *Proc. of 4th ACM SIGPLAN Conf. on Principles and Practice of Declarative Programming, PPDP 2002 (Pittsburgh, PA, Oct. 2002)*, pp. 26–37. ACM Press, 2002.

[14] M. Fiore, G. Plotkin, and D. Turi. Abstract syntax and variable binding. In *Proc. of 14th Ann. IEEE Symp. on Logic in Computer Science, LICS '99 (Trento, July 1999)*, pp. 193–202. IEEE CS Press, 1999.

[15] M. J. Gabbay and A. Nanevski. Denotation of contextual modal type theory (CMTT): Syntax and meta-programming. *J. of Appl. Log.*, 11(1):1–29, 2013.

[16] C. Grellois. Algebraic theories, monads, and arities. Master's thesis, Université Paris 6, 2011.

[17] C. A. Gunter. *Semantics of Programming Languages*. MIT Press, 1992.

[18] C. Heunen and B. Jacobs. Arrows, like monads, are monoids. In S. Brookes and M. Mislove, eds., *Proc. of 22nd Ann. Conf. on Mathematical Foundations of Programming Semantics, MFPS XXII (Genova, May 2006)*, vol. 158 of *Electron. Notes in Theor. Comput. Sci.*, pp. 219–236. Elsevier, 2006.

[19] J. R. Hindley and J. P. Seldin. *Introduction to Combinators and Lambda-Calculus*, vol. 1 of *London Mathematical Society Student Texts*. Cambridge University Press, 1986.

[20] J. Hughes. Generalising monads to arrows. *Sci. of Comput. Program.*, 37(1–3):67–111, 2000.

[21] B. Jacobs, C. Heunen, and I. Hasuo. Categorical semantics for arrows. *J. of Funct. Program.*, 19(3–4):403–438, 2009.

[22] S. Lack and R. Street. Skew monoidales, skew warpings and quantum categories. *Theory and Appl. of Categ.*, 26:385–402, 2012.

[23] S. Lack and R. Street. Triangulations, orientals, and skew monoidal categories. *Adv. in Math*, 258:351–396, 2012.

[24] E. G. Manes. *Algebraic Theories*, vol. 26 of *Graduate Texts in Mathematics*. Springer, 1976.

[25] C. McBride and R. Paterson. Applicative programming with effects. *J. of Funct. Program.*, 18(1):1–13, 2008.

[26] P. Morris and T. Altenkirch. Indexed containers. In *Proc. of 24th Ann. IEEE Symp. on Logic in Computer Science, LICS 2009 (Los Angeles, CA, Aug. 2009)*, pp. 277–285. IEEE CS Press, 2009.

[27] U. Norell. Dependently typed programming in Agda. In P. Koopman, R. Plasmeijer, and D. Swierstra, eds., *Revised Lectures from 6th Int. School on Advanced Functional Programming, AFP 2008 (Heijen, May 2008)*, vol. 5832 of *Lect. Notes in Comput. Sci.*, pp. 230–266. Springer, 2009.
[28] D. Orchard and A. Mycroft. Categorical programming for data types with restricted parametricity. Manuscript, 2012.
[29] D. Piponi. Commutative monads, diagrams and knots. In *Proc. of 14th Int. Conf. on Functional Programming, ICFP '09*, pp. 231–232. ACM Press, 2009.
[30] J. Power and E. Robinson. Premonoidal categories and notions of computation. *Math. Struct. in Comput. Sci.*, 7(5):453–468, 1997.
[31] J. M. Spivey. Algebras for combinatorial search. *J. of Funct. Program.*, 19(3–4):469–487, 2009.
[32] S. Staton. Enriched clones, monoids, and relative monads. Unpublished note, 2012.
[33] K. Szlachányi. Skew-monoidal categories and bialgebroids. *Adv. in Math.*, 231(3–4):1694–1730, 2012.
[34] T. Uustalu. Strong relative monads. In B. Jacobs, M. Niqui, J. Rutten, and A. Silva, eds., *Short Contributions of 10th Int. Wksh. on Coalgebraic Methods in Computer Science, CMCS '10 (Paphos, March 2010)*, Technical Report SEN-1004, pp. 23–24. CWI, 2011.
[35] T. Uustalu. Coherence for skew-monoidal categories. In N. Krishnaswamy and P. Levy, eds., *Proc. of 5th Wksh. on Mathematically Structured Functional Programming, MSFP 2014 (Grenoble, Apr. 2014)*, vol. 153 of *Electron. Proc. in Theor. Comput. Sci.*, pp. 68–77. Open Publ. Assoc., 2014.
[36] T. Uustalu and V. Vene. Comonadic notions of computation. In J. Adamék and C. Kupke, eds., *Proc. of 9th Int. Wksh. on Coalgebraic Methods in Computer Science, CMCS 2008 (Budapest, Apr. 2008)*, vol. 203(5) of *Electron. Notes in Theor. Comput. Sci.*, pp. 263–284. Elsevier, 2008.
[37] J. K. Vizzotto, T. Altenkirch, and A. Sabry. Structuring quantum effects: Superoperators as arrows. *Math. Struct. in Comput. Sci.*, 16(3):453–468, 2006.