Theory and Applications of Categories, Vol. 16, No. 24, 2006, pp. 668–699.

DESCENT FOR MONADS

PIETER HOFSTRA AND FEDERICO DE MARCHI

ABSTRACT. Motivated by a desire to gain a better understanding of the "dimension-by-dimension" decompositions of certain prominent monads in higher category theory, we investigate descent theory for endofunctors and monads. After setting up a basic framework of indexed monoidal categories, we describe a suitable subcategory of Cat over which we can view the assignment $\mathsf{C} \mapsto \mathrm{Mnd}(\mathsf{C})$ as an indexed category; on this base category, there is a natural topology. Then we single out a class of monads which are well-behaved with respect to reindexing. The main result is now, that such monads form a stack. Using this, we can shed some light on the free strict ω-category monad on globular sets and the free operad-with-contraction monad on the category of collections.

1. Introduction

The original motivation for the investigations presented here is a talk given by Eugenia Cheng during the PSSL 79 meeting in Utrecht, 2003. (See the paper [Che04] for a first-hand account.) She outlined an explicit construction for the free operad-with-contraction on a collection. There are two monads on the category of collections, namely the free operad monad and the free collection-with-contraction monad, and the construction amounts to the formation of their coproduct. Given a collection X, the problem is to freely add both operadic and contraction structure to X. The collection X itself can be "truncated", by throwing away all dimensions higher than some given n. Similarly, both monads in question can be restricted to the category of such n-dimensional collections. Now, we can build the free algebra structure on X dimension-by-dimension: we truncate X at dimension 0, and apply in order the (restriction of) the free operad monad and the (restriction of) the free contraction monad. Then, we move up and consider the 1-dimensional collection whose 0-dimensional part consists of the result of the previous step, whereas its 1-dimensional part is that of X. To this, we apply the (restrictions of) both monads, in the same order. Iterating the process, we give a description of the n-th dimension of the free operad-with-contraction on X for every natural number n. Intuitively, we could say that the category of collections admits a certain "decomposition", which is respected both by the free operad and the free collection-with-contraction monad.

We started wondering whether this picture could be formalised at a more abstract level. In particular, what does it mean for a category to admit such a dimension-by-

We are grateful to Eugenia Cheng and Mark Weber for stimulating discussions and useful suggestions. We have also benefited from the helpful comments by the referee.

Received by the editors 2005-09-16 and, in revised form, 2006-09-01.

Transmitted by Richard Blute. Published on 2006-09-30.

2000 Mathematics Subject Classification: 18C15, 18D10, 18D30.

Key words and phrases: Descent theory, monads, globular sets.

dimension decomposition? And what does it mean for a monad on such a category to be defined "dimension-by-dimension"? More generally, given a category C and a family of subcategories D_i (which are thought of as giving a covering of C), when can a monad on C be reconstructed from a family of monads on the D_i's?

The question found a natural formulation in terms of descent conditions. How can we view monads as indexed over a suitable base category, and under which conditions (and for which kind of topology on the base) is this indexed category a stack? Since the examples we tried to understand were fairly specific (in that the categories involved there are all presheaf toposes), completely general results were not to be expected. Indeed, in order to make the assignment $\mathsf{C} \mapsto \mathrm{Mnd}(\mathsf{C})$ into an indexed category, several assumptions on the categories involved and the functors between them are needed. Even though it turns out that these assumptions are fairly restrictive, we think that the examples we present make it worth considering them.

Of course, the questions we ask about monads are equally meaningful for endofunctors and in fact, we can first study this (easier) case and later deduce the results for monads by abstract methods.

The approach we took to the problem and its solution can be sketched in the following way. First, we describe conditions on a sub-2-category of Cat that guarantee that the assignment $\mathsf{C} \mapsto \mathrm{End}(\mathsf{C})$ becomes an indexed category. This indexed category will then have several special features; namely, it will have indexed coproducts and be and monoidal, in the sense that its fibres are monoidal categories (via the composition of endofunctors) and reindexing functors are strong monoidal. By "taking monoids in each fibre", we obtain an indexed category $\mathsf{C} \mapsto \mathrm{Mnd}(\mathsf{C})$, whose descent problem, for abstract reasons, reduces to the one for endofunctors.

At this point, we turn our attention to a specific topology, which canonically arises on the base category, and ask whether the indexed category $\mathrm{End}(-)$ is a stack for this topology. It turns out that, except for some degenerate cases, the answer to this question is negative. However, we can single out a class of endofunctors having a certain stability property, for which we can prove that they form not only an indexed monoidal category, but even a stack for the topology on the base category. Moreover, the abstract machinery developed before allows us to deduce that monads with the stability property also form a stack. Finally, it will be seen that the introduced topology is unique with this property.

Intuitively, we can interpret these results as follows: the topology is defined in such a way that, when a category C in the base is covered by a family R, this can be seen as a decomposition of the category. The fact that stable endofunctors are of effective descent for this family means that this geometrical property of the category C lifts to give an analogous decomposition of stable endofunctors, and the same goes for monads.

The paper is structured as follows. We start off by explaining our terminology and notations regarding 2-categories, indexed categories, and descent theory, in section 2. In particular, we look at various presentations of categories of descent data in terms of pseudo-limits. In section 3, we introduce the basic framework we will be working in, namely that of indexed monoidal categories. We exhibit an elementary construction which takes an indexed monoidal category and produces an ordinary indexed category by taking monoids in each fibre.

In section 4, we establish the results on descent categories for indexed monoidal categories which will be needed in the rest of the paper. This is the most technical part of the paper, and details of the proofs can be skipped by the reader who is only interested in the main results.

Section 5 focuses on the specific indexed monoidal category End(−). We find sufficient conditions on the base category in order to actually make End(−) into an indexed category, and to ensure cocompleteness. Passing to monoids will then give us the indexed category Mnd(−). We conclude the section by introducing a topology on the base category, which arises naturally by considerations on the trivial indexing of the base over itself.

In section 6, we study the property of stability, which enables us to view endofunctors as a stack for the aforementioned topology. The contents of all these sections add up to give the main results of this paper, which are collected in section 7. These say that stable endofunctors and stable monads form a stack, and that the topology on the base category is the only topology for which this is the case.

Finally, section 8 returns to our original motivation and shows that the free operad-with-contraction can be understood in this manner.

In order to improve the digestibility of this technical work and to provide the reader with a more concrete grip on the abstract concepts involved, we have also included a running example, namely the free strict ω-category monad on the category of globular sets.

2. Preliminaries and Notation

Throughout the paper, the following notation will be used: Cat denotes the 2-category of small categories (but in general we shall tend to ignore all size issues). Its objects will be denoted $\mathsf{C}, \mathsf{D}, \ldots$, its arrows by $x, y, \ldots$ and the 2-cells by $\alpha, \beta, \ldots$ The notation MonCat will be used for the 2-category of (not necessarily strict) monoidal categories, (lax) monoidal functors and monoidal transformations. At times, we shall restrict to strong monoidal functors (i.e. monoidal functors for which the coherence maps are all isomorphisms); then, we will use MonCat_s. If C is a monoidal category, we usually write $\otimes$ for the tensor product and I for the tensor unit. A monoid in C will be an object $X = (X, \mu, \iota)$ equipped with a multiplication $\mu : X \otimes X \to X$ and a unit $\iota : I \to X$, subject to the usual monoid axioms.

As for indexed category-theoretical matters, we mostly follow the notation of [Joh02]. That is, we have a base category denoted $\mathcal{S}$; its objects will be denoted $J, K, L, \ldots$ (but not I, since we already reserved that for tensor units). In our applications, the base category will actually be a 2-category, but the 2-categorical structure will not be used. Usually, we assume our base category to have pullbacks: this is helpful in dealing with descent categories. Pseudo-functors $\mathcal{S}^{op} \to \mathsf{Cat}$ are called $\mathcal{S}$-indexed categories, and we write $\mathbb{C}, \mathbb{D}, \ldots$ for them. If J is an object of the base category $\mathcal{S}$, then the fibre of $\mathbb{C}$ over

J will be denoted C^J. If $x : J \to K$ is a map in $\mathcal{S}$, its image under an indexed category $\mathbb{C}$ will be the reindexing functor $x^* : \mathsf{C}^K \to \mathsf{C}^J$. If x^* has a right (left) adjoint we will write Π_x (Σ_x) for it. An $\mathcal{S}$-indexed category $\mathbb{C}$ is called *complete* if all fibres are finitely complete, all reindexing functors have right adjoints and the Beck-Chevalley Condition (BCC) is satisfied, i.e. if for each pullback square

$$\begin{tikzcd} L \arrow[r, "v"] \arrow[d, "u"'] & M \arrow[d, "x"] \\ J \arrow[r, "y"'] & K \end{tikzcd}$$

the canonical natural transformation $x^*\Pi_y \to \Pi_v u^*$ is an isomorphism. Similarly, $\mathbb{C}$ is called *cocomplete* if all fibres are finitely cocomplete and reindexing functors have left adjoints and satisfy the BCC, which this time means that, for each pullback as above, the canonical map $\Sigma_u v^* \to y^*\Sigma_x$ is an isomorphism.

We write $\mathcal{S}-\mathsf{Cat}$ for the 2-category of $\mathcal{S}$-indexed categories, indexed functors and indexed transformations (see [Joh02] for the relevant definitions).

When we speak of a pseudo-diagram in a 2-category $\mathcal{M}$, we mean a pseudo-functor F from a (small) category into $\mathcal{M}$. A pseudo-cone with vertex X over such a pseudo-diagram is a pseudo-natural transformation $\Delta_X \to F$, where Δ_X is the constant functor with value X. For fixed X, we have a category $\mathsf{PsCone}(X, F)$ where the objects are pseudo-cones over F with vertex X and arrows are modifications. We also recall that, in the case where $\mathcal{M} = \mathsf{Cat}$, the pseudo-limit of such a pseudo-diagram F can be taken to be the category $\mathsf{PsCone}(1, F)$ of pseudo-cones over F with vertex 1, the terminal category.

We say that a pseudo-functor $F : \mathcal{S} \to \mathsf{Cat}$ is *representable* if it is equivalent (and not necessarily isomorphic) to a one of the form $\mathcal{S}(A, -)$ for some object A. Representable functors preserve all weighted limits (see [Str80, §1.20]). In the special case where the weight is the terminal category, this means that representable functors preserve pseudo-limits.

If $\mathbb{C}$ is an $\mathcal{S}$-indexed category and $R = \{x_\gamma : J_\gamma \to K \mid \gamma \in \Gamma\}$ is a collection of maps in the base $\mathcal{S}$ with common codomain K, then we have a category $\mathsf{Desc}(R, \mathbb{C})$ of *descent data*, described as follows. First, for each pair $\gamma, \delta \in \Gamma$, we have a pullback in $\mathcal{S}$

$$\begin{tikzcd} L_{\gamma\delta} \arrow[r, "z=z_{\gamma\delta}"] \arrow[d, "y=y_{\gamma\delta}"'] & J_\delta \arrow[d, "x_\delta"] \\ J_\gamma \arrow[r, "x_\gamma"'] & K. \end{tikzcd}$$

Now, the objects of $\mathsf{Desc}(R, \mathbb{C})$ are systems $(\{X_\gamma\}_{\gamma\in\Gamma}, \{f_{\gamma\delta}\}_{\gamma,\delta\in\Gamma})$, or more briefly $(X_\gamma, f_{\gamma\delta})$, where X_γ is an object of C^{J_γ}, and $f_{\gamma\delta} : y^*(X_\gamma) \to z^*(X_\delta)$ is an isomorphism in the fibre $\mathsf{C}^{L_{\gamma\delta}}$. The $f_{\gamma\delta}$ are required to satisfy certain coherence conditions, usually referred to as the *unit* and *cocycle* conditions; see [Joh02, Jan94, Jan97] for details.

There is a canonical functor $\Theta = \Theta_{R,\mathbb{C}} : \mathsf{C}^K \to \mathsf{Desc}(R, \mathbb{C})$, which sends an object X to the family $(x_\gamma^* X, f_{\gamma\delta})$, where $f_{\gamma\delta}$ is the isomorphism arising from the coherence of $\mathbb{C}$ as a pseudo-functor. Also, for each $\gamma \in \Gamma$, there is a forgetful functor: $U_\gamma : \mathsf{Desc}(R, \mathbb{C}) \to \mathsf{C}^{J_\gamma}$. These functors are related via the equality

$$x_\gamma^* = U_\gamma \circ \Theta : \mathsf{C}^K \to \mathsf{C}^{J_\gamma}. \tag{1}$$

At a more abstract level, we can introduce $\mathsf{Desc}(R, \mathbb{C})$ as the pseudo-limit of the composite

$$\mathbb{C}_R \;:\; \mathcal{R}^{op} \longrightarrow (\mathcal{S}/K)^{op} \longrightarrow \mathcal{S}^{op} \xrightarrow{\mathbb{C}} \mathsf{Cat}$$

where $\mathcal{R}$ is the full subcategory of the slice $\mathcal{S}/K$ on the objects $M \to K$ which factor through a map in R, and where $(\mathcal{S}/K)^{op} \to \mathcal{S}^{op}$ is the forgetful functor.

In other words, the category of descent data is $\mathsf{PsCone}(1, \mathbb{C}_R)$. This provides another description of $\mathsf{Desc}(R, \mathbb{C})$, where objects are systems $(\alpha_\gamma \in \mathsf{C}^{J_\gamma})$ for any $x_\gamma : J_\gamma \to K$ in R, together with isomorphisms $\alpha_f : f^*(\alpha_\gamma) \to \alpha_\delta$ where f runs over all maps $f : J_\delta \to J_\gamma$ such that $x_\delta = x_\gamma \circ f$. Of course, these α_f are supposed to satisfy the coherence conditions $\alpha_1 = 1, \alpha_f \alpha_g = \alpha_{gf}$.

In this perspective, the functor $\Theta : \mathsf{C}^K \to \mathsf{Desc}(R, \mathbb{C})$ is nothing but the mediating arrow from the pseudo-cone with vertex C^K and projections the restriction functors $x_\gamma^* : \mathsf{C}^K \to \mathsf{C}^{J_\gamma}$ to the pseudo-limit $\mathsf{Desc}(R, \mathbb{C}) = \mathsf{PsCone}(1, \mathbb{C}_R)$. On objects, this can be described as sending X in C^K to the system $(x_\gamma^* X)$ equipped with the coherence isomorphisms from the indexed category $\mathbb{C}$. The conditions expressed by (1) are saying precisely that Θ is a morphism of pseudo-cones.

If the functor Θ is an equivalence of categories (respectively, full and faithful), then we say that $\mathbb{C}$ is of *effective (pre-)descent* for R. Usually we drop the predicate "effective".

Let $\mathcal{J}$ be a Grothendieck topology on $\mathcal{S}$, or, more generally, a collection of families of maps with common codomain. Families $R \in \mathcal{J}(K)$ will be called covering families for K. If an indexed category $\mathbb{C}$ is of (pre-)descent for all families in $\mathcal{J}$, then $\mathbb{C}$ is called a *$\mathcal{J}$-(pre-)stack*. If the topology is clear, then we simply call $\mathbb{C}$ a (pre-)stack.

Running Example. We end this section by briefly introducing our running example (which, by the way, will not start running until section 5): globular sets and the free strict ω-category monad. For more explicit and elaborate information on these matters, see [Lei04, Str00].

Write $\mathcal{G}$ for the category generated by the natural numbers as objects, and two arrows $s, t : n \to n+1$ for each n, subject to the equations $ss = ts, st = tt$. For every n, we also consider $\mathcal{G}_n$, which is the full subcategory on the objects $0, \ldots, n$. For each, $n < m$, there is an inclusion $\mathcal{G}_n \to \mathcal{G}_m$. For convenience, we will sometimes write $\mathcal{G} = \mathcal{G}_\infty$.

The category of presheaves on $\mathcal{G}$ will be denoted GSet, and those on $\mathcal{G}_n$ by GSet_n. A typical object A of GSet will be written

$$\cdots \rightrightarrows A_2 \rightrightarrows A_1 \rightrightarrows A_0.$$

The inclusion functors now induce essential geometric morphisms. The inverse image of such a morphism can be viewed as *truncation*, and its left adjoint is given by left Kan extension.

More explicitly, for $n < m$, where we allow $m = \infty$, we write

$$r_{n,m} : \mathsf{GSet}_m \to \mathsf{GSet}_n, \qquad (r_{n,m}A)_k = A_k$$

and

$$i_{n,m} : \mathsf{GSet}_n \to \mathsf{GSet}_m, \qquad (i_{n,m}A)_k = \begin{cases} A_k & \text{if } k \leq n \\ \emptyset & \text{otherwise} \end{cases}$$

for the inverse image and its left adjoint.

The left adjoints are fully faithful, so that we have the isomorphism $r_{n,m}i_{n,m} \cong 1$ (which may actually be taken to be the identity).

On the category GSet there is a monad T, the free strict ω-category monad. This monad is familially represented by a family of objects, which we will describe first, following the presentation of [Lei04].

Let $\mathbf{pd}(0) = *$ and $\mathbf{pd}(n+1)$ be the free monoid on $\mathbf{pd}(n)$. Elements of $\mathbf{pd}(n)$ are called *n-dimensional pasting diagrams*. To each $\pi \in \mathbf{pd}(n)$ we associate an n-dimensional globular set $\hat{\pi}$, which has in dimension k the set of k-cells of π. Now T, the free strict ω-category monad, can be defined by

$$(TA)_n = \coprod_{\pi \in \mathbf{pd}(n)} \mathsf{GSet}[\hat{\pi}, A]. \tag{2}$$

Informally, this is the set of all possible ways of labelling n-dimensional pasting diagrams with suitably typed cells from A.

The globular pasting diagrams can now be recovered via $\mathbf{pd}(n) = (T1)_n$, where 1 is the terminal globular set.

3. Indexed Monoidal Categories

We begin by setting up the basic framework of monoidal categories indexed over a base $\mathcal{S}$, and providing a number of examples. We explain that to each indexed monoidal category $\mathbb{C}$ we can associate another indexed category $\mathbf{MON}(\mathbb{C})$ (in general not monoidal), which associates to an object J the category of monoids in the fibre of $\mathbb{C}$ over J. Then we look at completeness and cocompleteness of such indexed structures. Finally, descent for indexed monoidal categories is studied. The main point here is that, whenever the reindexing functors of $\mathbb{C}$ are strong monoidal, the indexed category $\mathbf{MON}(\mathbb{C})$ is a stack when $\mathbb{C}$ is.

3.1. Basic Definitions. At first, we found it hard to imagine that the notion of an indexed monoidal category was a new concept. In fact, it was not. It has already been introduced in the study of the semantics of logic programming languages (see for instance [Asp93, Ama01]). Since we use it in a different context, we find it appropriate to present the definitions here and develop some of the basic theory. We also recall some results about the 2-functor **Mon** : $\mathsf{MonCat} \to \mathsf{Cat}$, which takes a monoidal category to the category of monoids in it. This will be used in section 4, where we study descent for monoidal indexed categories. Some of the results are not stated at the highest possible level of abstraction, as that would carry us too far astray from the main focus of this paper.

3.2. Definition. [Indexed Monoidal Categories] *Let $\mathcal{S}$ be any category. An $\mathcal{S}$*-indexed monoidal category *is pseudofunctor $\mathcal{S}^{op} \to \mathsf{MonCat}$. Explicitly, this is an $\mathcal{S}$-indexed category $\mathbb{C}$ such that all fibres C^J are monoidal categories, all reindexing functors are monoidal functors and all coherence isomorphisms are monoidal transformations.*

Even more explicitly, the last requirement amounts to the commutativity of the following diagrams (where $J \xrightarrow{s} K \xrightarrow{t} L$ is a composable pair in $\mathcal{S}$)

$$\begin{tikzcd} s^*t^*A \otimes s^*t^*B \arrow[r] \arrow[d] & s^*t^*(A \otimes B) \arrow[d] \\ (ts)^*A \otimes (ts)^*B \arrow[r] & (ts)^*(A \otimes B) \end{tikzcd} \qquad \begin{tikzcd} I \arrow[r] \arrow[drr] & s^*I \arrow[r] & s^*t^*I \arrow[d] \\ & & (ts)^*I \end{tikzcd}$$

in the fibre C^J, for all objects A, B in C^L. The various I denote the unit for the tensor product. (We do not bother distinguishing notationally between tensor units in different fibres, since there is only one way for the diagram to pass type-checking.)

Similarly, $\mathbb{C}$ is called an *$\mathcal{S}$-indexed strong monoidal category*, when, as a pseudofunctor, it takes values in MonCat_s; that is, when all reindexing functors are strong monoidal functors.

3.3. Examples. Of course, examples of these structures abound:

1. Take any category with finite limits C and consider the canonical indexing $J \mapsto \mathsf{C}/J$. Reindexing now preserves finite products, so when we define the tensor to be the ordinary cartesian product, the canonical indexing is an indexed (strong) monoidal category.

2. As a base category, consider the category of rings, Rng. Then $\mathrm{Mod}(-)$, which assigns to a ring its category of modules, is a (strong) Rng-indexed monoidal category.

3. For each topos $\mathcal{E}$, we may consider the category of Complete Sup-Lattices in $\mathcal{E}$, denoted $\mathrm{CSL}(\mathcal{E})$. This is a monoidal category (in fact, it is *-autonomous), see [Joy84].

   For each geometric morphism $x : \mathcal{E} \to \mathcal{F}$, we have an induced functor $x_* : \mathrm{CSL}(\mathcal{E}) \to \mathrm{CSL}(\mathcal{F})$, which has a left adjoint x^*. Both of these functors are monoidal, and the assignment $\mathcal{E} \mapsto \mathrm{CSL}(\mathcal{E})$, $x \mapsto x^*$ becomes a (strong) indexed monoidal category over the base category $\mathcal{TOP}$ of toposes and geometric morphisms.

4. As a base category consider MonCat_s. To each monoidal category M, we may associate a category of representations $\mathrm{Mod}(\mathsf{M})$, i.e. strong monoidal functors

$\mathsf{M} \to \mathrm{End}(\mathsf{A})$, for some category A. Of course, maps in $\mathrm{Mod}(\mathsf{M})$ are monoidal transformations, and this category is monoidal, with tensor product induced by the maps $\mathrm{End}(A) \times \mathrm{End}(B) \to \mathrm{End}(A \times B)$, $(f, g) \mapsto f \otimes g$, where $(f \otimes g)(a, b) = (fa, gb)$. All this is a straightforward generalisation of M-sets, where M is a monoid. (For more, see [Jan01].)

Anyway, a strong monoidal functor $\mathsf{N} \to \mathsf{M}$ will now induce, by precomposition, a reindexing functor $\mathrm{Mod}(\mathsf{M}) \to \mathrm{Mod}(\mathsf{N})$, making $\mathrm{Mod}(-)$ into a MonCat_s-indexed category. Variations can be obtained by replacing "strong" by "strict" or by "lax".

5. Let $\mathcal{S}$ be a sub-2-category of Cat such that every functor $x : \mathsf{D} \to \mathsf{C}$ in $\mathcal{S}$ has a right adjoint $x \dashv y$ in Cat. As noted by Bénabou [Ben89], such an adjunction $x \dashv y$ determines one between the associated categories of endofunctors:

$$\begin{tikzcd} \mathrm{End}(\mathsf{D}) \arrow[r, shift left, "x-y"] \arrow[r, phantom, "\perp"] & \mathrm{End}(\mathsf{C}) \arrow[l, shift left, "y-x"] \end{tikzcd} \tag{3}$$

where $y{-}x$ sends an endofunctor T to the composite yTx (and analogously for $x{-}y$). Moreover, the functor $y - x$ is monoidal: the unit and counit of the adjunction $x \dashv y$ give us natural transformations $1 \to y1x = yx$ and $yTxySy \to yTSx$. Therefore, we can define an $\mathcal{S}$-indexed monoidal category by mapping each category C to $\mathrm{End}(\mathsf{C})$, and each functor $x : \mathsf{D} \to \mathsf{C}$ with a left adjoint y to the functor $y - x : \mathrm{End}(\mathsf{C}) \to \mathrm{End}(\mathsf{D})$.

This example will be crucial to our paper, and we shall expand on it in the following sections.

We will now describe a basic construction on indexed monoidal categories. Recall that there is a 2-functor $\mathbf{Mon} : \mathsf{MonCat} \to \mathsf{Cat}$, which sends a monoidal category C to the category of monoids in C. We stress, probably unnecessarily, that this functor is not the same as the obvious forgetful functor $\mathsf{MonCat} \to \mathsf{Cat}$. We recall in the following lemma all the properties of $\mathbf{Mon}$ which we shall be concerned with in this paper. Proofs are admittedly sketchy; after all, these are well-known facts, and their proof is a mere exercise in the theory of 2-categories.

3.4. Lemma. *Denoting by U the inclusion functor $\mathsf{MonCat}_s \hookrightarrow \mathsf{MonCat}$, the following properties hold:*

1. $\mathbf{Mon}$ *has a left adjoint* $\mathrm{FM} : \mathsf{Cat} \to \mathsf{MonCat}$*;*

2. $\mathbf{Mon}$ *is represented by the one-object category* 1*;*

3. *the composite* $\mathbf{Mon} \circ U$ *is represented by the simplicial category* Δ*.*

PROOF.

1. To a category C we associate a monoidal category $\mathrm{FM}(\mathsf{C})$, with the same objects as the free monoidal category on C. As arrows, it will have all arrows generated by those of C, but also new arrows $C \otimes C \to C$ and $1 \to C$ making each object C of C into a monoid. Hence, $\mathrm{FM}(\mathsf{C})$ is the free monoidal category on C in which each object has a monoid structure.

   Notice that, for the special case $\mathsf{C} = 1$, the image $\mathrm{FM}(1)$ is the free monoidal category generated by one object containing a monoid. This is easily seen to be the simplicial category Δ.

2. This is presented in [Ben89], and it is just an exercise on monoidal categories and lax monoidal functors.

3. If we restrict our attention to strong monoidal functors, 1 is no longer a representative of the functor, because a strong functor $1 \to \mathsf{C}$ is the same as a monoid in C for which all the structure maps are isomorphisms. However, we can classify monoids by means of functors from the simplicial category Δ (see [Mac98]).

■

We shall henceforth not distinguish notationally between the functor **Mon** and its restriction $\mathbf{Mon} \circ U$ to the 2-category MonCat_s.

3.5. COROLLARY. *The functor* **Mon** *preserves all weighted limits that exist in* MonCat *(and* MonCat_s*). In particular, it preserves all existing pseudo-limits.*

PROOF. There is one clarification to make; silently, we work in the meta-setting of 2-categories, pseudo-functors, pseudo-natural transformations and modifications. A proof of the fact that representable functors preserve weighted limits can be found in [Kel82], but the setting there is that of 2-categories, 2-functors, 2-natural transformations and modifications. It is, however, not difficult to extend this to our setting; in addition, working with pseudo-functors instead of 2-functors has the advantage that a pseudo-limit becomes a special case (namely where the weight is trivial) of a weighted limit. ■

Having these facts at our disposal, we now use **Mon** to define a useful construction on indexed monoidal categories.

3.6. CONSTRUCTION. Let $\mathbb{C}$ be an $\mathcal{S}$-indexed monoidal category. Then, the assignment $J \mapsto \mathbf{Mon}(\mathsf{C}^J)$, the category of monoids in C^J, is the object part of an $\mathcal{S}$-indexed category, which we denote by $\mathbf{MON}(\mathbb{C})$. Moreover, there exists an $\mathcal{S}$-indexed forgetful functor $\mathbf{MON}(\mathbb{C}) \to \mathbb{C}$.

In fact, the assignment $\mathbb{C} \mapsto \mathbf{MON}(\mathbb{C})$ is a 2-functor from $\mathcal{S}$-indexed monoidal categories to $\mathcal{S}$-indexed categories, given by postcomposition with the 2-functor **Mon**. Moreover, the left adjoint FM to **Mon** induces a left adjoint on the level of indexed categories.

Recalling that we write $\mathcal{S}$-Cat for the category of $\mathcal{S}$-indexed categories and $\mathcal{S}$-MonCat_s for the category of $\mathcal{S}$-indexed strong monoidal categories, we thus have:

$$\begin{tikzcd} \mathcal{S}\text{-}\mathsf{MonCat}_s \arrow[r, bend right, "\mathbf{MON}"'] \arrow[r, phantom, "\perp"] & \mathcal{S}\text{-}\mathsf{Cat} \arrow[l, bend right, "\mathrm{FM}\circ-"'] \end{tikzcd}$$

In particular, this entails that $\mathbf{MON}$ preserves weighted limits, because it is a right adjoint. The properties of $\mathbf{MON}$ will be used later to describe descent categories for $\mathbf{MON}(\mathbb{C})$.

3.7. Completeness and Cocompleteness. Next, we consider adjoints to reindexing functors for an indexed monoidal category. Although only cocompleteness will be relevant to the proof of our main results, we include here a detailed analysis of both cases. This is to notice that there is a significant difference between the case of a right adjoint and that of a left adjoint, and the two situations are not dual to each other.

3.8. Definition. [Monoidal Cocompleteness] *Let $\mathbb{C}$ be an $\mathcal{S}$-indexed monoidal category. Then, $\mathbb{C}$ is said to have* monoidal $\mathcal{S}$-indexed coproducts *if every reindexing functor x^* has a monoidal left adjoint Σ_x, such that the Beck-Chevalley Condition (BCC) holds, and the unit and counit of the adjunction $\Sigma_x \dashv x^*$ are monoidal.*

If, in addition, $\mathbb{C}$ has finitely cocomplete fibres, then it will be called monoidally cocomplete.

Similarly, monoidal completeness is defined.

More briefly, for $\mathbb{C}$ to be monoidally (co)complete, we ask that each reindexing functor has a right (left) adjoint in the category MonCat, that the fibres are finitely (co)complete and that the Beck-Chevalley Condition (BCC) is satisfied. Notice that, in the rest of this paper, this notion will be used alongside the one of (co)completeness for indexed categories, as defined for example in [Joh02]. The two notions differ in that in the monoidal case, all aspects of the relevant adjunctions are required to be monoidal.

The following lemma appears in [Jan01], and characterises right adjoints to monoidal functors.

3.9. Lemma. *Let F be a monoidal functor. Then F has a right adjoint in* Cat *if and only if it has a right adjoint in* MonCat.

This implies that, for an $\mathcal{S}$-indexed monoidal category to be monoidally complete, we only need to ask that it is complete as an ordinary $\mathcal{S}$-indexed category. For left adjoints, the analogous statement does not hold.

Let us briefly revisit the examples of indexed monoidal categories from the previous section from the point of view of completeness and cocompleteness:

1. The standard indexing $J \mapsto \mathsf{C}/J$, even though it is a cocomplete indexed category, it is not monoidally cocomplete, since the left adjoints do not preserve products, and hence fail to be monoidal.

2. The indexing $\mathcal{E} \mapsto \mathrm{CSL}(\mathcal{E})$, $x \mapsto x^*$ becomes a complete indexed monoidal category, which is monoidally complete.

3. The indexed category Mod$(-)$, which assigns to a ring its category of modules, is complete and cocomplete as a Rng-indexed monoidal category.

4. The MonCat_s-indexed category Mod$(-)$ is monoidally complete, since reindexing has a right adjoint by (a 2-categorical) Kan extension. It is cocomplete, but not monoidally cocomplete.

5. Let $\mathcal{S}$ be a sub-2-category of Cat in which every functor $x : \mathsf{C} \to \mathsf{D}$ has a right adjoint y. Then, adjunction (3) says that End$(-)$ has $\mathcal{S}$-indexed coproducts. If in addition each category C in $\mathcal{S}$ is finitely cocomplete, then so is the category End(C). In this case, End$(-)$ is a cocomplete $\mathcal{S}$-indexed category, but it is not monoidally cocomplete, because the left adjoints to the reindexing functors are generally not monoidal.

We briefly turn to completeness and cocompleteness of the indexed category $\mathbf{MON}(\mathbb{C})$. Again, the situations for completeness and cocompleteness are not dual. In the first case we have:

3.10. Lemma. *If $\mathbb{C}$ is a monoidally complete $\mathcal{S}$-indexed monoidal category, then* $\mathbf{MON}(\mathbb{C})$ *is complete (as an ordinary $\mathcal{S}$-indexed category).*

Proof. Any 2-functor preserves adjunctions. This implies that the reindexing functors of $\mathbf{MON}(\mathbb{C})$ have right adjoints. The BCC is inherited likewise. It remains to verify that each fibre of $\mathbf{MON}(\mathbb{C})$ is finitely complete. It is, however, not hard to see that for each monoidal category D, the forgetful functor $\mathbf{Mon}(\mathsf{D}) \to \mathsf{D}$ creates finite limits. ■

For cocompleteness, we do get the required adjoints to reindexing functors and the BCC, but we don't get finitely cocomplete fibres. In order to get those, assumptions about the interaction of the tensor products with the colimits in the fibres are needed. Also, even if $\mathbf{MON}(\mathbb{C})$ is cocomplete, it does not follow automatically that $\mathbb{C}$ is monoidally cocomplete.

4. Descent Categories

In this section, we prove some results about descent categories. First, we give a presentation for the descent categories of an indexed monoidal category. As a consequence, we obtain the result that $\mathbf{MON}(\mathbb{C})$ is a stack when $\mathbb{C}$ is. This will be useful later, when we wish to deduce the descent properties of monads from those of endofunctors. After that, we prove a technical result about descent categories for cocomplete indexed categories. To be more precise, we show that the canonical functor $\Theta = \Theta_R : \mathsf{C}^K \to \mathsf{Desc}(R, \mathbb{C})$ has a left adjoint, under the assumption of the existence of a certain type of colimit in the fibre C^K. This colimit should be thought of as a glueing of descent data. This result follows by abstract considerations as well, namely by applying the adjoint functor theorem, but we need an explicit description of the left adjoint. Moreover, our proof is based on a fact about pseudo-limits that seems worth recording. The usefulness of this result lies in the fact that it provides an extra tool for proving effective descent: rather than showing that the canonical comparison functor $\Theta : \mathsf{C}^K \to \mathsf{Desc}(R, \mathbb{C})$ (which takes an object X to the family $(x_\gamma^* X, f_{\gamma\delta})$, where the $f_{\gamma\delta}$ are the isomorphisms arising from the indexed category structure) is an equivalence, one can now show that a certain adjunction is an adjoint equivalence.

As mentioned in the Introduction, this section is going to be rather technical. The results presented here will play a fundamental role in the proof of the main results in section 7. However, it is not essential to read all the proofs in great detail, in order to understand the results therein.

4.1. Descent for Monoidal Indexed Categories. We investigate the category of descent data for a strong monoidal indexed category, starting off with a lemma.

4.2. Lemma. *Let $\mathbb{C}$ be an $\mathcal{S}$-indexed strong monoidal category and let R be a family of the form $R = \{x_\gamma : J_\gamma \to K\}$. Then, the following hold:*

1. *The category $\mathsf{Desc}(R, \mathbb{C})$ is monoidal;*
2. *The functor $\Theta : \mathsf{C}^K \to \mathsf{Desc}(R, \mathbb{C})$ is strong monoidal;*
3. *The functors $U_\gamma : \mathsf{Desc}(R, \mathbb{C}) \to \mathsf{C}^{J_\gamma}$ are strong monoidal.*

Proof.

1. The category of descent data is a pseudo-limit, and MonCat_s has pseudo-limits. Or, for those preferring a more direct approach, we define the monoidal structure in $\mathsf{Desc}(R, \mathbb{C})$ by saying that the tensor product of two objects $X = (X_\gamma, f_{\gamma\delta}), Y = (Y_\gamma, g_{\gamma\delta})$ is $X \otimes Y = (X_\gamma \otimes Y_\gamma, h_{\gamma\delta})$, where $h_{\gamma\delta}$ is the composite
$$p_{\gamma\delta}^*(X_\gamma \otimes Y_\gamma) \cong p_{\gamma\delta}^* X_\gamma \otimes p_{\gamma\delta}^* Y_\gamma \xrightarrow{f_{\gamma\delta} \otimes g_{\gamma\delta}} q_{\gamma\delta}^* X_\delta \otimes q_{\gamma\delta}^* Y_\delta \cong q_{\gamma\delta}^*(X_\delta \otimes Y_\delta).$$
Notice that for this definition to make sense, it is essential that the reindexing functors are strong monoidal.
2. As stated earlier, the canonical functor Θ sends X to the family $(x_\gamma^* X, f_{\gamma\delta})$, where the $f_{\gamma\delta}$ are the isomorphisms arising from the indexed category structure. Of course, this is the mediating arrow from the pseudo-cone $(\mathsf{C}^K, x_\gamma^*)$ with vertex C^K to the pseudo-limit $\mathsf{Desc}(R, \mathbb{C})$; hence, it is a strong monoidal functor.
3. The functors $U_\gamma : \mathsf{Desc}(R, \mathbb{C}) \to \mathsf{C}^{J_\gamma}$, taking $(X_\gamma, g_{\gamma\delta})$ to X_γ, are the ones that define the universal pseudo-cone with vertex $\mathsf{Desc}(R, \mathbb{C})$; therefore, they are strong monoidal by definition.

Next, we have the following useful “interchange” result, which says that Desc and **MON** commute.

4.3. Lemma. *Let $\mathbb{C}$ be an $\mathcal{S}$-indexed strong monoidal category and R a family of maps with common codomain. Then, the functor Θ induces an equivalence of categories*

$$\mathsf{Desc}(R, \mathbf{MON}(\mathbb{C})) \simeq \mathbf{Mon}(\mathsf{Desc}(R, \mathbb{C})).$$

Proof. This can be proved directly, but it is much quicker to observe that $\mathsf{Desc}(R, \mathbb{C})$ is a pseudo-limit in MonCat_s, and that taking monoids preserves pseudo-limits since $\mathbf{Mon}$ is a representable functor. ■

The situation can be summarised by the following commutative diagram:

$$\begin{tikzcd}
 & \mathbf{Mon}(\mathsf{C}^K) \arrow[r] \arrow[dl] \arrow[d, "\mathbf{Mon}(\Theta)"] & \mathsf{C}^K \arrow[d, "\Theta"] \\
\mathsf{Desc}(R, \mathbf{MON}(\mathbb{C})) \arrow[r, "\simeq"] \arrow[dr] & \mathbf{Mon}(\mathsf{Desc}(R, \mathbb{C})) \arrow[r] \arrow[d, "\mathbf{Mon}(U_\gamma)"] & \mathsf{Desc}(R, \mathbb{C}) \arrow[d, "U_\gamma"] \\
 & \mathbf{Mon}(\mathsf{C}^{J_\gamma}) \arrow[r] & \mathsf{C}^{J_\gamma}.
\end{tikzcd}$$

The horizontal arrows on the right-hand side of the diagram are the forgetful functors, and the commutativity of the two squares is just the fact that we have a forgetful functor $\mathbf{MON}(\mathbb{C}) \to \mathbb{C}$. Also, note that the fact that $\mathsf{Desc}(R, -)$ is a functor applies here to the forgetful functor $\mathbf{MON}(\mathbb{C}) \to \mathbb{C}$.

¿From the point of view of the study of effective descent, we have the following important consequence:

4.4. Proposition. *Let $\mathcal{J}$ be a coverage on the base category $\mathcal{S}$. If $\mathbb{C}$ is a stack for $\mathcal{J}$, then so is* $\mathbf{MON}(\mathbb{C})$.

Proof. If $\mathbb{C}$ is a stack, then by definition, the functor $\Theta_R : \mathsf{C}^K \to \mathsf{Desc}(R, \mathbb{C})$ is an equivalence of categories for each family R in the coverage. Because of the previous lemma and the fact that $\mathbf{Mon}$, being a 2-functor, preserves equivalences of categories, the functor $\mathbf{Mon}(\mathsf{C}^K) \to \mathsf{Desc}(R, \mathbf{MON}(\mathbb{C}))$ is an equivalence as well. ■

A quick word about the converse of this proposition: it is false. As an example, take any indexed category which is not a stack, and apply the free monoidal category functor fibrewise. Since the free monoidal category on a category contains exactly one monoid, passing to monoids will give (up to isomorphism) the terminal indexed category, which is a stack.

4.5. Descent and Cocompleteness. We now present a technical result on descent in certain cocomplete indexed categories that, to our best knowledge, has not been noticed before. Although in the stated form it follows from the adjoint functor theorem, we will find use for an explicit construction later on.

4.6. Proposition. *Let $\mathbb{C}$ be a cocomplete $\mathcal{S}$-indexed category, such that the fibres C^J have small colimits (and not just the finite ones). Then, for any family $R = \{x_\gamma : J_\gamma \to K\}$, the canonical functor $\Theta_R : \mathsf{C}^K \to \mathsf{Desc}(R, \mathbb{C})$ has a left adjoint.*

The left adjoint should be thought of as gluing together the descent data to form an object in C^K, which is why we call it Glue. The picture is thus:

$$\begin{tikzcd} \mathsf{Desc}(R,\mathbb{C}) \arrow[r, bend left, "\mathrm{Glue}"] \arrow[r, phantom, "\perp"] & \mathsf{C}^K \arrow[l, bend left, "\Theta"] \end{tikzcd}.$$

The result follows from the following more general fact about pseudo-limits.

4.7. Lemma. *Let $T : \mathsf{X} \to \mathsf{Cat}$ be a pseudo-functor with the property that for each map x in X, the functor Tx has a left adjoint. Let M be any category and let $\phi : \Delta_{\mathsf{M}} \to T$ be a pseudo-cone over T with vertex M such that all components of ϕ (that is, all functors $\phi_X : \mathsf{M} \to TX$) have a left adjoint. If M has colimits of shape X, then the induced functor $\mathsf{M} \to PsLim(T)$ has a left adjoint as well.*

Proof. (Sketch.) We break up the proof in two cases: that where T is a product diagram and that where it is a pseudo-equaliser diagram. Since all pseudo-limits can be constructed from those, this will be sufficient. (See [Str80].)

First we deal with products (there is no difference between a genuine product and a pseudo-product, since the diagram is discrete). So, consider a family D_i of categories $(i \in I)$ and a cone $\phi_i : \mathsf{M} \to \mathsf{D}_i$, such that all ϕ_i have a left adjoint $\widetilde{\phi_i}$. Assuming that M has I-indexed products, we can define a left adjoint to the induced functor $\langle \phi_i \rangle : \mathsf{M} \to \prod_i \mathsf{D}_i$ by the assignment

$$(D_i)_{i\in I} \mapsto \coprod_i \widetilde{\phi_i}(D_i).$$

It is not hard to see that this is functorial and that the resulting functor is left adjoint to $\langle \phi_i \rangle$.

Next, we deal with pseudo-equalisers. Suppose we have a pseudo-equaliser diagram in Cat:

$$\begin{tikzcd} \mathsf{E} \arrow[r, "e"] & \mathsf{A} \arrow[r, bend left, "F"] \arrow[r, bend right, "G"'] & \mathsf{B} \end{tikzcd}$$

in which both F and G have left adjoints, denoted by $\tilde{F}$ and $\tilde{G}$, respectively. Recall that a canonical representative for the pseudo-equaliser is a category E whose objects are pairs (A, θ), where A is an object of A and $\theta : FA \to GA$ is an isomorphism in B, and a map $(A, \theta) \to (B, \sigma)$ consists of a map $k : A \to B$ such that $Gk \circ \theta = \sigma \circ Fk$.

If M is a category, $\phi : \mathsf{M} \to \mathsf{A}$ is a functor and $\gamma : F\phi \to G\phi$ is a natural isomorphism, then there is an induced map $K : \mathsf{M} \to \mathsf{E}$ making $eK = \phi$; this map K is simply given by the assignment $C \mapsto (\phi C, \gamma_C)$.

Suppose now that ϕ has a left adjoint $\tilde{\phi}$ and M has coequalisers. We wish to show that K has a left adjoint $\tilde{K}$. On an object (A,θ) of E, we define $\tilde{K}(A,\theta)$ by the coequaliser diagram

$$\begin{tikzcd} \tilde{\phi}\tilde{F}FA \arrow[rr, "\tilde{\phi}\epsilon^F_A"] \arrow[d, "\tilde{\phi}\tilde{F}\theta"'] & & \tilde{\phi}A \arrow[r] & \tilde{K}(A,\theta) \\ \tilde{\phi}\tilde{F}GA \arrow[r, "\tilde{\phi}\mu_A"'] & \tilde{\phi}\phi\tilde{\phi}A \arrow[ur, "\epsilon^{\phi}_{\tilde{\phi}A}"'] & & \end{tikzcd} \tag{4}$$

where $\mu_A : \tilde{F}GA \to \phi\tilde{\phi}A$ is the transpose of $GA \to F\phi\tilde{\phi}A$, which in turn is obtained as the composite $GA \xrightarrow{G\eta^{\phi}_A} G\phi\tilde{\phi}A \xrightarrow{\gamma_{\tilde{\phi}A}} F\phi\tilde{\phi}A$. The various ϵ's and η's are counits and units of the various adjunctions, with superscripts indicating which adjunction is meant; so, for example, η^{ϕ} is the unit of $\tilde{\phi} \dashv \phi$, etcetera. On maps, $\tilde{K}$ is defined using the universal property and the naturality of the maps involved.

It is now a long diagram chase to verify the required adjointness. ∎

Proof of Proposition 4.6. This is now easy: if the indexed category is cocomplete, then all reindexing functors have left adjoints. This means that, in the pseudo-diagram of which $\mathsf{Desc}(R,\mathbb{C})$ is a pseudo-limit, all maps have left adjoints. Moreover, C^K is the vertex of a pseudo-cone over this diagram, where the the reindexing functors form the components, which by assumption have left adjoints. Now, if C^K has colimits of shape $\mathcal{R}$, then we get the desired left adjoint Glue, by Lemma 4.7. ∎

We shall make use of the following description of the functor Glue. As explained in section 2, we may identify $\mathsf{Desc}(R,\mathbb{C})$ with $\mathsf{PsCone}(*,\mathbb{C}R)$, the category of pseudo-cones over $\mathbb{C}R$ with vertex $*$, where $\mathbb{C}R$ is the composite of which $\mathsf{Desc}(R,\mathbb{C})$ is the pseudo-limit. Assume that $R = \{x_\gamma : J_\gamma \to K \mid \gamma \in \Gamma\}$ is a sieve (there is no loss of generality in doing so).

Fix an object X of the category $\mathsf{Desc}(R,\mathbb{C})$. That is, $X = (X_\gamma)_{\gamma\in\Gamma}$ is a family of objects X_γ in C^{J_γ}, together with coherent isomorphisms $\phi_f : X_\gamma \to f^*(X_\delta)$ for each map f with $x_\delta \circ f = x_\gamma$.

Now, X induces a functor $\mathcal{R} \to \mathsf{C}^K$, where $\mathcal{R}$ is the full subcategory of the slice $\mathcal{S}/K$ on the objects of R. This functor sends an object x_γ of $\mathcal{R}$ to the object $\Sigma_{x_\gamma}X_\gamma$, and a map $f : x_\gamma \to x_\delta$ to the composite

$$\Sigma_{x_\gamma}X_\gamma \xrightarrow{\Sigma_{x_\gamma}\phi_f} \Sigma_{x_\gamma}f^*X_\delta \xrightarrow{\cong} \Sigma_{x_\delta}\Sigma_f f^*X_\delta \xrightarrow{\Sigma_{x_\delta}\epsilon} \Sigma_{x_\delta}X_\delta$$

It is tedious (but otherwise straightforward) to check that this construction is functorial in X, so we get a functor

$$\mathsf{PsCone}(*,\mathbb{C}R) \times \mathcal{R} \to \mathsf{C}^K$$

which, by transposing and taking colimits, gives the desired left adjoint to Θ:

$$\mathsf{Desc}(R, \mathbb{C}) = \mathsf{PsCone}(*, \mathbb{C}R) \longrightarrow (\mathsf{C}^K)^{\mathcal{R}} \xrightarrow{Colim} \mathsf{C}^K.$$

In this situation, we will also have the following result, which will be used in the proof of Theorem 7.1, to show that a certain indexed category is a stack: it gives a sufficient condition for the unit of the adjunction Glue $\dashv \Theta$ to be an isomorphism.

4.8. Lemma. *With the same assumptions and notation as above, assume moreover that, for every $x_\gamma \in R$, the unit of $\Sigma_{x_\gamma} \dashv x_\gamma^*$ is an isomorphism and that the reindexing functors preserve colimits of shape $\mathcal{R}$. Then, the unit of* Glue $\dashv \Theta$ *is an isomorphism as well.*

Proof. We feel that there should be some abstract reason why this is the case, but a proof by hand suffices our need, and we did not investigate this further.

Let $X = (X_\gamma, m_f)$ be an object of $\mathsf{Desc}(R, \mathbb{C})$. That is, for each $\gamma \in \Gamma$ we have X_γ in C^{J_γ}, and for each f as in the diagram

$$\begin{tikzcd} J_\gamma \arrow[rr, "f"] \arrow[dr, "x_\gamma"'] & & J_\delta \arrow[dl, "x_\delta"] \\ & K & \end{tikzcd}$$

we have an isomorphism $m_f : f^* X_\delta \to X_\gamma$, subject to the usual coherence conditions. We apply the functor Glue, as described above. To the result we wish to apply Θ. In order to see what the result is, we first investigate what happens when we apply a reindexing functor to Glue(X). So, fix an index $\sigma \in \Gamma$. The reindexing functor x_σ^* preserves colimits of shape $\mathcal{R}$, so that x_σ^*Glue(X) can be computed by taking the colimit in C^{J_σ} of the $\mathcal{R}$-shaped diagram which sends an object $x_\gamma : J_\gamma \to K$ to the object $x_\sigma^* \Sigma_{x_\gamma} X_\gamma$ and a map f as above to the map

$$x_\sigma^* \Sigma_{x_\gamma} X_\gamma \xrightarrow{x_\sigma^* \Sigma_{x_\gamma} m_f^{-1}} x_\sigma^* \Sigma_{x_\gamma} f^* X_\delta \xrightarrow{\cong} x_\sigma^* \Sigma_{x_\delta} \Sigma_f f^* X_\delta \xrightarrow{x_\sigma^* \Sigma_{x_\delta} \epsilon} x_\sigma^* \Sigma_{x_\delta} X_\delta.$$

We will show that the object X_σ is a colimiting cocone over this diagram.

Let's compute first what $x_\sigma^* \Sigma_{x_\delta} X_\delta$ is. Form the pullback in $\mathcal{S}$:

$$\begin{tikzcd} M = J_{u(\delta,\sigma)} \arrow[r, "z"] \arrow[d, "y"'] & J_\sigma \arrow[d, "x_\sigma"] \\ J_\delta \arrow[r, "x_\delta"'] & K. \end{tikzcd}$$

The BCC gives us a canonical isomorphism $x_\sigma^* \Sigma_{x_\delta} X_\delta \cong \Sigma_z y^* X_\delta$. Because R is a sieve, the pullback M is of the form $J_{u(\delta,\sigma)}$ for some index $u(\delta, \sigma) \in \Gamma$. Hence, we have isomorphisms $X_{u(\delta,\sigma)} \cong z^* X_\sigma$ and $X_{u(\delta,\sigma)} \cong y^* X_\delta$. We can then form the composite

$$x_\sigma^* \Sigma_{x_\delta} X_\delta \to \Sigma_z y^* X_\delta \to \Sigma_z X_{u(\delta,\sigma)} \to X_\sigma. \tag{5}$$

If we look at the special case where we apply x_σ^* to $\Sigma_{x_\sigma} X_\sigma$, then we use the fact that $x_\sigma^* \Sigma_{x_\sigma} \cong 1$ and get $x_\sigma^* \Sigma_{x_\sigma} X_\sigma \cong X_\sigma$.

It is readily checked that this construction is functorial: if f is a map with $x_\gamma = x_\delta f$, then x_σ^* sends the induced map $\Sigma_{x_\gamma} X_\gamma \to \Sigma_{x_\delta} X_\delta$ to the map $x_\sigma^* \Sigma_{x_\gamma} X_\gamma \to x_\sigma^* \Sigma_{x_\delta} X_\delta$ over X_σ. Therefore, the composites in (5) form a cocone with vertex X_σ. Moreover, one of the components of this cocone is an isomorphism. Hence, we can conclude that the colimit is isomorphic to X_σ.

We have shown that for an object X in $\mathsf{Desc}(R, \mathbb{C})$, we have $x_\sigma^* \mathrm{Glue}(X) \cong U_\sigma(X)$ for any σ in Γ, where U_σ is the canonical projection functor. It is straightforward to show, using the naturality of the maps involved in the construction, that for a map of descent objects $t : X \to Y$, we get an induced transformation between the diagrams associated to X_σ and Y_σ, and that the induced map between the colimits $X_\sigma \to Y_\sigma$ is exactly the component t_σ of t. This shows that there is an isomorphism of functors $U_\sigma \Theta \mathrm{Glue} = x_\sigma^* \mathrm{Glue} \cong U_\sigma$.

We conclude the proof by remarking that we have two endofunctors on $\mathsf{Desc}(R, \mathbb{C})$, the identity and $\Theta \circ \mathrm{Glue}$, which become isomorphic upon composition with U_σ for each $\sigma \in \Gamma$. That means that we have two naturally isomorphic pseudo-cones over the diagram of which $\mathsf{Desc}(R, \mathbb{C})$ is the pseudo-limit, so that the universal property gives that 1 and $\Theta \circ \mathrm{Glue}$ are isomorphic. ∎

5. Endofunctors and Monads

In this section we start the investigation of the assignments $\mathsf{C} \mapsto \mathrm{End}(\mathsf{C})$ and $\mathsf{C} \mapsto \mathrm{Mnd}(\mathsf{C})$, where C is a category and $\mathrm{End}(\mathsf{C})$ and $\mathrm{Mnd}(\mathsf{C})$ denote the categories of endofunctors and monads on C, respectively. The main question we address is: under which circumstances can this be viewed as an indexed category, and what are the (co)completeness properties of such an indexed category?

Of course, in order to talk about an indexed category, we should first of all consider a base category $\mathcal{S}$. This will be a sub-2-category of Cat (size issues are thoroughly ignored), on which we shall impose conditions to ensure the properties we will need.

5.1. Endofunctors as an indexed category. As we saw in Example 3.3.5, any adjunction

$$\begin{tikzcd} \mathsf{D} \arrow[r, bend left, "x"] \arrow[r, phantom, "\perp"] & \mathsf{C} \arrow[l, bend left, "y"] \end{tikzcd}$$

between categories induces an adjunction between the associated categories of endofunctors:

$$\begin{tikzcd} \mathrm{End}(\mathsf{D}) \arrow[r, bend left, "x-y"] \arrow[r, phantom, "\perp"] & \mathrm{End}(\mathsf{C}), \arrow[l, bend left, "y-x"] \end{tikzcd}$$

where the right adjoint is monoidal. So, if we assume that every functor in $\mathcal{S}$ has a right adjoint in Cat, we can then consider the $\mathcal{S}$-indexed monoidal indexed category $\mathrm{End}(-)$

which takes a category C to $\mathrm{End}(\mathsf{C})$ and a functor $x : \mathsf{D} \to \mathsf{C}$ (with a right adjoint y) to $y - x : \mathrm{End}(\mathsf{C}) \to \mathrm{End}(\mathsf{D})$.

Moreover, under these assumptions, we can define a trivial indexing $\mathbb{U}$, by mapping a category C to itself, and a functor x as above to its right adjoint y. Trivially, all reindexing functors of $\mathbb{U}$ have left adjoints (given by the x's). This is analogous to a dual version of the trivial indexing on the category of toposes, and will prove very useful in translating properties of the base $\mathcal{S}$ into properties of the indexed category $\mathrm{End}(-)$. It is also used in the following definition.

5.2. Definition. *The category $\mathcal{S}$ is said to* admit indexed endofunctors *when (i) $\mathcal{S}$ is closed under pullbacks in* Cat *and (ii) the indexed category $\mathbb{U}$ is $\mathcal{S}$-cocomplete.*

The point of this definition is that it is a sufficient condition for the operation $\mathrm{End}(-)$ to be a cocomplete $\mathcal{S}$-indexed category. Note that $\mathcal{S}$, as a sub-2-category of Cat, is really a 2-category, but we do not take the 2-categorical structure into account, when speaking of $\mathcal{S}$-indexed categories.

5.3. Proposition. *Let $\mathcal{S}$ be such that it admits indexed endofunctors. Then*

1. *the assignment $\mathsf{C} \mapsto \mathrm{End}(\mathsf{C})$ is the object part of an $\mathcal{S}$-cocomplete $\mathcal{S}$-indexed category* $\mathrm{End}(-)$*;*

2. $\mathrm{End}(-)$ *is an indexed monoidal category.*

Proof.

1. The pseudo-functoriality is immediate from the fact that right adjoints compose uniquely up to unique isomorphism. For cocompleteness, we already mentioned that the left adjoint to a reindexing functor $y - x$ is given by $x - y$. The BCC follows from (and is in fact equivalent to) that for the $\mathcal{S}$-indexed category $\mathbb{U}$. Finally, since $\mathbb{U}$ is cocomplete, each category C in $\mathcal{S}$ is finitely cocomplete. Therefore, so are the categories $\mathrm{End}(\mathsf{C})$, because colimits of endofunctors are computed pointwise.

2. Each $\mathrm{End}(\mathsf{C})$ is monoidal: the tensor is composition, the unit is the identity functor. The monoidality of the reindexing functors was already indicated.

■

5.4. Remark. However, notice that $\mathrm{End}(-)$ is not monoidally cocomplete; the left adjoint $x-y$ to the reindexing map $y-x$ is in general not monoidal. In spite of this, the existence of the left adjoints will prove important in the rest of the paper.

5.5. Monads as an indexed category. As a particular consequence of Proposition 5.3, we have that $\mathrm{End}(-)$ is a pseudo-functor $\mathcal{S}^{op} \to \mathsf{MonCat}$. Composing with the functor **Mon** of section 3.1 and observing that monoids in $\mathrm{End}(\mathsf{C})$ are monads on C, we get at once the following result:

5.6. LEMMA. *Let $\mathcal{S}$ be a base category admitting indexed endofunctors. Then the assignment* $\mathsf{C} \mapsto \mathrm{Mnd}(\mathsf{C})$ *is the object part of an* $\mathcal{S}$*-indexed category* $\mathrm{Mnd}(-)$.

Since the indexed category $\mathrm{End}(-)$ is cocomplete but not monoidally cocomplete, we cannot expect, in general, that $\mathrm{Mnd}(-)$ is cocomplete. (Loosely speaking, the left adjoints to reindexing functors will not preserve monoids; incidentally, they preserve comonoids.) However, we can identify additional conditions on $\mathcal{S}$ which make $\mathrm{Mnd}(-)$ cocomplete. These are collected in the following definition.

5.7. DEFINITION. *The base category $\mathcal{S}$ is said to* admit indexed monads *if it admits indexed endofunctors, all functors in $\mathcal{S}$ are fully faithful and their right adjoints preserve colimits.*

These conditions may appear awkward, at first sight, but they are forced upon us by the arguments that follow. However, they are not too restrictive, at least for the class of examples we have in mind.

Because colimits of monads need not exist, in general, we can not expect the indexed category $\mathrm{Mnd}(-)$ to be cocomplete. Under these assumptions on $\mathcal{S}$, though, we get the following partial result:

5.8. PROPOSITION. *If $\mathcal{S}$ admit indexed monads, then* $\mathrm{Mnd}(-)$ *has $\mathcal{S}$-indexed coproducts.*

PROOF. For clarity we split these up into separate lemmas.

For a category C in the base $\mathcal{S}$, define $\mathrm{End}_*(\mathsf{C})$ to be the category of pointed endofunctors of C, i.e. endofunctors T equipped with a natural transformation $1 \to T$.

5.9. LEMMA. *The construction* $\mathsf{C} \mapsto \mathrm{End}_*(\mathsf{C})$ *is an $\mathcal{S}$-indexed category and there is an $\mathcal{S}$-indexed forgetful functor* $\mathrm{End}_*(-) \to \mathrm{End}(-)$.

PROOF. The reindexing functors are the same as for $\mathrm{End}(-)$; given a pointed endofunctor $1 \to T$, its image is given by the composite $1 \to yx \to yTx$, where the first map is the unit of $x \dashv y$. ∎

Note that $\mathrm{End}_*(\mathsf{C})$ is the same as the coslice $1/\mathrm{End}(\mathsf{C})$. Moreover, $\mathrm{End}_*(-)$ is the coslice $1/\mathrm{End}(-)$, where 1 is the trivial monoidal indexed category. $\mathrm{End}_*(-)$ is a monoidal indexed category and the monoidal structure is preserved by the forgetful functor from $\mathrm{End}_*(-)$ to $\mathrm{End}(-)$. Moreover, since representable functors preserve weighted limits (and coslicing is one), we get at once:

5.10. LEMMA. *The forgetful functor* $\mathrm{End}_*(-) \to \mathrm{End}(-)$ *induces an equivalence of indexed categories*

$$\mathbf{MON}(\mathrm{End}_*(-)) \simeq \mathbf{MON}(\mathrm{End}(-)) = \mathrm{Mnd}(-).$$

The advantage of shifting our attention from $\mathrm{End}(-)$ to $\mathrm{End}_*(-)$ is that the latter has better cocompleteness properties.

5.11. LEMMA. $\mathrm{End}_*(-)$ *is a monoidally cocomplete $\mathcal{S}$-indexed strong monoidal category.*

PROOF. Given a full inclusion $x : \mathsf{D} \to \mathsf{C}$ with right adjoint y, we first define a functor $\Sigma_x : \mathrm{End}_*(\mathsf{D}) \to \mathrm{End}_*(\mathsf{C})$. Take a pointed endofunctor $\eta : 1 \to T$ on D. Then, form the pushout

$$\begin{tikzcd} xy \arrow[r, "\epsilon"] \arrow[d, "x\eta y"'] & 1 \arrow[d] \\ xTy \arrow[r] & \Sigma_x T \end{tikzcd}$$

in the category $\mathrm{End}(\mathsf{C})$ (this pushout is taken pointwise). Clearly, $\Sigma_x T$ has a point, namely the right-hand map in the pushout diagram. It is rather laborious, but otherwise straightforward, to show that each Σ_x is also strong monoidal (notice, though, that for this it is crucial that the right adjoint functors preserve pushouts, which follows from Definition 5.7; in fact, this is the only form of right exactness we need for the result to hold). This operation is made into a functor by using the universal property of the pushout.

The same universal property is also used to verify that the functor Σ_x is left adjoint to $y - x$. The BCC is inherited from that for $\mathcal{S}$.

Finally, the fact that reindexing functors are strong monoidal is an easy exercise. ∎

Now we have everything we need to prove Proposition 5.8. The functors Σ_x are the required left adjoints to the reindexing functors, and the BCC holds. ∎

RUNNING EXAMPLE. Let $\mathcal{S}$ be the following category. The objects are the categories GSet_n, for each $n \geq 0$, together with the category $\mathsf{Gset}_\infty = \mathsf{Gset}$. The morphisms are the inclusions $i_{n,m} : \mathsf{Gset}_n \to \mathsf{GSet}_m$ (where $n \leq m \in \mathbb{N} \cup \infty$).

Then, $\mathcal{S}$ is closed under pullbacks in Cat (given two subcategories $\mathsf{Gset}_n, \mathsf{Gset}_m$, their pullback is the intersection GSet_k, with $k = m \wedge n$). Since each object of $\mathcal{S}$ is a topos, all the cocompleteness requirements are fulfilled. All morphisms are fully faithful and are left Kan extensions, so have right adjoints. These right adjoints preserve all colimits, because they have further right adjoints.

This shows that $\mathcal{S}$ satisfies all the requirements necessary to view monads as an indexed category.

5.12. A TOPOLOGY ON THE BASE CATEGORY. In the next section, we are going to investigate descent conditions for the indexed categories $\mathrm{End}(-)$ and $\mathrm{Mnd}(-)$. It will soon be clear that we have to restrict our attention to a specific class of endofunctors, namely the stable ones. However, at this point we can already introduce a topology on $\mathcal{S}$, which we shall work with in the rest of this paper. In order to describe this topology, we shall make use of the indexed category $\mathbb{U}$ introduced before Definition 5.2.

5.13. DEFINITION. *We call a family $R = \{\mathsf{D}_\gamma \to \mathsf{C}\}$ in $\mathcal{S}$* generating *if $\mathbb{U}$ is of descent for R.*

While this is a concise definition, there is a more practical and intuitive description possible, based on the fact that R induces, for each object X of C, a diagram in C, together with a cocone with vertex X.

In more detail, write $R = \{x_\gamma : \mathsf{D}_\gamma \to \mathsf{C} \mid \gamma \in \Gamma\}$ for the sieve and y_γ for the right adjoint to x_γ.

For a certain object X of C, we have the descent object $\Theta(X)$. To see what $\Theta(X)$ is, consider a map f in the subcategory $\mathcal{R}$ generated by R:

$$\begin{tikzcd} \mathsf{D}_\gamma \arrow[rr, "f"] \arrow[dr, "x_\gamma"'] & & \mathsf{D}_\delta \arrow[dl, "x_\delta"] \\ & \mathsf{C}. & \end{tikzcd}$$

All these maps have right adjoints, which we will denote by $x_\gamma \dashv y_\gamma, x_\delta \dashv y_\delta, f \dashv f_*$. Because $x_\gamma = x_\delta f$, there is a canonical isomorphism $y_\gamma \cong f_* y_\delta$. Now, using the characterisation of $\mathsf{Desc}(R, -)$ given in the Introduction, we can describe $\Theta(X)$ as the family $y_\gamma X$, with γ running over all elements of Γ, together with, for each f as in the diagram above, the canonical isomorphism $y_\gamma X \to f_* y_\delta X$. (Mind: f_* is the reindexing functor along f here.)

After applying the functor Glue discussed in section 4.5 to this descent object, we can take a colimit in C of the diagram defined by the following functor $\mathcal{R} \to \mathsf{C}$, which sends an object $x_\gamma : \mathsf{D}_\gamma \to \mathsf{C}$ to the object $x_\gamma y_\gamma X$ and a map f as in the above diagram to the composite

$$x_\gamma y_\gamma X \xrightarrow{=} x_\delta f y_\gamma X \xrightarrow{\cong} x_\delta f f_* y_\delta X \xrightarrow{x_\delta \epsilon y_\delta} x_\delta y_\delta X.$$

The counits $x_\gamma y_\gamma X \to X$ form a cocone over this diagram, which we will refer to as the *canonical cocone with vertex X*. Now we have:

5.14. PROPOSITION. *A family R is generating precisely when for every object X, the canonical cocone with vertex X is colimiting.*

PROOF. This is immediate from the description of the functor Glue given in section 4.5: the value of this functor on the descent object associated to X is precisely the colimit of the above diagram. ∎

Note that the diagram constructed above is in fact a filtered diagram.

Using this characterisation of generating families, it is not difficult to see that the generating families form a Grothendieck topology on the base category; identities are trivial, pullback-stability follows from the fact that filtered colimits commute with pullbacks, and transitivity follows from the fact that colimits distribute over each other.

We wish to think of a category C with a generating family $R = \{x_\gamma : \mathsf{D}_\gamma \to \mathsf{C}\}$ as a category which is approximated by the categories D_γ. In a certain sense, this can be viewed as a geometrical property of a category.

RUNNING EXAMPLE. Let us verify what a generating family is when $\mathcal{S}$ is the category consisting of the categories GSet_n. There is only one non-trivial generating family, namely the family of inclusions $i_{n,\infty} : \mathsf{Gset_n} \to \mathsf{GSet}$, where $n \neq \infty$. Why is this a generating family? Simply because any globular set A, written

$$\cdots \rightrightarrows A_2 \rightrightarrows A_1 \rightrightarrows A_0$$

is the colimit of the diagram

$$
\begin{tikzcd}
\cdots \arrow[r, shift left] \arrow[r, shift right] & 0 \arrow[r, shift left] \arrow[r, shift right] \arrow[d] & 0 \arrow[r, shift left] \arrow[r, shift right] \arrow[d] & A_0 \arrow[d] & = & i_{0,\infty} r_{0,\infty} A \arrow[d] \\
\cdots \arrow[r, shift left] \arrow[r, shift right] & 0 \arrow[r, shift left] \arrow[r, shift right] \arrow[d] & A_1 \arrow[r, shift left] \arrow[r, shift right] \arrow[d] & A_0 \arrow[d] & = & i_{1,\infty} r_{1,\infty} A \arrow[d] \\
\cdots \arrow[r, shift left] \arrow[r, shift right] & A_2 \arrow[r, shift left] \arrow[r, shift right] \arrow[d] & A_1 \arrow[r, shift left] \arrow[r, shift right] \arrow[d] & A_0 \arrow[d] & = & i_{2,\infty} r_{2,\infty} A \arrow[d] \\
 & \vdots & \vdots & \vdots & & \ddots
\end{tikzcd}
$$

6. Stability and Descent

Now that we have described the setting in which $\mathrm{End}(-)$ and $\mathrm{Mnd}(-)$ can be viewed as indexed categories, it is natural to wonder whether either of them is a stack for the topology on the base category described in the previous section. In the case of endofunctors, this would mean, informally, that for a covering family $R = \{x_\gamma : \mathsf{D}_\gamma \to \mathsf{C}\}$ in the base category, the category $\mathrm{End}(\mathsf{C})$ can be recovered from the categories $\mathrm{End}(\mathsf{D}_\gamma)$. In general, this is not the case. We shall first give a simple example to show why we cannot expect to get a stack, and then single out a class of well-behaved endofunctors, called *stable* endofunctors. This section investigates their properties and shows that they form a sub-$\mathcal{S}$-indexed monoidal category of $\mathrm{End}(-)$.

6.1. AN EXAMPLE. Let us start by giving an example to illustrate why we cannot expect $\mathrm{End}(-)$ to be a stack.

For our base category $\mathcal{S}$, we take the category with three objects, 1, the terminal category, Set and $\mathsf{Set} \times \mathsf{Set}$. There will be two non-trivial maps, namely the functors $i_0, i_1 : \mathsf{Set} \to \mathsf{Set} \times \mathsf{Set}$, given by $i_0 A = (A, \emptyset)$ and $i_1 A = (\emptyset, A)$. These functors are fully faithful and have right adjoints π_0, π_1, respectively, given by the projections.

Since every object (A, B) of $\mathsf{Set} \times \mathsf{Set}$ is the coproduct of $\pi_0 i_0(A, B) = (A, \emptyset)$ and $\pi_1 i_1(A, B) = (\emptyset, B)$, the pair $R = \{i_0, i_1\}$ is now a generating family of $\mathsf{Set} \times \mathsf{Set}$, in the sense of Definition 5.13.

An object of $\mathsf{Desc}(R, \mathrm{End})$ is just a pair of endofunctors on Set. The canonical functor $\Theta_R : \mathrm{End}(\mathsf{Set} \times \mathsf{Set}) \to \mathsf{Desc}(R, \mathrm{End})$ sends an endofunctor T of $\mathsf{Set} \times \mathsf{Set}$ to its restrictions $(\pi_0 T i_0, \pi_1 T i_1)$.

Now, we can see why $\mathrm{End}(-)$ is not a stack for this generating family: consider, for example, the endofunctor $W \in \mathrm{End}(\mathsf{Set} \times \mathsf{Set})$, given by

$$W(A, B) = (A, A + B).$$

We have $\Theta_R(W) = (1, 1) = \Theta_R(1)$, but clearly W and the identity are not isomorphic as functors on $\mathsf{Set} \times \mathsf{Set}$, hence Θ_R can not be an equivalence.

6.2. Stability for Endofunctors. The example we have just presented indicates why we cannot expect arbitrary endofunctors to be recoverable. Therefore, we look for a property that endofunctors (and monads) can have, which allows them to be recovered from their restriction along a generating family. This property will be called stability.

Throughout this section, we shall assume that $\mathcal{S}$ admits indexing of monads.

6.3. Definition. [Stability of Endofunctors] *Let* C, D *be categories in* $\mathcal{S}$, $x : \mathsf{D} \to \mathsf{C}$ *a functor with right adjoint* y *and* T *an endofunctor on* C.

1. *We say that* T *is* x-stable *if the map*

$$yT\epsilon : yTxy \to yT$$

is an isomorphism (here, $\epsilon : xy \to 1$ *is the counit of* $x \dashv y$*).*

2. *If* R *is a family of arrows in* $\mathcal{S}$ *with common codomain* C*, then* T *is called* R*-stable if* T *is* x*-stable for all* x *in* R *and all binary and ternary pullbacks from* R*.*

3. *Moreover, if* $\mathcal{J}$ *is a topology on the base* $\mathcal{S}$*, then* T *is called* $\mathcal{J}$-stable*, or simply* stable*, if* T *is* x*-stable for all possible* x *in all covering families on* C*.*

In this terminology, we can now see that the endofunctor W in the example above is not i_1-stable, since $\pi_1 W(A, B) = A + B$, whereas $\pi_1 W i_1 \pi_1(A, B) = \pi_1 W(\emptyset, B) = B$.

6.4. Lemma. *Let* $T \in \mathrm{End}(\mathsf{C})$ *be stable and let* $x : \mathsf{D} \to \mathsf{C}$ *be a fully faithful map with a right adjoint* y*. Then, the restriction* yTx *is stable.*

Proof. Take a D-cover R and a map $p : \mathsf{E} \to \mathsf{D}$ in R with right adjoint q. We need to show that the map

$$q(yTx)\epsilon : q(yTx)pq \to q(yTx)$$

is an isomorphism. But T is (xp)-stable, and, inserting some (unique) isomorphisms, the result follows. ∎

6.5. Lemma. *Let* $S \in \mathrm{End}(\mathsf{D})$ *be stable (for a certain coverage) and let* $x : \mathsf{D} \to \mathsf{C}$ *be a map with right adjoint* y*. Then, the lifting* xSy *is stable.*

Proof. Consider a family R which covers C, and a map $p : \mathsf{E} \to \mathsf{C}$ in R with right adjoint q. Consider the pullback

$$\begin{tikzcd} \mathsf{D}' \arrow[r, "m"] \arrow[d, "p'"'] & \mathsf{E} \arrow[d, "p"] \\ \mathsf{D} \arrow[r, "x"'] & \mathsf{C} \end{tikzcd}$$

in $\mathcal{S}$. Because $p \in R$ and covers are pullback-stable, the map p' is in a D-cover R'. By assumption, S is p'-stable, so the map $q'Sp'q' \to q'S$ is an isomorphism (where q' is the right adjoint to p'). Now the result follows using the BCC. ∎

These lemmas suggest the following definition:

6.6. Definition. *Let* $\mathrm{StEnd}(-)$ *denote the full* $\mathcal{S}$*-indexed subcategory of* $\mathrm{End}(-)$ *on the stable endofunctors. Similarly,* $\mathrm{StMnd}(-)$ *denotes the full* $\mathcal{S}$*-indexed subcategory of* $\mathrm{Mnd}(-)$ *on the stable monads (where, of course, a stable monad is a monad which is stable as an endofunctor).*

6.7. Lemma. *The indexed category* $\mathrm{StEnd}(-)$ *is* $\mathcal{S}$*-cocomplete.*

Proof. For any map $x : \mathsf{D} \to \mathsf{C}$ we have the picture:

$$\begin{tikzcd} \mathrm{StEnd}(\mathsf{D}) \arrow[r, shift left, "x-y"] \arrow[r, phantom, "\perp"] \arrow[d] & \mathrm{StEnd}(\mathsf{C}) \arrow[l, shift left, "y-x"] \arrow[d] \\ \mathrm{End}(\mathsf{D}) \arrow[r, shift left, "x-y"] \arrow[r, phantom, "\perp"] & \mathrm{End}(\mathsf{C}); \arrow[l, shift left, "y-x"] \end{tikzcd}$$

the BCC is inherited from $\mathrm{End}(-)$. Finally, colimits in the fibres $\mathrm{StEnd}(\mathsf{C})$ are computed as colimits in $\mathrm{End}(\mathsf{C})$. To see this, assume that $T = \operatorname{colim} T_i \in \mathrm{End}(\mathsf{C})$ and that $x : \mathsf{D} \to \mathsf{C}$ is a functor with right adjoint y, where y preserves colimits. Then, we have a commutative diagram

$$\begin{tikzcd} yTxy \arrow[r, "\cong"] \arrow[d, "="'] & yT \arrow[d, "="] \\ y(\operatorname{colim} T_i)xy \arrow[r, "\cong"] \arrow[d, "\cong"'] & y(\operatorname{colim} T_i) \arrow[d, "\cong"] \\ \operatorname{colim} yT_ixy \arrow[r, "\cong"'] & \operatorname{colim} yT_i, \end{tikzcd}$$

in which the bottom map is an isomorphism because of the stability of the T_i's. So, the top map is an isomorphism as well, hence T is stable. ∎

As a consequence of this fact, we can apply Proposition 4.6, which gives us at once the following diagram:

$$\begin{tikzcd} \mathsf{Desc}(R, \mathrm{StEnd}) \arrow[r, shift left, "\mathrm{Glue}"] \arrow[r, phantom, "\perp"] \arrow[d] & \mathrm{StEnd}(\mathsf{C}) \arrow[l, shift left, "\Theta_R"] \arrow[d] \\ \mathsf{Desc}(R, \mathrm{End}) \arrow[r, shift left, "\mathrm{Glue}"] \arrow[r, phantom, "\perp"] & \mathrm{End}(\mathsf{C}), \arrow[l, shift left, "\Theta_R"] \end{tikzcd}$$

which can be summarised by saying that the gluing operation on endofunctors restricts to stable endofunctors.

6.8. Lemma. *Let* $\mathcal{S}$ *admit indexed monads. Then, the indexed category* $\mathrm{StEnd}(-)$ *is monoidal, with tensor product inherited from* $\mathrm{End}(-)$*.*

PROOF. The identity functor, which is the unit for the tensor, is stable: given any map $x : \mathsf{D} \to \mathsf{C}$ with right adjoint y, the map $y1_{\mathsf{C}}\epsilon : y1_{\mathsf{C}}xy \to y1_{\mathsf{C}}$ is an isomorphism when the unit of $x \dashv y$ is.

For the tensor product, consider two stable endofunctors $T, S \in \mathrm{End}(\mathsf{C})$. We have to show that $yTSxy \to yTS$ is an isomorphism. But in the following naturality square, stability of T and of S forces the three remaining arrows to be isomorphisms.

$$\begin{tikzcd} yTxySxy \arrow[r, "\cong"] \arrow[d, "\cong"'] & yTxyS \arrow[d, "\cong"] \\ yTSxy \arrow[r] & yTS \end{tikzcd}$$

■

Of course, the concept of stability readily extends from endofunctors to monads: a monad is stable if the underlying endofunctor is. Therefore, we can take the indexed category $\mathrm{StMnd}(-)$ to be $\mathbf{MON}(\mathrm{StEnd}(-))$.

RUNNING EXAMPLE. The free strict ω-category monad T on globular sets is stable for the family of inclusions $\mathsf{GSet_n} \to \mathsf{GSet}$. To see this, recall that T is given by

$$(TA)_m = \coprod_{\pi \in \mathbf{pd}(m)} \mathsf{GSet}[\hat{\pi}, A].$$

Because $\hat{\pi}$ has no k-cells for $k > m$, this means that

$$(Ti_{m,\infty}r_{m,\infty}A)_m = \mathsf{GSet}[\hat{\pi}, i_{m,\infty}r_{m,\infty}A] = (TA)_m.$$

This shows that TA and $Ti_{m,\infty}r_{m,\infty}A$ agree in dimension m, and the argument is easily extended to lower dimensions. Therefore T is $i_{m,\infty}$-stable. Informally, this just says that in order to construct $(TA)_m$, one only needs information about $r_{m,\infty}A$, and not about the k-cells of A for $k > m$, so one might just as well assume that $A_k = \emptyset$ for $k > m$.

7. Main Results

We have introduced all concepts and facts necessary to state the main theorems. These tell us that the indexed categories of stable endofunctors and stable monads over a base category which admits indexed monads are of effective descent for a family R in the base when this family R is a generating in the sense of Definition 5.13. Informally, this says that when objects descend, then so do stable endofunctors and stable monads. It turns out that the converse also holds. The first theorem states this for endofunctors.

7.1. THEOREM. *Let $\mathcal{S}$ be a sub-2-category of* Cat *which admits indexed monads and let $R = \{x_\gamma : \mathsf{D}_\gamma \to \mathsf{C}\}$ be a family of maps with common codomain. Then, the following are equivalent:*

1. *the indexed category* StEnd *is of descent for* R*;*

2. *the family* R *is generating (i.e. is a covering family for the topology described in section 5.12).*

Proof. Assume first that StEnd is of descent for R. To prove that R is generating, we consider the identity functor on C, which is stable. Because StEnd is of descent for R, the functors Θ_R and Glue are both equivalences, whence $\mathrm{Glue}(\Theta_R(1)) \cong 1$ in $\mathrm{StEnd}(\mathsf{C})$. Clearly $\Theta_R(1)$ is (isomorphic to) the family (1_γ) of identity functors on the categories D_γ. Now, recall from the remarks preceding Lemma 4.8 that Glue is computed as a colimit of the liftings $x_\gamma 1_\gamma y_\gamma \cong x_\gamma y_\gamma$. To say that the colimit of this diagram is isomorphic to 1 is the same as saying that R is a cover, by the remark after Definition 5.13.

For the other direction, we start by observing that the unit of the adjunction $\mathrm{Glue} \dashv \Theta_R$ is an isomorphism, as we saw in Lemma 4.8.

Next, we show that the counit is also an isomorphism. To this end, take a stable endofunctor T on C. The functor Θ_R sends this to the family $(y_\gamma T x_\gamma)$, equipped with the canonical descent data. The glueing functor will form the colimit of the liftings $x_\gamma y_\gamma T x_\gamma y_\gamma$. More precisely, it will take the colimit of the diagram of shape $\mathcal{R}$ that sends an object $x_\gamma : \mathsf{D}_\gamma \to \mathsf{C}$ of $\mathcal{R}$ to $x_\gamma y_\gamma T x_\gamma y_\gamma$, and a map f as in

$$\begin{tikzcd} \mathsf{D}_\gamma \arrow[rr, "f"] \arrow[dr, "x_\gamma"'] & & \mathsf{D}_\delta \arrow[dl, "x_\delta"] \\ & \mathsf{C} & \end{tikzcd}$$

to the composite

$$x_\gamma y_\gamma T x_\gamma y_\gamma = x_\delta f y_\gamma T x_\delta f y_\gamma \xrightarrow{\cong} x_\delta f f_* y_\delta T x_\delta f f_* y_\delta \to x_\delta y_\delta T x_\delta y_\delta.$$

By stability, we have $x_\gamma y_\gamma T x_\gamma y_\gamma \cong x_\gamma y_\gamma T$ for each $\gamma \in \Gamma$, so we get, for each f as above, a commutative diagram

$$\begin{tikzcd} x_\gamma y_\gamma T x_\gamma y_\gamma = x_\delta f y_\gamma T x_\delta f y_\gamma \arrow[r, "\cong"] \arrow[d, "\cong"'] & x_\delta f f_* y_\delta T x_\delta f f_* y_\delta \arrow[r] & x_\delta y_\delta T x_\delta y_\delta \arrow[d, "\cong"] \\ x_\gamma y_\gamma T \arrow[r] & x_\delta f f_* y_\delta T \arrow[r] & x_\delta y_\delta T. \end{tikzcd} \tag{6}$$

Now, when we compute $\mathrm{Glue}(\Theta_R(T))$, we have to compute the colimit of the diagram with the $x_\gamma y_\gamma T x_\gamma y_\gamma$ (a typical fragment of which is the top row of (6) above), but the commutativity of the above diagram and the fact that the vertical arrows are isomorphisms means that we can just as well compute the colimit over the $x_\gamma y_\gamma T$ (i.e. the diagram of which the bottom row of 6 is a typical fragment), and since R is generating, we know that this colimit is isomorphic to T (see section 5.12). ∎

Next, we have the result for monads.

7.2. THEOREM. *Let $\mathcal{S}$ be a category admitting indexed monads and let $R = \{x_\gamma : \mathsf{D}_\gamma \to \mathsf{C}\}$ be a family of maps with common codomain. Then the following are equivalent:*

1. *the indexed category* StMnd *is of descent for R;*
2. *the family R is generating.*

PROOF. This follows from Theorem 7.1; if stable monads are of descent for R, then we may again consider the identity monad to show that R must be a covering family. For the other direction we use proposition 4.4, since a stable monad is by definition a monoid in the category of stable endofunctors. ∎

The following corollary is now immediate from the definitions.

7.3. COROLLARY. *Let $\mathcal{J}$ be any coverage on $\mathcal{S}$. Then, the following are equivalent:*

1. *the indexed category* StEnd *is a stack for $\mathcal{J}$;*
2. *the indexed category* StMnd *is a stack for $\mathcal{J}$;*
3. *the (trivial) indexed category $\mathbb{U}$ is a stack for $\mathcal{J}$;*
4. *every family in $\mathcal{J}$ is generating (in the sense of Definition 5.13).*

7.4. REMARK. A natural question is, whether the condition of stability is necessary to obtain the results of section 7. The answer is: no, it is not necessary and it can be weakened, probably in different ways. Let us indicate one possible direction. Let $R = \{x_\gamma : \mathsf{D}_\gamma \to \mathsf{C}\}$ be a generating family. Instead of asking that an endofunctor T on C is stable for all x_γ, one could ask that for each γ there exists a δ such that x_γ factors through x_δ and that the action of T on D_γ is determined by its action on D_δ.

8. An application: Cheng's interleaving

We can now use the results of the last section to view the results of Cheng [Che04] in our general framework. Let us first recall that the category Coll of collections is defined as the slice category $\mathsf{GSet}/T1$, where GSet is the category of globular sets (with terminal object 1), and T is the strict ω-category monad on it. Its objects are therefore diagrams of the form

$$\begin{tikzcd} \cdots \arrow[r, shift left] \arrow[r, shift right] & A_2 \arrow[r, shift left] \arrow[r, shift right] \arrow[d] & A_1 \arrow[r, shift left] \arrow[r, shift right] \arrow[d] & A_0 \arrow[d] \\ \cdots \arrow[r, shift left] \arrow[r, shift right] & T(1)_2 \arrow[r, shift left] \arrow[r, shift right] & T(1)_1 \arrow[r, shift left] \arrow[r, shift right] & T(1)_0, \end{tikzcd} \tag{7}$$

where each row is a globular set. The *underlying globular set* of such a collection is the one represented by the top row in the diagram.

We shall denote by Coll_n ($n \in \mathbb{N}$) the category of n-dimensional collections, i.e. those collections whose underlying globular set is n-dimensional; that is, A_m is the empty set for each $m > n$. There are obvious inclusion functors $J_n : \mathsf{Coll}_n \to \mathsf{Coll}$, and $J_{n,m} : \mathsf{Coll}_n \to \mathsf{Coll}_m$, for $n < m \in \mathbb{N}$. These are clearly full and faithful, and they have right adjoints $Tr_n : \mathsf{Coll} \to \mathsf{Coll}_n$ and $Tr_{n,m} : \mathsf{Coll}_m \to \mathsf{Coll}_n$, respectively, which simply truncate a collection (or an m-dimensional one) to dimension n, replacing all the higher dimensional sets by $\emptyset$.

For simplicity, we shall write Coll_∞ for Coll, and we shall speak only of functors $J_{n,m}$ and $Tr_{n,m}$. These will coincide with the functors J_n and Tr_n when $m = \infty$.

Since finite colimits of collections are defined pointwise, Coll_n is a finitely cocomplete category for all $n \in \mathbb{N} \cup \{\infty\}$; moreover, the truncation functors preserve colimits (and, in particular, pushouts), since they have right adjoints.

We now define a category $\mathcal{S}$ by taking the categories Coll_n as objects (for $n \in \mathbb{N} \cup \{\infty\}$) and the inclusion functors $J_{n,m}$ as maps.

8.1. Lemma. *The 2-category $\mathcal{S}$ admits indexed monads.*

Proof. It is clear by what we just said that each functor in $\mathcal{S}$ is full and faithful and has a right adjoint, and that each category in $\mathcal{S}$ has finite colimits and these are preserved by the right adjoints. So, all we need to show is that $\mathcal{S}$ is closed under pullbacks in Cat and the BCC holds.

For pullbacks, note that, given a diagram

$$\begin{tikzcd} & \mathsf{Coll}_n \arrow[d, shift right, "J_{n,p}"'] \arrow[d, phantom, "\dashv"] \\ \mathsf{Coll}_m \arrow[r, shift left, "J_{m,p}"] \arrow[r, phantom, "\perp"] & \mathsf{Coll}_p, \arrow[u, shift right, "Tr_{n,p}"'] \arrow[l, shift left, "Tr_{m,p}"] \end{tikzcd}$$

its pullback is given by Coll_q, where $q = n \wedge m$, with projections given by the inclusion functors $J_{q,m}$ and $J_{q,n}$ (of course, the functor $J_{n,n}$ is the identity for any $n \in \mathbb{N}$). For one such pullback

$$\begin{tikzcd} \mathsf{Coll}_q \arrow[r, "J_{q,n}"] \arrow[d, "J_{q,m}"'] & \mathsf{Coll}_n \arrow[d, "J_{n,p}"] \\ \mathsf{Coll}_m \arrow[r, "J_{m,p}"'] & \mathsf{Coll}_p, \end{tikzcd}$$

verifying the BCC amounts to showing that the following square commutes in Cat:

$$\begin{tikzcd} \mathsf{Coll}_n \arrow[r, "Tr_{q,n}"] \arrow[d, "J_{n,p}"'] & \mathsf{Coll}_q \arrow[d, "J_{q,m}"] \\ \mathsf{Coll}_p \arrow[r, "Tr_{m,p}"'] & \mathsf{Coll}_m. \end{tikzcd}$$

To this end, notice that for A in Coll_n one has

$$Tr_{m,p}J_{n,p}(A) = \begin{cases} J_{n,m}(A) & n \leq m \\ Tr_{m,n}(A) & n > m \end{cases}$$

and in both cases the result is equal to $J_{q,m}Tr_{q,n}(A)$, since $q = m \wedge n$. ■

Now, we consider on $\mathcal{S}$ the family of maps

$$R = \{J_n : \mathsf{Coll}_n \to \mathsf{Coll} \mid n \in \mathbb{N}\}.$$

Given a collection X in Coll, it is clear that the cocone

$$J_nTr_nX \xrightarrow{\epsilon_n} X$$

over the diagram $J_nTr_nX \to J_mTr_mX$ ($n < m$) is colimiting; for suppose we have another cocone

$$J_nTr_nX \xrightarrow{\alpha_n} Y,$$

then the first n objects of the underlying globular set of J_nTr_nX are the same as those of X, and α_n fixes the action of a map $\alpha : X \to Y$ on them.

Therefore, R is a generating family, in the sense of Definition 5.13; hence, StMnd$(-)$ is of descent for R.

Now, we concentrate on the two monads in [Che04]: Opd and Contr. First, we shall show that they are stable monads, thus deducing that they can be recovered as the glueing of their n-dimensional restrictions Opd_n and Contr_n. Then, we shall show how the computations of Cheng make it possible to form a coproduct of the two families (Opd_n) and (Contr_n) in $\mathsf{Desc}(R, \mathrm{StMnd})$, whose glueing is precisely the monad OWC of operads with contraction described in her paper.

To see that Opd and Contr are stable monads, we need to show that the maps

$$Tr_n\mathrm{Opd}\,\epsilon_n : Tr_n\mathrm{Opd}\,J_nTr_n \to Tr_n\mathrm{Opd}$$
$$Tr_n\mathrm{Contr}\,\epsilon_n : Tr_n\mathrm{Contr}\,J_nTr_n \to Tr_n\mathrm{Contr}$$

are isomorphisms for all $n \in \mathbb{N}$. For a collection A as in (7) above, we have

$$Tr_n\mathrm{Opd}\,J_nTr_n(A) = Tr_n\mathrm{Opd}\left(\begin{array}{ccccccc} \emptyset & \rightrightarrows & A_n & \cdots & A_1 & \rightrightarrows & A_0 \\ \cdots\downarrow & & \downarrow & & \downarrow & & \downarrow \\ T(1)_{n+1} & \rightrightarrows & T(1)_n & \cdots & T(1)_1 & \rightrightarrows & T(1)_0 \end{array}\right)$$

and

$$Tr_n\mathrm{Opd}\,(A) = Tr_n\mathrm{Opd}\left(\begin{array}{ccccccc} A_{n+1} & \rightrightarrows & A_n & \cdots & A_1 & \rightrightarrows & A_0 \\ \cdots\downarrow & & \downarrow & & \downarrow & & \downarrow \\ T(1)_{n+1} & \rightrightarrows & T(1)_n & \cdots & T(1)_1 & \rightrightarrows & T(1)_0 \end{array}\right).$$

Notice that the counit $\epsilon_n : J_nTr_nA \to A$ is the identity in the first n components. Moreover, Cheng points out that the k-cells of the free operad on a collection A are determined only by the j-cells of A for $j \leq k$. From this, we can deduce that

$$Tr_n\mathrm{Opd}\,J_nTr_n(A) \cong Tr_n\mathrm{Opd}(A).$$

A similar argument holds for Contr, since the k-cells of the free collection with contraction over A depend solely on the j-cells of A for $j < k$.

So, the monad Opd is the glueing of its restrictions $\mathrm{Opd}_n = Tr_n \mathrm{Opd}\, J_n : \mathsf{Coll}_n \to \mathsf{Coll}_n$, and the monad Contr is the glueing of the family $\mathrm{Contr}_n = Tr_n \mathrm{Contr}\, J_n : \mathsf{Coll}_n \to \mathsf{Coll}_n$. In particular, the families $(\mathrm{Opd}_n)_{n\in\mathbb{N}}$ and $(\mathrm{Contr}_n)_{n\in\mathbb{N}}$ form objects of $\mathsf{Desc}(R, \mathrm{StMnd}) \simeq \mathrm{StMnd}(\mathsf{Coll})$. We do not specify the coherence maps because, by the aforementioned properties of the monads Opd and Contr these are always the identity. We are now going to show that Cheng's interleaving construction of the operad-with-contraction monad OWC is substantially calculating the glueing of the coproducts of these two families.

As we have already mentioned, in general the indexed category StMnd is not $\mathcal{S}$-cocomplete, since the fibres are not finitely cocomplete (after all, colimits of monads are hard to find, and even when they should exist, stability does not automatically follow). However, we can say something in the specific example at hand. First of all, recall from [Kel80] that, given two monads S and T on a category C, one can form the product $U : P \to \mathsf{C}$ of the two forgetful functors U_T : T-Alg $\to \mathsf{C}$ and U_S : S-Alg $\to \mathsf{C}$ in Cat/C (here, T-Alg is the category of Eilenberg-Moore algebras of T, and likewise for S). If U is monadic, then the monad arising from it is called the *algebraic coproduct* of S and T in $\mathrm{Mnd}(\mathsf{C})$, and in particular it is a coproduct of S and T.

Now, using the notations in [Che04], it follows that our monad Opd_n is the restriction to Coll_n of the monad Opd^n resulting from the monadic adjunction

$$\begin{tikzcd} \mathsf{Coll} \arrow[r, shift left] \arrow[r, phantom, "\perp"] & \mathrm{Opd}_0 \arrow[l, shift left] \arrow[r, shift left] \arrow[r, phantom, "\perp"] & \mathrm{Opd}_1 \arrow[l, shift left] \arrow[r, shift left] \arrow[r, phantom, "\perp"] & \cdots \arrow[l, shift left] \arrow[r, shift left] \arrow[r, phantom, "\perp"] & \mathrm{Opd}_n \arrow[l, shift left] \end{tikzcd}$$

Likewise, we have $\mathrm{Contr}_n = Tr_n \mathrm{Contr}^n J_n$, where Contr^n is the monad arising from the monadic adjunction

$$\begin{tikzcd} \mathsf{Coll} = \mathrm{Contr}_0 \arrow[r, shift left] \arrow[r, phantom, "\perp"] & \mathrm{Contr}_1 \arrow[l, shift left] \arrow[r, shift left] \arrow[r, phantom, "\perp"] & \cdots \arrow[l, shift left] \arrow[r, shift left] \arrow[r, phantom, "\perp"] & \mathrm{Contr}_n \arrow[l, shift left] \end{tikzcd}$$

Moreover, the "interleaving construction" in that paper shows a monadic adjunction

$$\begin{tikzcd} \mathsf{Coll} \arrow[r, shift left] \arrow[r, phantom, "\perp"] & \mathrm{OWC}_{n,n} = \mathrm{OWC}_n \arrow[l, shift left] \end{tikzcd} \tag{8}$$

where OWC_n is precisely the product of the two forgetful functors $\mathrm{Opd}_n \to \mathsf{Coll}$ and $\mathrm{Contr}_n \to \mathsf{Coll}$ in $\mathsf{Cat}/\mathsf{Coll}$.

Therefore, the monad OWC^n arising from (8) is the (algebraic) coproduct of Contr^n and Opd^n. In particular, this coproduct is preserved by the restriction to Coll_n, since an S-algebra structure on an object $J_n(X)$ in Coll (for S=Opd^n, Contr^n or OWC^n) is the same as a $Tr_n S\, J_n$-algebra on X. We denote by OWC_n the restriction $Tr_n \mathrm{OWC}^n J_n$ of the monad OWC^n to Coll_n.

Then, the identity maps make the collection (OWC_n) into an object of $\mathsf{Desc}(R, \mathrm{StMnd})$, which is the coproduct of (Opd_n) and (Contr_n). Finally, we see that the glueing functor Glue : $\mathsf{Desc}(R, \mathrm{StMnd}) \to \mathrm{StMnd}(\mathsf{Coll})$ takes this family to a monad OWC : $\mathsf{Coll} \to \mathsf{Coll}$ which is the coproduct of Opd and Contr, i.e. the free operad-with-contraction monad.

References

[Ama01] G. Amato and J. Lipton. *Indexed categories and bottom-up semantics of logic programs.* Lecture Notes in Computer Science, Vol. 2250. Springer Verlag, New York-Berlin, 2001.

[Asp93] A. Asperti and A. Corradini. A categorical modal for logic programs: indexed monoidal categories. In: J.W. de Bakker and W.P. de Roever, Ed. REX Workshop, 110–137, Lecture Notes in Computer Science, Vol. 666. Springer Verlag, New York-Berlin, 1993.

[Bar85] M. Barr and C. Wells. *Toposes, Triples, Theories.* Springer-Verlag, New York-Berlin, 1985.

[Ben89] J. Bénabou. Some remarks on 2-categorical algebra. *Bulletin de la Société de Belgique* (1989).

[Bor81] F. Borceux. *Handbook of categorical algebra.* Encyclopedia of mathematics and its applications, Vol. 50,51. Cambridge University Press, 1981.

[Che04] E. Cheng. Monad interleaving: a construction of the operad for Leinster's weak ω-categories. Available at: `http://www.math.uchicago.edu/~eugenia/interleaving.pdf`

[Jan94] G. Janelidze and W. Tholen. Facets of descent I. *Applied Categorical Structures* 2 (1994), 1–37.

[Jan97] G. Janelidze and W. Tholen. Facets of descent II. *Applied Categorical Structures* 5 (1997), 229–248.

[Jan01] G. Janelidze and G.M. Kelly. A note on actions of a monoidal category. *Theory and Applications of Categories* 9 (2001), 61–91.

[Joh02] P.T. Johnstone. *Sketches of an Elephant: a Topos Theory Compendium.* Oxford University Press, 2002.

[Joy84] A. Joyal and M. Tierney. An extension of the Galois theory of Grothendieck. *Memoirs of the American. Mathematical Society* 309 (1984).

[Kel80] G.M. Kelly. A unified treatment for transfinite constructions for free algebras, free monoids, colimits, associated sheaves, and so on. *Bulletin of the Australian Mathematical Society* 22 (1980).

[Kel82] G.M. Kelly. *Basic concepts of enriched category theory.* LMS Lecture Note Series, Vol. 64. Cambridge University Press, 1982.

[Kel89] G.M. Kelly. Elementary observations on 2-categorical limits. *Bulletin of the Australian Mathematical Society* 39 (1989), 301–317.

[Kel04] G.M. Kelly and S. Lack. Monoidal functors generated by adjunctions, with applications to transport of structure. *Fields Institute Communications* 43 (2004), University of Sidney.

[Lei04] T. Leinster. *Higher Operads, Higher Categories.* LMS Lecture Note Series, Vol. 298. Cambridge University Press, 2004.

[Mac98] S. Mac Lane. *Categories for the working mathematician.* Graduate Texts in Mathematics, Vol. 5. Springer Verlag, New York-Berlin, 1998

[Str80] R. Street. Fibrations in bicategories. *Cahiers de Topologie et Géométrie Différentielle Catégoriques* 21 (1980), 111–160.

[Str00] R. Street. The petit topos of globular sets. *Journal of Pure and Applied Algebra* 154 (2000), 299–315.

Department of Mathematics and Statistics, University of Ottawa
585 King Edward Avenue, Ottawa, ON, K1N 6N5, Canada

Department of Mathematics, Utrecht University
Budapestlaan 6, 3584 CD Utrecht, The Netherlands

Email: pieter.hofstra@mathstat.uottawa.ca
marchi@math.uu.nl

This article may be accessed via WWW at http://www.tac.mta.ca/tac/ or by anonymous ftp at ftp://ftp.tac.mta.ca/pub/tac/html/volumes/16/24/16-24.{dvi,ps,pdf}

THEORY AND APPLICATIONS OF CATEGORIES (ISSN 1201-561X) will disseminate articles that significantly advance the study of categorical algebra or methods, or that make significant new contributions to mathematical science using categorical methods. The scope of the journal includes: all areas of pure category theory, including higher dimensional categories; applications of category theory to algebra, geometry and topology and other areas of mathematics; applications of category theory to computer science, physics and other mathematical sciences; contributions to scientific knowledge that make use of categorical methods.
Articles appearing in the journal have been carefully and critically refereed under the responsibility of members of the Editorial Board. Only papers judged to be both significant and excellent are accepted for publication.
The method of distribution of the journal is via the Internet tools WWW/ftp. The journal is archived electronically and in printed paper format.

SUBSCRIPTION INFORMATION. Individual subscribers receive (by e-mail) abstracts of articles as they are published. Full text of published articles is available in .dvi, Postscript and PDF. Details will be e-mailed to new subscribers. To subscribe, send e-mail to tac@mta.ca including a full name and postal address. For institutional subscription, send enquiries to the Managing Editor, Robert Rosebrugh, rrosebrugh@mta.ca.

INFORMATION FOR AUTHORS. The typesetting language of the journal is TeX, and LaTeX2e is the preferred flavour. TeX source of articles for publication should be submitted by e-mail directly to an appropriate Editor. They are listed below. Please obtain detailed information on submission format and style files from the journal's WWW server at http://www.tac.mta.ca/tac/. You may also write to tac@mta.ca to receive details by e-mail.