



HAL
open science

Central Submonads and Notions of Computation

Titouan Carette, Louis Lemonnier, Vladimir Zamdzhiev

► **To cite this version:**

Titouan Carette, Louis Lemonnier, Vladimir Zamdzhiev. Central Submonads and Notions of Computation. 2022. hal-03662565v3

HAL Id: hal-03662565

<https://hal.science/hal-03662565v3>

Preprint submitted on 20 Jul 2022 (v3), last revised 10 Oct 2023 (v4)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

Central Submonads and Notions of Computation

TITOUAN CARETTE, ENS Paris-Saclay, Université Paris-Saclay, LMF, Inria, France

LOUIS LEMONNIER, ENS Paris-Saclay, Université Paris-Saclay, LMF, Inria, France

VLADIMIR ZAMDZHEV, ENS Paris-Saclay, Université Paris-Saclay, LMF, Inria, France

The notion of "centre" has been introduced for many algebraic structures in mathematics. A notable example is the centre of a monoid which always determines a commutative submonoid. Monads in category theory are important algebraic structures that may be used to model computational effects in programming languages and in this paper we show how the notion of centre may be extended to strong monads acting on symmetric monoidal categories. We show that the centre of a strong monad \mathcal{T} , if it exists, determines a commutative submonad \mathcal{Z} of \mathcal{T} , such that the Kleisli category of \mathcal{Z} is isomorphic to the premonoidal centre (in the sense of Power and Robinson) of the Kleisli category of \mathcal{T} . We provide three equivalent conditions which characterise the existence of the centre of \mathcal{T} and we show that every strong monad on many well-known naturally occurring categories does admit a centre, thereby showing that this new notion is ubiquitous. We also provide a computational interpretation of our ideas which consists in giving a refinement of Moggi's monadic metalanguage. The added benefit is that this allows us to immediately establish a large class of contextually equivalent programs for computational effects that are described via monads that admit a non-trivial centre by simply considering the richer syntactic structure provided by the refinement.

CCS Concepts: • **Theory of computation** → **Operational semantics; Denotational semantics.**

Additional Key Words and Phrases: strong monads, commutative monads, computational effects, centre, category theory, denotational semantics, operational semantics

1 INTRODUCTION

The importance of monads in programming semantics has been demonstrated in seminal work by Moggi [Moggi 1989, 1991]. The main idea is that monads allow us to introduce computational effects (e.g., state, input/output, recursion, probability, continuations) into pure type systems in a controlled way. The mathematical development surrounding monads has been very successful and it directly influenced modern programming language design through the introduction of monads as a programming abstraction into languages such as Haskell, Scala and others (see [Benton 2015]). The results reported in this paper follow the same spirit as the above developments. We start with a simple and natural mathematical question about monads, then we develop the required mathematical theory and provide the answer to our original question. Finally, we present a computational interpretation of this idea and illustrate its usefulness.

The mathematical question that we ask is simple:

Is there a suitable notion of centre that may be formulated for monads? (1)

The answer is analogous to the same question that may be asked for monoids. Just as every monoid M (on **Set**) has a centre, which is a *commutative* submonoid of M , so does every (canonically strong) monad \mathcal{T} on **Set** and the centre of \mathcal{T} is a *commutative* submonad of \mathcal{T} . Generalising away from the category **Set**, the answer is a little bit more complicated, but still analogous to monoids: not every monoid object M on a symmetric monoidal category \mathbf{C} has a centre, and neither does every (strong) monad on \mathbf{C} . However, we show that under some basic reasonable assumptions, the answer is "yes", and we show that for many categories of interest all strong monads are centralisable and so the newly introduced notion of centre is ubiquitous. The computational significance of these ideas is easy to understand: given a computational effect, such that not *every* pair of effectful operations commute (i.e., the order of sequencing matters), identify only those effectful operations which do commute with any other possible effectful operation. The effectful operations that satisfy this property are called *central*. The computational interpretation of our ideas consists in giving a

refinement of Moggi’s monadic metalanguage that makes it syntactically clear which are the central effectful operations and organising them into a commutative submonad of the original one. This allows us to easily establish large classes of contextually equivalent programs, simply by considering the additional syntactic structure, and this in turn may be used for program optimisation.

1.1 The Centre of a Strong Monad: Computationally

In a seminal paper [Moggi 1991], Eugenio Moggi introduced a computational interpretation for (strong) monads, which are algebraic structures previously discovered by mathematicians in the field of category theory. In particular, Moggi showed that if one wishes to introduce computational effects into a type system with terms that may have finitely many free variables (i.e., terms of the form $x_1 : A_1, \dots, x_n : A_n \vdash M : B$), then this may be achieved via the structure of a *strong monad*¹. This allows us to introduce computational effects in a controlled and systematic way by using monadic sequencing. For instance, in Haskell, this may be achieved via the familiar `do` notation (see Listing 1).

1	do		do
2	x <- op1		y <- op2
3	y <- op2		x <- op1
4	f x y		f x y

Listing 1. Two examples of monadic sequencing in Haskell.

Depending on the choice of monad, the two code fragments in Listing 1 need not have the same effect. However, when the monad in question is not only strong, but also *commutative*, then the two code fragments in Listing 1 would *always* have the same effect. Thus, in a commutative monad, we can safely incorporate program transformations that exchange the order of monadic sequencing which can be useful for program optimisation. Therefore, the commutativity of the monad matters from a computational perspective.

However, many monads that are computationally interesting in computer science are strong, but not commutative. Examples include the continuations monad, state monad, list monad, the writer monad for a monoid M (when the monoid is not commutative) and many others. Nevertheless, just because a monad is strong, but not commutative, it does not mean that *every* monadic operation does not satisfy the above commutative property with respect to any other monadic operation. Even if the monad is not commutative, it might be the case that there are *specific* choices for the monadic operations `op1` and `op2` where the above code fragments are equivalent. In fact, it is useful to determine which monadic operations do satisfy this additional property, because then we can partially recover some of the benefits of a commutative monad.

Our main idea is to approach this problem by taking inspiration from the algebraic notion of “centre”, which we show how to extend to strong monads. From a computational perspective, the centre of the strong monad consists of all *central* monadic operations, i.e., those monadic operations for which the commutativity property above holds against any other (non-central) monadic operation. Thus, in Listing 1, if at least one of `op1` or `op2` is central, then the two code fragments would have the same effect, even if the monad is not commutative. In fact, the central monadic operations determine a commutative submonad of the original one, so we may utilise them using a similar syntax.

The computational interpretation of our ideas consists in giving a refinement of Moggi’s monadic metalanguage² and we provide an excerpt of it in Figure 1. In addition to the usual monadic

¹The monad strength is crucial for terms with *more than one* free variable; see [Moggi 1991] for more details.

²The monadic syntax we chose is standard, but not identical to Moggi’s syntax.

$$\begin{array}{c}
\frac{\Gamma, x : A \vdash M : B}{\Gamma \vdash \lambda x^A. M : A \rightarrow B} \quad \frac{\Gamma \vdash M : A}{\Gamma \vdash \text{ret}_{\mathcal{Z}} M : \mathcal{Z}A} \quad \frac{\Gamma \vdash M : \mathcal{Z}A \quad \Gamma, x : A \vdash N : \mathcal{Z}B}{\Gamma \vdash \text{do}_{\mathcal{Z}} x \leftarrow M ; N : \mathcal{Z}B} \\
\frac{\Gamma \vdash M : A \rightarrow B \quad \Gamma \vdash N : A}{\Gamma \vdash MN : B} \quad \frac{\Gamma \vdash M : \mathcal{Z}A}{\Gamma \vdash \iota M : \mathcal{T}A} \quad \frac{\Gamma \vdash M : \mathcal{T}A \quad \Gamma, x : A \vdash N : \mathcal{T}B}{\Gamma \vdash \text{do}_{\mathcal{T}} x \leftarrow M ; N : \mathcal{T}B}
\end{array}$$

Fig. 1. An excerpt of the refinement of Moggi's monadic metalanguage.

types $\mathcal{T}A$, which represent effectful computations for the ambient \mathcal{T} monad, we also introduce the types $\mathcal{Z}A$, which represent *central* effectful computations. The monadic unit allows us to see any pure term $M : A$ as a *central* computation, represented by the term $\text{ret}_{\mathcal{Z}} M : \mathcal{Z}A$, and any term $M : \mathcal{Z}A$ may also be seen to be of type $\mathcal{T}A$, which is represented by the $\iota M : \mathcal{T}A$ term (in the sequel, we use ι for the submonad inclusion $\iota : \mathcal{Z} \Rightarrow \mathcal{T}$, which explains the choice of notation). The monadic unit for the \mathcal{T} monad corresponds to $\iota(\text{ret}_{\mathcal{Z}} M)$, so there is no need to include a specific term for it. We also add two terms, via the familiar *do* notation, for monadic sequencing for both monads. This more refined syntax allows us to easily establish and keep track of many contextually equivalent terms in a systematic way which is otherwise difficult to achieve via standard methods when the monad \mathcal{T} is not commutative. For example, we can prove that $\llbracket \text{do}_{\mathcal{T}} x \leftarrow \iota M ; \text{do}_{\mathcal{T}} y \leftarrow N ; P \rrbracket = \llbracket \text{do}_{\mathcal{T}} y \leftarrow N ; \text{do}_{\mathcal{T}} x \leftarrow \iota M ; P \rrbracket$. The ιM term is necessarily central, whereas the term N may not be. It follows that ιM commutes with any other effectful operation and thus we can safely reorder those code fragments guided by the syntax, without carrying any further verifications involving the semantics. Other contextual equivalences may also be established (e.g., central effectful operations are closed under composition), see §6.3.

1.2 Centres of Algebraic Structures

The notion of *centre* has been introduced for many algebraic structures in mathematics (e.g., monoids, groups, rings) and the definition is usually quite simple. For example, given a monoid $(M, \cdot, 1)$, its centre, denoted $Z(M)$, is the set

$$Z(M) \stackrel{\text{def}}{=} \{x \in M \mid \forall y \in M. x \cdot y = y \cdot x\}. \quad (2)$$

In fact, the centre $Z(M)$, equipped with the same monoid structure as that of M , is a *commutative submonoid* of M . Similarly, the centre of a group G is a commutative (i.e., abelian) subgroup of G and the centre of a ring R is a commutative subring of R . Indeed, taking the centre of an algebraic structure is a well-known construction and a simple way to produce a commutative substructure of the original one. This naturally leads to the main question that we posed above: what about the centre of a monad? After all, monads are algebraic structures, and in fact, they are precisely monoids in categories of endofunctors. How can we then construct the centre of a monad?

The answer to this question is more complicated compared to the simple construction of the centre of a monoid. First, we have to recognise that in (2), there is an implicit symmetric monoidal structure involved, which is just the obvious cartesian structure of the category **Set**. The equation in (2) can be written in the following equivalent, but more convoluted, way:

$$(- \cdot -) \circ (x \times y) = (- \cdot -) \circ (y \times x), \quad (3)$$

where we see the elements x, y as maps $x, y : 1 \rightarrow M$ from the singleton set into M . However, Equation (3) gives a more categorical way of representing the same idea, and makes it clear that the definition of the centre of a monoid has to do with the interaction between the monoid operation and the (symmetric) monoidal structure of the category **Set**. For the same reason, we see that the

definition of the centre of a monad has to do with how the monad interacts with the monoidal structure of its base category. Because of this, we have to consider *strong* monads [Kock 1970, 1971, 1972], which are monads that come equipped with additional structure that ensures the monad satisfies appropriate coherence conditions with respect to the monoidal structure of the underlying category. Furthermore, constructing the centre of a strong monad, as we present in this paper, aligns perfectly with the computational view that we informally described above and which we formally describe in the sequel. Indeed, in our opinion, there is no suitable notion of "centre" that can be formulated for non-strong monads that appropriately represents the intended computational intuition. Because of these reasons, we focus on strong monads throughout the rest of the paper and we note that every monad on the category **Set** is canonically strong.

1.3 The Centre of a Strong Monad: Mathematically

As we just explained, we are interested in *strong* monads, which are also the monads that are relevant for programming languages with support for higher-order functions. Given a strong monad $\mathcal{T} : \mathbf{C} \rightarrow \mathbf{C}$ on a symmetric monoidal category (\mathbf{C}, \otimes, I) , there are two obvious maps $\text{dst}, \text{dst}' : \mathcal{T}A \otimes \mathcal{T}B \rightarrow \mathcal{T}(A \otimes B)$ with the indicated domain and codomain that may be defined. However, these maps are not equal, in general. When the two maps do coincide, then we say the monad \mathcal{T} is *commutative* [Kock 1970, 1971, 1972]. The (lack of) commutativity of a monad is important: in general, the Kleisli category $\mathbf{C}_{\mathcal{T}}$ of a strong monad \mathcal{T} has a canonical *premonoidal* structure [Power and Robinson 1997], which is slightly weaker than a monoidal one, as the name suggests; moreover, this premonoidal structure is a monoidal one iff \mathcal{T} is commutative [Power and Robinson 1997]. What is also interesting, is that Power and Robinson introduced the notion of *centre of a premonoidal category* which has an important role in the theory of premonoidal categories. Given a premonoidal category \mathbf{C} , its premonoidal centre $Z(\mathbf{C})$ is always a *monoidal* subcategory of \mathbf{C} , but this subcategory does *not* induce a monad, in general. In the sequel, we show that the centre of every strong monad \mathcal{T} , whenever it exists, is strongly related to $Z(\mathbf{C}_{\mathcal{T}})$, i.e., the premonoidal centre of the Kleisli category of \mathcal{T} .

Many monads of interest in mathematics and computer science are strong, but not commutative. Thus, by identifying the centres of such monads, we can recover some of the benefits of commutativity. Every monad $\mathcal{T} : \mathbf{Set} \rightarrow \mathbf{Set}$ on the category **Set** is canonically strong and we can define its centre in a straightforward way: given a set X , the *centre* of \mathcal{T} at X , written $\mathcal{Z}X$, is the set

$$\mathcal{Z}X \stackrel{\text{def}}{=} \{t \in \mathcal{T}X \mid \forall Y \in \text{Ob}(\mathbf{Set}). \forall s \in \mathcal{T}Y. \text{dst}_{X,Y}(t, s) = \text{dst}'_{X,Y}(t, s)\}. \quad (4)$$

In other words, the centre of \mathcal{T} at X contains only those monadic elements for which the commutativity requirement holds with respect to any other possible choice for the second monadic argument of dst and dst' . With some additional mathematical effort, we can then prove the assignment $\mathcal{Z}(-)$ extends to a functor on **Set** and when equipped with the (co)restricted monad data of \mathcal{T} , it follows that \mathcal{Z} becomes a *commutative* monad on **Set** (Theorem 3.6). Furthermore, writing $\iota_X : \mathcal{Z}X \subseteq \mathcal{T}X$ for the indicated subset inclusion, it follows that \mathcal{Z} is a submonad of \mathcal{T} with ι giving the submonad monomorphism. We call the commutative submonad \mathcal{Z} *the central submonad* of \mathcal{T} which is justified by the fact that there exists a canonical isomorphism $\mathbf{Set}_{\mathcal{Z}} \cong Z(\mathbf{Set}_{\mathcal{T}})$ between the Kleisli category of \mathcal{Z} and the *premonoidal centre* [Power and Robinson 1997] of the Kleisli category of \mathcal{T} . That \mathcal{Z} is *the* central submonad of \mathcal{T} is even further justified by considering concrete algebraic examples:

- Given a monoid M , the free monad induced by M is the monad $(- \times M) : \mathbf{Set} \rightarrow \mathbf{Set}$. This monad is also known as the writer monad. Its central submonad is given by the monad $(- \times Z(M)) : \mathbf{Set} \rightarrow \mathbf{Set}$, where $Z(M)$ is the centre (in the usual sense) of the monoid M . See Example 5.10 for more details.

- Given a semiring S , a monad $\mathcal{T} : \mathbf{Set} \rightarrow \mathbf{Set}$ can be similarly defined through formal sums with scalars in S . Its central submonad is obtained with the same formal sums, but where the scalars are in $Z(S)$, the centre of S in the usual sense. Details can be found in Example 5.12.

Generalising away from the category \mathbf{Set} , our results stated above still hold, but establishing this requires more effort. In order to do so, we introduce the notion of *central cone* (Definition 4.2) which generalises the simple definition presented in (4). Our main mathematical result is to characterise the centre of a strong monad in three equivalent ways, as shown by Theorem 4.10.

THEOREM 4.10 (CENTRALISABILITY). *Let \mathbf{C} be a symmetric monoidal category and \mathcal{T} a strong monad on it. The following are equivalent:*

- (1) *For any object X of \mathbf{C} , \mathcal{T} admits a terminal central cone at X ;*
- (2) *There exists a commutative submonad \mathcal{Z} of \mathcal{T} such that the canonical embedding functor $I : \mathbf{C}_{\mathcal{Z}} \rightarrow \mathbf{C}_{\mathcal{T}}$ corestricts to an isomorphism of categories $\mathbf{C}_{\mathcal{Z}} \cong Z(\mathbf{C}_{\mathcal{T}})$;*
- (3) *The corestriction of the Kleisli left adjoint $\mathcal{J} : \mathbf{C} \rightarrow \mathbf{C}_{\mathcal{T}}$ to the premonoidal centre $\hat{\mathcal{J}} : \mathbf{C} \rightarrow Z(\mathbf{C}_{\mathcal{T}})$ also is a left adjoint.*

This theorem gives us three equivalent conditions for the existence of the centre of a *specific* monad. In general, not every strong monad admits a centre and we show this by specifically constructing a category and a strong monad on it for this purpose. However, we are not aware of any other *naturally* occurring monad described in the literature that does not admit a centre. Moreover, by using Theorem 4.10 it is easy to prove that every strong monad on many categories of interest (e.g., \mathbf{Set} , \mathbf{DCPO} , \mathbf{Meas} , \mathbf{Top} , \mathbf{Vect}) admits a centre (see §5.2). Because of this, we believe the notion of centre for a strong monad is ubiquitous. We also provide further concrete examples of monads of interest that admit interesting centres, e.g., the valuations monad on \mathbf{DCPO} (see §5.3).

1.4 Monads as Monoids

It is well-known that monads on a category \mathbf{C} are exactly the *monoid objects*³ on the category of endofunctors of \mathbf{C} [Mac Lane 1998, pp. 138]. Because of this, we might think of monoid objects as being more general than monads. So, a natural question to ask is why not determine how to form the centre of an arbitrary monoid object, and then, as a special case, recover the centre of a monad? We thought about this possibility, but we decided against it for two reasons:

- (1) As we already explained, it is the *strong* monads that are computationally relevant. However, the "monads as monoids" correspondence does not allow us to recover the monadic strength. We personally see no way to achieve this and we do not know how to relate strong monads to special kinds of monoids.
- (2) *Commutative* monoid objects are usually defined via a *symmetric* monoidal structure. But in the "monads as monoids" correspondence, the monoidal structure of the category of endofunctors on \mathbf{C} is *not* symmetric in general. Even worse, restricting to endofunctors of the form \mathcal{T}^n , a symmetry would then be a distributive law of \mathcal{T} over itself which is known to not always exist, an example being the power-set monad [Klin and Salamanca 2018], which nevertheless is commutative in the usual sense.

Furthermore, we think that monads play a more important computational role, compared to monoids, so we prefer to directly work with monads. A different notion of centre may be defined in this direction but it seems not directly related to ours and not having any computational interpretation.

³A monoid object in \mathbf{Set} is a monoid in the usual sense. Thus, monoid objects are a generalisation of the usual notion.

1.5 Overview and Summary of Results

The paper is organised as follows. In §2 we recall background material on strong and commutative monads and the premonoidal structure of their Kleisli categories. In §3 we show how to construct the centre of any monad on **Set**. This construction provides the main intuition for the more general and abstract results that follow. In §4 we state the main mathematical results of the paper: we provide three equivalent characterisations of the centre of a strong monad acting on a symmetric monoidal category. In §5 we illustrate our ideas with many examples. We show how to use our main results to conclude that for many categories of interest, all strong monads are centralisable. We also provide concrete examples of strong monads, construct their centres and discuss their computational significance. In §6 we provide a computational interpretation of our ideas by presenting a refinement of Moggi’s monadic metalanguage. We also provide a categorical model together with a denotational semantics for it. In §7 we illustrate how our computational interpretation may be used through a case study involving a non-commutative writer monad for which we describe a sound and adequate semantics and provide examples of easily established contextual equivalences through the richer type system. In §8 we discuss related work. Finally, in §9 we provide concluding remarks and discuss future work.

2 BACKGROUND

We start by introducing some background on strong and commutative monads and their premonoidal structure. We also use this as an opportunity to fix notation. In this paper we assume some knowledge of category theory and we implicitly assume throughout the paper that all categories we are working with are locally small.

2.1 Strong and Commutative Monads

We begin by recalling the definition of a monad.

Definition 2.1 (Monad). A monad over a category \mathbf{C} is an endofunctor $\mathcal{T} : \mathbf{C} \rightarrow \mathbf{C}$ equipped with two natural transformations $\eta : \text{Id} \Rightarrow \mathcal{T}$ and $\mu : \mathcal{T}^2 \Rightarrow \mathcal{T}$ such that the following diagrams

$$\begin{array}{ccc}
 \mathcal{T}^3 X & \xrightarrow{\mu_{\mathcal{T}X}} & \mathcal{T}^2 X \\
 \mathcal{T}\mu_X \downarrow & & \downarrow \mu_X \\
 \mathcal{T}^2 X & \xrightarrow{\mu_X} & \mathcal{T}X
 \end{array}
 \qquad
 \begin{array}{ccc}
 \mathcal{T}X & \xrightarrow{\mathcal{T}\eta_X} & \mathcal{T}^2 X \\
 \eta_{\mathcal{T}X} \downarrow & \searrow & \downarrow \mu_X \\
 \mathcal{T}^2 X & \xrightarrow{\mu_X} & \mathcal{T}X
 \end{array}$$

commute. We call η the *unit* of \mathcal{T} and we say that μ is the *multiplication* of \mathcal{T} .

Next, we recall the definition of a *strong* monad, which is the main object of study in this paper. As we already explained in the introduction, these monads are more computationally relevant (compared to non-strong ones) for most use cases. The additional structure, called the *monadic strength*, ensures the monad interacts appropriately with the monoidal structure of the base category.

Definition 2.2 (Strong Monad). A *strong monad* over a monoidal category $(\mathbf{C}, \otimes, I, \alpha, \lambda, \rho)$ is a monad (\mathcal{T}, η, μ) equipped with a natural transformation $\tau_{X,Y} : X \otimes \mathcal{T}Y \rightarrow \mathcal{T}(X \otimes Y)$, called *strength*,

such that the following diagrams commute:

$$\begin{array}{ccc}
 I \otimes \mathcal{T}X & \xrightarrow{\tau_{I,X}} & \mathcal{T}(I \otimes X) \\
 \searrow \lambda_{\mathcal{T}X} & & \downarrow \mathcal{T}\lambda_X \\
 & & \mathcal{T}X
 \end{array}
 \quad
 \begin{array}{ccc}
 (W \otimes X) \otimes \mathcal{T}Y & \xrightarrow{\tau_{W \otimes X, Y}} & \mathcal{T}((W \otimes X) \otimes Y) \\
 \downarrow \alpha_{W, X, \mathcal{T}Y} & & \downarrow \mathcal{T}\alpha_{W, X, Y} \\
 W \otimes (X \otimes \mathcal{T}Y) & \xrightarrow{W \otimes \tau_{X, Y}} & W \otimes \mathcal{T}(X \otimes Y) \xrightarrow{\tau_{W, X \otimes Y}} \mathcal{T}(W \otimes (X \otimes Y))
 \end{array}$$

$$\begin{array}{ccc}
 X \otimes Y & \xrightarrow{X \otimes \eta_Y} & X \otimes \mathcal{T}Y \\
 \searrow \eta_{X \otimes Y} & & \downarrow \tau_{X, Y} \\
 & & \mathcal{T}(X \otimes Y)
 \end{array}
 \quad
 \begin{array}{ccc}
 X \otimes \mathcal{T}^2 Y & \xrightarrow{\tau_{X, \mathcal{T}Y}} & \mathcal{T}(X \otimes \mathcal{T}Y) \xrightarrow{\mathcal{T}\tau_{X, Y}} \mathcal{T}^2(X \otimes Y) \\
 \downarrow X \otimes \mu_Y & & \downarrow \mu_{X \otimes Y} \\
 X \otimes \mathcal{T}Y & \xrightarrow{\tau_{X, Y}} & \mathcal{T}(X \otimes Y)
 \end{array}$$

We now recall the definition of a *commutative monad* which is of central importance in this paper. Compared to a strong monad, a commutative monad enjoys even stronger coherence properties with respect to the monoidal structure of the base category (see also §2.2).

Definition 2.3 (Commutative Monad). Let $(\mathcal{T}, \eta, \mu, \tau)$ be a strong monad on a *symmetric monoidal category* $(\mathbf{C}, \otimes, I, \gamma)$. The costrength $\tau'_{X, Y}: \mathcal{T}X \otimes Y \rightarrow \mathcal{T}(X \otimes Y)$ of \mathcal{T} is given by the assignment $\tau'_{X, Y} \stackrel{\text{def}}{=} \mathcal{T}(\gamma_{Y, X}) \circ \tau_{Y, X} \circ \gamma_{\mathcal{T}X, Y}$. Then, \mathcal{T} is said to be *commutative* if the following diagram commutes:

$$\begin{array}{ccccc}
 \mathcal{T}X \otimes \mathcal{T}Y & \xrightarrow{\tau_{\mathcal{T}X, Y}} & \mathcal{T}(\mathcal{T}X \otimes Y) & \xrightarrow{\mathcal{T}\tau'_{X, Y}} & \mathcal{T}^2(X \otimes Y) \\
 \tau'_{X, \mathcal{T}Y} \downarrow & & & & \downarrow \mu_{X \otimes Y} \\
 \mathcal{T}(X \otimes \mathcal{T}Y) & \xrightarrow{\mathcal{T}\tau_{X, Y}} & \mathcal{T}^2(X \otimes Y) & \xrightarrow{\mu_{X \otimes Y}} & \mathcal{T}(X \otimes Y)
 \end{array}$$

The appropriate notion of morphism between two strong monads is given by our next definition.

Definition 2.4 (Morphism of Strong Monads [Jacobs 2016]). Given two strong monads $(\mathcal{T}, \eta^{\mathcal{T}}, \mu^{\mathcal{T}}, \tau^{\mathcal{T}})$ and $(\mathcal{P}, \eta^{\mathcal{P}}, \mu^{\mathcal{P}}, \tau^{\mathcal{P}})$ over a category \mathbf{C} , then a *morphism of strong monads* is a natural transformation $\iota: \mathcal{T} \Rightarrow \mathcal{P}$ that makes the following diagrams commute:

$$\begin{array}{ccc}
 \begin{array}{ccc} X & & \\ \eta_X^{\mathcal{T}} \swarrow & & \searrow \eta_X^{\mathcal{P}} \\ \mathcal{T}X & \xrightarrow{\iota_X} & \mathcal{P}X \end{array} &
 \begin{array}{ccc} \mathcal{T}^2 X & \xrightarrow{\iota_{\mathcal{T}X}} \mathcal{P}\mathcal{T}X & \xrightarrow{\mathcal{P}\iota_X} \mathcal{P}^2 X \\ \mu_X^{\mathcal{T}} \downarrow & & \downarrow \mu_X^{\mathcal{P}} \\ \mathcal{T}X & \xrightarrow{\iota_X} & \mathcal{P}X \end{array} &
 \begin{array}{ccc} A \otimes B & \xrightarrow{A \otimes \iota_B} & A \otimes \mathcal{P}B \\ \tau_{A, B}^{\mathcal{T}} \downarrow & & \downarrow \tau_{A, B}^{\mathcal{P}} \\ \mathcal{T}(A \otimes B) & \xrightarrow{\iota_{A \otimes B}} & \mathcal{P}(A \otimes B) \end{array}
 \end{array}$$

It is easy to see that strong monads over a (symmetric) monoidal category \mathbf{C} and strong monad morphisms between them form a category which we denote by writing $\mathbf{StrMnd}(\mathbf{C})$. In the situation

of Definition 2.4, if ι is a monomorphism in $\mathbf{StrMnd}(\mathbf{C})$, then \mathcal{T} is said to be a *strong submonad* of \mathcal{P} and ι is said to be a *submonad morphism*. The submonad relation induces an embedding between the Kleisli categories of the corresponding monads that we recall next.

Definition 2.5 (Kleisli category). Given a monad (\mathcal{T}, η, μ) over a category \mathbf{C} , the *Kleisli category* $\mathbf{C}_{\mathcal{T}}$ of \mathcal{T} is the category whose objects are the same as those of \mathbf{C} , but whose morphisms are given by $\mathbf{C}_{\mathcal{T}}[X, Y] = \mathbf{C}[X, \mathcal{T}Y]$. Composition in $\mathbf{C}_{\mathcal{T}}$ is given by $g \circ f \stackrel{\text{def}}{=} \mu_Z \circ \mathcal{T}g \circ f$ where $f : X \rightarrow \mathcal{T}Y$ and $g : Y \rightarrow \mathcal{T}Z$. The identity at X is given by the monadic unit $\eta_X : X \rightarrow \mathcal{T}X$.

PROPOSITION 2.6. *If $\iota : \mathcal{T} \Rightarrow \mathcal{P}$ is a submonad morphism, then the functor $I : \mathbf{C}_{\mathcal{T}} \rightarrow \mathbf{C}_{\mathcal{P}}$, which is defined by $I(X) = X$ on objects and on morphisms by $I(f : X \rightarrow \mathcal{T}Y) = \iota_Y \circ f : X \rightarrow \mathcal{P}Y$, is an embedding of categories.*

The functor I described in the above proposition is the canonical embedding of $\mathbf{C}_{\mathcal{T}}$ into $\mathbf{C}_{\mathcal{P}}$ induced by the submonad morphism $\iota : \mathcal{T} \Rightarrow \mathcal{P}$.

2.2 Premonoidal Structure of Strong Monads

Let \mathcal{T} be a strong monad on a symmetric monoidal category (\mathbf{C}, I, \otimes) . Then, its Kleisli category $\mathbf{C}_{\mathcal{T}}$ does *not* necessarily have a canonical monoidal structure. However, it does have a canonical *premonoidal structure* as shown by Power and Robinson [Power and Robinson 1997]. In fact, they show that this premonoidal structure is monoidal iff the monad \mathcal{T} is commutative. Next, we briefly recall the premonoidal structure of \mathcal{T} as outlined by Power and Robinson.

For every two objects X and Y of $\mathbf{C}_{\mathcal{T}}$, their tensor product $X \otimes Y$ is also an object of $\mathbf{C}_{\mathcal{T}}$. But, the monoidal product \otimes of \mathbf{C} does not necessarily induce a monoidal functor on $\mathbf{C}_{\mathcal{T}}$. However, by using the strength and the costrength of \mathcal{T} , we can define two families of functors as follows:

- for any object X , a functor $(- \otimes_l X) : \mathbf{C}_{\mathcal{T}} \rightarrow \mathbf{C}_{\mathcal{T}}$ whose action on objects sends Y to $Y \otimes X$, and sends $f : Y \rightarrow \mathcal{T}Z$ to $\tau'_{Z, X} \circ (f \otimes X) : Y \otimes X \rightarrow \mathcal{T}(Z \otimes X)$;
- for any object X , a functor $(X \otimes_r -) : \mathbf{C}_{\mathcal{T}} \rightarrow \mathbf{C}_{\mathcal{T}}$ whose action on objects sends Y to $X \otimes Y$, and sends $f : Y \rightarrow \mathcal{T}Z$ to $\tau_{X, Z} \circ (X \otimes f) : X \otimes Y \rightarrow \mathcal{T}(X \otimes Z)$.

This categorical data satisfies the axioms and coherence properties of *premonoidal categories* as explained in [Power and Robinson 1997], but which we omit here because it is not important for the development of our results. What is important, is to note that in a premonoidal category, $f \otimes_l X'$ and $X \otimes_r g$ do not always commute. This leads us to the next definition, which plays a crucial role in the theory of premonoidal categories and which has important links to our development as well.

Definition 2.7 (Premonoidal Centre [Power and Robinson 1997]). Given a strong monad $(\mathcal{T}, \eta, \mu, \tau)$ on a symmetric monoidal category (\mathbf{C}, I, \otimes) , we say that a morphism $f : X \rightarrow Y$ in $\mathbf{C}_{\mathcal{T}}$ is *central* if for any morphism $f' : X' \rightarrow Y'$ in $\mathbf{C}_{\mathcal{T}}$, the diagram

$$\begin{array}{ccc} X \otimes X' & \xrightarrow{f \otimes_l X'} & Y \otimes X' \\ \downarrow X \otimes_r f' & & \downarrow Y \otimes_r f' \\ X \otimes Y' & \xrightarrow{f \otimes_l Y'} & Y \otimes Y' \end{array}$$

commutes in $\mathbf{C}_{\mathcal{T}}$; or equivalently, the diagram

$$\begin{array}{ccccccc}
 X \otimes X' & \xrightarrow{f \otimes X'} & \mathcal{T}Y \otimes X' & \xrightarrow{\tau'_{Y,X'}} & \mathcal{T}(Y \otimes X') & \xrightarrow{\mathcal{T}(Y \otimes f')} & \mathcal{T}(Y \otimes \mathcal{T}Y') \\
 X \otimes f' \downarrow & & & & & & \downarrow \mathcal{T}\tau_{Y,Y'} \\
 X \otimes \mathcal{T}Y' & & & & & & \mathcal{T}^2(Y \otimes Y') \\
 \tau_{X,Y'} \downarrow & & & & & & \downarrow \mu_{Y \otimes Y'} \\
 \mathcal{T}(X \otimes Y') & \xrightarrow{\mathcal{T}(f \otimes Y')} & \mathcal{T}(\mathcal{T}Y \otimes Y') & \xrightarrow{\mathcal{T}\tau'_{Y,Y'}} & \mathcal{T}^2(Y \otimes Y') & \xrightarrow{\mu_{Y \otimes Y'}} & \mathcal{T}(Y \otimes Y')
 \end{array}$$

commutes in \mathbf{C} . The *premonoidal centre* of $\mathbf{C}_{\mathcal{T}}$ is the subcategory $Z(\mathbf{C}_{\mathcal{T}})$ which has the same objects as those of $\mathbf{C}_{\mathcal{T}}$ and whose morphisms are the central morphisms of $\mathbf{C}_{\mathcal{T}}$.

In [Power and Robinson 1997], the authors prove that $Z(\mathbf{C}_{\mathcal{T}})$, is a symmetric *monoidal* subcategory of $\mathbf{C}_{\mathcal{T}}$. In particular, this means that Kleisli composition and the tensor functors $(- \otimes_l X)$ and $(X \otimes_r -)$ preserve central morphisms. However, it does not necessarily hold that the subcategory $Z(\mathbf{C}_{\mathcal{T}})$ determines a monad over \mathbf{C} . Nevertheless, in this situation, the left adjoint of the Kleisli adjunction $\mathcal{J} : \mathbf{C} \rightarrow \mathbf{C}_{\mathcal{T}}$ always corestricts to $Z(\mathbf{C}_{\mathcal{T}})$ and we write $\hat{\mathcal{J}} : \mathbf{C} \rightarrow Z(\mathbf{C}_{\mathcal{T}})$ to indicate this corestriction (which need not be a left adjoint).

REMARK 2.8. In [Power and Robinson 1997], the subcategory $Z(\mathbf{C}_{\mathcal{T}})$ is called the *centre* of $\mathbf{C}_{\mathcal{T}}$. However, we refer to it as the *premonoidal centre* of a premonoidal category in order to avoid confusion with the new notion of *centre* of a monad that we introduce next. In the sequel, we show that the two notions are very strongly related to each other (Theorem 4.10).

3 CENTRAL SUBMONADS ON THE CATEGORY SET

In this section we show how we can construct the centre of any strong monad acting on the category \mathbf{Set} . The results here are a special case of our more general results from §4, but we choose to devote special attention to \mathbf{Set} for illustrative purposes and because the construction of the centre is the easiest to understand for this category (in our view). We note that every monad on \mathbf{Set} is *canonically strong* [Jakl et al. 2022, Remark 4.1] and therefore we show that every monad on \mathbf{Set} admits a centre.

NOTATION 3.1. Throughout the remainder of the section, we write $(\mathcal{T}, \eta, \mu, \tau)$ to indicate an arbitrary strong monad on the category \mathbf{Set} and we write τ' to indicate the costrength of \mathcal{T} .

Definition 3.2 (Centre). Given a set X , the *centre* of \mathcal{T} at X , written $\mathcal{Z}X$, is defined to be the set

$$\mathcal{Z}X \stackrel{\text{def}}{=} \{t \in \mathcal{T}X \mid \forall Y \in \text{Ob}(\mathbf{Set}). \forall s \in \mathcal{T}Y. \mu(\mathcal{T}\tau'(\tau(t, s))) = \mu(\mathcal{T}\tau(\tau'(t, s)))\}.$$

We write $\iota_X : \mathcal{Z}X \subseteq \mathcal{T}X$ for the indicated subset inclusion.

In other words, the centre of \mathcal{T} at X is the subset of $\mathcal{T}X$ which contains all monadic elements for which (2.3) holds when the set X is fixed.

REMARK 3.3. Notice that $\mathcal{Z}X \supseteq \eta_X(X)$, i.e., the centre of \mathcal{T} at X always contains all monadic elements which are in the image of the monadic unit. This follows easily from the axioms of strong monads.

In fact, the assignment $\mathcal{Z}(-)$ extends to a *commutative submonad* of \mathcal{T} . This is made precise by the following lemmas and theorems.

LEMMA 3.4. *The assignment $\mathcal{Z}(-)$ extends to a functor $\mathcal{Z} : \mathbf{Set} \rightarrow \mathbf{Set}$ when we define*

$$\mathcal{Z}f \stackrel{\text{def}}{=} \mathcal{T}f|_{\mathcal{Z}X} : \mathcal{Z}X \rightarrow \mathcal{Z}Y,$$

for any function $f : X \rightarrow Y$, where $\mathcal{T}f|_{\mathcal{Z}X}$ indicates the restriction of $\mathcal{T}f : \mathcal{T}X \rightarrow \mathcal{T}Y$ to the subset $\mathcal{Z}X$.

PROOF. The validity of this definition is equivalent to showing that $\mathcal{T}f(\mathcal{Z}X) \subseteq \mathcal{Z}Y$. This follows as a special case of Theorem 4.8. \square

LEMMA 3.5. *For any two sets X and Y , the monadic unit $\eta_X : X \rightarrow \mathcal{T}X$, the monadic multiplication $\mu_X : \mathcal{T}^2X \rightarrow \mathcal{T}X$, and the monadic strength $\tau_{X,Y} : X \times \mathcal{T}Y \rightarrow \mathcal{T}(X \times Y)$ (co)restrict respectively to functions $\eta_X^{\mathcal{Z}} : X \rightarrow \mathcal{Z}X$, $\mu_X^{\mathcal{Z}} : \mathcal{Z}^2X \rightarrow \mathcal{Z}X$ and $\tau_{X,Y}^{\mathcal{Z}} : X \times \mathcal{Z}Y \rightarrow \mathcal{Z}(X \times Y)$.*

PROOF. Special case of Theorem 4.8. \square

With a little bit more effort, it is possible to prove that the data we described above constitutes a commutative submonad of \mathcal{T} .

THEOREM 3.6. *The assignment $\mathcal{Z}(-)$ extends to a commutative submonad $(\mathcal{Z}, \eta^{\mathcal{Z}}, \mu^{\mathcal{Z}}, \tau^{\mathcal{Z}})$ of \mathcal{T} with $\iota_X : \mathcal{Z}X \subseteq \mathcal{T}X$ the required submonad morphism. Furthermore, there exists a canonical isomorphism $\mathbf{Set}_{\mathcal{Z}} \cong \mathcal{Z}(\mathbf{Set}_{\tau})$ ⁴.*

PROOF. Special case of Theorem 4.10. \square

The final statement of the above theorem is very important. It shows that the Kleisli category of \mathcal{Z} is canonically isomorphic to the premonoidal centre of the Kleisli category of \mathcal{T} . Because of this, we are justified in saying that \mathcal{Z} is not just a commutative submonad of \mathcal{T} , but rather it is *the central submonad* of \mathcal{T} , which is necessarily commutative (just like the centre of a monoid is a commutative submonoid). In §5.3 we provide concrete examples of monads on \mathbf{Set} and their central submonads and we see that the construction of the centre aligns nicely with our intuition.

4 CENTRALISABLE MONADS

In this section we show how to define the central submonad of a strong monad on a symmetric monoidal category. This submonad does not always exist (but it *usually* does) and we present three equivalent conditions that characterise its existence. In Subsection 4.1 we present the first such characterisation in terms of *central cones*. Then, in Subsection 4.2 we present the remaining ones that allow us to establish a link to the theory of premonoidal categories of Power and Robinson.

4.1 Central cones

In this subsection we show how the construction of the central submonad can be generalised to many categories other than \mathbf{Set} .

NOTATION 4.1. *Throughout the remainder of the section, we assume we are given a symmetric monoidal category (\mathbf{C}, \otimes, I) . We also assume that $(\mathcal{T}, \eta, \mu, \tau)$ is an arbitrary strong monad on the category \mathbf{C} . We write τ' to indicate the costrength of \mathcal{T} , which is induced by the strength τ and the symmetry of \mathbf{C} in the usual way. All theorems and definitions in this section are stated with respect to this monad structure.*

⁴We explain later (see Theorem 4.10) in what sense this isomorphism is canonical.

In **Set**, the centre is defined pointwise through subsets of $\mathcal{T}X$ which only contain elements that satisfy the coherence condition for a commutative monad. However, \mathbf{C} is an arbitrary symmetric monoidal category, so we cannot easily form subobjects in the required way. This leads us to the definition of a *central cone* which allows us to overcome this problem.

Definition 4.2 (Central Cone). Let X be an object of \mathbf{C} . A *central cone* of \mathcal{T} at X is given by a pair (Z, ι) of an object Z and a morphism $\iota : Z \rightarrow \mathcal{T}X$, such that for any object Y , the diagram

$$\begin{array}{ccccc}
 Z \otimes \mathcal{T}Y & \xrightarrow{\iota \otimes \mathcal{T}Y} & \mathcal{T}X \otimes \mathcal{T}Y & \xrightarrow{\tau'_{X, \mathcal{T}Y}} & \mathcal{T}(X \otimes \mathcal{T}Y) \\
 \iota \otimes \mathcal{T}Y \downarrow & & & & \downarrow \mathcal{T}\tau_{X, Y} \\
 \mathcal{T}X \otimes \mathcal{T}Y & & & & \mathcal{T}^2(X \otimes Y) \\
 \tau_{\mathcal{T}X, Y} \downarrow & & & & \downarrow \mu_{X \otimes Y} \\
 \mathcal{T}(\mathcal{T}X \otimes Y) & \xrightarrow{\mathcal{T}\tau'_{X, Y}} & \mathcal{T}^2(X \otimes Y) & \xrightarrow{\mu_{X \otimes Y}} & \mathcal{T}(X \otimes Y)
 \end{array}$$

commutes. If (Z, ι) and (Z', ι') are two central cones of \mathcal{T} at X , then a *morphism of central cones* $\varphi : (Z', \iota') \rightarrow (Z, \iota)$ is a morphism $\varphi : Z' \rightarrow Z$, such that $\iota \circ \varphi = \iota'$. A *terminal central cone* of \mathcal{T} at X is a central cone (Z, ι) for \mathcal{T} at X , such that for any central cone (Z', ι') of \mathcal{T} at X , there exists a unique morphism of central cones $\varphi : (Z', \iota') \rightarrow (Z, \iota)$.

The names "central morphism" (in the premonoidal sense, see §2.2) and "central cone" (above) also hint that there should be a relation between them. This is indeed the case and we show that the two definitions are equivalent.

PROPOSITION 4.3. *Let $f : X \rightarrow \mathcal{T}Y$ be a morphism in \mathbf{C} . The pair (X, f) is a central cone of \mathcal{T} at Y if and only if f is central in $\mathcal{C}_{\mathcal{T}}$ in the premonoidal sense (see Def. 2.7).*

PROOF. The naturality of τ and τ' allow us to rewrite each definition into the other. See Appendix A for more details. \square

From now on, we rely heavily on the fact that central cones and central morphisms are equivalent notions and we use Proposition 4.3 implicitly in the sequel. On the other hand, *terminal* central cones are crucial for our development, but it is unclear how to introduce a similar notion of "terminal central morphism" that is useful. For this reason, we have a preference to work with (terminal) central cones in this paper.

Central cones (or central morphisms) have several nice properties. To begin, we show that central cones are closed under certain compositions.

LEMMA 4.4. *If $(X, f : X \rightarrow \mathcal{T}Y)$ is a central cone of \mathcal{T} at Y , then for any $g : Z \rightarrow X$, it follows that $(Z, f \circ g)$ is a central cone of \mathcal{T} at Y .*

PROOF. This is obtained by precomposing the definition of central cone by $g \otimes \text{id}$. See Appendix A for more details. \square

LEMMA 4.5. *If $(X, f : X \rightarrow \mathcal{T}Y)$ is a central cone of \mathcal{T} at Y then for any $g : Y \rightarrow Z$, it follows that $(X, \mathcal{T}g \circ f)$ is a central cone of \mathcal{T} at Z .*

PROOF. The naturality of τ and μ allow us to push the application of g to the last postcomposition, in order to use the central property of f . See Appendix A for more details. \square

Next, a very important property for *terminal* central cones.

LEMMA 4.6. *If (Z, ι) is a terminal central cone of \mathcal{T} at X , then ι is a monomorphism.*

PROOF. Let us consider $f, g : Y \rightarrow Z$ such that $\iota \circ f = \iota \circ g$; this morphism is a central cone at X (Lemma 4.4), and since (Z, ι) is a terminal central cone, it factors uniquely through ι . Thus $f = g$ and therefore ι is monic. \square

It is easy to see that if a terminal central cone for \mathcal{T} at X exists, then it is unique up to a unique isomorphism of central cones. We also note that Lemma 4.6 is crucial for defining the centre of \mathcal{T} through terminal central cones, because the morphisms ι would be the components of a submonad morphism. The main definition of this subsection follows next and gives the foundation for the construction of the central submonad.

Definition 4.7 (Centralisable Monad). We say that the monad \mathcal{T} is *centralisable* if for any object X , a terminal central cone of \mathcal{T} at X exists. In this situation, we write $(\mathcal{Z}X, \iota_X)$ for the terminal central cone of \mathcal{T} at X .

In fact, for a centralisable monad \mathcal{T} , its terminal central cones induce a commutative submonad \mathcal{Z} of \mathcal{T} . This is the main theorem of this subsection, which is stated next, and its proof reveals constructively how the monad structure arises from the terminal central cones.

THEOREM 4.8. *If the monad \mathcal{T} is centralisable, then the assignment $\mathcal{Z}(-)$ extends to a commutative monad $(\mathcal{Z}, \eta^{\mathcal{Z}}, \mu^{\mathcal{Z}}, \tau^{\mathcal{Z}})$ on \mathbf{C} . Moreover, \mathcal{Z} is a commutative submonad of \mathcal{T} in the sense that the morphisms $\iota_X : \mathcal{Z}X \rightarrow \mathcal{T}X$ constitute a monomorphism of strong monads $\iota : \mathcal{Z} \Rightarrow \mathcal{T}$.*

PROOF. First, we extend the assignment $\mathcal{Z}(-)$ to an endofunctor on \mathbf{C} . Let $f : X \rightarrow Y$ be a morphism in \mathbf{C} . Recall that \mathcal{Z} maps every object X to its terminal central cone at X . We know that $\mathcal{T}f \circ \iota_X : \mathcal{Z}X \rightarrow \mathcal{T}Y$ is a central cone (Lemma 4.5), therefore, we define $\mathcal{Z}f$ as the unique map such that the following diagram commutes:

$$\begin{array}{ccc} & \mathcal{Z}f & \\ & \dashrightarrow & \\ \mathcal{Z}X & & \mathcal{Z}Y \\ \iota_X \downarrow & & \downarrow \iota_Y \\ \mathcal{T}X & \xrightarrow{\quad} & \mathcal{T}Y \\ & \mathcal{T}f & \end{array}$$

It follows directly that $\mathcal{Z}(\text{id}_X) = \text{id}_{\mathcal{Z}X}$ and that $\iota : \mathcal{Z} \Rightarrow \mathcal{T}$ is a natural transformation. Moreover, \mathcal{Z} preserves composition, which follows after recognising that the diagram

$$\begin{array}{ccc} & \xrightarrow{\iota_A} & \mathcal{T}A \\ \mathcal{Z}A & \xrightarrow{\quad} & \mathcal{T}A \\ \mathcal{Z}g \downarrow & & \downarrow \mathcal{T}g \\ & \xrightarrow{\iota_B} & \mathcal{T}B \\ \mathcal{Z}B & \xrightarrow{\quad} & \mathcal{T}B \\ \mathcal{Z}f \downarrow & & \downarrow \mathcal{T}f \\ & \xrightarrow{\iota_C} & \mathcal{T}C \\ \mathcal{Z}C & \xrightarrow{\quad} & \mathcal{T}C \end{array} \quad \mathcal{T}(f \circ g)$$

(A dashed arrow labeled $\mathcal{Z}(f \circ g)$ points from $\mathcal{Z}A$ to $\mathcal{Z}C$.)

commutes. This shows that \mathcal{Z} is indeed a functor. Next, we describe its monad structure. The monadic unit η_X is central, because it is the identity morphism in $Z(\mathcal{C}_{\mathcal{T}})$, thus it factors uniquely through ι_X to define $\eta_X^{\mathcal{Z}}$ as in the following diagram:

$$\begin{array}{ccc} X & \xrightarrow{\eta_X^{\mathcal{Z}}} & \mathcal{Z}X \\ \eta_X \searrow & & \swarrow \iota_X \\ & \mathcal{T}X & \end{array}$$

Next, observe that, by definition, $\mu_X \circ \mathcal{T}\iota_X \circ \iota_{\mathcal{Z}X} = \iota_X \circ \iota_{\mathcal{Z}X}$, where $(- \circ -)$ indicates Kleisli composition. Since ι is central and Kleisli composition preserves central morphisms, it follows this morphism factors uniquely through ι_X and we use this to define $\mu_X^{\mathcal{Z}}$ as in the following diagram:

$$\begin{array}{ccc} \mathcal{Z}^2X & \xrightarrow{\mu_X^{\mathcal{Z}}} & \mathcal{Z}X \\ \iota_{\mathcal{Z}X} \downarrow & & \downarrow \iota_X \\ \mathcal{T}\mathcal{Z}X & \xrightarrow[\mathcal{T}\iota_X]{} \mathcal{T}^2X \xrightarrow[\mu_X]{} & \mathcal{T}X \end{array}$$

Again, by definition, $\tau_{A,B} \circ (A \otimes \iota_B) = A \otimes_f \iota_B$. Central morphisms are preserved by the premonoidal products (as we noted in Section 2) and therefore, this morphism factors uniquely through $\iota_{A \otimes B}$ which we use to define $\tau_{A,B}^{\mathcal{Z}}$ as in the following diagram:

$$\begin{array}{ccc} A \otimes \mathcal{Z}B & \xrightarrow{\tau_{A,B}^{\mathcal{Z}}} & \mathcal{Z}(A \otimes B) \\ A \otimes \iota_B \downarrow & & \downarrow \iota_{A \otimes B} \\ A \otimes \mathcal{T}B & \xrightarrow[\tau_{A,B}]{} & \mathcal{T}(A \otimes B) \end{array}$$

Note that the last three diagrams are exactly those of a morphism of strong monads (see Definition 2.4). This defines the monad structure of \mathcal{Z} . See Appendix A for the remainder of the proof. \square

This theorem shows that centralisable monads always induce a canonical commutative submonad. However, we still have not precisely explained in what sense this submonad is "central". We justify this next. Note that, since \mathcal{Z} is a submonad of \mathcal{T} , we know that $\mathcal{C}_{\mathcal{Z}}$ canonically embeds into $\mathcal{C}_{\mathcal{T}}$ (see Proposition 2.6). The next theorem shows that this embedding factors through the premonoidal centre of $\mathcal{C}_{\mathcal{T}}$, and moreover, the two categories are isomorphic.

THEOREM 4.9. *In the situation of Theorem 4.8, the canonical embedding functor $I : \mathcal{C}_{\mathcal{Z}} \rightarrow \mathcal{C}_{\mathcal{T}}$ corestricts to an isomorphism of categories $\mathcal{C}_{\mathcal{Z}} \cong Z(\mathcal{C}_{\mathcal{T}})$.*

PROOF. That I corestricts as indicated follows easily: for any morphism $f : X \rightarrow \mathcal{Z}Y$, we have that $I f = \iota_Y \circ f$ which is central by Lemma 4.4. Let us write \hat{I} for the corestriction of I to $Z(\mathcal{C}_{\mathcal{T}})$. Next, to prove that $\hat{I} : \mathcal{C}_{\mathcal{Z}} \rightarrow Z(\mathcal{C}_{\mathcal{T}})$ is an isomorphism, we define the inverse functor $G : Z(\mathcal{C}_{\mathcal{T}}) \rightarrow \mathcal{C}_{\mathcal{Z}}$.

On objects, $G(X) \stackrel{\text{def}}{=} X$. To define its mapping on morphisms, observe that if $f : X \rightarrow \mathcal{T}Y$ is a central morphism (in the premonoidal sense), then (X, f) is a central cone of \mathcal{T} at Y (Proposition

4.3) and therefore there exists a unique morphism $f^{\mathcal{Z}} : X \rightarrow \mathcal{Z}Y$ such that $\iota_Y \circ f^{\mathcal{Z}} = f$; we define $Gf \stackrel{\text{def}}{=} f^{\mathcal{Z}}$. The proof that G is a functor is direct considering that any $f^{\mathcal{Z}}$ is a morphism of central cones and that all components of ι are monomorphisms.

To show that \hat{I} and G are mutual inverses, let $f : X \rightarrow \mathcal{T}Y$ be a morphism of $Z(\mathcal{C}_{\mathcal{T}})$, i.e., a central morphism. Then, $\hat{I}Gf = \iota_Y \circ f^{\mathcal{Z}} = f$ by definition of morphism of central cones (see Definition 4.2). For the other direction, let $g : X \rightarrow \mathcal{Z}Y$ be a morphism in \mathcal{C} . Then, $\iota_Y \circ G\hat{I}g = \iota_Y \circ (\iota_Y \circ g)^{\mathcal{Z}} = \iota_Y \circ g$ by Definition 4.2 and thus $G\hat{I}g = g$ since ι_Y is a monomorphism (Lemma 4.6). \square

It should now be clear that Theorem 4.8 and Theorem 4.9 show that we are justified in naming the submonad \mathcal{Z} as *the* central submonad of \mathcal{T} .

4.2 Characterising the Centre of Strong Monads

In the previous subsection we showed that the existence of terminal central cones is sufficient to construct the central submonad and we provided a constructive proof of this fact. Next, we show that the existence of these central cones is also necessary for this. Furthermore, we provide another equivalent characterisation in terms of the premonoidal structure of the monad. This is precisely formulated in the main theorem of this paper which is presented next.

THEOREM 4.10 (CENTRALISABILITY). *Let \mathcal{C} be a symmetric monoidal category and \mathcal{T} a strong monad on it. The following are equivalent:*

- (1) *For any object X of \mathcal{C} , \mathcal{T} admits a terminal central cone at X ;*
- (2) *There exists a commutative submonad \mathcal{Z} of \mathcal{T} such that the canonical embedding functor $I : \mathcal{C}_{\mathcal{Z}} \rightarrow \mathcal{C}_{\mathcal{T}}$ corestricts to an isomorphism of categories $\mathcal{C}_{\mathcal{Z}} \cong Z(\mathcal{C}_{\mathcal{T}})$;*
- (3) *The corestriction of the Kleisli left adjoint $\mathcal{J} : \mathcal{C} \rightarrow \mathcal{C}_{\mathcal{T}}$ to the premonoidal centre $\hat{\mathcal{J}} : \mathcal{C} \rightarrow Z(\mathcal{C}_{\mathcal{T}})$ also is a left adjoint.*

PROOF.

(1 \Rightarrow 2) : By Theorem 4.8 and Theorem 4.9.

(2 \Rightarrow 3) : Let us consider the Kleisli left adjoint $\mathcal{J}^{\mathcal{Z}}$ associated to the monad \mathcal{Z} . All our hypotheses can be summarised by the diagram

$$\begin{array}{ccc}
 \mathcal{C} & \xrightarrow{\mathcal{J}} & \mathcal{C}_{\mathcal{T}} \\
 \mathcal{J}^{\mathcal{Z}} \downarrow & \searrow \hat{\mathcal{J}} & \uparrow \\
 \mathcal{C}_{\mathcal{Z}} & \xrightarrow[\hat{I}]{\cong} & Z(\mathcal{C}_{\mathcal{T}})
 \end{array}$$

where $\hat{I} : \mathcal{C}_{\mathcal{Z}} \cong Z(\mathcal{C}_{\mathcal{T}})$ is the corestriction of I . This diagram commutes, because \mathcal{Z} is a submonad of \mathcal{T} (recall also that $\hat{\mathcal{J}}$ is the indicated corestriction of \mathcal{J} , see §2.2). Since \hat{I} is an isomorphism, then $\hat{\mathcal{J}} = \hat{I} \circ \mathcal{J}^{\mathcal{Z}}$ is the composition of two left adjoints and it is therefore also a left adjoint.

(3 \Rightarrow 1) : Let $\mathcal{R} : Z(\mathcal{C}_{\mathcal{T}}) \rightarrow \mathcal{C}$ be the right adjoint of $\hat{\mathcal{J}}$ and let ε be the counit of the adjunction. We will show that the pair $(\mathcal{R}X, \varepsilon_X)$ is the terminal central cone of \mathcal{T} at X .

First, since ε_X is a morphism in $Z(\mathcal{C}_{\mathcal{T}})$, it follows that it is central. Thus the pair $(\mathcal{R}X, \varepsilon_X)$ is a central cone of \mathcal{T} at X . Next, let $\Phi : Z(\mathcal{C}_{\mathcal{T}})[\hat{\mathcal{J}}Y, X] \cong \mathcal{C}[Y, \mathcal{R}X]$ be the natural bijection induced by the adjunction. If $f : Y \rightarrow \mathcal{T}X$ is central, meaning a morphism of $Z(\mathcal{C}_{\mathcal{T}})$, the diagram below

left commutes in $Z(\mathcal{C}_{\mathcal{T}})$, or equivalently, the diagram below right commutes in \mathcal{C} :

$$\begin{array}{ccc}
 \hat{\mathcal{J}}Y & & Y \\
 \hat{\mathcal{J}}\Phi(f) \downarrow & \searrow f & \downarrow \Phi(f) \\
 \hat{\mathcal{J}}\mathcal{R}X & \xrightarrow{\varepsilon_X} & X \\
 & & \mathcal{R}X \xrightarrow{\varepsilon_X} \mathcal{T}X
 \end{array}$$

Note that the pair (Y, f) is equivalently a central cone for \mathcal{T} at X (by Proposition 4.3). Thus f uniquely factors through the counit $\varepsilon_X : \mathcal{R}X \rightarrow \mathcal{T}X$ and therefore $(\mathcal{R}X, \varepsilon_X)$ is the terminal central cone of \mathcal{T} at X . \square

This theorem shows that Definition 4.7 may be stated by choosing any one of the above equivalent criteria. We note that the first condition is the easiest to verify in practice. The second condition is the most useful for providing a computational interpretation, as we do in the sequel. The third condition provides an important link to premonoidal categories.

5 EXAMPLES

In this section we show how we can make use of the mathematical results we already established in order to reason about the centres of monads of interest.

5.1 A Non-centralisable Monad

In **Set**, we heavily relied on the notion of subset to define the central submonad. One may wonder what happens if not every subset of a given set is an object of the category. The following example describes such a situation, which gives rise to a non-centralisable strong monad.

Example 5.1. Consider the Dihedral group \mathbb{D}_4 , which has 8 elements. Its centre $Z(\mathbb{D}_4)$ is non-trivial and has 2 elements. Let \mathcal{C} be the full subcategory of **Set** with objects that are finite products of the set \mathbb{D}_4 with itself. This category has a cartesian structure and the terminal object is the singleton set (which is the empty product). Notice that every object in this category has cardinality which is a power of 8. Therefore the cardinality of every homset of \mathcal{C} is also a power of 8. Since \mathcal{C} has a cartesian structure and since \mathbb{D}_4 is a monoid, we can consider the writer monad $\mathcal{M} \stackrel{\text{def}}{=} (\mathbb{D}_4 \times -) : \mathcal{C} \rightarrow \mathcal{C}$ induced by \mathbb{D}_4 , which can be defined in exactly the same way as in Example 5.10 (see also §1.3). It follows that \mathcal{M} is a strong monad on \mathcal{C} . However, it is easy to show that this monad is not centralisable. Let us assume, for the purpose of reaching a contradiction, that there is a monad $\mathcal{Z} : \mathcal{C} \rightarrow \mathcal{C}$ such that $\mathcal{C}_{\mathcal{Z}} \cong Z(\mathcal{C}_{\mathcal{M}})$ (see Theorem 4.10). Next, observe that the homset $Z(\mathcal{C}_{\mathcal{M}})[1, 1]$ has the same cardinality as the centre of the monoid \mathbb{D}_4 , i.e., its cardinality is 2. However, $\mathcal{C}_{\mathcal{Z}}$ cannot have such a homset since $\mathcal{C}_{\mathcal{Z}}[X, Y] = \mathcal{C}[X, \mathcal{Z}Y]$ which must have cardinality a power of 8. Therefore there exists no such monad \mathcal{Z} and \mathcal{M} is not centralisable.

Besides this example and any further attempts at constructing non-centralisable monads for this sole purpose, we do not know of any other strong monad in the literature that is not centralisable. Throughout the remainder of the section, we present many examples of centralisable monads and classes of centralisable monads which show that our results are widely applicable.

5.2 Categories whose Strong Monads are Centralisable

We saw earlier that every (strong) monad on **Set** is centralisable. In fact, this is also true for many other naturally occurring categories. For example, in many categories of interest, the objects of the

category have a suitable notion of subobject (e.g., subsets in **Set**, subspaces in **Vect**) and the centre can then be constructed in a similar way than in **Set**.

Example 5.2. Every strong monad on the category **DCPO** (whose objects are directed-complete partial orders and the morphisms are Scott-continuous maps between them) is centralisable. The easiest way to see this is to use Theorem 4.10 (1). Writing $\mathcal{T} : \mathbf{DCPO} \rightarrow \mathbf{DCPO}$ for an arbitrary strong monad on **DCPO**, the terminal central cone of \mathcal{T} at X is given by the subdcpo $\mathcal{Z}X \subseteq \mathcal{T}X$ which has the underlying set

$$\mathcal{Z}X \stackrel{\text{def}}{=} \{t \in \mathcal{T}X \mid \forall Y \in \text{Ob}(\mathbf{DCPO}). \forall s \in \mathcal{T}Y. \mu(\mathcal{T}\tau'(\tau(t, s))) = \mu(\mathcal{T}\tau(\tau'(t, s)))\}.$$

That $\mathcal{Z}X$ (with the inherited order) is a subdcpo of $\mathcal{T}X$ follows easily by using the fact that μ, τ, τ' and \mathcal{T} are Scott-continuous. So, we see the construction is analogous to the one in **Set**, but with some additional proof obligations.

Example 5.3. Every strong monad on the category **Top** (whose objects are topological spaces and the morphisms are continuous maps between them) is centralisable. Using Theorem 4.10 (1) and writing $\mathcal{T} : \mathbf{Top} \rightarrow \mathbf{Top}$ for an arbitrary strong monad on **Top**, the terminal central cone of \mathcal{T} at X is given by the space $\mathcal{Z}X \subseteq \mathcal{T}X$ which has the underlying set

$$\mathcal{Z}X \stackrel{\text{def}}{=} \{t \in \mathcal{T}X \mid \forall Y \in \text{Ob}(\mathbf{Top}). \forall s \in \mathcal{T}Y. \mu(\mathcal{T}\tau'(\tau(t, s))) = \mu(\mathcal{T}\tau(\tau'(t, s)))\}$$

and whose topology is the subspace topology inherited from $\mathcal{T}X$ (recall that the subspace topology is the coarsest topology that makes the subset inclusion map continuous).

Example 5.4. Every strong monad on the category **Meas** (whose objects are measurable spaces and the morphisms are measurable maps between them) is centralisable. The construction is fully analogous to the previous example, but instead of the subspace topology, we equip the underlying set with the subspace σ -algebra inherited from $\mathcal{T}X$ (which is the smallest σ -algebra that makes the subset inclusion map measurable).

Example 5.5. Every strong monad on the category **Vect** (whose objects are vector spaces and the morphisms are linear maps between them) is centralisable. One simply defines the subset $\mathcal{Z}X$ as in the other examples and shows that this is a linear subspace of $\mathcal{T}X$. That this is the terminal central cone is then obvious (as in the other examples).

The above categories, together with the category **Set**, are not meant to provide an exhaustive list of categories for which all strong monads are centralisable. Indeed, there are many more categories for which we can carry on similar arguments. The purpose of these examples is to illustrate how we may use Theorem 4.10 (1) to construct the centre of a strong monad. Next, we show how we may use Theorem 4.10 (3) to show that a strong monad is centralisable. Our next proposition does exactly this.

PROPOSITION 5.6. *Let \mathbf{C} be a symmetric monoidal closed category that is total⁵. Then all strong monads over \mathbf{C} are centralisable.*

PROOF. For any strong monad $\mathcal{T} : \mathbf{C} \rightarrow \mathbf{C}$, we first prove that the corestriction of the Kleisli inclusion $\mathcal{J} : \mathbf{C} \rightarrow \mathbf{C}_{\mathcal{T}}$ to the premonoidal centre $\hat{\mathcal{J}} : \mathbf{C} \rightarrow Z(\mathbf{C}_{\mathcal{T}})$ also is cocontinuous. Then by the adjoint functor theorem for total categories [Street and Walters 1978], $\hat{\mathcal{J}}$ is a left adjoint, and by Theorem 4.10 (3) it follows that the corresponding strong monad is centralisable. See Appendix A for more details. \square

⁵Recall that a locally small category is total if its Yoneda embedding has a left adjoint.

As a special case of the above proposition, we get another proof of the fact that every monad on **Set** and **Vect** is centralisable.

Example 5.7. Any category which is the Eilenberg-Moore category of a commutative monad over **Set** is total [Kelly 1986]. Furthermore it is cocomplete [Hermelink 2019] and then symmetric monoidal closed [Keigher 1978], thus all strong monads on it are centralisable, this includes:

- The category \mathbf{Set}_* of pointed sets and point preserving functions (algebras of the lift monad).
- The category \mathbf{CMon} of commutative monoids and monoid homomorphisms (algebras of the commutative monoid monad).
- The category \mathbf{Conv} of convex sets and linear functions (algebras of the distribution monad).
- The category \mathbf{Sup} of complete semilattices and sup-preserving functions (algebras of the powerset monad).

Example 5.8. Any presheaf category $\mathbf{Set}^{C^{op}}$ over a small category C is total [Kelly 1986] and cartesian closed, thus all strong monads on it are centralisable, this includes:

- The category $\mathbf{Set}^{A^{op}}$, where A is the category with two objects and two parallel arrows, which can be seen as the category of directed multi-graphs and graph homomorphisms.
- The category $\mathbf{Set}^{G^{op}}$, where G is a group seen as a category, which can be seen as the category of G -sets (sets with an action of G) and equivariant maps.
- The topos of trees $\mathbf{Set}^{\mathbb{N}^{op}}$.

More generally the conditions of Proposition 5.6 are also satisfied by any Grothendieck topos.

5.3 Specific Examples of Centralisable Monads

In this subsection, we consider specific monads and construct their centres.

Example 5.9. Let \mathcal{T} be a commutative monad. Then its central submonad is \mathcal{T} itself.

Example 5.10. Given a monoid (M, e, m) , the free monad induced by M , also known as the *writer monad*, is the monad $\mathcal{T} = (- \times M) : \mathbf{Set} \rightarrow \mathbf{Set}$ whose monad data is given by:

- $\eta_X : X \rightarrow X \times M :: x \mapsto (x, e)$;
- $\mu_X : (X \times M) \times M \rightarrow X \times M :: ((x, z), z') \mapsto (x, m(z, z'))$;
- $\tau_{X,Y} : X \times (Y \times M) \rightarrow (X \times Y) \times M :: (x, (y, z)) \mapsto ((x, y), z)$.

The central submonad \mathcal{Z} of \mathcal{T} is given by the commutative monad $(- \times Z(M)) : \mathbf{Set} \rightarrow \mathbf{Set}$, where $Z(M)$ is the centre of the monoid M and where the monad data is given by the (co)restrictions of the monad data of \mathcal{T} . Note that \mathcal{T} is a commutative monad iff M is a commutative monoid.

Example 5.11. Let S be a set. The (well-known) continuation monad is given by the functor $\mathcal{T} = [[-, S], S] : \mathbf{Set} \rightarrow \mathbf{Set}$, equipped with the monad data:

- $\eta_X : X \rightarrow [[X, S], S] :: x \mapsto \lambda f. f(x)$;
- $\mu_X : [[[[X, S], S], S], S] \rightarrow [[X, S], S] :: F \mapsto \lambda g. F(\lambda h. h(g))$;
- $\tau_{X,Y} : X \times [[Y, S], S] \rightarrow [[X \times Y, S], S] :: (x, f) \mapsto \lambda g. f(\lambda y. g(x, y))$.

Note that, if S is the empty set or a singleton set, then \mathcal{T} is commutative, so we are in the situation of Example 5.9. Otherwise, when S is not trivial, one can prove (details omitted here) that $\mathcal{Z}X = \eta_X(X) \cong X$. Therefore, the central submonad of \mathcal{T} is trivial and it is naturally isomorphic to the identity monad.

Example 5.11 shows that the centre of a monad may be trivial in the sense that it is precisely the image of the monadic unit, and by Remark 3.3, this is the least it can be. Therefore, the central submonad of such a monad is not very useful, because it does not contain any additional information

about the nature of the specific monadic effect. At the other extreme, Example 5.9 shows that the centre of a commutative monad coincides with itself (as one would expect) and therefore we also do not get anything new. Therefore, the monads that have interesting central submonads are those monads which are strong, but not commutative, and which have non-trivial centres, such as the one in Example 5.10. Another interesting example of a strong monad with a non-trivial centre is provided next.

Example 5.12. Every semiring $(S, +, 0, \cdot, 1)$ induces a monad $\mathcal{T} : \mathbf{Set} \rightarrow \mathbf{Set}$ [Jakl et al. 2022]. This monad maps a set X to the set of finite formal sums of the form $\sum s_i x_i \in \mathcal{T}X$, where s_i are elements of S and x_i are elements of X . The unit of the monad $\eta_X : X \rightarrow \mathcal{T}X$ maps an element of X to a singleton sum. The multiplication $\mu_X : \mathcal{T}^2 X \rightarrow \mathcal{T}X$ takes a sum of sums as an input and flattens it to a single sum: $\sum_i s_i (\sum_j s_{i,j} x_{i,j}) \mapsto \sum_{i,j} (s_i \cdot s_{i,j}) x_{i,j}$. The strength $\tau_{X,Y} : X \times \mathcal{T}Y \rightarrow \mathcal{T}(X \times Y)$ and the costrength $\tau'_{X,Y} : \mathcal{T}X \otimes Y \rightarrow \mathcal{T}(X \otimes Y)$ are respectively defined as follows: $(x, \sum_i s_i y_i) \mapsto \sum_i s_i (x, y_i)$ and $(\sum_i s_i x_i, y) \mapsto \sum_i s_i (x_i, y)$. The monad \mathcal{T} is commutative if and only if S is commutative. One can prove that the central submonad \mathcal{Z} of \mathcal{T} is induced by the commutative semiring $Z(S)$, made of elements of S that commute with any other element of S through its multiplication $(-\cdot-)$. Thus for a given set X , the central submonad is $\mathcal{Z}X = \{\sum_{i \leq n} s_i x_i \mid n \in \mathbb{N}, s \in Z(S)^n, x \in X^n\}$.

Next, we consider an important example from domain theory [Gierz et al. 2012].

Example 5.13. The valuations monad $\mathcal{V} : \mathbf{DCPO} \rightarrow \mathbf{DCPO}$ [Jones 1990; Jones and Plotkin 1989] is similar in spirit to the Giry monad on measurable spaces [Giry 1982]. It is used to combine probability and recursion for dcpo's. Given a dcpo X , the valuations monad \mathcal{V} assigns the dcpo $\mathcal{V}X$ of all Scott-continuous *valuations* on X , which are Scott-continuous functions $\nu : \sigma(X) \rightarrow [0, 1]$ from the Scott-open sets of X into the unit interval that satisfy some additional properties that make them suitable to model probability (details omitted here, see [Jones 1990] for more information). The category \mathbf{DCPO} is cartesian closed and the valuations monad $\mathcal{V} : \mathbf{DCPO} \rightarrow \mathbf{DCPO}$ is strong, but its commutativity on \mathbf{DCPO} has been an open problem since 1989 [Jones 1990; Jones and Plotkin 1989]. Multiple experts in the field believe that it is *not* commutative⁶. More recently, progress has been made in identifying commutative submonads of \mathcal{V} on the category \mathbf{DCPO} [Goubault-Larrecq et al. 2021; Jia et al. 2021a,b]. The difficulty in (dis)proving the commutativity of \mathcal{V} boils down to (dis)proving the following Fubini-style equation

$$\int_X \int_Y \chi_U(x, y) d\nu d\xi = \int_Y \int_X \chi_U(x, y) d\xi d\nu$$

holds for any dcpo's X and Y , any Scott-open subset $U \in \sigma(X \times Y)$ and any two valuations $\xi \in \mathcal{V}X$ and $\nu \in \mathcal{V}Y$. In the above equation, the notion of integration is given by the *valuation integral* (see [Jones 1990] for more information). Several different submonads of \mathcal{V} are constructed in [Goubault-Larrecq et al. 2021; Jia et al. 2021a] by using topological methods that allow the authors to restrict the space of valuations to particularly nice ones where the above Fubini-style equation may be shown to hold.

However, the authors of [Jia et al. 2021b] take a different approach which is of a more algebraic nature. The *central valuations monad*, as defined in [Jia et al. 2021b], is defined to be the monad $\mathcal{Z} : \mathbf{DCPO} \rightarrow \mathbf{DCPO}$ which maps a dcpo X to the dcpo $\mathcal{Z}X$ which has all *central valuations* as

⁶Personal Communication.

elements. More precisely, their definition is equivalent to writing:

$$\mathcal{Z}X \stackrel{\text{def}}{=} \left\{ \xi \in \mathcal{V}(X) \mid \forall Y \in \text{Ob}(\mathbf{DCPO}). \forall U \in \sigma(X \times Y). \forall v \in \mathcal{V}(Y). \right. \\ \left. \int_X \int_Y \chi_U(x, y) dv d\xi = \int_Y \int_X \chi_U(x, y) d\xi dv \right\}.$$

But this is precisely the central submonad of \mathcal{V} , which can be seen using Theorem 4.10 (1) after unpacking the definition of the monad data of \mathcal{V} . Therefore, we see that the main result of [Jia et al. 2021b] is a special case of our more general categorical treatment. We wish to note, that the central submonad of \mathcal{V} is not trivial, but it is actually quite large. It contains all three commutative submonads identified in [Jia et al. 2021a], neither of which are trivial, and all of which may be used to model lambda calculi with recursion and discrete probabilistic choice (see [Jia et al. 2021a,b]).

6 COMPUTATIONAL INTERPRETATION

In this section we provide a computational interpretation of our ideas by presenting a refinement of Moggi's metalanguage [Moggi 1991].

6.1 Syntax

We begin by describing the type system. The grammar of types (see Figure 2) are just the usual ones with one addition – we extend the grammar by adding the family of types $\mathcal{Z}A$. The types should be understood in the following way: 1 is the unit type; $A \times B$ represents pair types; $A + B$ represents sum types; $A \rightarrow B$ represents function types; $\mathcal{T}A$ represents the type of monadic computations for our monad \mathcal{T} that produce values of type A (together with a potential side effect described by \mathcal{T}); $\mathcal{Z}A$ represent the type of *central* monadic computations for our monad \mathcal{T} that produce values of type A (together with a potential *central* side effect that is necessarily in \mathcal{Z}).

The grammar of terms for our system are described in Figure 2 and the formation rules for well-formed terms are described in Figure 3. The first nine rules in Figure 3 are just the usual formation rules for a simply-typed lambda calculus with pair types and sum types. We focus on the terms for monadic computation.

The $\text{ret } M$ term is used as an introduction rule for the monadic types and it allows us to see the pure (i.e., non-effectful) computation described by the term M as a monadic one. Semantically, this corresponds to the application of the monadic unit and it is written $\text{ret}_{\mathcal{Z}}$ because the monadic unit is always central (and so is the pure computation described by the term M). The term ιM allows us to view a *central* monadic computation as a monadic (not necessarily central) one. Semantically, it corresponds to an application of the ι submonad inclusion. Because of this, we can introduce some syntactic sugar and define the term $\text{ret}_{\mathcal{T}} M \stackrel{\text{def}}{=} \iota \text{ret}_{\mathcal{Z}} M$. Finally, we have two terms for monadic sequencing that use the familiar do -notation. The monadic sequencing of two central computations remains central, which is represented via the $\text{do}_{\mathcal{Z}}$ terms; the $\text{do}_{\mathcal{T}}$ terms are used for monadic sequencing of (not necessarily central) computations, as usual.

For simplicity, and for brevity, we do not add any specific constants of monadic types $\mathcal{Z}A$ and $\mathcal{T}A$ to keep our presentation more manageable. Of course, it is clear, that in practice this has to be done when a specific monad \mathcal{T} is chosen and these constants will be application specific. We provide such a use case via the writer monad in §7, where we formally present a suitable operational semantics together with soundness and adequacy results.

(Types) $A, B ::= 1 \mid A \rightarrow B \mid A \times B \mid A + B \mid \mathcal{Z}A \mid \mathcal{T}A$
 (Terms) $M, N ::= x \mid * \mid \lambda x^A.M \mid MN \mid \langle M, N \rangle \mid \pi_i M \mid \text{left } M \mid \text{right } M$
 $\mid \text{case } M \text{ of } \{ \text{left } x \rightarrow N \parallel \text{right } x' \rightarrow N' \}$
 $\mid \text{ret}_{\mathcal{Z}} M \mid \iota M \mid \text{do}_{\mathcal{Z}} x \leftarrow M; N \mid \text{do}_{\mathcal{T}} x \leftarrow M; N$

Fig. 2. Grammars for the types and terms of our system.

$$\begin{array}{c}
 \frac{}{\Gamma, x : A \vdash x : A} \quad \frac{}{\Gamma \vdash * : 1} \quad \frac{\Gamma \vdash M : A \rightarrow B \quad \Gamma \vdash N : A}{\Gamma \vdash MN : B} \quad \frac{\Gamma \vdash M : A \quad \Gamma \vdash N : B}{\Gamma \vdash \langle M, N \rangle : A \times B} \\
 \frac{\Gamma \vdash M : A_1 \times A_2}{\Gamma \vdash \pi_i M : A_i} \quad \frac{\Gamma, x : A \vdash M : B}{\Gamma \vdash \lambda x^A.M : A \rightarrow B} \quad \frac{\Gamma \vdash M : A}{\Gamma \vdash \text{left } M : A + B} \quad \frac{\Gamma \vdash M : B}{\Gamma \vdash \text{right } M : A + B} \\
 \frac{\Gamma \vdash M : A + B \quad \Gamma, x : A \vdash N : C \quad \Gamma, x' : B \vdash N' : C}{\text{case } M \text{ of } \{ \text{left } x \rightarrow N \parallel \text{right } x' \rightarrow N' \} : C} \quad \frac{\Gamma \vdash M : A}{\Gamma \vdash \text{ret}_{\mathcal{Z}} M : \mathcal{Z}A} \\
 \frac{\Gamma \vdash M : \mathcal{Z}A \quad \Gamma, x : A \vdash N : \mathcal{Z}B}{\Gamma \vdash \text{do}_{\mathcal{Z}} x \leftarrow M; N : \mathcal{Z}B} \quad \frac{\Gamma \vdash M : \mathcal{Z}A}{\Gamma \vdash \iota M : \mathcal{T}A} \quad \frac{\Gamma \vdash M : \mathcal{T}A \quad \Gamma, x : A \vdash N : \mathcal{T}B}{\Gamma \vdash \text{do}_{\mathcal{T}} x \leftarrow M; N : \mathcal{T}B}
 \end{array}$$

Fig. 3. Formation rules for well-formed terms.

6.2 Denotational semantics

The categorical model that we use to interpret our type system consists of the following data: a cartesian closed category \mathbf{C} with coproducts and a centralisable strong monad \mathcal{T} , whose central submonad is denoted \mathcal{Z} and whose submonad inclusion is given by $\iota : \mathcal{Z} \Rightarrow \mathcal{T}$. We write π_A to indicate the projections for the cartesian product and we write in_i for the coproduct injections. In this situation, the cartesian product distributes over the coproduct, and we write the corresponding isomorphism as $\delta : A \times (B + C) \cong (A + B) \times (A + C)$. We write $\lambda : \mathbf{C}[X \times Y, Z] \cong \mathbf{C}[X, Z^Y]$ for the currying natural isomorphism and we write ev for the corresponding evaluation map. The types are interpreted as objects in \mathbf{C} , as usual:

$$\begin{aligned}
 \llbracket 1 \rrbracket &= 1 \\
 \llbracket A \rightarrow B \rrbracket &= \llbracket B \rrbracket^{\llbracket A \rrbracket} \\
 \llbracket A + B \rrbracket &= \llbracket A \rrbracket + \llbracket B \rrbracket \\
 \llbracket A \times B \rrbracket &= \llbracket A \rrbracket \times \llbracket B \rrbracket \\
 \llbracket \mathcal{Z}A \rrbracket &= \mathcal{Z} \llbracket A \rrbracket \\
 \llbracket \mathcal{T}A \rrbracket &= \mathcal{T} \llbracket A \rrbracket
 \end{aligned}$$

Variable contexts $\Gamma = x_1 : A_1 \dots x_n : A_n$ are interpreted as usual as $\llbracket \Gamma \rrbracket \stackrel{\text{def}}{=} \llbracket A_1 \rrbracket \times \dots \times \llbracket A_n \rrbracket$. Terms are interpreted as morphisms $\llbracket \Gamma \vdash M : A \rrbracket : \llbracket \Gamma \rrbracket \rightarrow \llbracket A \rrbracket$ of \mathbf{C} . When the context and the type of a term M are understood, then we simply write $\llbracket M \rrbracket$ as a shorthand for $\llbracket \Gamma \vdash M : A \rrbracket$. The

interpretation of the terms is defined by induction on the typing derivation as follows:

$$\begin{aligned}
\llbracket \Gamma, x : A \vdash x : A \rrbracket &= \pi_{[A]} & \llbracket \Gamma \vdash \pi_i M : A \rrbracket &= \pi_{[A]} \circ \llbracket M \rrbracket & \llbracket \Gamma \vdash \langle M, N \rangle : A \times B \rrbracket &= \langle \llbracket M \rrbracket, \llbracket N \rrbracket \rangle \\
\llbracket \Gamma \vdash MN : B \rrbracket &= \text{ev}_{[A],[B]} \circ \langle \llbracket M \rrbracket, \llbracket N \rrbracket \rangle & \llbracket \Gamma \vdash \lambda x^A. M : A \rightarrow B \rrbracket &= \lambda_{[\Gamma],[A],[B]}(\llbracket M \rrbracket) \\
\llbracket \Gamma \vdash \text{left } M : A + B \rrbracket &= \text{in}_1 \circ \llbracket M \rrbracket & \llbracket \Gamma \vdash \text{right } M : A + B \rrbracket &= \text{in}_2 \circ \llbracket M \rrbracket \\
\llbracket \Gamma \vdash \text{case } M \text{ of } \{ \text{left } x \rightarrow N \mid \text{right } x' \rightarrow N' \} : A \rrbracket &= \llbracket \llbracket N \rrbracket, \llbracket N' \rrbracket \rrbracket \circ \delta_{[\Gamma],[A],[B]} \circ \langle \text{id}, \llbracket M \rrbracket \rangle \\
\llbracket \Gamma \vdash \text{ret}_{\mathcal{Z}} M : \mathcal{Z}A \rrbracket &= \eta_{[A]}^{\mathcal{Z}} \circ \llbracket M \rrbracket & \llbracket \Gamma \vdash \iota M : \mathcal{T}A \rrbracket &= \iota_{[A]} \circ \llbracket M \rrbracket \\
\llbracket \Gamma \vdash \text{do}_{\mathcal{Z}} x \leftarrow M ; N : \mathcal{Z}B \rrbracket &= \mu_{[B]}^{\mathcal{Z}} \circ \mathcal{Z} \llbracket N \rrbracket \circ \tau_{[\Gamma],[A]}^{\mathcal{Z}} \circ \langle \text{id}, \llbracket M \rrbracket \rangle \\
\llbracket \Gamma \vdash \text{do}_{\mathcal{T}} x \leftarrow M ; N : \mathcal{T}B \rrbracket &= \mu_{[B]} \circ \mathcal{T} \llbracket N \rrbracket \circ \tau_{[\Gamma],[A]} \circ \langle \text{id}, \llbracket M \rrbracket \rangle
\end{aligned}$$

6.3 Observational equivalence

Our system and our semantics can be used to validate some important equivalences.

PROPOSITION 6.1. *The following equalities:*

$$\begin{aligned}
\llbracket \iota(\text{ret}_{\mathcal{Z}} M) \rrbracket &= \llbracket \text{ret}_{\mathcal{T}} M \rrbracket \\
\llbracket \text{do}_{\mathcal{T}} x \leftarrow \iota M ; \iota N \rrbracket &= \llbracket \iota(\text{do}_{\mathcal{Z}} x \leftarrow M ; N) \rrbracket \\
\llbracket \text{do}_{\mathcal{T}} x \leftarrow \iota M ; \text{do}_{\mathcal{T}} y \leftarrow N ; P \rrbracket &= \llbracket \text{do}_{\mathcal{T}} y \leftarrow N ; \text{do}_{\mathcal{T}} x \leftarrow \iota M ; P \rrbracket
\end{aligned}$$

hold in our categorical model.

PROOF. The first equation is obvious. The second one is a direct consequence of the coherence properties between ι and μ and τ . The last equality follows directly by the naturality of τ and of course, the fact that ι is central. \square

We now explain the importance of the above proposition. Assuming that our system is equipped with an operational semantics that is sound and adequate with respect to the denotational semantics (which is the case for our case study in the next section), then standard arguments may be used to show that the terms we identified above are *contextually equivalent* (or *observationally equivalent*). This means that the corresponding terms cannot be distinguished from the point of view of an observer who examines the result of the computation in any possible observable context. These equivalences might therefore be used to perform program optimisation or program transformation in a safe way.

A few more words on the operational significance of the second and third equations. The second one shows that central computations may continue to be seen as central even when we use monadic sequencing of the ambient monad \mathcal{T} . The third equation is perhaps the one of highest interest. It shows that central computations commute with any other (not necessarily central) computation. Note that this holds even when the monad \mathcal{T} is just strong, not necessarily commutative, and this allows us to exploit this useful equivalence in a wider range of use cases. Furthermore, in practice, this equivalence may be easily established by just looking at the more refined syntactic structure (due to the ιM terms) and without having to compute the semantic interpretation of the corresponding terms.

Of course, the above equivalences are not meant to provide an exhaustive list, but only an illustrative selection. For example, we may also prove that

$$\llbracket \text{case } M \text{ of } \{ \text{left } x \rightarrow \iota N \mid \text{right } x' \rightarrow \iota N' \} \rrbracket = \llbracket \iota(\text{case } M \text{ of } \{ \text{left } x \rightarrow N \mid \text{right } x' \rightarrow N' \}) \rrbracket$$

any many others, but for brevity, we keep our selection concise.

7 CASE STUDY: WRITER MONAD

In this section we illustrate our ideas by fixing a specific monad – the well-known writer monad from Example 5.10 (also known as the *action* monad). We fix a monoid $(\mathcal{M}, \bullet, e)$, with a non-trivial centre $Z(\mathcal{M})$. The elements of the monoid may be thought of as representing external actions on some system. For generality, we do not specify here what these actions are.

The possible actions are introduced into our language via terms $\text{act}(c)$ in the syntax where c ranges over elements of \mathcal{M} . These actions are viewed as computational effects and can be cleanly and systematically introduced into our system via the writer monad $\mathcal{T} \stackrel{\text{def}}{=} (- \times \mathcal{M}) : \mathbf{Set} \rightarrow \mathbf{Set}$ and its central submonad $\mathcal{Z} \stackrel{\text{def}}{=} (- \times Z(\mathcal{M})) : \mathbf{Set} \rightarrow \mathbf{Set}$.

7.1 Syntax

The syntax is the same as in the previous section and we simply add a new term to represent the possible actions. The grammar of terms is extended by adding

$$(\text{Terms}) M, N ::= \dots \mid \text{act}(c)$$

where c is an element of \mathcal{M} . The formation rules of the type system are the same as in the previous section together with the following two additions for the newly added terms:

$$\frac{c \in \mathcal{M} \setminus Z(\mathcal{M})}{\Gamma \vdash \text{act}(c) : \mathcal{T}1} \quad \frac{c \in Z(\mathcal{M})}{\Gamma \vdash \text{act}(c) : \mathcal{Z}1}$$

The two formation rules distinguish between the central and non-central elements of \mathcal{M} . The corresponding types are $\mathcal{T}1$ and $\mathcal{Z}1$, because these terms do not return any values, they only perform an action on the external system.

NOTATION 7.1. *In what follows, much of the treatment is common to both monads \mathcal{T} and \mathcal{Z} . We use the symbol X to refer to either one of the two monads.*

In order to define an operational semantics for this (effectful) language, we have to consider *program configurations* that represent the current state of execution. Program configurations are pairs (M, c) of a term M and an element $c \in \mathcal{M}$. A configuration (M, c) is *well-formed* of type XA , written $(M, c) : XA$, whenever $\cdot \vdash M : XA$, i.e., when M is a closed term of type XA , which we also write as $\vdash M : XA$. Note that we only consider configurations of monadic type to be well-formed, because those can potentially perform effectful actions, whereas configurations of non-monadic types are pure and will never act on the external system via the monoid \mathcal{M} . Finally, we add some syntactic sugar to help the reader, by setting “ $\text{do}_X \text{act}(c) ; N \stackrel{\text{def}}{=} \text{do}_X x \leftarrow \text{act}(c) ; N$ ”, where the variable x of type 1 should be chosen to be fresh such that it does not appear freely in the term N . Recall also that we use the syntactic sugar $\text{ret}_{\mathcal{T}} M \stackrel{\text{def}}{=} \iota \text{ret}_{\mathcal{Z}} M$.

7.2 Operational Semantics

The operational semantics has two different modes of reduction. The first one is on pure (non-monadic) terms, which we call *pure reduction*, which is given by the usual reduction rules for a

call-by-name lambda calculus:

$$\frac{}{(\lambda x^A.M)N \rightarrow M[N/x]} \quad \frac{M \rightarrow M'}{\pi_i M \rightarrow \pi_i M'} \quad \frac{}{\pi_i \langle M_1, M_2 \rangle \rightarrow M_i} \quad \frac{M \rightarrow M'}{MN \rightarrow M'N}$$

$$\frac{M \rightarrow M'}{\text{case } M \text{ of } \{ \text{left } x \rightarrow N \mid \text{right } x' \rightarrow N' \} \rightarrow \text{case } M' \text{ of } \{ \text{left } x \rightarrow N \mid \text{right } x' \rightarrow N' \}}$$

$$\frac{}{\text{case (left } M) \text{ of } \{ \text{left } x \rightarrow N \mid \text{right } x' \rightarrow N' \} \rightarrow N[M/x]}$$

$$\frac{}{\text{case (right } M) \text{ of } \{ \text{left } x \rightarrow N \mid \text{right } x' \rightarrow N' \} \rightarrow N'[M/x']}$$

The second mode of reduction is on configurations and thus it represents effectful computation.

$$\frac{M \rightarrow M'}{(M, c) \rightsquigarrow (M', c)} \quad \frac{(M, c) \rightsquigarrow (M', c')}{(\iota M, c) \rightsquigarrow (\iota M', c')} \quad \frac{}{(\text{act}(c), c') \rightsquigarrow (\text{ret}_X *, c' \bullet c)}$$

$$\frac{(M, c) \rightsquigarrow (M', c')}{(\text{do}_X x \leftarrow M; N, c) \rightsquigarrow (\text{do}_X x \leftarrow M'; N, c')} \quad \frac{}{(\text{do}_X x \leftarrow \text{ret}_X M; N, c) \rightsquigarrow (N[M/x], c)}$$

We also have to specify what are the normal forms for the two modes of reduction. The normal forms for pure reduction are just the usual ones for the call-by-name lambda calculus together with the newly added monadic terms (because those cannot be reduced in the pure reduction mode). The normal forms for effectful reduction (on configurations) are determined by the $\text{ret}_X M$ terms.

$$\begin{aligned} \text{(Pure reduction NF)} \quad V, W ::= & * \mid \lambda x^A.M \mid \langle M, N \rangle \mid \text{left } M \mid \text{right } M \\ & \mid \text{do}_X x \leftarrow M; N \mid \text{act}(c) \mid \text{ret}_X M \mid \iota M \\ \text{(Effectful reduction NF)} \quad \mathcal{V}, \mathcal{W} ::= & (\text{ret}_X M, c) \end{aligned}$$

The next two lemmas show that our language is type-safe.

LEMMA 7.2 (TYPE PRESERVATION).

- If $\Gamma \vdash M : A$ and $M \rightarrow M'$, then $\Gamma \vdash M' : A$.
- If $(M, c) : \mathcal{X}A$ and $(M, c) \rightsquigarrow (M', c')$, then $(M', c') : \mathcal{X}A$.

LEMMA 7.3 (PROGRESS).

- If $\cdot \vdash M : A$ is a closed term, then either there exists M' such that $M \rightarrow M'$ or M is a pure reduction normal form.
- If $(M, c) : \mathcal{X}A$, then either there exists (M', c') such that $(M, c) \rightsquigarrow (M', c')$ or (M, c) is an effectful normal form.

The final lemma in this subsection also should not be surprising.

LEMMA 7.4 (STRONG NORMALISATION). *Pure reductions and effectful reductions are strongly normalising.*

7.3 Denotational Semantics

The interpretation of types and terms is the same as in the previous section, where the category \mathbf{C} is now fixed to \mathbf{Set} and the monads \mathcal{T} and \mathcal{Z} are fixed to the writer monad and its central submonad specified at the beginning of this section. The newly added terms are interpreted as follows

$$\llbracket \Gamma \vdash \text{act}(c) : \mathcal{T}1 \rrbracket \stackrel{\text{def}}{=} \underline{c}: \llbracket \Gamma \rrbracket \rightarrow 1 \times \mathcal{M} :: \gamma \mapsto (*, c)$$

$$\llbracket \Gamma \vdash \text{act}(c) : \mathcal{Z}1 \rrbracket \stackrel{\text{def}}{=} \underline{c}: \llbracket \Gamma \rrbracket \rightarrow 1 \times Z(\mathcal{M}) :: \gamma \mapsto (*, c)$$

where we abuse notation and write \underline{c} for both of the indicated constant maps. The denotation of a well-formed configuration (M, c) is defined through the Kleisli composition of \underline{c} and $\llbracket M \rrbracket$ as follows:

$$\llbracket (M, c) : \mathcal{T}A \rrbracket = \mu_A \circ (\llbracket M \rrbracket \times \text{id}_{\mathcal{M}}) \circ \underline{c} \qquad \llbracket (M, c) : \mathcal{Z}A \rrbracket = \mu_A^Z \circ (\llbracket M \rrbracket \times \text{id}_{Z(\mathcal{M})}) \circ \underline{c}.$$

THEOREM 7.5 (SOUNDNESS). *If $\Gamma \vdash M : A$ and $M \rightarrow M'$, then $\llbracket M \rrbracket = \llbracket M' \rrbracket$. Furthermore, if $(M, c) : \mathcal{X}A$ and $(M, c) \rightsquigarrow (M', c')$, then $\llbracket (M, c) \rrbracket = \llbracket (M', c') \rrbracket$.*

Thanks to strong normalisation, we can now define the overall *action* of a term $M : \mathcal{X}A$. By writing \rightsquigarrow^* for the reflexive and transitive closure of \rightsquigarrow , we may define

$$\text{action}(M) \stackrel{\text{def}}{=} c, \text{ where } c \text{ is the unique monoid element, s.t. } (M, e) \rightsquigarrow^* (\text{ret}_{\mathcal{X}} M', c).$$

THEOREM 7.6 (ADEQUACY). *If $\vdash M : \mathcal{X}1$, then $\llbracket M \rrbracket (*) = (*, \text{action}(M))$.*

Since our semantics is sound and adequate, standard arguments may be used to show that the observational equivalences suggested in §6 hold for this language. In particular, this means that monadic actions that make use of the central elements may be freely interchanged with any other (not necessarily central) monadic actions and this will not affect the observational behaviour of the programs.

8 RELATED WORK

The work which is closest to ours is [Power and Robinson 1997] which introduces premonoidal categories. We have already established important links between our development and the premonoidal centre (Theorem 4.10). While premonoidal categories have been influential in our understanding of effectful computation, it was less clear how to formulate an appropriate computational interpretation of the premonoidal centre for higher-order languages. Our paper shows that under some mild assumptions (which are easily satisfied see §5), the premonoidal centre of the Kleisli category of a strong monad induces an adjunction into the base category (Theorem 4.10) and this allows us to formulate a suitable computational interpretation by using monads, which are already well-understood [Moggi 1989, 1991] and well-integrated into many programming languages [Benton 2015].

Other related work includes [Staton and Levy 2013] where Staton and Levy introduce the novel notion of *premulticategories* in order to axiomatise impure/effectful computation in programming languages. What they achieve is to show how both monads and premonoidal categories arise in terms of universal properties of premulticategories. The notion of centrality plays an important role in the development of the theory there as well. Its computational interpretation aligns with the expected one, i.e., certain terms always commute with other effectful/impure terms. However, they do not focus, like us, on providing suitable programming abstractions that identify both central and non-central computations (e.g., by separating them into different types like us) and from what we can tell from our reading, there are no universal properties stated for the collection of central morphisms. Indeed, this is one of the main results of our paper (see Theorem 4.10) and it would be interesting to consider, as part of future work, whether similar results can be established in the framework of premulticategories. Also, our results provide a computational interpretation in terms of monads, which are standard and well-understood, so we think it is easier to incorporate them into existing languages.

A notion of commutants for enriched algebraic theories has been defined in [Lucyshyn-Wright 2018] from which the author derives a notion of centre of an enriched algebraic theory. In the cases of enriched monads, in other words, strong monads, arising from enriched algebraic theories, his notion of commutant extends to monad morphisms. While not explicitly stated in the paper, applying the commutant construction on the identity monad morphism from a monad to itself

provides a notion of centre of a monad that appears to coincide with ours. The existence of such commutants is also not automatic and depends on conditions similar to our centralisability conditions given in Theorem 4.10. The main differences between that result and ours is that the approach in [Lucyshyn-Wright 2018] relies on algebraic theories while our definition applies directly to the monad formalism. Furthermore, Lucyshyn-Wright’s work does not provide any computational interpretation and it does not provide any links with the premonoidal centre of [Power and Robinson 1997], both of which are a major focus of our work. Another difference is that we construct the centre in very different ways – in our case through central cones (which are novel) or through the premonoidal centre, whereas [Lucyshyn-Wright 2018] uses other methods based on algebraic theories.

Identifying commutative submonads of a given strong monad has been done elsewhere, e.g., [Goubault-Larrecq et al. 2021; Vákár et al. 2019]. The cited works do not produce the central submonad of the ambient one, but rather a different commutative submonad. Compared to these works, our method presented here works in a more general sense (due to the mild assumptions that are required), as we have already shown, but we only construct the central submonad. Another major difference, is that the cited works are concerned only with the constructed submonads for applications, whereas in our development we use both the ambient monad and its central submonad. Indeed, one of the main applications of our approach is to identify both the central and non-central monadic operations and use both to our advantage.

9 CONCLUSION AND FUTURE WORK

In this paper we asked an interesting algebraic question: *"Can the notion of centre be extended to monads?"* We showed that, under some mild assumptions, strong monads do indeed admit a centre, which is a commutative submonad, and we provided three equivalent characterisations for the existence of this centre (Theorem 4.10) which also establish important links to the theory of premonoidal categories. In particular, every (canonically strong) monad on **Set** is centralisable (§3) and we showed that the same is true for many other categories of interest (§5.2) and we identified specific monads with interesting centres (§5.3). We provided a computational interpretation of our ideas (§6) which has the added benefit of allowing us to easily establish important contextual equivalences (Proposition 6.1), where we may commute any central effectful operation with any other (not necessarily central) effectful operation. We also illustrated our ideas through a case study involving the writer monad that shows how to exploit the contextual equivalences we previously identified (§7).

For future work, it would be interesting to identify more monads that have non-trivial centres and to consider similar case studies for them as we did in §7. Another possible direction for future work is to find additional sufficient conditions for a symmetric monoidal category \mathbf{C} , such that every strong monad on it is centralisable. This would complement Proposition 5.6. Yet another possible direction is to consider how to develop a suitable theory of *commutants* of submonads (in the spirit of [Lucyshyn-Wright 2018]) and their computational interpretation. This is challenging mathematically, because it would be nice to identify under what conditions these commutants exist and determine commutative submonads. It is also challenging from a computational perspective, because the class of effectful operations described by a commutant does not necessarily commute with *all* other effectful operations. Another opportunity for future work includes studying the relationship between the centres of strong monads and distributive laws. In particular, given two strong monads and a strong/commutative distributive law between them, can we show that the distributive law also holds for their centres? If so, this would allow us to use the distributive law to combine not just the original monads, but their centres as well.

REFERENCES

- Nick Benton. 2015. Categorical monads and computer programming. *LMS Impact150 Stories* 1 (2015), 9–13.
- Gerhard Gierz, Karl Heinrich Hofmann, Klaus Keimel, Jimmie D Lawson, Michael Mislove, and Dana S Scott. 2012. *A compendium of continuous lattices*. Springer Science & Business Media.
- Michèle Giry. 1982. A categorical approach to probability theory. In *Categorical Aspects of Topology and Analysis*, B. Banaschewski (Ed.). Springer Berlin Heidelberg, Berlin, Heidelberg, 68–85.
- Jean Goubault-Larrecq, Xiaodong Jia, and Clément Théron. 2021. A Domain-Theoretic Approach to Statistical Programming Languages. *CoRR abs/2106.16190* (2021). arXiv:2106.16190 <https://arxiv.org/abs/2106.16190>
- Ada Hermelink. 2019. Cocompleteness of Categories of Algebras-An Overview. (2019).
- Bart Jacobs. 2016. *Introduction to Coalgebra: Towards Mathematics of States and Observation*. Cambridge Tracts in Theoretical Computer Science, Vol. 59. Cambridge University Press. <https://doi.org/10.1017/CBO9781316823187>
- Tomáš Jakl, Dan Marsden, and Nihil Shah. 2022. Generalizations of Bilinear Maps – Technical Report. <https://doi.org/10.48550/ARXIV.2205.05382>
- Xiaodong Jia, Bert Lindenhovius, Michael W. Mislove, and Vladimir Zamdzhiev. 2021a. Commutative Monads for Probabilistic Programming Languages. In *36th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2021, Rome, Italy, June 29 - July 2, 2021*. IEEE, 1–14. <https://doi.org/10.1109/LICS52264.2021.9470611>
- Xiaodong Jia, Michael W. Mislove, and Vladimir Zamdzhiev. 2021b. The Central Valuations Monad (Early Ideas). In *9th Conference on Algebra and Coalgebra in Computer Science, CALCO 2021, August 31 to September 3, 2021, Salzburg, Austria (LIPIcs, Vol. 211)*, Fabio Gadducci and Alexandra Silva (Eds.). Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 18:1–18:5. <https://doi.org/10.4230/LIPIcs.CALCO.2021.18>
- Claire Jones. 1990. *Probabilistic Non-determinism*. Ph. D. Dissertation. University of Edinburgh, UK. <http://hdl.handle.net/1842/413>
- C. Jones and Gordon D. Plotkin. 1989. A Probabilistic Powerdomain of Evaluations. In *Proceedings of the Fourth Annual Symposium on Logic in Computer Science (LICS '89), Pacific Grove, California, USA, June 5-8, 1989*. IEEE Computer Society, 186–195. <https://doi.org/10.1109/LICS.1989.39173>
- William F Keigher. 1978. Symmetric monoidal closed categories generated by commutative adjoint monads. *Cahiers de topologie et géométrie différentielle catégoriques* 19, 3 (1978), 269–293.
- G. M. Kelly. 1986. A survey of totality for enriched and ordinary categories. *Cahiers de Topologie et Géométrie Différentielle Catégoriques* 27 (1986), 109–132.
- Bartek Klin and Julian Salamanca. 2018. Iterated Covariant Powerset is not a Monad. In *Proceedings of the Thirty-Fourth Conference on the Mathematical Foundations of Programming Semantics, MFPS 2018, Dalhousie University, Halifax, Canada, June 6-9, 2018 (Electronic Notes in Theoretical Computer Science, Vol. 341)*, Sam Staton (Ed.). Elsevier, 261–276. <https://doi.org/10.1016/j.entcs.2018.11.013>
- Anders Kock. 1970. Monads on symmetric monoidal closed categories. *Archiv der Mathematik* 21, 1 (1970), 1–10.
- Anders Kock. 1971. Closed categories generated by commutative monads. *Journal of the Australian Mathematical Society* 12, 4 (1971), 405–424.
- Anders Kock. 1972. Strong functors and monoidal monads. *Archiv der Mathematik* 23, 1 (1972), 113–120.
- Rory B. B. Lucyshyn-Wright. 2018. Commutants for Enriched Algebraic Theories and Monads. *Appl. Categorical Struct.* 26, 3 (2018), 559–596. <https://doi.org/10.1007/s10485-017-9503-1>
- Saunders Mac Lane. 1998. *Categories for the Working Mathematician (2nd ed.)*. Springer.
- Eugenio Moggi. 1989. Computational Lambda-Calculus and Monads. In *Proceedings of the Fourth Annual Symposium on Logic in Computer Science (LICS '89), Pacific Grove, California, USA, June 5-8, 1989*. IEEE Computer Society, 14–23. <https://doi.org/10.1109/LICS.1989.39155>
- Eugenio Moggi. 1991. Notions of Computation and Monads. *Inf. Comput.* 93, 1 (1991), 55–92. [https://doi.org/10.1016/0890-5401\(91\)90052-4](https://doi.org/10.1016/0890-5401(91)90052-4)
- John Power and Edmund P. Robinson. 1997. Premonoidal Categories and Notions of Computation. *Math. Struct. Comput. Sci.* 7 (1997), 453–468.
- Sam Staton and Paul Blain Levy. 2013. Universal properties of impure programming languages. In *The 40th Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL '13, Rome, Italy - January 23 - 25, 2013*, Roberto Giacobazzi and Radhia Cousot (Eds.). ACM, 179–192. <https://doi.org/10.1145/2429069.2429091>
- Ross Street and Robert Walters. 1978. Yoneda structures on 2-categories. *Journal of Algebra* 50, 2 (1978), 350–379.
- Matthijs Vákár, Ohad Kammar, and Sam Staton. 2019. A domain theory for statistical probabilistic programming. *Proc. ACM Program. Lang.* 3, POPL (2019), 36:1–36:29. <https://doi.org/10.1145/3290349>

A PROOFS

PROOF OF PROPOSITION 4.3. Let (X, f) be a central cone and let $f' : X' \rightarrow \mathcal{T}Y'$ be a morphism. The following diagram:

$$\begin{array}{ccccccc}
 X \otimes X' & \xrightarrow{f \otimes X'} & \mathcal{T}Y \otimes X' & \xrightarrow{\tau'_{Y, X'}} & \mathcal{T}(Y \otimes X') & \xrightarrow{\mathcal{T}(Y \otimes f')} & \mathcal{T}(Y \otimes \mathcal{T}Y') \\
 \downarrow X \otimes f' & \text{(1)} & \downarrow \mathcal{T}Y \otimes f' & & \text{(2)} & \nearrow \tau'_{Y, \mathcal{T}Y'} & \downarrow \mathcal{T}\tau_{Y, Y'} \\
 X \otimes \mathcal{T}Y' & \xrightarrow{f \otimes \mathcal{T}Y'} & \mathcal{T}Y \otimes \mathcal{T}Y' & & & & \mathcal{T}^2(Y \otimes Y') \\
 \downarrow \tau_{X, Y'} & \text{(3)} & \downarrow \tau_{\mathcal{T}Y, Y'} & & \text{(4)} & & \downarrow \mu_{Y \otimes Y'} \\
 \mathcal{T}(X \otimes Y') & \xrightarrow{\mathcal{T}(f \otimes Y')} & \mathcal{T}(\mathcal{T}Y \otimes Y') & \xrightarrow{\mathcal{T}\tau'_{Y, Y'}} & \mathcal{T}^2(Y \otimes Y') & \xrightarrow{\mu_{Y \otimes Y'}} & \mathcal{T}(Y \otimes Y')
 \end{array}$$

commutes because: (1) \mathcal{C} is monoidal; (2) τ' is natural; (3) τ is natural; and (4) the pair (X, f) is a central cone. Therefore, the morphism f is central in the premonoidal sense.

For the other direction, if f is central in $\mathcal{C}_{\mathcal{T}}$, the following diagram:

$$\begin{array}{ccccccc}
 Z \otimes \mathcal{T}Y & \xrightarrow{f \otimes \mathcal{T}Y} & \mathcal{T}X \otimes \mathcal{T}Y & \xrightarrow{\tau'_{X, \mathcal{T}Y}} & \mathcal{T}(X \otimes \mathcal{T}Y) & & \\
 \downarrow f \otimes \mathcal{T}Y & \searrow f \otimes \mathcal{T}Y & \downarrow \tau_{X, \mathcal{T}Y} & & \downarrow \mathcal{T}\tau_{X, Y} & & \\
 \mathcal{T}X \otimes \mathcal{T}Y & \xleftarrow{f \otimes \mathcal{T}Y} & Z \otimes \mathcal{T}Y & & \mathcal{T}^2(X \otimes Y) & & \\
 \downarrow \tau_{\mathcal{T}X, Y} & \text{(1)} & \downarrow \tau_{Z, Y} & & \text{(2)} & & \downarrow \mu_{X \otimes Y} \\
 \mathcal{T}(\mathcal{T}X \otimes Y) & \xrightarrow{\mathcal{T}\tau'_{X, Y}} & \mathcal{T}^2(X \otimes Y) & \xrightarrow{\mu_{X \otimes Y}} & \mathcal{T}(X \otimes Y) & &
 \end{array}$$

commutes because: (1) τ is natural; (2) f is a central morphism; all remaining subdiagrams commute trivially. This shows the pair (X, f) is a central cone. \square

unique map such that the following diagram commutes:

$$\begin{array}{ccc}
 \mathcal{Z}X & \xrightarrow{\mathcal{Z}f} & \mathcal{Z}Y \\
 \iota_X \downarrow & & \downarrow \iota_Y \\
 \mathcal{T}X & \xrightarrow{\mathcal{T}f} & \mathcal{T}Y
 \end{array}$$

It follows directly that \mathcal{Z} maps the identity to the identity, and that ι is natural. \mathcal{Z} also preserves composition, which follows by the commutative diagram below.

$$\begin{array}{ccc}
 & \iota_A & \\
 & \mathcal{Z}A \longrightarrow \mathcal{T}A & \\
 \mathcal{Z}g \downarrow & & \downarrow \mathcal{T}g \\
 \mathcal{Z}(f \circ g) & \mathcal{Z}B \xrightarrow{\iota_B} \mathcal{T}B & \mathcal{T}(f \circ g) \\
 \mathcal{Z}f \downarrow & & \downarrow \mathcal{T}f \\
 & \mathcal{Z}C \xrightarrow{\iota_C} \mathcal{T}C &
 \end{array}$$

This proves that \mathcal{Z} is a functor. Next, we describe its monad structure and after that we show that it is commutative.

The monadic unit η_X is central, because it is the identity morphism in $\mathcal{Z}(\mathcal{C}\mathcal{T})$, thus it factors through ι_X to define $\eta_X^{\mathcal{Z}}$.

$$\begin{array}{ccc}
 X & \xrightarrow{\eta_X^{\mathcal{Z}}} & \mathcal{Z}X \\
 \eta_X \searrow & & \swarrow \iota_X \\
 & \mathcal{T}X &
 \end{array}$$

Next, observe that, by definition, $\mu_X \circ \mathcal{T}\iota_X \circ \iota_{\mathcal{Z}X} = \iota_X \circ \iota_{\mathcal{Z}X}$, where $(- \circ -)$ indicates Kleisli composition. Since ι is central and Kleisli composition preserves central morphisms, it follows that this morphism factors through ι_X and we use this to define $\mu_X^{\mathcal{Z}}$ as in the diagram below.

$$\begin{array}{ccc}
 \mathcal{Z}^2X & \xrightarrow{\mu_X^{\mathcal{Z}}} & \mathcal{Z}X \\
 \iota_{\mathcal{Z}X} \downarrow & & \downarrow \iota_X \\
 \mathcal{T}\mathcal{Z}X & \xrightarrow{\mathcal{T}\iota_X} \mathcal{T}^2X \xrightarrow{\mu_X} & \mathcal{T}X
 \end{array}$$

Again, by definition, $\tau_{A,B} \circ (A \otimes \iota_B) = A \otimes_r \iota_B$. Central morphisms are preserved by the premonoidal products (as we noted in Section 2) and therefore, this morphism factors through $\iota_{A \otimes B}$ which we

use to define $\tau_{A,B}^{\mathcal{Z}}$ as in the diagram below.

$$\begin{array}{ccc}
 A \otimes \mathcal{Z}B & \xrightarrow{\tau_{A,B}^{\mathcal{Z}}} & \mathcal{Z}(A \otimes B) \\
 A \otimes \iota_B \downarrow & & \downarrow \iota_{A \otimes B} \\
 A \otimes \mathcal{T}B & \xrightarrow{\tau_{A,B}} & \mathcal{T}(A \otimes B)
 \end{array}$$

Note that the last three diagrams are exactly those of a morphism of strong monads (see Definition 2.4).

Using the fact that ι is monic (see Lemma 4.6) we show that the following commutative diagram shows that $\eta^{\mathcal{Z}}$ is natural.

$$\begin{array}{ccc}
 X & \xrightarrow{\eta_X^{\mathcal{Z}}} & \mathcal{Z}X \\
 \downarrow f & \searrow \eta_X & \swarrow \iota_X \\
 & \mathcal{T}X & \\
 & \searrow \mathcal{T}f & \downarrow \mathcal{Z}f \\
 & & \mathcal{Z}Y \\
 & & \downarrow \iota_Y \\
 Y & \xrightarrow{\eta_Y^{\mathcal{Z}}} & \mathcal{Z}Y \xrightarrow{\iota_Y} \mathcal{T}Y
 \end{array}$$

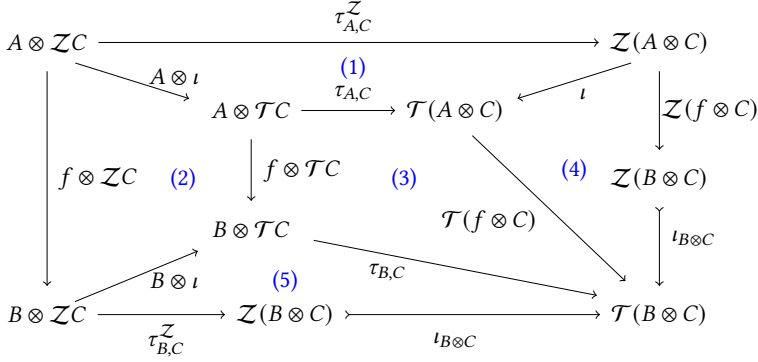
(1) (2) (3) (4)

(1) definition of $\eta^{\mathcal{Z}}$, (2) ι is natural, (3) η is natural and (4) definition of $\eta^{\mathcal{Z}}$. Thus we have proven that for any $f : X \rightarrow Y$, $\iota_Y \circ \mathcal{Z}f \circ \eta_X^{\mathcal{Z}} = \iota_Y \circ \eta_Y^{\mathcal{Z}} \circ f$. Besides, ι is monic, thus $\mathcal{Z}f \circ \eta_X^{\mathcal{Z}} = \eta_Y^{\mathcal{Z}} \circ f$ which proves that $\eta^{\mathcal{Z}}$ is natural. We will prove all the remaining diagrams will the same reasoning. The following commutative diagram shows that $\mu^{\mathcal{Z}}$ is natural.

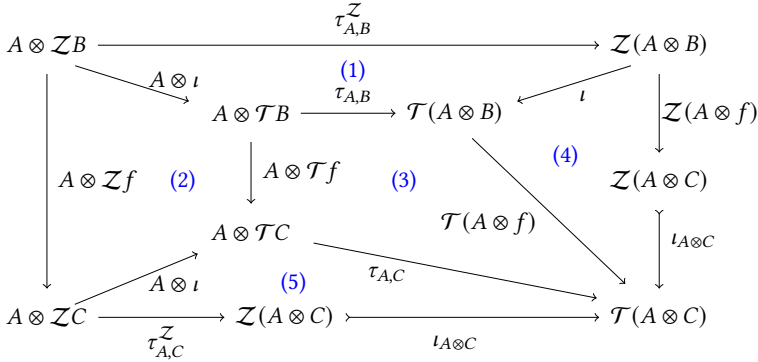
$$\begin{array}{ccc}
 \mathcal{Z}^2X & \xrightarrow{\mu_X^{\mathcal{Z}}} & \mathcal{Z}X \\
 \downarrow \mathcal{Z}^2f & \searrow u & \swarrow \iota \\
 & \mathcal{T}^2X & \xrightarrow{\mu_X} \mathcal{T}X \\
 & \searrow \mathcal{T}^2f & \downarrow \mathcal{Z}f \\
 & & \mathcal{Z}Y \\
 & & \downarrow \iota_Y \\
 \mathcal{Z}^2Y & \xrightarrow{\mu_Y^{\mathcal{Z}}} & \mathcal{Z}Y \xrightarrow{\iota_Y} \mathcal{T}Y
 \end{array}$$

(1) (2) (3) (4) (5)

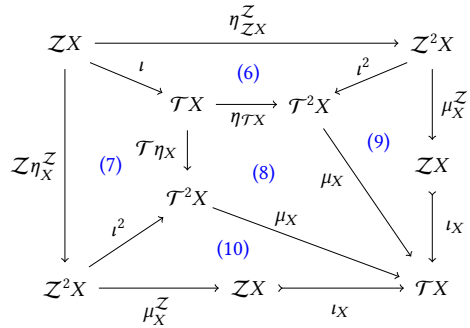
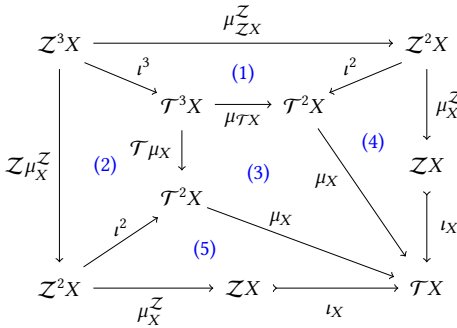
(1) definition of $\mu^{\mathcal{Z}}$, (2) ι is natural, (3) μ is natural, (4) ι is natural and (5) definition of $\mu^{\mathcal{Z}}$.
 The following commutative diagrams shows that $\tau^{\mathcal{Z}}$ is natural.



(1) definition of $\tau^{\mathcal{Z}}$, (2) ι is natural, (3) τ is natural, (4) ι is natural and (5) definition of $\tau^{\mathcal{Z}}$.



(1) definition of $\tau^{\mathcal{Z}}$, (2) ι is natural, (3) τ is natural, (4) ι is natural and (5) definition of $\tau^{\mathcal{Z}}$.
 The following commutative diagrams prove that \mathcal{Z} is a monad.



(1) and (2) involve the definition of $\mu^{\mathcal{Z}}$ and the naturality of ι and $\mu^{\mathcal{Z}}$, (3) is Def. 2.1, (4) definition of $\mu^{\mathcal{Z}}$ and (5) also. (6) and (7) involve the definition of $\eta^{\mathcal{Z}}$ and the naturality of ι and $\eta^{\mathcal{Z}}$, (8) is Def. 2.1, (9) definition of $\mu^{\mathcal{Z}}$ and (10) also.

of the diagram below amounts to check the commutativity of the following diagrams for each components:

$$\begin{array}{ccccc}
 A \otimes X & \xrightarrow{\mathcal{J}(\epsilon_A \otimes X)} & c \otimes X & \xrightarrow{h \otimes_l X} & c' \otimes X \\
 & & \downarrow c \otimes_r f & & \downarrow c' \otimes_r f \\
 & & c \otimes Y & \xrightarrow{h \otimes_l Y} & c' \otimes Y
 \end{array}$$

Since, the composition $f \circ \mathcal{J}(g)$ in $\mathbf{C}_{\mathcal{T}}$ corresponds to $f \circ g$ in \mathbf{C} , this is equivalent to the following diagram in \mathbf{C} :

$$\begin{array}{ccccccc}
 & & c \otimes X & & & & \\
 & & \epsilon_A \otimes X \nearrow & (1) & \searrow h \otimes X & & \\
 c \otimes X & \xleftarrow{\epsilon_A \otimes X} & A \otimes X & \xrightarrow{\epsilon'_A \otimes X} & \mathcal{T}c' \otimes X & \xrightarrow{\tau'_{c',X}} & \mathcal{T}(c' \otimes X) \xrightarrow{\mathcal{T}(c' \otimes f)} \mathcal{T}(c' \otimes \mathcal{T}Y) \\
 c \otimes f \downarrow & (2) & \downarrow A \otimes f & & & & \downarrow \mathcal{T}\tau_{c',Y} \\
 c \otimes \mathcal{T}Y & \xleftarrow{\epsilon_A \otimes \mathcal{T}Y} & A \otimes \mathcal{T}Y & & (5) & & \mathcal{T}^2(c' \otimes Y) \\
 \tau_{c,Y} \downarrow & (3) & \downarrow \tau_{A,Y} & & & & \downarrow \mu_{c' \otimes Y} \\
 \mathcal{T}(c \otimes Y) & \xleftarrow{\mathcal{T}(\epsilon_A \otimes Y)} & \mathcal{T}(A \otimes Y) & \xrightarrow{\mathcal{T}(\epsilon'_A \otimes Y)} & \mathcal{T}(\mathcal{T}c' \otimes Y) & \xrightarrow{\mathcal{T}\tau'_{c',Y}} & \mathcal{T}^2(c' \otimes Y) \xrightarrow{\mu_{c' \otimes Y}} \mathcal{T}(c' \otimes Y) \\
 & & \searrow & (4) & \nearrow & & \\
 & & & \mathcal{T}(h \otimes Y) & & &
 \end{array}$$

Where: (1) is the definition of h , (2) is the exchange law, (3) is the naturality of the strength, (4) is again the definition of h together with functoriality of $\mathcal{T}(- \otimes Y)$, and (5) is the fact that ϵ'_A is by definition central.

We can then conclude that h is central and so that the co-restriction $\hat{\mathcal{J}}$ is co-continuous.

Then by the adjoint functor theorem for total categories [Street and Walters 1978], $\hat{\mathcal{J}}$ is a left adjoint, and by Theorem 4.10 it follows that the corresponding strong monad is centralisable. \square