

Electronic Notes in Theoretical Computer Science 264 (2010) 125–140

[www.elsevier.com/locate/entcs](http://www.elsevier.com/locate/entcs)

From Coalgebraic to Monoidal Traces

Bart Jacobs

*Institute for Computing and Information Sciences, Radboud University of Nijmegen*

*P.O. Box 9010, 6500 GL Nijmegen, The Netherlands Email:* [*bart@ cs. ru. nl*](mailto:bart@cs.ru.nl) *URL:* [*www. cs. ru. nl/ ~ bart*](http://www.cs.ru.nl/~bart)

Abstract

The main result of this paper shows how coalgebraic traces, in suitable Kleisli categories, give rise to traced monoidal structure in those Kleisli categories, with finite coproducts as monoidal structure. At the heart of the matter lie partially additive monads inducing partially additive structure in their Kleisli categories. By applying the standard “Int” construction one obtains compact closed categories for “bidirectional monadic computation”.

*Keywords:* Coalgebra, execution trace, monoidal trace, iteration, Kleisli category

# Introduction

The notion of trace occurs prominently in the (classical) categorical work on traced monoidal categories [[13](#_bookmark39)]. It generalises the trace operator in linear algebra and captures fixed points for operations with feedback. Recently, also a coalgebraic approach to traces emerged [[12](#_bookmark37)], where traces are maps in Kleisli categories in- duced by monads that capture the observable behaviour in for instance sequences of (monadic) computations. Such traces are often described by removing states from execution traces. Naturally one wonders if there is a connection between these monoidal and coalgebraic traces. This paper addresses this question and shows how coalgebraic traces give rise to monoidal traces. The word ‘trace’ thus different meanings in this context, but hopefully without generating too much confusion.

The way this result is obtained is via the work of Haghverdi [[9](#_bookmark34)], where it is shown that partially additive categories (see also [[5](#_bookmark30)]) are traced monoidal, via what is called the execution (or trace) formula. Thus the paper proceeds by proving that under certain assumptions on a monad *T* , firstly the Kleisli category of *T* is such a partially additive category, and secondly the execution formula coincides with the

٨ ENTCS Proceedings of *Coalgebraic Methods in Computer Science* (CMCS 2010).

1571-0661 © 2010 Elsevier B.V. Open access under [CC BY-NC-ND license.](http://creativecommons.org/licenses/by-nc-nd/3.0/)

doi:10.1016/j.entcs.2010.07.017

coalgebraic trace. The technical core of the paper involves the identification of the notion of a “partially additive monad”, see Definition [4.3](#_bookmark15), and the proof that the Kleisli categories of such monads are partially additive.

We describe the organisation of this paper and at the same time the flow of developments. The paper starts with an elementary initial algebra in Section [2](#_bookmark0) that gives rise to a final coalgebra in suitably order-enriched Kleisli categories in Section [3](#_bookmark3), and thus to coalgebraic trace semantics, following [[12](#_bookmark37)]. For this particular coalgebra it also yields an iteration operation as in [[8](#_bookmark32),[6](#_bookmark31)]. Section [4](#_bookmark11) then shows that what we call partially additive monads in such a setting additionally yields partially additive structure *M* on Kleisli homsets, as studied earlier in [[5](#_bookmark30)]. They enable us to obtain the main result in Section [5](#_bookmark19), namely that Kleisli categories of suitable monads, with finite coproducts, are traced monoidal, via [[9](#_bookmark34)]. The “Int” construction from [[13](#_bookmark39)] can then be applied and yields in Section [6](#_bookmark24) new categories *Bd*(*T* ) of “bidirectional monadic computations”, with connections to game semantics and quantum computation. This forms a topic of its own that will be further investigated elsewhere. Throughout the paper there is a series of running examples, consisting of powerset, lift, distribution and quantale monads. The latter eventually yields examples of strongly compact closed categories.

# A basic initial algebra

i*∈*I i

Assume C is a category with countable coproducts, written as *X* with co- projections *κ*i: *X*i *→* i*∈*I *X*i. In order to further fix the notation, we shall write []X:0 *→ X* or simply []: 0 *→ X* (without subscript) for the unique arrow (the empty cotuple) out of an initial object 0. The two coprojections for a binary coproduct

*Æ*

κ

are written as *X*

*−→ X* + *Y*

κ

*←− Y*

*r*

, with cotupling of *f* : *X → Z* and *g*: *Y → Z*

denoted by [*f, g*]: *X* + *Y → Z*. Hence on morphisms, *h* + *k* = [*κ*l *◦ h, κ*r *◦ k*].

This C with its finite coproducts (0*,* +) yields a symmetric monoidal category (SMC). In general, for an SMC (A*,I, ⊗*) we write the familiar isomorphisms as:

1. *X ⊗* (*Y ⊗ Z*) *α*  (*X* *⊗ Y* ) *⊗ Z X ⊗ I*  *ρ*  *X* *X ⊗ Y*  *γ*  *Y* *⊗ X*

*∼*= *∼*= *∼*=

A copower *I · X* = i*∈*I *X* comes with coprojections *κ*i: *X → I · X* and cotupling [*f*i]i*∈*I: *I · X → Y* for an *I*-indexed collection of maps *f*i: *X → Y* .

**Proposition 2.1** *Let* C *have countable coproducts, as above. For a ﬁxed object Y ∈* C*, the functor Y* + (*−*): C *→* C *has the copower* N *· Y* = n*∈*N *Y as initial algebra, with structure map:*

*defn*

*ξ* = *κ ,* [*κ*

0

*Y* + N *· Y ∼*=

n+1]n*∈*N

N *· Y*

**Proof** For an arbitrary algebra [*a, b*]: *Y* + *X → X* we define *f* : N *· Y → X* as

*f* = [*b*n *◦ a*]n*∈*N. It forms the unique algebra homomorphism from *ξ* to [*a, b*].

The copower object N *· Y* may be understood in the standard way (see [[16](#_bookmark40)]) as

the colimit of repeated application of the functor *Y* +(*−*) to the initial object 0 *∈* C, as in:

[] *Y* + [] *Y* + (*Y* + [])

0 1 *·* *Y*  2 *·* *Y*

*· Y*  *· ·* *·*

3

We write 0 *· Y* = 0 and (*n* + 1) *· Y* = *Y* + *n · Y* . The resulting colimit cone

*λ*n: *n · Y →* N *· Y* is then defined as:

1. *λ*0 = []: 0 *−→* N *· Y* and *λ*n+1 = [*κ*n*, λ*n]: *Y* + *n · Y −→* N *· Y.*

The “twist” in this definition of *λ*n is needed to ensure that the “oldest” element in *n · Y* is put at the first position in N *· Y* . Indeed, in this way we get *λ*n+1 *◦ κ*r = *λ*n for the chain maps *κ*r: *Y*n *→ Y*n+1.

# A final coalgebra in a Kleisli category: trace seman- tics

We now assume that our category C (with coproducts) carries a monad *T* : C *→* C, with unit *η* and multiplication *μ*. We shall write *Kl*(*T* ) for the resulting Kleisli category, with forgetful functor *Kl*(*T* ) *→* C and left adjoint *J*: C *→ Kl*(*T* ). Trivially, *Kl*(*T* ) inherits coproducts from C. They behave like in C on objects, but have slightly different coprojections and coproducts of maps. In order to disambiguate them we shall write a dot for operations in a Kleisli category, as in:

*g ⊙ f* = *μ ◦ Tg ◦ f*

*κ*˙ l = *J*(*κ*l) = *η ◦ κ*l

*h*+˙ *k* = [*T* (*κ*l) *◦ h, T* (*κ*r) *◦ k*]*,* so that *J*(*a* + *b*) = *J*(*a*) +˙

*J*(*b*)*.*

This dot-notation is meant to prevent confusion. We shall use it with prudence and shall write for instance identity maps in Kleisli categories simply as idX and not as id˙X = *η*X. The (obvious) identities *g ⊙ J*(*f* ) = *g ◦ f* and *J*(*g*) *⊙ f* = *T* (*g*) *◦ f* are often used.

For an object *Y ∈* C we thus also get a functor *Y* + (*−*): *Kl*(*T* ) *→ Kl*(*T* ). Its initial algebra is the copower N *· Y* , by Proposition [2.1](#_bookmark1), but in *Kl*(*T* ). Its final coalgebra will be of more interest here.

In [[12](#_bookmark37)] a general framework is developed for generic trace semantics, which works for coalgebras of the form *X → TFX*, where *T* is a monad and *F* an endofunctor. The main result in [[12](#_bookmark37)] says that, under suitable order-theoretic assumptions, the initial algebra in C yields a final coalgebra in *Kl*(*T* ). Here we shall only be interested in the special case where the functor *F* is of the form *Y* + (*−*).

**Proposition 3.1 (From [**[**12**](#_bookmark37)**])** *Let T be a monad on a category* C *with coproducts. Assume that the Kleisli category Kl*(*T* ) *is dcpo-enriched, that (Kleisli) homsets have bottom elements ⊥ which are left strict (i.e. satisfy ⊥ ⊙ f* = *⊥) and that cotupling is monotone (i.e.* [*−, −*] *preserves the order in both coordinates).*

*The initial algebra ξ*: *Y* +N *· Y*

*—*=*→* N *· Y in* C *from Proposition* [*2.1*](#_bookmark1) *then yields a*

*ﬁnal coalgebra J*(*ξ−*1): N*·Y —*=*→ T*

(*Y* +N*·Y* ) *of the functor Y* +(*−*): *Kl*(*T* ) *→ Kl*(*T* )*.*

*Concretely, this means that for every coalgebra c*: *X → T* (*Y* + *X*) *there is a unique map* tr(*c*): *X → T* (N *· Y* ) *forming a unique coalgebra homomorphism in the Kleisli category Kl*(*T* ) *as in:*

*id* +˙ tr(*c*)

*Y* +,*X*,\_ \_ \_ \_ \_ \_ *Y* + N,*·*,*Y*

*c*

tr(*c*)

*∼*= *J*(*ξ−*1)

*X* \_ \_ \_ \_ \_ \_ \_ \_ N *· Y*

Intuitively, this trace map tr(*c*) sends an element *x ∈ X* to the “set” of those (*n, y*) *∈* N *· Y* for which *c* reaches *y ∈ Y* from *x* in *n* cycles through *X*, see the examples below.

We shall write *c*# = *∇ ⊙* tr(*c*): *X → Y* in *Kl*(*T* ) for the “iterate” of *c*, like in [[8](#_bookmark32),[5](#_bookmark30)] [1](#_bookmark9) , where *∇* = [id]n*∈*N: N *· Y → Y* is the codiagonal in *Kl*(*T* ). It yields an operator between Kleisli homsets of the form:

(*−*)#

*Kl*(*T* )(*X, Y* + *X*) *Kl*(*T* )(*X, Y* )

Clearly, such an iterate *c*# does not keep track of the number of rounds that are made to reach a result in *Y* —like tr(*c*) does.

Here we omit the proof and refer to [[12](#_bookmark37)] for details but we shall explicitly describe the definition of the trace map tr(*c*) so that we can use it later on. It uses the fact that the initial object 0 *∈* C is final in Kleisli categories as in the proposition, with

*⊥*: *X →* 0 in *Kl*(*T* ) as unique map (see also Lemma [4.1](#_bookmark12) (1) below). This allows us to define a sequence of maps *c*n: *X → n · Y* in *Kl*(*T* ) as:



*c*0 = *⊥* : *X −→* 0 = 0 *· Y*

*c*n+1 = (id +˙

*c*n) *⊙ c* : *X −→ Y* + *X −→ Y* + *n · Y* = (*n* + 1) *· Y*

Then we can define the trace map as join:

1. ​

tr(*c*) = n*∈*N

*J*(*λ*n) *⊙ c*n

in the Kleisli homset of maps *X →* N *· Y* , with *λ*n as defined in ([2](#_bookmark2)).

**Example 3.2** We shall consider what the above result amounts to for our four main examples for the monad *T* , namely *P, D, L* and *Q*(*−*) on **Sets**.

1. The Kleisli category *Kl*(*P*) of the powerset monad *P*: **Sets** *→* **Sets** is the category of sets with relations as arrows between them. Homsets are ordered by pointwise inclusion, and form complete lattices. Commutation of diagram ([3](#_bookmark5)) means that for a coalgebra *c*: *X → P*(*Y* + *X*) the resulting trace map tr(*c*): *X → P*(N *· Y* ) satisfies:

(*n, y*) *∈* tr(*c*)(*x*0) *⇔ ∃x*1*,..., x*n *∈ X. x*1 *∈ c*(*x*0) *∧ ... ∧ x*n*−*1 *∈ c*(*x*n) *∧ y ∈ c*(*x*n)

*⇔ ∃x*1*,..., x*n *∈ X.* i<n *x*i+1 *∈ c*(*x*i) *∧ y ∈ c*(*x*n)

1 In [[8](#_bookmark32),[5](#_bookmark30)] the notation c*†* is used, instead of c#, but we prefer to reserve the dagger *†* for involutions, see Lemma [5.4](#_bookmark23).

where we have left out the coprojections *κ*l*, κ*r for simplicity.

1. For the lift monad *L* = 1 + (*−*) we write *⊥∈* 1+ *X* for the bottom element

*⊥∈* 1 and up(*x*) *∈* 1+ *X* for an element *x ∈ X*. These sets 1 + *X* are “flat” dcpos. For *c*: *X →* 1+ (*Y* + *X*) we then get a trace map tr(*c*): *X →* 1+ N *· Y* with:

tr(*c*)(*x*0) = up(*n, y*) *⇔ ∃x*1*,..., x*n *∈ X.* i<n *c*(*x*i) = up(*x*i+1) *∧ c*(*x*n) = up(*y*)

1. We shall write *D* for the (sub)distribution monad on **Sets** given by:

*D*(*X*) = *{ϕ*: *X →* [0*,* 1] *|* x*∈*X *ϕ*(*x*) *≤* 1*}.*

Σ

Notice that we do not require that such *ϕ ∈ D*(*X*) have finite support (*i.e.* have finitely many elements *x ∈ X* that are not mapped to 0). The sets *D*(*X*) are dcpos with pointwise order and bottom element *⊥* = *λx.* 0. The Kleisli maps *X → D*(*Y* ) can then also be ordered, pointwise.

For a coalgebra *c*: *X → D*(*Y* + *X*) we obtain a trace map tr(*c*): *X → D*(N *· Y* ) as in diagram ([3](#_bookmark5)), given explicitly by the following probability formula.

tr(*c*)(*x*0)(*n, y*) = Σ

= Σ

x1,...,x*n∈*X

*c*(*x*0)(*x*1) *· ... · c*(*x*n*−*1)(*x*n) *· c*(*x*n)(*y*)

*c*(*x*i)(*x*i+1) *· c*(*x*n)(*y*)

x1,...,x*n∈*X i<n

1. Let *Q* be a quantale, *i.e.* a complete lattice with a monoid structure (1*, ·*) where multiplication *·* preserves suprema in both arguments (see [[14](#_bookmark41)]). The mapping *X '→ Q*X is then a monad on **Sets** with unit and multiplication given by:

*η*  X *Q*(Q*X* ) *μ*  X

*X*

*x*  *λx* *'.*

*Q*

1 if *x'* = *x*

Φ *λx* *.*

*Q*

Φ(*ϕ*) *· ϕ*(*x*)

*⊥* otherwise

ϕ*∈*Q*X*

A function *f* : *X → Y* yields *Q*f : *Q*X *→ Q*Y by *ϕ '→ λy.* x*∈*f *−*1(y) *ϕ*(*x*). The powerset monad *P* from (1) is a special case for *Q* = 2.

For a coalgebra *c*: *X → Q*Y +X diagram ([3](#_bookmark5)) now yields a trace map tr(*c*): *X →*

*Q*N*·*Y that formally resembles the previous one:

tr(*c*)(*x*0)(*n, x*n+1) = *c*(*x*i)(*x*i+1)

x1,...,x*n∈*X i*≤*n

We collect some basic results about coalgebraic traces tr(*c*) and iterates *c*#.

**Lemma 3.3** *In the situation of the previous proposition:*

1. *Uniformity: if f is a homomorphism of coalgebras c → d in Kl*(*T* )*,*

tr(*c*) = tr(*d*) *◦ f and so c*# = *d*# *◦ f.*

1. *Naturality in Y : for g*: *Y → T* (*V* )*,*

tr((*g* +˙

*id*) *⊙ c*) = N *· g ⊙* tr(*c*) *and* ((*g* +˙

*id*) *⊙ c*)# = *g ⊙ c*#*.*

1. *Dinaturality in X: for f* : *U → T* (*X*)*,*

tr(*c ⊙ f* ) = tr((*id* +˙

*f* ) *⊙ c*) *◦ f and* (*c ⊙ f* )# = ((*id* +˙

*f* ) *⊙ c*)# *⊙ f.*

**Proof** Everything follows from (the uniqueness part of) finality. For instance the second point involves the diagram:

*V* + *X*  id +˙ tr(*c*) *V* + N *· Y*  id +˙ N *· g*  *V* + N *· V*

,,

*g* +˙ id

,, ,,

*g* +˙ id

*Y* + *X*  id +˙ tr(*c*) *Y* + N *· Y ∼*=

,, ,,

*c ∼*=

tr(*c*) N *· g*

*X*  N *· Y* ˛ N *· V*

tr((*g* +˙

id) *⊙ c*)

The diagram on the right commutes by definition of N *· g*.

# Additive structure on Kleisli homsets

We start this section by some preparatory observations about the structure in- duced by order on Kleisli homsets, making coproducts behave a bit like products (*i.e.* biproducts). It will lead to a description of additive structure (certain sums) in such homsets, which we shall write with a separate symbol *M* in order to prevent confusion with the sum *f* + *g* = [*κ*l *◦ f, κ*r *◦ g*] induced by coproducts +. The main contribution of this section lies in the notion of partially additive monad, see Definition [4.2](#_bookmark13), and in the result that the Kleisli categories of such monads form partially additive categories.

The first point of the next lemma has already been used, but will be repeated here for completeness.

**Lemma 4.1** *Assume* C *is a category with countable coproducts. Let T* : C *→* C *be a monad whose Kleisli homsets Kl*(*T* )(*X, Y* ) = C(*X, T* (*Y* )) *are partially ordered.*

* 1. *If each Kleisli homset has a bottom element ⊥*: *X → T* (*Y* ) *which is left strict (i.e. satisﬁes ⊥ ⊙ f* = *⊥), then* 0 *is a ﬁnal object in Kl*(*T* )*. Since* 0 *is obviously initial in Kl*(*T* )*, it becomes a zero object (or “nullary” biproduct).*
  2. *If ⊥ is “bi-strict”, i.e. is preserved by both pre- and post-composition in Kl*(*T* )*, then there are natural “projection” maps p*j: i*∈*I *X*i *→ T* (*X*j) *satisfying:*

*p*j *⊙ κ*˙ j = *id and p*j *⊙ κ*˙ m = *⊥ for j /*= *m.*

*In the binary case we shall write p*l*, p*r*, just like for coprojections κ*l*, κ*r*.*

**Proof** (1) There is only *⊥*: *X →* 0 in *Kl*(*T* ) because each *f* : *X →* 0 satisfies:

*f* = *f ⊙* id0 = *f ⊙ ⊥* = *⊥*, by left strictness.

(2) One takes *p*j = [*p*i,j]i*∈*I: i*∈*I *X*i *→ T* (*X*j) where *p*j,j = *η*X*j* and *p*i,j = *⊥* for *i /*= *j*. Then clearly *p*j *⊙ κ*˙ j = *p*j *◦ κ*j = *p*j,j = *η*, which is the identity in *Kl*(*T* ), and *p*j *⊙ κ*˙ m = *⊥* for *j /*= *m*. Naturality follows from (right) strictness.

For the formulation of the following notion it is convenient to assume that our category C has set-indexed products. The definition can be given without such prod- ucts, using “jointly monic families”. But that only makes it harder to understand the matter.

**Definition 4.2** Assume projections *p*i as in the previous lemma, for a monad *T* on a category C with countable coproducts and products. By bc, for ‘bicartesian’, we denote the following map.

bc = *T*

( i*∈*I

*⟨ p*b *⟩*i*∈*I

*X*i) i

i*∈*I

*T* (*X*i

) where

b *μ ◦ T*

(*p*i)*.*

The monad *T* is called partially additive if these bc’s form cartesian natural trans- formations with monic components. This means that all naturality squares:

*p*

=

i

*T* ( i

*X* ) *T* ( i *f*i) *T* (

*Y*i)

J J

i

i

bc bc

i *T* (*Y*i)

J

i *T* (*X*i)

i *T* (*f*i) J

are pullbacks in C, for collections of maps *f*i: *X*i *→ Y*i in C.

The monad *T* may be called additive if these bc’s are isomorphisms. Such mon- ads are investigated further in [[7](#_bookmark33)]. The next definition of sums on Kleisli homsets is based on [[5](#_bookmark30)].

**Definition 4.3** Let *T* be a partially additive monad on C, as in the previous defi- nition. For countably many *f*i: *X → Y* in *Kl*(*T* ) write *M*i*∈*I *f*i = *∇*I *⊙ b*: *X → Y* in *Kl*(*T* ) if there is a “bound” map *b*: *X → T* (*I · Y* ) = *T* ( *Y* ) with *p ⊙ b* = *f* .

i*∈*I i i

This bound property can be expressed as: bc *◦ b* = *⟨f*i*⟩*i*∈*I: *X →* i*∈*I *T* (*Y* ) =

*T* (*Y* )I. By the mono requirement on bc there is at most one such bound *b*.

We may observe that certain joins always exist: for a map *f* : *X → T* (*Y* + *Z*),

one has *f* = (*κ*˙ l *⊙ p*l *⊙ f* ) *M* (*κ*˙ r *⊙ p*r *⊙ f* ), via the bound (*κ*˙ l +˙ *κ*˙ r) *⊙ f* : *X →*

*T* ((*Y* + *Z*)+ (*Y* + *Z*)).

Before further investigation of this sum *M* we check what it means in the exam- ples.

**Example 4.4** We shall consider the powerset monad as special case of the quantale monad *Q*(*−*), namely for *Q* = 2. For convenience, we consider the binary sum *M* only.

1. For the lift monad *L*, recall that Kleisli homsets are flat orders, in which very few joins (or sums) exist. The projections *Y*l + *Y*r *→* 1 + *Y*i are given by *p*i(*w*) = up(*y*) iff *w* = *κ*i(*y*), for *i ∈ {l, r}*. For *b*: *X →* 1+ (*Y* + *Y* ) one has:

(*p*i

*⊙ b*)(*x*) = up(*y*) if *b*(*x*) = up(*κ*i*y*)

*⊥* otherwise.

Hence *b* is completely determined by these *p*i *⊙ b*, so that projections are jointly monic—and bc from ([6](#_bookmark14)) is monic. The pullback property for bc is left to the reader.

Now if *f*i: *X →* 1+ *Y* are given, and we have a bound *b*: *X →* 1+ (*Y* + *Y* ) with

*p*i *⊙ b* = *f*i, then we know:

* if *f*l(*x*) = up(*y*), then (*p*l *⊙ b*)(*x*) = up(*y*) so that *b*(*x*) = up(*κ*l*y*) and thus (*p*r *⊙ b*)(*x*) = *⊥*, so that *f*r(*x*) = *⊥*.
* if *f*r(*x*) = up(*y*), then similarly *f*l(*x*) = *⊥*.

The existence of this bound *b* thus guarantees that both *f*l(*x*) */*= *⊥* and *f*r(*x*) */*= *⊥* does not happen. Hence their join exists, namely the non-bottom value, if any. This value is given by *∇ ⊙ b*.

1. The Kleisli category *Kl*(*Ð*) of the subdistribution monad *Ð* inherits its point- wise order from the unit interval [0*,* 1]. This interval has joins, but it turns out that *M* describes the partially defined + on [0*,* 1]. The projections *Y*l + *Y*r *→ Ð*(*Y*i) are given by *p*i(*w*)(*y*) = if *w* = *κ*i*y* then 1 else 0. Thus for *b*: *X → Ð*(*Y* + *Y* ) we have (*p*i *⊙ b*)(*x*)(*y*) = w*∈*У +У *p*i(*w*)(*y*) *· b*(*x*)(*w*) = *b*(*x*)(*κ*i*y*). And bc: *Ð*(*Y*l + *Y*r) *→ Ð*(*Y*l) *× Ð*(*Y*r) is given by bc(*ϕ*) = *⟨ϕ ◦ κ*l*,ϕ ◦ κ*r*⟩*. It is thus clearly monic.

Σ

For the pullback property for bc, assume a collection *f*i: *X*i *→ Y*i together with maps *⟨α*l*, α*r*⟩*: *A → Ð*(*X*l) *× Ð*(*X*r) and *β*: *A → Ð*(*Y*l + *Y*r) satisfying *Ð*(*f*i) *◦ α*i = *p*b *◦ β*. The only possible mediating map *γ*: *A → Ð*(*X*l + *X*r) is defined as *γ*(*a*)(*κ*l*x*) = *α*l(*a*)(*x*) and *γ*(*a*)(*κ*r*x*) = *α*r(*a*)(*x*). We have to check that *γ*(*a*) is a subdistribution. This follows from because *β*(*a*) is a subdistribution:

i

=

y*∈*У*Æ*

l

y*∈*У*r*

(*p*b *◦ β*)(*a*)(*y*)

= y*∈*У (*Ð*(*f*l) *◦ α*l)(*a*)(*y*)+ y*∈*У (*Ð*(*f*r) *◦ α*r)(*a*)(*y*)

= Σx*∈*X*Æ α*l(*a*)(*x*)+ Σx*∈*X*r α*r(*a*)(*x*)

r

Σ *Æ* Σ *r*

*Æ*

(*p*b *◦ β*)(*a*)(*y*)+ Σ

1 *≥* Σz *β*(*a*)(*z*) = Σy*∈*У*Æ β*(*a*)(*κ*l*y*)+ Σy*∈*У*r β*(*a*)(*κ*r*y*)

= Σy*∈*У*Æ* Σx*∈*ƒ *−*1(y) *α*l(*a*)(*x*)+ Σy*∈*У*r* Σx*∈*ƒ *−*1(y) *α*r(*a*)(*x*)

=

w*∈*X*Æ* +X*r γ*(*a*)(*w*)*.*

*r*

Further, if *f*i: *X → Ð*(*Y* ) are given with *f*i = *p*i *⊙ b*, then:

(*f*l *M f*r)(*x*)(*y*) = (*∇ ⊙ b*)(*x*)(*y*) = w*∈*У +У *∇*(*w*)(*y*) *· b*(*x*)(*w*)

Σ

= *b*(*x*)(*κ*l*y*)+ *b*(*x*)(*κ*r*y*)

= *f*l(*x*)(*y*)+ *f*r(*x*)(*y*)*.*

1. For the quantale monad *Q*(*−*) we have projections *Y*l + *Y*r *→ Q*У*i* given by *p*i(*w*)(*y*) = if *w* = *κ*i*y* then 1 else *⊥*, so that for *b*: *X → Q*У +У we get (*p*i *⊙ b*)(*x*)(*y*) =  w*∈*У +У *p*i(*w*)(*y*) *· b*(*x*)(*w*) = *b*(*x*)(*κ*i*y*). The map bc is in this case an

isomorphism *Q*У*Æ* +У*r ∼*=

У*Æ × Q*У*r* , so that *Q*(*−*) is an additive monad. And if the

*—→ Q*

*f*i have a bound, then their sum is given by union: (*f*l *M f*r)(*x*)(*y*) = *f*l(*x*)(*y*) *V*

*f*r(*x*)(*y*).

These examples illustrate that the sum operation *M* is determined by (Kleisli) composition, and hence ultimately by the monad involved.

We continue with some basic properties of *M*.

**Lemma 4.5** *In the situation of the previous deﬁnition, one has:*

1. *M is preserved by both pre- and post-composition;*
2. *The sum of the singleton family {f} if f itself; the sum over the empty family is ⊥;*
3. *If cotupling* [*—, —*] *is monotone, then f*j *≤ M*i*∈*I *f*i*;*
4. *Assume the Kleisli category is* ***Dcpo****-enriched. Let I be a countable set such that M*i*∈*J *f*i *exists for each ﬁnite subset J ⊆ I. Then M*i*∈*I *f*i *exists.*

**Proof** (1) Suppose *M*i*f*i exists for *f*i: *X → T* (*Y* ), say with bound *b*: *X → T* (*I · Y* ). For *g*: *U → T* (*X*) the composite *b ⊙ g*: *U → T* (*I · Y* ) is obviously a bound for *f*i *⊙ g* and yields *M*i(*f*i *⊙ g*) = *∇ ⊙ b ⊙ g* = (*M*i*f*i) *⊙ g*.

Similarly, for *h*: *Y → T* (*U* ) the map *I · h ⊙ b* is a bound for *h ⊙ f*i, by naturality of projections, so that *M*i(*h ⊙ f*i) = *∇ ⊙ I · h ⊙ b* = *h ⊙ ∇ ⊙ b* = *h ⊙* (*M*i*f*i).

* 1. The map *f* is a bound for *{f}* and *⊥* is a bound for the empty family.
  2. If cotupling is monotone we get *p*i *≤∇* and thus for a bound *b*,

*f*i = *p*i *⊙ b ≤∇ ⊙ b* = *M*i*f*i*.*

* 1. Assume for convenience that our index set is N. Let *f*n: *X → T* (*Y* ), for *n ∈* N, be a collection such that the sum *M* exists for each finite subset. There are sums *f*0 *M f*1 *M· · ·M f*n*−*1, say via bound *b*n: *X → T* (*n · Y* ). It is not hard to see that the collection *κ*˙ i *⊙ f*i: *X → T* (N *· Y* ), for *i < n*, also has a bound, namely

*'* = (*κ*˙ 0 +˙ *···* +˙ *κ*˙ n*−*1) *⊙ b*n: *X → T* (*n ·* N *· Y* ). We then define

*b*

n

*g*n = *∇ ⊙ b'* = (*κ*˙ 0 *⊙ f*0) *M ··· M* (*κ*˙ n*−*1 *⊙ f*n*−*1) : *X —→* N *· Y.*

n

This yields a monotone collection *g*n *≤ g*n+1 by the previous point. Hence we get a map *f* = n *g*n: *X →* N *· Y* as directed join, which is the intended sum.

One further property of *M* is required, which is sometimes called “partition associativity”. It is non-trivial and depends on the pullback requirement from Def- inition [4.2](#_bookmark13).

**Lemma 4.6** *If a (countable) collection I can be written as disjoint union I* =

k*∈*K *I*k*, then M*i*∈*I *f*i *exists if and only each sum f*k = *M*i*∈*I*k f*i *exists and M*i*∈*I *f*i =

*M*k*∈*K *f*k*.*

*As a result, M is commutative and associative.*

**Proof** If *I* = k*∈*K *I*k is a disjoint union, then *I · Y ~*= k*∈*K *I*k *· Y* . Hence it is more convenient to consider a collection of maps *f*k,i: *X → Y* for *k ∈ K* and *i ∈ I*k. In one direction, suppose *b*: *X →* k*∈*K *I*k *· Y* is bound for the collection (*f*k,i),

so that *f*k,i = *p*i *⊙ p*k *⊙ b*. Write *b*k = *p*k *⊙ b*: *X → I*k *· Y* . It forms a bound for

the collection (*f*k,i)i*∈*I*k* , since *p*i *⊙ b*k = *p*i *⊙ p*k *⊙ b* = *f*i, for each *i ∈ I*k. The sums *f*k = *M*i*∈*I*k f*i = *∇*I*k ⊙ b*k have a bound *a* = ( k*∈*K *∇*I*k* ) *⊙ b*: *X → K · Y* , since for each *k ∈ K*,

*p*k *⊙ a* = *p*k *⊙* ( k*∈*K *∇*I*k* ) *⊙ b* = *∇*I*k ⊙ p*k *⊙ b* by naturality of projections

= *∇*I*k ⊙ b*k = *M*i*∈*I*k f*i = *f*k*.*

Hence *M*k*∈*K*f*k exists as *∇*K *⊙ a* = *∇*K *⊙* ( k*∈*K *∇*I*k* ) *⊙ b* = *∇*I *⊙ b* = *M*i*∈*I *f*i.

For the other direction assume that the sums *f*k = *M*i*∈*I*k f*k,i and *M*k*∈*K*f*k exist; we need to show that also *M*k*∈*I,i*∈*I *f*k,i exists—and is equal to *M*k*∈*K*f*k. So let

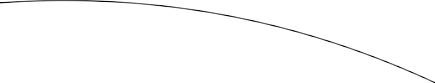
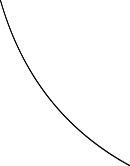
*b*k: *X → I*k *· Y* be a bound for the collection (*f*k,i)i*∈*I*k* and *a*: *X → K · Y* be a bound

for these *f*k = i*∈*I*k f*i = *∇*I*k b*k. We need a bound *c*: *X →* k*∈*K *I*k *· Y* , which

*⊙*

we obtain via the following naturality pullback, as required in Definition [4.3](#_bookmark15).

*X* ¸ ¸ ¸



*c*¸ ¸

¸z

*T* ( *I*

*a*

*· Y* )

*T* ( k *∇*I*k* ) z˛ )

*⟨b*k*⟩*k*∈*K

k*∈*K k

J

\_

bc

*T* ( *K · Y*

J

bc

k*∈*K *T* (*I*k *· Y* )

k*∈*K *T* (*Y* )

z J

k *T* (*∇*I*k* ) J

Hence the mediating map *c* is a bound for these *b*k and thus for the *f*k,i. The resulting sum is: *M*k*∈*K,i*∈*I*k f*k,i = *∇*K *⊙* k*∈*K *∇*I*k ⊙ c* = *∇*K *⊙ a* = *M*k*∈*K*f*k.

We are now ready to collect the requirements that we need in this paper.

**Requirements 4.7** *The category* C *is assumed to have countable coproducts and the monad T* : C *→* C *satisﬁes:*

1. *its Kleisli category Kl*(*T* ) *is* ***Dcpo****⊥-enriched, so that Kleisli homsets have (countable) directed joins and a bottom element, which are preserved by com- position;*
2. *this Kleisli category also has monotone cotupling;*
3. *the monad T is partially additive, as in Deﬁnition* [*4.3*](#_bookmark15)*.*

¿From Lemma [4.5](#_bookmark16) we may now conclude a basic result.

**Proposition 4.8** *Let category* C *with monad T satisfy Requirement* [*4.7*](#_bookmark17)*. The Kleisli category Kl*(*T* ) *is then partially additive. Further, it is additive (has all countable sums M) iff it has countable strict biproducts.*

For what it precisely means to be partially additive we refer to the literature [[5](#_bookmark30)]. Here we shall simply use that Kleisli homsets have certain sums *M*, with properties as described in Lemma [4.5](#_bookmark16). The projections *p*i make the Kleisli categories into what are called ‘unique decomposition categories’, see also [[10](#_bookmark35)]. The “further” part of the proposition is [[9](#_bookmark34), Theorem 3.0.17]. It applies to the Kleisli category of quantale monads.

# Kleisli categories are traced monoidal

Now that we have seen additive structure on Kleisli homsets we can conclude from [[9](#_bookmark34)] that we have traced monoidal structure in these Kleisli categories. But before we do so we return to Section [3](#_bookmark3) and re-describe the iterate *c*# of a coalgebra *c* in terms of the newly discovered sums. This will be used (in the proof of Theorem [5.2](#_bookmark21)) to show that the induced traced monoidal structure coincides with the coalgebraic trace.

**Lemma 5.1** *For* C*,T satisfying Requirements* [*4.7*](#_bookmark17) *the iterate c*# *of a coalgebra*

*c*: *X → T* (*Y* + *X*)*, from Proposition* [*3.1*](#_bookmark4)*, can be described as sum:*

*c*# = *c*l *⊙ M*n*∈*N *c*n = *c*l *⊙ c*٨*,*

r r

*where c*l = *p*l *⊙ c*: *X → T* (*Y* ) *and c*r = *p*r *⊙ c*: *X → T* (*X*)*, and h*٨ = *M*n*∈*N *h*n*.*

**Proof** Recall that the iterate is defined as *c*# = *∇ ⊙* tr(*c*): *X →* N *· Y → Y* . Hence it is a sum *M* by construction. So we only have to check that *p*i *⊙* tr(*c*) = *c*l *⊙ c*i , for *i ∈* N. But before we can do so we need a better handle on the projections *p*i: *n · Y → Y* in *Kl*(*T* ), for *i < n*. They are given inductively by:

r

1. *p*0 = [*η, ⊥*]: *Y* + *n · Y —→ T* (*Y* ) and *p*i+1 = [*⊥, p*i]: *Y* + *n · Y —→ T* (*Y* )

Then it is not hard to see that *p*i *◦ λ*n = *p*n*−*i*−*1: *n · Y → T* (*Y* ), for *i < n*, and

*p*i *◦ λ*n = *⊥*, for *i ≥ n*.

Next we use the explicit description of tr(*c*) as directed join from ([5](#_bookmark7)):

*p*i *⊙* tr(*c*) = *p*i *⊙*

n*∈*N

*J*(*λ*n) *⊙ c*n

= n*∈*N

=

n*∈*N

*p*i *⊙ J*(*λ*n) *⊙ c*n

*p*n*−*i*−*1 *⊙ c*n as we have just seen, where *i < n*

(=*∗*) *c*

l

r

*⊙ c*i *.*

The equation (*∗*) is obtained by induction on *n*, using ([4](#_bookmark6)).

The main result of this paper now shows how coalgebraic traces in Kleisli cat- egories yield a traced monoidal structure with respect to this monoidal structure (0*,* +). The result is actually a direct consequence of Proposition [4.8](#_bookmark18), using [[9](#_bookmark34), Theorem 3.1.4] (which dualises Hasegawa’s result that uniform fixed point opera- tors are uniform traces [[11](#_bookmark36)]). We should point out that the induced trace structure is of a very special kind, since the monoidal structure consists of coproducts, and

the obtained trace operators are uniform. Hence it can equivalently be presented in terms of iteration operators a` la Bloom-E´sik, *i.e.* as the duals of uniform fixed point operators, see [[6](#_bookmark31)]. So we are basically looking at an instance of Elgot iterative theories, see [[4](#_bookmark29)].

**Theorem 5.2** *For* C *and T satisfying Requirements* [*4.7*](#_bookmark17)*, the Kleisli category Kl*(*T* ) *with* (0*,* +) *is traced monoidal (see [*[*13*](#_bookmark39)*]). For a map f* : *X* + *U → Y* + *U in Kl*(*T* ) *we deﬁne* Tr(*f* ): *X → Y as the composite ∇ ⊙* tr(*f* ) *⊙ κ*l = *f* # *⊙ κ*l *at the bottom in:*

^ ^

*Y* + (*X*+ *U* ) *id* + tr(*f*^)

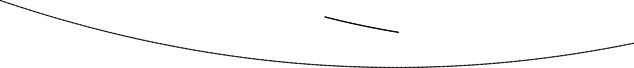
˙

,,

\_ \_ \_ \_ \_ \_ \_

*Y* + N *· Y*

,,

*f*^ = (*id*У

*κ*˙l

*X*

*X* + *U* \_ \_

+˙ *κ*˙ r) *⊙ f*

^

tr(*f* )

\_

\_

\_

\_

\_ N *· Y*

\_ *Y*

*~*= *J*

\_

\_

(*ξ−*1)

*∇*

Tr(*f* )

*f*^#

*This monoidal trace operation* Tr *then satisﬁes standard requirements from [*[*13*](#_bookmark39)*], and also the following special properties.*

***Identity*** Tr(*id*X+U ) = *id*X *;*

***Uniformity*** Tr(*f* ) = Tr(*g*)*, if* (*id* +˙

1. *⊙ f* = *g ⊙* (*id* +˙

*h*)*,*

*for f* : *X* + *U → Y* + *U, g*: *X* + *V → Y* + *V and h*: *U → V (see [*[*11*](#_bookmark36)*]).*

**Proof** The result follows from the properties of iteration (*—*)#, see [[9](#_bookmark34)] [2](#_bookmark22) , once we know that the definition of trace in [[9](#_bookmark34)] coincides with the coalgebraic one described in the theorem. This follows from Lemma [5.1](#_bookmark20) using a matrix description of *f* : *X* +*U → Y* + *U* . Write *f*ij = *π*j *⊙ f ⊙ κ*˙ i, for *i, j ∈ {l, r}*, so that:

⎛⎜ *X f*ll *T* ( *Y* ) *U f*lr *T* ( *Y* ) ⎞⎟

*f*

*f* =

⎝ *X f*

rl

*T* ( *U* ) *U*

rr

*T* ( *U* ) ⎠

We have to show that Tr(*f* ) = *f* # *⊙ κ*˙ l as defined above can be written as the (regular) expression *f*ll *M f*lr*f* ٨ *f*rl that is used in [[9](#_bookmark34)], and called the execution (or

^

rr

2 which, in dual form for products and a fixed point operator, should also be attributed to Masahito Hasegawa [[11](#_bookmark36)] and to Martin Hyland, see also [[15](#_bookmark42)].

Tr(*f* ) = *f*^# *⊙ κ*˙ l = *p*l *⊙ f*^ *⊙* *M*n (*p*r *⊙ f*^)n *⊙ κ*˙ l trace) formula. This follows from the description of iteration (*—*)# in Lemma [5.1](#_bookmark20):

= *p*l *⊙ f ⊙* *M*n (*κ*˙ r *⊙ p*r *⊙ f* )n *⊙ κ*˙ l

l n r r l

= *p ⊙ f ⊙* id *M M* (*κ*˙ *⊙ p ⊙ f* )n+1 *⊙ κ*˙

= (*p*l *⊙ f ⊙ κ*˙ l) *M* (*p*l *⊙ f ⊙* *M*n (*κ*˙ r *⊙ p*r *⊙ f* )n+1 *⊙ κ*˙ l)

(*∗*)

= *f*

ll

*M* (*p*l

*⊙ f ⊙* *M*n

*κ*˙ r

*⊙* (*p*r

*⊙ f ⊙ κ*˙

r)n *⊙ p*r

*⊙ f ⊙ κ*˙ l)

= *f*ll *M* *p*l *⊙ f ⊙ κ*˙ r *⊙* *M*n (*p*r *⊙ f ⊙ κ*˙ r)n *⊙ f*rl

= *f*ll *M f*lr*f* ٨ *f*rl*.*

rr

The marked equation holds because

(*κ*˙ r *⊙ p*r *⊙ f* )n+1 *⊙ κ*˙ l = *κ*˙ r *⊙* (*p*r *⊙ f ⊙ κ*˙ r)n *⊙ p*r *⊙ f ⊙ κ*˙ l*,*

which is obtained by induction.

The identity and uniformity properties are a consequence of Lemma [3.3](#_bookmark10).

**Example 5.3** We shall quickly review what this monoidal trace amounts to for a map *f* : *X* + *U → T* (*Y* + *U* ) where *T* is one of the monads *У, L, Ð, Q*(*−*) from Example [3.2](#_bookmark8).

* 1. For the powerset monad *У* we get Tr(*f* ): *X → У*(*Y* ) given by:

*y ∈* Tr(*f* )(*x*) *⇐⇒ En ∈* N*.* (*n, y*) *∈* tr(*f* )(*x*)

*⇐⇒ Eu*1*,..., u*n *∈ U. u*1 *∈ f* (*x*) *Λ u*2 *∈ f* (*u*1) *Λ · · · Λ u*n *∈ f* (*u*n*−*1) *Λ y ∈ f* (*u*n)*.*

* 1. The lift monad yields Tr(*f* ): *X →* 1+ *Y* as

Tr(*f* )(*x*) = up(*y*) *⇐⇒ En ∈* N*.f* (*x*) = up(*u*1) *Λ f* (*u*1) = up(*u*2) *Λ · · ·Λ*

*f* (*u*n*−*1) = up(*u*n) *Λ f* (*u*n) = up(*y*)*.*

* 1. The subdistribution monad yields Tr(*f* ): *X → Ð*(*Y* ) with: Tr(*f* )(*x*)(*y*)

= Σ Σ

*f* (*x*)(*u*1) *· f* (*u*1)(*u*2) *· ... · f* (*u*n*−*1)(*u*n) *· f* (*u*n)(*y*)*.*

n*∈*N u1,...,u*n∈*U

* 1. Similarly, the quantale monad yields Tr(*f* ): *X → Q*У with:

Tr(*f* )(*x*)(*y*)

= *f* (*x*)(*u*1) *· f* (*u*1)(*u*2) *· ... · f* (*u*n*−*1)(*u*n) *· f* (*u*n)(*y*)*.*

n*∈*N u1,...,u*n∈*U

We have already seen that Kleisli categories of quantale monads are special, because they have biproducts. But there is more.

**Lemma 5.4** *The Kleisli category Kl*(*Q*(*−*)) *of the monad Q*(*−*) *for a commutative*

*quantale Q has an involution* (*—*)*†*: *Kl*(*Q*(*−*))*op —*=*→ Kl*(*Q*(*−*)) *that preserves biprod-*

*ucts and (monoidal) traces.*

**Proof** On objects one has *X†* = *X* and on a morphism *f* : *X → Q*У one gets *f†*: *Y → Q*X by *f†*(*y*)(*x*) = *f* (*x*)(*y*). Clearly, (*—*)*††* = id. Commutativity of *Q*’s monoid (1*, ·*) is needed to show that (*—*)*†* preserves composition and traces.

# A category for bidirectional monadic computation

In this section we continue to work with a monad *T* on a category C as in Require- ments [4.7](#_bookmark17) for which we thus have both coalgebraic traces (as in Proposition [3.1](#_bookmark4)) and monoidal traces (by Theorem [5.2](#_bookmark21)). Then we can apply the standard “Int” con- struction from [[13](#_bookmark39)]. We shall write *Bd*(*T* ) for the resulting category Int(*Kl*(*T* )) of “bidirectional computations of type *T* ”.

This final section only contains an explicit description of this category *Bd*(*T* ) and a brief examination of our standard examples.

**Definition 6.1** Let *Bd*(*T* ) be the category with:

**Objects** *A* = (*A*l*, A*r) consisting of pairs of objects *A*l*, A*r *∈* C;

**Morphisms** *f* : *A → B* are maps *f* : *A*l + *B*r *→ T* (*B*l + *A*r) in C. Of course they may also be described as maps *A*l + *B*r *→ B*l + *A*r in the Kleisli category *Kl*(*T* );

**Identities** idA: *A → A* are (Kleisli) identities *A*l + *A*r *→ T* (*A*l + *A*r);

**Composition** For *f* : *A → B* and *g*: *B → C*, that is for *f* : *A*l + *B*r *→ T* (*B*l + *A*r) and *g*: *B*l + *C*r *→ T* (*C*l + *B*r), the composite *g ◦ f* is the (monoidal) trace of the following “obvious” map: (*A*l + *C*r)+ *B*r *→ T* ((*C*l + *A*r)+ *B*r), given explicitly in *Kl*(*T* ) as:

[(*κ*˙ 1 +˙ id) *⊙ g ⊙ κ*˙ 1*, κ*˙ 1 *⊙ κ*˙ 2] *⊙ f ⊙ κ*˙ 1*,* (*κ*˙ 1 +˙

id) *⊙ g ⊙ κ*˙ 2

[(*κ*˙ 1 +˙

id) *⊙ g ⊙ κ*˙ 1*, κ*˙ 1 *⊙ κ*˙ 2] *⊙ f ⊙ κ*˙ 2 *.*

We refer to [[13](#_bookmark39)] for the proof of the fact that this yields a compact closed category, with a full and faithful functor *Kl*(*T* ) *→ Bd*(*T* ) given by *A '→* (*A,* 0). Such proofs are non-trivial, and can best be done using a suitable graphical notation.

In the remainder we briefly review our running examples. For the lift monad *L* the category *Bd*(*L*) contains the essence of the category of games *G* as described in [[3](#_bookmark28)]. There, the objects can be described in terms of pairs of sets (*A*l*, A*r) of moves, of a player (left, say) and opponent (right), together with additional structure, given by a set of plays, as suitable subset of the set of (*A*l + *A*r)٨ sequences of moves. Morphisms *A → B* in *G* are “strategies”, that can be described as certain partial

functions *A*l + *B*r *- B*l + *A*r, that is [3](#_bookmark38) , as Kleisli maps *A*l + *B*r *→* 1+ (*B*l + *A*r). Composition of these strategies takes place via Girard’s “execution formula”, which corresponds to composition as described in Definition [6.1](#_bookmark25).

The category *Bd*(*Ð*) for the distribution monad *Ð* does not seem to have been studied yet. The other example involving quantale monads gives rise to a separate result, yielding a setting for quantum computation, see [[2](#_bookmark27)]. It includes the familiar situation of relations.

**Proposition 6.2** *The category Bd*(*Q*(*−*)) *obtained from the quantale monad Q*(*−*)

*for a commutative quantale Q is strongly compact closed.*

**Proof** The involution (*—*)*†* from Lemma [5.4](#_bookmark23) is preserved by the “Int” construction, as claimed in [[1](#_bookmark26)].

# Acknowledgement

Thanks to are due to Masahito Hasegawa for helpful comments, and to referees of (earlier versions of) this paper.

# References

1. S. Abramsky. Abstract scalars, loops, and free traced and strongly compact closed categories. In J.L. Fiadeiro, N. Harman, M. Roggenbach, and J. Rutten, editors, *Conference on Algebra and Coalgebra in Computer Science (CALCO 2005)*, number 3629 in Lect. Notes Comp. Sci., pages 1–31. Springer, Berlin, 2005.
2. S. Abramsky and B. Coecke. A categorical semantics of quantum protocols. In *Logic in Computer Science*, pages 415–425. IEEE, Computer Science Press, 2004.
3. S. Abramsky, R. Jagadeesan, and P. Malacaria. Full abstraction for PCF. *Inf. & Comp.*, 163:409–470, 2000.
4. J. Ad´amek, S. Milius, and J. Velebil. Elgot theories: A new perspective of iteration theories (extended abstract). In S. Abramsky, M. Mislove, and C. Palamidessi, editors, *Mathematical Foundations of Programming Semantics*, number 249 in Elect. Notes in Theor. Comp. Sci., pages 407–427. Elsevier, Amsterdam, 2009.
5. M.A. Arbib and E.G. Manes. *Algebraic Approaches to Program Semantics*. Texts and Monogr. in Comp. Sci. Springer, Berlin, 1986.
6. S.L. Bloom and Z. E´sik. *Iteration Theories: The Equational Logic of Iterative Processes*. EATCS Monographs. Springer, Berlin, 1993.
7. D. Coumans and B. Jacobs. Scalars, monads and categories, 2010. Manuscript.
8. C.C. Elgot. Monadic computation and iterative algebraic theories. In H.E. Rose and J.C. Shepherson, editors, *Logic Colloquium ’73*, pages 175–230, Amsterdam, 1975. North-Holland.
9. E. Haghverdi. *A categorical approach to linear logic, geometry of proofs and full completeness*. PhD thesis, Univ. of Ottawa, Canada, 2000.
10. E. Haghverdi. Unique decomposition categories, geometry of interaction and combinatory logic. *Math. Struct. in Comp. Sci.*, 10:205–231, 2000.
11. M. Hasegawa. The uniformity principle on traced monoidal categories. In *Category Theory and Computer Science*, number 69 in Elect. Notes in Theor. Comp. Sci. Elsevier, Amsterdam, 2003.
12. I. Hasuo, B. Jacobs, and A. Sokolova. Generic trace theory. *Logical Methods in Comp. Sci.*, 3(4:11), 2007.

3 These strategies are maps ƒ: M *P* + M *O* ~ M *O* + M *P* in the notation of [[3](#_bookmark28), Section 2.4].

*A B A B*

1. A. Joyal, R. Street, and D. Verity. Traced monoidal categories. *Math. Proc. Cambridge Phil. Soc.*, 119(3):425–446, 1996.
2. K.I. Rosenthal. *Quantales and their applications*. Number 234 in Pitman Research Notes in Math. Longman Scientific & Technical, 1990.
3. A. Simpson and G. Plotkin. Complete axioms for categorical fixed-point operators. In *Logic in Computer Science*, pages 30–41. IEEE, Computer Science Press, 2000.
4. M.B. Smyth and G.D. Plotkin. The category theoretic solution of recursive domain equations. *SIAM Journ. Comput.*, 11:761–783, 1982.