

# A complete logic for behavioural equivalence in coalgebras of finitary set functors

David Sprunger

February 19, 2016

## Abstract

This paper presents a sound and complete sequent-style deduction system for determining behavioural equivalence in coalgebras of finitary set functors preserving weak pullbacks. We select finitary set functors because they are quotients of polynomial functors: the polynomial functor provides a ready-made signature and the quotient provides necessary additional axioms. We also show that certain operations on functors can be expressed with uniform changes to the presentations of the input functors, making this system compositional for a range of widely-studied classes of functors, such as the Kripke polynomial functors. Our system has roots in the  $FLR_0$  proof system of Moschovakis et al., particularly as used by Moss, Wennstrom, and Whitney for non-wellfounded sets. Similarities can also be drawn to expression calculi in the style of Bonsangue, Rutten, and Silva.

## 1 Introduction

In this paper, we propose a logic for detecting bisimilar states in coalgebras of finitary Set-endofunctors. We focus on finitary functors because they have presentations whereby they can be represented as the quotient of a signature functor by a collection of equations.<sup>1</sup> The signature provides a syntax in which the coalgebras can be expressed, and the equations add the axioms necessary to distinguish reasoning among functors of similar syntax.

In particular, we will consider *specifications* on sets of variables in the signature of finitary functors. These are total assignments of variables to terms which serve as definitions, and may be considered a recasting of the longstanding tradition of systems of simultaneous equations going back to Kahn, Manna and Vuillemin, and Lawvere. We show every coalgebra of a finitary functor has at least one corresponding specification.

Our system is comparable to  $FLR_0$ , as considered by Moss in [2] and Moss et al. in [3].  $FLR_0$  has distinctive terms of the form  $x_i$  **where**  $\{x_1 =$

---

<sup>1</sup>Finitary functors in finitely presentable categories outside of Set may also have finitary presentations, see [1].

$A_1, \dots, x_n = A_n\}$ . This **where** operator binds variables, but also allows terms to carry their context. As a consequence, many of the  $FLR_0$  rules concern moving a definition in and out of a subscope or evolving the term before the **where** clause. We avoid these issues by fixing a specification sending  $x_i$  to  $A_i$ , roughly, and considering terms with this context backgrounded. Additionally,  $FLR_0$  and the full  $FLR$  language are intended as general languages of recursion with semantics of various flavors. The application here to coalgebras distinguishes our version somewhat.

We might also compare this work to that of Bonsangue et al. in [4] and [5] or Milius' related work in the setting of vector spaces [6], where a  $\mu$  operator provides a similar variable binding. The work of Bonsangue et al. feature an inductive class of functors, the so-called Kripke polynomial functors, and a syntax of expressions based on the inductive class. They build a sound and complete axiomatization for these expressions which is compositional, meaning the laws involved are built in parallel with the definition of the functor and the expressions. We show that the presentations involved in our setting enjoy similar compositional properties.

In Chapter 5 of Silva's PhD thesis [7] and the related paper [8], she gives an extension of this  $\mu$  calculus to finitary functors, demonstrating that the expressions of this calculus exactly coincide with the behaviours of locally finite coalgebras. However, at the end of this work, questions regarding axiomatization and uniform proofs of soundness and completeness for the system are left open.

We are able to prove soundness and completeness for our logic for the finitary functors which preserve weak pullbacks, a common condition with numerous pleasant coalgebraic consequences including that bisimilarity and behavioural equivalence coincide, see Rutten [9]. In particular, polynomial functors and the finite powerset functor preserve weak pullbacks, so the functors in our setting properly include those of Bonsangue et al. and Moss et al.

*Outline.* In Section 2, we briefly recall some background on coalgebras, signatures, and finitary functors. This section introduces the interplay between coalgebras of a finitary functor and coalgebras of its related signature functor that are of central importance to later sections. In Section 3, we introduce bisimulation up-to-presentation, a novel up-to technique which permits expressing bisimulations for finitary functors in terms of bisimulations for their signatures. In Section 4, we present a formal proof system capturing the notion of bisimulation-up-to. We show this system is sound and complete in the sense that it detects the bisimilarity of states in coalgebras for finitary functors preserving weak pullbacks. In Section 5, we note that signatures for previously studied inductive classes of these functors—including the Kripke polynomial functors—can be constructed compositionally. This allows the proof system developed in Section 4 to be constructed compositionally as well.

## 2 Background

In this section, we recall definitions and basic results about coalgebras, finitary signatures, finitary functors, and introduce the notion of a specification. Our setting is the category  $\mathbf{Set}$ , and all functors are assumed to be  $\mathbf{Set}$ -endofunctors. Additionally, we will often assume functors preserve weak pullbacks, but make a special note when this assumption is needed.

### 2.1 Coalgebras

Given a  $\mathbf{Set}$ -endofunctor  $F$ , an  $F$ -coalgebra is a set  $X$  together with a map  $f : X \rightarrow FX$ . The set is often called the *carrier* of the coalgebra, while  $f$  gives its *structure* or *dynamics*.

A *coalgebra morphism* from an  $F$ -coalgebra  $(X, f)$  to another  $F$ -coalgebra  $(Y, g)$  is a map  $\varphi : X \rightarrow Y$  such that the following diagram commutes:

$$\begin{array}{ccc} X & \xrightarrow{f} & FX \\ \varphi \downarrow & & \downarrow F\varphi \\ Y & \xrightarrow{g} & FY \end{array}$$

$F$ -coalgebras together with coalgebra morphisms between them form a category, often denoted  $\mathbf{Coalg}_F$ . Of particular interest are the  $F$  where  $\mathbf{Coalg}_F$  has a final object. This final coalgebra has a natural interpretation as a semantic object: since there is a unique coalgebra morphism from any  $F$ -coalgebra into it, points in coalgebras which have the same image in the final coalgebra can be considered (behaviourally) equivalent to one another.

A related notion is that of an (Aczel-Mendler)  $F$ -bisimulation on a coalgebra. An  $F$ -bisimulation is a relation  $R \subseteq X \times X$  such that there is an  $F$ -coalgebra structure on  $R$ ,  $\rho$ , such that the following diagram holds:<sup>2</sup>

$$\begin{array}{ccccc} X & \xleftarrow{\pi_1} & R & \xrightarrow{\pi_2} & X \\ f \downarrow & & \downarrow \rho & & \downarrow f \\ FX & \xleftarrow{F\pi_1} & FR & \xrightarrow{F\pi_2} & FX \end{array}$$

Roughly speaking, a relation is a bisimulation if two points in a coalgebra being related implies their structures are also related. This gives a different notion of equivalence, which is known to coincide with the behavioural equivalence for weak pullback preserving functors. For more details, we refer the reader to Rutten [9].

---

<sup>2</sup>Throughout this paper we will write  $\pi_i$  for the more cumbersome  $\pi_i|_R$ .

## 2.2 Finitary signatures and functors

A *finitary signature* is a set  $\Sigma$  with a map  $ar : \Sigma \rightarrow \omega$ . In the sequel, we often abbreviate “finitary signature” to “signature” and refer to the signature as  $\Sigma$  instead of  $(\Sigma, ar)$  when there is no risk of confusion. The elements of  $\Sigma$  are the *symbols* of the signature, and each symbol  $f \in \Sigma$  has the *arity*  $ar(f)$ . The collection of all symbols with arity  $n$  is denoted  $\Sigma_n$ .

Each finitary signature has an associated *signature functor*,  $H_\Sigma$ , given by  $\prod_n \Sigma_n \times X^n$ . We denote a typical element of  $H_\Sigma X$  by  $f(x_1, \dots, x_{ar(f)})$  if  $x_i \in X$  or  $f(\vec{x})$  if  $\vec{x} : ar(f) \rightarrow X$ .  $H_\Sigma X$  is often referred to as the set of all “flat terms” using symbols from  $\Sigma$  with variables from  $X$ .

$F$  is a *finitary functor* if there is a finitary signature  $\Sigma$  together with a (pointwise) epic natural transformation  $\epsilon : H_\Sigma \rightarrow F$ .<sup>3</sup> If  $F$  is a finitary functor, we say  $(\Sigma, \epsilon)$  is a *presentation* of  $F$ . Finitary functors have a number of alternate characterizations, including functors which preserve  $\omega$ -filtered colimits [1].

**Example 2.1.** For each set  $A$ , the constant functor  $FX = A$  is finitary with signature  $\Sigma = \Sigma_0 = A$  and the transformation  $\epsilon$  with components  $\epsilon_X : a() \mapsto a$ .

**Example 2.2.** The identity functor is finitary with signature  $\Sigma = \Sigma_1 = \{*\}$  and the transformation  $\epsilon$  with components  $\epsilon_X : *(x) \mapsto x$ .

**Example 2.3.** The finite powerset functor  $\mathcal{P}_\omega$  is finitary with signature  $\Sigma_n = \{\sigma_n\}$  and the transformation  $\epsilon_X : \sigma_n(x_1, \dots, x_n) \mapsto \{x_1, \dots, x_n\}$ . Note that unlike the previous two transformations, this  $\epsilon$  is not an isomorphism.

**Example 2.4.** The 3 powerset functor  $\mathcal{P}_3$ , which assigns each set to the set of its subsets of cardinality  $< 3$ , is finitary with signature  $\Sigma_n = \{\sigma_n\}$  for  $0 \leq n < 3$  and the same  $\epsilon$  as in  $\mathcal{P}_\omega$ , restricted to the smaller set of terms.

**Example 2.5.** The functor  $ZX = \{0, 1\} \times X \times X$  is finitary with signature  $\Sigma = \Sigma_2 = \{0 : \text{zip}, 1 : \text{zip}\}$  and transformation given by

$$\epsilon_X(i : \text{zip}(x_1, x_2)) = (i, x_1, x_2).$$

We call this the “zip functor” in the sequel, though this is not standard terminology.

A *specification in the signature*  $\Sigma$  is an  $H_\Sigma$ -coalgebra, a set  $X$  together with a function  $d : X \rightarrow H_\Sigma X$ . Elements of  $X$  are called *variables*, and  $d$  gives their *definition*. Every specification in the signature of a finitary functor  $F$  gives rise to an  $F$ -coalgebra: given  $d : X \rightarrow H_\Sigma X$  postcomposing with  $\epsilon_X$  yields  $\epsilon_X \circ d : X \rightarrow FX$ .

---

<sup>3</sup>That is, we are assuming each component  $\epsilon_X$  is epic. For transformations between functors into  $\text{Set}$ , pointwise epic and epic in the functor category coincide. [10, p. 91]

A single  $F$ -coalgebra  $(X, f)$  will correspond to potentially many specifications in its signature. For each section  $s$  of  $\epsilon_X$ , the composition  $s \circ f : X \rightarrow H_\Sigma X$  is a specification in  $\Sigma$ . The  $F$ -coalgebra related to each of these specifications will be, not surprisingly,  $(X, f)$ . Note at least one section of  $\epsilon_X$  is guaranteed to exist since  $\text{Set}$  has split epis.

At a broad level, this paper could be seen as an attempt to use the quotient relationship  $\epsilon : H_\Sigma \twoheadrightarrow F$  between the functors  $H_\Sigma$  and  $F$  to understand the relationships between  $H_\Sigma$ - and  $F$ -coalgebras and particularly the relationships between  $H_\Sigma$ - and  $F$ -bisimulations. Rather than constantly clarifying which functor we are considering, we hereafter reserve “specification” to mean an  $H_\Sigma$ -coalgebra, and the undecorated “coalgebra” to mean  $F$ -coalgebra.

Since we can readily recast specifications and coalgebras for finitary  $\text{Set}$  functors, we translate standard notions from coalgebras to specifications. For example,  $R$  is an  $F$ -bisimulation on the specification  $(X, d)$  when it is an  $F$ -bisimulation on the coalgebra  $(X, \epsilon_X \circ d)$ , the *standard semantics* for a variable in a specification is its image in a given final  $F$ -coalgebra, and two variables are *behaviourally equivalent* when they have the same standard semantics. Note that though the standard semantics of a variable depends on the final coalgebra under consideration, whether two variables are behaviourally equivalent is independent of this choice.

We write  $\vDash_{(X,d)} x = y$  when  $x$  and  $y$  are behaviourally equivalent states in the specification. When the specification is clear from context, we write  $\vDash x = y$ . (This notation will be relevant mostly in Section 4.)

**Example 2.6.** We can give a specification for the zip functor with  $X = \{x, y, z, w\}$  and

$$\begin{array}{ll} d(x) = 0 : \text{zip}(y, z) & d(y) = 1 : \text{zip}(x, w) \\ d(z) = 1 : \text{zip}(z, w) & d(w) = 0 : \text{zip}(w, z) \end{array}$$

As shown by Kupke and Rutten in [11] and Grabmayer et al. in [12], a final coalgebra for this functor is the set of streams in  $\{0, 1\}$ .<sup>4</sup> With this final coalgebra in mind the standard semantics for  $x$  is the Thue-Morse sequence.

**Example 2.7.** Another zip specification for  $Y = \{x, y, z, w, u, v, q\}$  is given by

$$\begin{array}{lll} d(x) = 0 : \text{zip}(y, z) & d(y) = 1 : \text{zip}(x, v) & d(v) = 0 : \text{zip}(w, u) \\ d(w) = 0 : \text{zip}(w, z) & d(z) = 1 : \text{zip}(z, w) & d(u) = 1 : \text{zip}(u, v) \\ & d(q) = 0 : \text{zip}(y, u) & \end{array}$$

---

<sup>4</sup>Kupke and Rutten actually considered a slight variation on  $\mathbf{Coalg}_Z$  which has the same final coalgebra but a slightly different final map.

In this specification, the states  $x$  and  $q$  are behaviourally equivalent. Our goal is to give a uniform account for detecting this behavioural equivalence.

Note also  $x$  in this example and  $x$  in Example 2.6 are behaviourally equivalent. We could consider the problem of showing the equivalence of two variables in two separate specifications, but by taking the disjoint union of the two specifications and determining equivalence within this single joint specification we get the same effect.

### 3 Bisimulation up to presentation

In this section, we introduce the notion of bisimulation up to presentation. Roughly, bisimulations up to presentation are  $H_\Sigma$ -bisimulations relaxed up to the kernel of  $\epsilon$  in such a way that they correspond nicely to  $F$ -bisimulations. This allows us to detect  $F$ -bisimulations using the more syntactic  $H_\Sigma$ -bisimulations and so-called  $\epsilon$  laws. We also give an alternate characterization of bisimulation up to presentation and several related sufficient criteria to conclude that a relation is a subset of the bisimilarity relation.

Since bisimulation up to presentation provides an alternate criterion which suffices for detecting bisimulations, we have intentionally named this type of relation in the style of enhanced coalgebraic bisimulations studied recently by Rot et al. [13] with veins of research going back to Milner, Park, Sangiorgi, and others. We also struck by the similarities between the results in Section 3.2 and the flavor of standard up-to results. However, we are unaware of a formal connection between these bodies of work since our setting relates bisimulations of two related functors, and the standard literature deals with bisimulations of a single functor. We would be very glad to learn of a connection, though.

Recall the standard (Aczel-Mendler) bisimulation diagram for an  $H_\Sigma$ -bisimulation on  $X$ :

$$\begin{array}{ccccc} X & \xleftarrow{\pi_1} & R & \xrightarrow{\pi_2} & X \\ d \downarrow & & \downarrow \rho & & \downarrow d \\ H_\Sigma X & \xleftarrow{H_\Sigma \pi_1} & H_\Sigma R & \xrightarrow{H_\Sigma \pi_2} & H_\Sigma X \end{array}$$

We say  $R \subseteq X \times X$  is a *bisimulation up to the presentation*  $(\Sigma, \epsilon)$  if there is a  $\rho : R \rightarrow H_\Sigma R$  such that  $\epsilon_X \circ d \circ \pi_i = \epsilon_X \circ H_\Sigma \pi_i \circ \rho$  for  $i \in \{1, 2\}$ . That is,  $\rho$  nearly gives  $R$  a  $H_\Sigma$ -coalgebra structure except that the paths in the diagram above are coequalized by  $\epsilon_X$  instead of commuting outright.

$$\begin{array}{ccccc} R & \xrightarrow{\pi_i} & X & \xrightarrow{d} & H_\Sigma X \\ \rho \downarrow & & & & \downarrow \epsilon_X \\ H_\Sigma R & \xrightarrow{H_\Sigma \pi_i} & H_\Sigma X & \xrightarrow{\epsilon_X} & FX \end{array}$$

**Theorem 1.** *For all specifications  $(X, d)$  for a finitary set functor with presentation  $(\Sigma, \epsilon)$ , a relation  $R \subseteq X \times X$  is an  $F$ -bisimulation if and only if it is a bisimulation up to the presentation  $(\Sigma, \epsilon)$ .*

*Proof.* ( $\Leftarrow$ ) Let  $\rho$  give  $R$  the structure of a bisimulation up to the presentation  $(\Sigma, \epsilon)$ . Then  $\epsilon_R \circ \rho$  gives  $R$  an  $F$ -coalgebra structure such that

$$\begin{array}{ccccc} X & \xleftarrow{\pi_1} & R & \xrightarrow{\pi_2} & X \\ \epsilon_X \circ d \downarrow & & \downarrow \epsilon_R \circ \rho & & \downarrow \epsilon_X \circ d \\ FX & \xleftarrow{F\pi_1} & FR & \xrightarrow{F\pi_2} & FX \end{array}$$

commutes, so  $R$  is an  $F$ -bisimulation.

( $\Rightarrow$ ) Given an  $F$ -bisimulation structure  $\phi$  on  $R$ , we claim  $s \circ \phi$  gives a bisimulation up to presentation structure to  $R$  where  $s$  is any section of  $\epsilon_R$ . To see this, note  $\epsilon_X \circ H_\Sigma \pi_i \circ s \circ \phi = F\pi_i \circ \epsilon_R \circ s \circ \phi = F\pi_i \circ \phi = \epsilon_X \circ d \circ \pi_i$   $\square$

We note that this statement is not that the  $F$ -bisimulation structures are in 1-1 correspondence with the bisimulation up to presentation structures, but that the carriers are in a 1-1 correspondence. Much like the correspondence between coalgebras and specifications, there is a possibly distinct bisimulation up to presentation structure for each section of  $\epsilon_R$ .

**Corollary 1.** *The biggest  $F$ -bisimulation on a coalgebra is the biggest bisimulation up to the presentation  $(\Sigma, \epsilon)$  on that coalgebra.*

We denote bisimilarity, the biggest  $F$ -bisimulation on a coalgebra, by  $\sim$ . The bisimilarity relation is known to be an equivalence relation on all coalgebras for all Set functors preserving weak pullbacks, see e.g. Rutten [9].

### 3.1 An explicit characterization

We defined a bisimulation up to presentation using a variation of the Aczel-Mendler diagram, but there is a more concrete characterization for bisimulations up to presentation which we describe in this section.

For flat terms  $\alpha, \beta \in H_\Sigma X$  we write  $\alpha =_\epsilon \beta$  to mean  $\epsilon_X(\alpha) = \epsilon_X(\beta)$ . If  $\alpha =_\epsilon \beta$  we say this is an  $\epsilon$  law, or that  $\alpha$  may be rewritten to  $\beta$  using  $\epsilon$  laws. Note the  $=_\epsilon$  relation on  $H_\Sigma X$  is an equivalence relation.

**Definition** ( $c(R)$ ). *The flat contextual closure of a relation  $R \subseteq X \times X$  is the relation  $c(R) \subseteq H_\Sigma X \times H_\Sigma X$  defined by  $f(x_1, \dots, x_{ar(f)}) c(R) f(y_1, \dots, y_{ar(f)})$  if and only if  $x_i R y_i$  for all  $1 \leq i \leq ar(f)$ .*

We denote the pointwise composition of relations  $R$  and  $S$  by  $R \bullet S$ . That is,  $x(R \bullet S)z$  iff there exists a  $y$  such that  $xRy$  and  $ySz$ .

**Definition** ( $\sim_R$ ). *Given a relation  $R$  on  $X$ , we define  $\sim_R$  to be  $=_\epsilon \bullet c(R) \bullet =_\epsilon$ , a relation on  $H_\Sigma X$ .*

Since  $\sim_R$  also depends on the transformation  $\epsilon$  it would be more proper to denote it  $\sim_{R,\epsilon}$ , but since  $\epsilon$  is standard for each functor we elide it from the notation.

Here we also emphasize the distinction between two very similar symbols:  $\sim$  denotes bisimilarity on  $X$ , and has no direct relationship with the symbol  $\sim_R$  just defined.

**Theorem 2.** *Given a finitary functor  $F$  and a specification  $(X, d)$  in that functor's signature,  $R \subseteq X \times X$  is a bisimulation up to the presentation  $(\Sigma, \epsilon)$  if and only if  $xRy$  implies  $d(x) \sim_R d(y)$ . More explicitly, for each  $(x, y) \in R$ , there is an  $f \in \Sigma_n$  and  $(x_1, y_1), \dots, (x_n, y_n) \in R$  such that:*

1.  $d(x) =_\epsilon f(x_1, \dots, x_n)$

2.  $d(y) =_\epsilon f(y_1, \dots, y_n)$

*Proof.* ( $\Rightarrow$ ) Suppose we have  $\rho : R \rightarrow H_\Sigma R$  such that  $\epsilon_X \circ d \circ \pi_i = \epsilon_X \circ H_\Sigma \pi_i \circ \rho$ . Let  $(x, y) \in R$  and write  $\rho(x, y) = f((x_1, y_1), \dots, (x_n, y_n))$  where  $f \in \Sigma$  and  $(x_i, y_i) \in R$ . Then  $(H_\Sigma \pi_1 \circ \rho)(x, y) = f(x_1, \dots, x_n)$ , so by the hypothesis on  $\rho$ ,  $d(x) =_\epsilon f(x_1, \dots, x_n)$ , as desired. Similarly considering  $(H_\Sigma \pi_2 \circ \rho)(x, y)$  yields item 2.

( $\Leftarrow$ ) Suppose we have a relation satisfying the latter condition, and define  $\rho : R \rightarrow H_\Sigma R$  by  $\rho(x, y) = f((x_1, y_1), \dots, (x_n, y_n))$ . Then by item 1,  $\epsilon_X \circ d \circ \pi_1 = \epsilon_X \circ H_\Sigma \pi_1 \circ \rho$ , and similarly for item 2.  $\square$

Theorem 2 gives an explicit characterization for bisimulations up to presentation. To check that a relation is a bisimulation up to presentation, for each pair  $(x, y)$  in the relation we need to rewrite  $d(x)$  and  $d(y)$  using  $\epsilon$  laws so that they have the same symbol and all corresponding variables are related.

We can now show  $x$  and  $q$  from Example 2.7 are related by a bisimulation up to presentation.

**Example 3.1.** Recall that the zip functor has function symbols  $\Sigma = \Sigma_2 = \{0 : \text{zip}, 1 : \text{zip}\}$  with no nontrivial  $\epsilon$  laws. Then for the specification

$$\begin{aligned} d(x) &= 0 : \text{zip}(y, z) & d(y) &= 1 : \text{zip}(x, v) & d(v) &= 0 : \text{zip}(w, u) \\ d(w) &= 0 : \text{zip}(w, z) & d(z) &= 1 : \text{zip}(z, w) & d(u) &= 1 : \text{zip}(u, v) \\ & & & & d(q) &= 0 : \text{zip}(y, u) \end{aligned}$$

we propose  $R = \{(x, q), (z, u), (w, v)\} \cup \Delta_X$  as a bisimulation up to presentation. The diagonal part clearly satisfies the required properties. Then

- $0 : \text{zip}(y, z) \sim_R 0 : \text{zip}(y, u)$  since  $yRy$  and  $zRu$ .
- $1 : \text{zip}(z, w) \sim_R 1 : \text{zip}(u, v)$  since  $zRu$  and  $wRv$ .

- $0 : \text{zip}(w, z) \sim_R 0 : \text{zip}(w, u)$  since  $wRw$  and  $zRu$ .

Since the zip functor has no nontrivial  $\epsilon$  laws this is just an ordinary  $H_\Sigma$  bisimulation. Matters are more complicated for non-polynomial functors.

**Example 3.2.** Recall that the functor  $\mathcal{P}_3$  from Example 2.4 has a presentation with three function symbols,  $\{\sigma_i\}_{i<3}$ , each with arity  $i$  and  $\epsilon$  laws of the forms  $\sigma_2(x, y) =_\epsilon \sigma_2(y, x)$  and  $\sigma_2(x, x) =_\epsilon \sigma_1(x)$ .

An example specification in this signature for  $X = \{x, y, z\}$  might be

$$d(x) = \sigma_2(x, y) \quad d(y) = \sigma_1(z) \quad d(z) = \sigma_2(z, z)$$

All of these are behaviourally equivalent, so they should be related by a bisimulation up to presentation. We propose  $R = \{(x, y), (y, z), (x, z), (z, z)\}$ . For this we need to check four things:

- $\sigma_2(x, y) \sim_R \sigma_1(z)$ : we use  $\sigma_1(z) =_\epsilon \sigma_2(z, z)$  to rewrite the RHS and note  $xRz$  and  $yRz$ .
- $\sigma_1(z) \sim_R \sigma_2(z, z)$ : uses the same rewrite and  $zRz$  twice.
- $\sigma_2(x, y) \sim_R \sigma_2(z, z)$ : immediate from  $xRz$  and  $yRz$ .
- $\sigma_2(z, z) \sim_R \sigma_2(z, z)$ : immediate from  $zRz$ .

Therefore all three of these variables are related by a bisimulation up to presentation.

### 3.2 Enhanced bisimulation up to presentation

In Example 3.1, we showed two variables in a specification were related by a bisimulation up to presentation, but in the course of this proof we added in the diagonal relation to make the bisimulation hypothesis go through. This is reminiscent of other combination bisimulation up to techniques, such as those studied by Rot et al. in [13]. In this section, we provide several enhancements to the bisimulation up to presentation technique which will be useful in the sequel.

First we note bisimulation up to presentation interacts well with union of bisimulations.

**Lemma 1.** *Suppose  $(X, d)$  is a specification for a finitary functor presented by  $(\Sigma, \epsilon)$ . Let  $S$  be any bisimulation on  $X$ ,  $T$  be a relation containing  $S$ , and  $R$  be a relation on  $X$  such that  $xRy$  implies  $d(x) \sim_T d(y)$ . Then  $R \cup S$  also has the property that  $(x, y) \in R \cup S$  implies  $d(x) \sim_T d(y)$ .*

*Proof.* Since  $S$  is a bisimulation on  $X$ , it is also a bisimulation up to presentation by Theorem 1. Then by Theorem 2,  $xSy$  implies  $d(x) \sim_S d(y)$ . Since  $S \subseteq T$ , we have  $c(S) \subseteq c(T)$  and therefore  $\sim_S \subseteq \sim_T$ . Hence  $xSy$  implies  $d(x) \sim_T d(y)$ . Combining this with the hypothesis on  $R$ , we have the desired result.  $\square$

**Corollary 2.** Recall that  $\sim$  is the biggest  $F$ -bisimulation on  $X$ . If  $R$  is a relation on a specification such that any of the following hold:

- $xRy \rightarrow d(x) \sim_{R \cup \Delta_X} d(y)$
- $xRy \rightarrow d(x) \sim_{R \cup \sim} d(y)$

then  $R \subseteq \sim$ .

*Proof.* Both  $\Delta_X$  and  $\sim$  are bisimulations so by Lemma 1, taking  $T = R \cup B$  where  $B \in \{\Delta_X, \sim\}$ , we get a relation  $T$  such that  $xTy \rightarrow d(x) \sim_T d(y)$ . Then  $T$  is a bisimulation up to presentation by Theorem 2, so  $R \subseteq T \subseteq \sim$ .  $\square$

Bisimulation up to presentation also behaves well with respect to symmetric closures:

**Lemma 2.** Suppose  $(X, d)$  is a specification for a finitary functor presented by  $(\Sigma, \epsilon)$ . Let  $T$  be any symmetric relation and  $R$  be any relation on  $X$  such that  $xRy$  implies  $d(x) \sim_T d(y)$ . Then  $s(R)$ , the symmetric closure of  $R$ , also has the property  $x s(R) y$  implies  $d(x) \sim_T d(y)$ .

*Proof.* It is easy to check that  $T$  symmetric implies  $c(T)$  symmetric, which in turn implies  $\sim_T$  symmetric. Then if  $x s(R) y$ , either  $xRy$  or  $yRx$  by definition of  $s(R)$ . The hypothesis on  $R$  yields  $d(x) \sim_T d(y)$  or  $d(y) \sim_T d(x)$ , respectively. Then  $\sim_T$  symmetric allows us to conclude that in either case  $d(x) \sim_T d(y)$ , as desired.  $\square$

**Corollary 3.** If  $R$  is a relation such that  $xRy \rightarrow d(x) \sim_{s(R)} d(y)$ , then  $R \subseteq \sim$ .

Bisimulation up to presentation for functors *preserving weak pullbacks* also plays well with equivalence closures. Preservation of weak pullbacks is a critical assumption here. We recall the following definition and theorem from [14, p.14], slightly recast to use our notation:

**Definition.** A presentation is dominated if for every  $\epsilon$  law  $f(\vec{x}) =_\epsilon g(\vec{y})$  where  $f \in \Sigma_n$ ,  $g \in \Sigma_m$ ,  $\vec{x} : n \rightarrow X$ ,  $\vec{y} : m \rightarrow X$ , there is a symbol  $h \in \Sigma_k$  and functions  $u : k \rightarrow n$  and  $v : k \rightarrow m$  such that  $h(u) =_\epsilon f(id_n)$ ,  $h(v) =_\epsilon g(id_m)$ , and  $\vec{x} \circ u = \vec{y} \circ v$ .

To give some intuition for dominated presentations, consider the presentation for  $\mathcal{P}_\omega$  from Example 2.3. We could say the symbol  $\sigma_4$  dominates the symbol  $\sigma_2$  since there is a  $u : 4 \rightarrow 2$ , namely  $u(i) = \lfloor \frac{i}{2} \rfloor$ , such that  $\sigma_4(u) = \sigma_4(0, 0, 1, 1) =_\epsilon \sigma_2(0, 1) = \sigma_2(id_2)$ . Then any time we have a term using  $\sigma_2$ , we could replace  $\sigma_2$  with  $\sigma_4$ , following the substitution scheme hinted at by  $u$ , and remain in the same component of the kernel of  $\epsilon$ . So, for example, this domination would imply  $\sigma_2(x, y) =_\epsilon \sigma_4(x, x, y, y)$ .

Note  $\sigma_3$  is also dominated by  $\sigma_4$ , for example by  $\sigma_3(0, 1, 2) = \sigma_4(0, 2, 1, 1)$ . This would allow us to rewrite  $\sigma_3(x, y, x) =_\epsilon \sigma_4(x, x, y, y)$ . Combining with the previous paragraph would allow us to derive  $\sigma_3(x, y, x) = \sigma_2(x, y)$  via rewrites to  $\sigma_4$ . We can then say that  $\sigma_3(x, y, x) =_\epsilon \sigma_2(x, y)$  is a consequence of the joint domination of  $\sigma_2$  and  $\sigma_3$  by  $\sigma_4$ .<sup>5</sup>

The verbiage for all this notation then is that a presentation is dominated means for every  $\epsilon$  law  $f(\vec{x}) =_\epsilon g(\vec{y})$  there is a dominating symbol  $h$  with two variable substitutions  $u$  and  $v$  such that the  $\epsilon$  law is a consequence of the joint domination of  $f$  and  $g$  by  $h$  via the substitutions  $u$  and  $v$ . We will be relying on facts about dominated presentations only in the proof of Lemma 3.

**Theorem 3** (Adámek, Gumm, Trnková). *A finitary functor weakly preserves pullbacks if and only if it has a dominated presentation.*

As a result of this theorem, for the next lemmas we can assume our presentation is dominated without loss of generality.

**Lemma 3.** *Suppose  $T$  is an equivalence relation and  $(\Sigma, \epsilon)$  is a dominated presentation. Then  $\sim_T$  is an equivalence relation.*

*Proof.* Reflexivity and symmetry are straightforward.  $T$  is reflexive and symmetric, therefore  $c(T)$  is reflexive and symmetric, and so  $=_\epsilon \bullet c(T) \bullet =_\epsilon$  is reflexive and symmetric.

Transitivity requires the dominated presentation. Suppose  $\alpha \sim_T \beta \sim_T \gamma$ . Then we can write

$$\alpha =_\epsilon \alpha' c(T) \beta' =_\epsilon \beta =_\epsilon \beta'' c(T) \gamma' =_\epsilon \gamma$$

Let  $\beta' = f(\vec{x})$  and  $\beta'' = g(\vec{y})$ . Then the above relations become:

$$\alpha =_\epsilon f(\vec{x}') c(T) f(\vec{x}) =_\epsilon g(\vec{y}) c(T) g(\vec{y}') =_\epsilon \gamma$$

Since we have a dominated presentation, we get  $h$ ,  $u$ , and  $v$  such that  $h(u) =_\epsilon f(id_n)$ ,  $h(v) =_\epsilon g(id_m)$  and  $\vec{x} \circ u = \vec{y} \circ v$ . The first two statements imply  $f(\vec{x}') =_\epsilon h(\vec{x}' \circ u)$  and  $g(\vec{y}') =_\epsilon h(\vec{y}' \circ v)$ .

We also know  $\vec{x}'(i) T \vec{x}(i)$  for all  $i \in [1, n]$  and  $\vec{y}(i) T \vec{y}'(i)$  for  $i \in [1, m]$ . Therefore,  $(\vec{x}' \circ u)(i) T (\vec{x} \circ u)(i) = (\vec{y} \circ v)(i) T (\vec{y}' \circ v)(i)$  for  $i \in [1, k]$ , where the middle equality is by the last condition guaranteed by the dominated presentation. Then since  $T$  is transitive we know  $(\vec{x}' \circ u)(i) T (\vec{y}' \circ v)(i)$  for  $i \in [1, k]$ .

Therefore we have produced terms such that

$$\alpha =_\epsilon h(\vec{x}' \circ u) c(T) h(\vec{y}' \circ v) =_\epsilon \gamma$$

and so  $\sim_T$  is transitive and hence is an equivalence relation.  $\square$

<sup>5</sup>Obviously, the joint domination is not unique in the case of  $\mathcal{P}_\omega$ .  $\sigma_2$  and  $\sigma_3$  are jointly dominated by  $\sigma_i$  for all  $i \geq 3$  and even for each dominating symbol there may be many different substitutions which yield the desired equation as a consequence of the joint domination.

**Lemma 4.** *Suppose  $(X, d)$  is a specification for a finitary functor preserving weak pullbacks with the dominated presentation  $(\Sigma, \epsilon)$ . Let  $T$  be any equivalence relation on  $X$  and  $R$  be any relation on  $X$  such that  $xRy$  implies  $d(x) \sim_T d(y)$ . Then  $e(R)$ , the equivalence closure of  $R$ , has the property  $x e(R) y$  implies  $d(x) \sim_T d(y)$ .*

*Proof.* By Lemmas 1 and 2, we know immediately that  $x sr(R) y$  implies  $d(x) \sim_T d(y)$ , where  $sr(R)$  is the symmetric reflexive closure of  $R$ . Hence we only have to consider  $(x, y) \in e(R) \setminus sr(R)$ . Therefore, suppose we have  $x sr(R) z sr(R) y$ . Then by the noted property of  $sr(R)$  we get  $d(x) \sim_T d(z) \sim_T d(y)$ . By the previous Lemma, since  $T$  is an equivalence relation and we have a dominated presentation,  $\sim_T$  is an equivalence relation and hence  $d(x) \sim_T d(y)$ .  $\square$

**Corollary 4.** *Suppose  $F$  preserves weak pullbacks and its presentation  $(\Sigma, \epsilon)$  is dominated. If  $R$  is a relation such that  $xRy$  implies  $d(x) \sim_{\epsilon(R)} d(y)$ , then  $R \subseteq \sim$ .*

The following corollary follows directly from the results above, but is of critical importance to our proof of soundness.

**Corollary 5.** *Suppose  $F$  preserves weak pullbacks and its presentation  $(\Sigma, \epsilon)$  is dominated. If  $R$  is a relation such that  $xRy$  implies  $d(x) \sim_{\epsilon(R \cup \sim)} d(y)$ , then  $R \subseteq \sim$ .*

## 4 A proof system for bisimulation up to presentation

In this section, we outline a formal proof system to capture the notion of bisimulation up to presentation. For this whole section, we assume  $F$  preserves weak pullbacks. We then prove this system to be sound and complete. Our system has judgements of the form  $R \vdash \sigma = \tau$  where  $R \subseteq X \times X$  and  $(\sigma, \tau) \in X \times X + H_\Sigma X \times H_\Sigma X$ . The inference rules are as follows:

$$\begin{array}{c}
\frac{}{R \vdash \sigma = \sigma} \textit{r} \qquad \frac{R \vdash \tau = \sigma}{R \vdash \sigma = \tau} \textit{s} \\
\frac{\alpha =_\epsilon \beta}{R \vdash \alpha = \beta} \textit{\epsilon} \qquad \frac{R \vdash \sigma = \tau \quad R \vdash \tau = \rho}{R \vdash \sigma = \rho} \textit{t} \\
\frac{}{\{\varphi\} \cup R \vdash \varphi} \textit{a} \qquad \frac{R \vdash x_1 = y_1 \quad \dots \quad R \vdash x_{ar(f)} = y_{ar(f)}}{R \vdash f(x_1, \dots, x_{ar(f)}) = f(y_1, \dots, y_{ar(f)})} \textit{c} \\
\frac{R \vdash \varphi \quad \forall (x, y) \in R. R \vdash d(x) = d(y)}{\vdash \varphi} \textit{b}
\end{array}$$

As usual, we say  $R \vdash \varphi$  when there is a proof tree using the above rules with the judgement  $R \vdash \varphi$  as the root. The notation  $\vdash \varphi$  is shorthand for  $\emptyset \vdash \varphi$ . Recall that  $\vDash x = y$  means that  $x$  and  $y$  are behaviourally equivalent (have the same image in the final coalgebra).

We should point out that  $R$  on the left side of the turnstile does not have the usual force of a full assumption. Rather, this  $R$  should be thought of as tracking unverified bisimulation hypotheses. In most rules this unverified bisimulation remains on both sides, except the axioms and the  $b$  rule, which essentially discharges a verified bisimulation from the left hand side. This  $b$  rule is probably the least intuitive, but is really just the coinductive proof principle. We also note its similarity to the Recursion Inference Rule from  $\text{FLR}_0$ , which was in mind as the system was constructed. [15]

Before we prove soundness and completeness, we give two example proofs using the system. This first example is based on Example 4.2 in Moss et al. [3].

**Example 4.1.** Consider the specification on  $X = \{x, y, r, s\}$  for  $\mathcal{P}_3$  defined by

$$d(x) = \sigma_2(x, y) \quad d(y) = \sigma_0 \quad d(r) = \sigma_2(r, s) \quad d(s) = \sigma_0$$

Let  $R = \{(x, r), (y, s)\}$ . The proof tree below witnesses  $\vdash x = r$ .

$$\frac{\frac{\frac{R \vdash x = r \quad a}{R \vdash \sigma_2(x, y) = \sigma_2(r, s)} \quad a \quad \frac{\frac{R \vdash y = s \quad a}{R \vdash \sigma_0 = \sigma_0} \quad c}{R \vdash \sigma_2(x, y) = \sigma_2(r, s)} \quad c}{\vdash x = r} \quad b$$

The next example is adapted from Example 3.2 in this paper and shows how some rules allow for shorter proofs.

**Example 4.2.** Consider the specification on  $X = \{x, y\}$  for  $\mathcal{P}_3$  defined by

$$d(x) = \sigma_2(x, y) \quad d(y) = \sigma_1(x)$$

Let  $R = \{(x, y)\}$ . The proof tree below witnesses  $\vdash x = y$ .

$$\frac{\frac{\frac{\frac{R \vdash x = y \quad a}{R \vdash x = x} \quad r \quad \frac{R \vdash y = x \quad s}{R \vdash \sigma_2(x, y) = \sigma_2(x, x)} \quad c}{R \vdash \sigma_2(x, y) = \sigma_2(x, x)} \quad c \quad \frac{R \vdash \sigma_2(x, x) = \sigma_1(x) \quad \epsilon}{R \vdash \sigma_2(x, y) = \sigma_1(x)} \quad t}{\vdash x = y} \quad b$$

Note that  $R$  contains a single pair though an unenhanced bisimulation proof would require three: the  $r$  rule allows us to omit  $(x, x)$ , and the  $s$  rule elides the mirror-image proof that  $d(y) = d(x)$  thereby allowing us to omit  $(y, x)$ .

## 4.1 Soundness

To help with our soundness proof, we define a new relational closure which we call the *presentational closure* of a relation  $R \subseteq X \times X$  on the carrier of a specification  $(X, d)$ . Recall from Section 3 that  $\sim_R$  is defined to be  $=_\epsilon \bullet c(R) \bullet =_\epsilon$ . That is,  $\alpha \sim_R \beta$  if we can rewrite  $\alpha$  and  $\beta$  using  $\epsilon$  laws so they have the same function symbol and all corresponding variables are related by  $R$ . The presentational closure of  $R$ , denoted  $pr(R)$ , is defined to be  $pr(R) \triangleq e(R \cup \sim) + \sim_{e(R \cup \sim)}$ , a relation on  $X + H_\Sigma X$ .

We note that  $e(R \cup \sim)$  is an equivalence relation on  $X$ , and  $\sim_{e(R \cup \sim)}$  is an equivalence relation on  $H_\Sigma X$  as a consequence. Since  $X$  and  $H_\Sigma X$  are disjoint,  $pr(R)$  is also an equivalence relation.

**Proposition 1.** *If  $R \vdash \varphi$ , then  $\varphi \in pr(R)$ .*

*Proof.* By induction on the proof tree. The base cases are  $r$ ,  $\epsilon$ , and  $a$ . We know  $pr(R)$  is an equivalence relation, hence  $r$ . The relation  $e(R \cup \sim)$  is reflexive, hence  $\sim_{e(R \cup \sim)}$  contains the relation  $=_\epsilon$ , hence  $\epsilon$ . All pairs in  $R$  are included in  $e(R \cup \sim)$ , hence  $a$ .

The induction steps are  $s$ ,  $t$ ,  $c$ , and  $b$ .  $s$  and  $t$  follow easily from the fact that  $pr(R)$  is an equivalence relation.

Suppose for all  $1 \leq i \leq ar(f)$ , we know  $(x_i, y_i) \in pr(R)$  and hence  $(x_i, y_i) \in e(R \cup \sim)$ . By definition of the flat contextual closure,  $f(x_1, \dots, x_{ar(f)}) c(e(R \cup \sim)) f(y_1, \dots, y_{ar(f)})$ . Since  $=_\epsilon$  is a reflexive relation,  $pr(R)$  contains  $c(R)$  and hence these two terms are related by  $pr(R)$ . Therefore the induction holds across the  $c$  rule.

Finally we consider the  $b$  rule, which allows one to discharge bisimulations from the left hand side. The induction hypothesis gives  $\varphi \in pr(R)$  and  $(d(x), d(y)) \in pr(R)$  for each  $(x, y) \in R$ . Since  $d(x), d(y) \in H_\Sigma X$  we know  $d(x) \sim_{e(R \cup \sim)} d(y)$ . Then by Corollary 5, we know  $R \subseteq \sim$ . Therefore,  $pr(R) = e(R \cup \sim) + \sim_{e(R \cup \sim)} = e(\sim) + \sim_{e(\sim)} = pr(\emptyset)$ . Then  $\varphi \in pr(\emptyset)$  since  $\varphi \in pr(R)$ .  $\square$

**Corollary 6 (Soundness).** *If  $\vdash x = y$ , then  $\vDash x = y$ .*

*Proof.* If  $\vdash x = y$ , then  $(x, y) \in e(\sim \cup \emptyset)$  by the previous proposition. However, clearly  $e(\sim \cup \emptyset) = e(\sim) = \sim$ , so  $x \sim y$ . A standard fact about functors preserving weak pullbacks is that two states in a coalgebra are bisimilar if and only if they are behaviourally equivalent [9], so  $x \sim y$  implies  $\vDash x = y$ .  $\square$

## 4.2 Completeness

**Lemma 5.** *If  $\alpha \sim_R \beta$ , then  $R \vdash \alpha = \beta$ .*

*Proof.*  $\alpha \sim_R \beta$  means there are  $f \in \Sigma_n$  and  $(x_1, y_1), \dots, (x_n, y_n) \in R$  such that

$$\alpha =_\epsilon f(x_1, \dots, x_n) \ c(R) \ f(y_1, \dots, y_n) =_\epsilon \beta.$$

then

$$\frac{\frac{\frac{R \vdash \alpha = f(x_1, \dots, x_n)}{\epsilon} \quad \frac{\frac{R \vdash x_1 = y_1}{a} \quad \dots \quad \frac{R \vdash x_n = y_n}{a}}{R \vdash f(x_1, \dots, x_n) = f(y_1, \dots, y_n)} \ c}{R \vdash \alpha = f(y_1, \dots, y_n)} \ t \quad \frac{R \vdash f(y_1, \dots, y_n) = \beta}{\epsilon} \ t}{R \vdash \alpha = \beta} \ t$$

is a witness for  $R \vdash \alpha = \beta$ .  $\square$

**Corollary 7** (Completeness). *If  $\vDash x = y$ , then  $\vdash x = y$ .*

*Proof.* Recall that  $\vDash x = y$  iff  $x \sim y$ . Since  $x$  and  $y$  are bisimilar, they are related by a bisimulation up to presentation, which we call  $R$ . By Theorem 2,  $uRv \rightarrow d(u) \sim_R d(v)$ . Syllogizing with the previous lemma yields  $uRv \rightarrow (R \vdash d(u) = d(v))$ . Trivially,  $R \vdash x = y$  by the  $a$  rule. Therefore,  $\vdash x = y$  by the  $b$  rule.  $\square$

## 5 Compositionality of presentations

Bisimulations up to presentation give a uniform way to reason about behavioural equivalence of variables in specifications for finitary functors, but there is a potential disadvantage. Certain inductive classes of functors have sound, complete, and **compositional** proof systems. That is, the rules for reasoning about coalgebras of a functor are built inductively in a manner corresponding to the definition of the functor. The prime example of this situation is that of the polynomial functors and the Kripke polynomial functors.

A functor is called *polynomial* if it is generated by the following BNF grammar:

$$P ::= A \mid Id \mid P + P \mid P \times P \mid P^B$$

where  $A$  is the constant functor having value  $A \in Set$  and  $B$  is a finite set. The *Kripke polynomial* class of functors adds the finite powerset functor:

$$K ::= A \mid Id \mid P_\omega(K) \mid K + K \mid K \times K \mid K^B$$

Bonsangue et al. build a sound, complete and compositional expression calculus to represent coalgebras of Kripke polynomial functors in [5]. We show presentations are similarly compositional, in that both the signature and the  $\epsilon$  transformation can be built inductively to parallel the construction of the functor. For the following three constructions, suppose  $F$  and  $G$  are finitary functors with presentations  $(\Sigma, \epsilon)$  and  $(\Sigma', \epsilon')$ .

## 5.1 Products

Let  $J = F \times G$ . Then we claim  $J$  has a presentation  $(\Sigma'', \epsilon'')$  where  $\Sigma''$  has all pairs of symbols,  $\Sigma''_n = \{(f, g) : f \in \Sigma, g \in \Sigma', ar(f) + ar(g) = n\}$ , and  $\epsilon'' : H_{\Sigma''} \rightarrow J$  has components

$$\epsilon''_X : (f, g)(x_1, \dots, x_n) \mapsto (\epsilon_X(f(x_1, \dots, x_{ar(f)})), \epsilon'_X(g(x_{ar(f)+1}, \dots, x_n))).$$

Then  $\epsilon''$  is an epic natural transformation as a consequence of  $\epsilon$  and  $\epsilon'$  being epic natural transformations.

We single out this particular presentation because it allows us to state the  $\epsilon''$  laws in terms of  $\epsilon$  and  $\epsilon'$  laws. By definition  $\epsilon''((f, g)(x_1, \dots, x_n)) = \epsilon''((f', g')(y_1, \dots, y_m))$  means

$$\begin{aligned} \epsilon(f(x_1, \dots, x_{ar(f)})) &= \epsilon(f'(y_1, \dots, y_{ar(f')})) \text{ and} \\ \epsilon'(g(x_{ar(f)+1}, \dots, x_n)) &= \epsilon'(g'(y_{ar(f')+1}, \dots, y_m)). \end{aligned}$$

Therefore,  $\epsilon''$  laws in this presentation are pairs of  $\epsilon$  and  $\epsilon'$  laws.

We note here that we could represent finite powers with a similar construction. If  $B$  is a finite set, a signature for  $F^B$  has symbols  $|B|$  tuples of symbols from  $\Sigma$  with arity the sum of the arities through the tuple. Then the  $\epsilon''$  laws are  $|B|$  tuples of  $\epsilon$  laws.

## 5.2 Coproducts

Let  $J = F + G$ . We write  $S + T = \{\text{inl}s : s \in S\} \cup \{\text{inr}t : t \in T\}$ . Then  $J$  has a presentation  $(\Sigma'', \epsilon'')$  where  $\Sigma''_n = \Sigma_n + \Sigma'_n$  and  $\epsilon''$  has components  $\epsilon''_X$  such that  $\begin{cases} \epsilon''_X(\text{inl}f(\vec{x})) = \text{inl}\epsilon_X(f(\vec{x})) \\ \epsilon''_X(\text{inr}g(\vec{x})) = \text{inr}\epsilon'_X(g(\vec{x})) \end{cases}$ . Since  $\epsilon$  and  $\epsilon'$  are epic natural transformations,  $\epsilon''$  is also an epic natural transformation.

Again, we can state the  $\epsilon''$  laws in terms of the  $\epsilon$  and  $\epsilon'$  laws. By definition,  $\epsilon''\alpha = \epsilon''\beta$  means  $\alpha$  and  $\beta$  are both labelled  $\text{inl}$  or are both labelled  $\text{inr}$ . In the former case, we have  $\text{inl}\epsilon(f(\vec{x})) = \text{inl}\epsilon(g(\vec{y}))$ , which is an  $\text{inl}$ -labelled instance of an  $\epsilon$ -law. Similarly, the latter case gives an  $\text{inr}$ -labelled instance of an  $\epsilon'$ -law.

## 5.3 Compositions

Let  $J = G \circ F$ . Then  $J$  has a presentation with symbols from the set  $\Sigma'' = \{(\sigma', (\sigma_1, \dots, \sigma_{ar(\sigma')})) : \sigma' \in \Sigma' \text{ and } \sigma_i \in \Sigma\}$ . For each symbol  $\sigma'' \in \Sigma''$  define  $w_{\sigma''}(i) = \sum_{j=1}^i ar(\sigma_j)$  for  $0 \leq i \leq ar(\sigma')$  and define  $\sigma''$  to have arity  $w_{\sigma''} = w_{\sigma''}(ar(\sigma'))$ . Given an  $ar(\sigma'')$  tuple from  $X$ , we let  $\vec{x}_i = (x_{w_{\sigma''}(i-1)+1}, \dots, x_{w_{\sigma''}(i)})$  for  $1 \leq i \leq ar(\sigma')$ , the slice of the variables

corresponding to  $\sigma_i$ . The natural transformation  $\epsilon''$  has components given by

$$\epsilon'' : \sigma''(x_1, \dots, x_{ar(\sigma'')}) \mapsto \epsilon'(\sigma'(\epsilon(\sigma_1(\vec{x}_1)), \dots, \epsilon(\sigma_{ar(\sigma')}(\vec{x}_{ar(\sigma'')})))).$$

This is an epic natural transformation and the  $\epsilon''$  laws can be stated again in terms of the  $\epsilon$  and  $\epsilon'$  laws.

#### 5.4 Kripke polynomial functors and other polynomial-like classes of functors

We have presentations for constant functors (Example 2.1), the identity functor (Example 2.2) and the finite power set functor (Example 2.3), which means the above constructions give a compositional presentation for each of the Kripke polynomial functors. Due to the previous section, we know bisimulation up to those presentations is a sound and complete proof system.

**Example 5.1.** Consider the functor  $F = A \times \text{Id}^B$  where  $B$  is finite. Following the constructions above,  $F$  has a presentation with function symbols  $\Sigma = \Sigma_{|B|} = \{(a, \beta) : a \in A, \beta : B \rightarrow \{*\}\}$  and equations  $\epsilon((a, \beta)(\vec{x})) = \epsilon((a', \beta')(\vec{x}'))$  iff  $a = a'$  and  $\vec{x} = \vec{x}'$ . Since there is only one function  $\beta : B \rightarrow \{*\}$ , we might as well omit that part of the function symbol and abbreviate  $(a, \beta)$  by just  $a$ , still with arity  $|B|$ .

Suppose we had a specification in this signature, like

$$d(x) = a(\overbrace{x, \dots, x}^{|B|}) \quad d(y) = a(\overbrace{z, \dots, z}^{|B|}) \quad d(z) = a(\overbrace{y, \dots, y}^{|B|})$$

The complete relation on  $\{x, y, z\}$  is a bisimulation up to presentation for this specification, since for each pair of variables, all corresponding variable pairs in the definitions are in the relation. Hence we can conclude these variables all have the same image in the final coalgebra.

The constructions above generalize previous results about inductive classes of functors since they assure us a compositional proof system exists for any inductive class using any of those formation rules.

## 6 Conclusion and future directions

In this paper, we presented a sequent-style deduction system for reasoning about behavioural equivalence of points in coalgebras and specifications of finitary functors in  $\text{Set}$ . This system was based on a relaxed version of  $H_\Sigma$ -bisimulations which nicely coincide with  $F$ -bisimulations called bisimulations up to presentation. We demonstrated this proof system was sound and

complete for finitary functors preserving weak pullbacks. We also demonstrated that three common operations on functors have uniform effects on both the signature and equations in a presentation.

One restriction in our setting we would like to remove is the totality restriction. Since our specifications are defined with a (total) function, there is exactly one related coalgebra and each variable has exactly one interpretation in a final coalgebra. This makes strong soundness and completeness results decidedly less satisfying—either the assumptions are true or false of the single model of the specification. By removing the totality restriction, we could get more meaningful strong soundness and completeness.

Another more practical advantage of partial specifications is the ability to detect equality even in circumstances where values of certain variables are not known or are irrelevant. For example, from

$$d(x) = 1 : \text{zip}(y, z) \qquad d(y) = 1 : \text{zip}(x, z)$$

we would like to be able to conclude  $\models x = y$  even if the value of  $z$  is not specified.

We implicitly used the fact that finitary functors have *flat* presentations. That is, each point in  $FX$  is an image of a point in  $H_\Sigma X$  and all equations necessary for the presentation are between flat terms. Flat signatures are not always the most natural though, sometimes one would like to use zip terms like  $0 : \text{zip}(1 : x, \text{zip}(y, y))$ , where we have three function symbols:  $\Sigma_1 = \{0 :, 1 : \}$  and  $\Sigma_2 = \{\text{zip}\}$ . Then some non-flat equations are necessary to capture all the truths of the system. How many more modifications are necessary to deal successfully with specifications in  $X \rightarrow T_\Sigma X$  instead of  $X \rightarrow H_\Sigma X$ ?

We are also interested in contexts beyond  $\text{Set}$ , particularly  $\text{Vect}$ . Milius [6] extended the expression calculi of Bonsangue et al. [4, 5] to vector space coalgebras of the functor  $FX = \mathbb{R} \times X$ , providing a sound and complete system for reasoning about stream circuits. We have some hope that by combining our approach with the general definition of signature from Kelly and Power [16] we might be able to devise a system usable in more categories than  $\text{Set}$ .

There are also a couple of generic questions suggested by this work. Here we utilized the quotient relationship  $\epsilon : H_\Sigma \twoheadrightarrow F$  to relate  $H_\Sigma$ - and  $F$ -bisimulations. Do other relationships between functors yield interesting interplay between their coalgebras? Additionally, many of the results here suggest a connection to bisimulation up to literature. How does bisimulation up to presentation fit into the theory of enhanced bisimulations?

## Acknowledgements

I owe thanks to Alexandra Silva for helpful conversations at WoLLIC at the start of this project and to Larry Moss for his expertise and encouragement throughout. Thanks also to the anonymous referees for their careful reading and thoughtful comments.

## References

- [1] Jiří Adámek, Stefan Milius, and Lawrence S Moss. On finitary functors and their presentations. In *Coalgebraic Methods in Computer Science*, pages 51–70. Springer, 2012.
- [2] Lawrence S Moss. Recursion and corecursion have the same equational logic. *Theoretical Computer Science*, 294(1):233–267, 2003.
- [3] Lawrence S Moss, Erik Wennstrom, and Glen T Whitney. A complete logical system for the equality of recursive terms for sets. In *Logic and Program Semantics*, pages 180–203. Springer, 2012.
- [4] Marcello Bonsangue, Jan Rutten, and Alexandra Silva. A Kleene theorem for polynomial coalgebras. In *Foundations of Software Science and Computational Structures*, pages 122–136. Springer, 2009.
- [5] Marcello Bonsangue, Jan Rutten, and Alexandra Silva. An algebra for Kripke polynomial coalgebras. In *Logic In Computer Science, 2009. LICS’09. 24th Annual IEEE Symposium on*, pages 49–58. IEEE, 2009.
- [6] Stefan Milius. A sound and complete calculus for finite stream circuits. In *Logic in Computer Science (LICS), 2010 25th Annual IEEE Symposium on*, pages 421–430. IEEE, 2010.
- [7] Alexandra M Silva. *Kleene coalgebra*. PhD thesis, CWI, 2010.
- [8] Alexandra M Silva, Marcello Maria Bonsangue, and Jan JMM Rutten. *Kleene coalgebras*. CWI. Software Engineering [SEN], 2010.
- [9] Jan JMM Rutten. Universal coalgebra: a theory of systems. *Theoretical computer science*, 249(1):3–80, 2000.
- [10] Saunders Mac Lane. *Categories for the working mathematician*, volume 5. Springer Science & Business Media, 1978.
- [11] Clemens Kupke and Jan JMM Rutten. On the final coalgebra of automatic sequences. 2011.

- [12] Clemens Grabmayer, Jorg Endrullis, Dimitri Hendriks, Jan Willem Klop, and Lawrence S Moss. Automatic sequences and zip-specifications. In *Proceedings of the 2012 27th Annual IEEE/ACM Symposium on Logic in Computer Science*, pages 335–344. IEEE Computer Society, 2012.
- [13] Jurriaan Rot, Filippo Bonchi, Marcello Bonsangue, Damien Pous, JJMM Rutten, and Alexandra Silva. Enhanced coalgebraic bisimulation. *Mathematical Structures in Computer Science (to appear, 2014)*, 2014.
- [14] Jirí Adámek, H Peter Gumm, and Vera Trnková. Presentation of set functors: a coalgebraic perspective. *Journal of Logic and Computation*, 20(5):991–1015, 2010.
- [15] Antonius J. C. Hurkens, Monica McArthur, Yiannis N Moschovakis, Lawrence S Moss, and Glen T Whitney. The logic of recursive equations. *The Journal of Symbolic Logic*, 63(02):451–478, 1998.
- [16] G Maxwell Kelly and A John Power. Adjunctions whose counits are coequalizers, and presentations of finitary enriched monads. *Journal of pure and applied algebra*, 89(1):163–179, 1993.