



# Coalgebraic Simulations and Congruences

H. Gumm, Mehdi Zarrad

► **To cite this version:**

H. Gumm, Mehdi Zarrad. Coalgebraic Simulations and Congruences. Marcello M. Bonsangue. 12th International Workshop on Coalgebraic Methods in Computer Science (CMCS), Apr 2014, Grenoble, France. Lecture Notes in Computer Science, LNCS-8446, pp.118-134, 2014, Coalgebraic Methods in Computer Science. <10.1007/978-3-662-44124-4\_7>. <hal-01408755>

**HAL Id: hal-01408755**

**<https://hal.inria.fr/hal-01408755>**

Submitted on 5 Dec 2016

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

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



Distributed under a Creative Commons Attribution 4.0 International License

# Coalgebraic Simulations and Congruences

H. Peter Gumm and Mehdi Zarrad

Philipps-Universität Marburg

**Abstract** In a recent article Gorín and Schröder[3] study  $\lambda$ -simulations of coalgebras and relate them to preservation of positive formulae. Their main results assume that  $\lambda$  is a set of monotonic predicate liftings and their proofs are set-theoretical. We give a different definition of simulation, called strong simulation, which has several advantages:

Our notion agrees with that of [3] in the presence of monotonicity, but it has the advantage, that it allows diagrammatic reasoning, so several results from the mentioned paper can be obtained by simple diagram chases. We clarify the role of  $\lambda$ -monotonicity by showing the equivalence of

- $\lambda$  is monotonic
- every simulation is strong
- every bisimulation is a (strong) simulation
- every  $F$ -congruence is a (strong) simulation.

We relate the notion to bisimulations and  $F$ -congruences - which are defined as pullbacks of homomorphisms. We show that

- if  $\lambda$  is a separating set, then each difunctional strong simulation is an  $F$ -congruence,
- if  $\lambda$  is monotonic, then the converse is true: if each difunctional strong simulation is an  $F$ -congruence, then  $\lambda$  is separating.

## 1 1. Introduction

Coalgebraic logic as introduced by D. Pattinson [6] and refined by L. Schröder [9], has been very successful in providing a common framework for quite a variety of modal logics, see for instance [2],[5], or [11]. In many cases, the type functor, used to model such coalgebras preserves weak pullbacks, so logical equivalence can be modeled by structural relations called bisimulations. Two states related by a bisimulation are equivalent. In a recent paper Gorín and Schröder have introduced a notion of  $\lambda$ -simulation, where  $\lambda$  is a (set of) predicate lifting(s). Their definition is set theoretical and their proofs are calculational. In all of their results they assumed that all predicate liftings are monotonic.

Here we offer a different notion of simulation, which we call strong simulation. The definition is amenable to diagrammatic reasoning, whose utility we show in a number of proofs. Moreover, we show that under the assumption of monotonicity our definition coincides with that of [3]. Since they used monotony as a general hypothesis in their work, their results could be proved as well with our definition. We relate our strong simulations to the notion of Aczel-Mendler bisimulation (called  $F$ -bisimulation) and to (generalized) congruences.

## 2 Basic notions and preparations

Given a binary relation  $R \subseteq A \times B$ , let  $R^- \subseteq B \times A$  be the converse relation. If  $S \subseteq B \times C$  is another relation, then  $R \circ S := \{(a, c) \mid \exists b \in B. aRb \wedge bSc\}$  is called the composition of  $R$  and  $S$ . Obviously,  $\circ$  is associative and  $(R \circ S)^- = S^- \circ R^-$ .

For  $R \subseteq A \times A$ , a relation *on* a set  $A$ , notice that  $R$  is transitive iff  $R \circ R \subseteq R$ . Let  $R^* \subseteq A \times A$  be the reflexive transitive closure of  $R$ . The smallest equivalence relation containing  $R$  is  $R^{eq} := (R \cup R^-)^*$ . It is well known that kernels of maps are equivalence relations, where for a map  $f : A \rightarrow B$  the kernel is defined as

$$\ker f := \{(a, a') \mid f(a) = f(a')\},$$

and conversely, any equivalence relation  $E \subseteq A \times A$  is the kernel of the projection map  $\pi_E : A \rightarrow A/E$ , sending each element  $a \in A$  to  $a/E$ , its equivalence class under  $E$ . With  $\Delta_A$  we denote the identity relation on a set  $A$ .

### 2.1 Difunctionality

*Difunctional* relations are generalizations of equivalence relations, for the case of relations  $R \subseteq A \times B$  between possibly different sets. Reflexivity, symmetry and transitivity make no sense for such relations, so a possible generalization is:

**Definition 1.** A relation  $R \subseteq A \times B$  is called *difunctional*, if it satisfies :

$$(a_1, b_1), (a_2, b_1), (a_2, b_2) \in R \implies (a_1, b_2) \in R.$$

Immediately from the definition we see ([7]):

**Lemma 1.**  $R$  is difunctional  $\iff R \circ R^- \circ R \subseteq R \iff R^- \circ R \circ R^- \subseteq R^- \iff R^-$  is difunctional. The difunctional closure of a relation  $R$  is obtained as  $R^d := R \circ (R^- \circ R)^* = (R \circ R^-)^* \circ R$ .

Each equivalence relation on  $A$  is obviously difunctional. More generally, let  $f : A \rightarrow C$  and  $g : B \rightarrow C$  be two maps then we define

$$\ker(f, g) := \{(a, b) \in A \times B \mid f(a) = g(b)\}.$$

It is easy to see that  $\ker(f, g)$  is a difunctional relation, and, in perfect analogy to the situation with equivalence relations, every difunctional relation arises that way:

**Lemma 2.** A relation  $R \subseteq A \times B$  is difunctional, if and only if there are maps  $f : A \rightarrow C$ ,  $g : B \rightarrow C$  with  $R = \ker(f, g)$ .

*Proof.* Let  $R \subseteq A \times B$  be difunctional. Let  $e_A : A \rightarrow A+B$  and  $e_B : B \rightarrow A+B$  be the canonical inclusions of  $A$  and  $B$  into their sum. On  $A+B$  define

$$\bar{R} := \{(e_A(x), e_B(y)) \mid (x, y) \in R\}.$$

Obviously,  $\bar{R}$  is difunctional, so

$$\bar{R} \circ \bar{R}^- \circ \bar{R} \subseteq \bar{R} \quad \text{and} \quad \bar{R}^- \circ \bar{R} \circ \bar{R}^- \subseteq \bar{R}^-.$$

Moreover,  $\bar{R} \circ \bar{R} = \emptyset = \bar{R}^- \circ \bar{R}^-$  by construction. Therefore,

$$E := \Delta_{A+B} \cup \bar{R} \cup \bar{R}^- \cup \bar{R} \circ \bar{R}^- \cup \bar{R}^- \circ \bar{R}$$

is an equivalence relation, since one easily calculates  $E \circ E \subseteq E$ . Notice that

$$(e_A[A] \times e_B[B]) \cap E = \bar{R}.$$

With the projection  $\pi_E : A + B \rightarrow (A + B)/E$  it is now easy to calculate:

$$\begin{aligned} (x, y) \in \ker(\pi_E \circ e_A, \pi_E \circ e_B) &\iff (e_A(x), e_B(y)) \in E \\ &\iff (e_A(x), e_B(y)) \in \bar{R} \\ &\iff (x, y) \in R. \end{aligned}$$

Thus,  $R = \ker(f, g)$  where  $f$  and  $g$  are constructed as the pushout of the projections  $\pi_A^R : R \rightarrow A$  and  $\pi_B^R : R \rightarrow B$ .

More generally, if  $f : A \rightarrow C$  and  $g : B \rightarrow D$ , then any difunctional relation  $R \subseteq C \times D$  gives rise to a difunctional relation  $\ker(f, g)_R := \{(a, b) \mid f(a) R g(b)\} \subseteq A \times B$ .

## 2.2 Directed diagrams

Each map  $\theta : A \rightarrow 2$ , where  $2 = \{0, 1\}$  is understood as an ordered set, is called a *predicate*. The *carrier* of predicate  $\theta : A \rightarrow 2$  is the subset

$$\llbracket \theta \rrbracket := \{a \in A \mid \theta(a) = 1\}$$

and conversely, every subset  $U \subseteq A$  arises as  $U = \llbracket \chi_U \rrbracket$  from its characteristic function  $\chi_U$ . We shall often use the same symbol for a predicate and its carrier, such as in  $\square : F(2) \rightarrow 2$  and  $\square \subseteq F(2)$ .

It is sometimes convenient to write  $a \models \theta$  rather than  $\theta(a) = 1$  or  $a \in \theta$ . Similarly,  $A \models \theta$  means that  $a \models \theta$  for each  $a \in A$ .

We say  $\theta \implies \psi$  provided  $\llbracket \theta \rrbracket \subseteq \llbracket \psi \rrbracket$ . It will be convenient to encode this diagrammatically, where the inclusion is indicated by an upwards arrow, as in

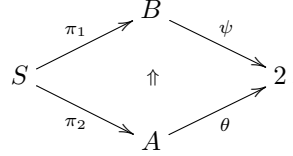
$$\begin{array}{ccc} & \psi & \\ & \curvearrowright & \\ A & & 2 \\ & \curvearrowleft & \\ & \theta & \end{array}$$

Establishing such a diagram amounts to showing that for any  $a \in A$ , taking the lower path in the diagram yields a result which is smaller than or equal to the result obtained by the upper path, i.e.  $\theta(a) \leq \psi(a)$  for all  $a \in A$ . We generalize this notation in the following way:

**Definition 2.** Given a relation  $S$  between sets  $A$  and  $B$  and predicates  $\theta : A \rightarrow 2$  and  $\psi : B \rightarrow 2$ , we introduce

$$\theta \xrightarrow{S} \psi \quad : \iff \quad \forall (x, y) \in S. (x \models \theta \implies y \models \psi) \quad (2.1)$$

which may be spelled as „ $\theta$  implies  $\psi$  modulo  $S$ “. With our above notation, we can visualize  $\theta \xrightarrow{S} \psi$  by the following „upwards-commuting“ diagram



Notice that with  $A = B$  and  $S = \Delta_A$ , we have  $\theta \implies \psi$  being the same as  $\theta \xrightarrow{\Delta_A} \psi$ .

Interpreting a relation  $S \subseteq A \times B$  as a map between the powersets  $S : \mathbb{P}(A) \rightarrow \mathbb{P}(B)$ , via  $S(U) := \{b \in B \mid \exists a \in U. (a, b) \in S\}$ , we could equivalently write:

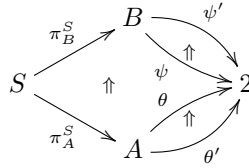
$$\theta \xrightarrow{S} \psi \iff S(\llbracket \theta \rrbracket) \subseteq \llbracket \psi \rrbracket.$$

This shows that this notation is closely related to the notation of Hoare triples, where the relation  $S$  would be given as the semantics of an imperative program. We can immediately gather a number of simple properties inspired by this association. These correspond to the rules of precondition strengthening/postcondition weakening and sequencing:

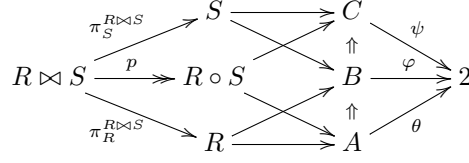
**Lemma 3.**

1.  $\theta' \subseteq \theta \xrightarrow{S} \psi$  implies  $\theta' \xrightarrow{S} \psi$
2.  $\theta \xrightarrow{S} \psi \subseteq \psi'$  implies  $\theta \xrightarrow{S} \psi'$
3.  $\theta \xrightarrow{R} \varphi$  and  $\varphi \xrightarrow{S} \psi$  implies  $\theta \xrightarrow{R \circ S} \psi$

*Proof.* The first two claims can be readily obtained by gluing diagrams where we use the obvious naming conventions for the projections of a relation to its components:



For the third claim, we note that if  $R \bowtie S$  is the pullback of  $\pi_B^R$  with  $\pi_B^S$ , then  $R \circ S$  is the image obtained by factoring the span  $(R \bowtie S, \pi_C^{R \bowtie S}, \pi_C^{R \bowtie S})$  into an epi followed by a mono source:



Explicitly, an upwards diagram chase, for instance in the right diagram, would be:

$$\begin{aligned}
\theta \circ \pi_A^{R \circ S} \circ p &= \theta \circ \pi_A^R \circ \pi_R^{R \bowtie S} \\
&\leq \varphi \circ \pi_B^R \circ \pi_R^{R \bowtie S} \\
&= \varphi \circ \pi_B^S \circ \pi_S^{R \bowtie S} \\
&\leq \psi \circ \pi_C^S \circ \pi_S^{R \bowtie S} \\
&= \psi \circ \pi_C^{R \circ S} \circ p.
\end{aligned}$$

Cancelling the epi  $p$  results in  $\theta \circ \pi_A^{R \circ S} \leq \psi \circ \pi_C^{R \circ S}$ .

### 3 Functors, Coalgebras and Bisimulations

Let  $F : Set \rightarrow Set$  be an endofunctor on the category of sets. We shall write  $F(X)$  for the action of  $F$  on an object  $X$  and  $Ff$  for the action of  $F$  on a map  $f$ .

Typical endofunctors describe set-theoretical constructions, such as *sets*, *lists*, *tuples*, *bags*, etc. In programming they include all generic collection classes such as `List<X>`, `Set<X>`, `Bag<X>` etc. The action of  $F$  on a map  $f : X \rightarrow Y$  is generically called `: map f`. It will be useful to keep the following visualization in mind:

- $F$  defines a type of „constructions“.
- Elements of  $F(X)$  are those „constructions“ whose elements are drawn from a set  $X$ ; we will call them  $X$  – *patterns*.
- Given a map  $f : X \rightarrow Y$ , the map  $Ff : F(X) \rightarrow F(Y)$  acts on an  $X$ -pattern  $p \in F(X)$  by replacing in  $p$  each  $x$  by  $f(x)$ .
- A pattern  $p \in F(X)$  is finite, if there is a subset  $\{x_1, \dots, x_n\} \subseteq X$  such that  $p \in F(\{x_1, \dots, x_n\})$ . In this case, we write  $p = p(x_1, \dots, x_n)$  and we let  $p(f(x_1), \dots, f(x_n))$  denote  $(Ff)p(x_1, \dots, x_n)$ .
- In particular, if  $\theta : X \rightarrow 2$  is a predicate, then  $F\theta$  acts on an element  $p \in F(X)$  by replacing in  $p$  each  $x$  by 1 if  $x \models \theta$  and by 0 otherwise.
- If  $p = p(x_1, \dots, x_n)$ , then  $(F\theta)p(x_1, \dots, x_n) = p(\theta(x_1), \dots, \theta(x_n))$  is called a *0 – 1 – pattern*.

If  $f : X \rightarrow Y$  is injective and  $X \neq \emptyset$ , then  $f$  is left-invertible, hence  $Ff$  is injective, too.  $F$  can always be modified just on the empty set and on empty mappings, so that it preserves injectivity for all mappings, including the empty one, see [13]. We therefore assume for the rest of this article, that  $F$  preserves all monos.

### 3.1 Coalgebras

**Definition 3.** An  $F$ -coalgebra  $\mathcal{A} = (A, \alpha)$  consists of a set  $A$  and a map  $\alpha : A \rightarrow F(A)$ .  $A$  is called the base set and  $\alpha$  the structure map. The functor  $F$  is called the type of coalgebra  $\mathcal{A}$ .

We shall keep  $F$  fixed and consider only coalgebras of that given type  $F$ .

**Definition 4.** A map  $\varphi : A \rightarrow B$  between two coalgebras  $\mathcal{A} = (A, \alpha)$  and  $\mathcal{B} = (B, \beta)$  is called a homomorphism, if  $\beta \circ \varphi = F\varphi \circ \alpha$ .

The functor properties immediately guarantee that the class of all  $F$ -coalgebras with homomorphisms as morphisms forms a category  $Set_F$ . The forgetful functor  $U : Set_F \rightarrow Set$  which associates with every coalgebra  $\mathcal{A}$  its underlying set  $A$  and with every homomorphism its underlying map is known to create and preserve colimits [8], so in particular the category  $Set_F$  is cocomplete and colimits have the same underlying set and mappings as the corresponding colimits in  $Set$ .

*Example 1.* Kripke frames are coalgebras of type  $\mathbb{P}$  where  $\mathbb{P}$  is the covariant powerset functor, acting on a map  $f : X \rightarrow Y$  as  $\mathbb{P}f : \mathbb{P}(X) \rightarrow \mathbb{P}(Y)$  where  $(\mathbb{P}f)(U) := f[U] := \{f(u) \mid u \in U\}$  for any  $U \in \mathbb{P}(X)$ .

Kripke structures come with a fixed set  $V$  of atomic properties, so they are modeled as coalgebras of type  $\mathbb{P}(-) \times \mathbb{P}(V)$ , where the second component is simply a constant. A coalgebra of type  $\mathbb{P}(-) \times \mathbb{P}(V)$  is therefore a base set  $A$  with a structure map  $\alpha : A \rightarrow \mathbb{P}(A) \times \mathbb{P}(V)$ . Its first component associates to a state  $a \in A$  the set of its successors  $succ_A(a) := (\pi_1 \circ \alpha)(a)$  and its second component yields the set of all atomic values  $val_A(a) := (\pi_2 \circ \alpha)(a)$  which are true for  $a$ .

Homomorphisms  $\varphi : A \rightarrow B$  between Kripke frames, resp. Kripke structures are also known as *bounded morphisms*. They are maps preserving and reflecting successors and atomic values in the following sense:  $\varphi[succ_A(a)] = succ_B(\varphi(a))$  and  $val_A(a) = val_B(\varphi(a))$ .

### 3.2 Bisimulations

In the structure theory of coalgebras, bisimulations play the role of compatible relations.

**Definition 5.** ([1]) A bisimulation between coalgebras  $\mathcal{A}$  and  $\mathcal{B}$  is a relation  $R \subseteq A \times B$  for which there exists a coalgebra structure  $\rho : R \rightarrow F(R)$  such that the projections  $\pi_A^R : R \rightarrow A$  and  $\pi_B^R : R \rightarrow B$  are homomorphisms.

Typical bisimulations are graphs of homomorphisms  $G(\varphi) := \{(a, \varphi(a)) \mid a \in A\}$ . In fact, a map  $f : A \rightarrow B$  is a homomorphism iff its graph is a bisimulation ([8]). If  $R \subseteq A \times B$  is a bisimulation between coalgebras  $\mathcal{A}$  and  $\mathcal{B}$ , then there could be several possible structure maps  $\rho : R \rightarrow F(R)$  establishing that  $R$  is a bisimulation.

The empty relation  $\emptyset \subseteq A \times B$  is always a bisimulation and (more generally) the union of bisimulations is a bisimulation, so that bisimulations between  $\mathcal{A}$  and  $\mathcal{B}$  form a complete lattice with largest element called  $\sim_{\mathcal{A},\mathcal{B}}$ .

The following proposition will be needed later in the proof of Theorem 3. It shows that bisimulations can be enlarged as long as the structure maps are not affected in the following sense:

**Proposition 1.** *Let  $\mathcal{A}_1$  and  $\mathcal{A}_2$  be coalgebras with corresponding structure maps  $\alpha_1$  and  $\alpha_2$ . Let  $R \subseteq \mathcal{A}_1 \times \mathcal{A}_2$  be a bisimulation and  $R'$  an enlargement i.e.  $R \subseteq R' \subseteq \ker \alpha_1 \circ R \circ \ker \alpha_2$ . Then  $R'$  is also a bisimulation.*

*Proof.*  $R$  is a bisimulation, so there exists a structure map  $\rho : R \rightarrow F(R)$  with  $\alpha_i \circ \pi_i^R = F\pi_i^R \circ \rho$ . Let  $\iota : R \rightarrow R'$  be the inclusion map, then clearly  $\pi_i^R = \pi_i^{R'} \circ \iota$ . By assumption, we find for every  $(x', y') \in R'$  a pair  $(x, y) \in R$  such that  $\alpha_1(x) = \alpha_1(x')$  and  $\alpha_2(y) = \alpha_2(y')$ . The axiom of choice provides for a map  $\mu : R' \rightarrow R$  satisfying

$$\alpha_i \circ \pi_i^{R'} \circ \iota \circ \mu = \alpha_i \circ \pi_i^R.$$

We now define  $\rho' : R' \rightarrow F(R')$  by  $\rho' := F\iota \circ \rho \circ \mu$ .

$$\begin{array}{ccccc}
 & & R' & & \\
 & \swarrow \mu & \downarrow \pi_i^{R'} & \searrow \pi_i^{R'} & \\
 R & \xrightarrow{\iota} & R' & \xrightarrow{\pi_i^{R'}} & A_i \\
 \downarrow \rho & & \downarrow \rho' & & \downarrow \alpha_i \\
 F(R) & \xrightarrow{F\mu} & F(R') & \xrightarrow{F\pi_i^{R'}} & F(A_i) \\
 & \swarrow F\iota & \downarrow F\pi_i^R & \searrow F\pi_i^R & \\
 & & F(R) & & 
 \end{array}$$

The rest is a simple calculation.

**Corollary 1.** *Let  $\mathcal{A} = (A, \alpha)$  be a coalgebra, then every reflexive relation  $R \subseteq \ker \alpha$  is a bisimulation.*

*Proof.* Since  $\Delta \subseteq A$  is always a bisimulation, we have  $\Delta \subseteq R \subseteq \ker \alpha = \ker \alpha \circ \ker \alpha = \ker \alpha \circ \Delta \circ \ker \alpha$ , because  $\ker \alpha$  is transitive.

### 3.3 Predicate liftings and boxes

We denote the contravariant powerset functor by  $2^-$ . Thus  $2^X$  is the set of all subsets of  $X$  and a map  $f : X \rightarrow Y$  induces a map  $2^f : 2^Y \rightarrow 2^X$  via  $2^f(V) := f^{-1}[V]$ . If we consider the elements of  $2^Y$  as predicates  $\tau : Y \rightarrow 2$ , we can write  $2^f(\tau) = \tau \circ f$ , or  $2^f = (-) \circ f$ .

The classical Kripke style modal logic introduces formulae expressing properties holding for all successors of a point  $x$ . If  $\varphi$  is a state formula then  $\Box\varphi$  holds at  $x$  if  $\varphi$  holds for each successor  $x'$  of  $x$ . The set of all successors of a point  $x$  is  $\alpha(x) \in \mathbb{P}(X)$ , in the case of Kripke frames. Thus  $\Box$  can be understood as lifting



a property  $\varphi$  from the base set  $A$  to a property  $\lambda_A(\varphi) \subseteq \mathbb{P}(A)$ , so  $x \models \Box\varphi$  iff  $\alpha(a)$  satisfies the lifted property  $\lambda_A(\varphi)$ . Generalizing this observation, Pattinson [6] introduced predicate liftings  $\lambda_A : 2^A \rightarrow 2^{F(A)}$  as natural transformations between the contravariant powerset functors  $2^{(-)}$  and  $2^{F(-)}$ .

**Definition 6.** A predicate lifting  $\lambda$  for  $F$  is a natural transformation  $\lambda : 2^- \rightarrow 2^{F(-)}$  where the latter is the composition of the functor  $F$  with  $2^-$ . For each  $X$  denote by  $\lambda_X$  its  $X$ -component  $\lambda_X : 2^X \rightarrow 2^{F(X)}$ .

The idea is that every property for elements of a set  $X$  is transformed to a property for elements of  $F(X)$ .

By the Yoneda lemma, such a natural transformation  $\lambda$  is uniquely determined by the action of  $\lambda_2$  on the input  $id_2$  where  $\llbracket id_2 \rrbracket = \{1\} \subseteq 2$ , i.e. by  $\lambda_2(id_2) : F(2) \rightarrow 2$ , which is a predicate on  $F(2)$ . This was observed in [9]. We shall from now on write  $[\lambda]$  or simply  $\Box$ , if  $\lambda$  is understood, for this predicate.

Conversely, given a predicate  $\Box : F(2) \rightarrow 2$  on  $F(2)$ , then  $\theta \mapsto \Box \circ F\theta$  defines a predicate transformer, and it is easy to see that  $id_2$  is sent to  $\Box$  again.

Intuitively, we think of  $\Box \subseteq F(2)$  as a selection of 0 – 1 – patterns. The map  $\lambda_A$  of the corresponding predicate transformer  $\lambda$ , when applied to  $\theta \in 2^A$  takes an  $A$ -pattern  $p(a_1, \dots, a_n) \in F(A)$  to 1 if  $p(\theta(a_1), \dots, \theta(a_n)) \in \Box$ , and to 0 otherwise.

In this paper we prefer to deal with predicates  $\Box : F(2) \rightarrow 2$  rather than with predicate transformers  $\lambda : 2^{(-)} \rightarrow 2^{F(-)}$ . Ignoring for a moment the map  $\alpha$ , the following figure visualizes the translation between these two views.

$$\begin{array}{ccccc}
 A & \xrightarrow{\theta} & 2 & \xrightarrow{id} & 2 \\
 \vdots & & & & \\
 \alpha \downarrow & & & & \\
 F(A) & \xrightarrow{F\theta} & F(2) & \xrightarrow[\lambda_2(id_2)]{\Box} & 2 \\
 & \searrow & & \nearrow & \\
 & & & \lambda_A(\theta) & 
 \end{array}$$

Let us now consider  $F$ -coalgebras  $\mathcal{A} = (A, \alpha)$ , where  $\alpha : A \rightarrow F(A)$  is the *structure map*. Every predicate transformer, i.e. every predicate  $\Box$  on  $F(2)$  defines a modality.

**Definition 7.** Given a predicate  $\theta$  on  $\mathcal{A} = (A, \alpha)$ , denote by  $\Box\theta$  the predicate  $\Box \circ F\theta \circ \alpha$ , that is for any  $a \in A$  we define

$$a \models \Box\theta : \iff (\Box \circ F\theta \circ \alpha)(a) = 1.$$

### 3.4 Coalgebraic modal logic

Given any choice of predicate liftings, equivalently, any choice of boxes  $\Box_i : F(2) \rightarrow 2$ ,  $i \in I$ , we obtain a logic  $\mathcal{L}$  (see [6]) whose formulae are defined inductively by

$$\varphi ::= \top \mid \varphi_1 \vee \varphi_2 \mid \varphi_1 \wedge \varphi_2 \mid \neg\varphi \mid \Box_i\varphi \text{ for each } i \in I$$

A formula is called positive, if it has no occurrence of  $\neg$ .

Given a coalgebra  $\mathcal{A} = (A, \alpha)$  each formula defines a predicate  $\varphi_{\mathcal{A}} : A \rightarrow 2$ , where the propositional connectors have their obvious interpretation and  $(\Box_i \varphi)_{\mathcal{A}} := \Box_i \circ (F\varphi_{\mathcal{A}}) \circ \alpha$ , which is short for saying

$$a \models \Box_i \varphi : \iff (F\varphi_{\mathcal{A}} \circ \alpha)(a) \in \Box_i.$$

## 4 Simulations

Given a predicate lifting  $\lambda$ , a  $\lambda$ -simulation  $S$  between coalgebras  $\mathcal{A} = (A, \alpha)$  and  $\mathcal{B} = (B, \beta)$  was defined in [3] as a relation  $S \subseteq A \times B$  such that for any  $(x, y) \in S$  and any predicate  $\theta : A \rightarrow 2$  one has  $\alpha(x) \models \lambda_A(\theta) \implies \beta(y) \models \lambda_B(S[\theta])$ , where  $S[\theta]$  is defined as  $b \models S[\theta] : \iff \exists a \in A. (aSb \wedge a \models \theta)$ . Most of the results in [3] assume that  $\lambda$  is monotonic, a notion to be discussed in Sect. 4.2. Amongst other things, for instance they prove:

- if  $\lambda$  is monotonic then bisimulations are  $\lambda$ -simulations
- if  $\lambda$  is monotonic, then each  $\lambda$ -simulation preserves positive formulae.

The proofs, in each case, are set theoretical, so it is difficult to see how the notions and results could possibly be lifted to situations beyond set-theoretical categories. Therefore, we introduce a new definition of „strong“ simulation which has the advantage that

- proofs are diagrammatical
- monotonicity need not be assumed.

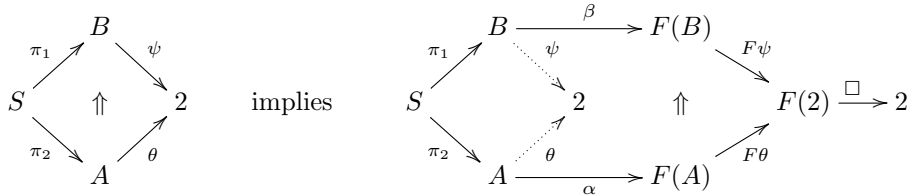
For notational reasons, we shall from now on fix a certain  $\Box$  and define simulations relative to that  $\Box$ . Thus a „simulation“ is the same as a  $\lambda$ -simulation from [3] with  $\lambda$  the predicate lifting defined by  $\Box$ . Next we shall define our new notion of „strong simulation“. It will turn out that monotonicity is the property relating simulations with strong simulations, see theorem 2 below.

### 4.1 Strong simulations

A *strong simulation* between coalgebras  $\mathcal{A} = (A, \alpha)$  and  $\mathcal{B} = (B, \beta)$  is a relation  $S \subseteq A \times B$  such that for any predicates  $\theta : A \rightarrow 2$  and  $\psi : B \rightarrow 2$  we have

$$\theta \xRightarrow{S} \psi \text{ implies } \Box\theta \xRightarrow{S} \Box\psi.$$

Diagrammatically:



Clearly, every strong simulation is a simulation. This is because the diagram in the premise, above, is trivially satisfied with  $\psi = S[\theta]$ .

**Lemma 4.** *Strong simulations are closed under unions and relational composition, i.e. if  $R \subseteq A \times B$  and  $S \subseteq B \times C$  are strong simulations, then so is  $R \circ S \subseteq A \times C$ .*

*Proof.* Closure under unions is easily checked. For closure under relational composition, let  $\theta \xrightarrow{R \circ S} \psi$  be given. Obviously, then  $\theta \xrightarrow{R} R[\theta]$  and  $R[\theta] \xrightarrow{S} \psi$ . Assuming that  $R$  and  $S$  are simulations, we obtain  $\Box\theta \xrightarrow{R} \Box R[\theta]$  as well as  $\Box R[\theta] \xrightarrow{S} \Box\psi$ , so Lemma 3 yields  $\Box\theta \xrightarrow{R \circ S} \Box\psi$ .

Simulations have a preferred direction. This is emphasized by the following logical fact:

**Theorem 1.** *Strong simulations preserve positive formulae.*

*Proof.* Let  $S$  be a strong simulation between coalgebras  $\mathcal{A}$  and  $\mathcal{B}$ , and  $(x, y) \in S$ . By structural induction, we show that for any positive formula  $\phi$  we have:  $x \models \phi \implies y \models \phi$ , that is we need to show  $\phi_{\mathcal{A}} \xrightarrow{S} \phi_{\mathcal{B}}$ . The only interesting case is when  $\phi = \Box\psi$  with  $\psi$  another positive formula. Let  $\psi_{\mathcal{A}}$ , resp  $\psi_{\mathcal{B}}$  be the predicates defined by  $\psi$  in  $\mathcal{A}$ , resp  $\mathcal{B}$ . By assumption then,  $\psi_{\mathcal{A}} \xrightarrow{S} \psi_{\mathcal{B}}$ , whence the definition of simulation yields  $\Box\psi_{\mathcal{A}} \xrightarrow{S} \Box\psi_{\mathcal{B}}$ , hence  $(\Box\psi)_{\mathcal{A}} \xrightarrow{S} (\Box\psi)_{\mathcal{B}}$ .

By a (strong) *bidirectional simulation* we understand a (strong) simulation  $S$  for which  $S^{-}$  is also a simulation. We must be careful not to confuse this with the notion of *bisimulation*.

From Lemmas 1 and 4 we obtain:

**Lemma 5.** *Let  $(S_i)_{i \in I}$  be a family of bidirectional simulations, then their difunctional closure is again a bidirectional simulation.*

## 4.2 Monotonicity

**Definition 8.** *A predicate lifting  $\lambda$  is called monotonic, if for all sets  $U, V, A$  with  $U \subseteq V \subseteq A$  one has  $\lambda_A(U) \subseteq \lambda_A(V)$ . We say that  $\Box : F(2) \rightarrow 2$  is monotonic, if the predicate lifting given by  $\Box$  is monotonic.*

We get the following characterization:

**Lemma 6.**  *$\Box : F(2) \rightarrow 2$  is monotonic, iff for any  $A$  and any predicates  $\theta, \psi$  on  $A$  with  $\theta \implies \psi$ , we obtain  $\Box \circ F\theta \implies \Box \circ F\psi$ .*

*Proof.* Suppose  $\lambda_A = \Box \circ F(-)$  is monotonic,  $\theta \implies \psi$  and  $\Box \circ F\theta = 1$ , that is  $\lambda_A(\theta) = 1$ . By monotonicity,  $\lambda_A(\psi) = 1$ , i.e.  $\Box \circ F\psi = 1$ . Conversely, assume  $U \subseteq V \subseteq A$  and  $u \in \lambda_A(U)$ , where  $\lambda_A(U) = \llbracket \Box \circ F\chi_U \rrbracket$ . Then  $\chi_U \implies \chi_V$  and  $(\Box \circ F\chi_U)(u) = 1$  whence by assumption  $(\Box \circ F\chi_V)(u) = 1$ , meaning  $u \in \lambda_A(V)$ . Thus  $\lambda_A$  is monotonic.

Graphically, monotonicity can be represented as

$$\begin{array}{ccc}
 \begin{array}{ccc}
 & \psi & \\
 A & \curvearrowright & 2 \\
 & \uparrow & \\
 & \theta & \\
 & \curvearrowleft & \\
 & & 
 \end{array}
 & \Longrightarrow &
 \begin{array}{ccc}
 & F\psi & \\
 F(A) & \curvearrowright & F(2) \xrightarrow{\square} 2 \\
 & \uparrow & \\
 & F\theta & \\
 & \curvearrowleft & \\
 & & 
 \end{array}
 \end{array}$$

The following observation was independently found by L. Schröder and appears in the journal version [10] of [9]. With our diagrammatic notation its proof becomes almost trivial:

**Lemma 7.**  $\square$  is monotonic if and only if for every ternary pattern  $p(x, y, z)$  we have that

$$p(1, 0, 0) \in \square \Longrightarrow p(1, 1, 0) \in \square.$$

*Proof.* When  $\theta \Longrightarrow \psi$ , we can obtain a joint factorization as  $\theta = \chi_{\{x\}} \circ f$  and  $\psi = \chi_{\{x, y\}} \circ f$ . Thus the above definition of monotonicity reduces to the following implication:

$$\begin{array}{ccc}
 \begin{array}{ccc}
 & \psi & \\
 A & \xrightarrow{f} \{x, y, z\} & \curvearrowright 2 \\
 & \uparrow \chi_{\{x, y\}} & \\
 & \uparrow \chi_{\{x\}} & \\
 & \theta & 
 \end{array}
 & \Longrightarrow &
 \begin{array}{ccc}
 & \psi & \\
 F(A) & \xrightarrow{Ff} F(\{x, y, z\}) & \curvearrowright F(2) \xrightarrow{\square} 2 \\
 & \uparrow F\chi_{\{x, y\}} & \\
 & \uparrow F\chi_{\{x\}} & \\
 & \theta & 
 \end{array}
 \end{array}$$

The outer diagrams are upward commutative iff the inner ones are. The one in the premise is automatically upward commutative. Therefore,  $\square$  is monotonic, if and only if the inner diagram on the right is upwards commutative.

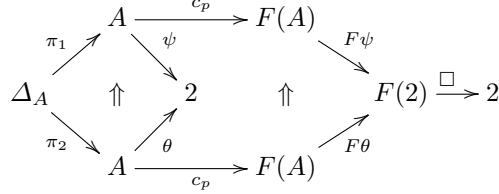
This means that monotonicity needs only be checked for  $\theta = \chi_{\{x\}}$  and  $\psi = \chi_{\{x, y\}}$ , which translates immediately into the statement  $p(1, 0, 0) \in \square \Longrightarrow p(1, 1, 0) \in \square$  for each  $p \in F(\{x, y, z\})$ .

**Theorem 2.**  $\square$  is monotonic iff each simulation is strong

*Proof.* Suppose that  $\square$  is monotonic and let  $S$  be a simulation between coalgebras  $\mathcal{A} = (A, \alpha)$  and  $\mathcal{B} = (B, \beta)$ . Suppose  $\theta \xrightarrow{S} \psi$ , then  $S[\theta] \leq \psi$  as shown in the left part of the following figure, where the left inner square trivially commutes. Since  $S$  is a simulation we get upwards commutativity of the outer figure with  $FS[\theta]$  instead of  $F\psi$ . Using monotonicity, we get upwards commutativity of the right upper figure and therefore of the whole diagram:

$$\begin{array}{ccccc}
 & & B & \xrightarrow{\beta} & F(B) & \xrightarrow{F\psi} & F(2) & \xrightarrow{\square} & 2 \\
 & \pi_B^S \nearrow & \downarrow \psi & & \downarrow F\psi & & \downarrow F\psi & & \\
 S & & \uparrow S[\theta] & & \uparrow FS[\theta] & & \uparrow FS[\theta] & & \\
 & \pi_A^S \searrow & A & \xrightarrow{\alpha} & F(A) & \xrightarrow{F\theta} & F(2) & \xrightarrow{\square} & 2 \\
 & & \downarrow \theta & & \downarrow F\theta & & \downarrow F\theta & & 
 \end{array}$$

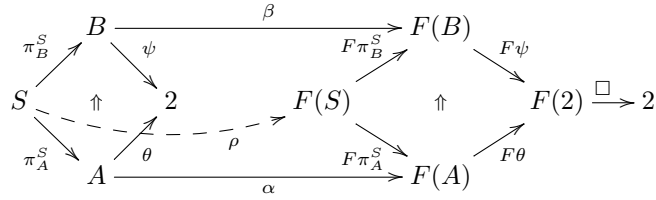
For the converse, consider the identity relation  $\Delta_A$  on  $A$ , which is obviously a simulation, hence it is a strong simulation by assumption. Given any  $p \in F(A)$  and  $\theta \leq \psi : A \rightarrow 2$  we choose the constant coalgebra structure  $c_p : A \rightarrow F(A)$ . Since the left square is upwards commuting, so must be the outer figure. This readily translates into  $\square$  being monotonic.



**Theorem 3.** *The following are equivalent:*

1.  $\square$  is monotonic
2. each bisimulation is a simulation
3. each bisimulation is a strong simulation

*Proof.* (1.  $\rightarrow$  3.) Suppose  $\square$  is monotonic and  $S \subseteq A \times B$  is a bisimulation between coalgebras  $\mathcal{A} = (A, \alpha)$  and  $\mathcal{B} = (B, \beta)$ . Given  $\theta \xrightarrow{S} \psi$ , the left square is upward commuting. Since  $\square$  is monotonic, applying  $F$  makes the right hand square (followed by  $\square$ ) upward commuting, too.



Inserting the bisimulation structure  $\rho$  into the picture, an upward diagram chase yields that the outer diagram is upward commuting, too:

$$\begin{aligned}
\square \circ F\theta \circ \alpha \circ \pi_A^S &= \square \circ F\theta \circ F\pi_A^S \circ \rho \\
&\leq \square \circ F\psi \circ F\pi_B^S \circ \rho \\
&= \square \circ F\psi \circ \beta \circ \pi_B^S
\end{aligned}$$

which means that  $S$  is a strong simulation.

(3  $\rightarrow$  2) being trivial, we prove (2  $\rightarrow$  1): By Lemma 7, we need to check monotonicity only for  $A = \{x, y, z\}$ ,  $\theta = \chi_{\{x\}}$  and  $\psi = \chi_{\{x, y\}}$ . Given  $p \in F(A)$  with  $p(1, 0, 0) \in \square$ , i.e.  $(\square \circ F\theta)(p) = 1$ , define a coalgebra  $\mathcal{A}_p$  on  $A$  with constant structure map  $c_p$ . By proposition 1,  $R := \Delta_A \cup \{(x, y), (y, x)\}$  is a bisimulation on

$\mathcal{A}_p$ , and  $\psi = R[\theta]$ . By hypothesis,  $R$  is a simulation, so  $\square \circ F\theta \circ c_p \xrightarrow{B} \square \circ F\psi \circ c_p$ , in particular,

$$\begin{aligned} (\square \circ F\psi)(p) &= (\square \circ F\psi \circ c_p \circ \pi_2)(x, x) \\ &\geq (\square \circ F\theta \circ c_p \circ \pi_1)(x, x) \\ &= (\square \circ F\theta)(p) \\ &= 1 \end{aligned}$$

i.e.  $p(1, 1, 0) \in \square$ , as can be read from the following diagram:

$$\begin{array}{ccccc} & & A & \xrightarrow{c_p} & F(A) \\ & \nearrow^{\pi_2} & \searrow^{\psi} & & \searrow^{F\psi} \\ R & & & & F(2) \xrightarrow{\square} 2. \\ & \nwarrow_{\pi_1} & \uparrow & & \uparrow \\ & & 2 & & \\ & & \nearrow^{\theta} & & \nearrow^{F\theta} \\ & & A & \xrightarrow{c_p} & F(A) \end{array}$$

## 5 Congruences and separability

### 5.1 Congruences

In classical examples of coalgebras, such as Kripke structures, deterministic and nondeterministic automata, etc., observational equivalence is definable via bisimulations. The reason is that the corresponding type functors preserve weak pullbacks (see [4]). This in turn has many structural consequences. In particular the largest bisimulation is always the same as the largest congruence relation, where a congruence is defined as the kernel of a homomorphism. Thus a congruence is a relation *on a single* coalgebra. Since we want to study relations between different coalgebras, we have to widen the notion of congruence and therefore introduce the notion of  $F$ -congruence. This notion has been studied by Sam Staton under the name *kernel bisimulation* [12]:

**Definition 9.** *An  $F$ -congruence  $\theta$  between coalgebras  $\mathcal{A}$  and  $\mathcal{B}$  is the pullback of two homomorphisms  $\varphi : \mathcal{A} \rightarrow \mathcal{C}$  and  $\psi : \mathcal{B} \rightarrow \mathcal{C}$ :*

$$\theta = \ker(\varphi, \psi).$$

**Theorem 4.** *The following are equivalent:*

1.  $\square$  is monotonic
2. each congruence is a simulation
3. each  $F$ -congruence is a strong simulation.

*Proof.* (1.→3.): An  $F$ -congruence  $\theta = \ker(\varphi, \psi)$  can be obtained as a composition of relations:  $\theta = G(\varphi) \circ G(\psi)^-$  where  $G(\varphi)$  and  $G(\psi)$  are the graphs of  $\varphi$  and  $\psi$ . The graphs of homomorphisms are bisimulations ([8]) and the converse of a bisimulation is a bisimulation. Assuming monotonicity of  $\square$ , Theorem 3 tells us that they are strong simulations. By Lemma 4, their composition is a strong simulation. In particular, each congruence is a simulation, too. (3.→2) is of course trivial, since each congruence is an  $F$ -congruence and each strong simulation is a simulation.

For (2.→1.), assuming that each congruence is a simulation, we can reuse the proof of (3→2) in Theorem 3. This time, we only need to observe that  $R$  happens to be a congruence relation, since it is the kernel of the obvious homomorphism from  $\mathcal{A}_p = \mathcal{A}_{p(x,y,z)}$  to the constant coalgebra  $\mathcal{A}_{p(x,x,z)}$  on  $\{x, z\}$ .

## 5.2 Separability

In this section we need to work with a family of boxes  $(\square_i)_{i \in I}$ . Such is usually required in order to render coalgebraic modal logic expressive. Separability is usually expressed for the functor and for the boxes separately. A functor is called 2-separable, if for any  $X$  and any  $p, q \in F(X)$  with  $p \neq q$  there is a predicate  $\phi : X \rightarrow 2$  such  $(F\phi)(p) \neq (F\phi)(q)$ . Next, we call a family  $(\square_i)_{i \in I}$  of predicate liftings *separating*, if the functor  $F$  is 2-separating and the predicates  $\square_i : F(2) \rightarrow 2$  combined with the unary boolean operations  $\theta : 2 \rightarrow 2$  form a mono-source. We can equivalently define this as follows:

**Definition 10.**  $(\square_i)_{i \in I}$  is separating if

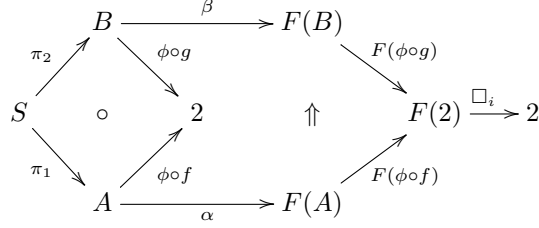
$$\forall p \neq q \in F(X). \exists \phi : X \rightarrow 2. \exists i \in I. (\square_i \circ F\phi)(p) \neq (\square_i \circ F\phi)(q).$$

**Theorem 5.** If  $(\square_i)_{i \in I}$  is separating then every difunctional bidirectional strong simulation is an  $F$ -congruence.

*Proof.* Let  $S$  be a difunctional strong simulation between coalgebras  $\mathcal{A} = (A, \alpha)$  and  $\mathcal{B} = (B, \beta)$  and  $\pi_1, \pi_2$  the projections of  $S$ . Form the pushout  $(P, f : A \rightarrow P, g : B \rightarrow P)$  of  $(S, \pi_1, \pi_2)$  in  $Set$ . Since  $S$  is difunctional,  $(S, \pi_1, \pi_2)$  is a pullback of  $f$  and  $g$  in  $Set$ . It suffices to show that there exists a coalgebra structure on  $P$  so that  $f$  and  $g$  are homomorphisms. We obtain such a coalgebra structure if we can show that  $(FP, Ff \circ \alpha, Fg \circ \beta)$  is a competitor of the pushout  $(P, f, g)$  in  $Set$ . For this it remains to show :  $Ff \circ \alpha \circ \pi_1 = Fg \circ \beta \circ \pi_2$ .

$$\begin{array}{ccccc}
 & & A & \xrightarrow{f} & P & \xrightarrow{\phi} & 2 \\
 & \nearrow \pi_1 & \downarrow & & \downarrow & & \\
 S & & & & & & \\
 & \searrow \pi_2 & B & \xrightarrow{g} & P & & \\
 & & \downarrow & & \downarrow & & \\
 & & F(A) & \xrightarrow{Ff} & F(P) & \xrightarrow{F\phi} & F(2) \xrightarrow{\square_i} 2 \\
 & & \downarrow & & \downarrow & & \\
 & & F(B) & & & & 
 \end{array}$$

Let  $(x, y) \in S$ . As  $(\sqcap_i)_{i \in I}$  is separating, it is enough to show that for each  $i \in I$  and each  $\phi : P \rightarrow 2$  we have  $Ff \circ \alpha(x) \models \sqcap_i \theta \iff Fg \circ \beta(y) \models \sqcap_i \theta$ . This we can read from the following diagram :

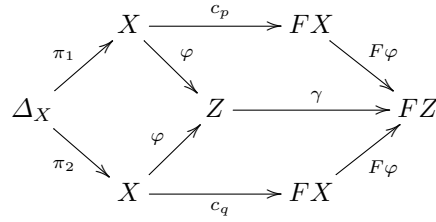


The left square in the diagram commutes, since  $(P, f, g)$  is a pushout, in particular it is upward commuting.  $S$  being a strong simulation, we obtain  $Ff \circ \alpha(x) \models \sqcap \theta \implies Fg \circ \beta(y) \models \sqcap \theta$ . Since  $S^-$  is a strong simulation, too, we similarly have  $Fg \circ \beta(y) \models \sqcap \theta \implies Ff \circ \alpha(x) \models \sqcap \theta$ .

**Theorem 6.** *If each difunctional simulation is an  $F$ -congruence, then  $(\sqcap_i)_{i \in I}$  is separating.*

*Proof.* Assume  $p, q \in FX$  such that  $p \models \sqcap_i \theta \iff q \models \sqcap_i \theta$  for each  $i \in I$  and each  $\theta : X \rightarrow 2$ . We must show  $p = q$ .

*Case 1.*  $X \neq \emptyset$ : On the set  $X$  define  $F$ -coalgebras  $\mathcal{X}_p = (X, c_p)$  and  $\mathcal{X}_q = (X, c_q)$ , where  $c_p$ , resp.  $c_q$ , are constant maps with value  $p$ , resp.  $q$ . Notice that the assumption is then equivalent to saying that  $\Delta_X$  is a (difunctional) simulation (with respect to each  $\sqcap_i$ ) between  $\mathcal{X}_p$  and  $\mathcal{X}_q$ . Therefore, by the theorem's premise,  $\Delta_X$  is an  $F$ -congruence. Consequently, there must be homomorphisms  $\varphi : \mathcal{X}_p \rightarrow \mathcal{Z} = (Z, \gamma)$  and  $\psi : \mathcal{X}_q \rightarrow \mathcal{Z}$  with  $\Delta_X = Pb(\varphi, \psi)$ . This immediately yields  $\varphi = \psi$  and  $\varphi$  injective.



The above diagram commutes, since  $\varphi$  is a homomorphism, so  $(F\varphi)(p) = (F\varphi \circ c_p \circ \pi_1)(x) = (F\varphi \circ c_q \circ \pi_2)(x) = (F\varphi)(q)$ . Therefore  $p = q$  as required.

*Case 2.*  $X = \emptyset$ : According to our general assumption,  $F\iota : F\emptyset \rightarrow F1$  is injective. Thus in order to separate  $p, q \in F\emptyset$ , it is enough to separate  $(F\iota)(p) \in F(1)$  from  $(F\iota)(q) \in F(1)$  which is possible due to the previous case.



**Corollary 2.** *If  $\square$  is monotonic and separating then every difunctional simulation is an  $F$ -congruence.*

As a further corollary, we obtain a converse to another result found in [3].

**Corollary 3.** *Let  $(\square_i)_{i \in I}$  be monotonic. Then  $(\square_i)_{i \in I}$  are separating and  $F$  weakly preserves pullbacks if and only if each difunctional simulation is an  $F$ -bisimulation.*

*Proof.* The direction from left to right is from [3]. For the converse, suppose that each difunctional simulation is an  $F$ -bisimulation. Then by monotony each  $F$ -congruence is an  $F$ -bisimulation. This is the same as saying that  $F$  weakly preserves pullbacks. Similarly, every difunctional simulation is an  $F$ -congruence, hence by the above proposition,  $(\square_i)_{i \in I}$  is separating.

## 6 Conclusion and further work

We have given a new definition of coalgebraic simulation, which has the advantage to be amenable to diagrammatic reasoning. We have demonstrated its use with a number of results and related our definition to that of Gorín and Schröder in [3]. In the case where our boxes (respectively predicate liftings) are monotonic, a general assumption in the paper [3], our definition agrees with that of the authors. We have related our simulations to 2-dimensional congruences (so called  $F$ -congruences). We suspect that the set of all  $F$ -congruences between fixed coalgebras  $\mathcal{A}$  and  $\mathcal{B}$  forms a complete lattice with the natural ordering. However we were only able to show it under the additional assumption that there exists a set of separating monotonic boxes  $(\square_i)_{i \in I}$ . In that case,  $F$ -congruences are bidirectional simulations and their supremum is given by difunctional closure. We leave it open whether the existence of a separating set  $(\square_i)_{i \in I}$  is needed.

## References

1. P. Aczel and N. Mendler. A final coalgebra theorem. In D.H. Pitt et al, editor, *Proceedings category theory and computer science*, Lecture Notes in Computer Science, pages 357–365. Springer, 1989.
2. C. Cirstea, A. Kurz, D. Pattinson, L. Schröder, and Y. Venema. Modal logics are coalgebraic. In *BCS Int. Acad. Conf.*, pages 128–140, 2008.
3. D. Gorín and L. Schröder. Simulations and bisimulations for coalgebraic modal logics. In R. Heckel and S. Milius, editors, *CALCO '13*, Lecture Notes in Computer Science. Springer, 2013.
4. H.P. Gumm and T. Schröder. Types and coalgebraic structure. *Algebra Universalis*, 53:229–252, 2005.
5. R.S.R. Myers, D. Pattinson, and L. Schröder. Coalgebraic hybrid logic. In *FOS-SACS*, pages 137–151, 2009.
6. D. Pattinson. Coalgebraic modal logic: soundness, completeness and decidability of local consequence. *Theoretical Computer Science*, 309(2-3):177–193, 2003.

7. J. Rignuet. Relations binaires, fermetures, correspondances de Galois. *Bulletin de la Société Mathématique de France*, 76:114–155, 1948.
8. J.J.M.M. Rutten. Universal coalgebra: a theory of systems. *Theoretical Computer Science*, (249):3–80, 2000.
9. L. Schröder. Expressivity of coalgebraic modal logic: The limits and beyond. In R. Amadio, editor, *Foundation of Software Science and Comp. Structures*, volume 3441 of *Lecture Notes in Computer Science*.
10. L. Schröder. Expressivity of coalgebraic modal logic: The limits and beyond. *Theoretical Computer Science*, 390(2-3):230–247, 2008.
11. L. Schröder and D. Pattinson. Coalgebraic correspondence theory. In *FOSSACS*, pages 328–342, 2010.
12. S. Staton. Relating coalgebraic notions of bisimulation. *Logical Methods in Computer Science*, 7(1), 2011.
13. V. Trnková. Some properties of set functors. *Comm. Math. Univ. Carolinae*, (10,2):323–352, 1969.