Bisimilar States in Uncertain Structures

Jurriaan Rot
Radboud University, Nijmegen, The Netherlands

Thorsten Wißmann
Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany
Radboud University, Nijmegen, The Netherlands

Abstract

We provide a categorical notion called uncertain bisimilarity, which allows to reason about bisimilarity in combination with a lack of knowledge about the involved systems. Such uncertainty arises naturally in automata learning algorithms, where one investigates whether two observed behaviours come from the same internal state of a black-box system that can not be transparently inspected. We model this uncertainty as a set functor equipped with a partial order which describes possible future developments of the learning game. On such a functor, we provide a lifting-based definition of uncertain bisimilarity and verify basic properties. Beside its applications to Mealy machines, a natural model for automata learning, our framework also instantiates to an existing compatibility relation on suspension automata, which are used in model-based testing. We show that uncertain bisimilarity is a necessary but not sufficient condition for two states being implementable by the same state in the black-box system. We remedy the lack of sufficiency by a characterization of uncertain bisimilarity in terms of coalgebraic simulations.

2012 ACM Subject Classification Theory of computation

Keywords and phrases Coalgebra, Relation Lifting, Bisimilarity, Mealy Machines, ioco

Digital Object Identifier 10.4230/LIPIcs.CALCO.2023.12

Related Version *Full Version*: https://arxiv.org/abs/2303.15279

Funding *Jurriaan Rot*: Partially supported by the NWO grant OCENW.M20.053.
Thorsten Wißmann: Supported by the NWO TOP project 612.001.852 (until 2022) and the DFG project 419850228 (since 2023)

1 Introduction

Inspired from constructive mathematics, Geuvers and Jacobs [6] introduced apartness relations on coalgebraic systems, complementing bisimilarity. While bisimilarity is a coinductive characterization of behavioural equivalence, apartness is inductive, and allows constructing *finite* proofs of difference in behaviour.

Although apartness and bisimilarity are just different sides of the same coin, the angle of 'apartness' turned out to be fruitful in the recent $L^\#$ automata learning algorithm [21]. This algorithm works in the active learning setting of Angluin [1], where a *learner* tries to reconstruct the implementation of an automaton (or concretely a Mealy machine in [21]) from only its black-box behaviour. In $L^\#$, a crucial task of the learner is to determine whether two input words w, v lead to the identical or to distinct states in the black box. Throughout the learning game, the learner makes more and more observations. If at some point the learner finds out that the states q_w, q_v reached by w and v respectively have different behaviours, then q_w and q_v are provably different – called *apart*. For that, it is not required that we know the entire semantics of q_w and q_v; instead, it suffices to observe one aspect of their behaviour in which they differ in incompatible ways.

10th Conference on Algebra and Coalgebra in Computer Science (CALCO 2023).
Editors: Paolo Baldan and Valeria de Paiva; Article No. 12; pp. 12:1–12:17
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

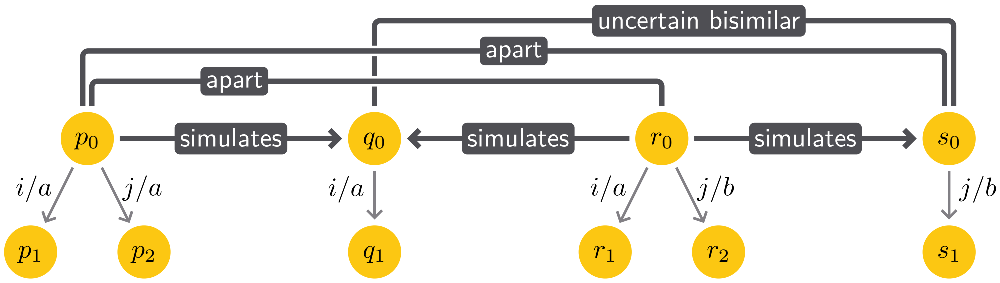


Figure 1 Partial Mealy machine for the input alphabet $\{i, j\}$ and output alphabet $\{a, b\}$.

Once states turn out to be apart, they stay so throughout the entire remaining learning game, no matter which further observations of the black box are made. Thus, the apartness relation grows monotonically as the learning game progresses. This beauty of monotonicity breaks if we consider *bisimilarity*: as long as states q_w, q_v have not been proven different yet, should they be considered bisimilar? Or do we just have insufficiently much information at hand? If we do not know the number of states in the black box, we can never consider states q_w, q_v bisimilar with 100% certainty during the learning process.

In the present work, we close this gap by introducing the notion of *uncertain bisimilarity*, which expresses that two states might be bisimilar – but we are not certain about it, because we simply did not observe any reason yet that would disprove bisimilarity. The main idea is exemplified by the Mealy machine in Figure 1: states p_0 and r_0 are apart, because p_0 has output a on input j whereas r_0 yields a different output b on input j. By the same input j, we can tell that p_0 and s_0 are apart, even though we do not yet know the behaviour of s_0 on input i. Furthermore, states q_0 and s_0 can either turn out to be apart or to be bisimilar, depending on their outputs on input j. Thus, we call q_0 and s_0 *uncertain bisimilar*. If we for instance try to explore the output of q_0 on input j, then depending on the output, q_0 will be identical to p_0 or r_0. Until we know this, q_0 is simulated by both p_0 and by r_0. Simulation is a special case of uncertain bisimilarity, because it not only says that the behaviours are compatible, but that one behaviour is even included in the other.

Our aim is to establish a theory of uncertain bisimilarity at the level of coalgebras, including the motivating example of Mealy machines. Working with bisimulation relations and bisimilarity benefits from a rich categorical theory. In particular, they are themselves coalgebras in the category of relations [8, 11]. Here, the coalgebraic type functor considered on relations is a lifting of the original coalgebraic type functor for the systems of interest.

In the present paper, we incorporate the explicit treatment of the lack of knowledge which is omnipresent in the learning setting. Formally, we do this by equipping the type functor with a partial order. This order $s \sqsubseteq t$ represents that the behaviour s observed so far might be extended to behaviour t after additional observations. This order immediately induces two further notions of lax coalgebra morphisms.

Contributions. With such a partial order on the type functor, we will:

- Define a generic system equality notion, called *uncertain bisimilarity*, derived from the relation lifting of the type functor.
- We show basic properties of the relation such as reflexivity and symmetry. It is immediate that uncertain bisimilarity is not transitive, and thus, no equivalence relation.
- As instances, we discuss (partial) Mealy machines as the running example. Moreover, we cover suspension automata, for which uncertain bisimilarity instantiates to an existing compatibility notion, used in the *ioco* conformance relation from model-based testing [22].

- It is a standard result that standard coalgebraic bisimilarity coincides with being identifiable by coalgebra morphisms (under often-met assumptions on the type functor, see e.g. [18]). We show that uncertain bisimilarity is not characterized via identifiability in lax coalgebra morphisms – for the running example of Mealy machines.
- Instead, we show that uncertain bisimilarity is characterized via coalgebraic simulations – for two definitions of coalgebraic simulation. Concretely, two states are uncertain bisimilar if there is a state in another coalgebra that simulates both (e.g. r_0 simulates both q_0 and s_0 in Figure 1).

2 Preliminaries

We first establish the basic coalgebraic notions used in the technical development later, see e.g. [11]. We assume that the reader is familiar with basic concepts from category theory.

▶ **Definition 2.1.** *For a functor $F\colon \mathcal{C} \to \mathcal{C}$, an F-*coalgebra *(C, c) is an object $C \in \mathcal{C}$ (the* carrier*) together with a morphism $c\colon C \to FC$ in $\mathcal{C}$ (the* structure*). For two F-coalgebras, a coalgebra morphism $h\colon (C, c) \to (D, d)$ is a morphism $h\colon C \to D$ satisfying $Fh \cdot c = d \cdot h$. We denote the category of F-coalgebras by $\mathsf{Coalg}(F)$.*

Most of our coalgebras will live in the category Set of sets and maps and our leading example is the functor modelling Mealy machines:

▶ **Example 2.2.**

1. For fixed sets I and O of input and output symbols, consider the functor

$$\mathcal{M}_T\colon \mathsf{Set} \to \mathsf{Set} \qquad \mathcal{M}_T X = (O \times X)^I.$$

   An $\mathcal{M}_T$-coalgebra is then a set C together with a map $c\colon C \to (O \times C)^I$ which sends each state $q \in C$ and input symbol $i \in I$ to a pair of an output symbol and a successor state to which the Mealy machine transitions: $c(q)(i) \in O \times C$. We write

$$q \xrightarrow{i/o} q'$$

   to specify that $c(q)(i) = (o, q')$. In the name of the functor, the index T shall indicate that the Mealy machine is *total*, in the sense that it is defined for every input $i \in I$.
2. The finitary powerset functor $\mathcal{P}_\mathsf{f}$ sends each set X to the set of its finite subsets $\mathcal{P}_\mathsf{f} X = \{S \subseteq X \mid S \text{ finite}\}$ and maps $f\colon X \to Y$ to direct images: $\mathcal{P}_\mathsf{f} f\colon \mathcal{P}_\mathsf{f} X \to \mathcal{P}_\mathsf{f} Y$, $\mathcal{P}_\mathsf{f} f(S) := \{f(x) \mid x \in S\}$.

A canonical domain for the semantics of coalgebras is the *final* coalgebra:

▶ **Definition 2.3.** *The final F-coalgebra is the final object in $\mathsf{Coalg}(F)$. Concretely, a coalgebra (D, d) is final if for every (C, c) in $\mathsf{Coalg}(F)$ there is a unique coalgebra morphism $h\colon (C, c) \to (D, d)$. If it exists, we denote the final coalgebra for F by $(\nu F, \tau)$ and the induced unique morphism for (C, c) by $[\![-]\!]\colon (C, c) \to (\nu F, \tau)$.*

▶ **Example 2.4.** The final $\mathcal{M}_T$-coalgebra is carried by the set $\nu\mathcal{M}_T = O^{I^+}$ – the set of all maps $I^+ \to O$ from non-empty words I^+ to O.

Equivalently, we can characterize the semantics $\nu\mathcal{M}_T$ in terms of maps $I^* \to O^*$ that interact nicely with the prefix-order on words:

▶ **Notation 2.5.** *For words $v, w \in I^*$ (in particular also for non-empty words $I^+ \subseteq I^*$), we write $v \leq w$ to denote that v is a prefix of w. The length of a word w is denoted by $|w| \in \mathbb{N}$.*

Then, we can characterize $\nu\mathcal{M}_T$ as maps $I^* \to O^*$ that preserve length and prefixes of words:

$$\nu\mathcal{M}_T \cong \{f\colon I^* \to O^* \mid \text{for all } w \in I^*\colon |f(w)| = |w| \text{ and for all } v \le w\colon f(v) \le f(w)\}$$

3 A Lax Coalgebra Morphism Lacks Knowledge

In the learning game for Mealy machines, the learner tries to reconstruct the internal implementation of a Mealy machine

$$c\colon C \to \mathcal{M}_T C = (O \times C)^I$$

by only its black-box behaviour. For that, one assumes a distinguished initial state $q_0 \in C$ and it is the task of the learner to construct a Mealy machine with the same behaviour $[\![q_0]\!]$ as that of q_0. Being in a block-box setting means that the learner knows neither C or c. Instead, the learner can enter a word $i_1, \ldots, i_n \in I$ of input symbols from the input alphabet I to the black box, referred to as a *query*, and then observe the output symbols $o_1, \ldots, o_n \in O$. More precisely, the learner observes the output symbols

$$o_k = [\![q_0]\!](\underbrace{i_1 \cdots i_k}_{\in I^+}) \in O \qquad \text{for every } 1 \le k \le n \qquad (\text{with } [\![-]\!] \text{ as in Example 2.4}).$$

On this query, the black box reveals the output $o_1 \in O$ of the initial state for input i_1. But after performing only this query, we still don't know the output for all the other input symbols $i_1' \in I$, $i_1' \neq i_1$ with which we could have started the input word.

After such a query, the black box returns to the initial state q_0 in order to be ready for the next query. In concrete learning scenarios this reset to initial state is for example realized by resetting the actual hardware of a system that is learned. When learning network protocol implementations, this reset-behaviour is realized by opening a separate network connection (or session) for each new input query.

The $L^\#$ algorithm (for Mealy machines) gathers all the information from the performed input queries in an *observation tree*. This bundles the observations of the experiments so far in a single data structure. However, this structure is not an $\mathcal{M}_T$-coalgebra itself, because the knowledge about the outputs for some inputs $i \in I$ in some states in the tree will be lacking.

We can model this lack of knowledge by the following functor

$$\mathcal{M}\colon \mathsf{Set} \to \mathsf{Set} \qquad \mathcal{M}X = (\{\mathbf{?}\} + O \times X)^I. \tag{1}$$

The element '**?**' models that we do not know the transition yet.

▶ **Notation 3.1.** *We abbreviate partial functions via* $(X \rightharpoonup Y) := (\{\mathbf{?}\} + Y)^X$.

So we can also write $\mathcal{M}X = (I \rightharpoonup O \times X)$. Compared to $\mathcal{M}_T$, a state q in an $\mathcal{M}$-coalgebra $d\colon D \to \mathcal{M}D$ is either undefined for an input $i \in I$, i.e. $d(q)(i) = \mathbf{?}$, or has a transition defined, i.e. we have both an output $o \in O$ and a successor state $q' \in D$ with

$$d(q)(i) = (o, q') \qquad \text{or using notation:} \qquad q \xrightarrow{i/o} q'.$$

This kind of partiality in $\mathcal{M}$ models that whenever the learner sends a word $w \in I^+$, the black box reveals the output symbols of all transitions along the way of processing w. Thus, the semantics of states $q \in D$ in such a partial Mealy machine, i.e. an $\mathcal{M}$-coalgebra, can be characterized by:

$$\nu\mathcal{M} := \{f\colon I^+ \rightharpoonup O \mid \text{for all } v \le w \text{ if } f(w) \in O \text{ then } f(v) \in O\}. \tag{2}$$

This monotonicity condition describes that whenever a learner has observed the behaviour for an input word $w \in I^+$, then we also have observed the outputs of all the prefixes $v \le w$.

$$\begin{array}{ccc} C & \xrightarrow{h} & D \\ {\scriptstyle c}\downarrow & \sqsubseteq & \downarrow{\scriptstyle d} \\ FC & \xrightarrow{Fh} & FD \end{array}$$

Figure 2 Diagrammatic notation of a lax F-coalgebra morphism $h\colon (C,c) \xrightarrow{\sqsubseteq} (D,d)$.

$$\begin{array}{ccc} C & \xrightarrow{h} & D \\ {\scriptstyle c}\downarrow & \sqsupseteq & \downarrow{\scriptstyle d} \\ FC & \xrightarrow{Fh} & FD \end{array}$$

Figure 3 Diagrammatic notation of an oplax F-coalgebra morphism $h\colon (C,c) \xrightarrow{\sqsupseteq} (D,d)$.

▶ **Proposition 3.2.** *The final $\mathcal{M}$-coalgebra $(\nu\mathcal{M}, \tau)$ is characterized by* (2) *and the map*

$$\tau\colon \nu\mathcal{M} \to (I \rightharpoonup O \times \nu\mathcal{M}) \qquad \tau(f) = \quad i \mapsto \begin{cases} \mathbf{?} & \text{if } f(i) = \mathbf{?} \\ (o, w \mapsto f(i\,w)) & \text{if } f(i) \in O \end{cases}$$

The structure sends every $f \in \nu\mathcal{M}$ to a successor structure of type $\mathcal{M}(\nu\mathcal{M})$. For $i \in I$, this successor structure yields $\tau(f)(i) \in \{\mathbf{?}\} + O \times \nu\mathcal{M}$.

During the learning process, the learner might be able to modify the coalgebra after the output of state q on input i has been observed. In this sense, we increase knowledge, and this can be modelled by the usual order on partial functions:

▶ **Definition 3.3.** *For partial functions $t, s\colon A \rightharpoonup B$, we fix the partial order*

$$t \sqsubseteq_{A \rightharpoonup B} s \quad \overset{\text{def}}{\iff} \quad \forall i \in I\colon t(i) \in \{s(i), \mathbf{?}\}.$$

The functor $\mathcal{M}X = (I \rightharpoonup O \times X)$ inherits the poset structure $(\mathcal{M}X, \sqsubseteq)$ from partial maps.

The equivalence means that for every input $i \in I$, the value of $t(i)$ is either undefined ($t(i) = \mathbf{?}$) or agrees with ith entry in the other successor structure ($t(i) = s(i)$). The partial order itself represents how the behaviour can possibly be completed if we found out more information about the full Mealy machines. That is, the partial order shows possible options in the future learning process.

This principle also works for other system types, so we generally assume (e.g. [9]):

▶ **Assumption 3.4.** *Fix a functor $F_{\mathsf{Pos}}\colon \mathsf{Set} \to \mathsf{Pos}$ and define:*

- *$F := U \cdot F_{\mathsf{Pos}}$, where $U\colon \mathsf{Pos} \to \mathsf{Set}$ is the usual forgetful functor.*
- *Let $\sqsubseteq_{FX}$ be the order on $F_{\mathsf{Pos}}X$, i.e. we have $F_{\mathsf{Pos}}X = (FX, \sqsubseteq_{FX})$.*

The functoriality of F_{Pos} means that for every $f\colon X \to Y$, the map $Ff\colon FX \to FY$ is monotone. This partial order gives rise to a lax notion of coalgebra morphisms:

▶ **Definition 3.5.** *A* lax F-coalgebra morphism $h\colon (C,c) \xrightarrow{\sqsubseteq} (D,d)$ *between F-coalgebras is a map $h\colon C \to D$ such that for all $x \in C$ we have $Fh(c(x)) \sqsubseteq_{FD} d(h(x))$. We write $\sqsubseteq$ in squares to indicate lax commutativity as shown in Figure 2. Dually, an* oplax F-coalgebra morphism $h\colon (C,c) \xrightarrow{\sqsupseteq} (D,d)$ *is a map $h\colon C \to D$ such that for all $x \in C$, we have $Fh(c(x)) \sqsupseteq_{FD} d(h(x))$, and denoted in diagrams as shown in Figure 3. In contrast, we write $\circlearrowleft$ to emphasize proper commutativity.*

Lax coalgebra morphisms could also be called functional simulations, because they are simulations (a special kind of relation) and they are functional (a property on relations). Intuitively, $h\colon (C,c) \xrightarrow{\sqsubseteq} (D,d)$ means that (D,d) has at least as many transitions as (C,c). Conversely, $h\colon (C,c) \xrightarrow{\sqsupseteq} (D,d)$ means that (D,d) has possibly fewer transitions than (C,c).

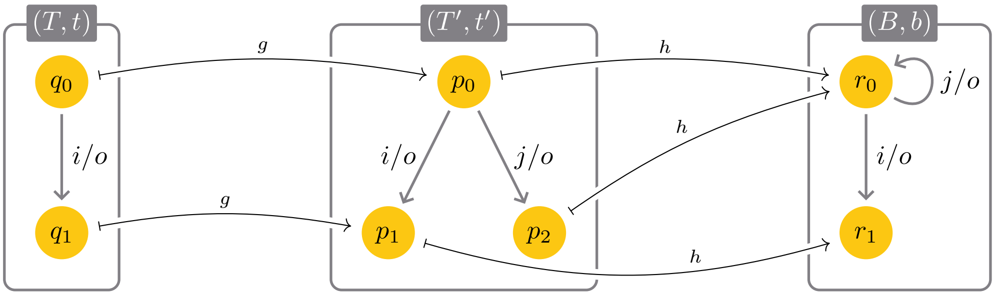


Figure 4 Two lax $\mathcal{M}$-coalgebra morphism $g\colon (T,t) \multimap (T',t')$ and $h\colon (T',t') \multimap (B,b)$ for $\mathcal{M}X = (\{\mathbf{?}\} + O \times X)^I$ with $I = \{i,j\}$, $O = \{o\}$.

In the learning game, lax coalgebra homomorphisms arise naturally, because there, all observations are collected in an observation tree (T,t). This observation tree is an F-coalgebra that admits a lax F-coalgebra morphism $h\colon (T,t) \to (B,b)$ to the black-box (B,b) that needs to be learned. An example of lax morphisms for Mealy machines is visualized in Figure 4.

Of course, the learner only sees the observation tree (T,t) but neither (B,b) nor h. But, the learner can make use of the fact that there is *some* lax coalgebra morphism, and can use it to deduce properties of (B,b). The correctness proof of the $L^{\#}$ learning algorithm [21] in fact relies on the existence of such a lax coalgebra morphism.

Suspension automata. Related to automata learning is the application of *conformance testing* of state-based systems. In particular, the *ioco* (**i**nput **o**utput **co**nformance) relation from testing theory [19] nicely fits into the coalgebraic theory too, while using a non-trivial order on the functor. Specifically, we will focus on *suspension automata*, and later recover the notion of *ioco*-compatibility from [22], see Definition 5.10. Suspension automata are a subclass of deterministic labelled transition systems. They are coalgebras for the following functor:

▶ **Definition 3.6.** *For partial maps which are defined on at least one input we write*

$$(A \rightharpoonup_{\mathsf{ne}} B) := \{f\colon A \rightharpoonup B \mid \exists a \in A\colon f(a) \neq \mathbf{?}\}.$$

For a fixed set of inputs I and outputs O, define the suspension automaton functor

$$\mathcal{S}X := (I \rightharpoonup X) \times (O \rightharpoonup_{\mathsf{ne}} X).$$

▶ **Notation 3.7.** *We denote the projections from (subsets of) the cartesian product $X \times Y$ by $\pi_1\colon X \times Y \to X$ and $\pi_2\colon X \times Y \to Y$.*

Following existing presentations [22, 19], it is not hard to see that suspension automata are coalgebras for this functor:

▶ **Definition 3.8.** *A* suspension automaton *is a finite $\mathcal{S}$-coalgebra, i.e. a finite set of states C and a map $c\colon C \to \mathcal{S}C$. For a given coalgebra (C,c), we write ? for input transitions and ! for output transitions:*

$$x \xrightarrow{?a} y \quad \overset{\mathsf{def}}{\iff} \quad \pi_1(c(x))(a) = y \qquad \textit{and} \qquad x \xrightarrow{!a} y \quad \overset{\mathsf{def}}{\iff} \quad \pi_2(c(x))(a) = y$$

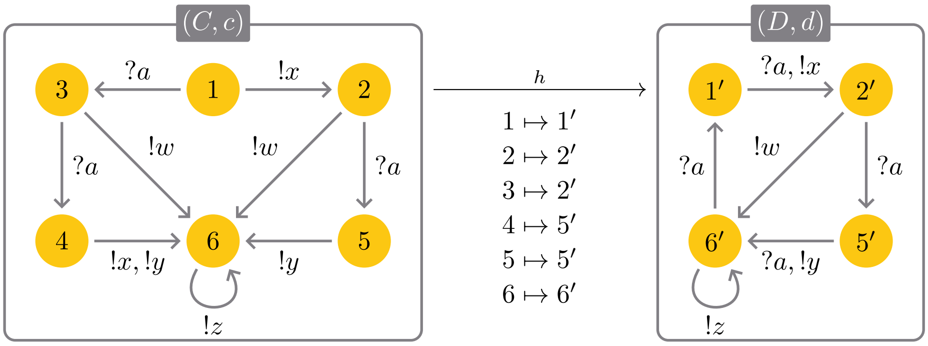


Figure 5 Examples of suspension automata and a lax $\mathcal{S}$-coalgebra morphism between them.

In other words, a suspension automaton is a deterministic LTS where the set of labels is partitioned into inputs and outputs. For some of the inputs and for some of the outputs, a suspension automaton in some state x can make a transition to another state. But every state is non-blocking in the sense that for every state x there is at least one output $o \in O$ such that x can make a transition $x \xrightarrow{!o} y$.

The *ioco* compatibility relation (recalled in Definition 5.10) is characterized by a bisimulation game with an alternating flavour, which can be captured by reversing the order in the output part of the functor $\mathcal{S}$:

▶ **Definition 3.9.** *For $(s_i, s_o) \in \mathcal{S}X$ and $(t_i, t_o) \in \mathcal{S}X$ we put:*

$$(s_i, s_o) \sqsubseteq_{\mathcal{S}X} (t_i, t_o) \quad \overset{\text{def}}{\iff} \quad \underbrace{s_i \sqsubseteq t_i}_{\text{in } I \rightharpoonup X} \text{ and } \underbrace{t_o \sqsubseteq s_o}_{\text{in } O \rightharpoonup X}$$

In other words, when ascending in the order of $\mathcal{S}$, input transitions can be added and output transitions can be removed if there is still at least one output transition afterwards.

▶ **Example 3.10.** We recall two examples of suspension automata from van den Bos et al. [22, Fig. 1] in Figure 5. With the order on $\mathcal{S}$ (Definition 3.9), there is a lax coalgebra morphism $h\colon (C, c) \to (D, d)$ between them that identifies some of the states: $h(3) = h(2)$ and $h(4) = h(5)$. The map h is only a lax coalgebra morphism because there is no input transition for a from 5 (or 4) to 6 in (C, c), but we have $5' \xrightarrow{?a} 6'$ in (D, d). Conversely, there is an output transition $4 \xrightarrow{!x} 6$ in (C, c) but there is no transition $h(4) = 5' \xrightarrow{!x} 6' = h(6)$ in (D, d). Summarizing the above two points, we have:

$$\pi_1(\mathcal{S}h(c(5))) \sqsubsetneq \pi_1(d(h(5))) \quad \text{and} \quad \pi_2(\mathcal{S}h(c(4))) \sqsupsetneq \pi_2(d(h(4)))$$

The partial order on the functor does not only relax the notion of morphism, but also gives rise to a new coalgebraic bisimulation notion, which we introduce in Section 5.

4 Bisimulation Notions are Liftings

In this section we recall coalgebraic bisimulation through the lens of relation liftings; for an extensive introduction see [11]. We start by fixing some notation regarding relations.

▶ **Notation 4.1.** *Given relations $R \subseteq X \times Y$ and $S \subseteq Y \times Z$, the composition $R \circ S$ is given by: $R \circ S := \{(x, z) \in X \times Z \mid \exists y \in Y : (x, y) \in R$ and $(y, z) \in Z\}$. We denote the converse of R by $R^{\mathsf{op}} = \{(y, x) \mid (x, y) \in R\}$. The equality relation (also called the diagonal) on a set*

X is denoted by $\mathsf{Eq}_X = \{(x,x) \mid x \in X\}$*. Given a map* $f\colon X \to Y$ *and a relation* $U \subseteq Y \times Y$*, inverse image is denoted by* $(f \times f)^{-1}(U) = \{(x_1,x_2) \mid (f(x_1), f(x_2)) \in U\}$*. The kernel relation of f is given by* $\ker f = \{(x_1,x_2) \mid f(x_1) = f(x_2)\}$*. Note that* $\ker f = (f \times f)^{-1}(\mathsf{Eq}_Y)$*.*

For a structural study of (bi)simulation notions on coalgebras, we consider the fibred category of *relations*:

▶ **Definition 4.2.** *The category* Rel *has objects* (X,R)*, where X is a set and* $R \subseteq X \times X$*, i.e. R is a relation on X. The morphisms* $f\colon (X,R) \to (Y,S)$ *in* Rel *are maps* $f\colon X \to Y$ *that preserve the relation, i.e.* $(x_1,x_2) \in R$ *implies* $(f(x_1), f(x_2)) \in S$*. The obvious forgetful functor is* $p\colon \mathsf{Rel} \to \mathsf{Set}$*, given by* $p(X,R) = X$*.*

The forgetful functor p is a fibration; for a thorough introduction, see the first chapter of Jacobs' book [10]. We can express the preservation property of the morphisms in Rel in a point-free way: $f\colon X \to Y$ is a map from (X,R) to (Y,S) in Rel if and only if

$$R \subseteq (f \times f)^{-1}(S).$$

Equality extends to a functor $\mathsf{Eq}\colon \mathsf{Set} \to \mathsf{Rel}$, given by $\mathsf{Eq}(X) = (X, \mathsf{Eq}_X)$ and $\mathsf{Eq}(f) = f$. This is well-defined since we have $\mathsf{Eq}_X \subseteq (f \times f)^{-1}(\mathsf{Eq}_Y)$ for every map $f\colon X \to Y$.

To study relations on F-coalgebras – most notably notions of behavioural equivalence and inclusion – we lift the type functor F from Set to an endofunctor $\hat{F}$ on Rel.

$$\begin{array}{ccc} \mathsf{Rel} & \xrightarrow{\hat{F}} & \mathsf{Rel} \\ {\scriptstyle p}\downarrow & & \downarrow{\scriptstyle p} \\ \mathsf{Set} & \xrightarrow{F} & \mathsf{Set} \end{array}$$

For all set functors, such a lifting exists in a canonical way:

▶ **Definition 4.3.** *For a functor* $F\colon \mathsf{Set} \to \mathsf{Set}$*, the relation lifting* $\hat{F}\colon \mathsf{Rel} \to \mathsf{Rel}$ *is given by*

$$\hat{F}(R \subseteq X \times X) = \{(x,y) \in FX \times FX \mid \exists t \in FR\colon F\pi_1(t) = x \text{ and } F\pi_2(t) = y\}.$$

Hence, $\hat{F}$ transforms relations on X into relations on FX. Before we proceed, we list several standard properties of the relation lifting:

▶ **Lemma 4.4** [11]**.** *For every functor* $F\colon \mathsf{Set} \to \mathsf{Set}$*, we have*

1. *Monotonicity: if* $R \subseteq S$ *then* $\hat{F}(R) \subseteq \hat{F}(S)$*.*
2. *Preservation of equality:* $\mathsf{Eq} \circ F \subseteq \hat{F} \circ \mathsf{Eq}$*.*
3. *Preservation of converse:* $\hat{F}(R^{\mathsf{op}}) = (\hat{F}(R))^{\mathsf{op}}$ *for all* $R \subseteq X \times X$*.*
4. *Preservation of inverse images: For a map* $f\colon X \to Y$ *and relation* $S \subseteq Y \times Y$*, we have*

$$\hat{F}((f \times f)^{-1}(S)) \subseteq (Ff \times Ff)^{-1}(\hat{F}(S))\,.$$

Moreover, if F preserves weak pullbacks, then this is an equality.

As a consequence of monotonicity and preservation of inverse images, $\hat{F}$ indeed extends to a lifting of F, given on morphisms by $\hat{F}(f) = F(f)$.

▶ **Example 4.5.** In our example of (partial) Mealy machines as coalgebras for $\mathcal{M}X = (\{\mathbf{?}\} + O \times X)^I$, a relation $R \subseteq X \times X$ is lifted to the relation $\hat{\mathcal{M}}R \subseteq \mathcal{M}X \times \mathcal{M}X$ given by

$$\begin{array}{lll} (s,t) \in \hat{\mathcal{M}}R \quad \text{iff} & \text{for all } i \in I\colon & (s(i) = \mathbf{?} \text{ and } t(i) = \mathbf{?}) \text{ or} \\ & & (s(i), t(i)) \in \{\big((o,x),(o,y)\big) \mid o \in O, (x,y) \in R\} \end{array}$$

Thus, successor structures $s, t \in \mathcal{M}R$ are related by $\hat{\mathcal{M}}R$ if s and t have transitions defined for the same inputs $i \in I$, and for all inputs $i \in I$ for which $s(i) = (o, x)$ and $t(i) = (o', y)$ are defined, both have the same output $o = o'$, and the successor states are related $(x, y) \in R$.

The relation lifting is reminiscent of the criterion of a relation $R \subseteq C \times C$ being a bisimulation on Mealy machines. However, in relation liftings, we can distinguish between the relation R on the successor states on the one hand and the relation on the predecessor states on the other hand. If we let the relation on predecessor and successor states coincide, then the relation lifting gives rise to bisimilarity as follows [8, 11].

▶ **Definition 4.6.** *A relation $R \subseteq C \times C$ on the state space of a coalgebra $c\colon C \to FC$ is a* bisimulation *if $R \subseteq (c \times c)^{-1}(\hat{F}(R))$. States $x, y \in C$ are called* bisimilar *if there is a bisimulation relating them.*

Note that $(c \times c)^{-1}(\hat{F}(-))\colon \mathsf{Rel}_C \to \mathsf{Rel}_C$ is a monotone map on the complete lattice $\mathsf{Rel}_C = \mathcal{P}(C \times C)$ of relations on C, ordered by inclusion. A bisimulation is thus a post-fixed point for this map, and bisimilarity is the greatest post-fixed point, which is also the greatest fixed point by the Knaster-Tarski theorem. Characterizing bisimilarity as the greatest fixed point of a monotone map is standard in the classical theory of coinduction [16].

▶ Remark 4.7 (Disjoint union of coalgebras). In the definition of bisimulation, we consider a relation R on the state space of a single coalgebra $c\colon C \to FC$. This bisimulation notion straightforwardly generalizes to states of different F-coalgebras $x \in C$, and $y \in D \xrightarrow{d} FD$, because we can consider the bisimulation notion on the disjoint union (i.e. coproduct) of the coalgebras (C, c) and (D, d):

$$C + D \xrightarrow{c+d} FC + FD \xrightarrow{[F\mathsf{inl}, F\mathsf{inr}]} F(C + D)$$

where $\mathsf{inl}\colon C \to C + D$ and $\mathsf{inr}\colon D \to C + D$ are the coproduct injections and $[-,-]$ is case distinction (i.e. the universal mapping property of the coproduct). So by the bisimilarity of x and y we mean the bisimilarity of $\mathsf{inl}(x)$ and $\mathsf{inr}(y)$ in the above combined coalgebra. One can easily see that this generalization is well-defined: in the special case where $(D, d) := (C, c)$, states x, y in C are bisimilar iff $\mathsf{inl}(x), \mathsf{inr}(y)$ are bisimilar in the coalgebra on $C + C$.

▶ **Example 4.8.** The relation lifting for $\mathcal{M}$ (Example 4.5) thus gives rise to the following: a bisimulation on a coalgebra $c\colon C \to \mathcal{M}C$ is a relation $R \subseteq C \times C$ such that

$$R \subseteq (c \times c)^{-1}(\hat{\mathcal{M}}R).$$

Spelling out the inclusion yields that R is a bisimulation iff for all $(x, y) \in R$ and $i \in I$:

1. $c(x)(i) = \textbf{?}$ iff $c(y)(i) = \textbf{?}$,
2. if $c(x)(i) = (o, x') \in O \times C$, then $c(y)(i) = (o, y')$ for some $y' \in C$ with $(x', y') \in R$, and
3. if $c(y)(i) = (o, y') \in O \times C$, then $c(x)(i) = (o, x')$ for some $x' \in C$ with $(x', y') \in R$.

For example, the leaf states q_1, p_1, p_2, r_1 in Figure 4 are all pairwise bisimilar. However, q_0 and p_0 are not bisimilar: q_0 can not mimic the j-transition of p_0. Similarly, q_0 and r_0 are not bisimilar (and also p_0 and r_0 are not bisimilar).

In order to still express the compatibility of q_0 and p_0, we relax the notion of coalgebraic bisimilarity in the next section.

5 Uncertain Bisimilarity

So far, we have not considered the order $\sqsubseteq$ when discussing bisimulations on coalgebras for a functor F satisfying Assumption 3.4. By taking the order into account, we introduce the notion of *uncertain bisimilarity*. In particular, in the example of Mealy machines it captures a notion of equivalence where 'unknown' transitions are ignored. Since we stick to the principle that bisimulation notions are coalgebras in Rel, we only need to make use of the order $\sqsubseteq$ when defining a functor on Rel. The desired bisimulations will then be coalgebras for this functor:

▶ **Definition 5.1.** *The* uncertain relation lifting *of F is defined on a relation $R \subseteq X \times X$ by*

$$\hat{F}_{\sqsubseteq}(R) \;:=\; \sqsubseteq_{FX} \circ \hat{F}(R) \circ \sqsupseteq_{FX}$$

▶ Remark 5.2. Definition 5.1 is inspired by the notion of *simulation* on coalgebras by Hughes and Jacobs [9]. In their work, a simulation on a coalgebra (C, c) is a relation R such that

$$R \subseteq (c \times c)^{-1}(\sqsubseteq_{FC} \circ \hat{F}(R) \circ \sqsubseteq_{FC})$$

The lifting $\sqsubseteq \circ \hat{F}(-) \circ \sqsubseteq$ of F is referred to in *op. cit.* as the *lax relation lifting.*

▶ **Definition 5.3.** *An* uncertain bisimulation *R on a coalgebra $c\colon C \to FC$ is a relation $R \subseteq C \times C$ with $R \subseteq (c \times c)^{-1}(\hat{F}_{\sqsubseteq}(R))$. States $x, y \in C$ are called* uncertain bisimilar *if there is an uncertain bisimulation relating them. Complementarily, $x, y \in C$ are called* apart *if there is no uncertain bisimulation relating them.*

The uncertainty here expresses that in the learning setting, we are not entirely certain that the two states are bisimilar. With an extension of the system by a future observation, they might turn out to be non-bisimilar. With this intuition, the opposite property is simply called *apart*: whenever two states are separated, they will stay so no matter how the system might be extended by further transitions.

When unfolding the definitions, we obtain the following explicit characterization:

▶ **Lemma 5.4.** *A relation $R \subseteq C \times C$ on $c\colon C \to FC$ is an uncertain bisimulation if and only if for every $(x, y) \in R$ there exists $t \in FR$ such that*

$$c(x) \sqsubseteq F\pi_1(t) \qquad \text{and} \qquad c(y) \sqsubseteq F\pi_2(t).$$

When representing the witnesses t as a choice function, then we equivalently have a map $r\colon R \to FR$ making the projections π_1, π_2 oplax coalgebra morphisms $(R, r) \overset{\sqsupseteq}{\longrightarrow} (C, c)$:

$$\begin{array}{ccccc} C & \xleftarrow{\pi_1} & R & \xrightarrow{\pi_2} & C \\ \downarrow{\scriptstyle c} & \sqsubseteq & \downarrow{\scriptstyle r} & \sqsupseteq & \downarrow{\scriptstyle c} \\ FC & \xleftarrow{F\pi_1} & FR & \xrightarrow{F\pi_2} & FC \end{array}$$

This characterization instantiates to partial Mealy machines as:

▶ **Lemma 5.5.** *For partial Mealy machines $c\colon C \to \mathcal{M}C$ a* reflexive *relation $R \subseteq C \times C$ is an uncertain bisimulation if and only if for all $(x, y) \in R$ and $i \in I$ we have:*

$$x \xrightarrow{i/o} x' \text{ and } y \xrightarrow{i/o'} y' \qquad \text{imply} \qquad o = o' \text{ and } (x', y') \in R. \tag{3}$$

This condition is vacuously satisfied for all $(x, y) \in R$ and $i \in I$ whenever x or y have no i-transition defined. In this characterization, we use the mild assumption of R being reflexive in order to be able to define the coalgebra structure $r\colon R \to FR$ of Lemma 5.4 in the case where only one of the related states has an i-transition defined. Even without R being reflexive, every uncertain bisimulation R satisfies (3). Conversely, for every relation R satisfying (3), the relation $\mathsf{Eq} \cup R$ is an uncertain bisimulation (i.e. we implicitly work with reflexive closure as an up-to technique [3]).

The characterization in Lemma 5.5 leads to the following coinduction principle:

▶ **Proposition 5.6.** *States x, y in a partial Mealy machine $c\colon C \to \mathcal{M}C$ are uncertain bisimilar iff*

$$\text{for all } w \in I^+\colon \qquad [\![x]\!](w) \in O \text{ and } [\![y]\!](w) \in O \qquad \Longrightarrow \qquad [\![x]\!](w) = [\![y]\!](w)$$

Dually, x and y are apart iff there is some $w \in I^+$ for which both are defined but differ: $\mathbf{?} \neq [\![x]\!] \neq [\![y]\!] \neq \mathbf{?}$. Thus, this instance matches the explicit definition of *apart states* in the context of the $L^\#$ learning algorithm [21, Def. 2.6].

Recall that the final coalgebra semantics of a state $x \in C$ is a partial map $[\![x]\!]\colon I^+ \rightharpoonup O$ (in $\nu\mathcal{M}$, Proposition 3.2). This map sends each input word $w \in I^+$ to the output symbol of the last transition of the run of w, if such a run exists. If not all required transitions exist, then the partial map is undefined (i.e. $[\![x]\!](w) = \mathbf{?}$). The characterization in Proposition 5.6 states that two states are uncertain bisimilar if for all input words $w \in I^+$, whenever both behaviours $[\![x]\!]$, $[\![y]\!]$ are defined, then they must agree.

▶ **Example 5.7.** If there is a natural transformation $\top\colon 1 \to F$ such that $\top_X$ is the greatest element of FX, then all states in any F-coalgebra are uncertain bisimilar.

▶ **Example 5.8.** For the inclusion order $\subseteq$ on the finitary powerset functor $\mathcal{P}_\mathsf{f}$, any two states x, y in any $\mathcal{P}_\mathsf{f}$-coalgebra $c\colon C \to \mathcal{P}_\mathsf{f}C$ are uncertain bisimilar. Essentially, the issue is that any pair of elements $s, t \in \mathcal{P}_\mathsf{f}C$ has an upper bound in $(\mathcal{P}_\mathsf{f}C, \subseteq)$.

▶ **Example 5.9.** We re-obtain ordinary bisimilarity as the instance where the order $\sqsubseteq$ on FX is the discrete poset structure: $\sqsubseteq_{FC} := \mathsf{Eq}_{FC}$.

The instance for suspension automata has explicitly been studied in the literature [22]:

▶ **Definition 5.10** [22, Def. 15]**.** *A relation $R \subseteq C \times C$ on a suspension automaton $c\colon C \to \mathcal{S}C$ is an* (ioco) compatibility relation *if for all $(x, y) \in R$ we have:*

1. *for all $x \xrightarrow{?a} x'$ and $y \xrightarrow{?a} y'$ we have $(x', y') \in R$*
2. *there exists $o \in O$ such that $x \xrightarrow{!o} x'$, $y \xrightarrow{!o} y'$, and $(x', y') \in R$.*

▶ **Proposition 5.11.** *A reflexive relation on a suspension automaton is a ioco compatibility relation iff it is an uncertain bisimulation (for $\mathcal{S}$ with the order from Definition 3.9).*

In the proof it is relevant that the output transitions of suspension automata are non-empty partial maps $C \to (O \rightharpoonup_{\mathsf{ne}} C)$. Non-emptiness means that whenever there is a coalgebra structure $r\colon R \to \mathcal{S}R$ on a relation $R \subseteq C \times C$, then all related states $(x, y) \in R$ have at least one common output $o \in O$. This is reflected by the existentially quantified condition in the definition of ioco compatibility.

5.1 Properties

Having discussed instances, we now uniformly establish general properties of uncertain bisimilarity. We start by listing properties of uncertain relation lifting, analogous to Lemma 4.4.

▶ **Lemma 5.12.** *For any functor* $F_{\mathsf{Pos}}\colon \mathsf{Set} \to \mathsf{Pos}$*, we have the following properties of uncertain relation lifting:*

1. *Monotonicity: if* $R \subseteq S$ *then* $\hat{F}_{\sqsubseteq}(R) \subseteq \hat{F}_{\sqsubseteq}(S)$*.*
2. *Preservation of equality:* $\mathsf{Eq} \circ F \subseteq \hat{F}_{\sqsubseteq} \circ \mathsf{Eq}$*.*
3. *Preservation of converse:* $\hat{F}_{\sqsubseteq}(R^{\mathsf{op}}) = (\hat{F}_{\sqsubseteq}(R))^{\mathsf{op}}$ *for all* $R \subseteq X \times X$*.*
4. *Preservation of inverse images: For a map* $f\colon X \to Y$ *and relation* $S \subseteq Y \times Y$*, we have*

$$\hat{F}_{\sqsubseteq}((f \times f)^{-1}(S)) \subseteq (Ff \times Ff)^{-1}(\hat{F}_{\sqsubseteq}(S))\,.$$

Similar to the case of $\hat{F}$, by monotonicity and preservation of inverse images, $\hat{F}_{\sqsubseteq}$ extends to a lifting of F. Uncertain bisimilarity is reflexive and symmetric:

▶ **Lemma 5.13.** *The equality relation* Eq_C *on any coalgebra* (C, c) *is an uncertain bisimulation, and if* $R \subseteq C \times C$ *is an uncertain bisimulation then so is* R^{op}*.*

Unsurprisingly, uncertain bisimilarity is not transitive: even though two states x and z are certainly non-bisimilar (i.e. not uncertain bisimilar), there can be a state y that is uncertain bisimilar to both x and z (e.g. p_0, q_0, s_0 in Figure 1). Similarly, ioco compatibility is known to not be transitive in general [22, Ex. 17].

This lack of transitivity makes it non-trivial to characterize uncertain bisimilarity in terms of being identifiable by a morphism, in the way it holds for normal bisimilarity. Still, we can show some preservation results that match the intuition that the order $\sqsubseteq$ adds transitions (or other information). Since uncertain bisimilarity of two states means that there is no conflict in their existing transition behaviour, they stay uncertain bisimilar if we omit transitions:

▶ **Lemma 5.14.** *Uncertain bisimilarity is preserved by oplax morphisms: whenever states* x, y *in* (C, c) *are uncertain bisimilar, then for every oplax coalgebra morphism* $h\colon (C, c) \to (D, d)$*, the states* $h(x)$ *and* $h(y)$ *are uncertain bisimilar in* (D, d)*.*

Conversely, we can show that if two states can be identified by a lax coalgebra morphism, then they are uncertain bisimilar. For the corresponding proof for (canonical) relation liftings $\hat{F}$, one uses weak pullback preservation as a sufficient condition for preservation of inverse images. For uncertain bisimilarity we will simply make preservation of inverse images an assumption, referred to as *stability*. This terminology follows Hughes and Jacobs [9], who define a similar condition for their lax relation lifting.

▶ **Definition 5.15.** *The functor* F_{Pos} *is called* stable *if* $\hat{F}_{\sqsubseteq}$ *commutes with inverse images on reflexive relations, i.e. the inclusion in Item 4 of Lemma 5.12 is an equality if* S *is reflexive.*

▶ Remark 5.16. Contrary to the variant in [9], we require the converse of Lemma 5.12.4 only for *reflexive* relations. The reason is that even for the case of Mealy machines, $F = \mathcal{M}$, the converse of Lemma 5.12.4 does not hold if we drop that assumption.

▶ **Example 5.17.** $\hat{\mathcal{M}}_{\sqsubseteq}$ is stable.

▶ **Lemma 5.18.** *Suppose that* F_{Pos} *is stable. Then any lax coalgebra morphism* $h\colon (C, c) \overset{\sqsubseteq}{\longrightarrow} (D, d)$ *reflects uncertain bisimilarity, that is, if* $R \subseteq D \times D$ *is a reflexive uncertain bisimulation relation then so is* $(h \times h)^{-1}(R)$*.*

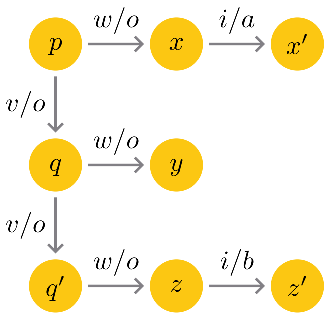


Figure 6 Mealy machine of Example 5.20 ....

$[\![p]\!](w) = o$	$[\![q]\!](w) = o$
$[\![p]\!](wi) = a$	$[\![q]\!](wi) = \textbf{?}$
$[\![p]\!](v) = o$	$[\![q]\!](v) = o$
$[\![p]\!](vw) = o$	$[\![q]\!](vw) = o$
$[\![p]\!](vv) = o$	$[\![q]\!](vv) = \textbf{?}$
$[\![p]\!](vvw) = o$	$[\![q]\!](vvw) = \textbf{?}$
$[\![p]\!](vvwi) = b$	$[\![q]\!](vvwi) = \textbf{?}$
$[\![p]\!](vwi) = \textbf{?}$	$[\![q]\!](vwi) = b$

Figure 7 ... and its semantics.

As a consequence, under the assumption of stability, if states x, y of a coalgebra are identified by a lax homomorphism h then they are uncertain bisimilar.

▶ **Corollary 5.19.** *Suppose that F_{Pos} is stable. Then the kernel* $\ker h$ *of a lax coalgebra morphism* $h\colon (C,c) \xrightarrow{\sqsubseteq} (D,d)$ *is an uncertain bisimulation.*

This gives half a characterization theorem of uncertain bisimilarity (assuming stability):

$$\begin{array}{c}\text{States } x,y \text{ can be identified}\\ \text{by a lax coalgebra morphism}\end{array} \implies \begin{array}{c}\text{States } x,y \text{ are}\\ \text{uncertain bisimilar}\end{array} \tag{4}$$

For standard bisimilarity, the converse direction also holds: whenever states x, y are bisimilar (in the ordinary sense), then they can be identified by an (ordinary) coalgebra morphism. For uncertain bisimilarity however, the converse direction even fails when restricting to tree-shaped Mealy machines:

▶ **Example 5.20.** Consider the partial Mealy machine (C, c) in Figure 6 for $I = \{v, w, i\}$ and $O = \{a, b, o\}$. In this machine, p and q are uncertain bisimilar, because their semantics matches on all defined input words, as verified in Figure 7 (using Proposition 5.6). However, there is no lax coalgebra morphism $f\colon (C,c) \xrightarrow{\sqsubseteq} (D,d)$ with $f(p) = f(q)$. To see this, first observe that for any such f, we can derive the following equalities:

$$f(p) \xrightarrow{w/o} f(x) \quad\text{and}\quad f(q) \xrightarrow{w/o} f(y) \quad\text{implies}\quad f(x) = f(y),$$

$$f(p) \xrightarrow{v/o} f(q) \xrightarrow{w/o} f(y) \quad\text{and}\quad f(q) \xrightarrow{v/o} f(q') \xrightarrow{w/o} f(z) \quad\text{implies}\quad f(y) = f(z)$$

and hence $f(x) = f(z)$. By Corollary 5.19 this means x and z are uncertain bisimilar; but i witnesses that x and z are apart – a contradiction! Thus, there is no $f\colon (C,c) \to (D,d)$ with $f(p) = f(q)$.

5.2 Characterization via Simulations

We can remedy the failure of the converse direction of (4) by going from functional simulations (i.e. lax coalgebra morphisms) to proper simulations in the sense of spans of (lax) morphisms.

There are multiple ways to define simulations between coalgebras for functors $F\colon \mathsf{Set} \to \mathsf{Set}$ equipped with an order $\sqsubseteq$. The way that Hughes and Jacobs [9] define simulations (see also Remark 5.2) between coalgebras (C, c) and (D, d) corresponds to a relation $R \subseteq C \times D$ and a structure $r\colon R \to FR$ making the projections oplax and lax morphisms:

$$\begin{array}{l} \pi_1 \colon (R, r) \mathrel{-\!\sqsupseteq\!\rightarrow} (C, c) \\ \pi_2 \colon (R, r) \mathrel{-\!\sqsubseteq\!\rightarrow} (D, d) \end{array} \quad \text{that is, diagrammatically:} \quad \begin{array}{ccccc} C & \xleftarrow{\pi_1} & R & \xrightarrow{\pi_2} & D \\ \downarrow{\scriptstyle c} & \sqsubseteq & \downarrow{\scriptstyle r} & \sqsubseteq & \downarrow{\scriptstyle d} \\ FC & \xleftarrow{F\pi_1} & FR & \xrightarrow{F\pi_2} & FD \end{array}$$

Note that due to using both lax and oplax morphisms, such a simulation is subtly different from the diagram in Lemma 5.4. We can now show that this span-based definition of simulation characterizes uncertain bisimilarity:

▶ **Proposition 5.21.** *Given that F_{Pos} is stable, the following are equivalent for all states x, y in a coalgebra $c\colon C \to FC$:*
1. *x and y are uncertain bisimilar.*
2. *There is a state $z \in D$ in another coalgebra (D, d) and a simulation $S \subseteq C \times D$ in the style of Hughes and Jacobs such that $(x, z) \in S$ and $(y, z) \in S$.*

The second item intuitively means that the states x and y can be 'identified' by a simulation. We obtained a converse to the implication in (4) when replacing 'lax coalgebra morphism' with 'simulation'. In the proof of the first direction (top to bottom), we use that in sets, every surjective function $e\colon X \to Y$ has a right-inverse $a\colon Y \to X$ (i.e. with $e \circ a = \mathsf{id}_Y$), using the axiom of choice. In the second direction (bottom to top), we use the stability of F_{Pos}.

Another slightly different notion of simulation on coalgebras arises from the approach to bisimilarity via open maps [12, 23]. Here, a simulation between (C, c) and (D, d) is again a relation $R \subseteq C \times D$ equipped with a coalgebra structure $r\colon R \to FR$ such that
1. the projection π_1 is a coalgebra morphism $\pi_1\colon (R, r) \to (C, c)$, and
2. the projection π_2 is a lax coalgebra morphism $\pi_2\colon (R, r) \mathrel{-\!\sqsubseteq\!\rightarrow} (D, d)$:

$$\begin{array}{ccccc} C & \xleftarrow{\pi_1} & R & \xrightarrow{\pi_2} & D \\ \downarrow{\scriptstyle c} & \circlearrowleft & \downarrow{\scriptstyle r} & \sqsubseteq & \downarrow{\scriptstyle d} \\ FC & \xleftarrow{F\pi_1} & FR & \xrightarrow{F\pi_2} & FD \end{array}$$

Hence, any open-map-style simulation is also a simulation in the style of Hughes and Jacobs. In our leading examples, the converse inclusion also holds, as we show in the following.

▶ Remark 5.22. The above definition of simulation is reminiscent of Fiore's *ordered categorical bisimulation* [5, Def. 6.1], for which the partial order comes from the base category being Pos-enriched, i.e. a partial order on every hom set $\mathsf{hom}(A, B)$ is assumed. In contrast, we only require a partial order on FX, i.e. the partial order is part of the functor data, not the category.

In order to show the equivalence of open-map-style simulations to that by Hughes and Jacobs, we impose another assumption on the functor:

▶ **Definition 5.23.** *We call the order $\sqsubseteq$ on the functor F* restricting *if for all maps $f\colon X \to Y$ and all $s \in FX$, $t \in FY$ we have*

$$t \sqsubseteq_{FY} Ff(s) \qquad \Longrightarrow \qquad \text{there is some } s' \sqsubseteq s \text{ with } t = Ff(s'). \tag{5}$$

The idea behind s' is that it is the restriction of s to those transitions that are defined in t, so that $Ff\colon FX \to FY$ maps s' to t:

▶ **Example 5.24.** The functor $\mathcal{M}$ for partial Mealy machines is restricting: for $f\colon X \to Y$, $s \in \mathcal{M}X$, and $t \sqsubseteq_{\mathcal{M}Y} \mathcal{M}f(s)$, define

$$s' \in \mathcal{M}X = (\{\mathbf{?}\} + O \times X)^I \quad \text{by} \quad s'(i) = \begin{cases} \mathbf{?} & \text{if } t(i) = \mathbf{?} \\ s(i) & \text{otherwise.} \end{cases}$$

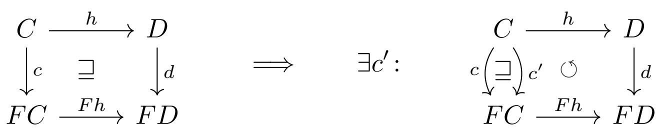
This definition makes $s' \sqsubseteq s$ true because for all $i \in I$, whenever $s'(i)$ is defined (i.e. $s'(i) \neq \mathbf{?}$) then $s(i)$ is defined, too. The inequality $t \sqsubseteq_{\mathcal{M}Y} \mathcal{M}f(s)$ implies

$$(s'(i) = \mathbf{?} \iff t(i) = \mathbf{?}) \qquad \text{for all } i \in I$$

and moreover, whenever $s'(i) = (o, x)$ for $i \in I$, then $t(i) = s(i) = s'(i)$. Hence, $\mathcal{M}f(s') = t$.

▶ Remark 5.25. In the definition of *restricting*, we have $t \sqsubseteq Ff(s)$ as the condition and then construct some s' with $s' \sqsubseteq s$. Thus, one might be tempted to think that there is a Galois connection hidden. Note however, that this is not the case in the example of partial Mealy machines because the construction of s' does depend on s! Hence, it is not possible to construct an adjoint map $FY \to FX$ in general.

▶ **Lemma 5.26.** *If F is restricting, then for every oplax morphism $h\colon (C, c) \mathrel{-\sqsupseteq\rightarrow} (D, d)$, there is a structure $c'\colon C \to FC$ such that $c'(x) \sqsubseteq c(x)$ for all $x \in C$ making h a (proper) coalgebra morphism.*



This lemma turns Hughes/Jacobs simulations into simulations in the style of open maps:

▶ **Lemma 5.27.** *Given that $\sqsubseteq$ is restricting, the following are equivalent for any relation $S \subseteq C \times D$ on coalgebras (C, c), (D, d):*
1. *there is a map $S \to FS$ making S a simulation in the style of Hughes and Jacobs.*
2. *there is a map $S \to FS$ making S a simulation in the style used in open maps.*

Thus, we can combine Lemma 5.27 and the previous characterization Proposition 5.21:

▶ **Theorem 5.28.** *Given that F_{Pos} is stable and that $\sqsubseteq$ is restricting, the following are equivalent for all states x, y in a coalgebra $c\colon C \to FC$:*
1. *x and y are uncertain bisimilar.*
2. *There is a state $z \in D$ in another coalgebra (D, d) and a simulation $S \subseteq C \times D$ in the style of Hughes and Jacobs such that $(x, z) \in S$ and $(y, z) \in S$.*
3. *There is a state $z \in D$ in another coalgebra (D, d) and an open-map-style simulation $S \subseteq C \times D$ such that $(x, z) \in S$ and $(y, z) \in S$.*

▶ **Example 5.29.** For partial Mealy machines, the abstract definitions of simulation instantiate to the usual notion of simulation between (C, c) and (D, d) when considering the Mealy machines as deterministic LTSs for the alphabet $I \times O$: a simulation is a relation $R \subseteq C \times D$ such that for all $(x, z) \in R$ and $x \xrightarrow{i/o} x'$ there is some $z' \in D$ such that $z \xrightarrow{i/o} z'$ and $(x', z') \in R$. The characterization in Proposition 5.21 shows that for all states $x, y \in C$ in $c\colon C \to \mathcal{M}C$ we have

x and y are uncertain bisimilar	$\iff$	There is a state z in some $d\colon D \to \mathcal{M}D$ such that z simulates x and y

▶ **Example 5.30.** For the compatibility relation on suspension automata, a similar equivalence holds. In the specific simulation notion (called *coinductive ioco relation* [22, Def. 4]), the input transitions are preserved in the usual direction and the output transitions are preserved in the converse direction. Then, it is shown that states x, y in a suspension automaton are compatible iff there is a state z in another suspension automaton which conforms (according to the *ioco* relation) to both x and y [22, Lem. 24.2].

6 Conclusions and Future Work

We introduced *uncertain bisimilarity*, a notion to talk about behavioural compatibility on a coalgebraic level of generality. Instances include both partial Mealy machines and the *ioco* conformance relation from model-based testing. The setting is tailored towards the lack of knowledge in automata learning games. We are optimistic that this generalization provides a step from the $L^{\#}$ learning algorithm [21] towards new coalgebra learning algorithms. While previous categorical frameworks [2, 20, 7, 4] generalize Angluin's classical L^* algorithm, the development of a variant of $L^{\#}$ at a high level of generality could be useful, as the experiments [21] point to a better performance in the case of Mealy machines.

So far, we have shown that uncertain bisimilarity is equivalent to being simulated by a common state. A similar observation might be lifted to the final coalgebra by defining a suitable simulation order on the final coalgebra. In this context it would be interesting to explicitly connect our results to the similarity quotients in [15].

Standard coalgebraic bisimilarity can also be characterized as indistinguishability via formulas of coalgebraic modal logic [17, 13]. We are confident that uncertain bisimilarity can be characterized in similar terms. Since it is a coinductively defined relation involving a non-standard relation lifting, a good starting point may be the framework in [14], although that does not provide a canonical construction of a logic but only the infrastructure for proving expressiveness and adequacy. To obtain such a logic, it should be feasible to transfer modalities from an existing system type functor (e.g. $\mathcal{M}_T$) to the functors involving order and partial behaviours (e.g. $\mathcal{M}$), such that properties like adequacy or even expressiveness are inherited. These distinguishing modal formulas can then serve as witnesses for disproving uncertain bisimilarity, that is, for showing apartness.

The relation lifting based definition of uncertain bisimilarity makes it amenable to the use of coalgebraic up-to techniques as developed in [3]. This could be helpful in the development of efficient algorithms for checking uncertain bisimilarity, which could be particularly interesting to check compatibility of ioco specifications as studied in [22].

References

1. Dana Angluin. Learning regular sets from queries and counterexamples. *Inf. Comput.*, 75(2):87–106, 1987.
2. Simone Barlocco, Clemens Kupke, and Jurriaan Rot. Coalgebra learning via duality. In *FoSSaCS*, volume 11425 of *Lecture Notes in Computer Science*, pages 62–79. Springer, 2019.
3. Filippo Bonchi, Daniela Petrisan, Damien Pous, and Jurriaan Rot. A general account of coinduction up-to. *Acta Informatica*, 54(2):127–190, 2017.
4. Thomas Colcombet, Daniela Petrisan, and Riccardo Stabile. Learning automata and transducers: A categorical approach. In *CSL*, volume 183 of *LIPIcs*, pages 15:1–15:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021.
5. Marcelo P. Fiore. A coinduction principle for recursive data types based on bisimulation. *Inf. Comput.*, 127(2):186–198, 1996. doi:10.1006/inco.1996.0058.
6. Herman Geuvers and Bart Jacobs. Relating apartness and bisimulation. *Logical Methods in Computer Science*, Volume 17, Issue 3, July 2021. doi:10.46298/lmcs-17(3:15)2021.
7. Gerco van Heerdt. *CALF: Categorical Automata Learning Framework*. Phd thesis, University College London, October 2020.
8. Claudio Hermida and Bart Jacobs. Structural induction and coinduction in a fibrational setting. *Inf. Comput.*, 145(2):107–152, 1998.
9. Jesse Hughes and Bart Jacobs. Simulations in coalgebra. *Theor. Comput. Sci.*, 327(1-2):71–108, 2004.

10 Bart Jacobs. *Categorical Logic and Type Theory.* Number 141 in Studies in Logic and the Foundations of Mathematics. North Holland, Amsterdam, 1999.

11 Bart Jacobs. *Introduction to Coalgebra: Towards Mathematics of States and Observation*, volume 59 of *Cambridge Tracts in Theoretical Computer Science.* Cambridge University Press, 2016.

12 André Joyal, Mogens Nielsen, and Glynn Winskel. Bisimulation from Open Maps. *Information and Computation*, 127:164–185, 1996. `doi:10.1006/inco.1996.0057`.

13 Clemens Kupke and Dirk Pattinson. Coalgebraic semantics of modal logics: An overview. *Theor. Comput. Sci.*, 412(38):5070–5094, 2011.

14 Clemens Kupke and Jurriaan Rot. Expressive logics for coinductive predicates. *Log. Methods Comput. Sci.*, 17(4), 2021.

15 Paul Blain Levy. Similarity quotients as final coalgebras. In *FoSSaCS*, volume 6604 of *Lecture Notes in Computer Science*, pages 27–41. Springer, 2011.

16 Davide Sangiorgi. *An introduction to Bisimulation and Coinduction.* Cambridge University Press, 2012.

17 Lutz Schröder. Expressivity of coalgebraic modal logic: The limits and beyond. *Theor. Comput. Sci.*, 390(2-3):230–247, 2008.

18 Sam Staton. Relating coalgebraic notions of bisimulation. *Log. Methods Comput. Sci.*, 7(1), 2011.

19 Jan Tretmans. Model based testing with labelled transition systems. In Robert M. Hierons, Jonathan P. Bowen, and Mark Harman, editors, *Formal Methods and Testing, An Outcome of the FORTEST Network, Revised Selected Papers*, volume 4949 of *Lecture Notes in Computer Science*, pages 1–38. Springer, 2008. `doi:10.1007/978-3-540-78917-8_1`.

20 Henning Urbat and Lutz Schröder. Automata learning: An algebraic approach. In *LICS*, pages 900–914. ACM, 2020.

21 Frits Vaandrager, Bharat Garhewal, Jurriaan Rot, and Thorsten Wißmann. A new approach for active automata learning based on apartness. In *Tools and Algorithms for the Construction and Analysis of Systems - 28th International Conference, TACAS 2022*, Lecture Notes in Computer Science. Springer, April 2022.

22 Petra van den Bos, Ramon Janssen, and Joshua Moerman. n-complete test suites for IOCO. *Softw. Qual. J.*, 27(2):563–588, 2019. `doi:10.1007/s11219-018-9422-x`.

23 Thorsten Wißmann, Jérémy Dubut, Shin-ya Katsumata, and Ichiro Hasuo. Path category for free. In Mikołaj Bojańczyk and Alex Simpson, editors, *FoSSaCS 2019*, pages 523–540, Cham, April 2019. Springer International Publishing. `doi:10.1007/978-3-030-17127-8_30`.