Reflection and preservation of properties in coalgebraic (bi)simulations*

Ignacio Fábregas, Miguel Palomino, and David de Frutos Escrig

Departamento de Sistemas Informáticos y Computación, UCM
fabregas@fdi.ucm.es {miguelpt, defrutos}@sip.ucm.es

Abstract. Our objective is to extend the standard results of preservation and reflection of properties by bisimulations to the coalgebraic setting, as well as to study under what conditions these results hold for simulations. The notion of bisimulation is the classical one, while for simulations we use that proposed by Hughes and Jacobs. As for properties, we start by using a generalization of linear temporal logic to arbitrary coalgebras suggested by Jacobs, and then an extension by Kurtz which includes atomic propositions too.

1 Introduction

To reason about computational systems it is customary to mathematically formalize them by means of state-based structures such as labelled transitions systems or Kripke structures. This is a fruitful approach since it allows to study the properties of a system by relating it to some other, possibly better-known system, by means of simulations and bisimulations (see e.g., [15, 14, 12, 3]).

The range of structures used to formalize computational systems is quite wide. In this context, coalgebras have emerged with a unifying aim [18]. A coalgebra is simply a function $c : X \longrightarrow FX$, where X is the set of states and FX is some expression on X (a functor) that describes the possible outcomes of a transition from a given state. Choosing different expressions for F one can obtain coalgebras that correspond to transition systems, Kripke structures, ...

Coalgebras can also be related by means of (bi)simulations. Our goal in this paper is to prove that, like their concrete instantiations, (bi)simulations between arbitrary coalgebras preserve some interesting properties. A first step in this direction consists in choosing an appropriate notion for both bisimulation and simulation, as well as a logic in which to express these properties.

Bisimulations were originally introduced by Aczel and Mendler [1], who showed that the general definition coincided with the standard ones when particularized; it is an established notion. Simulations, on the other hand, were defined by Hughes and Jacobs [8] and lack such canonicity. Their notion of simulation depends on the use of orders that allow (perhaps too) much flexibility in what it can be considered as a simulation; in order to show that simulations preserve

* Research supported by the Spanish projects DESAFIOS TIN2006-15660-C02-01 and PROMESAS S-0505/TIC/0407

properties, we will have to impose certain restrictions on such orders. As for the logic used for the properties, there is likewise no canonical choice at the moment. Jacobs proposes a temporal logic (see [9]) that generalizes linear temporal logic (LTL), though without atomic propositions; a clever insight of Pattinson [17] provides us with a way to endow Jacobs' logic with atomic propositions.

Since our original motivation was the generalization of the results about simulations and preservation of LTL properties, we will focus on Jacobs' logic and its extension with atomic propositions. Actually, modal logic seems to be the right logic to express properties of coalgebras and several proposals have been made in this direction, among them those in [10, 13, 17], which are invariant under behavioral equivalence. The reason for studying preservation/reflection of properties by bisimulations here is twofold: on the one hand, some of the operators in Jacobs' logic do not seem to fall under the framework of those general proposals; on the other hand, some of the ideas and insights developed for that study are needed when tackling simulations. As far as we know, reflection of properties by simulations in coalgebras has not been considered before in the literature.

2 Preliminaries

In this section we summarize definitions and concepts from [8, 11, 9], and introduce the notation we are going to use.

Given a category $\mathbb{C}$ and an endofunctor F in $\mathbb{C}$, an F-coalgebra, or just a coalgebra, consists of an object $X \in \mathbb{C}$ together with a morphism $c : X \longrightarrow FX$. We often call X the state space and c the transition or coalgebra structure.

Example 1. We show how two well-known structures can be seen as coalgebras:

- Labelled transition systems are coalgebras for the functor $F = \mathcal{P}(id)^A$, where A is the set of labels.
- Kripke structures are coalgebras for the functor $F = \mathcal{P}(AP) \times \mathcal{P}(id)$, where AP is a set of atomic propositions.

It is well-known that an arbitrary endofunctor F on **Sets** can be lifted to a functor in the category **Rel** of relations, that is, $\mathrm{Rel}(F) : \mathbf{Rel} \longrightarrow \mathbf{Rel}$. Given a relation $R \subseteq X \times Y$, its lifting is defined by

$$\mathrm{Rel}(F)(R) = \{\langle u, v\rangle \in FX_1 \times FX_2 \mid \exists w \in F(R).\, F(r_1)(w) = u, F(r_2)(w) = v\}\ ,$$

where $r_i : R \longrightarrow X_i$ are the projection morphisms.

A predicate P of a coalgebra $c : X \longrightarrow FX$ is just a subset of the state space. Also, a predicate $P \subseteq X$ can be lifted to a functor structure using the relation lifting:

$$\mathrm{Pred}(F)(P) = \coprod_{\pi_1}(\mathrm{Rel}(F)(\coprod_{\delta}(P))) = \coprod_{\pi_2}(\mathrm{Rel}(F)(\coprod_{\delta}(P))),$$

where $\delta = \langle id, id \rangle$ and $\coprod_f(X)$ is the image of X under f, so $\coprod_{\delta_x}(P) = \{(x,x) \mid x \in P\}$, $\coprod_{\pi_1}(R) = \{x_1 \mid \exists x_2 . x_1 R x_2\}$ is the domain of the relation R, and $\coprod_{\pi_2}(R) = \{x_2 \mid \exists x_1 . x_1 R x_2\}$ is its codomain.

The class of polynomial endofunctors is defined as the least class of endofunctors on **Sets** such that it contains the identity and constant functors, and is closed under product, coproduct, constant exponentiation, powerset and finite sequences. For polynomial endofunctors, $\mathrm{Rel}(F)$ and $\mathrm{Pred}(F)$ can be defined by induction on the structure of F. For further details on these definitions see [9]; we will introduce some of those when needed. For example, for the cases of labelled transition systems and Kripke structures we have:

$$\mathrm{Rel}(\mathcal{P}(id)^A)(R) = \{(f,g) \mid \forall a \in A.\, (f(a), g(a)) \in \{(U,V) \mid \forall u \in U.\, \exists v \in V.\, uRv \wedge \forall v \in V.\, \exists u \in U.\, uRv\}\}$$

$$\mathrm{Pred}(\mathcal{P}(id)^A)(P) = \{f \mid \forall a \in A.\, f(a) \in \{U \mid \forall u \in U.\, Pu\}\}$$

$$\mathrm{Rel}(\mathcal{P}(AP) \times \mathcal{P}(id))(R) = \{((u_1,u_2),(v_1,v_2)) \mid (u_1 = v_1.\, u_1, v_1 \in \mathcal{P}(AP)) \wedge (u_2,v_2) \in \{(U,V) \mid \forall u \in U.\, \exists v \in V.\, uRv \wedge \forall v \in V.\, \exists u \in U.\, uRv\}\}$$

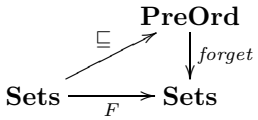
$$\mathrm{Pred}(\mathcal{P}(AP) \times \mathcal{P}(id))(P) = \{(u,v) \mid (u \subseteq \mathcal{P}(AP)) \wedge (v \in \{U \mid \forall u \in U.\, Pu\}\}$$

A bisimulation for coalgebras $c : X \longrightarrow FX$ and $d : Y \longrightarrow FY$ is a relation $R \subseteq X \times Y$ which is "closed under c and d":

$$\text{if } (x,y) \in R \text{ then } (c(x), d(y)) \in \mathrm{Rel}(F)(R)\,.$$

In the same way, an invariant for a coalgebra $c : X \longrightarrow FX$ is a predicate $P \subseteq X$ such that it is "closed under c", that is, if $x \in P$ then $c(x) \in \mathrm{Pred}(F)(P)$.

We will use the definition of simulation introduced by Hughes and Jacobs in [8] which uses an order $\sqsubseteq$ for functors F that makes the following diagram commute



Given an order $\sqsubseteq$ on F, a simulation for the coalgebras $c : X \longrightarrow FX$ and $d : Y \longrightarrow FY$ is a relation $R \subseteq X \times Y$ such that

$$\text{if } (x,y) \in R \text{ then } (c(x), d(y)) \in \mathrm{Rel}(F)_{\sqsubseteq}(R)\,,$$

where $\mathrm{Rel}(F)_{\sqsubseteq}(R)$ is defined as

$$\mathrm{Rel}(F)_{\sqsubseteq}(R) = \{(u,v) \mid \exists w \in F(R).\, u \sqsubseteq Fr_1(w) \wedge Fr_2(w) \sqsubseteq v\}\,.$$

To express properties we will use a generalization of LTL proposed by Jacobs (see [9]) that applies to arbitrary coalgebras, whose formulas are given by the following BNF expression:

$$\varphi = P \subseteq X \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \varphi \Rightarrow \varphi \mid \bigcirc\varphi \mid \Diamond\varphi \mid \Box\varphi \mid \varphi\, \mathcal{U}\, \varphi$$

$\bigcirc$ is the *nexttime* operator and its semantics (abusing notation) is defined as $\bigcirc P = c^{-1}(\mathrm{Pred}(F)(P)) = \{x \in X \mid c(x) \in \mathrm{Pred}(F)(P)\}$; $\Box$ is the *henceforth* operator defined as $\Box P$ if exists an invariant for c, such that $Q \subseteq P$ with $x \in Q$ or, equivalently by means of the greatest fixed point ν, $\Box P = \nu S.(P \wedge \bigcirc S)$; $\Diamond$ is the *eventually* operator defined as $\Diamond P = \neg\Box\neg P$; and $\mathcal{U}$ is the *until* operator defined as $P\,\mathcal{U}\,Q = \mu S.(Q \vee (P \wedge \neg \bigcirc \neg S))$, where μ is the least fixed point.

We denote the set of states in X that satisfies φ as $[\![\varphi]\!]_X$. That is, if $P \subseteq X$ is a predicate, then $[\![P]\!]_X = P$; if $\alpha \in \{\neg, \bigcirc, \Box, \Diamond\}$ then $[\![\alpha\varphi]\!]_X = \alpha[\![\varphi]\!]_X$, and if $\beta \in \{\wedge, \vee, \Rightarrow, \mathcal{U}\}$ then $[\![\varphi_1 \beta \varphi_2]\!]_X = [\![\varphi_1]\!]_X \beta [\![\varphi_2]\!]_X$. We will usually omit the reference to the set X when it is clear from the context. We say that an element x satisfies a formula φ, and we denote it by $c, x \models \varphi$, when $x \in [\![\varphi]\!]$. Again, we will usually omit the reference to the coalgebra c.

3 Reflection and preservation in bisimulations

These definitions of reflection and preservation are slightly more involved than for classical LTL because the logic proposed by Jacobs does not use atomic propositions, but predicates (subsets of the set of states). Later, we will see how atomic propositions can be introduced in the logic.

Given a predicate P on X and a binary relation $R \subseteq X \times Y$, we will say that an element $y \in Y$ is in the direct image of P, and we will denote it by $y \in RP$, if there exists $x \in X$ with $x \in P$ and xRy. The inverse image of R is just the direct image for the relation R^{-1}.

Definition 1. *Given two formulas φ on X and ψ on Y, built over predicates $P_1, \ldots P_n$ and $Q_1, \ldots Q_n$, respectively, and a binary relation $R \subseteq X \times Y$, we define the image of φ as a formula φ^* on Y, obtained by substituting in φ RP_i for P_i. Likewise, we build ψ^{-1}, the inverse of ψ, substituting $R^{-1}Q_i$ for Q_i in ψ.*

Remark 1. It is important to notice that φ^* coincides with φ^{-1} when we consider R^{-1} instead of R. Analogously, φ^{-1} is just φ^* when we consider R^{-1} instead of R.

Now we can define when a relation preserves or reflects properties.

Definition 2. *Let $R \subseteq X \times Y$ be a binary relation and a and b elements such that aRb. We say that R* preserves *the property φ on X if, whenever $a \models \varphi$, $b \models \varphi^*$. We say that R* reflects *the property φ on Y if $b \models \varphi$ implies $a \models \varphi^{-1}$.*

Let us first state a couple of technical lemmas whose proofs appear in [6].

Lemma 1. *Let F be a polynomial functor, $R \subseteq X \times Y$ a bisimulation between coalgebras $c : X \longrightarrow FX$ and $d : Y \longrightarrow FY$, $P \subseteq Y$, $Q \subseteq X$ and xRy. If $d(y) \in \mathrm{Pred}(F)(P)$, then $c(x) \in \mathrm{Pred}(F)(R^{-1}P)$; and if $c(x) \in \mathrm{Pred}(F)(Q)$, then $d(y) \in \mathrm{Pred}(F)(RQ)$.*

Another auxiliary lemma we need to prove the main result of this section is the following:

Lemma 2. *The direct and inverse images of an invariant are also invariants.*

Proof. Let R be a bisimulation between $c : X \longrightarrow FX$ and $d : Y \longrightarrow FY$. Let us suppose that $P \subseteq X$ is an invariant and let us prove that so is RP; that is, for all $y \in RP$ it must be the case that $d(y) \in \mathrm{Pred}(F)(RP)$. If $y \in RP$, then there exists $x \in P$ such that xRy. Since P is an invariant, we also have $c(x) \in \mathrm{Pred}(F)(P)$ and by Lemma 1 we get $d(y) \in \mathrm{Pred}(F)(RP)$.

On the other hand, since R^{-1} is also a bisimulation, the inverse image of an invariant is an invariant too. □

At this point it is interesting to recall that our objective is to prove that bisimulations preserve and reflect properties of a temporal logic, that is, if we have xRy and $y \models \varphi$ then we must also have $x \models \varphi^{-1}$; and, analogously, if $x \models \varphi$ then $y \models \varphi^*$. We will show this result for all temporal operators except for the negation; it is well-known that negation is reflected and preserved by standard bisimulations, but not here because of the lack of atomic propositions in the coalgebraic temporal logic.

To prove the result for the rest of temporal operators, we will see that if $y \in [\![\varphi]\!]$ then we also have $x \in R^{-1}[\![\varphi]\!]$ and, analogously, if $x \in [\![\varphi]\!]$ then $y \in R[\![\varphi]\!]$. Ideally, we would like to have both $R^{-1}[\![\varphi]\!] = [\![\varphi^{-1}]\!]$ and $R[\![\varphi]\!] = [\![\varphi^*]\!]$ but, in general, only the inclusion $\subseteq$ is true. Fortunately this is enough to prove our propositions, since the temporal operators are all monotonic except for the negation. In fact, here is where the problem with negation appears.

Lemma 3 ([6]). *Let R be a bisimulation between coalgebras $c : X \longrightarrow FX$ and $d : Y \longrightarrow FY$. For all temporal formulas φ and ψ which do not contain the negation operator, it follows that*

$$R^{-1}[\![\varphi]\!]_Y \subseteq [\![\varphi^{-1}]\!]_X \quad \textit{and} \quad R[\![\psi]\!]_X \subseteq [\![\psi^*]\!]_Y \,.$$

Finally we can show that bisimulations reflect and preserve properties given by any temporal operator except for the negation.

Proposition 1. *Let ψ be a formula over a set Y which does not use the* negation *operator and let R be a bisimulation between coalgebras $c : X \longrightarrow FX$ and $d : Y \longrightarrow FY$. Then the property ψ is reflected by R.*

Proof. The result is proved by structural induction over the formula ψ using the first half of Lemmas 1 and 3, and Lemma 2. See [6] for further details. □

Preservation of properties is a consequence of the reflection of properties together with the fact that if R is a bisimulation then R^{-1} is also a bisimulation. We have thus proved the following theorem.

Theorem 1. *Let ψ and φ be formulas over sets Y and X, respectively, which do not use the* negation *operator and let R be a bisimulation between coalgebras $c : X \longrightarrow FX$ and $d : Y \longrightarrow FY$. Then ψ is reflected by R and φ is preserved by R.*

4 Reflection and preservation in simulations

In [3, 16] it is proved not only that bisimulations reflect and preserve properties but also that simulations reflect them: it turns out that this result does not generalize straightforwardly to the coalgebraic setting.

The main problem that we have found concerning this is that the coalgebraic definition of simulation uses an arbitrary functorial order $\sqsubseteq$, and in general reflection of properties will not hold for all orders.

Let us show a counterexample that will convince us that simulations may not reflect properties without restricting the orders. Let us take $F = \mathcal{P}(id)$, $X = \{x_1, x_2\}$, $Y = \{y_1, y_2\}$ and the coalgebras c and d defined as $c(x_1) = \{x_1, x_2\}$, $c(x_2) = \{x_2\}$, $d(y_1) = y_2$ and $d(y_2) = y_2$. We define $u \sqsubseteq v$ whenever $v \subseteq u$ and consider the formula $\varphi = \bigcirc P$, where $P = \{y_2\}$, and the simulation $R = \{(x_1, y_2)\}$. It is immediate to check that R is a simulation and $y_2 \in [\![\varphi]\!]$, but $x_1 \notin [\![\varphi^{-1}]\!]$.

- $y_2 \in [\![\varphi]\!]$. Indeed, since $d(y_2) = y_2$ then $y_2 \in [\![\varphi]\!] = \bigcirc P$ is equivalent to $y_2 \in P = \{y_2\}$, which is trivially true.
- $x_1 \notin [\![\varphi^{-1}]\!]$. By definition, $\varphi^{-1} = \bigcirc R^{-1} P = \bigcirc\{x_1\}$. Since $c(x_1) = \{x_1, x_2\}$, it is enough to see that $x_2 \notin \{x_1\}$, which is also true.

As a consequence, we will need to restrict the functorial orders that are involved in the definition of simulation. In a first approach we will impose an extra requirement that the order must fulfill, and later we will not only restrict the orders but also the functors that are involved.

4.1 Restricting the orders

The idea is that we are going to require an extra property for each pair of elements which are related by the order. In particular, we are particularly interested in the following property (which is defined in [8]):

Definition 3. *Given a functor F : **Sets** $\longrightarrow$ **Sets**, we say that an order $\sqsubseteq$ associated to it is "down-closed" whenever $a \sqsubseteq b$, with $a, b \in FX$, implies that*

$$b \in \mathrm{Pred}(F)(P) \implies a \in \mathrm{Pred}(F)(P), \quad \textit{for all predicates } P \subseteq X\,.$$

We can show some examples of down-closed orders:

Example 2. 1. Kripke structures are defined by the functor $F = \mathcal{P}(AP) \times \mathcal{P}(id)$, so a down-closed order must fulfill that if $(u, v) \sqsubseteq (u', v')$, then $(u', v') \in \mathrm{Pred}(F)(P)$ implies $(u, v) \in \mathrm{Pred}(F)(P)$; that is, by definition of $\mathrm{Pred}(\mathcal{P}(AP) \times \mathcal{P}(id))$, $u, u' \subseteq \mathcal{P}(AP)$ and, if $v' \in \mathrm{Pred}(\mathcal{P}(id))(P) = \{U \mid \forall u \in U.\, u \in P\}$ then $v \in \mathrm{Pred}(\mathcal{P}(id))(P)$. In other words, for all $b \in v$ and $b' \in v'$, if $b' \in P$ then $b \in P$. Therefore, what is needed in this case is that the set of successors v of the smaller pair is contained in the set of successors v' of the bigger pair, that is, if $(u, v) \sqsubseteq (u', v')$ then $v \subseteq v'$.

2. Labelled transition systems are defined by the functor $F = \mathcal{P}(id)^A$, so the order must fulfill the following: if $u \sqsubseteq v$ then $\forall a \in A.\ u(a) \subseteq u'(a)$.

Those examples show that there are not many down-closed orders, but it does not seem clear how to further extend this class in such a way that we could still prove the reflection of properties by simulations. Unfortunately, even under this restriction we can only prove reflection (or preservation) of formulas that only use the operators $\vee$, $\wedge$, $\bigcirc$ and $\Box$.

To convince us of this fact, we present a counterexample with operator $\Diamond$. Let $X = \{x_1, x_2\}, Y = \{y_1, y_2\}$ and the functor $F = \mathcal{P}(id)$. We consider the following down-closed order: $u \sqsubseteq v$ if $u \subseteq v$. We also define the coalgebras $c : X \longrightarrow FX$ and $d : Y \longrightarrow FY$ as $c(x_1) = \{x_1\}$, $c(x_2) = \{x_2\}$, $d(y_1) = \{y_1, y_2\}$ and $d(y_2) = \{y_2\}$. Obviously $R = \{(x_1, y_1)\}$ is a simulation since $c(x_1) = \{x_1\} \sqsubseteq \{x_1\}$ and $\{y_1\} \sqsubseteq \{y_1, y_2\} = d(y_1)$ and, also, $\{x_1\}\mathrm{Rel}(F)(R)\{y_1\}$. We have $y_1 \in \Diamond\{y_2\}$, since we can reach y_2 from y_1, but $x_1 \notin \Diamond R^{-1}\{y_2\} = \Diamond\emptyset$. Indeed, $x_1 \notin \Diamond\emptyset$ is equivalent to $x_1 \in \Box\neg\emptyset$ and this is true since $\{x_1\}$ is an invariant such that $x_1 \in \{x_1\}$, with $\{x_1\} \subseteq \neg\emptyset$.

In order to prove reflection of properties that only use the operators $\vee$, $\wedge$, $\bigcirc$ and $\Box$, we will need a previous elementary result involving binary relations.

Proposition 2. *Let $R \subseteq X \times Y$ be a binary relation and $P \subseteq Y$ a predicate. Let us suppose that $u\mathrm{Rel}(F)(R)v$; then, if $v \in \mathrm{Pred}(F)(P)$ it is also true that $u \in \mathrm{Pred}(F)(R^{-1}P)$.*

Proof. Once again the proof will proceed by structural induction on the functor F. See [6] for further details. □

We will also need a subtle adaptation of Lemmas 2 and 3 from the framework of bisimulations to the framework of simulations. In particular, we can adapt Lemma 2 to prove that if Q is an invariant and R a simulation, $R^{-1}Q$ is still an invariant, whereas the first half of Lemma 3 will also be true in the framework of simulations for formulas that only use the operators $\vee$, $\wedge$, $\bigcirc$ and $\Box$.

Lemma 4. *Let R be a simulation between coalgebras $c : X \longrightarrow FX$ and $d : Y \longrightarrow FY$, with a down-closed order, and let $Q \subseteq Y$ be an invariant. Then $R^{-1}Q$ is also an invariant.*

Proof. We are going to show that for all $x \in R^{-1}Q$ we have $c(x) \in \mathrm{Pred}(F)(R^{-1}Q)$. Let us take an arbitrary $x \in R^{-1}Q$; then, by definition there exists $y \in Q$ such that xRy and, since Q is an invariant, $d(y) \in \mathrm{Pred}(F)(Q)$. On the other hand, since R is a simulation, $c(x) \sqsubseteq u\mathrm{Rel}(F)(R)v \sqsubseteq d(y)$. Henceforth, since we are working with a down-closed order and $d(y) \in \mathrm{Pred}(F)(Q)$, then $v \in \mathrm{Pred}(F)(Q)$. Also, by Proposition 2 we have $u \in \mathrm{Pred}(F)(R^{-1}Q)$ and, using again that the order is down-closed, it follows that $c(x) \in \mathrm{Pred}(F)(R^{-1}Q)$. □

Lemma 5 ([6]). *Let R be a simulation between coalgebras $c : X \longrightarrow FX$ and $d : Y \longrightarrow FY$, with a down-closed order. If φ is a temporal formula constructed only with operators $\vee$, $\wedge$, $\bigcirc$ and $\Box$, then*

$$R^{-1}[\![\varphi]\!]_Y \subseteq [\![\varphi^{-1}]\!]_X .$$

Now we can state the corresponding theorem:

Theorem 2 ([6]). *Let R be a simulation between coalgebras $c : X \longrightarrow FX$ and $d : Y \longrightarrow FY$ with a down-closed order. If φ is a temporal formula constructed only with operators $\vee$, $\wedge$, $\bigcirc$ and $\Box$, then the property φ is reflected by the simulation.*

Instead of considering down-closed orders, we could have imposed the converse implication, that is, those orders that satisfy that if $a \in \mathrm{Pred}(F)(P)$ then $b \in \mathrm{Pred}(F)(P)$.

Definition 4. *Given a functor F : **Sets** $\longrightarrow$ **Sets** we say that an order $\sqsubseteq$ is* up-closed *if whenever $a \sqsubseteq b$ then*

$$a \in \mathrm{Pred}(F)(P) \implies b \in \mathrm{Pred}(F)(P), \quad \textit{for all predicates } P\,.$$

Obviously up-closed is symmetrical to down-closed, that is, it is equivalent to taking $\sqsubseteq^{op}$ instead of $\sqsubseteq$ in Definition 3. So, for example, in the case of Kripke structures an up-closed order would satisfy $(u, v) \sqsubseteq (u', v')$ if $v' \subseteq v$.

The interesting thing about up-closed orders is that they allow us to prove *preservation* of properties; again, this result will hold only for formulas constructed with the operators $\vee$, $\wedge$, $\bigcirc$ and $\Box$. We need the following auxiliary result whose proof is analogous to the case of down-closed orders. Since if R is a simulation for the order $\sqsubseteq$, then R^{-1} is a simulation for the oposite order $\sqsubseteq^{op}$, we can apply Theorem 2 to get the following (see [6] for more details):

Theorem 3. *Let R be a simulation between coalgebras $c : X \longrightarrow FX$ and $d : Y \longrightarrow FY$ carrying an up-closed order. If φ is a temporal formula constructed only with the operators $\vee$, $\wedge$, $\bigcirc$ and $\Box$, then R preserves the property φ.*

4.2 Restricting the class of functors

As we have just seen, it is not enough to restrict ourselves to down-closed (or up-closed) orders to get a valid result for all properties. What we want is a necessary and sufficient condition over functorial orders that implies reflection (or preservation) of properties by simulations. So far we have not found such a condition, but we have a sufficient one for simulations to reflect properties (and, in fact, also so that they preserve properties).

Recalling the structure of lemmas and propositions used to prove reflection and preservation of properties by bisimulations, we notice that the key ingredient was Lemma 1. With this lemma we were able to prove directly preservation of invariants (Lemma 2) and the relation between R^{-1} (respectively R) of a formula and the inverse of a formula (respectively direct image of a formula). Also, Lemma 1 was essential to prove directly reflection and preservation of formulas built with the *nexttime* operator and the rest of temporal operators.

In the previous section the problem we faced was that either the second half of Lemma 1 (for down-closed orders) or the first half of Lemma 1 (for up-closed

orders) held, but not both simultaneously. As a consequence, the results for the operators *eventually* and *until* did not hold. So, if we were capable of finding a subclass of functors and orders such that they fulfill results analogous to Lemma 1 then, translating those proofs, we would get reflection and preservation of arbitrary properties.

We are going to define a subclass of functors and orders in the way that Hughes and Jacobs did in [8] for the subclass **Poly**.

Definition 5. *The class* **Order** *is the least class of functors closed under the following operations:*

1. *For every preorder $(A, \leq)$, the constant functor $X \mapsto A$ with the order given by $\sqsubseteq_X = \leq_A$.*
2. *The identity functor with equality order.*
3. *Given two polynomial functors F_1 and F_2 with orders $\sqsubseteq^1$ and $\sqsubseteq^2$, the product functor $F_1 \times F_2$ with order $\sqsubseteq_X$ given by*

$$(u, v) \sqsubseteq_X (u', v') \quad \text{if} \quad u \sqsubseteq^1 u' \quad \text{and} \quad v \sqsubseteq^2 v' .$$

4. *Given the polynomial functor F with order $\sqsubseteq^F$ and the set A, the functor F^A with order $\sqsubseteq_X$ given by*

$$u \sqsubseteq_X v \text{ if } u(a) \sqsubseteq^F v(a) \text{ for all } a \in A.$$

5. *Given two polynomial functors F_1 and F_2 with orders $\sqsubseteq^1$ and $\sqsubseteq^2$, the coproduct functor $F_1 + F_2$ with order $\sqsubseteq_X$ given by*

$$\begin{aligned} u \sqsubseteq_X v \text{ if } & u = \kappa_1(u_0) \text{ and } v = \kappa_1(v_0) \text{ with } u_0 \sqsubseteq^1 v_0 \\ \text{or } & u = \kappa_2(u_0) \text{ and } v = \kappa_2(v_0) \text{ with } u_0 \sqsubseteq^2 v_0 . \end{aligned}$$

6. *Given the polynomial functor F with order $\sqsubseteq^F$, the powerset functor $\mathcal{P}(F)$ with order $\sqsubseteq_X$ given by*

$$\begin{aligned} u \sqsubseteq_X v \quad \text{if} \quad & \forall a \in u \; \exists b \in v \quad \text{such that} \quad a \sqsubseteq^F b \\ \text{and also } & \forall b \in v \; \exists a \in u \quad \text{such that} \quad a \sqsubseteq^F b . \end{aligned}$$

For example the usual order for Kripke structures is not in the class **Order**. Besides, in the definition of **Poly** in [8] the authors did not consider the powerset functor but we do, although we are not using the *usual* order for this functor.

At first, to obtain that simulations not only reflect but also preserve properties may seem a little surprising. If we think about the elements in the subclass **Order** we notice that we have restricted the orders to equality-like orders, that is, almost all possible orders in **Order** are the equality. However, since the class **Order** is very similar to the class **Poly**, it has the same good properties shown in [8] (like the stablility of the orders and functors).

Example 3. 1. If we consider the functor $\mathcal{P}(id)$, then the order $\sqsubseteq$ defined in Definition 5 says that $u \sqsubseteq v$ if and only if for each $a \in u$ there exists $b \in v$ such that $a = b$, and if for each $b \in v$ there exists $a \in u$ such that $a = b$. This means that $\sqsubseteq$ is the identity relation. As an immediate consequence for transition systems the only possible **Order** simulations are bisimulations.

2. If we consider the functor $A \times id$ where A has a preorder $\leq_A$ different from the identity, the order $\sqsubseteq$ from Definition 5 is the following: $(u, v) \sqsubseteq (u', v')$ iff $v = v'$ and $u \leq_A u'$. So, if $\leq_A$ is not the identity, neither is $\sqsubseteq$. For example, let us take $X = \{x_1, x_2, x_3\}$, $Y = \{y_1, y_2\}$, $AP = \{p_1, p_2, p_3\}$ and consider the functor $F = \mathcal{P}(id) \times \mathcal{P}(AP)$ and the coalgebras $c : X \longrightarrow FX$ and $d : Y \longrightarrow FY$ defined by $c(x_1) = (\{x_2, x_3\}, \{p_1\})$, $c(x_2) = (\{x_3\}, \{p_2\})$, $c(x_3) = (\{x_2\}, \{p_3\})$, $d(y_1) = (\{y_2\}, \{p_2\})$ and $d(y_2) = (\{y_2\}, \{p_1\})$. Obviously there is no bisimulation between x_1 and y_1 since this atomic propositions are not the same, but taking the order $\sqsubseteq$ defined as $(u, v) \sqsubseteq (u', v')$ iff $u = u'$ (that is, taking as the preorder $\leq_{AP}$ the total relation) we have that there exists a simulation R in **Order** between x_1 and y_1.

Lemma 6 ([6]). *Let $R \subseteq X \times Y$ be a simulation between coalgebras $c : X \longrightarrow FX$ and $d : Y \longrightarrow FY$, such that the functor F is in the class* **Order**. *Let us also suppose that $P \subseteq Y$ and xRy; then, if $d(y) \in \mathrm{Pred}(F)(P)$ we have $c(x) \in \mathrm{Pred}(F)(R^{-1}P)$.*

In a similar way we have the corresponding lemma involving direct predicates.

Lemma 7. *Let $R \subseteq X \times Y$ be a simulation between coalgebras $c : X \longrightarrow FX$ and $d : Y \longrightarrow FY$, such that the functor F is in* **Order**. *Let us suppose also that $P \subseteq X$ and xRy. Then, if $c(x) \in \mathrm{Pred}(F)(P)$, $d(y) \in \mathrm{Pred}(F)(RP)$.*

Now we can conclude that under these hypothesis simulations reflect and preserve properties, simultaneously! This fact is a straightforward result from Lemmas 6 and 7.

Theorem 4. *Let R be a simulation between coalgebras $c : X \longrightarrow FX$ and $d : Y \longrightarrow FY$, with F a polynomial functor in the class* **Order**. *Then, the simulation R reflects and preserves properties.*

5 Including atomic propositions

A consequence of the fact that the logic proposed by Jacobs does not introduce atomic propositions was the need of giving non-standard definitions of reflection and preservation of properties. Kurz, in his work [13] includes atomic propositions in a temporal logic for coalgebras by means of natural transformations.

Definition 6. *Given a set AP of atomic propositions, the formulas of the temporal logic associated to a coalgebra $c : X \longrightarrow FX$ are given by the BNF expression:*

$$\varphi = p \mid \neg\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \varphi \Rightarrow \varphi \mid \bigcirc\varphi \mid \Diamond\varphi \mid \Box\varphi \mid \varphi \,\mathcal{U}\, \varphi$$

where $p \in AP$ is an atomic proposition.

Kurz also defines when a state x satisfies an atomic proposition p, that is, he defines the semantics of an atomic proposition.

Definition 7. *Let* F : **Sets** $\longrightarrow$ **Sets** *be a functor and* AP *a set of atomic propositions. Let* $\nu : F \Rightarrow \mathcal{P}(AP)$ *be a natural transformation and* $c : X \longrightarrow FX$ *a coalgebra. We say that* x *satisfies an atomic proposition* $p \in AP$*, and denote it* $x \models p$*, when* $p \in (\nu_X \circ c)(x)$*. This way* $[\![p]\!] = \{x \mid p \in (\nu_X \circ c)(x)\}$*.*

Notice that in fact this defines not only a semantics but a family of possible semantics that depends on the natural transformation. For example, we can define a natural transformation for the functor for Kripke structures in this way:

$$\begin{array}{ccc} \nu_X : \mathcal{P}(AP) \times \mathcal{P}(X) & \longrightarrow & \mathcal{P}(AP) \\ (P, Q) & \longmapsto & P \end{array}$$

With ν_X we have characterized the standard semantics of LTL for Kripke structures. Analogously, we could define the following interpretation: $\nu'_X(P, Q) = \mathcal{P}(AP) \setminus P$.

Introducing in our temporal logic the semantics of the atomic propositions, we can prove the following theorem involving bisimulations:

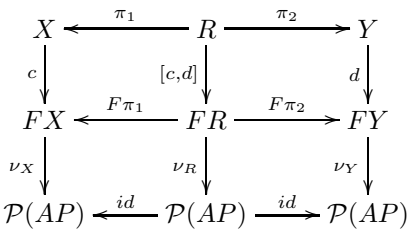
Theorem 5. *Let* R *be a bisimulation between coalgebras* $c : X \longrightarrow FX$ *and* $d : Y \longrightarrow FY$*. Let* φ *be a temporal formula; then, the following is true for all* $x \in X$ *and* $y \in Y$ *such that* xRy*:*

$$x \in [\![\varphi]\!]_X \iff y \in [\![\varphi]\!]_Y \,.$$

Here we have captured in the same theorem the classical ideas of reflection and preservation of properties: we have some property in the lefthand side of a bisimulation if and only if we have the property in its righthand side. In this case the theorem is true also for the negation operator thanks to the atomic propositions. Intuitively, this is because now we have an "if and only if" theorem, whereas in Theorem 1 we needed to reason separately for each implication using monotonicity, and negation lacks it. Also notice that even though we could think that in Theorem 1 our predicates played the role of atomic propositions, there are some essential differences: first, predicates are not independent of each other, unlike atomic propositions, and secondly, while atomic propositions stay the same predicates vary with each set of states.

Proof. Once again the proof will proceed by structural induction on the formula φ. We only show some of the cases (the complete proof can be found in [6]).

1. Let $\varphi = p$ where p is an arbitrary atomic proposition. This way we have the following diagram, for ν an arbitrary natural trasformation:



This diagram is commutative. Indeed, since R is a bisimulation the upper side commutes, while the lower side commutes because ν is a natural transformation.
So, $x \in [\![\varphi]\!]_X$ means by definition that $p \in (\nu_X \circ c)(x)$. Since the diagram commutes then $p \in (\nu_R \circ [c,d])(x,y) \Leftrightarrow p \in (\nu_Y \circ d)(y)$, that is, $y \in [\![\varphi]\!]_Y$.

2. Let us suppose $\varphi = \neg\varphi_0$. In this case we must show that $x \in \neg[\![\varphi_0]\!]_X$ if and only if $y \in \neg[\![\varphi_0]\!]_Y$, that is, we must see that $x \notin [\![\varphi_0]\!]_X$ if and only if $y \notin [\![\varphi_0]\!]_Y$. By induction hypothesis we have $x \in [\![\varphi_0]\!]_X$ if and only if $y \in [\![\varphi_0]\!]_Y$.
3. Let us suppose now that $\varphi = \bigcirc\varphi_0$. We must prove that $x \in \bigcirc[\![\varphi_0]\!]_X$ is equivalent to $y \in \bigcirc[\![\varphi_0]\!]_Y$, that is, $c(x) \in \mathrm{Pred}(F)([\![\varphi_0]\!]_X)$ is equivalent to $d(y) \in \mathrm{Pred}(F)([\![\varphi_0]\!]_Y)$. The latter will be proved by structural induction on the functor F. As an example we show the case of $F = G^A$. Let us prove only one implication since the other one is almost identical. We have

$$\mathrm{Pred}(F)([\![\varphi_0]\!]_X) = \{f \mid \forall a \in A.\ f(a) \in \mathrm{Pred}(G)([\![\varphi_0]\!]_X)\}\,.$$

Once again, as we have shown in other proofs, we define for each $a \in A$ and each F-coalgebra $c : X \longrightarrow F(X)$ a G-coalgebra, $c^a : X \longrightarrow G(X)$ where for each $x \in X$ we have $c^a(x) = c(x)(a)$. In this way, we have xRy and $c^a(x) = c(x)(a) \in \mathrm{Pred}(G)([\![\varphi_0]\!]_X)$. By induction hypothesis we have that $d^a(y) \in \mathrm{Pred}(G)([\![\varphi_0]\!]_Y)$. Since this is a valid argument for all $a \in A$, we obtain $d(y) \in \mathrm{Pred}(F)([\![\varphi_0]\!]_Y)$.

4. $\varphi = \Box\varphi_0$. Assuming that $x \in [\![\varphi]\!]_X$ we get that there exists

$$Q \subseteq X \text{ an invariant for } c \text{ with } Q \subseteq [\![\varphi_0]\!]_X \text{ and } x \in Q.$$

Now, RQ is a invariant for d and, also, such that $RQ \subseteq [\![\varphi_0]\!]_Y$ with $y \in RQ$. Indeed, if $x \in Q$ then $y \in RQ$ and if $b \in RQ$ there must exists some $a \in Q \subseteq [\![\varphi_0]\!]_X$ such that aRb. So, by induction hypothesis we get that $b \in [\![\varphi_0]\!]_Y$
On the other hand, if $y \in [\![\varphi]\!]_Y$ there must exists some invariant T on Y, such that $T \subseteq [\![\varphi_0]\!]_Y$ with $y \in T$, hence for proving $x \in [\![\varphi]\!]_X$ it is enough to consider the invariant $R^{-1}T$. □

To obtain a similar result for simulations, we will need again to restrict the class of functors and orders as we did in Sections 4.1 and 4.2. In particular we are interested in the following antimonotonicity property: if $u \sqsubseteq u'$ then $\nu(u') \subseteq \nu(u)$.

Definition 8. *Let F : **Sets** $\longrightarrow$ **Sets** be a functor, AP a set of atomic propositions and $\nu : F \Rightarrow \mathcal{P}(AP)$ a natural transformation. We say that $\sqsubseteq$ is a down-natural ν-order if, whenever $u \sqsubseteq u'$ then $\nu(u') \subseteq \nu(u)$.*

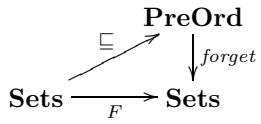
Obviously this definition depends on the natural transformation that we consider in each case. For example, for Kripke structures we have the following natural transformation: $\nu_X((A_X, B_X)) = A_X \subseteq AP$. To obtain a down-natural

ν-order the following must hold: $(u, v) \sqsubseteq (u', v')$ then $\nu((u', v')) \subseteq \nu((u, v))$, that is, it will be enough to require $(u, v) \sqsubseteq (u', v')$ iff $u' \subseteq u$.

This way, if we combine the down-closed and the down-natural orders we get:

$$\text{If} \quad (u, v) \sqsubseteq (u', v') \quad \text{then} \quad u' \subseteq u \text{ and } v \subseteq v' .$$

This characterization is not as restrictive as one could think. Indeed, if we recall the definition of functorial order we had:



This diagram means that the functor F and the order $\sqsubseteq$ almost have the same structure and indeed, we could use a natural transformation between $\sqsubseteq$ and $\mathcal{P}(AP)$ in Definition 7 instead of a natural transformation between F and $\mathcal{P}(AP)$, that is, $\nu :\sqsubseteq \Rightarrow \mathcal{P}(AP)$. Considering ν in this way, an immediate consequence is that if we take as order in $\mathcal{P}(AP)$ the relation $\supseteq$ (as is done in [16]), then $u \sqsubseteq v$ implies $\nu(u) \sqsubseteq \nu(v)$.

We can tackle the proof of reflection of properties (with atomic propositions) by simulations as we did in Section 4.1, imposing to the order not only to be down-natural but also down-closed. But, if we do that we will find the same difficulties we faced in Section 4.1 (that is, we would not be able to prove reflection of formulas built with the operators *until* and *eventually*). Therefore, we must restrict the class of functors and orders, as we did with the class **Order** in Section 4.2, but imposing also that the orders must be down-natural.

Definition 9. *The class* **Down-Natural ν-Order** *is the subclass of* **Order** *where all orders are down-natural.*

Notice that we are defining a different class for each natural transformation ν. Under this condition we state the corresponding theorem involving simulations and the reflection of properties (with atomic propositions); for the proof see [6].

Theorem 6. *Let R be a simulation between coalgebras $c : X \longrightarrow FX$ and $d : Y \longrightarrow FY$ on the same polynomial functor F from* **Sets** *to* **Sets** *belonging to the class* **Down-Natural ν-Order** *and let φ be a temporal formula. Then, for each $x \in X$ and $y \in Y$ such that xRy:*

$$y \in [\![\varphi]\!]_Y \quad \Longrightarrow \quad x \in [\![\varphi]\!]_X .$$

We showed above that simulations for functors in the class **Order** reflected and preserved all kinds of properties. Instead, now we can only prove one implication, that corresponding to the reflection of properties. This is so because down-natural ν-orders have a natural direction.

Exactly in the same way as we did with down-natural ν-orders, we can define the corresponding class of up-natural ν-orders:

Definition 10. *Let $F : \mathbf{Sets} \longrightarrow \mathbf{Sets}$ be a functor, AP a set of atomic propositions and $\nu : F \Rightarrow \mathcal{P}(AP)$ a natural transformation. We say that $\sqsubseteq$ is an up-natural ν-order if $u \sqsubseteq u'$ implies $\nu(u) \subseteq \nu(u')$.*

As we did for down-natural ν-orders, we define a subclass of **Order**:

Definition 11. *The class* **Up-Natural ν-Order** *is the subclass of* **Order** *where all orders are up-natural.*

Theorem 7. *Let R be a simulation between coalgebras $c : X \longrightarrow FX$ and $d : Y \longrightarrow FY$ on the same polynomial functor F in the class* **Up-Natural ν-Order**, *and let φ be a temporal formula. Then, for all $x \in X$ and $y \in Y$ such that xRy:*

$$x \in [\![\varphi]\!]_X \quad \Longrightarrow \quad y \in [\![\varphi]\!]_Y \, .$$

6 Conclusions

The main goal of this paper was to study under what assumptions coalgebraic simulations reflect properties. In our way towards the proof of this result, we were also able to prove reflection and preservation of properties by coalgebraic bisimulations. For expressing the properties we used Jacobs' temporal logic [9], later extended with atomic propositions using the idea presented in [13].

That coalgebraic bisimulations reflect and preserve properties expressed in modal logic is a well-known topic (e.g, [10, 13, 17]), but not so the corresponding results for simulations. The main difficulty is that Hughes and Jacobs' notion of simulation is defined by means of an arbitrary functorial order which bestows them with a high degree of freedom. We have dealt with this by restricting the class of functorial orders (although even so we are not able of obtaining a general result) and by restricting also the class of allowed functors.

In order to get more general results on the subject, an interesting path that we intend to explore is the search for a canonical notion of simulation. This definition would provide us, not only with a "natural" way to understand simulations but, hopefully, would also give rise to "natural" general results about reflection of properties.

Another promising direction of research is the study of reflection and preservation of properties in probabilistic systems, following our results of [4] in combination with the ideas presented in [7, 5, 2].

Acknowledgement

The authors would like to thank the anonymous referees for their comments and suggestions.

References

1. P. Aczel and N. P. Mendler. A final coalgebra theorem. In D. H. Pitt, D. E. Rydeheard, P. Dybjer, A. M. Pitts, and A. Poigné, editors, *Category Theory and Computer Science*, volume 389, pages 357–365, 1989.

2. F. Bartels, A. Sokolova, and E.P. de Vink. A hierarchy of probabilistic system types. *Theor. Comput. Sci.*, 327(1-2):3–22, 2004.
3. E. M. Clarke, O. Grumberg, and D. A. Peled. *Model Checking*. MIT Press, 1999.
4. D. de Frutos Escrig, M. Palomino, and I. Fábregas. Multiset bisimulation as a common framework for ordinary and probabilistic bisimulations. Submitted, 2007.
5. E.P. de Vink and J.J.M.M. Rutten. Bisimulation for probabilistic transition systems: a coalgebraic approach. In P. Degano et al, editors, *ICALP'97*, volume 1256 of *Lect. Notes Comput. Sci.*, pages 4460–470. Springer, Berlin, 1997.
6. I. Fábregas, M. Palomino, and D. de Frutos Escrig. Reflection and preservation of properties in coalgebraic (bi)simulations (Extended). 2007 `http://maude.sip.ucm.es/~miguelpt/`.
7. I. Hasuo. Generic forward and backward simulations. In *International Conference on Concurrency Theory (CONCUR 2006)*, volume 4137 of *Lect. Notes Comput. Sci.*, pages 406–420. Springer, 2006.
8. J. Hughes and B. Jacobs. Simulations in coalgebra. *Theor. Comput. Sci.*, 327(1-2):71–108, 2004.
9. B. Jacobs. *Introduction to Coalgebra. Towards Mathematics of States and Observations*. Book in preparation. Draft available in the web. `http://www.cs.ru.nl/B.Jacobs/CLG/JacobsCoalgebraIntro.pdf`
10. B. Jacobs. *Categorical Logic and Type Theory*, volume 141 of *Studies in Logic and the Foundations of Mathematics*. North-Holland, 1999.
11. B. Jacobs and J.J.M.M. Rutten. A tutorial on (co)algebras and (co)induction. *Bulletin of the European Association for Theoretical Computer Science*, 62:222–259, 1997.
12. Y. Kesten and A. Pnueli. Control and data abstraction: The cornerstones of practical formal verification. *International Journal on Software Tools for Technology Transfer*, 4(2):328–342, 2000.
13. A. Kurz. *Logics for coalgebras and applications to computer science*. PhD thesis, Universität München, 2000.
14. C. Loiseaux, S. Graf, J. Sifakis, A. Bouajjani, and S. Bensalem. Property preserving abstractions for the verification of concurrent systems. *Formal Methods in System Design*, 6:1–36, 1995.
15. R. Milner. *Communication and Concurrency*. Prentice Hall, 1989.
16. M. Palomino. *Reflexión, abstracción y simulación en la lógica de reescritura*. PhD thesis, Universidad Complutense de Madrid, Spain, March 2005.
17. D. Pattinson. *Expressivity Results in the Modal Logic of Coalgebras*. PhD thesis, Universität München, 2001.
18. J.J.M.M. Rutten. Universal coalgebra: a theory of systems. *Theor. Comput. Sci.*, 249(1):3–80, 2000.