

Non-strongly Stable Orders Also Define Interesting Simulation Relations[★]

Ignacio Fábregas, David de Frutos Escrig, and Miguel Palomino

Departamento de Sistemas Informáticos y Computación, UCM
fabregas@fdi.ucm.es, {migue1pt, defrutos}@sip.ucm.es

Abstract. We present a study of the notion of coalgebraic simulation introduced by Hughes and Jacobs. Although in their original paper they allow any functorial order in their definition of coalgebraic simulation, for the simulation relations to have good properties they focus their attention on functors with orders which are strongly stable. This guarantees a so-called “composition-preserving” property from which all the desired good properties follow. We have noticed that the notion of strong stability not only ensures such good properties but also “distinguishes the direction” of the simulation. For example, the classic notion of simulation for labeled transition systems, the relation “ p is simulated by q ”, can be defined as a coalgebraic simulation relation by means of a strongly stable order, whereas the opposite relation, “ p simulates q ”, cannot. Our study was motivated by some interesting classes of simulations that illustrate the application of these results: covariant-contravariant simulations and conformance simulations.

1 Introduction and Presentation of Our New Results

Simulations are a very natural way to compare systems defined by transition systems or other related mechanisms based on the description of systems by means of the actions they can execute at each of their states [11]. They can be enriched in several ways to obtain, in particular, the important ready simulation semantics [2,8], as well as other more elaborated ones such as nested simulations [5]. Quite recently we have studied the general concept of constrained simulation [3], proving that all the simulation relations constrained by an adequate condition have similar properties. The semantics of these constrained simulations is also the basis for our unified presentation of the semantics of processes [4], where all the semantics in the lbtb-spectrum [13] (and other new semantics) are classified in a systematic way.

Hughes and Jacobs [6] have also developed a systematic study of simulation-like relations, this time in a purely coalgebraic context, so that simulations are studied in connection with bisimulations [11], the fundamental concept to define equivalence in the coalgebraic world. Their coalgebraic simulations are defined in terms of an order \sqsubseteq associated to the functor F corresponding to the coalgebra $c : X \rightarrow FX$ that we want to observe. In this way they obtain a very general notion of coalgebraic simulation, not only because all functors F are considered, including in particular the important

[★] Research supported by the Spanish projects DESAFIOS TIN2006-15660-C02-01, WEST TIN2006-15578-C02-01, PROMESAS S-0505/TIC/0407 and UCM-BSCH GR58/08/910606.

class of polynomial functors, but also because by changing the family of orders \sqsubseteq_X many different families of simulation relations can be obtained. The general properties of these simulations can be studied in the defined coalgebraic framework, thus avoiding the need of similar proofs for each of the particular classes of simulations.

Certainly, this generic presentation of the notion of coalgebraic simulation has as advantage that it provides a wide and abstract framework where one can try to isolate and take advantage of the main properties of all the simulation-like relations. However, at the same time it can be argued that the proposal fails to capture in a tight manner the spirit of simulation relations because, in addition to the natural notions of simulations, the framework also allows for other less interesting relations. This has as a result that some natural properties of simulations cannot be proved in general, simply due to the fact that they are not satisfied by all of the permitted coalgebraic simulation relations. For instance, the induced similarity relation between systems is not always an order because transitivity is not always satisfied. In order to guarantee transitivity, and other related properties of coalgebraic simulations, Jacobs and Hughes introduce in [7] the composition-preserving property to the order \sqsubseteq that induces the simulation relation. In [6] they continue with the study of the topic and present *stability* of orders as a natural categorical property to guarantee that an order is composition-preserving. They also comment that stability is not easy to check and introduce a stronger condition (that we will call right-stability) so that, whenever applicable, the checking of the main properties of coalgebraic simulations becomes much simpler than in the general case.

Roughly speaking, given an order \sqsubseteq_X on FX for each set X , the induced coalgebraic simulations are defined in the same way as bisimulations for F , but allowing a double application of \sqsubseteq on the two sides of the defined relation. More precisely, instead of the functor $\text{Rel}(F)$ defining plain bisimulations, $\text{Rel}_{\sqsubseteq}(F)$ defined as $\sqsubseteq_Y \circ \text{Rel}(F) \circ \sqsubseteq_X$ is used. There are several interesting facts hidden behind the apparent simplicity of this definition. The first one is that, in general, it only defines an order and not an equivalence relation, even if it is based on bisimulations (that always define an equivalence relation, namely, bisimilarity). The reason is that the order \sqsubseteq appears “in the same direction” on both sides of the definition, thus breaking its symmetry. However, we can also define some equivalence relations weaker than bisimilarity by using an equivalence relation \equiv as the order \sqsubseteq . Another interesting fact is that whenever we define a coalgebraic simulation by using \sqsubseteq , the inverse order \supseteq defines the inverse relation of that defined by \sqsubseteq once we also interchange the roles of the related sets X and Y (so we could say that we are defining in fact the same relation but looking at it from the other side). Stability is also a symmetric condition, so that whenever an order \sqsubseteq on a functor F is stable, the inverse order \supseteq is stable for F , too. This is quite reasonable, since stability is imposed in order to guarantee transitivity of the generated similarity relation and the inverse of a transitive relation is also transitive, so that whenever \sqsubseteq generates an “admissible” similarity relation (meaning that it is an order), the inverse order \supseteq must be also admissible.

It is worth noting that the stronger condition guaranteeing stability is asymmetric. In fact, Hughes and Jacobs prove in [6] that “right-stability” implies that

$$\text{Rel}(F)(R) \circ \sqsubseteq_X \subseteq \sqsubseteq_Y \circ \text{Rel}(F)(R), \quad (1)$$

which in fact motivates our name for the condition.

A second surprise was to notice that, in most cases, right-stability induces a “natural direction” on the orders defining the coalgebraic simulation. For instance, for plain similarity over labeled transition systems, the inclusion order \subseteq induces the classic simulation relation while the reversed inclusion \supseteq induces the opposite “simulated by” relation: the first one is right-stable while the second is not.

All these general results arose when trying to integrate two new simulation-like notions as coalgebraic simulations definable by a stable order, so that we could obtain for free all the good properties that have been proved in [6] for this class of relations.

The first new simulation notion is that of covariant-contravariant simulations, where the alphabet of actions Act is partitioned into three disjoint sets Act^l , Act^r , and Act^{bi} . The intention is for the simulation to treat the actions in Act^l like in the ordinary case, to interchange the role of the related processes for those actions in Act^r , and to impose a symmetric condition like that defining bisimulation for the actions in Act^{bi} .

The second notion, conformance simulations, captures the conformance relations [9,12] that several authors introduced in order to formalize the notion of possible implementations. Like covariant-contravariant simulations, they can be defined as coalgebraic simulations for some stable order which is not right-stable neither left-stable. We show that the good properties of these two classes of orders are preserved in those orders that can be seen as a kind of composition of right-stable and left-stable orders. We use this fact to derive the stability of the orders defining both covariant-contravariant and conformance simulations.

2 Coalgebraic Simulations and Stability

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 $c : X \rightarrow FX$. We often call X the state space and c the transition or coalgebra structure.

An arbitrary endofunctor $F : \mathbf{Sets} \rightarrow \mathbf{Sets}$ can be lifted to a functor in the category \mathbf{Rel} over $\mathbf{Sets} \times \mathbf{Sets}$ of relations, $\mathbf{Rel}(F) : \mathbf{Rel} \rightarrow \mathbf{Rel}$. In set-theoretic terms, for a relation $R \subseteq X_1 \times X_2$,

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

A *bisimulation* for coalgebras $c : X \rightarrow FX$ and $d : Y \rightarrow FY$ is a relation $R \subseteq X \times Y$ which is “closed under c and d ”:

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

where the r_i are the projections of R into X and Y . Sometimes we shall use the term F -bisimulation to emphasize the functor we are working with.

Bisimulations can also be characterized by means of spans, using the general categorical definition by Aczel and Mendler [1]:

$$\begin{array}{ccccc}
 X & \xleftarrow{r_1} & R & \xrightarrow{r_2} & Y \\
 c \downarrow & & \downarrow e & & \downarrow d \\
 FX & \xleftarrow{Fr_1} & FR & \xrightarrow{Fr_2} & FY
 \end{array}$$

R is a bisimulation iff it is the carrier of some coalgebra e making the above diagram commute. Alternatively, bisimulations can also be defined as the $\text{Rel}(F)$ -coalgebras in the category **Rel**.

We will also need the general concept of simulation introduced by Hughes and Jacobs [6] using orders on functors. Let $F : \mathbf{Sets} \rightarrow \mathbf{Sets}$ be a functor. An order on F is defined by means of a functorial collection of preorders $\sqsubseteq_X \subseteq FX \times FX$ that must be preserved by renaming: for every $f : X \rightarrow Y$, if $u \sqsubseteq_X u'$ then $Ff(u) \sqsubseteq_Y Ff(u')$.

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

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

where the lax relation lifting $\text{Rel}_{\sqsubseteq}(F)(R)$ is $\sqsubseteq_Y \circ \text{Rel}(F)(R) \circ \sqsubseteq_X$, which can be expanded to

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

Alternatively, \sqsubseteq -simulations are just the $\text{Rel}_{\sqsubseteq}(F)$ -coalgebras in **Rel**.

Sometimes, when $f : X \rightarrow Y$ and $A \subseteq X$ we will simply write $f(A)$ for the image $\coprod_f(A)$.

A functor with order \sqsubseteq is *stable* [6] if the relation lifting $\text{Rel}_{\sqsubseteq}(F)$ commutes with substitution, that is, if for every $f : X \rightarrow Z$ and $g : Y \rightarrow W$, $\text{Rel}_{\sqsubseteq}(F)((f \times g)^{-1}(R)) = (Ff \times Fg)^{-1}(\text{Rel}_{\sqsubseteq}(F)(R))$.¹ They also define a stronger condition that we are going to call right-stability.

Definition 1 ([6]). We will say that a functor F with order \sqsubseteq is *right-stable* if, for every function $f : X \rightarrow Y$, we have²

$$(id \times Ff)^{-1} \sqsubseteq_Y \subseteq \coprod_{Ff \times id} \sqsubseteq_X. \tag{2}$$

According to [6], condition (2) is equivalent to (a) F being stable and (b) for every relation $R \subseteq X \times Y$,

$$\text{Rel}(F)(R) \circ \sqsubseteq_X \subseteq \sqsubseteq_Y \circ \text{Rel}(F)(R). \tag{3}$$

Right-stability was introduced by arguing that it is easier to check than plain stability, while being satisfied by nearly all orders discussed in that paper. Surprisingly, one cannot find in [6] a clear explanation of the reason why right-stable orders are easier to analyze. In our opinion, the crucial fact is that from (3) we can immediately conclude that

$$\sqsubseteq_Y \circ \text{Rel}(F)(R) \circ \sqsubseteq_X = \sqsubseteq_Y \circ \text{Rel}(F)(R), \tag{4}$$

so that the coalgebraic simulations for a right-stable order \sqsubseteq can be equivalently defined by means of the asymmetric definition on the right-hand side of equality (4). If the order \sqsubseteq can be used only on one of the sides of the definition, the verification of the

¹ In fact, the inclusion \subseteq always holds.

² Again, the other inclusion is always true since \sqsubseteq functorial means that $Ff(u) \sqsubseteq_Y Ff(v)$ if $u \sqsubseteq_X v$.

properties of the induced coalgebraic simulations becomes much easier than when using the original definition.

It was quite surprising to discover that the easiest way to prove the properties of the “simulated by” relations which come from symmetric properties such as composition-preserving (that are also satisfied by the corresponding inverse relations “simulates”) is to break that symmetry by considering the asymmetric definition of coalgebraic simulations that only use \sqsubseteq_Y ; certainly, this is only possible when the defining order \sqsubseteq is right-stable.

Stability is used in [6, Lemma 5.3] to prove that lax relation lifting preserves composition of relations, which is needed to prove [6, Lemma 5.4(2)], the crucial fact that the induced similarity relation is transitive; this need not be the case for the simulation notion defined by an arbitrary order \sqsubseteq .

3 On Stability of Simulation and Anti-simulation

Plain simulations between labeled transition systems can be defined as coalgebraic simulations considering the functor $F = \mathcal{P}^A$ (G^A denote the functor $X \mapsto (G(X))^A$) with the order \sqsubseteq given by $\alpha \sqsubseteq \beta$ for $\alpha, \beta : A \rightarrow \mathcal{P}X$ iff for all $a \in A$, $\alpha(a) \sqsubseteq \beta(a)$.

Lemma 1. *The order \sqsubseteq defining plain simulations for labeled transition systems is right-stable.*

Corollary 1. *Plain simulations between labeled transition systems can be defined as the $(\sqsubseteq_Y \circ \text{Rel}(F))$ -coalgebras.*

It is worth examining the consequences of the removal of \sqsubseteq_X from the original definition of coalgebraic simulations in this particular case. Both \sqsubseteq_X and \sqsubseteq_Y correspond to the inclusion order, but when applied at the right-hand side it means that we can reduce the set of successors of the simulating process q when simulating the execution of a by p . This means that starting from a set $Y' \subseteq Y$ we can obtain an adequate subset $Y'' \subseteq Y'$. Instead, the application of \sqsubseteq_X at the left-hand side allows to enlarge the set of successors of the simulated process p and this produces a set X'' larger than the given X' : one could say that we need to consider “new” information not in X' , while going from Y' to Y'' just “removes” some known information.

Another interesting point arises from the fact that every use of \sqsubseteq_X at the left-hand side can be “compensated” by removing at Y the added states and this is why Corollary 1 was correct, because we can always avoid the introduction of new successors in the simulated process by simply removing them at the right-hand side. However, the opposite procedure, to compensate the removal of states by adding them at the simulated process side is not always possible, since in general X could be not big enough.

The anti-simulations can be defined as coalgebraic simulations by taking the reversed inclusion order instead of \subseteq . It is interesting to note that it is not right-stable as the following counterexample shows. Let $X = \{x\}$ and $Y = \{y_1, y_2\}$ and let $f : X \rightarrow Y$ be such that $f(x) = y_1$. With these definitions the pair $(Y, X) \in (id \times \mathcal{P}f)^{-1}(\supseteq)$, since $Y \supseteq \{y_1\} = \mathcal{P}f(X)$, but it is obvious that there is no $A \subseteq X$ such that $Y = f(A)$ because f is not surjective.

However, the order defining anti-simulations is stable as a consequence of the following general result.

Lemma 2. *F with an order \sqsubseteq is stable iff it is stable with the order \sqsubseteq^{op} .*

Proof. It is shown in [6, Lemma 4.2(4)] that $\text{Rel}_{\sqsubseteq^{op}}(F)(R) = (\text{Rel}_{\sqsubseteq}(F)(R^{op}))^{op}$. Then, on the one hand,

$$\begin{aligned} (Ff \times Fg)^{-1}(\text{Rel}_{\sqsubseteq^{op}}(F)(R)) &= (Ff \times Fg)^{-1}(\text{Rel}_{\sqsubseteq}(F)(R^{op}))^{op} \\ &= ((Fg \times Ff)^{-1}\text{Rel}_{\sqsubseteq}(F)(R^{op}))^{op}, \end{aligned}$$

and on the other hand,

$$\begin{aligned} \text{Rel}_{\sqsubseteq^{op}}(F)((f \times g)^{-1}(R)) &= (\text{Rel}_{\sqsubseteq}(F)((f \times g)^{-1}(R))^{op})^{op} \\ &= (\text{Rel}_{\sqsubseteq}(F)((g \times f)^{-1}(R^{op}))^{op}). \end{aligned}$$

Since $R^{op} \subseteq Y \times X$ is a relation whenever $R \subseteq X \times Y$ is so, and f, g , and R are arbitrary, we have shown that

$$\text{Rel}_{\sqsubseteq}(F)((f \times g)^{-1}(R)) = (Ff \times Fg)^{-1}(\text{Rel}_{\sqsubseteq}(F)(R))$$

if and only if

$$\text{Rel}_{\sqsubseteq^{op}}(F)((f \times g)^{-1}(R)) = (Ff \times Fg)^{-1}(\text{Rel}_{\sqsubseteq^{op}}(F)(R)),$$

and therefore F is stable for \sqsubseteq iff it is stable for \sqsubseteq^{op} . □

Corollary 2. *The order \sqsubseteq^{op} defining anti-simulations for transition systems as coalgebraic simulations is stable.*

One could conclude from the observation above that there is indeed a natural argument supporting plain similarity as a “right” coalgebraic similarity, definable by a right-stable order. This criterion could be adopted to define right coalgebraic simulations, which plain similarity would satisfy while the opposite relation “is simulated by” would not. However, we immediately noticed that we could define “left-stable” orders by interchanging the roles of Ff and id in the definition of right-stable order, obtaining the inverse inclusion in (1).

Definition 2. *We will say that a functor F with order \sqsubseteq is **left-stable** if, for every function $f : X \rightarrow Y$, we have*

$$(Ff \times id)^{-1} \sqsubseteq_Y \subseteq \coprod_{id \times Ff} \sqsubseteq_X. \tag{5}$$

It is immediate to check that an order \sqsubseteq is left-stable iff the inverse order \sqsubseteq^{op} is right-stable. Moreover, left-stable orders have the same structural properties that right-stable ones so that, in particular, they are also stable and hence composition-preserving. But in this case it would be the inverse simulations, corresponding to the “is simulated by” notion, that would be natural instead of plain simulations. As a conclusion, we could use right or left-stability as a criterion to choose a natural direction for the simulation order. But the important fact in both cases is that the simplified asymmetric definitions (using either \sqsubseteq_X or \sqsubseteq_Y) of coalgebraic simulations are much easier to handle than the symmetric original definition (where both \sqsubseteq_X and \sqsubseteq_Y have to be used).

4 Covariant-Contravariant Simulations and Conformance Simulations

Covariant-contravariant simulations are defined by combining the conditions “to simulate” and “be simulated by”, using a partition of the alphabet of actions of the compared labeled transition systems.

Definition 3. Given $c : X \rightarrow \mathcal{P}(X)^{Act}$ and $d : Y \rightarrow \mathcal{P}(Y)^{Act}$ labeled transition systems for the alphabet Act , and $\{Act^r, Act^l, Act^{bi}\}$ a partition of this alphabet, a (Act^r, Act^l) -**simulation** between c and d is a relation $S \subseteq X \times Y$ such that for every $(x, y) \in S$ we have:

- for all $a \in Act^r \cup Act^{bi}$ and all $x \xrightarrow{a} x'$ there exists $y \xrightarrow{a} y'$ with $(x', y') \in S$.
- for all $a \in Act^l \cup Act^{bi}$, and all $y \xrightarrow{a} y'$ there exists $x \xrightarrow{a} x'$ with $(x', y') \in S$.

We write $x \mathcal{S}_{Act^l} \mathcal{S}_{Act^r} y$, and say that x is (Act^r, Act^l) -simulated by y , if and only if there exists some (Act^r, Act^l) -simulation S with xSy .

A very interesting application of this kind of simulations is related with the definition of adequate simulation notions for input/output (I/O) automata [10]. The classic approach to simulations is based on the definition of semantics for reactive systems, where all the actions of the processes correspond to input actions that the user must trigger. Instead, whenever we have explicit output actions the situation is the opposite: it is the system that produces the actions and the user who is forced to accept the produced output. Then, it is natural to conclude that in the simulation framework we have to dualize the simulation condition when considering output actions, and this is exactly what our anti-simulation relations do.

Covariant-contravariant simulations can be easily obtained as coalgebraic simulations, as the following proposition proves.

Proposition 1. (Act^r, Act^l) -simulations can be defined as the coalgebraic simulations for the functor $F = \mathcal{P}^{Act}$ with functorial order $_{Act^r} \sqsubseteq_{Act^l}$ where, for each set X and $\alpha, \alpha' : Act \rightarrow \mathcal{P}(X)$, we have $\alpha \mathcal{S}_{Act^r} \sqsubseteq_{Act^l} \alpha'$ if:

- for all $a \in Act^r \cup Act^{bi}$, $\alpha(a) \subseteq \alpha'(a)$, and
- for all $a \in Act^l \cup Act^{bi}$, $\alpha(a) \supseteq \alpha'(a)$.

Note that in particular we have $\alpha(a) = \alpha'(a)$ for all $a \in Act^{bi}$.

Proof. Intuitively, using the order $_{Act^r} \sqsubseteq_{Act^l}$ on the left-hand side of $\text{Rel}_{\sqsubseteq}(F)(R)$ allows us to remove a' -transitions when $a' \in Act^l$, whereas using it on the right-hand side of $\text{Rel}_{\sqsubseteq}(F)(R)$ allows us to remove a -transitions when $a \in Act^r$.

Let us suppose that we have a classic covariant-contravariant simulation $_{Act^r} \mathcal{S}_{Act^l}$ between labeled transition systems $c : P \rightarrow \mathcal{P}(P)^{Act}$ and $d : Q \rightarrow \mathcal{P}(Q)^{Act}$ defined by $c(p)(a) = \{p' \mid p \xrightarrow{a} p'\}$ and $d(q)(a) = \{q' \mid q \xrightarrow{a} q'\}$. We must show that if $p \mathcal{S}_{Act^r} \mathcal{S}_{Act^l} q$ then there exist p^* and q^* such that

$$c(p) \mathcal{S}_{Act^r} \sqsubseteq_{Act^l} p^* \text{Rel}(\mathcal{P}^{Act})_{(Act^r \mathcal{S}_{Act^l})} q^* \mathcal{S}_{Act^r} \sqsubseteq_{Act^l} d(q). \quad (6)$$

We define p^* and q^* as follows:

- p^* has the same transitions as $c(p)$, except for those transitions $p \xrightarrow{a'} p'$ with $a' \in Act^l$ such that there is no q' with $q \xrightarrow{a'} q'$ and $p' \mathcal{S}_{Act^l} q'$.
- q^* has the same transitions as $d(q)$, except for those transitions $q \xrightarrow{a} q'$ with $a \in Act^r$ such that there is no p' with $p \xrightarrow{a} p'$ and $p' \mathcal{S}_{Act^r} q'$.

It is immediate from these definitions that $c(p) \mathcal{A}_{Act^r} \sqsubseteq_{Act^l} p^*$ and $q^* \mathcal{A}_{Act^r} \sqsubseteq_{Act^l} d(q)$, so we are left with checking that $p^* \text{Rel}(\mathcal{P}^{Act}) q^*$.

Let $p' \in p^*(a)$ with $a \in Act^r$. By construction of p^* , since we have not dropped any a -transitions from p^* , $p \xrightarrow{a} p'$. Using the fact that \mathcal{S}_{Act^l} is a classic covariant-contravariant simulation, there exists q' such that $q \xrightarrow{a} q'$ with $p' \mathcal{S}_{Act^l} q'$, and, again by construction, $q' \in q^*(a)$ because there is some $p \xrightarrow{a} p'$ with $p' \mathcal{S}_{Act^l} q'$. Similarly, if $p' \in p^*(a)$ with $a' \in Act^l$, by construction of p^* there must exist some q' such that $q \xrightarrow{a'} q'$ with $p' \mathcal{S}_{Act^l} q'$. Again, since we have not removed any a' -transitions from $d(q)$ in q^* , it must be true that $q' \in q^*(a)$. Finally, if $p' \in p^*(a)$ with $a \in Act^{bi}$ we have that $p \xrightarrow{a} p'$ and hence there exists q' such that $q \xrightarrow{a} q'$ with $p' \mathcal{S}_{Act^l} q'$, but also $q' \in q^*(a)$.

The argument that shows that for every $q' \in q^*(a)$ there exists some $p' \in p^*(a)$ with $p' \mathcal{S}_{Act^l} q'$ is analogous.

We show now the other implication, that a coalgebraic covariant-contravariant simulation is a classic one. In this case we start from coalgebras c and d that satisfy relation (6) whenever $p \mathcal{S}_{Act^l} q$.

If $p \xrightarrow{a} p'$ for $a \in Act^r$, then $p' \in p^*(a)$ because $c(p) \mathcal{A}_{Act^r} \sqsubseteq_{Act^l} p^*$ and, since $p^* \text{Rel}(\mathcal{P}^{Act})_{(Act^r \mathcal{S}_{Act^l})} q^*$, there is some $q' \in q^*(a)$ with $p' \mathcal{S}_{Act^l} q'$. Again, the definition of $\mathcal{A}_{Act^r} \sqsubseteq_{Act^l}$ ensures that $q^*(a) \subseteq d(q)(a)$ and hence $q \xrightarrow{a} q'$ as required. Similarly, if $q \xrightarrow{a'} q'$ for $a' \in Act^l$, then $q' \in q^*(a)$ because $q^* \mathcal{A}_{Act^r} \sqsubseteq_{Act^l} d(q)$ and thus, as in the previous case, there exists $p' \in p^*(a)$ with $p' \mathcal{S}_{Act^l} q'$ and $p \xrightarrow{a'} p'$. Finally if $p \xrightarrow{a} p'$ for $a \in Act^{bi}$ (resp. $q \xrightarrow{a} q'$), again by the definition of $\mathcal{A}_{Act^r} \sqsubseteq_{Act^l}$ we have $p' \in p^*(a)$ (resp. $q' \in q^*(a)$) and, from $p^* \text{Rel}(\mathcal{P}^{Act})_{(Act^r \mathcal{S}_{Act^l})} q^*$, it follows that there exists $q' \in q^*(a)$ (resp. $p' \in p^*(a)$) such that $p' \mathcal{S}_{Act^l} q'$; by the definition of $\mathcal{A}_{Act^r} \sqsubseteq_{Act^l}$, $q \xrightarrow{a} q'$ (resp. $p \xrightarrow{a} p'$). \square

The other new kind of simulations in which we are interested is that of conformance simulations, where the conformance relation in [9,12] meets the simulation world in a nice way. In the definition below we will write $p \xrightarrow{a}$ if $p \xrightarrow{a} p'$ for some p' .

Definition 4. Given $c : X \rightarrow \mathcal{P}(X)^A$ and $d : Y \rightarrow \mathcal{P}(Y)^A$ two labeled transition systems for the alphabet A , a **conformance simulation** between them is a relation $R \subseteq X \times Y$ such that whenever pRq , then:

- For all $a \in A$, if $p \xrightarrow{a}$ we must also have $q \xrightarrow{a}$ (this means, using the usual notation for process algebras, that $I(p) \subseteq I(q)$).
- For all $a \in A$ such that $q \xrightarrow{a} q'$ and $p \xrightarrow{a}$, there exists some p' with $p \xrightarrow{a} p'$ and $p'Rq'$.

Conformance simulations allow the extension of the set of actions offered by a process, so that in particular we will have $a < a + b$, but they also consider that a process can be “improved” by reducing the nondeterminism in it, so that $ap + aq < ap$. In this way we have again a kind of covariant-contravariant simulation, not driven by the alphabet of actions executed by the processes but by their nondeterminism.

Once again, conformance simulations can be defined as coalgebraic simulations taking the adequate order on the functor defining labeled transition systems.

Proposition 2. *Conformance simulations can be obtained as the coalgebraic simulations for the order \sqsubseteq^{Conf} on the functor \mathcal{P}^A , where for any set X we have $u \sqsubseteq_X^{Conf} v$ if for every $u, v : A \rightarrow \mathcal{P}X$ and $a \in A$:*

- either $u(a) = \emptyset$, or
- $u(a) \supseteq v(a)$ and $v(a) \neq \emptyset$.

Proof. Let us first prove that \sqsubseteq_X^{Conf} is indeed an order. It is clear that the only not immediate property is transitivity. To check it, let us take $u \sqsubseteq_X^{Conf} v \sqsubseteq_Y^{Conf} w$: if $u(a) = \emptyset$ we are done; otherwise, we have $u(a) \supseteq v(a)$ and $v(a) \neq \emptyset$, so that we also have $v(a) \supseteq w(a)$ and $w(a) \neq \emptyset$, obtaining $u(a) \supseteq w(a)$ and $w(a) \neq \emptyset$.

Now, we can interpret that using the order \sqsubseteq^{Conf} on the left-hand side of $\text{Rel}_{\sqsubseteq}(F)(R)$ allows us to remove all a -transitions except for the last one, whereas using it on the right-hand side allows us to remove all b -transitions for $b \in B$, where B is any set of actions. But again, as in the proof of Proposition 1, we can compensate these additions with the corresponding removals at the other side and the proof follows in an analogous way. □

Next we check that the order $_{Act^r} \sqsubseteq_{Act^l}$ defining covariant-contravariant simulations is stable.

Lemma 3. *Given a partition $\{Act^r, Act^l, Act^{bi}\}$ of Act the order $_{Act^r} \sqsubseteq_{Act^l}$ for the functor \mathcal{P}^{Act} defining covariant-contravariant simulations for transition systems is stable.*

Proof. It is clear that the order $_{Act^r} \sqsubseteq_{Act^l}$ can be obtained as the product of a family of orders \sqsubseteq^a for the functor \mathcal{P} , with $a \in Act$. This is indeed the case taking $\sqsubseteq_X^a = \subseteq_X$ for $a \in Act^r$, $\sqsubseteq_X^a = \supseteq_X$ for $a \in Act^l$ and $\sqsubseteq_X^a = =_X$ for $a \in Act^{bi}$. Then it is easy to see that to obtain that $_{Act^r} \sqsubseteq_{Act^l}$ is stable it is enough to prove that each of the orders \sqsubseteq^a is stable.

This latter requirement is straightforward because, for $a \in Act^r$, \sqsubseteq^a is right-stable; for $a \in Act^l$ the order \sqsubseteq^a is left-stable; and for $a \in Act^{bi}$, \sqsubseteq^a is the equality relation, which is both right and left-stable, for every functor F . □

Certainly, the order defining covariant-contravariant simulations is not right-stable nor left-stable, but in the proof above we have used the power of these two properties thanks to the fact that the order $_{Act^r} \sqsubseteq_{Act^l}$ can be factorised as the product of a family of orders that are either right-stable or left-stable. Then we can obtain the following sequence of general definitions and results, from which Lemma 3 could be obtained as a simple particular case.³

³ Instead of removing the above, we have preferred to maintain the sequence of results in the order in which we got them, starting with our motivating example.

Definition 5. We say that an order \sqsubseteq on a functor F^A is **action-distributive** if there is a family of orders \sqsubseteq^a on F such that

$$f \sqsubseteq g \iff f(a) \sqsubseteq^a g(a) \text{ for all } a \in A.$$

Whenever \sqsubseteq can be distributed in this way we will write $\sqsubseteq = \prod_{a \in A} \sqsubseteq^a$.

Definition 6. We say that an action-distributive order \sqsubseteq on F^A is **side stable** if for the decomposition $\sqsubseteq = \prod_{a \in A} \sqsubseteq^a$ we have that each order \sqsubseteq^a is either right-stable or left-stable.

By separating the right-stable and the left-stable components we obtain $\sqsubseteq = \sqsubseteq^l \times \sqsubseteq^r$, where A^r (resp. A^l) collects the set of arguments⁴ $a \in A$ with \sqsubseteq^a right-stable (resp. left-stable). We extend \sqsubseteq^l and \sqsubseteq^r to obtain a pair of orders on F^A , \sqsubseteq^l and \sqsubseteq^r , defined by:

- $f \sqsubseteq^r g$ iff $f(a) \sqsubseteq^a g(a)$ for all $a \in A^r$ and $f(a) = g(a)$ for all $a \in A^l$.
- $f \sqsubseteq^l g$ iff $f(a) \sqsubseteq^a g(a)$ for all $a \in A^l$ and $f(a) = g(a)$ for all $a \in A^r$.

Proposition 3. The order \sqsubseteq^l is left-stable, while \sqsubseteq^r is right-stable. We have $\sqsubseteq = (\sqsubseteq^l \circ \sqsubseteq^r) = (\sqsubseteq^r \circ \sqsubseteq^l)$, and therefore we also have $\sqsubseteq = (\sqsubseteq^l \cup \sqsubseteq^r)^*$.

Proposition 4. For any side stable order \sqsubseteq on F^A , if we have a decomposition $\sqsubseteq = \sqsubseteq^l \times \sqsubseteq^r$ based on a partition of A into a set of right-stable components A^r and another set of left-stable components A^l , then we can obtain the coalgebraic simulations for \sqsubseteq as the $(\sqsubseteq_Y^r \circ \text{Rel}(F) \circ \sqsubseteq_X^l)$ -coalgebras.

Proof. By definition, $\text{Rel}_\sqsubseteq(F)(R) = \sqsubseteq_Y \circ \text{Rel}(F)(R) \circ \sqsubseteq_X$. Since $\sqsubseteq = (\sqsubseteq^r \circ \sqsubseteq^l) = (\sqsubseteq^l \circ \sqsubseteq^r)$, we have:

$$\begin{aligned} \sqsubseteq_Y \circ \text{Rel}(F)(R) \circ \sqsubseteq_X &= (\sqsubseteq_Y^l \circ \sqsubseteq_Y^r) \circ \text{Rel}(F)(R) \circ (\sqsubseteq_X^r \circ \sqsubseteq_X^l) \\ &= \sqsubseteq_Y^l \circ (\sqsubseteq_Y^r \circ \text{Rel}(F)(R) \circ \sqsubseteq_X^r) \circ \sqsubseteq_X^l \\ &= (\sqsubseteq_Y^l \circ \sqsubseteq_Y^r) \circ \text{Rel}(F)(R) \circ \sqsubseteq_X^l && \text{(by right-stability of } \sqsubseteq^r) \\ &= \sqsubseteq_Y^r \circ (\sqsubseteq_Y^l \circ \text{Rel}(F)(R) \circ \sqsubseteq_X^l) && \text{(since } \sqsubseteq^r \text{ and } \sqsubseteq^l \text{ commute)} \\ &= \sqsubseteq_Y^r \circ \text{Rel}(F)(R) \circ \sqsubseteq_X^l && \text{(by left-stability of } \sqsubseteq^l) \end{aligned}$$

□

The characterization above still requires the use of the order on both sides of the $\text{Rel}(F)(R)$ operator. However, the fact that \sqsubseteq_Y^r (resp. \sqsubseteq_X^l) is right-stable (resp. left-stable) makes the application of this decomposition as simple as when coping with either a right or left-stable order.

Proposition 5. If $\sqsubseteq = \prod_{a \in A} \sqsubseteq^a$ and \sqsubseteq^a is stable for all $a \in A$, then \sqsubseteq is stable.

⁴ We have assumed here a partition $\{A^l, A^r\}$ of the set A into two sets of right-stable and left-stable components. Obviously, if there were some arguments $a \in A$ on which \sqsubseteq^a is both right-stable and left-stable then the decomposition would not be unique, but the result would be valid for any such decomposition.

Proof. The result follows from the following chain of implications:

$$\begin{aligned}
& (u, v) \in (Ff \times Fg)^{-1} \text{Rel}_{\sqsubseteq}(F)(R) \\
\iff & Ff(u) \sqsubseteq z' \text{Rel}(F)(R)w' \sqsubseteq Fg(v) \\
\iff & Ff(u)(a) \sqsubseteq^a z'(a) \text{Rel}(F^a)(R)w'(a) \sqsubseteq^a Fg(v)(a), \text{ for all } a \\
\iff & (u(a), v(a)) \in (Ff \times Fg)^{-1} \text{Rel}_{\sqsubseteq^a}(F)(R), \text{ for all } a \\
\implies & (u(a), v(a)) \in \text{Rel}_{\sqsubseteq^a}(F)((f \times g)^{-1}R), \text{ for all } a \\
\iff & u(a) \sqsubseteq^a x'(a) \text{Rel}(F)((f \times g)^{-1}R)y'(a) \sqsubseteq^a v(a), \text{ for all } a \\
\iff & (u, v) \in \text{Rel}_{\sqsubseteq}(F)((f \times g)^{-1}R) \quad \square
\end{aligned}$$

Corollary 3. *Any side stable order is stable.*

Corollary 4. *The order $\text{Act}^r \sqsubseteq_{\text{Act}^l}$ defining covariant-contravariant simulations is side stable and therefore it is stable too.*

Next we consider the case of conformance simulations, for which we can obtain similar results to those proved for covariant-contravariant simulations.

Lemma 4. *The order $\sqsubseteq^{\text{Conf}}$ defining conformance simulations for transition systems is stable.*

Proof. Let $R \subseteq Z \times W$ be a relation and $f : X \rightarrow Z, g : Y \rightarrow W$ arbitrary functions. If $(u, v) \in (\mathcal{P}^A f \times \mathcal{P}^A g)^{-1}(\text{Rel}_{\sqsubseteq^{\text{Conf}}}(\mathcal{P}^A)(R))$, then there exist z and w such that

$$\mathcal{P}^A f(u) \sqsubseteq^{\text{Conf}} z \text{Rel}(\mathcal{P}^A)(R) w \sqsubseteq^{\text{Conf}} \mathcal{P}^A g(v). \quad (7)$$

We have to show that $(u, v) \in \text{Rel}_{\sqsubseteq^{\text{Conf}}}(\mathcal{P}^A)((f \times g)^{-1}(R))$, that is, there exist x and y such that

$$u \sqsubseteq^{\text{Conf}} x \text{Rel}(\mathcal{P}^A)((f \times g)^{-1}(R)) y \sqsubseteq^{\text{Conf}} v.$$

Let us define $x : A \rightarrow \mathcal{P}(X)$ by $x(a) = u(a) \cap f^{-1}(z(a))$ and $y : A \rightarrow \mathcal{P}(Y)$ by $y(a) = g^{-1}(w(a))$. Then we have:

1. $u \sqsubseteq^{\text{Conf}} x$.

If $w(a) = \emptyset$, there is nothing to prove. Otherwise, since $\mathcal{P}^A f(u) \sqsubseteq^{\text{Conf}} z$ and $f(u(a)) \neq \emptyset$, we have $f(u(a)) \supseteq z(a) \neq \emptyset$ and hence $u(a) \supseteq u(a) \cap f^{-1}(z(a)) = x(a) \neq \emptyset$.

2. $y \sqsubseteq^{\text{Conf}} v$.

If $w(a) = \emptyset$, then $y(a) = g^{-1}(w(a)) = \emptyset$. Otherwise, since $w \sqsubseteq^{\text{Conf}} \mathcal{P}^A g(v)$, we have $w(a) \supseteq g(v(a)) \neq \emptyset$, so that $v(a) \neq \emptyset$ and $y(a) = g^{-1}(w(a)) \supseteq g^{-1}(g(v(a))) \supseteq v(a)$.

3. $x \text{Rel}(\mathcal{P}^A)((f \times g)^{-1}(R)) y$.

For every $a \in A$ we need to show that $x(a) \text{Rel}(\mathcal{P})((f \times g)^{-1}(R)) y(a)$, which means:

(a) for every $p \in x(a)$ there exists $q \in y(a)$ such that $p (f \times g)^{-1}(R) q$, that is, $f(p)Rg(q)$; and

(b) for every $q \in y(a)$ there exists $p \in x(a)$ such that $p (f \times g)^{-1}(R) q$, that is, $f(p)Rg(q)$.

In the first case, let $p \in x(a)$; by definition of x , $f(p) \in z(a)$. Now, from $z \text{Rel}(\mathcal{P}^A)(R) w$ we obtain that for each $p' \in z(a)$ there exists $q' \in w(a)$ such that $p'Rq'$. Then, for $f(p) \in z(a)$ there exists $q' \in w(a)$ with $f(p)Rq'$; and by definition of y , there exists $q \in y(a)$ with $q' = g(q)$ as required.

In the second case, let $q \in y(a)$ so that $g(q) \in w(a)$. Again, from $z \text{Rel}(\mathcal{P}^A)(R) w$ it follows that there is $p' \in z(a)$ with $p'Rg(q)$. Now, $f(u(a)) \supseteq z(a)$ because $u \sqsubseteq^{\text{Conf}} z$, so there exists $p \in u(a) \cap f^{-1}(z(a))$ with $f(p) = p'$, as required. \square

As in the case of covariant-contravariant simulations, conformance simulations cannot be defined as coalgebraic simulations using neither a right-stable order nor a left-stable order. But we can find in the arguments above the basis for a decomposition of the involved order \sqsubseteq^{Conf} , according to the two cases in its definition. Once again \sqsubseteq^{Conf} is an action-distributive order on \mathcal{P}^A , but in order to obtain the adequate decomposition of \sqsubseteq^{Conf} now we also need to decompose the component orders \sqsubseteq^a .

Definition 7. We define the conformance orders \sqsubseteq^{C-0} , \sqsubseteq^{C0} , and \sqsubseteq^C on the functor \mathcal{P} by:

- $x_1 \sqsubseteq^{C0} x_2$ if $x_1 = \emptyset$ or $x_1 = x_2$.
- $x_1 \sqsubseteq^{C-0} x_2$ if $x_1 \supseteq x_2$ and $x_2 \neq \emptyset$, or $x_1 = x_2$.
- $x_1 \sqsubseteq^C x_2$ if $x_1 \sqsubseteq^{C-0} x_2$ or $x_1 \sqsubseteq^{C0} x_2$.

Proposition 6. The two relations \sqsubseteq^{C0} and \sqsubseteq^{C-0} commute with each other:

$$(\sqsubseteq^{C0} \circ \sqsubseteq^{C-0}) = (\sqsubseteq^{C-0} \circ \sqsubseteq^{C0}),$$

from where it follows that $(\sqsubseteq^{C0} \cup \sqsubseteq^{C-0})^* = (\sqsubseteq^{C0} \circ \sqsubseteq^{C-0}) = (\sqsubseteq^{C-0} \circ \sqsubseteq^{C0})$. We also have $\sqsubseteq^C = (\sqsubseteq^{C0} \circ \sqsubseteq^{C-0})$, from where we conclude that \sqsubseteq^C is indeed an order relation.

Proof. Let $u (\sqsubseteq^{C0} \circ \sqsubseteq^{C-0}) v$: there is some w such that $u \sqsubseteq^{C-0} w$ and $w \sqsubseteq^{C0} v$. We need to find w' such that $u \sqsubseteq^{C0} w'$ and $w' \sqsubseteq^{C-0} v$. If $w = \emptyset$ then it must be $u = \emptyset$ too, and we can take $w' = v$; otherwise, it must be $v = w$ and we can take $w' = u$. The other inclusion is similar. □

Corollary 5. The order \sqsubseteq^{Conf} defining conformance simulations can be decomposed into $\prod_{a \in A} \sqsubseteq^a$ where, for each $a \in A$, we have $\sqsubseteq^a = \sqsubseteq^C$ as defined above. Then, $\sqsubseteq^{Conf} = \prod_{a \in A} (\sqsubseteq^{a,-0} \cup \sqsubseteq^{a,0})^* = \prod_{a \in A} (\sqsubseteq^{a,-0}) \circ \prod_{a \in A} (\sqsubseteq^{a,0}) = \prod_{a \in A} (\sqsubseteq^{a,0}) \circ \prod_{a \in A} (\sqsubseteq^{a,-0})$, so that we obtain \sqsubseteq^{Conf} as the composition of a right-stable order and a left-stable order that commute with each other.

Proposition 7. For any pair of right (resp. left)-stable orders $\sqsubseteq^1, \sqsubseteq^2$ on F , their composition also defines a right (resp. left)-stable order on F .

Proof. Given $f : X \rightarrow Y$ we must show that

$$(id \times Ff)^{-1}(\sqsubseteq_Y^1 \circ \sqsubseteq_Y^2) \subseteq \coprod_{(Ff \times id)} (\sqsubseteq_X^1 \circ \sqsubseteq_X^2).$$

Let us assume that $(y, x) \in (id \times Ff)^{-1}(\sqsubseteq_Y^1 \circ \sqsubseteq_Y^2)$, that is, $y (\sqsubseteq^1 \circ \sqsubseteq^2) y' = Ff(x)$; then, there exists $y'' \in FY$ such that $y \sqsubseteq_Y^2 y''$ and $y'' \sqsubseteq_Y^1 y'$. Graphically,

$$\begin{array}{ccc}
 y & \sqsubseteq_Y^2 & y'' & \sqsubseteq_Y^1 & y' & & (8) \\
 & & & & \uparrow & & \\
 & & & & Ff & & \\
 & & & & \uparrow & & \\
 & & & & x & &
 \end{array}$$

Since \sqsubseteq_Y^1 is right-stable we have that $(id \times Ff)^{-1} \sqsubseteq_Y^1 \subseteq \coprod_{(Ff \times id)} \sqsubseteq_X^1$. Hence, there exists $x'' \in FX$ such that $Ff(x'') = y''$ and $x'' \sqsubseteq_X^1 x$, thus turning diagram (8) into the following:

$$\begin{array}{ccc}
 y & \sqsubseteq_Y^2 & y'' \\
 & & \uparrow Ff \\
 & & x'' \\
 & & \sqsubseteq_X^1 \quad x
 \end{array} \tag{9}$$

Now, we can apply right-stability of \sqsubseteq_Y^2 : since we have $(y, x'') \in (id \times Ff)^{-1} \sqsubseteq_Y^2 \subseteq \coprod_{(Ff \times id)} \sqsubseteq_X^2$, there exists $x' \in FX$ such that $Ff(x') = y$ and $x' \sqsubseteq_X^2 x''$. Thus, diagram (9) becomes

$$\begin{array}{ccc}
 y & & \\
 \uparrow Ff & & \\
 x' & \sqsubseteq_X^2 & x'' \sqsubseteq_X^1 x
 \end{array} \tag{10}$$

which means that there exist $x', x'' \in FX$ such that $Ff(x') = y$, $x' \sqsubseteq_X^2 x''$ and $x'' \sqsubseteq_X^1 x$, or equivalently, that $(y, x) \in \coprod_{(Ff \times id)} (\sqsubseteq_X^1 \circ \sqsubseteq_X^2)$, as we had to prove. \square

Proposition 8. *If \sqsubseteq^r is a right-stable order on F and \sqsubseteq^l is a left-stable order on F that commute with each other, then their composition defines a stable order on F . Moreover, the coalgebraic simulations for the order $\sqsubseteq = \sqsubseteq^r \circ \sqsubseteq^l$ can be equivalently defined as the $(\sqsubseteq^r \circ \text{Rel}(F)(R) \circ \sqsubseteq^l)$ -coalgebras.*

Proof. Let $R \subseteq Z \times W$ be a relation, $f : X \rightarrow Z$ and $g : Y \rightarrow W$ arbitrary functions, and $\sqsubseteq = \sqsubseteq^r \circ \sqsubseteq^l$. Let us suppose that $(u, v) \in (Ff \times Fg)^{-1}(\text{Rel}_{\sqsubseteq}(F)(R))$. Then, since \sqsubseteq^r and \sqsubseteq^l commute with each other, using Proposition 4, there exist z', w' such that

$$Ff(u) \sqsubseteq_Z^l z' \text{Rel}(F)(R) w' \sqsubseteq_W^r Fg(v). \tag{11}$$

If we write z for $Ff(u)$ and w for $Fg(v)$, then equation (11) is equivalent to

$$\begin{array}{ccc}
 z & \sqsubseteq_Z^l & z' \text{Rel}(F)(R) w' \sqsubseteq_W^r w \\
 \uparrow Ff & & \uparrow Fg \\
 u & & v
 \end{array} \tag{12}$$

and we have to show that $(u, v) \in \text{Rel}_{\sqsubseteq}(F)((f \times g)^{-1}(R))$, that is, that there exist x and y such that

$$u \sqsubseteq_X^l x \text{Rel}(F)((f \times g)^{-1}(R)) y \sqsubseteq_Y^r v.$$

Using that \sqsubseteq^r is right-stable on the rhs of equation (11), we get $(w', v) \in (id \times Fg)^{-1} \sqsubseteq_W^r \subseteq \coprod_{(Fg \times id)} \sqsubseteq_Y^r$, so that there is some $y \in FY$ such that $Fg(y) = w'$, with $y \sqsubseteq_Y^r v$. Graphically, diagram (12) becomes

$$\begin{array}{ccc}
 z & \sqsubseteq_Z^l & z' \text{Rel}(F)(R) w' \\
 \uparrow Ff & & \uparrow Fg \\
 u & & y \sqsubseteq_Y^r v
 \end{array} \tag{13}$$

Analogously, applying the left-stability of order \sqsubseteq_Z^l we get that there is some $x \in FX$ with $Ff(x) = z'$ such that $u \sqsubseteq_X^l x$. Or graphically,

$$\begin{array}{ccc}
 & z' & \text{Rel}(F)(R) & w' & \\
 & \uparrow Ff & & \uparrow Fg & \\
 u & \sqsubseteq_X^l & x & & y & \sqsubseteq_Y^r & v
 \end{array} \tag{14}$$

But diagram (14) is just what we had to prove, since we have found x, y such that $(x, y) \in (Ff \times Fg)^{-1}(\text{Rel}(F)(R)) = \text{Rel}(F)((f \times g)^{-1}(R))$ with $u \sqsubseteq_X^l x, y \sqsubseteq_Y^r v$ or, in other words, $(u, v) \in \text{Rel}_{\sqsubseteq}(F)((f \times g)^{-1}(R))$. \square

In particular, for our running example of conformance simulations we obtain the corresponding factorization of the definition of coalgebraic simulations for the order \sqsubseteq^{Conf} :

Corollary 6. *Coalgebraic simulations for the conformance order \sqsubseteq^{Conf} can be equivalently defined as the $(\prod_{a \in A} (\sqsubseteq_Y^{a, \neg 0}) \circ \text{Rel}(F)(R) \circ \prod_{a \in A} (\sqsubseteq_X^{a, 0}))$ -coalgebras.*

5 Conclusion

We have presented in this paper two new simulation orders induced by two criteria that capture the difference between input and output actions and the implementation notions that are formalized by the conformance relations.

In order to apply the general theory of coalgebraic simulations to them, we identified the corresponding orders on the functor defining labeled transition systems. However, it was not immediate to prove that the obtained orders had the desired good properties since the usual way to do it, namely, by establishing stability as a consequence of a stronger property that we have called right-stability, is not applicable in this case.

Trying to adapt that property to our situation we have discovered several interesting consequences. We highlight the fact that right-stability is an assymmetric property which has proved to be very useful for the study of a “reversible” concept such as that of relation, since it is clear that any structural result on the theory of relations should remain true when we reverse the relations, simply “observing” them “from the other side”. Two consequences of that assymmetric approach followed: first we noticed that we could use it to point the simulation orders in some natural way; secondly we also noticed that by dualizing the right-stability condition we could obtain left-stability.

But the crucial result in order to be able to manage more complicated simulation notions, as proved to be the case for our new covariant-contravariant simulations and the conformance simulations, was the discovery of the fact that both of them could be factorized into the composition of a right-stable and a left-stable component. Exploiting this decomposition we have been able to easily adapt all the techniques that had proved to be very useful for the case of right-stable orders.

We plan to expand our work here in two different directions. The first one is concerned with the two new simulated notions introduced in this paper: once we know that they can be defined as stable coalgebraic simulations and therefore have all the desired basic properties of simulations, we will continue with their study by integrating them

into our unified presentation of the semantics for processes [4]. Hence we expect to obtain, in particular, a clear relation between conformance similarity and the classic similarity orders as well as an algebraic characterization for the new semantics. In addition, we plan to continue with our study of stability, which has proved to be a crucial property in order to understand the notion of coalgebraic simulation, thus making it possible to apply the theory to other examples like those studied in this paper.

References

1. Aczel, P., Mendler, N.P.: A final coalgebra theorem. In: Dybjer, P., Pitts, A.M., Pitt, D.H., Poigné, A., Rydeheard, D.E. (eds.) *Category Theory and Computer Science*. LNCS, vol. 389, pp. 357–365. Springer, Heidelberg (1989)
2. Bloom, B., Istrail, S., Meyer, A.R.: Bisimulation can't be traced. *J. ACM* 42(1), 232–268 (1995)
3. de Frutos-Escrig, D., Gregorio-Rodríguez, C.: Universal coinductive characterisations of process semantics. In: Ausiello, G., Karhumäki, J., Mauri, G., Ong, C.-H.L. (eds.) *IFIP TCS*. IFIP, vol. 273, pp. 397–412. Springer, Heidelberg (2008)
4. de Frutos Escrig, D., Gregorio-Rodríguez, C., Palomino, M.: On the unification of semantics for processes: observational semantics. In: Nielsen, M., Kucera, A., Miltersen, P.B., Palamidessi, C., Tuma, P., Valencia, F.D. (eds.) *SOFSEM 2009*. LNCS, vol. 5404, pp. 279–290. Springer, Heidelberg (2009)
5. Groote, J.F., Vaandrager, F.W.: Structured operational semantics and bisimulation as a congruence. *Inf. Comput.* 100(2), 202–260 (1992)
6. Hughes, J., Jacobs, B.: Simulations in coalgebra. *TCS* 327(1-2), 71–108 (2004)
7. Jacobs, B., Hughes, J.: Simulations in coalgebra. In: Gumm, H.P. (ed.) *CMCS 2003: 6th International Workshop on Coalgebraic Methods in Computer Science*, vol. 82 (2003)
8. Larsen, K.G., Skou, A.: Bisimulation through probabilistic testing. *Inf. Comput.* 94(1), 1–28 (1991)
9. Leduc, G.: A framework based on implementation relations for implementing LOTOS specifications. *Computer Networks and ISDN Systems* 25(1), 23–41 (1992)
10. Lynch, N.A., Tuttle, M.R.: Hierarchical correctness proofs for distributed algorithms. In: *Sixth Annual ACM Symposium on Principles of Distributed Computing*, pp. 137–151 (1987)
11. Park, D.: Concurrency and automata on infinite sequences. In: Deussen, P. (ed.) *GI-TCS 1981*. LNCS, vol. 104, pp. 167–183. Springer, Heidelberg (1981)
12. Tretmans, J.: Conformance testing with labelled transition systems: Implementation relations and test generation. *Computer Networks and ISDN Systems* 29(1), 49–79 (1996)
13. van Glabbeek, R.J.: The linear time-branching time spectrum I: The semantics of concrete, sequential processes. In: Bergstra, J.A., Ponse, A., Smolka, S.A. (eds.) *Handbook of process algebra*, pp. 3–99 (2001)