Categorical logics for contravariant simulations, partial bisimulations, modal refinements and mixed transition systems⋆

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. Covariant-contravariant simulation and conformance simulation generalize plain simulation and try to capture the fact that it is not always the case that "the larger the number of behaviors, the better". We have previously studied some of their properties, showing that they can be presented as particular instances of the general notion of categorical simulation developed by Hughes and Jacobs and constructing the axiomatizations of the preorders defined by the simulation relations and their induced equivalences. We have also studied their logical characterizations and in this paper we continue with that study, presenting them as instantiations of the categorical results on simulation logics by Cîrstea. In addition, we continue exploring, now in this categorical framework, the relationship between covariant-contravariant simulation, partial bisimulation over labeled transition systems, refinement over modal transition systems and mixed transition systems.

1 Introduction and related work

Simulations are a very natural way to compare systems defined by labeled transition systems of other related mechanisms based on describing the behavior of states by means of the actions they can execute. However, the classic notion of simulation does not take into account the fact that whenever a system has several possibilities for the execution of an action, it will choose in an unpredictable manner resulting in more non-determinism and less control.

We have proposed two new simulation notions which are more suitable to deal with non-determinism [7]. On the one hand, covariant-contravariant simulations were designed to manage systems in which non-determinism arises because of the presence of both input and output actions; on the other hand, conformance simulations cope with having several options for the same action. In previous works we have proved that these simulations can be presented as instances of the coalgebraic simulation framework [7] and have also described their logical characterizations [8].

In this paper we continue with the study of the logics that characterize these two simulation notions, but now within the general categorical framework developed by Cîrstea in [5]. In addition, we also consider partial bisimulation [2], which turns out to

⋆ Research supported by the Spanish projects DESAFIOS10 TIN2009-14599-C03-01, TESIS TIN2009-14321-C02-01 and PROMETIDOS S2009/TIC-1465.

be just a particular case of covariant-contravariant simulation, as well as modal transition systems, a concept introduced by Boudol and Larson [3] and whose associated notion of refinement clearly resembles our covariant-contravariant simulations; in doing so, we expand on the comparison we started in [1] between these related notions. Actually, although more interesting, modal transition systems are just a particular case of mixed transition systems; by reusing many of the concepts used for the former, we show how to also obtain a logic for the latter for which, unlike the others, we were not aware of a previous non-coalgebraic logical characterization.

Now, besides describing a method for obtaining logical characterizations, [5] also explains how to build new logics in a compositional manner out of known ones. Unfortunately, our simulations were not amenable to this methodology and we were forced to start from scratch. As a consequence, and besides the characterization for mixed transition systems, the main contribution of this work is the application of the ideas in [5] to interesting case studies such as modal refinement or contravariant simulation, in what we believe is a nice illustration of the methods involved.

All the missing proofs can be found in the extended version at `http://maude.sip.ucm.es/~miguelpt/bibliography.html`.

2 Preliminaries

In this section we summarize some definitions and concepts from [5, 7, 1, 3] and introduce the notation we are going to use. Let us recall our two simulation notions:

Definition 1. *Given $P = (P, A, \rightarrow_P)$ and $Q = (Q, A, \rightarrow_Q)$, two labeled transition systems (LTS) for the alphabet A, and $\{A^r, A^l, A^{bi}\}$ a partition of this alphabet, a (A^r, A^l)-**simulation** (or just a covariant-contravariant simulation) between them is a relation $S \subseteq P \times Q$ such that for every pSq we have:*

- *For all $a \in A^r \cup A^{bi}$ and all $p \xrightarrow{a} p'$ there exists $q \xrightarrow{a} q'$ with $p' S q'$.*
- *For all $a \in A^l \cup A^{bi}$, and all $q \xrightarrow{a} q'$ there exists $p \xrightarrow{a} p'$ with $p' S q'$.*

We will write $p \lesssim_{CC} q$ if there exists a covariant-contravariant simulation S such that pSq.

Definition 2. *Given $P = (P, A, \rightarrow_P)$ and $Q = (Q, A, \rightarrow_Q)$ two labeled transition systems for the alphabet A, a **conformance simulation** between them is a relation $R \subseteq P \times Q$ such that whenever pRq, then:*

- *For all $a \in A$, if $p \xrightarrow{a}$, then $q \xrightarrow{a}$ (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 $p'Rq'$.*

We will write $p \lesssim_{CS} q$ if there exists a conformance simulation R such that pRq.

Now, we recall the definitions for modal transition systems.

Definition 3. *For a set of actions A, a* modal transition *system (MTS) is defined by the triple* $(P, \rightarrow_\diamond, \rightarrow_\Box)$*, where P is a set of states and* $\rightarrow_\diamond, \rightarrow_\Box \subseteq P \times A \times P$ *are transition relations such that* $\rightarrow_\Box \subseteq \rightarrow_\diamond$.

The transitions in $\rightarrow_\Box$ are called the *must transitions* and those in $\rightarrow_\diamond$ are the *may transitions*. In an MTS, each must transition is also a may transition, which intuitively means that any required transition is also allowed.

The notion of (modal) refinement $\sqsubseteq$ over MTSs that we now proceed to introduce is based on the idea that if $p \sqsubseteq q$ then q is a 'refinement' of the specification p. In that case, intuitively, q may be obtained from p by possibly turning some of its may transitions into must transitions.

Definition 4. *A relation* $R \subseteq P \times Q$ *is a* refinement relation *between two modal transition systems if, whenever* $p\ R\ q$*:*

- $p \xrightarrow{a}_\Box p'$ *implies that there exists some* q' *such that* $q \xrightarrow{a}_\Box q'$ *and* $p'\ R\ q'$*;*
- $q \xrightarrow{a}_\diamond q'$ *implies that there exists some* p' *such that* $p \xrightarrow{a}_\diamond p'$ *and* $p'\ R\ q'$.

We write $\sqsubseteq$ *for the largest refinement relation.*

Finally, we briefly recall the basic concepts on categorical simulations that we are going to use in Section 3. First, we will model finitary LTS by coalgebras $c : X \longrightarrow \mathcal{P}_\omega X^A$ for the finite powerset functor $\mathcal{P}_\omega^A$, where, as usually, we will denote $x' \in c(x)(a)$ by $x \xrightarrow{a} x'$. We can also see modal transition systems as coalgebras for the functor $F = \mathcal{P}(id \times \{\diamond, \Box\})^A$, where $\{\diamond, \Box\}$ is a set with two elements where $\Box$ stands for must transitions and $\diamond$ for may transitions. We will make intensive use of the following notation along the paper.

$c(x)(a)_\Box = \{x' \in X \mid (x', \Box) \in c(x)(a)\}$, and
$c(x)(a)_\diamond = \{x' \in X \mid (x', \sigma') \in c(x)(a), \text{ with } \sigma' \in \{\diamond, \Box\}\}$.

Note that with the previous definition we do not have necessarily $\rightarrow_\Box \subseteq \rightarrow_\diamond$, but that requirement is built-in in our notation since we have that $c(x)(a)_\Box \subseteq c(x)(a)_\diamond$.

We will denote by **Sets** the category of sets and by **Rel** the category of relations. Given an endofunctor T : **Sets** $\longrightarrow$ **Sets**, a *monotonic T-relator* [14, 5] is an endofuntor Γ : **Rel** $\longrightarrow$ **Rel** such that $\mathsf{U} \circ \Gamma = (T \times T) \circ \mathsf{U}$, $=_{TX} \subseteq \Gamma(=_X)$, and $\Gamma(S \circ R) = \Gamma(S) \circ \Gamma(R)$, where U : **Rel** $\longrightarrow$ **Sets** $\times$ **Sets** is the forgetful functor. A Γ-*simulation* between coalgebras (X, c) and (Y, d) is just a Γ-coalgebra of the form $(R, (c, d))$, i.e, a relation R such that xRy implies $c(x)\Gamma(R)d(y)$.

3 Logical characterizations of the semantics

For the logical characterization of the covariant-contravariant and conformance simulations we will follow the general inductive methodology introduced in [5]. First, we will define the syntax and semantics of the logics by means of a "language constructor" and its associated notion of semantics. In fact, both constructions only define a single step that must be successively applied in an iterative process that ends up with the definitive syntax and semantics. The next stage consists in showing that the "one-step" semantics is adequate for the corresponding simulation notions. Finally, we will build the concrete logics for coalgebras which characterize the new similarities, which are equivalent to the logics we defined in [8].

We begin with the covariant-contravariant simulation because we consider it more illustrative.

3.1 Covariant-contravariant simulations

Before starting with the methodology in [5], we must show that covariant-contravariant simulations can be modeled using monotonic relators [14, 5].

Definition 5 (Covariant-contravariant simulation relator). *Let $R \subseteq Q \times P$ be a relation, $g : Q \longrightarrow \mathcal{P}_\omega Q^A$ and $f : P \longrightarrow \mathcal{P}_\omega P^A$ LTS, and $\{A^r, A^l, A^{bi}\}$ a partition of A. We define the $\mathcal{P}_\omega^A$-relator $\Gamma_{\mathrm{CC}} : \mathbf{Rel} \longrightarrow \mathbf{Rel}$ for covariant-contravariant simulations by $g\ \Gamma_{\mathrm{CC}}(R)\ f$ iff:*

- *for all $a \in A^r \cup A^{bi}$ and all $p \in f(a)$ there exists $q \in g(a)$ with qRp.*
- *for all $a \in A^l \cup A^{bi}$, and all $q \in g(a)$ there exists $p \in f(a)$ with qRp.*

Proposition 1. *The simulation notion defined by the relator Γ_{CC} coincides with the notion of covariant-contravariant simulation.*

Proof (Sketch). First, for the implication from right to left, let us see the case of $a \in A^r \cup A^{bi}$ and $p' \in f(p)(a)$ then $p \xrightarrow{a} p'$. Using that pRq, there exists q' such that $q \xrightarrow{a} q'$ with $p'Rq'$, that is, there is $q' \in g(q)(a)$ with $q'R^{op}p'$.

For the other implication, again, let $a \in A^r \cup A^{bi}$ and $p \xrightarrow{a} p'$, then $p' \in f(p)(a)$ with $a \in A^r \cup A^{bi}$ and by definition of the relator there exists $q' \in g(q)(a)$ with $q'Rp'$, that is, we have $q \xrightarrow{a} q'$ with $p'R^{op}q'$. □

The first step for defining the logic is to define its syntax by means of what is called a *language constructor*. From now on we work with a signature $\Sigma_B \subseteq \{\mathsf{tt}, \mathsf{ff}, \wedge, \vee, \bigwedge, \bigvee\}$[1] and its corresponding category $Alg(\Sigma_B)$ of algebras.

Definition 6 ([5]). *A language constructor is an accessible endofunctor $\mathsf{S} : Alg(\Sigma_B) \longrightarrow Alg(\Sigma_B)$ and the language $\mathcal{L}(\mathsf{S})$ induced by S is the initial algebra of S.*

In most interesting cases the language $\mathcal{L}(\mathsf{S})$ is given by $\bigcup_n L_n(\mathsf{S})$, with $L_0(\mathsf{S})$ the initial Σ_B-algebra and $L_{n+1}(\mathsf{S}) = \mathsf{S}(L_n(\mathsf{S}))$.

In order to define the logic for covariant-contravariant simulations we proceed as in [8]. First, given $\Sigma = \{\mathsf{tt}, \wedge\}$, the language $\mathcal{L}(\mathsf{S}_\sqsupseteq)$ characterizing the simulation semantics is defined in [5] as the language constructor taking the Σ-algebra L to the free Σ-algebra over the set $\{\diamond\varphi \mid \varphi \in L\}$. It is also shown in [5] that for LTS we could define $\mathcal{L}(\mathsf{S}_\sqsupseteq^{\mathsf{A}})$ as the language constructor taking the Σ-algebra L to the free Σ-algebra over the set $\{\langle a\rangle\varphi \mid \varphi \in L\}$.

[1] Although in [5] the element ff is not used, we will need it for the logics characterizing covariant-contravariant simulations and modal refinement.

If we compare it with the Hennessy-Milner logic $\mathcal{L}_{HM}$ [10], it can be noted that the main difference is that negation is not present. Obviously, this must be the case to capture a strict order that is not an equivalence relation, such as $\lesssim_{CC}$. However, adding both the constant ff and the disjunction $\vee$ to Σ does no harm, thus obtaining $\mathcal{L}(\bar{\mathsf{S}}^{\mathsf{A}}_{\supseteq})$ which also characterizes $\lesssim_S$ for LTS.

As we did in [8], the inspiration to obtain the logic characterizing $\lesssim_{CC}$ comes from the fact that if we only have contravariant actions, then $\lesssim_{CC}$ becomes $\lesssim_S^{-1}$, and therefore by negating all the formulas in $\mathcal{L}(\bar{\mathsf{S}}^{\mathsf{A}}_{\supseteq})$ we would obtain the desired characterization (that is why we need ff). In particular, for the modal operator $\langle a\rangle$ we would obtain its dual form $[a]$.

Then, in the presence of both covariant and contravariant actions, we need to consider the existential operator $\langle a\rangle$ for $a \in A^r \cup A^{bi}$ and the universal operator $[b]$ for $b \in A^l \cup A^{bi}$, thus obtaining the following definition of the syntax of the logic for covariant-contravariant simulations.

Definition 7. *Let $\Sigma_B = \{\mathsf{tt}, \mathsf{ff}, \wedge, \vee\}$ and let $\mathrm{S}_{CC} : \mathrm{Alg}(\Sigma_B) \to \mathrm{Alg}(\Sigma_B)$ denote the language constructor taking a Σ_B-algebra L to the free Σ_B-algebra over the set $\{[b]\varphi \mid b \in A^l \cup A^{bi}, \varphi \in L\} \cup \{\langle a\rangle\varphi \mid a \in A^r \cup A^{bi}, \varphi \in L\}$. Then, the language $\mathcal{L}(\mathrm{S}_{CC})$ can be generated using the following syntax:*

$$\varphi ::= \mathsf{tt} \mid \mathsf{ff} \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid [b]\varphi \mid \langle a\rangle\varphi\,.$$

Now, in order to define the semantics of the operators above we need some technical definitions.

Definition 8 ([5]). *An interpretation of a Σ_B-algebra L over a set X is a Σ_B-algebra morphism $d : L \longrightarrow \mathcal{P}X$.*

Intuitively, an interpretation gives for each operator in the syntax (that is, of the language $\mathcal{L}(\mathrm{S}_{CS})$) all the elements (of a given set X) that satisfy a formula, that is, $x \in d(\varphi)$ means that the formula φ holds in x. Interpretations define a category. A map between interpretations $d : L \longrightarrow \mathcal{P}X$ and $d' : L' \longrightarrow \mathcal{P}X'$ is a pair (l, f) with $l : L \longrightarrow L'$ a Σ_B-algebra morphism and $f : X' \longrightarrow X$ a function such that $\hat{\mathcal{P}}f \circ d = d' \circ l$ (where $\hat{\mathcal{P}}$ denotes the contravariant powerset functor). We denote this category of interpretations by $\mathbf{Int}_B$, with $\mathsf{L} : \mathbf{Int}_B \longrightarrow Alg(\Sigma_B)$ the functor taking d to L and $\mathsf{E} : \mathbf{Int}_B \longrightarrow \mathbf{Sets}^{op}$ the functor taking d to X.

Recall that in order to define the semantics for logics, we must first define the semantics of a single step. This single step is formalized as follows.

Definition 9 ([5]). *A T-semantics for a language constructor S is a functor $\mathbb{S} : \mathbf{Int}_B \longrightarrow \mathbf{Int}_B$ such that $\mathsf{L} \circ \mathbb{S} = \mathsf{S} \circ \mathsf{L}$ and $\mathsf{E} \circ \mathbb{S} = T^{op} \circ \mathsf{E}$. Thus, a T-semantics for S takes an interpretation $d : L \longrightarrow \mathcal{P}X$ to an interpretation $d' : \mathsf{S}L \longrightarrow \mathcal{P}TX$.*

For our concrete case of covariant-contravariant simulations the interesting cases are the definition of the semantic for the two modal operators. In [5] the semantics for the operator $\diamond$ is defined as $d'(\diamond\varphi) = \{Y \in \mathcal{P}_\omega X \mid Y \cap d(\varphi) \neq \emptyset\}$. So, it is easy to see that if we consider the operator $\langle a\rangle$ we have $d'(\langle a\rangle\varphi) = \{f : A \longrightarrow \mathcal{P}_\omega X \mid f(a) \cap d(\varphi) \neq$

$\emptyset\}$. Analogously, following the classical definitions of the modal operators in [10] and our work in [8], in order to define the semantics for $[b]$ we must consider not just $f(b) \cap d(\varphi) \neq \emptyset$ but $f(b) \subseteq d(\varphi)$ since with the classical interpretation $p \models [b]\varphi$ means that $p' \models \varphi$ for all $p \stackrel{b}{\longrightarrow} p'$; thus, all the successors must be in the interpretation and not just one.

Hence, we have the following.

Definition 10. *A $\mathcal{P}_\omega^A$-semantics for S_{CC} is given by the functor $\mathbb{S}_{CC} : \mathbf{Int}_B \longrightarrow \mathbf{Int}_B$ taking an interpretation $d : L \longrightarrow \mathcal{P}X$ to an interpretation $d' : \mathsf{S}_{CC}(L) \longrightarrow \mathcal{P}(\mathcal{P}_\omega X^A)$ defined by:*

- $d'(\mathsf{tt}) = \mathcal{P}_\omega X^A$.
- $d'(\mathsf{ff}) = \emptyset$.
- $d'(\varphi \wedge \psi) = d'(\varphi) \cap d'(\psi)$.
- $d'(\varphi \vee \psi) = d'(\varphi) \cup d'(\psi)$.
- $d'([b]\varphi) = \{f : A \longrightarrow \mathcal{P}_\omega X \mid f(b) \subseteq d(\varphi)\}$.
- $d'(\langle a \rangle \varphi) = \{f : A \longrightarrow \mathcal{P}_\omega X \mid f(a) \cap d(\varphi) \neq \emptyset\}$.

Note that, since an interpretation between the Σ_B-algebras $\mathcal{L}(\mathsf{S}_{CC})$ and $\mathcal{P}X$ is a morphism, the value of d' on tt, ff, $\wedge$ and $\vee$ is imposed.

Next, we show that the semantics $\mathbb{S}_{CC}$ is adequate for covariant-contravariant simulations. The notion of adequacy is given by "preserving expressiveness". Informally, a preorder is expressive if from $a \leq b$ it follows that b satisfies a logical formula (according to the interpretation) whenever a does; a semantics preserve expressiveness whenever it maps expressive interpretations and preorders into expressive ones. The following definition makes these concepts precise.

Definition 11 ([5]). *Given an interpretation $d : L \longrightarrow \mathcal{P}X$, for $x, y \in X$ we write $y \geq_L x$ if $y \in d(\varphi)$ whenever $x \in d(\varphi)$. We will say that d is expressive for a preorder $R \subseteq X \times X$ if $R = \geq_L$, in other words, yRx if and only if $y \in d(\varphi)$ whenever $x \in d(\varphi)$.*

*Given a T-relator Γ : **Rel** $\longrightarrow$ **Rel** and a language constructor $\mathsf{S} : Alg(\Sigma_B) \longrightarrow Alg(\Sigma_B)$, we will say that a T-semantics $\mathbb{S}$ for S preserves expressiveness w.r.t. Γ if it maps an interpretation $d : L \longrightarrow \mathcal{P}X$ which is expressive for $R \subseteq X \times X$, into an interpretation $d' : \mathsf{S}(L) \longrightarrow \mathcal{P}TX$ which is expressive for ΓR.*

Proposition 2. *The semantics $\mathbb{S}_{CC}$ for S_{CC} preserves expressiveness w.r.t. Γ_{CC}.*

Finally, the last step of the construction in [5] is the definition of the "definitive" logic for a coalgebra induced by a relator. The semantics of this logic will be built as the limit of the "single step" semantics.

Definition 12 ([5]). *For any ordinal α, given $(Z_\alpha), (\rho_\alpha^\beta : Z_\alpha \longrightarrow Z_\beta)_{\beta \leq \alpha}$, the final sequence of the functor T, an interpretation $d : \mathcal{L} \longrightarrow \mathcal{P}Z_\alpha$ induces a logic $(\mathcal{L}, \models)$ for T-coalgebras with*

$$c \models_\gamma \varphi \text{ if and only if } \gamma_\alpha(c) \in d(\varphi),$$

where $(\gamma_\alpha : C \longrightarrow Z_\alpha)$ denotes the cone over the final sequence of T defined as follows:

- $\gamma_0 : C \longrightarrow 1$ *is the unique such map.*

– $\gamma_\alpha = T\gamma_\beta \circ \gamma$.
– γ_ω *is the unique arrow satisfying* $\rho^\omega_\alpha \circ \gamma_\omega = \gamma_\alpha$ *for each* $\alpha < \omega$.

In particular, if the final sequence of Γ : **Rel** $\longrightarrow$ **Rel** stabilizes at α, then the logic induced by $\mathbb{S}$ and Γ [5] is the logic induced by the interpretation $d_\alpha : L_\alpha \longrightarrow \mathcal{P}Z_\alpha$. Then, if $\mathbb{S}$ preserves expressiveness w.r.t. Γ, the final sequence of T stabilizes at α, and the initial sequence of S stabilizes at α, the final sequence of Γ also stabilizes at α [5, Prop. 61]. If that is the case, the logic induced by $\mathbb{S}$ and Γ characterizes the similarity relation [5, Cor. 60].

In our case, we finally obtain the following proposition.

Proposition 3. *For an LTS* $\gamma : C \longrightarrow \mathcal{P}_\omega C^A$*, the logic which characterizes covariant-contravariant simulation is given by:*

– $c \models_\gamma \mathsf{tt}$.
– $c \not\models_\gamma \mathsf{ff}$.
– $c \models_\gamma \varphi_1 \wedge \varphi_2$ *if and only if* $c \models_\gamma \varphi_1$ *and* $c \models_\gamma \varphi_2$.
– $c \models_\gamma \varphi_1 \vee \varphi_2$ *if and only if* $c \models_\gamma \varphi_1$ *or* $c \models_\gamma \varphi_2$.
– $c \models_\gamma \langle a\rangle\varphi$ *if and only if* $c' \models_\gamma \varphi$ *for some* $c' \in \gamma(c)(a)$.
– $c \models_\gamma [b]\varphi$ *if and only if* $c' \models \varphi$ *for all* $c' \in \gamma(c)(b)$.

Proof (Sketch). First, we have to consider the initial sequence of S_{CC}, $(L_\alpha), (\iota^\alpha_\beta : L_\beta \longrightarrow L_\alpha)_{\beta\leq\alpha}$, the final sequence of $T = \mathcal{P}^A_\omega$, $(Z_\alpha), (\rho^\beta_\alpha : Z_\alpha \longrightarrow Z_\beta)_{\beta\leq\alpha}$, and the sequence $\hat\rho^\alpha_\beta = \hat{\mathcal{P}}(\rho^\beta_\alpha)$ built from the terminal sequence of Z_α by applying the contravariant functor $\hat{\mathcal{P}}$. Note that Z_α contains all the possible (up to bisimilarity) A-trees of depth at most $\alpha + 1$ with branching at most $|A|\alpha$, whereas Z_ω contains all the A-trees, possibly with infinitely many branches and/or infinite depth, and $Z_{\omega+k}$ is the set of A-trees (possibly infinite) such that their first k floors are finitely branching.

Next, following [5, Prop. 57], we have to define the initial segment of the initial sequence $(d_\alpha), ((\iota^\alpha_\beta, \rho^\beta_\alpha) : d_\beta \longrightarrow d_\alpha)_{\beta\leq\alpha}$ of $\mathbb{S}_{CC}$, where $d_{\alpha+1} = \mathbb{S}_{CC}(d_\alpha)$. For each $0 \leq \alpha < \omega$, we have to follow Definition 10 and, for example, we obtain $d_{\alpha+1}(\mathsf{tt}) = \mathcal{P}_\omega Z^A_\alpha$, $d_{\alpha+1}(\mathsf{ff}) = \emptyset$, $d_{\alpha+1}(\varphi_1 \wedge \varphi_2) = d_{\alpha+1}(\varphi_1) \cap d_{\alpha+1}(\varphi_2)$ or $d_{\alpha+1}(\langle a\rangle\varphi) = \{f : A \longrightarrow \mathcal{P}_\omega Z_\alpha \mid f(a) \cap d_\alpha(\varphi) \neq \emptyset\}$.

For the case of d_ω we have $d_\omega([a]\varphi) = \{u \in Z_\omega \mid \rho^\omega_\alpha(u) \in d_\alpha([a]\varphi)\} = \hat\rho^\alpha_\omega(d_\alpha([a]\varphi))$ and $d_\omega(\langle a\rangle\varphi) = \{u \in Z_\omega \mid \rho^\omega_\alpha(u) \in d_\alpha(\langle a\rangle\varphi)\} = \hat\rho^\alpha_\omega(d_\alpha(\langle a\rangle\varphi))$. An important step in the proof is to check that d_ω is well-defined and that it is a limiting element of the sequence (see the details at [9]).

Now, since the initial sequence of S_{CC} stabilizes at ω and the final sequence of T stabilizes at $\omega + \omega$, we also have that the initial sequence of $\mathbb{S}_{CC}$ stabilizes at $\omega + \omega$ [5, Prop. 61]. Let $\gamma : C \longrightarrow \mathcal{P}_\omega C^A$ be a labeled transition system, with $\{A^r, A^l, A^{bi}\}$ a partition of A: by Definition 12, when considering the induced logic, we must work with $d_{\omega+\omega}$ and $\gamma_{\omega+\omega}$. But, in fact, $\rho^{\omega+k}_\omega : Z_{\omega+k} \longrightarrow Z_\omega$ is a monomorphism [9] so, since $\gamma_\omega = \rho^{\omega+\omega}_\omega \circ \gamma_{\omega+\omega}$, $\gamma_{\omega+\omega}(c) = \gamma_\omega(c)$. On the other hand, since $\hat\rho^\omega_{\omega+k} = \hat{\mathcal{P}}(\rho^{\omega+k}_\omega) : \mathcal{P}Z_\omega \longrightarrow \mathcal{P}Z_{\omega+k}$ is an epimorphism [9], we have that $d_{\omega+\omega} : L_\omega \longrightarrow \mathcal{P}Z_{\omega+\omega}$ and $d_{\omega+\omega}(\varphi) \subsetneq d_\omega(\varphi)$, because in $d_\omega(\varphi)$ we also have infinitely branching A-trees as possible behaviors of φ. However, since $\gamma_\omega(c) = \gamma_{\omega+\omega}(c)$ is a finitely branching A-tree, it turns out that $\gamma_{\omega+\omega}(c) \in d_{\omega+\omega}(\varphi)$ if and only if $\gamma_\omega(c) \in d_\omega(\varphi)$, that is, we can just consider γ_ω and d_ω.

Finally, the remaining of the proof consists in the application of Definition 12 for γ_ω and d_ω, that is, $c \models_\gamma \varphi$ if and only if $\gamma_\omega(c) \in d_\omega(\varphi)$ (see again [9] for the details). □

Hence, by Proposition 3, the logic induced by $\mathbb{S}_{CC}$ and Γ_{CC} is equivalent to the logic for covariant-contravariant simulation in [8].

3.2 Partial bisimulation

Partial bisimulation is defined in [2] as a behavioural relation over LTSs for studying the theory of supervisory control [13] in a concurrency-theoretic framework. In [2], the authors considered LTSs that also include a termination predicate ↓ over states. For the sake of simplicity, since its role is orthogonal to our aims in this paper, we simply omit it in what follows.

Definition 13. *A* partial bisimulation with bisimulation set *B between two LTSs P and Q is a relation $R \subseteq P \times Q$ such that, whenever $p\ R\ q$:*

- *For all $a \in A$, if $p \xrightarrow{a} p'$ then there exists some $q \xrightarrow{a} q'$ with $p'\ R\ q'$.*
- *For all $b \in B$, if $q \xrightarrow{b} q'$ then there exists some $p \xrightarrow{b} p'$ with $p'\ R\ q'$.*

We write $p \lesssim_B q$ if $p\ R\ q$ for some partial bisimulation with bisimulation set B.

In [1] we proved that partial bisimulation is a particular case of covariant-contravariant simulation, when the LTS P has signature $A^r = A \setminus B$, $A^l = \emptyset$ and $A^{bi} = B$. Hence, instantiating Proposition 3 with this particular case we obtain the same logic as in [1], which is simpler than that proposed in [2].

3.3 Conformance simulations

As we did in Section 3.1, we can apply the methodology in [5] to obtain the logical characterization of conformance simulations. First, we define the corresponding relator and prove that it defines the same simulation notion as the non-coalgebraic one.

Definition 14 (Conformance simulation relator). *Given $R \subseteq Q \times P$, $f \in \mathcal{P}_\omega P^A$ and $g \in \mathcal{P}_\omega Q^A$, we define the $\mathcal{P}_\omega^A$-relator $\Gamma_{\mathrm{CS}} : \mathbf{Rel} \longrightarrow \mathbf{Rel}$ for conformance simulation by $g\ \Gamma_{\mathrm{CS}}(R)\ f$ iff*

- *for each $a \in A$, $f(a) \neq \emptyset$ implies $g(a) \neq \emptyset$.*
- *for all $a \in A$, if $q' \in g(a)$ and $f(a) \neq \emptyset$ then there is $p' \in f(a)$ such that $q'Rp'$.*

Proposition 4. *The simulation notion defined by the relator Γ_{CS} coincides with the notion of conformance simulation.*

Next, we define the corresponding syntax.

Definition 15. *Let $\Sigma_B = \{\mathsf{tt}, \wedge, \vee\}$ and $\mathrm{S}_{CS} : \mathrm{Alg}(\Sigma_B) \rightarrow \mathrm{Alg}(\Sigma_B)$ denote the language constructor taking a Σ_B-algebra L to the free Σ_B-algebra over the set $\{[a]\varphi \mid a \in A, \varphi \in L\}$. Then, the language $\mathcal{L}(\mathrm{S}_{CS})$ is that generated using the following syntax:*

$$\varphi ::= \mathsf{tt} \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid [a]\varphi\,.$$

Note that in order to define the syntax for conformance simulation logic we do not consider ff since we do not have two kinds of modal operators with different nature (as opposed to the case of covariant-contravariant simulation). Nevertheless, we could add ff to our logic without changing its meaning. This is make clearer in the following definition that gives us the semantics.

Definition 16. *The $\mathcal{P}_\omega^A$-semantics for S_{CS} is given by the functor $\mathbb{S}_{CS} : \mathbf{Int}_B \longrightarrow \mathbf{Int}_B$ taking an interpretation $d : L \longrightarrow \mathcal{P}X$ to an interpretation $d' : \mathrm{S}_{CS}(L) \longrightarrow \mathcal{P}(\mathcal{P}_\omega X^A)$ defined by:*

- $d'(\mathsf{tt}) = \mathcal{P}_\omega X^A$.
- $d'(\varphi \wedge \psi) = d'(\varphi) \cap d'(\psi)$.
- $d'(\varphi \vee \psi) = d'(\varphi) \cup d'(\psi)$.
- $d'([a]\varphi) = \{f : A \longrightarrow \mathcal{P}_\omega X \mid f(a) \neq \emptyset \text{ and } f(a) \subseteq d(\varphi)\}$.

Again, as we saw in [8], in order to define the semantics for conformance simulation we need to define the operator $[a]$, which captures the idea of "having just one a-action is better than having more", by imposing that all the elements in $f(a)$ must (non-trivially) satisfy the formula φ. The next step is to prove that it is adequate for conformance simulations.

Proposition 5. *The semantics $\mathbb{S}_{CS}$ for S_{CS} preserves expressiveness w.r.t.* Γ_{CS}.

Finally, we obtain the following logic.

Proposition 6. *For an LTS $\gamma : C \longrightarrow \mathcal{P}_\omega C^A$, the logic which characterizes conformance simulation is given by:*

- $c \models_\gamma \mathsf{tt}$.
- $c \models_\gamma \varphi_1 \wedge \varphi_2$ *if and only if* $c \models_\gamma \varphi_1$ *and* $c \models_\gamma \varphi_2$.
- $c \models_\gamma \varphi_1 \vee \varphi_2$ *if and only if* $c \models_\gamma \varphi_1$ *or* $c \models_\gamma \varphi_2$.
- $c \models_\gamma [a]\varphi$ *if and only if* $\gamma(c)(a) \neq \emptyset$ *and* $c_i \models_\gamma \varphi$*, for all* $c_i \in \gamma(c)(a)$.

Proof (Sketch). The proof is essentially the same as that for Proposition 3. The only main difference is in the final proof of $c \models_\gamma [a]\varphi$, where we have to take into account that $\gamma(c)(a) \neq \emptyset$. □

Hence, Proposition 6 shows that the logic induced by $\mathbb{S}_{CS}$ and Γ_{CS} is equivalent to the logic for conformance simulation defined at [8].

3.4 Modal refinement

Again, we can apply the methodology in [5] to obtain the logical characterization of modal refinement between modal transition systems. First, we define the corresponding relator and prove that it defines the same simulation notion as the non-coalgebraic one.

Definition 17 (Modal refinement relator). *Given $R \subseteq Q \times P$, $g : Q \longrightarrow \mathcal{P}_\omega(Q \times \{\diamond, \Box\})^A$ and $f : P \longrightarrow \mathcal{P}_\omega(P \times \{\diamond, \Box\})^A$, we define the $\mathcal{P}_\omega(id \times \{\diamond, \Box\})^A$-relator Γ_{ref} :* **Rel** $\longrightarrow$ **Rel** *for modal refinement by $g\ \Gamma_{\mathrm{ref}}(R)\ f$ iff*

– *for all* $a \in A$, *if* $p' \in f(a)_\Box$ *then there is* $q' \in g(a)_\Box$ *such that* $q'Rp'$.
– *for all* $a \in A$, *if* $q' \in g(a)_\diamond$ *then there is* $p' \in f(a)_\diamond$ *such that* $q'Rp'$.

Proposition 7. *The simulation notion defined by the relator* Γ_{ref} *coincides with the notion of modal refinement.*

Next, we define the corresponding syntax and semantics.

Definition 18. *Let* $\Sigma_B = \{\mathsf{tt}, \mathsf{ff}, \wedge, \vee\}$ *and* $\mathrm{S}_{ref} : \mathrm{Alg}(\Sigma_B) \to \mathrm{Alg}(\Sigma_B)$ *denote the language constructor taking a* Σ_B*-algebra* L *to the free* Σ_B*-algebra over the set* $\{[a]\varphi \mid a \in A, \varphi \in L\} \cup \{\langle a\rangle\varphi \mid a \in A, \varphi \in L\}$*. Then, the language* $\mathcal{L}(\mathrm{S}_{ref})$ *is that generated using the following syntax:*

$$\varphi ::= \mathsf{tt} \mid \mathsf{ff} \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid [a]\varphi \mid \langle a\rangle\varphi\,.$$

Definition 19. *The* $\mathcal{P}_\omega(id \times \{\diamond, \Box\})^A$*-semantics for* S_{ref} *is given by the functor* $\mathbb{S}_{ref} : \mathbf{Int}_B \longrightarrow \mathbf{Int}_B$ *taking an interpretation* $d : L \longrightarrow \mathcal{P}X$ *to an interpretation* $d' : \mathrm{S}_{ref}(L) \longrightarrow \mathcal{P}(\mathcal{P}_\omega(X \times \{\diamond, \Box\})^A)$ *defined by:*

– $d'(\mathsf{tt}) = \mathcal{P}_\omega(X \times \{\diamond, \Box\})^A$.
– $d'(\mathsf{ff}) = \emptyset$.
– $d'(\varphi \wedge \psi) = d'(\varphi) \cap d'(\psi)$.
– $d'(\varphi \vee \psi) = d'(\varphi) \cup d'(\psi)$.
– $d'([a]\varphi) = \{f : A \longrightarrow \mathcal{P}_\omega(X \times \{\diamond, \Box\}) \mid f(a)_\diamond \subseteq d(\varphi)\}$.
– $d'(\langle a\rangle\varphi) = \{f : A \longrightarrow \mathcal{P}_\omega(X \times \{\diamond, \Box\}) \mid f(a)_\Box \cap d(\varphi) \neq \emptyset\}$.

In order to define the semantics for modal refinements we have needed to define two operators $[a]$ and $\langle a\rangle$, where the first one captures the transitions that the process may do, whereas the second one captures the transitions that the process must do. It is not surprising to note that, in particular, the definitions of these two modal operators are essentially the same as those for covariant-contravariant simulation, but taking into account in each case the "must" or "may" transitions.

The next step is to prove that it is adequate for modal refinement, and to obtain the corresponding logic.

Proposition 8. *The semantics* $\mathbb{S}_{ref}$ *for* S_{ref} *preserves expressiveness w.r.t.* Γ_{ref}.

Proposition 9. *For an MTS* $\gamma : C \longrightarrow \mathcal{P}_\omega(C \times \{\diamond, \Box\})^A$*, the logic which characterizes modal refinement is given by:*

– $c \models_\gamma \mathsf{tt}$.
– $c \models_\gamma \varphi_1 \wedge \varphi_2$ *if and only if* $c \models_\gamma \varphi_1$ *and* $c \models_\gamma \varphi_2$.
– $c \models_\gamma \varphi_1 \vee \varphi_2$ *if and only if* $c \models_\gamma \varphi_1$ *or* $c \models_\gamma \varphi_2$.
– $c \models_\gamma \langle a\rangle\varphi$ *if and only if* $c' \models_\gamma \varphi$*, for some* $c' \in \gamma(c)(a)_\Box$.
– $c \models_\gamma [a]\varphi$ *if and only* $c' \models_\gamma \varphi$*, for all* $c' \in \gamma(c)(a)_\diamond$.

Proof (Sketch). Again, the proof is essentially the same of that as Proposition 3. The main difference is that we have to consider the final sequence of $T = \mathcal{P}_\omega(id \times \{\diamond, \Box\})^A$ instead of that of the functor $\mathcal{P}_\omega^A$, that is, we consider A-trees with two kinds of transitions. For the remaining of the proof we have just to consider the specific definitions of $\mathbb{S}_{ref}$ and d_ω in order to get the desired result. □

Hence, Proposition 9 shows that the logic induced by $\mathbb{S}_{ref}$ and Γ_{ref} is equivalent to the logic for modal refinements between modal transition systems as defined in [3].

3.5 Mixed transition systems

Mixed transition systems [11, 6] generalize MTS by considering two kinds of transitions that need not be related at all.

Definition 20 ([6]). *For a set of actions A, a* mixed transition system *(MiTS) is a triple* $(P, \rightarrow_1, \rightarrow_2)$*, where P is a set of states and* $\rightarrow_1, \rightarrow_2 \subseteq P \times A \times P$ *are transition relations.*

As for the associated simulation notion, it requires one transition relation to behave covariantly and the other one contravariantly.

Definition 21 ([6]). *A relation* $R \subseteq P \times Q$ *is a* mixed simulation *between two MiTS if, whenever* $p\ R\ q$*:*

- $p \xrightarrow{a}_1 p'$ *implies that there exists some* q' *such that* $q \xrightarrow{a}_1 q'$ *and* $p'\ R\ q'$*;*
- $q \xrightarrow{a}_2 q'$ *implies that there exists some* p' *such that* $p \xrightarrow{a}_2 p'$ *and* $p'\ R\ q'$*.*

Thus, MTS are obtained as the particular case in which $\rightarrow_1 \subseteq \rightarrow_2$. Other than that, MiTS behave as MTS and can be described in similar coalgebraic terms. An MiTS arises as a coalgebra for the functor $F = \mathcal{P}(id \times \{1, 2\})^A$, where 1 stands for $\rightarrow_1$ transitions and 2 for $\rightarrow_2$ transitions; given $c : X \longrightarrow \mathcal{P}_\omega(X \times \{1, 2\})^A$, we shall use the following notation:

$c(x)(a)_1 = \{x' \in X \mid (x', 1) \in c(x)(a)\}$, and
$c(x)(a)_2 = \{x' \in X \mid (x', 2) \in c(x)(a)\}$.

Then, the definition of the relator that captures MiTS simulations is straightforward, by mimicking that for MTS.

Definition 22 (Mixed relator). *Given* $R \subseteq Q \times P$*,* $g : Q \longrightarrow \mathcal{P}_\omega(Q \times \{1, 2\})^A$ *and* $f : P \longrightarrow \mathcal{P}_\omega(P \times \{1, 2\})^A$*, we define the* $\mathcal{P}_\omega(id \times \{1, 2\})^A$*-relator* $\Gamma_{\text{mix}} : \mathbf{Rel} \longrightarrow \mathbf{Rel}$ *for mixed simulation by* $g\ \Gamma_{\text{mix}}(R)\ f$ *if and only if:*

- *for all* $a \in A$*, if* $p' \in f(a)_1$ *then there is* $q' \in g(a)_1$ *such that* $q'Rp'$*;*
- *for all* $a \in A$*, if* $q' \in g(a)_2$ *then there is* $p' \in f(a)_2$ *such that* $q'Rp'$*.*

From here, the same steps taken for building a logic that characterizes MTS can be retraced. For example, the functor for the $\mathcal{P}_\omega(id \times \{1, 2\})^A$-semantics maps an interpretation d to d' as on page 10, just replacing $f(a)_\diamond$ and $f(a)_\Box$ with $f(a)_1$ and $f(a)_2$. This way, the resulting logic for MiTS is:

Proposition 10. *For an MiTS* $\gamma : C \longrightarrow \mathcal{P}_\omega(C \times \{1, 2\})^A$*, the logic which characterizes mixed simulation is given by:*

- $c \models_\gamma \mathsf{tt}$.
- $c \models_\gamma \varphi_1 \wedge \varphi_2$ *if and only if* $c \models_\gamma \varphi_1$ *and* $c \models_\gamma \varphi_2$*.*
- $c \models_\gamma \varphi_1 \vee \varphi_2$ *if and only if* $c \models_\gamma \varphi_1$ *or* $c \models_\gamma \varphi_2$*.*
- $c \models_\gamma \langle a \rangle \varphi$ *if and only if* $c' \models_\gamma \varphi$*, for some* $c' \in \gamma(c)(a)_1$*.*
- $c \models_\gamma [a]\varphi$ *if and only* $c' \models_\gamma \varphi$*, for all* $c' \in \gamma(c)(a)_2$*.*

As a consequence, Proposition 9 turns out to be a corollary of this result.

4 Conclusion and future work

Following [5], we have built the characterizing logics for covariant-contravariant and conformance simulations, partial bisimulation (which can be considered as a particular case of the covariant-contravariant notion), modal refinement and mixed transition systems. In particular, we have presented a novel (to the best of our knowledge) coalgebraic characterization of modal and mixed transition systems. Even though most of the results are not new (except for the logical characterization of mixed transition systems), we believe that their proofs constitute a nice illustration of the method developed in [5], with non-trivial systems.

As future work, we intend to explore the relationship between covariant-contravariant simulation and modal refinement at the institution level that we sketched in [1]. Our idea would be to check whether the machinery of borrowing [4, 12] could be used to express our results in [1] relating the logics for covariant-contravariant simulation and modal transition systems in a more precise manner at the categorical level.

References

1. L. Aceto, I. Fábregas, D. de Frutos Escrig, A. Ingólfsdóttir, and M. Palomino. Relating modal refinements, covariant-contravariant simulations and partial bisimulations. In F. Arbab and M. Sirjani, editors, *Fundamentals of Software Engineering, FSEN 2011*, Lecture Notes in Computer Science. Springer, 2011.
2. J. Baeten, D. van Beek, B. Luttik, J. Markovski, and J. Rooda. Partial bisimulation. SE Report 2010-04, Systems Engineering Group, Department of Mechanical Engineering, Eindhoven University of Technology, 2010.
3. G. Boudol and K. G. Larsen. Graphical versus logical specifications. *Theor. Comput. Sci.*, 106(1):3–20, 1992.
4. M. Cerioli and J. Meseguer. May I borrow your logic? (Transporting logical structures along maps). *Theor. Comput. Sci.*, 173(2):311–347, 1997.
5. C. Cîrstea. A modular approach to defining and characterising notions of simulation. *Inf. Comput.*, 204(4):469–502, 2006.
6. D. Dams, R. Gerth, and O. Grumberg. Abstract interpretation of reactive systems. *ACM Transactions on Programming Languages and Systems*, 19:253–291, 1997.
7. I. Fábregas, D. de Frutos-Escrig, and M. Palomino. Non-strongly stable orders also define interesting simulation relations. In *Proceedings of CALCO 2009*, volume 5728 of *Lecture Notes in Computer Science*, pages 221–235. Springer, 2009.
8. I. Fábregas, D. de Frutos-Escrig, and M. Palomino. Logics for contravariant simulations. In *Proceedings of FORTE 2010*, volume 6117 of *Lecture Notes in Computer Science*, pages 224–231. Springer-Verlag, 2010.
9. I. Fábregas, M. Palomino, and D. de Frutos-Escrig. Categorical logics for contravariant simulations, partial bisimulations, modal refinements and mixed transition systems (extended). `http://maude.sip.ucm.es/~miguelpt/bibliography.html`
10. M. Hennessy and R. Milner. Algebraic laws for nondeterminism and concurrency. *J. ACM*, 32(1):137–161, 1985.
11. K. G. Larsen and L. Xinxin. Equation solving using modal transition systems. In *LICS*, pages 108–117. IEEE Computer Society, 1990.
12. T. Mossakowski. Relating CASL with other specification languages: the institution level. *Theor. Comput. Sci.*, 286(2):367–475, 2002.

13. P. J. Ramadge and W. M. Wonham. Supervisory control of a class of discrete event processes. *SIAM J. Control Optim.*, 25(1):206–230, 1987.
14. A. Thijs. *Simulation and fixpoint semantics*. PhD thesis, Rijksuniversiteit Groningen, 1996.