

A Modular Approach to Defining and Characterising Notions of Simulation

Corina Cîrstea

*School of Electronics and Computer Science,
University of Southampton
Southampton SO17 1BJ, United Kingdom
Email: cc2@ecs.soton.ac.uk
URL: <http://www.ecs.soton.ac.uk/~cc2>*

Abstract

We propose a modular approach to defining notions of simulation, and modal logics which characterise them. We use *coalgebras* to model state-based systems, *relators* to define notions of simulation for such systems, and inductive techniques to define the syntax and semantics of modal logics for coalgebras. We show that the expressiveness of an inductively-defined logic for coalgebras w.r.t. a notion of simulation follows from an expressivity condition involving one step in the definition of the logic, and the relator inducing that notion of simulation. Moreover, we show that notions of simulation and associated characterising logics for increasingly complex system types can be derived by lifting the operations used to combine system types, to a relational level as well as to a logical level. We use these results to obtain Baltag's logic for coalgebraic simulation, as well as notions of simulation and associated logics for a large class of non-deterministic and probabilistic systems.

Key words: coalgebra, simulation, modal logic, probabilistic system

1 Introduction

Simulations have long been used in the semantics of computational systems, to formalise refinement relationships between such systems. The choice of a notion of simulation depends on which aspects of the system behaviour are of interest in a particular context. For example, in the semantics of non-deterministic, sequential processes, notions such as *standard*, *complete* or *ready simulation* are used to compare the ability/inability of processes to perform certain/any computations (see [29] for an overview). Similarly, in the semantics of probabilistic processes, various notions of probabilistic simulation are

used to compare the probabilities with which processes are able to ensure certain outcomes as a result of computations (see [9,18]). In each of these cases, logical characterisations of simulation, using various modal logics, have also been proposed (again, see [29,9,18]). Such logics offer useful insights into the kinds of properties which are preserved by particular notions of simulation, and at the same time provide a basis for more expressive temporal logics used in formal verification [24,10,19,12].

What is missing from existing approaches are compositional techniques for deriving notions of simulation, and logics which characterise them, for a large class of system types. The system types of interest are often combinations of simple system types – this is, for instance, the case for the *probabilistic automata* of [27,26] (see also [18]), or for the *alternating probabilistic systems* of [11], where a combination of non-deterministic and probabilistic features is present. In such cases, the task of defining suitable notions of simulation, and logics which characterise them, becomes increasingly challenging. A framework which allows the automatic derivation of such definitions, together with proofs of expressiveness, would therefore prove valuable in the treatment of complex system types. The present paper develops such a framework, using *coalgebras* as a general setting.

Coalgebras have, in recent years, been shown to provide suitable abstract models for a large class of state-based systems, which includes non-deterministic systems, probabilistic systems, and various kinds of automata [25]. The emphasis in such modelling is on the observations which can be performed on a system in one step. The coalgebraic notion of *bisimulation* then formalises the indistinguishability of system states by experiments involving a series of one-step observations. Notions of *simulation* between systems modelled as coalgebras have also been studied, initially as a tool for proving refinements between recursively-defined programs [28,14], and later with the aim to provide a coalgebraic theory of simulations [1,17]. Logics which characterise simulation have been proposed in [1], generalising earlier work on logics which characterise bisimulation [22]. Additional properties of coalgebraic simulations, including their relationship to *final coalgebras*, were subsequently studied in [17].

This paper further develops the coalgebraic theory of simulations, by providing a modular approach to defining notions of simulation and modal logics which characterise them. This approach lifts the operations used to combine coalgebraic types to a relational level as well as to a logical level, thereby allowing notions of simulation and characterising logics for combinations of coalgebraic types to be derived from notions of simulation and characterising logics for the types being combined. The structure of coalgebraic types is thus reflected both in the notions of simulation defined, and in the modal operators employed by the corresponding logics.

A similar approach to defining logics for coalgebras was taken in [5] (see also [4]), where logics capturing bisimulation were investigated from a compositional perspective. Specifically, it was shown in loc. cit. that the expressiveness w.r.t. bisimulation of an inductively-defined logic for coalgebras follows from an expressivity condition involving one step in the definition of the logic. In the case of logics capturing simulation, the situation is more complex, since, unlike bisimulation, simulation is not uniquely determined by the underlying coalgebraic type. Thus, in order to derive characterising logics for simulation, these logics must be tailored to particular notions of simulation, and the expressivity condition of [5] has to be generalised accordingly. Also, a method for deriving notions of simulation for combinations of coalgebraic types from notions of simulation for the types being combined needs to be developed.

The structure of the paper is as follows.

Section 2 contains some prerequisites for subsequent sections, including basic facts about relations, coalgebras, and coalgebraic simulation.

Section 3 provides an alternative characterisation of *monotonic relators* (the concept underlying the definition of coalgebraic simulation [28]), and uses it to obtain a coalgebraic characterisation of simulation on unlabelled probabilistic transition systems.

Section 4 develops a modular approach for defining simulation relations between coalgebras. Various operations on coalgebraic types, including functor composition, product, coproduct and exponentiation are shown to induce corresponding operations on monotonic relators, thereby yielding notions of simulation for increasingly complex coalgebraic types. The notions of standard/ready simulation on labelled transition systems [29], simulation on probabilistic transition systems [9], and strong simulation on probabilistic automata [27] are all shown to arise in this way.

Section 5 discusses an inductive method for defining logics which characterise simulation, much in the spirit of [5]. *Language constructors* and associated semantics are used to capture one step in the definition of a logic for coalgebras. Moreover, an expressivity condition involving (a) a language constructor and an associated semantics, and (b) a monotonic relator, is used to ensure the expressiveness of the induced logic w.r.t. the induced notion of simulation. This method can be applied to derive Baltag's logic for coalgebraic simulation [1], as well as a logic capturing simulation on unlabelled probabilistic transition systems.

Section 6 develops a modular approach to defining expressive logics for simulation. The previously-considered operations on coalgebraic types are shown to induce corresponding operations on language constructors and their associated semantics, with the above-mentioned expressivity condition being

preserved by these operations. This allows the modular derivation of logics which characterise standard/ready simulation on labelled transition systems, simulation on probabilistic transition systems, and strong (probabilistic) simulation on probabilistic automata. The resulting logics are similar (as regards their syntax, semantics and expressiveness) to the logics known to characterise these notions of simulation (as described e.g. in [29,9,18]).

Section 7 concludes with a summary of the results obtained.

This paper is an extended version of [6]. It differs from [6] in the treatment of language constructors and their associated semantics: these were regarded as a single concept in loc. cit., but are here separated to allow for a clearer distinction between syntax and semantics. (This alternative formulation was first proposed in [7].) Compared to [6], the present paper also provides additional examples, including a more comprehensive treatment of notions of simulation for labelled transition systems, and a treatment of notions of simulation for probabilistic automata.

2 Preliminaries

Here we fix the notation for subsequent sections, recall some basic facts about relations and coalgebras, and summarise the coalgebraic approach to defining simulation.

2.1 Relations

We let \mathbf{Set} denote the category of sets and functions, and write $X_1 \times X_2$ ($X_1 + X_2$) for the cartesian product (disjoint union) of X_1 and X_2 , and $\pi_i : X_1 \times X_2 \rightarrow X_i$ ($\iota_i : X_i \rightarrow X_1 + X_2$), with $i = 1, 2$, for the canonical projections (injections). We also let $1 = \{*\}$ denote a one-element set (final object in \mathbf{Set}). Finally, we write X^A for the set of functions $A \rightarrow X$.

We let \mathbf{Rel} denote the category having, as objects, binary relations $R \subseteq A \times B$, and as arrows from $R \subseteq A \times B$ to $S \subseteq C \times D$, pairs of functions (f, g) with $f : A \rightarrow C$ and $g : B \rightarrow D$ being such that $(f \times g)(R) \subseteq S$. We note that this is not the only way of defining a category of relations between sets. Another possibility is to consider the category having sets as objects, and relations $R \subseteq A \times B$ as arrows from A to B . This is, for instance, the approach taken in [1]. Our definition of \mathbf{Rel} follows [17].

Given a relation $R \subseteq A \times B$, we write π_1^R and π_2^R for $\pi_1 \circ \iota : R \rightarrow A$ and $\pi_2 \circ \iota : R \rightarrow B$, respectively, where $\iota : R \hookrightarrow A \times B$ is the inclusion map. Also,

we write R^{op} for the converse of a relation R , and $\text{Gr}f \subseteq A \times B$ for the relation defining the graph of a function $f : A \rightarrow B$. The composition of two relations $R \subseteq A \times B$ and $S \subseteq B \times C$ is denoted $S \circ R \subseteq A \times C$.

If $\mathbf{U} : \mathbf{Rel} \rightarrow \mathbf{Set} \times \mathbf{Set}$ denotes the functor taking relations to their underlying sets, then \mathbf{U} defines a *fibration* (see [3,15] for a definition of fibrations). For, given functions $f : A \rightarrow C$, $g : B \rightarrow D$ and a relation $S \subseteq C \times D$, letting $a R b$ iff $f(a) S g(b)$ makes $(f, g) : (R \subseteq A \times B) \rightarrow (S \subseteq C \times D)$ a cartesian map. (Equivalently, R can be defined as $\text{Gr}(g)^{\text{op}} \circ S \circ \text{Gr}(f)$.) The cartesian maps of \mathbf{U} are thus the relation-reflecting maps. We also note that our particular definition of \mathbf{Rel} results in the uniqueness of cartesian maps over $(f, g) : A \times B \rightarrow C \times D$ with codomain $S \subseteq C \times D$, for any such (f, g) and S . Consequently, there is only one *reindexing functor* (see [15] for a definition), denoted $(f, g)^*$, for each $(f, g) : A \times B \rightarrow C \times D$, and only one *cleavage* (again, see [15] for a definition) for the fibration \mathbf{U} .

We now let \mathbf{Preord} denote the category of preorders and monotonic maps. Then, \mathbf{Preord} is (isomorphic to) a sub-category of \mathbf{Rel} . Moreover, if $\mathbf{V} : \mathbf{Preord} \rightarrow \mathbf{Set}$ takes preorders to their underlying sets, then \mathbf{V} also defines a fibration. The cartesian maps of \mathbf{V} are the (pre)order-reflecting maps. The following also holds:

Proposition 1 *Rel and Preord are complete categories.*

Limits in \mathbf{Rel} and \mathbf{Preord} are constructed from limits in \mathbf{Set} and limits in certain fibres of \mathbf{U} and \mathbf{V} , respectively.

2.2 Coalgebras

The coalgebraic approach to modelling systems involves the use of an endofunctor to specify the type of information which can be observed about a system in one step. Particular models of the system are then formalised as *coalgebras*.

Definition 2 (Coalgebra, coalgebra morphism) *Let $\mathbf{T} : \mathbf{C} \rightarrow \mathbf{C}$ be an endofunctor. A \mathbf{T} -coalgebra is a pair (C, γ) with C a \mathbf{C} -object (the carrier of the coalgebra) and $\gamma : C \rightarrow \mathbf{T}C$ a \mathbf{C} -arrow (the coalgebra map). Also, a \mathbf{T} -coalgebra morphism from (C, γ) to (D, δ) is a \mathbf{C} -arrow $f : C \rightarrow D$ such that $\mathbf{T}f \circ \gamma = \delta \circ f$. The category of \mathbf{T} -coalgebras and \mathbf{T} -coalgebra morphisms is denoted $\mathbf{Coalg}(\mathbf{T})$.*

In what follows, we will consider (coalgebras of) endofunctors on the categories \mathbf{Set} , \mathbf{Rel} and \mathbf{Preord} . Coalgebras over \mathbf{Set} will be used to model state-based systems, whereas coalgebras over \mathbf{Rel} and \mathbf{Preord} will be used to define simulation

relations between systems modelled in this way.

For convenience, we will restrict attention to *standard* (that is, inclusion preserving) endofunctors on **Set**. A similar assumption is made in [28] when defining simulation relations. The results in this paper could also be formulated for arbitrary endofunctors on **Set**, but this would involve defining relations as monomorphic spans, which, in turn, would complicate our exposition. The assumption regarding standard endofunctors will be implicit in all the endofunctors $\mathbb{T} : \mathbf{Set} \rightarrow \mathbf{Set}$ considered in this paper, and true of all concrete such endofunctors.

Example 3 *Let A be a set, and let $\mathcal{P}_\omega : \mathbf{Set} \rightarrow \mathbf{Set}$ denote the finite powerset functor, taking a set to the set of its finite subsets. Then, any $(\mathcal{P}_\omega)^A$ -coalgebra (S, γ) defines an (image-finite) A -labelled transition system, with set of states S and transition relation given by $s \xrightarrow{a} t$ iff $t \in \gamma(s)(a)$, for $s, t \in S$ and $a \in A$. Moreover, any image-finite, A -labelled transition system can be modelled in this way.*

Example 4 *Image-finite, A -labelled probabilistic transition systems¹ can be modelled as coalgebras of the functor $(1 + \mathcal{D}_\omega)^A : \mathbf{Set} \rightarrow \mathbf{Set}$, where $\mathcal{D}_\omega : \mathbf{Set} \rightarrow \mathbf{Set}$ is the finite probability distribution functor, defined by:*

$$\mathcal{D}_\omega X = \{ \mu : X \rightarrow [0, 1] \mid \text{supp}(\mu) \text{ finite}, \sum_{x \in X} \mu(x) = 1 \} \text{ for } X \in |\mathbf{Set}|$$

with $\text{supp}(\mu) = \{ x \in X \mid \mu(x) \neq 0 \}$, for $\mu : X \rightarrow [0, 1]$, and:

$$(\mathcal{D}_\omega f)(\mu)(y) = \mu[f^{-1}(\{y\})] \text{ for } f : X \rightarrow Y, \mu \in \mathcal{D}_\omega X, \text{ and } y \in Y$$

with $\mu[Z] = \sum_{x \in Z} \mu(x)$, for $\mu : X \rightarrow [0, 1]$ and $Z \subseteq X$.

The coalgebraic approach to modelling systems provides a canonical notion of observational equivalence between system states, in the form of *bisimilarity*.

Definition 5 (Bisimulation, bisimilarity) *Let $\mathbb{T} : \mathbf{Set} \rightarrow \mathbf{Set}$ be an endofunctor. A \mathbb{T} -bisimulation between \mathbb{T} -coalgebras (C, γ) and (D, δ) is a relation $R \subseteq C \times D$ carrying a \mathbb{T} -coalgebra structure $\rho : R \rightarrow \mathbb{T}R$ which makes $\pi_1^R : R \rightarrow C$ and $\pi_2^R : R \rightarrow D$ \mathbb{T} -coalgebra morphisms. The largest \mathbb{T} -bisimulation between (C, γ) and (D, δ) , given by the union of all bisimulations between (C, γ) and (D, δ) , is called \mathbb{T} -bisimilarity and is denoted \simeq .*

Example 6 *$(\mathcal{P}_\omega)^A$ -bisimulation coincides with Park-Milner bisimulation, as defined in [23, 21].*

¹ These are similar to transition systems, except that transitions also carry probability values $p \in [0, 1]$, with $\sum_{s \xrightarrow{a,p} t} p \in \{0, 1\}$ for each $s \in S$ and $a \in A$.

Example 7 A notion of bisimulation equivalence for probabilistic transition systems was defined in [20]. Moreover, it was shown in [8] that this notion is essentially the same as $(1 + \mathcal{D}_\omega)^A$ -bisimulation. The following characterisation of $1 + \mathcal{D}_\omega$ -bisimulation was also given in [8]: a relation $R \subseteq C \times D$ is a $1 + \mathcal{D}_\omega$ -bisimulation between (C, γ) and (D, δ) iff $c R d$ implies $\gamma(c)[X] = \delta(d)[Y]$ ² for any $X \subseteq C$ and $Y \subseteq D$ such that $(\pi_1^R)^{-1}(X) = (\pi_2^R)^{-1}(Y)$.

Of particular interest in the coalgebraic modelling of systems are *final T-coalgebras*, that is, final objects in the category $\mathbf{Coalg}(\mathbb{T})$. The elements of a final T-coalgebra provide abstract descriptions of all observable behaviours w.r.t. T. A general method for constructing the final coalgebra of an endofunctor is via its *final sequence*.

Definition 8 (Final sequence) For an endofunctor $\mathbb{T} : \mathbf{C} \rightarrow \mathbf{C}$ on a complete category, the final sequence of \mathbb{T} is an ordinal-indexed sequence (Z_α) of C-objects, together with a family $(p_\beta^\alpha : Z_\alpha \rightarrow Z_\beta)_{\beta \leq \alpha}$ of C-arrows, subject to the following conditions:

- $Z_0 = 1$,
- $p_0^\alpha : Z_\alpha \rightarrow 1$ is the unique such map,
- $Z_{\alpha+1} = \mathbb{T}Z_\alpha$,
- $p_{\beta+1}^{\alpha+1} = \mathbb{T}p_\beta^\alpha$ for $\beta \leq \alpha$,
- $p_\alpha^\alpha = 1_{Z_\alpha}$,
- $p_\gamma^\alpha = p_\gamma^\beta \circ p_\beta^\alpha$ for $\gamma \leq \beta \leq \alpha$,
- if α is a limit ordinal, the cone $Z_\alpha, (p_\beta^\alpha)_{\beta < \alpha}$ for $(p_\gamma^\beta)_{\gamma \leq \beta < \alpha}$ is limiting.

The final sequence of \mathbb{T} is uniquely defined by these conditions.

For an ordinal α , the α -element of the final sequence of \mathbb{T} describes the abstract T-behaviours observable in α steps. Elements of arbitrary T-coalgebras can be mapped to such partial observable behaviours by using α unfoldings of the coalgebra structure, as illustrated next.

Remark 9 Given a T-coalgebra (C, γ) , one can define a cone $(\gamma_\alpha : C \rightarrow Z_\alpha)$ over the final sequence of \mathbb{T} as follows:

- $\gamma_0 : C \rightarrow 1$ is the unique such map;
- $\gamma_\alpha = \mathbb{T}\gamma_\beta \circ \gamma$, if $\alpha = \beta + 1$;
- γ_α is the unique C-arrow satisfying $p_\beta^\alpha \circ \gamma_\alpha = \gamma_\beta$ for each $\beta < \alpha$, if α is a limit ordinal.

Moreover, T-coalgebra morphisms $f : (C, \gamma) \rightarrow (D, \delta)$ define morphisms of cones $f : (\gamma_\alpha : C \rightarrow Z_\alpha) \rightarrow (\delta_\alpha : D \rightarrow Z_\alpha)$; that is, $\delta_\alpha \circ f = \gamma_\alpha$ for any α .

² By convention, $\gamma(c)[X] = 0$ if $\gamma(c) \in \iota_1(1)$.

Under some mild constraints on \mathbf{C} and \mathbf{T} , the final sequence of \mathbf{T} stabilises, yielding a final \mathbf{T} -coalgebra.

Proposition 10 ([31]) *If $\mathbf{T} : \mathbf{C} \rightarrow \mathbf{C}$ is an accessible endofunctor³ on a locally presentable category⁴, and if \mathbf{T} preserves monics, then the final sequence of \mathbf{T} stabilises at some α (that is, $p_\alpha^{\alpha+1} : Z_{\alpha+1} \rightarrow Z_\alpha$ is an isomorphism), and moreover, Z_α is the carrier of a final \mathbf{T} -coalgebra.*

In the case of ω -accessible endofunctors on \mathbf{Set} , the cardinal α of Proposition 10 does not exceed $\omega + \omega$.

Proposition 11 ([31]) *If $\mathbf{T} : \mathbf{Set} \rightarrow \mathbf{Set}$ is ω -accessible, the map $p_{\omega+\omega}^{\omega+\omega+1} : Z_{\omega+\omega+1} \rightarrow Z_{\omega+\omega}$ is an isomorphism, whereas the maps $p_{\omega+n}^{\omega+n+1} : Z_{\omega+n+1} \rightarrow Z_{\omega+n}$ with $n = 0, 1, \dots$ are all injective.*

2.3 Simulations

Notions of simulation between coalgebras have been studied in [28,14,1,17]. A summary of these approaches is given in the following. To this end, we fix an endofunctor $\mathbf{T} : \mathbf{Set} \rightarrow \mathbf{Set}$.

The concept which lies at the heart of defining coalgebraic simulation is that of a relator. A *(\mathbf{T} -)relator* [28] is a mapping from relations to relations, taking relations on $A \times B$ to relations on $\mathbf{T}A \times \mathbf{T}B$. A *monotonic (\mathbf{T} -)relator* [28] is required to satisfy some additional constraints, including preservation of inclusions between relations with the same carrier, and preservation of relational composition. These constraints result in the following equivalent definition of monotonic relators:

Definition 12 (Monotonic relator) *Let $\mathbf{T} : \mathbf{Set} \rightarrow \mathbf{Set}$. A monotonic \mathbf{T} -relator is an endofunctor $\Gamma : \mathbf{Rel} \rightarrow \mathbf{Rel}$ additionally satisfying:*

- (i) $\mathbf{U} \circ \Gamma = (\mathbf{T} \times \mathbf{T}) \circ \mathbf{U}$;
- (ii) $=_{\mathbf{T}A} \subseteq \Gamma(=A)$;
- (iii) $\Gamma(S \circ R) = \Gamma(S) \circ \Gamma(R)$ for any $R \subseteq A \times B$ and $S \subseteq B \times C$.

For any \mathbf{T} -relator $\Gamma : \mathbf{Rel} \rightarrow \mathbf{Rel}$, the *transposed relator* $\Gamma^\sim : \mathbf{Rel} \rightarrow \mathbf{Rel}$ takes a relation $R \subseteq A \times B$ to the relation $(\Gamma(R^{\text{op}}))^{\text{op}} \subseteq \mathbf{T}A \times \mathbf{T}B$.

Any monotonic \mathbf{T} -relator induces a notion of simulation between \mathbf{T} -coalgebras. The following is a reformulation of the definition of simulation given in [28]

³ Given a regular cardinal κ , an endofunctor is κ -accessible if it preserves κ -filtered colimits.

⁴ Each of the categories \mathbf{Set} , \mathbf{Rel} and \mathbf{Preord} are locally ω -presentable.

(see also [17]).

Definition 13 (Simulation, similarity) *Let $\Gamma : \mathbf{Rel} \rightarrow \mathbf{Rel}$ be a monotonic \mathbb{T} -relator. A Γ -simulation between \mathbb{T} -coalgebras (C, γ) and (D, δ) is a Γ -coalgebra of the form $(R, (\gamma, \delta))$. The largest Γ -simulation between (C, γ) and (D, δ) is called Γ -similarity and is denoted \gtrsim . If $c \in C$, $d \in D$ are such that $c \gtrsim d$, we say that c simulates d .*

A Γ -simulation between (C, γ) and (D, δ) is thus given by a relation $R \subseteq C \times D$ such that $c R d$ implies $\gamma(c) \Gamma(R) \delta(d)$ for any $c \in C$ and $d \in D$.

A generic example of a relator is the *minimal relator induced by \mathbb{T}* [28], denoted $\Gamma_{\mathbb{T}} : \mathbf{Rel} \rightarrow \mathbf{Rel}$, and defined by:

$$\Gamma_{\mathbb{T}}(R) = \langle \mathbb{T}\pi_1^R, \mathbb{T}\pi_2^R \rangle(\mathbb{T}R) \subseteq \mathbb{T}A \times \mathbb{T}B \text{ for } R \subseteq A \times B$$

We note in passing that the minimal relator induced by an endofunctor corresponds to the notion of *relation lifting* of an endofunctor [13,16], used in defining coalgebraic bisimulation for so-called *polynomial* endofunctors.

The minimal relator induced by \mathbb{T} is monotonic if and only if \mathbb{T} preserves weak pullbacks. This observation is an immediate consequence of the results in [28, Section 2.2]. Irrespective of the preservation of weak pullbacks by \mathbb{T} , the minimal relator $\Gamma_{\mathbb{T}}$ is contained in any monotonic relator Γ ; that is, $\Gamma_{\mathbb{T}}(R) \subseteq \Gamma(R)$ for any relation R . Moreover, any monotonic relator Γ can be defined in terms of its action on equality relations and of $\Gamma_{\mathbb{T}}$:

$$\Gamma(R) = \Gamma(=_{\mathbb{B}}) \circ \Gamma_{\mathbb{T}}(R) \circ \Gamma(=_{\mathbb{A}}) \text{ for any } R \subseteq A \times B \quad (1)$$

(See e.g. [28, Theorem 2.1.4] for a proof.)

The notion of simulation induced by the minimal \mathbb{T} -relator coincides with \mathbb{T} -bisimulation.

Proposition 14 *Let $\mathbb{T} : \mathbf{Set} \rightarrow \mathbf{Set}$ be a weak pullback preserving endofunctor. Then, the $\Gamma_{\mathbb{T}}$ -simulations are exactly the \mathbb{T} -bisimulations.*

PROOF (Sketch). Let (C, γ) and (D, δ) be \mathbb{T} -coalgebras, and let $R \subseteq C \times D$. The existence of a \mathbb{T} -coalgebra $\rho : R \rightarrow \mathbb{T}R$ making the projections $\pi_1^R : R \rightarrow C$ and $\pi_2^R : R \rightarrow D$ \mathbb{T} -coalgebra morphisms is equivalent to $R \subseteq C \times D$ being such that $\gamma(c) \langle \mathbb{T}\pi_1^R, \mathbb{T}\pi_2^R \rangle(\mathbb{T}R) \delta(d)$ whenever $c R d$, for $c \in C$ and $d \in D$.

Also, since $\Gamma_{\mathbb{T}}(R) \subseteq \Gamma(R)$ for any relation R , any \mathbb{T} -bisimulation (or equivalently, $\Gamma_{\mathbb{T}}$ -simulation) is also a Γ -simulation, for any monotonic relator Γ .

Example 15 The minimal \mathcal{P}_ω -relator $\Gamma_{\mathcal{P}_\omega} : \text{Rel} \rightarrow \text{Rel}$ takes a relation $R \subseteq A \times B$ to the relation $\Gamma_{\mathcal{P}_\omega}(R) \subseteq \mathcal{P}_\omega A \times \mathcal{P}_\omega B$ defined by:

$$X \Gamma_{\mathcal{P}_\omega}(R) Y \quad \text{iff} \quad (\forall x \in X . \exists y \in Y . x R y \text{ and } \forall y \in Y . \exists x \in X . x R y)$$

for $X \in \mathcal{P}_\omega A$ and $Y \in \mathcal{P}_\omega B$. Another \mathcal{P}_ω -relator $\Gamma_{\supseteq} : \text{Rel} \rightarrow \text{Rel}$ can be defined by:

$$X \Gamma_{\supseteq}(R) Y \quad \text{iff} \quad \forall y \in Y . \exists x \in X . x R y$$

Both $\Gamma_{\mathcal{P}_\omega}$ and Γ_{\supseteq} are monotonic relators. Moreover, $\Gamma_{\supseteq}(R) = \supseteq_B \circ \Gamma_{\mathcal{P}_\omega}(R) \circ \supseteq_A$, where \supseteq_A and \supseteq_B are the containment relations on $\mathcal{P}_\omega A$ and $\mathcal{P}_\omega B$, respectively. Finally, the transposed relator $\Gamma_{\subseteq} = (\Gamma_{\supseteq})^\sim$ is given by:

$$X \Gamma_{\subseteq}(R) Y \quad \text{iff} \quad \forall x \in X . \exists y \in Y . x R y$$

By Proposition 14, $\Gamma_{\mathcal{P}_\omega}$ -simulations are the same as \mathcal{P}_ω -bisimulations. Also, a relation $R \subseteq C \times D$ is a Γ_{\supseteq} -simulation between \mathcal{P}_ω -coalgebras (C, γ) and (D, δ) if, whenever $c R d$ and $d' \in \delta(d)$, there exists $c' \in \gamma(c)$ such that $c' R d'$.

It is shown in [28] that monotonic relators are in one-to-one correspondence with so-called *monotonic extensions* of \mathbb{T} .

Definition 16 (Extension) Let $\mathbb{T} : \text{Set} \rightarrow \text{Set}$. An extension of \mathbb{T} is a functor $\supseteq : \text{Set} \rightarrow \text{Preord}$ such that:

- (i) $\mathbb{V} \circ \supseteq = \mathbb{T}$;
- (ii) if $A \subseteq B$ then $u \supseteq_A v$ iff $u \supseteq_B v$, for any $u, v \in \mathbb{T}A$ ⁵.

Any extension of \mathbb{T} induces a \mathbb{T} -relator $\Gamma_{\supseteq} : \text{Rel} \rightarrow \text{Rel}$, defined by:

$$\Gamma_{\supseteq}(R) = \supseteq_B \circ \Gamma_{\mathbb{T}}(R) \circ \supseteq_A \quad \text{for } R \subseteq A \times B$$

(Functoriality of Γ_{\supseteq} follows from the functoriality of \supseteq and $\Gamma_{\mathbb{T}}$.)

Definition 17 (Monotonic extension) An extension \supseteq of \mathbb{T} is monotonic if the following holds for any $f : A \rightarrow C$, $g : B \rightarrow C$, $u \in \mathbb{T}A$ and $v \in \mathbb{T}B$:

$$(\mathbb{T}f)(u) \supseteq_C (\mathbb{T}g)(v) \Rightarrow u (\Gamma_{\supseteq} \{ (a, b) \in A \times B \mid f(a) = g(b) \}) v$$

Remark 18 By the functoriality of Γ_{\supseteq} , the converse implication in Definition 17 always holds.

The above definition ensures that monotonic extensions induce monotonic relators, and moreover, that any monotonic relator Γ arises from a unique monotonic extension \supseteq_Γ , defined by:

$$\supseteq_{\Gamma, A} = \Gamma(=_A) \quad \text{for } A \in |\text{Set}| \tag{2}$$

⁵ The assumption that \mathbb{T} is standard gives $\mathbb{T}A \subseteq \mathbb{T}B$ whenever $A \subseteq B$.

Finally, we note that any monotonic relator Γ restricts to an endofunctor on \mathbf{Preord} (itself denoted Γ). (See [28, Theorem 2.2.4] for a proof.)

Example 19 *The functor $\supseteq : \mathbf{Set} \rightarrow \mathbf{Preord}$ taking a set A to the containment relation \supseteq_A on $\mathcal{P}_\omega A$ defines a monotonic extension of \mathcal{P}_ω . The corresponding monotonic relator is Γ_{\supseteq} , as defined in Example 15.*

A notion of *weak monotonic relator* was defined in [1], based on ideas from [28]. This notion is similar to that of a monotonic relator, only in [1] a different category of relations, having sets as objects and relations as arrows, is considered. In this setting, the notion of relator does not depend on an endofunctor $\mathbb{T} : \mathbf{Set} \rightarrow \mathbf{Set}$. Instead, the fact that \mathbf{Set} is a sub-category of the above-mentioned category of relations can be used to define what it means for a weak monotonic relator to *extend* an endofunctor \mathbb{T} . A notion of *simulation* induced by a weak monotonic relator extending an endofunctor \mathbb{T} can then be defined for \mathbb{T} -coalgebras. This notion is essentially the same as that of Definition 13. However, the fact that different categories of relations are used in the two definitions makes it impossible to directly transfer results between the two approaches.

In [17], functors $\sqsubseteq : \mathbf{Set} \rightarrow \mathbf{Preord}$ satisfying $\mathbb{V} \circ \sqsubseteq = \mathbb{T}$ were taken as primitive, and *lax relation lifting functors* $\mathbf{Rel}_{\sqsubseteq}(\mathbb{T}) : \mathbf{Rel} \rightarrow \mathbf{Rel}$, defined similarly to the relators Γ_{\sqsubseteq} , were used to define notions of simulation. Only the first condition in Definition 16 was required of the functors $\sqsubseteq : \mathbf{Set} \rightarrow \mathbf{Preord}$. As a result, the induced lax relation lifting functors were not necessarily monotonic relators. However, after restricting attention to monotonic extensions, the notion of simulation defined in [17] was the same as that of [28]. Moreover, several properties of Γ -similarity were proved in [17], in the presence of this restriction.

Proposition 20 ([17]) *The following hold for a monotonic relator $\Gamma : \mathbf{Rel} \rightarrow \mathbf{Rel}$:*

- (i) Γ -similarity on a \mathbb{T} -coalgebra (C, γ) is a preorder on C ;
- (ii) given \mathbb{T} -coalgebra morphisms $f : (A, \alpha) \rightarrow (B, \beta)$ and $g : (C, \gamma) \rightarrow (D, \delta)$,
 $a \succsim c$ iff $f(a) \succsim g(c)$, for $a \in A$ and $c \in C$;
- (iii) Γ -similarity on the final \mathbb{T} -coalgebra is the final Γ -coalgebra.

Remark 21 *By taking f and g in (ii) of Proposition 20 to be the unique morphisms $!_\alpha : (A, \alpha) \rightarrow (Z, \zeta)$ and $!_\gamma : (C, \gamma) \rightarrow (Z, \zeta)$ into the final \mathbb{T} -coalgebra, we obtain that Γ -similarity between (A, α) and (C, γ) is the domain of the cartesian map $(!_\alpha, !_\gamma)$ induced by the Γ -similarity relation on the final \mathbb{T} -coalgebra. This observation, together with (iii) of Proposition 20, will later allow us to define logics which characterise Γ -similarity.*

We conclude this section by noting that any Γ -relator also induces a notion of simulation equivalence, defined as Γ -similarity in both directions.

Definition 22 (Simulation equivalence) *Let $\Gamma : \mathbf{Rel} \rightarrow \mathbf{Rel}$ be a monotonic \mathbb{T} -relator, and let (C, γ) and (D, δ) be \mathbb{T} -coalgebras. Two states $c \in C$ and $d \in D$ are Γ -simulation equivalent (written $c \sim_{\Gamma} d$) if $c \succeq_{\Gamma} d$ and $d \succeq_{\Gamma} c$.*

As already noted in [17], Γ -simulation equivalence is generally weaker than \mathbb{T} -bisimulation. To see this, it suffices to consider the \mathcal{P}_{ω} -relator Γ_{\supseteq} of Example 15. In this case, Γ_{\supseteq} -simulation equivalence is the standard two-way simulation relation on unlabelled transition systems, which is known to be weaker than bisimulation (see e.g. [29]).

3 Monotonic Relators Revisited

In this section, we first give an alternative characterisation of monotonic relators, and then use this characterisation to define a notion of simulation for probabilistic transition systems. The alternative characterisation has a more categorical flavour than the original definition (Definition 12), as it replaces the preservation of relational composition by the preservation of a property of arrows in \mathbf{Rel} .

Lemma 23 *Let $\Gamma : \mathbf{Rel} \rightarrow \mathbf{Rel}$ be a monotonic relator. Then, the following hold:*

- (i) $\Gamma \mathbf{Gr}(f) = \Xi_{\Gamma, C} \circ \mathbf{Gr}(\mathbb{T}f)$
- (ii) $\Gamma(\mathbf{Gr}(g)^{\text{op}}) = \mathbf{Gr}(\mathbb{T}g)^{\text{op}} \circ \Xi_{\Gamma, C}$

PROOF (Sketch). The statements follow by taking $g = 1_C$ and $f = 1_C$, respectively, in Definition 17 (see also Remark 18).

Proposition 24 *Let $\mathbb{T} : \mathbf{Set} \rightarrow \mathbf{Set}$, and let $\Gamma : \mathbf{Rel} \rightarrow \mathbf{Rel}$ be such that:*

- (i) $\mathbf{U} \circ \Gamma = (\mathbb{T} \times \mathbb{T}) \circ \mathbf{U}$;
- (ii) $=_{\mathbb{T}A} \subseteq \Gamma(=A)$.

Then, Γ is a monotonic relator if and only if Γ preserves cartesian maps.

PROOF. Any monotonic relator Γ is uniquely determined by its induced monotonic extension Ξ_{Γ} , defined by (2) of Section 2.3. It therefore suffices to prove that, in the presence of (i) and (ii) above, the condition in the definition of monotonic extensions (Definition 17) is equivalent to the preservation by Γ of cartesian maps.

We begin by noting that the previously-mentioned condition is equivalent to Γ preserving cartesian maps of the form $(f, g) : (R \subseteq A \times B) \rightarrow (=_{\mathcal{C}} \subseteq C \times C)$ (see also Remark 18). Thus, one half of the previously-mentioned equivalence follows immediately. To prove the other half, assume that Γ is a monotonic relator, and let $(f, g) : (R \subseteq A \times B) \rightarrow (S \subseteq C \times D)$ be a cartesian map. Thus, $R = \text{Gr}(g)^{\text{op}} \circ S \circ \text{Gr}(f)$. The fact that $(\mathbb{T}f, \mathbb{T}g) : (\Gamma R \subseteq \mathbb{T}A \times \mathbb{T}B) \rightarrow (\Gamma S \subseteq \mathbb{T}C \times \mathbb{T}D)$ is itself a cartesian map, i.e. $\Gamma R = \text{Gr}(\mathbb{T}g)^{\text{op}} \circ \Gamma S \circ \text{Gr}(\mathbb{T}f)$, now follows from:

$$\begin{aligned}
\Gamma R &= \\
\Gamma(\text{Gr}(g)^{\text{op}}) \circ \Gamma S \circ \Gamma(\text{Gr}(f)) &= \quad (\text{Lemma 23}) \\
\text{Gr}(\mathbb{T}g)^{\text{op}} \circ \sqsubseteq_{\Gamma, D} \circ \Gamma S \circ \sqsubseteq_{\Gamma, C} \circ \text{Gr}(\mathbb{T}f) &= \quad (1) \\
\text{Gr}(\mathbb{T}g)^{\text{op}} \circ \sqsubseteq_{\Gamma, D} \circ \sqsubseteq_{\Gamma, D} \circ \Gamma_{\mathbb{T}} S \circ \sqsubseteq_{\Gamma, C} \circ \sqsubseteq_{\Gamma, C} \circ \text{Gr}(\mathbb{T}f) &= \\
\text{Gr}(\mathbb{T}g)^{\text{op}} \circ \sqsubseteq_{\Gamma, D} \circ \Gamma_{\mathbb{T}} S \circ \sqsubseteq_{\Gamma, C} \circ \text{Gr}(\mathbb{T}f) &= \quad (1) \\
\text{Gr}(\mathbb{T}g)^{\text{op}} \circ \Gamma S \circ \text{Gr}(\mathbb{T}f) &=
\end{aligned}$$

The first of the above equalities uses the preservation of relational composition by Γ , whereas the fourth equality exploits the fact that $\sqsubseteq_{\Gamma, C}$ and $\sqsubseteq_{\Gamma, D}$ are preorders. Hence, Γ preserves cartesian maps. This concludes our proof.

We note that the standard approach to proving preservation of cartesian maps involves the use of reindexing functors, and amounts to proving an isomorphism between $\Gamma((f, g)^* S)$ and $(\mathbb{T}f, \mathbb{T}g)^*(\Gamma S)$ in the fibre over $\mathbb{T}A \times \mathbb{T}B$, for any $(f, g) : A \times B \rightarrow C \times D$ and any $S \subseteq C \times D$. As already mentioned in Section 2.1, our definition of Rel results in all reindexing functors being uniquely defined. In particular, we have $(f, g)^* S = \text{Gr}(g)^{\text{op}} \circ S \circ \text{Gr}(f)$, and $(\mathbb{T}f, \mathbb{T}g)^*(\Gamma S) = \text{Gr}(\mathbb{T}g)^{\text{op}} \circ \Gamma S \circ \text{Gr}(\mathbb{T}f)$. The sequence of equalities in the proof of Proposition 24 thus shows that $\Gamma((f, g)^* S) = (\mathbb{T}f, \mathbb{T}g)^*(\Gamma S)$.

Throughout this paper, preservation of cartesian maps by various endofunctors on Rel will be proved directly, i.e. without reference to the reindexing functors. We believe such proofs to be shorter and more insightful in the context of this paper.

As a result of Proposition 24, monotonic relators can alternatively be defined as functors satisfying (i) and (ii) of Proposition 24, and preserving cartesian maps. We will make extensive use of this observation in what follows.

We also note that condition (i) of Proposition 24 together with the requirement that Γ preserves cartesian maps amount to Γ defining a *fibred functor over $\mathbb{T} \times \mathbb{T}$* , or to $(\Gamma, \mathbb{T} \times \mathbb{T})$ defining a *morphism between fibrations* (see [15] for a definition). We thus obtain yet another characterisation of monotonic relators, namely as fibred functors over $\mathbb{T} \times \mathbb{T}$, additionally satisfying (ii) of

Proposition 24. Finally, a fully categorical characterisation of monotonic relators can be given by replacing condition (ii) of Proposition 24 by the requirement that Γ restricts to an endofunctor on \mathbf{Preord} . However, for the purpose of this paper, the characterisation provided by Proposition 24 is the most useful one.

Remark 25 *The proof of Proposition 24 also gives:*

$$\Gamma(\mathbf{Gr}(g)^{\text{op}}) \circ \Gamma S = \mathbf{Gr}(\mathbf{T}g)^{\text{op}} \circ \Gamma S \qquad \Gamma S \circ \Gamma(\mathbf{Gr}(f)) = \Gamma S \circ \mathbf{Gr}(\mathbf{T}f)$$

for any $f : A \rightarrow C$, $g : B \rightarrow D$ and $S \subseteq C \times D$.

Since all the relators considered in the following are monotonic, from now on we will simply use the term $(\mathbf{T}\text{-})$ relator to refer to a monotonic $(\mathbf{T}\text{-})$ relator.

The remainder of this section is dedicated to defining a monotonic relator, and hence a notion of simulation, for unlabelled probabilistic transition systems. We have seen in Example 4 that such systems can be modelled as coalgebras of the endofunctor $1 + \mathcal{D}_\omega$. However, for the purpose of defining simulation relations, it will prove more convenient to work with a slightly more general type of coalgebras. Specifically, we will consider the *finite sub-probability distribution functor* $\mathcal{S}_\omega : \mathbf{Set} \rightarrow \mathbf{Set}$, defined by:

$$\begin{aligned} \mathcal{S}_\omega X &= \{ \mu : X \rightarrow [0, 1] \mid \text{supp}(\mu) \text{ finite, } \sum_{x \in X} \mu(x) \leq 1 \} \text{ for } X \in |\mathbf{Set}| \\ (\mathcal{S}_\omega f)(\mu)(y) &= \mu[f^{-1}(\{y\})] \text{ for } f : X \rightarrow Y, \mu \in \mathcal{S}_\omega X, \text{ and } y \in Y. \end{aligned}$$

The coalgebraic type \mathcal{S}_ω is a generalisation of the coalgebraic type $1 + \mathcal{D}_\omega$, in a sense made precise below.

Remark 26 *Any $1 + \mathcal{D}_\omega$ -coalgebra can be regarded as an \mathcal{S}_ω -coalgebra. To see this, let $\eta : 1 + \mathcal{D}_\omega \Rightarrow \mathcal{S}_\omega$ be the natural transformation given by:*

$$\begin{aligned} \eta_X(\iota_1(*))(x) &= 0 \text{ for } x \in X \\ \eta_X(\iota_2(\mu)) &= \mu \text{ for } \mu \in \mathcal{D}_\omega X \end{aligned}$$

with $X \in |\mathbf{Set}|$. Then, η induces a functor $\mathbf{U}_\eta : \mathbf{Coalg}(1 + \mathcal{D}_\omega) \rightarrow \mathbf{Coalg}(\mathcal{S}_\omega)$, which takes a $1 + \mathcal{D}_\omega$ -coalgebra (C, γ) to the \mathcal{S}_ω -coalgebra $(C, \eta_C \circ \gamma)$.

The use of \mathcal{S}_ω -coalgebras in modelling unlabelled probabilistic transition systems allows a unified treatment of terminating states (i.e. states for which no transition is possible) and non-terminating ones.

An \mathcal{S}_ω -relator can now be defined by relaxing the conditions in the characterisation of $1 + \mathcal{D}_\omega$ -bisimulation (see Example 7).

Definition 27 (Relator for probabilistic simulation) *The \mathcal{S}_ω -relator $\Gamma_P : \mathbf{Rel} \rightarrow \mathbf{Rel}$ takes a relation $R \subseteq A \times B$ to the relation $\Gamma_P R \subseteq \mathcal{S}_\omega A \times \mathcal{S}_\omega B$ defined*

by:

$$\mu(\Gamma_P R) \nu \quad \text{iff} \quad \mu[X] \geq \nu[Y] \text{ for any } X \subseteq A \text{ and } Y \subseteq B \text{ s.t.} \\ (\pi_1^R)^{-1}(X) \supseteq (\pi_2^R)^{-1}(Y)$$

with $\mu \in \mathcal{S}_\omega A$ and $\nu \in \mathcal{S}_\omega B$.

For Γ_P to be well-defined, we must prove that, if $(f, g) : (R \subseteq A \times B) \rightarrow (S \subseteq C \times D)$ is an arrow in Rel , then so is $(\mathcal{S}_\omega f, \mathcal{S}_\omega g) : \Gamma_P R \rightarrow \Gamma_P S$. To see this, let $\mu \in \mathcal{S}_\omega A$, $\nu \in \mathcal{S}_\omega B$ be such that $\mu(\Gamma_P R) \nu$, and let $U \subseteq C$, $V \subseteq D$ be such that $(\pi_1^S)^{-1}(U) \supseteq (\pi_2^S)^{-1}(V)$. An easy calculation shows that $(\pi_1^R)^{-1}(f^{-1}(U)) \supseteq (\pi_2^R)^{-1}(g^{-1}(V))$. This, together with $\mu(\Gamma_P R) \nu$ gives $\mu[f^{-1}(U)] \geq \nu[g^{-1}(V)]$, that is, $(\mathcal{S}_\omega f)(\mu)[U] \geq (\mathcal{S}_\omega g)(\nu)[V]$. Thus, $(\mathcal{S}_\omega f)(\mu) (\Gamma_P S) (\mathcal{S}_\omega g)(\nu)$.

Proposition 28 Γ_P is a relator.

PROOF. The first two requirements in the definition of a relator (see (i) and (ii) of Proposition 24) are immediately verified. To see that Γ_P preserves cartesian maps, let $(f, g) : (R \subseteq A \times B) \rightarrow (S \subseteq C \times D)$ be a relation-reflecting map, let $\mu \in \mathcal{S}_\omega A$, $\nu \in \mathcal{S}_\omega B$ be such that $(\mathcal{S}_\omega f)(\mu) (\Gamma_P S) (\mathcal{S}_\omega g)(\nu)$, and let $X \subseteq A$, $Y \subseteq B$ be such that $(\pi_1^R)^{-1}(X) \supseteq (\pi_2^R)^{-1}(Y)$. Also, let $U = \{c \in C \mid c = f(a) \text{ implies } a \in X\}$ and $V = g(Y)$. Then, $X \supseteq f^{-1}(U)$, $g^{-1}(V) \supseteq Y$, and $(\pi_1^S)^{-1}(U) \supseteq (\pi_2^S)^{-1}(V)$. The fact that $(\mathcal{S}_\omega f)(\mu) (\Gamma_P S) (\mathcal{S}_\omega g)(\nu)$ now gives $(\mathcal{S}_\omega f)(\mu)[U] \geq (\mathcal{S}_\omega g)(\nu)[V]$, and hence $\mu[X] \geq \mu[f^{-1}(U)] \geq \nu[g^{-1}(V)] \geq \nu[Y]$. We have thus proved that $\mu(\Gamma_P R) \nu$.

Next, we characterise the restriction of Γ_P to Preord .

Proposition 29 Let R be a preorder on A , and let $\mu, \nu \in \mathcal{S}_\omega A$. Then:

$$\mu(\Gamma_P R) \nu \quad \text{iff} \quad \mu[Y] \geq \nu[Y] \text{ for any } R^{\text{op}}\text{-closed } Y \subseteq A \quad (3)$$

where $Y \subseteq A$ is called R^{op} -closed if $y \in Y$ and aRy imply $a \in Y$.

PROOF. We begin by noting that, if $X, Y \subseteq A$, then $(\pi_1^R)^{-1}(X) \supseteq (\pi_2^R)^{-1}(Y)$ translates to $X \supseteq \bar{Y}$, where $\bar{Y} = \{a \in A \mid \exists y \in Y. aRy\}$. Also, the reflexivity and transitivity of R^{op} give $\bar{Y} \supseteq Y$ and \bar{Y} R^{op} -closed. First, let $Y \subseteq A$ be an R^{op} -closed set. Then, $(\pi_1^R)^{-1}(Y) \supseteq (\pi_2^R)^{-1}(Y)$ (as $Y \supseteq \bar{Y}$), and hence, by the definition of Γ_P , $\mu[Y] \geq \nu[Y]$. Next, let $X, Y \subseteq A$ be such that $X \supseteq \bar{Y}$. Then, since \bar{Y} is R^{op} -closed, it follows by (3) that $\mu[\bar{Y}] \geq \nu[\bar{Y}]$. We also have $\mu[X] \geq \mu[\bar{Y}]$ (as $X \supseteq \bar{Y}$) and $\nu[\bar{Y}] \geq \nu[Y]$ (as $\bar{Y} \supseteq Y$). Hence, $\mu[X] \geq \nu[Y]$.

We now investigate the notion of simulation induced by Γ_P (see Definition 13). For simplicity, we consider Γ_P -simulation on a single \mathcal{S}_ω -coalgebra (C, γ) . In this case, a relation $R \subseteq C \times C$ is a Γ_P -simulation if, whenever $c R d$ and $X \subseteq C$ is R^{op} -closed, we have $\gamma(c)[X] \geq \gamma(d)[X]$. The condition that X is R^{op} -closed amounts to X being closed under the simulation R , that is, if $x \in X$ and y simulates x via R , then also $y \in X$. The condition $\gamma(c)[X] \geq \gamma(d)[X]$ requires that a one-step transition from c is at least as likely to result in a state in X as a one-step transition from d is, whenever X is closed under simulation.

The restriction of Γ_P to **Preord** satisfies the hypotheses of Proposition 10. (This will later allow us to construct a final Γ_P -coalgebra using the final sequence of Γ_P .)

Proposition 30 $\Gamma_P : \text{Preord} \rightarrow \text{Preord}$ *preserves monics and is ω -accessible.*

PROOF (Sketch). The key observation for proving ω -accessibility is that, for $\mu, \nu \in \mathcal{S}_\omega A$, we have:

$$\mu(\Gamma_P R)\nu \quad \text{iff} \quad \mu|_Z(\Gamma_P(R|_{Z \times Z}))\nu|_Z$$

where $Z = \text{supp}(\mu) \cup \text{supp}(\nu)$, and $\mu|_Z, \nu|_Z \in \mathcal{S}_\omega(A \cap Z)$.

Remark 31 *A notion of simulation for probabilistic transition systems has also been defined in [9], namely as a preorder R on the set S of states of a probabilistic transition system, such that $s R t$ implies $\tau_a(s, X) \leq \tau_a(t, X)$ for any R -closed $X \subseteq S$ (with $\tau_a(s, X)$ giving the probability of reaching a state in X via an a -labelled transition from s). It then follows by the previous characterisation of $\Gamma_P : \text{Preord} \rightarrow \text{Preord}$ that R is a simulation preorder according to [9] (in the unlabelled case) if and only if R^{op} is a Γ_P -simulation preorder.*

4 Compositionality of Simulations

In this section we show that various operations on coalgebraic types induce corresponding operations on relators, thereby allowing the compositional derivation of notions of simulation for increasingly complex coalgebraic types. As a result, we obtain various notions of simulation for labelled transition systems, probabilistic transition systems, and probabilistic automata.

We begin by recalling the definition of products and coproducts in **Rel**. If $R_i \subseteq X_i \times Y_i$ with $i = 1, 2$, then the relations $R_1 \times R_2 \subseteq (X_1 \times X_2) \times (Y_1 \times Y_2)$

and $R_1 + R_2 \subseteq (X_1 + X_2) \times (Y_1 + Y_2)$ are defined by:

$$\begin{aligned} (x_1, x_2) (R_1 \times R_2) (y_1, y_2) & \text{ iff } x_1 R_1 y_1 \text{ and } x_2 R_2 y_2 \\ \iota_i(x_i) (R_1 + R_2) \iota_j(y_j) & \text{ iff } i = j \text{ and } x_i R_i y_i \end{aligned}$$

with $x_i \in X_i$ and $y_i \in Y_i$, for $i = 1, 2$. Similarly to products, one can define, for each relation $R_1 \subseteq X_1 \times Y_1$, a relation $(R_1)^A \subseteq (X_1)^A \times (Y_1)^A$ by:

$$f (R_1)^A g \text{ iff } f(a) R_1 g(a) \text{ for all } a \in A$$

with $f \in (X_1)^A$ and $g \in (Y_1)^A$.

The above operations on relations can be used to derive $(\mathbb{T}_1 \times \mathbb{T}_2)$ -, $(\mathbb{T}_1 + \mathbb{T}_2)$ - and $(\mathbb{T}_1)^A$ -relators from \mathbb{T}_1 - and \mathbb{T}_2 -relators.

Definition 32 (Operations on relators) *Let Γ_1 and Γ_2 be \mathbb{T}_1 - and \mathbb{T}_2 -relators, respectively. Define $\Gamma_1 \oplus \Gamma_2$, $\Gamma_1 \otimes \Gamma_2$, $(\Gamma_1)^A : \mathbf{Rel} \rightarrow \mathbf{Rel}$ by:*

- $R \subseteq X \times Y \xrightarrow{\Gamma_1 \oplus \Gamma_2} \Gamma_1(R) + \Gamma_2(R) \subseteq (\mathbb{T}_1 + \mathbb{T}_2)X \times (\mathbb{T}_1 + \mathbb{T}_2)Y$
- $R \subseteq X \times Y \xrightarrow{\Gamma_1 \otimes \Gamma_2} \Gamma_1(R) \times \Gamma_2(R) \subseteq (\mathbb{T}_1 \times \mathbb{T}_2)X \times (\mathbb{T}_1 \times \mathbb{T}_2)Y$
- $R \subseteq X \times Y \xrightarrow{(\Gamma_1)^A} \Gamma_1(R)^A \subseteq (\mathbb{T}_1 X)^A \times (\mathbb{T}_1 Y)^A$.

We note in passing that similar operations on relations were used in [13] (see also [16]) to inductively define the notion of *relation lifting* of a polynomial endofunctor.

In addition, relators can be combined using functor composition.

Proposition 33 $\Gamma_1 \circ \Gamma_2$, $\Gamma_1 \oplus \Gamma_2$, $\Gamma_1 \otimes \Gamma_2$, $(\Gamma_1)^A$ are $\mathbb{T}_1 \circ \mathbb{T}_2$ -, $\mathbb{T}_1 + \mathbb{T}_2$ -, $\mathbb{T}_1 \times \mathbb{T}_2$ -, and $(\mathbb{T}_1)^A$ -relators, respectively.

PROOF (Sketch). Some easy calculations show that all the conditions in the definition of monotonic relators (Definition 12) hold for $\Gamma_1 \circ \Gamma_2$, $\Gamma_1 \oplus \Gamma_2$, $\Gamma_1 \otimes \Gamma_2$ and $(\Gamma_1)^A$.

This allows us to derive relators (and hence notions of simulation) for combinations of coalgebraic types, from relators for the types being combined. In particular, we can derive notions of simulation for \mathbb{T} -coalgebras, with the endofunctor \mathbb{T} being generated by the following syntax:

$$\mathbb{T} ::= A \mid \text{Id} \mid \mathcal{P}_\omega \mid \mathcal{S}_\omega \mid \mathbb{T}_1 \circ \mathbb{T}_2 \mid \mathbb{T}_1 + \mathbb{T}_2 \mid \mathbb{T}_1 \times \mathbb{T}_2 \mid \mathbb{T}^A$$

(Here, $A : \mathbf{Set} \rightarrow \mathbf{Set}$ denotes the constant functor $X \mapsto A$, while $\text{Id} : \mathbf{Set} \rightarrow \mathbf{Set}$ denotes the identity functor.) For $\mathbb{T} = A$ or $\mathbb{T} = \text{Id}$, the minimal relators provide obvious choices of relators to be used. For $\mathbb{T} = \mathcal{P}_\omega$, the relator Γ_{\supseteq} of

Example 46, inducing the standard notion of simulation on unlabelled transition systems, is the obvious choice. Finally, for $\mathbb{T} = \mathcal{S}_\omega$, the relator Γ_P from Definition 27 can be used.

Example 34 Recall from Example 3 that $(\mathcal{P}_\omega)^A$ -coalgebras are essentially image-finite, A -labelled transition systems. Now let $\Gamma_{\supseteq} : \mathbf{Rel} \rightarrow \mathbf{Rel}$ be as in Example 15. A relation $R \subseteq C \times D$ is a $(\Gamma_{\supseteq})^A$ -simulation between $(\mathcal{P}_\omega)^A$ -coalgebras (C, γ) and (D, δ) if, whenever $c R d$ and $d \xrightarrow{a} d'$ with $d' \in D$, then $c \xrightarrow{a} c'$, with $c' \in C$ being such that $c' R d'$. Thus, $(\Gamma_{\supseteq})^A$ -simulation coincides with standard simulation on A -labelled transition systems.

Two other notions of simulation, namely *complete* and *ready simulation* are used in the semantics of sequential processes (see e.g. [29]). Each of these can be derived using a suitable choice of relator. However, while the notion of ready simulation can be derived compositionally, the notion of complete simulation can not. This is illustrated below.

Example 35 Let $\Gamma_{\supseteq}^R : \mathbf{Rel} \rightarrow \mathbf{Rel}$ be the \mathcal{P}_ω -relator defined by:

$$X \Gamma_{\supseteq}^R(Y) \quad \text{iff} \quad X \Gamma_{\supseteq}(Y) \text{ and } (Y = \emptyset \Rightarrow X = \emptyset)$$

for $X \in \mathcal{P}_\omega C$ and $Y \in \mathcal{P}_\omega D$. Then, a relation $R \subseteq C \times D$ is a $(\Gamma_{\supseteq}^R)^A$ -simulation between $(\mathcal{P}_\omega)^A$ -coalgebras (C, γ) and (D, δ) if, whenever $c R d$, the following hold for each $a \in A$:

- if $d \xrightarrow{a} d'$, then $c \xrightarrow{a} c'$ for some $c' \in C$ such that $c' R d'$;
- $c \xrightarrow{a} \cdot$ implies $d \xrightarrow{a} \cdot$.

where $c \xrightarrow{a} \cdot$ stands for the existence of an a -labelled transition from c . Thus, $(\Gamma_{\supseteq}^R)^A$ -simulation between $(\mathcal{P}_\omega)^A$ -coalgebras coincides with ready simulation between the associated labelled transition systems (as defined e.g. in [29]).

Example 36 Let $\Gamma_{\supseteq}^C : \mathbf{Rel} \rightarrow \mathbf{Rel}$ be the $(\mathcal{P}_\omega)^A$ -relator defined by:

$$f \Gamma_{\supseteq}^C(g) \quad \text{iff} \quad f (\Gamma_{\supseteq})^A(g) \text{ and } \left(\bigcup_{a \in A} g(a) = \emptyset \Rightarrow \bigcup_{a \in A} f(a) = \emptyset \right)$$

for $f \in (\mathcal{P}_\omega C)^A$ and $g \in (\mathcal{P}_\omega D)^A$. Then, a relation $R \subseteq C \times D$ is a Γ_{\supseteq}^C -simulation between $(\mathcal{P}_\omega)^A$ -coalgebras (C, γ) and (D, δ) if, whenever $c R d$, the following hold:

- for each $a \in A$, if $d \xrightarrow{a} d'$, then $c \xrightarrow{a} c'$ for some $c' \in C$ such that $c' R d'$;
- $c \longrightarrow \cdot$ implies $d \longrightarrow \cdot$.

where $c \longrightarrow \cdot$ stands for the existence of a transition from c . Thus, Γ_{\supseteq}^C -simulation between $(\mathcal{P}_\omega)^A$ -coalgebras coincides with complete simulation between the associated labelled transition systems (as defined e.g. in [29]).

In the case of probabilistic systems, we can also recover familiar notions of simulation, as illustrated in the following.

Example 37 Let $\Gamma_P : \mathbf{Rel} \rightarrow \mathbf{Rel}$ be as in Definition 27, and recall that A -labelled, probabilistic transition systems can be modelled as $(\mathcal{S}_\omega)^A$ -coalgebras. Then, a relation $R \subseteq C \times C$ is a $(\Gamma_P)^A$ -simulation on an $(\mathcal{S}_\omega)^A$ -coalgebra (C, γ) if, whenever $c R d$, $a \in A$ and $X \subseteq C$ is R^{op} -closed, we have $\gamma(c)(a)[X] \geq \gamma(d)(a)[X]$ (or, using the notation in Remark 31, $\tau_a(c, X) \geq \tau_a(d, X)$). Thus, $(\Gamma_P)^A$ -simulation coincides with standard simulation on A -labelled probabilistic transition systems, as defined e.g. in [9].

Example 38 The simple probabilistic automata of [27,26] can be modelled as coalgebras of the functor $\mathbb{T} = (\mathcal{P}_\omega \circ \mathcal{D}_\omega)^A$. Here we consider the slightly more general case of $(\mathcal{P}_\omega \circ \mathcal{S}_\omega)^A$ -coalgebras, and derive a notion of simulation for such coalgebras by combining the \mathcal{P}_ω -relator Γ_\supseteq and the \mathcal{S}_ω -relator Γ_P . Specifically, we consider the $(\mathcal{P}_\omega \circ \mathcal{S}_\omega)^A$ -relator $(\Gamma_\supseteq \circ \Gamma_P)^A$. Then, a relation $R \subseteq C \times D$ is a $(\Gamma_\supseteq \circ \Gamma_P)^A$ -simulation between $(\mathcal{P}_\omega \circ \mathcal{S}_\omega)^A$ -coalgebras (C, γ) and (D, δ) iff $c R d$ implies:

$$\forall a \in A. \forall \nu \in \delta(d)(a). \exists \mu \in \gamma(c)(a). (\mu[X] \geq \nu[Y] \text{ whenever } (\pi_1^R)^{-1}(X) \supseteq (\pi_2^R)^{-1}(Y))$$

In Section 6, we will derive a characterising logic for the notion of simulation obtained in Example 38, and will use that logic to compare our notion of simulation with the notion of strong simulation defined in [27] (see also [18]).

5 Expressive Logics for Simulation

We now describe an inductive method for defining logics which characterise simulation. Following [5,7], we use a notion of language constructor and an associated notion of semantics (w.r.t. an endofunctor \mathbb{T}) to formalise one step in the definition of a logic for \mathbb{T} -coalgebras. The syntax and semantics of the induced logic are then obtained by successive applications of the language constructor and of the associated semantics, respectively. We subsequently show that the expressiveness of the resulting logic w.r.t. a given notion of simulation follows from an expressivity condition involving the semantics of the language constructor, and the monotonic relator inducing that notion of simulation. Finally, we apply our results to derive Baltag's logic for coalgebraic simulation, as well as an expressive logic for simulation on unlabelled probabilistic transition systems.

We fix an endofunctor $\mathbb{T} : \mathbf{Set} \rightarrow \mathbf{Set}$ and a \mathbb{T} -relator $\Gamma : \mathbf{Rel} \rightarrow \mathbf{Rel}$, and let

$\succsim = \succsim_\Gamma$ denote the similarity relation induced by Γ . We are interested in *logics for T-coalgebras* which *characterise* Γ -similarity.

Definition 39 (Logic for coalgebras) A logic for T-coalgebras is a pair (\mathcal{L}, \models) , with \mathcal{L} a set (of formulae) and $\models = (\models_\gamma)$ a $|\text{Coalg}(\mathbb{T})|$ -indexed family of satisfaction relations $\models_\gamma \subseteq C \times \mathcal{L}$ for each $\gamma : C \rightarrow \mathbb{T}C$, such that $f(c) \models_\delta \varphi$ iff $c \models_\gamma \varphi$, for any $f : (C, \gamma) \rightarrow (D, \delta)$, $c \in C$ and $\varphi \in \mathcal{L}$.

Definition 40 (Logic which characterises similarity) Let (\mathcal{L}, \models) denote a logic for T-coalgebras. Given T-coalgebras (C, γ) and (D, δ) , we say that $c \in C$ logically simulates $d \in D$ (and write $c \geq_{\mathcal{L}} d$) if $c \models_\gamma \varphi$ whenever $d \models_\delta \varphi$, for any $\varphi \in \mathcal{L}$. The logic (\mathcal{L}, \models) characterises Γ -similarity if, for any T-coalgebras (C, γ) and (D, δ) , the logical simulation relation $\geq_{\mathcal{L}} \subseteq C \times D$ coincides with the Γ -similarity relation $\succsim \subseteq C \times D$.

It is worth noting that, if one was interested in characterising *equivalence* relations between the states of coalgebras (including Γ -simulation equivalence, or T-bisimulation), a different notion of characterising logic would be used – one would first define *logical equivalence* between the states of coalgebras as logical simulation in both directions, and subsequently require that logical equivalence coincides with the given notion of (simulation) equivalence. Such an approach was used in [5] to define characterising logics for T-bisimulation. We also note that, under the above definition, any logic which characterises Γ -simulation also characterises Γ -simulation equivalence.

Now assume that \mathbb{T} admits a final coalgebra (F, ζ) , and recall from Remark 21 that, if c and d are as in Definition 40, then $c \geq_{\mathcal{L}} d$ iff $!_\gamma(c) \succsim !_\delta(d)$. Also, Definition 39 gives $c \models_\gamma \varphi$ iff $!_\gamma(c) \models_\zeta \varphi$ (and similarly for d), and hence $c \geq_{\mathcal{L}} d$ iff $!_\gamma(c) \geq_{\mathcal{L}} !_\delta(d)$. Thus, in order to define a logic for T-coalgebras which characterises Γ -similarity, it suffices to define a set \mathcal{L} of formulae together with an interpretation of these formulae over the carrier F of the final T-coalgebra, such that the logical simulation relation induced by this interpretation coincides with the Γ -similarity relation on (F, ζ) . Now, by (iii) of Proposition 20, Γ -similarity on (F, ζ) is the final Γ -coalgebra. Also, by Proposition 10, this coalgebra can be approximated using the final sequence of Γ . We can therefore use induction over the final sequence of Γ to define the set of formulae \mathcal{L} and their interpretation over F .

Proposition 41 *The final sequence of Γ belongs to Preord.*

PROOF (Sketch). The statement follows by transfinite induction. Proposition 1 is used in the case of limit ordinals.

As a result, the final sequence of Γ coincides with the final sequence of the

restriction of Γ to **Preord**. This justifies the following definition.

Definition 42 (Relation sequence) *The relation sequence induced by Γ is the final sequence of $\Gamma : \mathbf{Preord} \rightarrow \mathbf{Preord}$.*

We can immediately infer the following:

Proposition 43 *The Set-sequence underlying the relation sequence induced by Γ is the final sequence of \mathbb{T} .*

PROOF (Sketch). The statement follows from Γ being a \mathbb{T} -relator, together with the observation that limits in **Preord** are computed from limits in **Set**.

Thus, the relation sequence induced by Γ can be written $(\succsim_\alpha), (p_\beta^\alpha : \succsim_\alpha \rightarrow \succsim_\beta)_{\beta \leq \alpha}$, where $\succsim_\alpha \subseteq Z_\alpha \times Z_\alpha$ for each α , and where $(Z_\alpha), (p_\beta^\alpha : Z_\alpha \rightarrow Z_\beta)_{\beta \leq \alpha}$ is the final sequence of \mathbb{T} (see Definition 8).

Now assume that the relation sequence induced by Γ stabilises at α . In this case, the final sequence of \mathbb{T} stabilises at, or before α . Moreover, by (iii) of Proposition 20, the α -element of the relation sequence induced by Γ gives Γ -similarity on the final \mathbb{T} -coalgebra. In the following, we will use induction along the final sequence of Γ to define a language whose formulae, when interpreted over Z_α , characterise the relation \succsim_α . The basic machinery for such inductive definitions is developed in Section 5.1. The induced logic for coalgebras is defined in Section 5.2, where a characterisability result for Γ -simulation is also formulated. Sections 5.3 and 5.4 instantiate this result to derive logics which characterise specific notions of simulation.

5.1 Language Constructors and their Semantics

Since our aim is to characterise simulation relations, the languages we are about to define only use conjunctions and (non-empty) disjunctions as logical connectives – adding negation would make it impossible to characterise pre-order relations which are not equivalence relations. We therefore use a subset $\Sigma_{\mathbf{B}} \subseteq \{\mathbf{tt}, \wedge, \vee, \bigwedge, \bigvee\}$ to indicate the logical connectives employed by a particular language (where we write \wedge and \vee for binary conjunction and disjunction, respectively, and \bigwedge and \bigvee for their infinitary versions).

Definition 44 (Language constructor, and induced language) *A language constructor is an accessible endofunctor $\mathbf{S} : \mathbf{Alg}(\Sigma_{\mathbf{B}}) \rightarrow \mathbf{Alg}(\Sigma_{\mathbf{B}})$. The language $\mathcal{L}(\mathbf{S})$ induced by \mathbf{S} is the initial algebra of \mathbf{S} .*

Remark 45 If S is an inclusion-preserving, ω -accessible endofunctor, then the language $\mathcal{L}(S)$ is given by $\bigcup_n \mathcal{L}_n(S)$, where the $\Sigma_{\mathbf{B}}$ -algebras $\mathcal{L}_n(S)$ are defined inductively by:

- $\mathcal{L}_0(S)$ is the initial $\Sigma_{\mathbf{B}}$ -algebra,
- $\mathcal{L}_{n+1}(S) = S(\mathcal{L}_n(S))$ for $n \in \omega$.

We also note that language constructors which are not ω -accessible will generally give rise to infinitary languages.

Example 46 Let $\Sigma_{\mathbf{B}} = \{\mathbf{tt}, \wedge\}$, and let $S_{\sqsupseteq} : \mathbf{Alg}(\Sigma_{\mathbf{B}}) \rightarrow \mathbf{Alg}(\Sigma_{\mathbf{B}})$ denote the language constructor taking a $\Sigma_{\mathbf{B}}$ -algebra L to the free $\Sigma_{\mathbf{B}}$ -algebra over the set $\{\diamond\varphi \mid \varphi \in L\}$. The language $\mathcal{L}(S_{\sqsupseteq})$ can alternatively be generated using the following syntax:

$$\varphi ::= \mathbf{tt} \mid \varphi \wedge \psi \mid \diamond\varphi$$

In order to define an associated semantics for a language constructor, we introduce the notion of *interpretation*. To this end, we note that for a set X , its power set $\mathcal{P}X$ can be endowed with $\Sigma_{\mathbf{B}}$ -algebra structure by interpreting \mathbf{tt} , \vee (or \bigvee) and \wedge (or \bigwedge) as X , union and intersection, respectively.

Definition 47 (Interpretation) An interpretation of a $\Sigma_{\mathbf{B}}$ -algebra L over a set X is a $\Sigma_{\mathbf{B}}$ -algebra morphism $d : L \rightarrow \mathcal{P}X$. A map between interpretations $d : L \rightarrow \mathcal{P}X$ and $d' : L' \rightarrow \mathcal{P}X'$ is a pair (l, f) with $l : L \rightarrow L'$ a $\Sigma_{\mathbf{B}}$ -algebra morphism and $f : X' \rightarrow X$ a function, such that $\hat{\mathcal{P}}f \circ d = d' \circ l$ (where $\hat{\mathcal{P}} : \mathbf{Set} \rightarrow \mathbf{Set}$ denotes the contravariant powerset functor). The category whose objects are interpretations of $\Sigma_{\mathbf{B}}$ -algebras, and whose arrows are maps between such interpretations is denoted $\mathbf{Int}_{\mathbf{B}}$.

Remark 48 Any interpretation $d : L \rightarrow \mathcal{P}X$ induces a logical map $s : X \rightarrow \mathcal{P}L$, defined by $s(x) = \{\varphi \in L \mid x \in d(\varphi)\}$ for $x \in X$. With this notation, the condition defining a map between $d : L \rightarrow \mathcal{P}X$ and $d' : L' \rightarrow \mathcal{P}X'$ becomes $s \circ f = \hat{\mathcal{P}}l \circ s'$ (where $s : X \rightarrow \mathcal{P}L$ and $s' : X' \rightarrow \mathcal{P}L'$ are the logical maps induced by d and d' , respectively).

We let $\mathbf{L} : \mathbf{Int}_{\mathbf{B}} \rightarrow \mathbf{Alg}(\Sigma_{\mathbf{B}})$ and $\mathbf{E} : \mathbf{Int}_{\mathbf{B}} \rightarrow \mathbf{Set}^{\text{op}}$ denote the functors taking interpretations $d : L \rightarrow \mathcal{P}X$ to L and X , respectively, and maps (l, f) between interpretations $d : L \rightarrow \mathcal{P}X$ and $d' : L' \rightarrow \mathcal{P}X'$ to $l : L \rightarrow L'$ and $f : X' \rightarrow X$, respectively. The following result was proved in [5] for a slightly less general notion of interpretation. The result and its proof generalise to the present setting.

Proposition 49 $\mathbf{Int}_{\mathbf{B}}$ is cocomplete, and \mathbf{E} preserves colimits.

Colimits in $\mathbf{Int}_{\mathbf{B}}$ are constructed from limits in \mathbf{Set} and colimits in certain comma categories of $\mathbf{Int}_{\mathbf{B}}$. For instance, an initial object in $\mathbf{Int}_{\mathbf{B}}$ is given by the

interpretation $d_0 : L_0 \rightarrow \mathcal{P}1$, with L_0 an initial $\Sigma_{\mathbf{B}}$ -algebra and d_0 the unique $\Sigma_{\mathbf{B}}$ -morphism arising from the initiality of L_0 .

The following result can also be proved in a similar way.

Proposition 50 *L preserves colimits.*

We now return to defining a semantics for a language constructor. We have seen that a language constructor \mathbf{S} induces a language $\mathcal{L}(\mathbf{S})$. Our aim is to interpret this language over \mathbf{T} -coalgebras. The following notion constitutes an intermediary step in this direction.

Definition 51 (Semantics for language constructor) *Let $\mathbf{T} : \mathbf{Set} \rightarrow \mathbf{Set}$ be an endofunctor, and let $\mathbf{S} : \mathbf{Alg}(\Sigma_{\mathbf{B}}) \rightarrow \mathbf{Alg}(\Sigma_{\mathbf{B}})$ be a language constructor. A \mathbf{T} -semantics for \mathbf{S} is a functor $\mathbb{S} : \mathbf{Int}_{\mathbf{B}} \rightarrow \mathbf{Int}_{\mathbf{B}}$ such that $L \circ \mathbb{S} = \mathbf{S} \circ L$ and $E \circ \mathbb{S} = \mathbf{T}^{\text{op}} \circ E$:*

$$\begin{array}{ccc}
 \mathbf{Alg}(\Sigma_{\mathbf{B}}) & \xrightarrow{\mathbf{S}} & \mathbf{Alg}(\Sigma_{\mathbf{B}}) \\
 \uparrow L & & \uparrow L \\
 \mathbf{Int}_{\mathbf{B}} & \xrightarrow{\mathbb{S}} & \mathbf{Int}_{\mathbf{B}} \\
 \downarrow E & & \downarrow E \\
 \mathbf{Set}^{\text{op}} & \xrightarrow{\mathbf{T}^{\text{op}}} & \mathbf{Set}^{\text{op}}
 \end{array}$$

Thus, a \mathbf{T} -semantics for \mathbf{S} takes an interpretation $d : L \rightarrow \mathcal{P}X$ to an interpretation $d' : \mathbf{S}L \rightarrow \mathcal{P}TX$, and a map (l, f) of interpretations to $(\mathbf{S}l, \mathbf{T}f)$. In particular, we note that the action of a \mathbf{T} -semantics on maps between interpretations is uniquely determined by the actions of \mathbf{S} and \mathbf{T} on arrows in $\mathbf{Alg}(\Sigma_{\mathbf{B}})$ and \mathbf{Set} , respectively.

Example 52 (\mathcal{P}_{ω} -semantics for \mathbf{S}_{\supseteq}) *Let $\mathbf{S}_{\supseteq} : \mathbf{Alg}(\Sigma_{\mathbf{B}}) \rightarrow \mathbf{Alg}(\Sigma_{\mathbf{B}})$ be as in Example 46. A \mathcal{P}_{ω} -semantics for \mathbf{S}_{\supseteq} is given by the functor $\mathbb{S}_{\supseteq} : \mathbf{Int}_{\mathbf{B}} \rightarrow \mathbf{Int}_{\mathbf{B}}$ taking $d : L \rightarrow \mathcal{P}X$ to $d' : \mathbf{S}_{\supseteq}(L) \rightarrow \mathcal{P}(\mathcal{P}_{\omega}X)$, with $d'(\diamond\varphi) = \{Y \in \mathcal{P}_{\omega}X \mid Y \cap d(\varphi) \neq \emptyset\}$. (The requirement that d' defines a $\Sigma_{\mathbf{B}}$ -algebra morphism uniquely determines the action of d' on formulae containing logical connectives.)*

Variations of the notions of language constructor and associated semantics (Definitions 44 and 51) have also been considered in [5,7]. There, we were interested in the ability of interpretations $d : L \rightarrow \mathcal{P}X$ to characterise elements of X using formulae in L . Here, our aim is to characterise certain preorders on X .

Definition 53 (Expressiveness of interpretation) *Let $d : L \rightarrow \mathcal{P}X$ be an interpretation. If $x, y \in X$, we write $y \geq_L x$ if $y \in d(\varphi)$ whenever $x \in d(\varphi)$, with $\varphi \in L$. Then, d is called adequate for a preorder $R \subseteq X \times X$ if $R \subseteq \geq_L$, and expressive for R if, in addition, $R \supseteq \geq_L$.*

Thus, adequacy of an interpretation $d : L \rightarrow \mathcal{P}X$ for a preorder R amounts to the logical map $s : X \rightarrow \mathcal{P}L$ induced by d (see Remark 48) defining a map $s : (R \subseteq X \times X) \rightarrow (\supseteq \subseteq \mathcal{P}L \times \mathcal{P}L)$ in **Preord**, whereas expressiveness of d for R amounts to s being cartesian (or order-reflecting).

The following condition involving the \mathbb{T} -semantics of a language constructor and a monotonic \mathbb{T} -relator Γ will later be used to ensure that a sequence of interpretations over the elements of the final sequence of \mathbb{T} are expressive w.r.t. the corresponding relations in the final sequence of Γ .

Definition 54 (Preservation of expressiveness) *Let $\Gamma : \text{Rel} \rightarrow \text{Rel}$ be a \mathbb{T} -relator, and let $\mathbb{S} : \text{Alg}(\Sigma_{\mathbb{B}}) \rightarrow \text{Alg}(\Sigma_{\mathbb{B}})$ be a language constructor. A \mathbb{T} -semantics \mathbb{S} for \mathbb{S} preserves expressiveness w.r.t. Γ if it maps an interpretation $d : L \rightarrow \mathcal{P}X$ which is expressive for $R \subseteq X \times X$ to an interpretation $d' : \mathbb{S}L \rightarrow \mathcal{P}\mathbb{T}X$ which is expressive for $\Gamma R \subseteq \mathbb{T}X \times \mathbb{T}X$.*

Example 55 *The \mathcal{P}_ω -semantics \mathbb{S}_\supseteq for \mathbb{S}_\supseteq defined in Example 52 preserves expressiveness w.r.t. Γ_\supseteq . To see this, let $d : L \rightarrow \mathcal{P}X$ be expressive for $R \subseteq X \times X$, and let $Y, Z \in \mathcal{P}_\omega X$. It follows easily that $Y (\Gamma_\supseteq R) Z$ implies $Y \geq_{\mathbb{S}_\supseteq L} Z$. Now assume that $Y (\Gamma_\supseteq R) Z$ does not hold. To show that $Y \not\geq_{\mathbb{S}_\supseteq L} Z$, we need to define a formula $\phi \in \mathbb{S}_\supseteq L$ such that $Z \in \mathbb{S}_\supseteq(d)(\phi)$ but $Y \notin \mathbb{S}_\supseteq(d)(\phi)$. First, the fact that $Y (\Gamma_\supseteq R) Z$ does not hold gives $z \in Z$ such that $y R z$ does not hold for any $y \in Y$. The expressiveness of d for R then gives, for each $y \in Y$, some $\varphi_y \in L$ such that $z \in d(\varphi_y)$ but $y \notin d(\varphi_y)$. Then, ϕ can be taken to be $\diamond(\bigwedge_{y \in Y} \varphi_y)$. (Since $Y \in \mathcal{P}_\omega X$, the conjunction defining ϕ is finite.)*

5.2 Induced Logics for Simulation

We will derive a logic for \mathbb{T} -coalgebras from an interpretation $d : \mathcal{L} \rightarrow \mathcal{P}F$, with F the carrier of a final \mathbb{T} -coalgebra. The choice of d will depend on the particular notion of simulation we aim to characterise, i.e. on the choice of Γ . Specifically, d will be defined as the α -element of the initial sequence of a \mathbb{T} -semantics \mathbb{S} for a language constructor \mathbb{S} , with \mathbb{S} and \mathbb{S} being chosen so as to yield expressive interpretations for the relations in the final sequence of Γ , and with the ordinal α being chosen so that the final sequence of \mathbb{T} stabilises at or before α (and hence Z_α is the carrier of a final \mathbb{T} -coalgebra).

Remark 56 *For any ordinal α , an interpretation $d : \mathcal{L} \rightarrow \mathcal{P}Z_\alpha$ induces a logic (\mathcal{L}, \models) for \mathbb{T} -coalgebras, with $c \models_\gamma \varphi$ iff $\gamma_\alpha(c) \in d(\varphi)$, for any \mathbb{T} -coalgebra (C, γ) , $c \in C$ and $\varphi \in \mathcal{L}$ (where $\gamma_\alpha : C \rightarrow Z_\alpha$ is as in Remark 9). The fact that coalgebra morphisms $f : (C, \gamma) \rightarrow (D, \delta)$ define morphisms of cones $f : (\gamma_\alpha) \rightarrow (\delta_\alpha)$ ensures the correctness of this definition.*

The following property of the initial sequence of \mathbb{S} will prove useful in what follows.

Proposition 57 *Let $\mathbb{S} : \mathbf{Alg}(\Sigma_{\mathbb{B}}) \rightarrow \mathbf{Alg}(\Sigma_{\mathbb{B}})$ be a language constructor, and let $\mathbb{S} : \mathbf{Int}_{\mathbb{B}} \rightarrow \mathbf{Int}_{\mathbb{B}}$ be a \mathbb{T} -semantics for \mathbb{S} . The $\mathbf{Alg}(\Sigma_{\mathbb{B}})$ - and \mathbf{Set} -sequences underlying the initial sequence of \mathbb{S} are the initial sequence of \mathbb{S} and the final sequence of \mathbb{T} , respectively.*

PROOF (Sketch). Immediate from Definition 51 and (the dual of) Definition 8.

Thus, if $(L_\alpha), (\iota_\beta^\alpha : L_\beta \rightarrow L_\alpha)_{\beta \leq \alpha}$ denotes the initial sequence of \mathbb{S} and $(Z_\alpha), (p_\beta^\alpha : Z_\alpha \rightarrow Z_\beta)_{\beta \leq \alpha}$ denotes the final sequence of \mathbb{T} , then the elements of the initial sequence of \mathbb{S} are interpretations of the form $d_\alpha : L_\alpha \rightarrow \mathcal{P}Z_\alpha$, while the arrows defining this initial sequence are of the form $(\iota_\beta^\alpha, p_\beta^\alpha) : d_\beta \rightarrow d_\alpha$, with $\beta \leq \alpha$. For an ordinal α , we write $s_\alpha : Z_\alpha \rightarrow \mathcal{P}L_\alpha$ for the logical map induced by d_α .

The next result concerns the expressiveness of the interpretations d_α w.r.t. the relations $\succsim_\alpha \subseteq Z_\alpha \times Z_\alpha$ in the relation sequence induced by Γ .

Theorem 58 *Let $\mathbb{S} : \mathbf{Alg}(\Sigma_{\mathbb{B}}) \rightarrow \mathbf{Alg}(\Sigma_{\mathbb{B}})$ be a language constructor, and let $\mathbb{S} : \mathbf{Int}_{\mathbb{B}} \rightarrow \mathbf{Int}_{\mathbb{B}}$ be a \mathbb{T} -semantics for \mathbb{S} . If \mathbb{S} preserves expressiveness w.r.t. Γ , then $d_\alpha : L_\alpha \rightarrow \mathcal{P}Z_\alpha$ is expressive for $\succsim_\alpha \subseteq Z_\alpha \times Z_\alpha$, for any ordinal α .*

PROOF. The proof is by transfinite induction on α . For $\alpha = 0$, the expressiveness of $d_0 : L_0 \rightarrow \mathcal{P}1$ for $\succsim_0 = =_1$ follows immediately. For $\alpha = \beta + 1$, the expressiveness of d_α for \succsim_α follows from the expressiveness of d_β for \succsim_β together with the preservation of expressiveness by \mathbb{S} . Finally, let α be a limit ordinal, and assume that d_β is expressive for \succsim_β , for any $\beta < \alpha$.

To show that d_α is adequate for \succsim_α , let $x, y \in Z_\alpha$ be such that $y \succsim_\alpha x$. Then, for $\beta < \alpha$, $p_\beta^\alpha(y) \succsim_\beta p_\beta^\alpha(x)$ (as $p_\beta^\alpha : (Z_\alpha, \succsim_\alpha) \rightarrow (Z_\beta, \succsim_\beta)$ defines a map in \mathbf{Preord}), and hence $s_\beta(p_\beta^\alpha(y)) \supseteq s_\beta(p_\beta^\alpha(x))$ (using the adequacy of d_β for \succsim_β). Now let $\varphi \in s_\alpha(x)$. Since the cocone $(\iota_\beta^\alpha)_{\beta < \alpha}$ is colimiting (see Proposition 57), it follows that φ is either of the form $\iota_\beta^\alpha(\psi)$ with $\beta < \alpha$ and $\psi \in L_\beta$, or a boolean combination of formulae of this form (The standard construction of colimits in categories of algebras, namely as quotients of the free algebras over the colimits in the underlying category, is used here.) We can therefore use induction on φ to prove that $\varphi \in s_\alpha(y)$. Moreover, only the cases $\varphi = \bigwedge_i \varphi_i$ and $\varphi = \bigvee_i \varphi_i$ need to be considered in the induction step, since binary conjunctions and disjunctions are already covered by the base case.

1. If $\varphi = \iota_\beta^\alpha(\psi)$ with $\beta < \alpha$ and $\psi \in L_\beta$, then $\psi \in (\hat{\mathcal{P}}\iota_\beta^\alpha)(s_\alpha(x)) = s_\beta(p_\beta^\alpha(x))$, and hence $\psi \in s_\beta(p_\beta^\alpha(y)) = (\hat{\mathcal{P}}\iota_\beta^\alpha)(s_\alpha(y))$. (Remark 48 is also used here.) This now gives $\varphi = \iota_\beta^\alpha(\psi) \in s_\alpha(y)$.
2. If $\varphi = \bigwedge_i \varphi_i \in s_\alpha(x)$, then $\varphi_i \in s_\alpha(x)$ for each $i \in I$ (using the fact that $d_\alpha : L_\alpha \rightarrow \mathcal{P}Z_\alpha$ preserves the $\Sigma_{\mathbf{B}}$ -structure). Hence, by the induction hypothesis, $\varphi_i \in s_\alpha(y)$ for each $i \in I$. This, in turn, gives $\varphi = \bigwedge_i \varphi_i \in s_\alpha(y)$. The case when $\varphi = \bigvee_i \varphi_i$ is treated similarly.

Hence, $s_\alpha(y) \supseteq s_\alpha(x)$. This concludes the proof of adequacy of d_α for \succeq_α .

To show that d_α is expressive for \succeq_α , let $x, y \in Z_\alpha$ be such that $s_\alpha(y) \supseteq s_\alpha(x)$. Then, for $\beta < \alpha$, Remark 48 gives $s_\beta(p_\beta^\alpha(y)) \supseteq s_\beta(p_\beta^\alpha(x))$, while the expressiveness of d_β for \succeq_β gives $p_\beta^\alpha(y) \succeq_\beta p_\beta^\alpha(x)$. The fact that the cone $(p_\beta^\alpha)_{\beta < \alpha}$ is limiting finally gives $y \succeq_\alpha x$. This concludes our proof.

The previous result, together with (iii) of Proposition 20 justify the following definition.

Definition 59 (Logic induced by \mathbb{S} and Γ) *Let $\Gamma : \text{Rel} \rightarrow \text{Rel}$ be such that its final sequence stabilises at α . The logic induced by \mathbb{S} and Γ is the logic induced by the interpretation $d_\alpha : L_\alpha \rightarrow \mathcal{P}Z_\alpha$, as defined in Remark 56.*

The next result allows us to derive logics which characterise Γ -similarity from \mathbb{T} -semantics which preserve expressiveness w.r.t. Γ .

Corollary 60 *Let $\mathbb{S} : \text{Alg}(\Sigma_{\mathbf{B}}) \rightarrow \text{Alg}(\Sigma_{\mathbf{B}})$, $\mathbb{S} : \text{Int}_{\mathbf{B}} \rightarrow \text{Int}_{\mathbf{B}}$ and $\Gamma : \text{Rel} \rightarrow \text{Rel}$ be as in Theorem 58, and assume that \mathbb{S} preserves expressiveness w.r.t. Γ . If the final sequence of Γ stabilises (at α), then the logic induced by \mathbb{S} and Γ characterises \succeq .*

PROOF. Let (C, γ) and (D, δ) be \mathbb{T} -coalgebras, and let $c \in C$ and $d \in D$. Then:

$$c \succeq d \text{ iff } !_\gamma(c) \succeq !_\delta(d) \text{ iff } !_\gamma(c) \succeq_\alpha !_\delta(d) \text{ iff } !_\gamma(c) \geq_{L_\alpha} !_\delta(d) \text{ iff } c \geq_{L_\alpha} d$$

The above equivalences follow from Remark 21, (iii) of Proposition 20, Theorem 58 and Definition 39, respectively.

In particular, if Γ preserves monics and is accessible, then by Proposition 10, the final sequence of Γ stabilises, and therefore Corollary 60 can be applied.

We conclude this section with some results concerning the final sequence of a \mathbb{T} -relator Γ , in case the hypotheses of Theorem 58 are satisfied.

Proposition 61 *Let $\mathbf{S} : \text{Alg}(\Sigma_{\mathbf{B}}) \rightarrow \text{Alg}(\Sigma_{\mathbf{B}})$, $\mathbb{S} : \text{Int}_{\mathbf{B}} \rightarrow \text{Int}_{\mathbf{B}}$ and $\Gamma : \text{Rel} \rightarrow \text{Rel}$ be as in Theorem 58, and assume that \mathbb{S} preserves expressiveness w.r.t. Γ . If the final sequence of \mathbf{T} stabilises at α , and the initial sequence of \mathbf{S} stabilises at, or before, α , then the final sequence of Γ also stabilises at α .*

PROOF. By Proposition 57, the $\text{Alg}(\Sigma_{\mathbf{B}})$ - and Set -sequences underlying the initial sequence of \mathbb{S} are the initial sequence of \mathbf{S} and the final sequence of \mathbf{T} , respectively. Moreover, the additional constraints on \mathbf{T} and \mathbf{S} together with the definition of arrows in $\text{Int}_{\mathbf{B}}$ ensure that the initial sequence of \mathbb{S} also stabilises at α .

On the other hand, by Theorem 58, $\geq_{L_{\alpha}}$ and $\geq_{L_{\alpha+1}}$ characterise \succsim_{α} and $\succsim_{\alpha+1}$, respectively. Hence, for $x, y \in Z_{\alpha+1}$, the following holds:

$$x \succsim_{\alpha+1} y \text{ iff } x \geq_{L_{\alpha+1}} y \text{ iff } p_{\alpha}^{\alpha+1}(x) \geq_{L_{\alpha}} p_{\alpha}^{\alpha+1}(y) \text{ iff } p_{\alpha}^{\alpha+1}(x) \succsim_{\alpha} p_{\alpha}^{\alpha+1}(y)$$

with the second equivalence following from the fact that $(\iota_{\alpha}^{\alpha+1}, p_{\alpha}^{\alpha+1})$ defines an isomorphism in $\text{Int}_{\mathbf{B}}$. As a result, $p_{\alpha}^{\alpha+1} : \succsim_{\alpha+1} \rightarrow \succsim_{\alpha}$ is an isomorphism in Rel ⁶, and hence the final sequence of Γ stabilises at α .

Thus, Proposition 61 allows us to make statements about the degree of accessibility of a \mathbf{T} -relator Γ , by exhibiting a language constructor \mathbf{S} , and a \mathbf{T} -semantics for \mathbf{S} which preserves expressiveness w.r.t. Γ . This, in turn, allows us to apply Corollary 60.

Now assume that \mathbf{T} is ω -accessible. Then, by Proposition 11, the final sequence of \mathbf{T} stabilises at (or before) $\omega + \omega$. Moreover, the maps $p_{\omega+n}^{\omega+n+1}$ with $n = 0, 1, \dots$ are all injective. Combining this observation with Proposition 61 yields the following result.

Corollary 62 *Let $\mathbf{T} : \text{Set} \rightarrow \text{Set}$ be an ω -accessible endofunctor, let $\mathbf{S} : \text{Alg}(\Sigma_{\mathbf{B}}) \rightarrow \text{Alg}(\Sigma_{\mathbf{B}})$, $\mathbb{S} : \text{Int}_{\mathbf{B}} \rightarrow \text{Int}_{\mathbf{B}}$ and $\Gamma : \text{Rel} \rightarrow \text{Rel}$ be as in Proposition 61, and assume that \mathbf{S} is ω -accessible and that \mathbb{S} preserves expressiveness w.r.t. Γ . Then:*

- (i) *The final sequence of Γ stabilises at (or before) $\omega + \omega$.*
- (ii) *The maps $p_{\omega+n}^{\omega+n+1} : \succsim_{\omega+n+1} \rightarrow \succsim_{\omega+n+1}$ with $n = 0, 1, \dots$ are order-reflecting.*

PROOF. The fact that \mathbf{S} is ω -accessible results in its initial sequence stabilising at ω . The first statement now follows from Propositions 11 and 61. The

⁶ Note that, since the final sequence of \mathbf{T} stabilises at α , $p_{\alpha}^{\alpha+1}$ is an isomorphism in Set .

second statement follows by an argument similar to the one in the proof of Proposition 61.

Thus, under the hypotheses of Corollary 62, the last ω -steps in the final sequence of Γ are determined by the corresponding steps in the final sequence of \mathbb{T} . The induced logic for coalgebras is also not influenced by these steps – the first ω -steps completely define the interpretations of formulae over \mathbb{T} -coalgebras.

Proposition 61 (or Corollary 62) can be applied to the \mathcal{P}_ω -relator Γ_{\supseteq} of Example 15.

Example 63 *Let Γ_{\supseteq} , \mathbb{S}_{\supseteq} and \mathbb{S}_{\supseteq} be as in Examples 15, 46 and 52, respectively. The initial sequence of \mathbb{S}_{\supseteq} stabilises at ω , while the final sequence of \mathcal{P}_ω stabilises at $\omega + \omega$; hence, by (the proof of) Proposition 61, the initial sequence of \mathbb{S}_{\supseteq} and final sequence of Γ also stabilise at $\omega + \omega$. Moreover, the last ω -steps in the initial sequence of \mathbb{S}_{\supseteq} do not affect the semantics of the logic induced by \mathbb{S}_{\supseteq} and Γ_{\supseteq} . As a result, the induced logic coincides with a fragment of standard modal logic. Its syntax is given by:*

$$\varphi ::= \mathbf{tt} \mid \diamond\varphi \mid \varphi \wedge \psi$$

whereas its (coalgebraic) semantics is defined by:

$$c \models_{\gamma} \diamond\varphi \quad \text{iff} \quad \exists d \in \gamma(c). d \models \varphi$$

(and the usual clauses for \mathbf{tt} and \wedge).

The next two subsections apply the results of this section in order to derive logics which characterise other specific notions of simulation.

5.3 Baltag's Logic for Coalgebraic Simulation

Here we define a language constructor and associated semantics which mirror the construction of Baltag's logic for Γ -simulation [1], and prove that the given semantics preserves expressiveness w.r.t. the relator Γ . We thus obtain an alternative definition of the logic in [1], as well as an alternative (inductive) proof of its expressiveness.

Throughout this section, $\Sigma_{\mathbb{B}} = \{\wedge\}$, $\mathbb{T} : \mathbf{Set} \rightarrow \mathbf{Set}$ denotes an inclusion-preserving endofunctor, and $\Gamma : \mathbf{Rel} \rightarrow \mathbf{Rel}$ denotes a \mathbb{T} -relator. Also, we identify interpretations $d : L \rightarrow \mathcal{P}X$ with relations $\models \subseteq X \times L$ subject to the additional constraint that $x \models \wedge \Phi$ iff $x \models \varphi$ for all $\varphi \in \Phi$, for any $\Phi \in \mathcal{P}L$ – any interpretation $d : L \rightarrow \mathcal{P}X$ determines such a relation, with

$x \models \varphi$ iff $x \in d(\varphi)$, and conversely, any relation $\models \subseteq X \times L$ subject to the previous constraint determines an interpretation $d : L \rightarrow \mathcal{P}X$, with $d(\varphi) = \{x \in X \mid x \models \varphi\}$.

As mentioned earlier, a different category of relations, and a different, but equivalent notion of monotonic relator were considered in [1]. Nonetheless, the logics proposed in [1] can also be derived using the approach in Section 5.2.

The logics in loc. cit. are parameterised by a monotonic relator Γ . Their syntax only depends on the endofunctor \mathbb{T} , but their semantics depends on the relator Γ .

Definition 64 ([1]) *The language $\mathcal{L}_{\mathbb{T}}$ is defined inductively by:*

$$\varphi ::= \bigwedge \Phi \mid \Box \Psi \quad \Phi \subseteq \mathcal{L}_{\mathbb{T}}, \Psi \in \mathbb{T}\Phi$$

The coalgebraic semantics of $\mathcal{L}_{\mathbb{T}}$ is defined inductively by:

$$\begin{aligned} c \models_{\gamma} \bigwedge \Phi & \text{ iff } c \models_{\gamma} \varphi \text{ for all } \varphi \in \Phi \\ c \models_{\gamma} \Box \Psi & \text{ iff } \gamma(c)(\Gamma \models) \Psi \end{aligned}$$

for each \mathbb{T} -coalgebra (C, γ) and each $c \in C$.

Note that, in the above definition, we have identified the span defined by $\Gamma \models$ with the relation it induces on $\mathbb{T}C \times \mathbb{T}\mathcal{L}_{\mathbb{T}}$ (in the second clause defining the semantics of $\mathcal{L}_{\mathbb{T}}$).

Remark 65 *The logic defined in [1] also contains infinitary disjunctions (with the standard interpretation), and a modal operator \Diamond (interpreted via the transposed relator Γ^{\sim}). Since neither infinitary disjunctions nor the \Diamond operator are needed to characterise Γ -simulation, in Definition 64 we have only considered a fragment of the logic in [1].*

The definition of $\mathcal{L}_{\mathbb{T}}$ can be captured using the following language constructor and associated \mathbb{T} -semantics.

Definition 66 ($\mathbb{S}_{\mathbb{T}}$ and \mathbb{S}_{Γ}) *The language constructor $\mathbb{S}_{\mathbb{T}} : \text{Alg}(\Sigma_{\mathbb{B}}) \rightarrow \text{Alg}(\Sigma_{\mathbb{B}})$ takes a $\Sigma_{\mathbb{B}}$ -algebra L to the free $\Sigma_{\mathbb{B}}$ -algebra over $\mathbb{T}L$. The \mathbb{T} -semantics $\mathbb{S}_{\Gamma} : \text{Int}_{\mathbb{B}} \rightarrow \text{Int}_{\mathbb{B}}$ for $\mathbb{S}_{\mathbb{T}}$ takes $\models \subseteq X \times L$ to the natural extension of the relation $(\Gamma \models) \subseteq \mathbb{T}X \times \mathbb{T}L$ to formulae containing infinitary conjunctions.*

For \mathbb{S}_{Γ} to be well-defined, we must prove that:

$$(\mathbb{T}f)(t)(\Gamma \models_1) \phi \text{ iff } t(\Gamma \models_2)(\mathbb{S}_{\mathbb{T}}l)(\phi) \quad (4)$$

for any map of interpretations $(l, f) : (\models_1 \subseteq X_1 \times L_1) \rightarrow (\models_2 \subseteq X_2 \times L_2)$, any $t \in \mathbb{T}X_2$ and any $\phi \in \mathbb{S}_{\mathbb{T}}L_1$. The following characterisation of the logical

map induced by $\mathbb{S}_\Gamma(\models)$ will be used to prove this.

Lemma 67 *Let $\models \subseteq X \times L$ be an interpretation with logical map $s : X \rightarrow \mathcal{P}L$, and let $e : \mathbb{T}\hat{\mathcal{P}} \Rightarrow \hat{\mathcal{P}}\mathbb{T}$ be given by $e_X(U) = \{t \in \mathbb{T}X \mid U(\Gamma\exists)t\}$ for $X \in |\mathbf{Set}|$ and $U \in \mathbb{T}\mathcal{P}X$ (where \exists denotes the converse of the membership relation). Then:*

- (i) *e is a natural transformation;*
- (ii) *The logical map $s' : \mathbb{T}X \rightarrow \mathcal{P}\mathbb{T}L$ induced by $(\Gamma\models) \subseteq \mathbb{T}X \times \mathbb{T}L$ is given by $e_L \circ \mathbb{T}s$.*

PROOF. We note that, for $f : X \rightarrow Y$, $\exists \circ \mathbf{Gr}(\hat{\mathcal{P}}f) = \mathbf{Gr}(f)^{\text{op}} \circ \exists$. Preservation of relational composition by Γ together with Remark 25 then give $(\Gamma\exists) \circ \mathbf{Gr}(\mathbb{T}\hat{\mathcal{P}}f) = \mathbf{Gr}(\mathbb{T}f)^{\text{op}} \circ (\Gamma\exists)$, i.e. $(\mathbb{T}\hat{\mathcal{P}}f)(V)(\Gamma\exists)t$ if and only if $V(\Gamma\exists)(\mathbb{T}f)(t)$, for $V \in \mathbb{T}\hat{\mathcal{P}}Y$ and $t \in \mathbb{T}X$. But this is equivalent to $e_X \circ \mathbb{T}\hat{\mathcal{P}}f = \hat{\mathcal{P}}\mathbb{T}f \circ e_Y$. Hence, e is natural.

We also note that $(s, 1_L) : (\models \subseteq X \times L) \rightarrow (\exists \subseteq \mathcal{P}L \times L)$ is a cartesian map in Rel. (This follows directly from the definition of s .) Preservation of cartesian maps by Γ then gives $t(\Gamma\models)\phi$ if and only if $(\mathbb{T}s)(t)(\Gamma\exists)\phi$, for $t \in \mathbb{T}X$ and $\phi \in \mathbb{T}L$. That is, $\phi \in s'(t)$ if and only if $\phi \in e_L((\mathbb{T}s)(t))$. Hence, $s' = e_L \circ \mathbb{T}s$.

Proposition 68 *\mathbb{S}_Γ is well-defined.*

PROOF. We use induction on ϕ to prove (4). If $\phi \in \mathbb{T}L$, (4) is equivalent to $s'_1 \circ \mathbb{T}f = \hat{\mathcal{P}}\mathbb{T}l \circ s'_2$, where $s'_1 : \mathbb{T}X_1 \rightarrow \mathcal{P}\mathbb{T}L_1$ and $s'_2 : \mathbb{T}X_2 \rightarrow \mathcal{P}\mathbb{T}L_2$ are the logical maps induced by $\Gamma\models_1$ and $\Gamma\models_2$. By (ii) of Lemma 67, this is equivalent to $e_{L_1} \circ \mathbb{T}s_1 \circ \mathbb{T}f = \hat{\mathcal{P}}\mathbb{T}l \circ e_{L_2} \circ \mathbb{T}s_2$, which, in turn, is a consequence of (i) of Lemma 67 and of Remark 48. Also, if $\phi = \bigwedge \Phi$, the fact that (4) holds follows from the induction hypothesis using the definitions of $\Gamma\models_1$ and $\Gamma\models_2$ on formulae containing infinitary conjunctions.

Proposition 69 *\mathbb{S}_Γ preserves expressiveness w.r.t. Γ .*

PROOF. We begin by showing that, if $\models \subseteq X \times L$ is adequate for $R \subseteq X \times X$, then $\Gamma\models \subseteq \mathbb{T}X \times \mathbb{T}L$ is adequate for $\Gamma R \subseteq \mathbb{T}X \times \mathbb{T}X$ (and hence so is $\mathbb{S}_\Gamma\models$). The adequacy of \models for R translates to $\models \circ R \subseteq \models$. The preservation of inclusions by \mathbb{T} and of relational composition by Γ then give $(\Gamma\models) \circ (\Gamma R) \subseteq (\Gamma\models)$. That is, $\Gamma\models$ is adequate for ΓR .

Now assume that \models is expressive for R , i.e. $R = \geq_L$. Following [1], we define

$\theta : X \rightarrow L$ by $\theta(x) = \bigwedge_{\varphi \in L, x \models \varphi} \varphi$. Then:

$$y R x \text{ iff } y \geq_L x \text{ iff } y \models \theta(x) \text{ iff } y (\text{Gr}(\theta)^{\text{op}} \circ \models) x \quad (5)$$

The definition of θ also gives $\text{Gr}(\theta) \subseteq \models$, and hence:

$$\text{Gr}(\top\theta) \subseteq (\Gamma =_L) \circ \text{Gr}(\top\theta) = \Gamma(\text{Gr}(\theta)) \subseteq (\Gamma \models) \quad (6)$$

The first inclusion follows from the definition of a relator (Definition 12), the subsequent equality follows by Lemma 23, and the final inclusion follows from the preservation of inclusions by \top and Γ . We then have:

$$\Gamma R = (\Gamma(\text{Gr}(\theta)^{\text{op}})) \circ (\Gamma \models) = \text{Gr}(\top\theta)^{\text{op}} \circ (\Gamma \models) \supseteq \geq_{\top L}$$

The first equality follows from (5) using the preservation of relational composition by Γ , while the second equality follows by Remark 25. To prove the containment relation, let $v \geq_{\top L} u$. By (6), $u (\Gamma \models) (\top\theta)(u)$, and hence also $v (\Gamma \models) (\top\theta)(u)$. This, together with $(\top\theta)(u) \text{Gr}(\top\theta)^{\text{op}} u$ now yields $v (\text{Gr}(\top\theta)^{\text{op}} \circ (\Gamma \models)) u$. We have thus proved that $\Gamma R \supseteq \geq_{\top L}$. Hence, $\Gamma \models$ is expressive for ΓR (and therefore so is $\mathbb{S}_{\Gamma \models}$).

If the final sequence of Γ stabilises, then the logic induced by \mathbb{S}_{Γ} and Γ coincides with the fragment of the logic in [1] considered in Definition 64. Moreover, by applying Corollary 60, we obtain an alternative proof of the expressiveness w.r.t. Γ -similarity of this logic.

Theorem 70 *The logic induced by \mathbb{S}_{Γ} and Γ characterises Γ -simulation.*

5.4 A Logic for Probabilistic Simulation

We now define a language constructor and associated semantics for probabilistic transition systems, and prove that this semantics preserves expressiveness w.r.t. Γ_P (see Definition 27). As a result, we obtain a logic which characterises simulation on unlabelled probabilistic transition systems.

We let $\Sigma_{\mathbb{B}} = \{\text{tt}, \wedge, \vee\}$, and let $\Gamma_P : \text{Rel} \rightarrow \text{Rel}$ be as in Definition 27. A language constructor and associated \mathcal{S}_{ω} -semantics can be defined as follows.

Definition 71 (\mathbb{S}_P and \mathbb{S}_P) *The language constructor $\mathbb{S}_P : \text{Alg}(\Sigma_{\mathbb{B}}) \rightarrow \text{Alg}(\Sigma_{\mathbb{B}})$ takes a $\Sigma_{\mathbb{B}}$ -algebra L to the free $\Sigma_{\mathbb{B}}$ -algebra over the set $\{\diamond_p \varphi \mid p \in \mathbb{Q} \cap [0, 1], \varphi \in L\}$. The \mathcal{S}_{ω} -semantics $\mathbb{S}_P : \text{Int}_{\mathbb{B}} \rightarrow \text{Int}_{\mathbb{B}}$ for \mathbb{S}_P takes an interpretation $d : L \rightarrow \mathcal{P}X$ to the interpretation $d' : \mathbb{S}_P(L) \rightarrow \mathcal{P}\mathcal{S}_{\omega}X$ defined by:*

$$d'(\diamond_p \varphi) = \{ \mu \in \mathcal{S}_{\omega}X \mid \mu[d(\varphi)] \geq p \} \text{ for } \varphi \in L.$$

Thus, a formula of the form $\diamond_p \varphi$ holds for a finite sub-probability distribution μ if a state satisfying φ is reached via μ with probability at least p . The action of \mathbb{S}_P on arrows is completely defined by the actions of \mathbb{S}_P and \mathcal{S}_ω on arrows. Moreover, Remark 48 can be used to show that \mathbb{S}_P is well-defined.

Now recall from Proposition 30 that the \mathcal{S}_ω -relator Γ_P preserves monics and is ω -accessible. Hence, by Proposition 10, its final sequence stabilises at some α . We will use Corollary 62 to show that α is at most $\omega + \omega$. We first note that the endofunctors \mathcal{S}_ω and \mathbb{S}_P are ω -accessible. Next, we show that the remaining hypothesis of Corollary 62 is satisfied.

Proposition 72 \mathbb{S}_P preserves expressiveness w.r.t. Γ_P .

PROOF. Let $d : L \rightarrow \mathcal{P}X$ be an interpretation, and $R \subseteq X \times X$ be a preorder on X . First, assume that d is adequate for R . We immediately infer that $d(\varphi)$ is R^{op} -closed for any $\varphi \in L$. To show that $\mathbb{S}_P(d) : \mathbb{S}_P(L) \rightarrow \mathcal{P}\mathcal{S}_\omega X$ is adequate for $\Gamma_P R \subseteq \mathcal{S}_\omega X \times \mathcal{S}_\omega X$, let $\mu, \nu \in \mathcal{S}_\omega X$ be such that $\mu (\Gamma_P R) \nu$. The proof of the fact that $\nu \in \mathbb{S}_P(d)(\phi)$ implies $\mu \in \mathbb{S}_P(d)(\phi)$ for all $\phi \in \mathbb{S}_P(L)$ (and hence $\mu \geq_L \nu$) is by induction on ϕ . The non-trivial case is when ϕ is of the form $\diamond_p \varphi$ with $\varphi \in L$. In this case, $\nu \in \mathbb{S}_P(d)(\phi)$ translates to $\nu[d(\varphi)] \geq p$. Also, since $d(\varphi)$ is R^{op} -closed, it follows that $\mu[d(\varphi)] \geq \nu[d(\varphi)]$. Hence, $\mu[d(\varphi)] \geq p$, that is, $\mu \in \mathbb{S}_P(d)(\phi)$.

Now assume that d is expressive for R . To show that $\mathbb{S}_P(d)$ is expressive for $\Gamma_P R$, we must prove that $\mu[Y] \geq \nu[Y]$ for any R^{op} -closed $Y \subseteq X$, whenever $\mu, \nu \in \mathcal{S}_\omega X$ are such that $\mu \geq_{\mathbb{S}_P(L)} \nu$. We can assume that $Y \neq \emptyset$ (otherwise $\mu[Y] = \nu[Y] = 0$ and we are done). We note that, for any R^{op} -closed $\emptyset \neq Y \subseteq X$, $Y = \bigcup_{y \in Y} \bigcap_{y \in d(\varphi)} d(\varphi)$: the left-to-right inclusion is immediate, whereas the right-to-left inclusion follows from the expressiveness of d for R together with Y being R^{op} -closed. Thus, if both Y and the sets $\{\varphi \mid \varphi \in L, y \in d(\varphi)\}$ with $y \in Y$ are finite, the formulae $\diamond_p \varphi_Y$, with $p \in \mathbb{Q} \cap [0, \nu[Y]]$ and $\varphi_Y = \bigvee_{y \in Y} \bigwedge_{y \in d(\varphi)} \varphi$ can be used to show that $\mu[Y] \geq \nu[Y]$. For, $\nu \in \mathbb{S}_P(d)(\diamond_p \varphi_Y)$ yields $\mu \in \mathbb{S}_P(d)(\diamond_p \varphi_Y)$ for any $p \in \mathbb{Q} \cap [0, \nu[Y]]$. That is, $\mu[Y] = \mu[d(\varphi_Y)] \geq p$ for any $p \in \mathbb{Q} \cap [0, \nu[Y]]$. This, in turn, gives $\mu[Y] \geq \nu[Y]$.

However, the previously-mentioned sets are not, in general, finite. Nevertheless, it is possible to define a formula $\varphi \in L$ with the property that $\mu[Y] = \mu[d(\varphi)]$ and $\nu[Y] = \nu[d(\varphi)]$. Then, the above reasoning can be applied to the formulae $\diamond_p \varphi$ with $p \in \mathbb{Q} \cap [0, \nu[Y]]$. In order to define φ , let $Z = \text{supp}(\mu) \cup \text{supp}(\nu)$, and let $\equiv \subseteq \mathcal{L} \times \mathcal{L}$ denote the equivalence relation on \mathcal{L} given by $\varphi_1 \equiv \varphi_2$ if and only if $d(\varphi_1) \cap Z = d(\varphi_2) \cap Z$. Since Z is finite, there are only finitely-many equivalence classes w.r.t. \equiv . For $y \in Y$, let $\Phi_y = \{\varphi \in \mathcal{L} \mid y \in d(\varphi)\}$, and let $\Phi_y^0 \subseteq \Phi_y$ consist of a set of rep-

representatives for Φ_y . Then, for $z \in Z$, $z \in d(\varphi)$ for all $\varphi \in \Phi_y$ if and only if $z \in d(\varphi)$ for all $\varphi \in \Phi_y^0$. Now let $\Phi = \{ \bigwedge_{\varphi \in \Phi_y^0} \varphi \mid y \in Y \}$, and let $\Phi^0 \subseteq \Phi$ consists of a set of representatives for Φ . Then, for $z \in Z$, $z \in d(\phi)$ for some $\phi \in \Phi$ if and only if $z \in d(\phi)$ for some $\phi \in \Phi^0$. One can therefore infer that, for $z \in Z$, $z \in Y$ if and only if $z \in d(\bigvee_{\phi \in \Phi^0} \phi)$. This, in turn, gives $\mu[Y] = \mu[d(\bigvee_{\phi \in \Phi^0} \phi)]$ and $\nu[Y] = \nu[d(\bigvee_{\phi \in \Phi^0} \phi)]$. Then, $\mu \geq_{\mathbb{S}_P(L)} \nu$ together with $\nu \in \mathbb{S}_P(d)(\diamond_p \bigvee_{\phi \in \Phi^0} \phi)$ gives $\mu \in \mathbb{S}_P(d)(\diamond_p \bigvee_{\phi \in \Phi^0} \phi)$, or equivalently $\mu[Y] \geq p$, for all $p \in \mathbb{Q} \cap [0, \nu[Y]]$. Hence, $\mu[Y] \geq \nu[Y]$. We have therefore proved that $\mathbb{S}_P(d)$ is expressive for $\Gamma_P R$.

Corollary 62 can now be applied to infer that the final sequence of Γ_P stabilises at $\omega + \omega$ (or earlier). We can also apply Corollary 60 to obtain:

Theorem 73 *The logic induced by \mathbb{S}_P and Γ_P characterises Γ_P -simulation.*

The syntax of the induced logic can alternatively be defined by:

$$\varphi ::= \mathbf{tt} \mid \diamond_p \varphi \mid \varphi \wedge \psi \mid \varphi \vee \psi$$

while the coalgebraic semantics of the logic is given by:

$$c \models_{\gamma} \diamond_p \varphi \quad \text{iff} \quad \gamma(c)[[\varphi]_{\gamma}] \geq p$$

with $[[\varphi]_{\gamma}] = \{c \in C \mid c \models_{\gamma} \varphi\}$ (plus the usual clauses for \mathbf{tt} , \wedge and \vee). This logic coincides with the unlabelled version of the logic considered in [9]. We have therefore obtained an alternative, inductive proof of the expressiveness w.r.t. similarity of that logic (in the unlabelled case).

6 Compositionality of Logics for Simulation

In this section, we show that expressive logics for simulation can be derived in a compositional fashion. Specifically, we show that language constructors and associated semantics for combinations of coalgebraic types can be derived from language constructors and corresponding semantics for the types being combined, and moreover, that the expressiveness of \mathbb{T} -semantics w.r.t. \mathbb{T} -relators is preserved by these constructions. As a result, we are able to derive, in a modular fashion, expressive logics for the notions of simulation defined in Section 4.

We begin by defining several ways to combine language constructors and their associated semantics. (Variations of these definitions have appeared in [7,5].)

Definition 74 (Operations on language constructors) *Given two language constructors $\mathbb{S}_1, \mathbb{S}_2 : \text{Alg}(\Sigma_{\mathbb{B}}) \rightarrow \text{Alg}(\Sigma_{\mathbb{B}})$, define $\mathbb{S}_1 \oplus \mathbb{S}_2, \mathbb{S}_1 \otimes \mathbb{S}_2, (\mathbb{S}_1)^A : \text{Alg}(\Sigma_{\mathbb{B}}) \rightarrow \text{Alg}(\Sigma_{\mathbb{B}})$ by:*

- $L \xrightarrow{\mathbb{S}_1 \oplus \mathbb{S}_2} \mathbb{S}_1 L \oplus \mathbb{S}_2 L$
- $L \xrightarrow{\mathbb{S}_1 \otimes \mathbb{S}_2} \mathbb{S}_1 L \otimes \mathbb{S}_2 L$
- $L \xrightarrow{(\mathbb{S}_1)^A} (\mathbb{S}_1 L)^A$

where, for $\Sigma_{\mathbb{B}}$ -algebras L_1 and L_2 , the $\Sigma_{\mathbb{B}}$ -algebras $L_1 \oplus L_2$, $L_1 \otimes L_2$ and $(L_1)^A$ are the free $\Sigma_{\mathbb{B}}$ -algebras over the sets $\{\langle \kappa_i \rangle \varphi_i \mid \varphi_i \in L_i, i \in \{1, 2\}\}$, $\{[\pi_i] \varphi_i \mid \varphi_i \in L_i, i \in \{1, 2\}\}$ and $\{[a] \varphi_1 \mid \varphi_1 \in L_1, a \in A\}$, respectively⁷.

Another way to combine two language constructors is to simply compose them (using functor composition).

Proposition 75 $\mathbb{S}_1 \circ \mathbb{S}_2, \mathbb{S}_1 \oplus \mathbb{S}_2, \mathbb{S}_1 \otimes \mathbb{S}_2, (\mathbb{S}_1)^A$ are language constructors.

PROOF (Sketch). All four operations on language constructors preserve accessibility.

Our aim is to derive logics for $\mathbb{T}_1 \circ \mathbb{T}_2$ -, $\mathbb{T}_1 \oplus \mathbb{T}_2$ -, $\mathbb{T}_1 \otimes \mathbb{T}_2$ - and $(\mathbb{T}_1)^A$ -coalgebras, from language constructors \mathbb{S}_1 and \mathbb{S}_2 and associated \mathbb{T}_1 - and \mathbb{T}_2 -semantics. This motivates the following definition.

Definition 76 (Operations on semantics for language constructors) *Let $\mathbb{T}_1, \mathbb{T}_2 : \text{Set} \rightarrow \text{Set}$, let $\mathbb{S}_1, \mathbb{S}_2 : \text{Alg}(\Sigma_{\mathbb{B}}) \rightarrow \text{Alg}(\Sigma_{\mathbb{B}})$ be language constructors, and let \mathbb{S}_i be a \mathbb{T}_i -semantics for \mathbb{S}_i , with $i = 1, 2$. Define $\mathbb{S}_1 \oplus \mathbb{S}_2, \mathbb{S}_1 \otimes \mathbb{S}_2, (\mathbb{S}_1)^A : \text{Int}_{\mathbb{B}} \rightarrow \text{Int}_{\mathbb{B}}$ by:*

- $d : L \rightarrow \mathcal{P}X \xrightarrow{\mathbb{S}_1 \oplus \mathbb{S}_2} \mathbb{S}_1 d \oplus \mathbb{S}_2 d : (\mathbb{S}_1 \oplus \mathbb{S}_2)L \rightarrow \mathcal{P}(\mathbb{T}_1 + \mathbb{T}_2)X$
- $d : L \rightarrow \mathcal{P}X \xrightarrow{\mathbb{S}_1 \otimes \mathbb{S}_2} \mathbb{S}_1 d \otimes \mathbb{S}_2 d : (\mathbb{S}_1 \otimes \mathbb{S}_2)L \rightarrow \mathcal{P}(\mathbb{T}_1 \times \mathbb{T}_2)X$
- $d : L \rightarrow \mathcal{P}X \xrightarrow{(\mathbb{S}_1)^A} (\mathbb{S}_1 d)^A : (\mathbb{S}_1)^A L \rightarrow \mathcal{P}((\mathbb{T}_1 X)^A)$

where, for interpretations $d_i : L_i \rightarrow \mathcal{P}X_i$ with $i = 1, 2$, the interpretations $d_1 \oplus d_2 : L_1 \oplus L_2 \rightarrow \mathcal{P}(X_1 + X_2)$, $d_1 \otimes d_2 : L_1 \otimes L_2 \rightarrow \mathcal{P}(X_1 \times X_2)$ and $(d_1)^A : (L_1)^A \rightarrow \mathcal{P}(X_1^A)$ are defined by:

- $\langle \kappa_i \rangle \varphi_i \xrightarrow{d_1 \oplus d_2} \{\iota_i(x_i) \mid x_i \in d_i(\varphi_i)\}$ for $i = 1, 2$
- $[\pi_i] \varphi_i \xrightarrow{d_1 \otimes d_2} \{(x_1, x_2) \mid x_i \in d_i(\varphi_i)\}$ for $i = 1, 2$

⁷ Taking free $\Sigma_{\mathbb{B}}$ -algebras amounts to closing under the logical operators specified by $\Sigma_{\mathbb{B}}$.

- $[a]\varphi_1 \xrightarrow{(d_1)^A} \{ (x_a)_{a \in A} \mid x_a \in d_1(\varphi_1) \}$ for $a \in A$.

(Note that the actions of $d_1 \oplus d_2$, $d_1 \otimes d_2$ and $(d_1)^A$ on formulae containing logical connectives is uniquely defined by the requirement that interpretations preserve the $\Sigma_{\mathbf{B}}$ -structure.)

Again, we can also combine \mathbf{T} -semantics by simply composing them.

Proposition 77 $\mathbb{S}_1 \circ \mathbb{S}_2$, $\mathbb{S}_1 \oplus \mathbb{S}_2$, $\mathbb{S}_1 \otimes \mathbb{S}_2$ and $(\mathbb{S}_1)^A$ are $\mathbf{T}_1 \circ \mathbf{T}_2$ -, $\mathbf{T}_1 + \mathbf{T}_2$ -, $\mathbf{T}_1 \times \mathbf{T}_2$ - and $(\mathbf{T}_1)^A$ -semantics for $\mathbb{S}_1 \circ \mathbb{S}_2$, $\mathbb{S}_1 \oplus \mathbb{S}_2$, $\mathbb{S}_1 \otimes \mathbb{S}_2$ and $(\mathbb{S}_1)^A$, respectively.

PROOF. Immediate from the definitions.

We note that the modal operators $\langle \kappa_i \rangle$, $[\pi_i]$ and $[a]$ (and their associated semantics) are similar to the modal operators used in [16] in the context of polynomial endofunctors.

The next result shows that the expressivity condition required to derive expressive logics for simulation (Definition 54) is preserved by the previously-defined operations.

Proposition 78 If \mathbb{S}_i preserves expressiveness w.r.t. Γ_i , for $i = 1, 2$, then $\mathbb{S}_1 \circ \mathbb{S}_2$, $\mathbb{S}_1 \oplus \mathbb{S}_2$, $\mathbb{S}_1 \otimes \mathbb{S}_2$ and $(\mathbb{S}_1)^A$ preserve expressiveness w.r.t. $\Gamma_1 \circ \Gamma_2$, $\Gamma_1 \oplus \Gamma_2$, $\Gamma_1 \otimes \Gamma_2$ and $(\Gamma_1)^A$, respectively.

PROOF (Sketch). In the case of $\mathbb{S}_1 \circ \mathbb{S}_2$, the statement follows immediately from the definition of preservation of expressiveness w.r.t. a relator. Of the remaining cases, we only consider that of coproducts. (The other two can be treated similarly.) Let $d : L \rightarrow \mathcal{P}X$ be expressive for $R \subseteq X \times X$. Hence, $\mathbb{S}_i d : \mathbb{S}_i L \rightarrow \mathcal{P}\mathbf{T}_i X$ is expressive for $\Gamma_i R \subseteq \mathbf{T}_i X \times \mathbf{T}_i X$. Now let $i, j \in \{1, 2\}$, $t_i \in \mathbf{T}_i X$ and $s_j \in \mathbf{T}_j X$ be such that $(\iota_i(t_i), \iota_j(s_j)) \notin (\Gamma_1 R \oplus \Gamma_2 R)$. If $i \neq j$, this is witnessed by the formula $\langle \kappa_j \rangle \mathbf{tt}$, which holds in $\iota_j(s_j)$ but not in $\iota_i(t_i)$. If $i = j$, this is witnessed by the formula $\langle \kappa_j \rangle \varphi_j$, where φ_j holds in s_j but not in t_j .

It is also possible to define language constructors and associated semantics for constant and identity functors. In the case of the constant functor $X \mapsto A$, the language constructor provides an atomic formula a for each $a \in A$, whereas the associated A -semantics takes any interpretation to the interpretation mapping a to $\{a\}$ for $a \in A$. In the case of the identity functor, both the language constructor and its associated semantics are identity functors (on $\text{Alg}(\Sigma_{\mathbf{B}})$)

and $\text{Int}_{\mathbb{B}}$, respectively). In both cases, the associated semantics preserves expressiveness w.r.t. the corresponding minimal relator. (Note that this holds independently of the choice of $\Sigma_{\mathbb{B}}$.) As a result, the compositional techniques described in this section can be applied to any endofunctor $\mathbb{T} : \text{Set} \rightarrow \text{Set}$ generated by the syntax:

$$\mathbb{T} ::= A \mid \text{Id} \mid \mathcal{P}_{\omega} \mid \mathcal{S}_{\omega} \mid \mathbb{T}_1 \circ \mathbb{T}_2 \mid \mathbb{T}_1 + \mathbb{T}_2 \mid \mathbb{T}_1 \times \mathbb{T}_2 \mid \mathbb{T}^A$$

in order to derive both a notion of simulation for \mathbb{T} -coalgebras, and a characterising logic for it. We therefore obtain notions of simulation and characterising logics for a large class of non-deterministic and probabilistic system types. These include image-finite labelled transition systems ($\mathbb{T} = (\mathcal{P}_{\omega})^A$), *reactive* ($\mathbb{T} = (\mathcal{S}_{\omega})^A$), *generative* ($\mathbb{T} = \mathcal{S}_{\omega} \circ (A \times \text{Id})$) and *stratified* ($\mathbb{T} = \mathcal{S}_{\omega} + (A \times \text{Id})$) models of probabilistic systems [30], Hansson's *alternating probabilistic systems* ($\mathbb{T} = \mathcal{S}_{\omega} + (\mathcal{P}_{\omega})^A$) [11], Segala's *simple* ($\mathbb{T} = (\mathcal{P}_{\omega} \circ \mathcal{S}_{\omega})^A$) and *general* ($\mathbb{T} = \mathcal{P}_{\omega} \circ \mathcal{S}_{\omega} \circ (A \times \text{Id})$) *probabilistic automata* [27], and several other types of non-deterministic and probabilistic models. (See [2] for a comprehensive survey of such models from a coalgebraic perspective.) The remainder of this section describes some of the logics obtained using our approach.

Example 79 *Let $\Sigma_{\mathbb{B}} = \{\text{tt}, \wedge\}$, and let $\Gamma_{\supseteq} : \text{Rel} \rightarrow \text{Rel}$, $\mathbb{S}_{\supseteq} : \text{Alg}(\Sigma_{\mathbb{B}}) \rightarrow \text{Alg}(\Sigma_{\mathbb{B}})$ and $\mathbb{S}_{\supseteq} : \text{Int}_{\mathbb{B}} \rightarrow \text{Int}_{\mathbb{B}}$ be as in Examples 15, 46 and 52, respectively. It then follows from Example 55 and Proposition 78 that $(\mathbb{S}_{\supseteq})^A$ preserves expressiveness w.r.t. $(\Gamma_{\supseteq})^A$. Also, since the initial sequence of $(\mathbb{S}_{\supseteq})^A$ stabilises at ω , the logic induced by $(\mathbb{S}_{\supseteq})^A$ and $(\Gamma_{\supseteq})^A$ has syntax $\mathcal{L} = \mathcal{L}((\mathbb{S}_{\supseteq})^A)$ given by:*

$$\begin{aligned} \mathcal{L} \ni \varphi &::= \text{tt} \mid [a]\psi \mid \varphi_1 \wedge \varphi_2 & (\psi \in \mathcal{L}_0) \\ \mathcal{L}_0 \ni \psi &::= \text{tt} \mid \diamond\varphi \mid \psi_1 \wedge \psi_2 & (\varphi \in \mathcal{L}) \end{aligned}$$

and coalgebraic semantics defined inductively by:

$$\begin{aligned} c \models_{\gamma} \varphi &\text{ iff } \gamma(c) \models \varphi & (c \in C) \\ f \models [a]\psi &\text{ iff } f(a) \models_0 \psi & (f \in (\mathcal{P}_{\omega}C)^A) \\ X \models_0 \diamond\varphi &\text{ iff } \exists c \in X . c \models_{\gamma} \varphi & (X \in \mathcal{P}_{\omega}C) \end{aligned}$$

with (C, γ) a $(\mathcal{P}_{\omega})^A$ -coalgebra. Note the use of an intermediary $\Sigma_{\mathbb{B}}$ -algebra \mathcal{L}_0 and of two intermediary relations $\models \subseteq (\mathcal{P}_{\omega}C)^A \times \mathcal{L}$ and $\models_0 \subseteq \mathcal{P}_{\omega}C \times \mathcal{L}_0$ in defining the syntax and semantics of the logic. These intermediary steps correspond to the steps used in deriving the underlying language constructor and its associated $(\mathcal{P}_{\omega})^A$ -semantics. We also note that the modal operator $[a]$ distributes (semantically) over tt and \wedge ; as a result, the induced logic is equivalent to a fragment of Hennessy-Milner logic, with $[a]\text{tt}$ being semantically equivalent to tt , $[a](\psi_1 \wedge \psi_2)$ being semantically equivalent to $[a]\psi'_1 \wedge [a]\psi'_2$, and $[a]\diamond\varphi$ being semantically equivalent to $\langle a \rangle \varphi'$, whenever ψ_1, ψ_2, φ are semantically equivalent to $\psi'_1, \psi'_2, \varphi'$, respectively. This logic characterises $(\Gamma_{\supseteq})^A$.

simulation, i.e. the standard notion of simulation on labelled transition systems.

The notions of ready and complete simulation between labelled transition systems (see Examples 35 and 36) also admit characterising logics. However, since complete simulation can not be derived compositionally, nor can its characterising logic.

Example 80 Let $\Sigma_B = \{\text{tt}, \wedge\}$, and let $\Gamma_{\geq}^R : \text{Rel} \rightarrow \text{Rel}$ be as in Example 35. Also, let $S_{\geq}^R : \text{Alg}(\Sigma_B) \rightarrow \text{Alg}(\Sigma_B)$ denote the language constructor which takes a Σ_B -algebra L to the free Σ_B -algebra over the set $\{\Delta\} \cup \{\diamond\varphi \mid \varphi \in L\}$. Finally, let $S_{\geq}^R : \text{Int}_B \rightarrow \text{Int}_B$ denote the \mathcal{P}_ω -semantics for S_{\geq}^R which takes an interpretation $d : L \rightarrow \mathcal{P}X$ to the interpretation $d' : S_{\geq}^R L \rightarrow \mathcal{P}(\mathcal{P}_\omega X)$ given by:

$$d'(\Delta) = \{\emptyset\} \quad d'(\diamond\varphi) = \{Y \in \mathcal{P}_\omega X \mid Y \cap d(\varphi) \neq \emptyset\}$$

Then, S_{\geq}^R preserves expressiveness w.r.t. Γ_{\geq}^R . To see this, let $d : L \rightarrow \mathcal{P}X$ be expressive for $R \subseteq X \times X$, and let $Y, Z \in \mathcal{P}_\omega X$. The fact that $Y (\Gamma_{\geq}^R R) Z$ iff $Y \geq_{S_{\geq}^R(L)} Z$ follows from Example 55, together with the observation that $(Z = \emptyset \Rightarrow Y = \emptyset)$ is equivalent to $(\Delta \in s'(Z) \Rightarrow \Delta \in s'(Y))$ (where $s' : \mathcal{P}_\omega X \rightarrow \mathcal{P}(S_{\geq}^R L)$ is the logical map induced by $d' : S_{\geq}^R L \rightarrow \mathcal{P}(\mathcal{P}_\omega X)$). Hence, by Proposition 78, $(S_{\geq}^R)^A$ preserves expressiveness w.r.t. $(\Gamma_{\geq}^R)^A$. This, in turn, yields a logic which characterises ready simulation. The syntax of this logic extends $\mathcal{L}((S_{\geq}^R)^A)$ with formulae of the form $[a]\Delta\varphi$, which hold in a state c of a $(\mathcal{P}_\omega)^A$ -coalgebra (C, γ) precisely when $\gamma(c)(a) = \emptyset$. The logic thus obtained differs from the logic typically used to characterise ready simulation equivalence (see e.g. [29]), where formulae of the form B , with $B \subseteq A$, are used to formalise the ability of states in an A -labelled transition system to perform exactly the actions in B .

Example 81 Let $\Sigma_B = \{\text{tt}, \wedge\}$, and let $\Gamma_{\geq}^C : \text{Rel} \rightarrow \text{Rel}$ be as in Example 36. Also, let $S_{\geq}^C : \text{Alg}(\Sigma_B) \rightarrow \text{Alg}(\Sigma_B)$ denote the language constructor which takes a Σ_B -algebra L to the free Σ_B -algebra over the set $\{\Delta\} \cup \{\langle a \rangle\varphi \mid a \in A, \varphi \in L\}$ (with A some fixed set of labels). Finally, let $S_{\geq}^C : \text{Int}_B \rightarrow \text{Int}_B$ denote the $(\mathcal{P}_\omega)^A$ -semantics for S_{\geq}^C which takes an interpretation $d : L \rightarrow (\mathcal{P}X)^A$ to the interpretation $d' : S_{\geq}^C L \rightarrow \mathcal{P}((\mathcal{P}_\omega X)^A)$ given by:

$$\begin{aligned} d'(\Delta) &= \{f : A \rightarrow \mathcal{P}_\omega X \mid f(a) = \emptyset \text{ for all } a \in A\} \\ d'(\langle a \rangle\varphi) &= \{f : A \rightarrow \mathcal{P}_\omega X \mid f(a) \cap d(\varphi) \neq \emptyset\} \end{aligned}$$

It is easy to show that S_{\geq}^C preserves expressiveness w.r.t. Γ_{\geq}^C . The logic induced by S_{\geq}^C and Γ_{\geq}^C characterises complete simulation on labelled transition systems.

We now turn to coalgebraic types which involve probabilistic features. Specifically, we consider labelled probabilistic transition systems ($\mathbb{T} = (\mathcal{S}_\omega)^A$) and simple probabilistic automata ($\mathbb{T} = (\mathcal{P}_\omega \circ \mathcal{S}_\omega)^A$).

Example 82 Let $\Sigma_{\mathbb{B}} = \{\mathbf{tt}, \wedge, \vee\}$, and let $\mathbb{S}_P : \mathbf{Int}_{\mathbb{B}} \rightarrow \mathbf{Int}_{\mathbb{B}}$ be as in Definition 71. It then follows by Propositions 72 and 78 that $(\mathbb{S}_P)^A$ preserves expressiveness w.r.t. $(\Gamma_P)^A$. The logic induced by $(\mathbb{S}_P)^A$ and $(\Gamma_P)^A$ has syntax $\mathcal{L} = \mathcal{L}((\mathbb{S}_P)^A)$ given by:

$$\begin{aligned} \mathcal{L} \ni \varphi &::= \mathbf{tt} \mid [a]\psi \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \vee \varphi_2 & (\psi \in \mathcal{L}_0) \\ \mathcal{L}_0 \ni \psi &::= \mathbf{tt} \mid \diamond_p \varphi \mid \psi_1 \wedge \psi_2 \mid \psi_1 \vee \psi_2 & (\varphi \in \mathcal{L}) \end{aligned}$$

and coalgebraic semantics defined inductively by:

$$\begin{aligned} c \models_{\gamma} \varphi &\text{ iff } \gamma(c) \models \varphi & (c \in C) \\ f \models [a]\psi &\text{ iff } f(a) \models_0 \psi & (f \in (\mathcal{S}_{\omega}C)^A) \\ \mu \models_0 \diamond_p \varphi &\text{ iff } \mu[\llbracket \varphi \rrbracket_{\gamma}] \geq p & (\mu \in \mathcal{S}_{\omega}C) \end{aligned}$$

with (C, γ) an $(\mathcal{S}_{\omega})^A$ -coalgebra. It then follows by Corollary 60 that this logic characterises probabilistic simulation. Moreover, this logic is equivalent to the logic used in [9], with $[a]\diamond_p \varphi$ being semantically equivalent to $\langle a \rangle_p \varphi'$, whenever φ is semantically equivalent to φ' .

Example 83 A logic which characterises the notion of simulation obtained in Example 38 for simple probabilistic automata can be derived by combining \mathbb{S}_{\supseteq} with \mathbb{S}_P , and \mathbb{S}_{\supseteq} with \mathbb{S}_P . The logic induced by $(\mathbb{S}_{\supseteq} \circ \mathbb{S}_P)^A$ and $(\Gamma_{\supseteq} \circ \Gamma_P)^A$ has syntax $\mathcal{L} = \mathcal{L}((\mathbb{S}_{\supseteq} \circ \mathbb{S}_P)^A)$ given by:

$$\begin{aligned} \mathcal{L} \ni \varphi &::= \mathbf{tt} \mid [a]\psi \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \vee \varphi_2 & (\psi \in \mathcal{L}_0) \\ \mathcal{L}_0 \ni \psi &::= \mathbf{tt} \mid \diamond \xi \mid \psi_1 \wedge \psi_2 \mid \psi_1 \vee \psi_2 & (\xi \in \mathcal{L}_1) \\ \mathcal{L}_1 \ni \xi &::= \mathbf{tt} \mid \diamond_p \varphi \mid \xi_1 \wedge \xi_2 \mid \xi_1 \vee \xi_2 & (\varphi \in \mathcal{L}) \end{aligned}$$

and coalgebraic semantics defined similarly to previous examples. This logic captures the notion of simulation induced by $(\Gamma_{\supseteq} \circ \Gamma_P)^A$ (see Example 38).

We conclude this section by comparing the notion of simulation derived in Example 38 with two existing notions of simulation for simple probabilistic automata, namely *strong simulation* and *strong probabilistic simulation*, as defined in [27] (see also [18]).

Strong simulation [27] is a generalisation of the notion of simulation for probabilistic transition systems, and can be characterised using a multi-sorted logic similar to the one obtained here (with *non-deterministic formulae* being used to formalise properties of states of a probabilistic automaton, and with *probabilistic formulae* being used to formalise properties of probability distributions over these states) [18]. Moreover, it follows easily that the logic obtained here is equivalent to the one in [18] (with formulae of the form $[a]\diamond \xi$ in our logic corresponding to non-deterministic formulae of the form $\langle a \rangle \xi'$ in [18]). This equivalence between the two logics, together with the fact these logics char-

acterise $(\Gamma_{\supseteq} \circ \Gamma_P)^A$ -simulation and strong simulation, respectively, results in $(\Gamma_{\supseteq} \circ \Gamma_P)^A$ -simulation being the same as strong simulation.

The definition of strong probabilistic simulation [27] uses *combined transitions* to weaken the requirements in the definition of strong simulation. A combined transition $s \xrightarrow{a}_C \mu$ involves a convex combination μ of a set $\{\mu_i \mid s \xrightarrow{a} \mu_i\}$ of probability distributions, that is, $\mu = \sum_i \lambda_i \mu_i$, with $\sum_i \lambda_i = 1$. Then, the definition of strong probabilistic simulation requires that any transition $s \xrightarrow{a} \mu$ in one probabilistic automaton is matched by a combined transition $t \xrightarrow{a}_C \nu$ in the other automaton (subject to some additional constraints on μ and ν). Strong probabilistic simulation can also be defined in our setting, as shown in the following example.

Example 84 Let $\Gamma_{PA} : \text{Rel} \rightarrow \text{Rel}$ be the $\mathcal{P}_\omega \circ \mathcal{S}_\omega$ -relator defined by:

$$S \Gamma_{PA}(R) T \text{ iff } \forall \nu \in \mathcal{S}. \exists \{\mu_i \mid i \in I\} \subseteq T. \exists \{\lambda_i \mid i \in I\} \subseteq [0, 1]. \sum_i \lambda_i = 1 \\ \text{and } \sum_i \lambda_i \mu_i[X] \geq \nu[Y] \text{ whenever } (\pi_1^R)^{-1}[X] \supseteq (\pi_2^R)^{-1}[Y]$$

for $R \subseteq X \times Y$, $S \in \mathcal{P}_\omega \mathcal{S}_\omega X$ and $T \in \mathcal{P}_\omega \mathcal{S}_\omega Y$. Then, $(\Gamma_{PA})^A$ -simulation coincides with strong probabilistic simulation. Again, this is proved by exhibiting a logic which characterises $(\Gamma_{PA})^A$ -simulation, and then showing that this logic is equivalent to the logic used in [18] to characterise strong probabilistic simulation. The sought logic has the same syntax as the logic in Example 83, but its semantics accounts for the use of combined transitions in defining strong probabilistic simulation. Specifically, the clause:

$$M \models_0 \diamond \xi \text{ iff } \exists \mu \in M. \mu \models_1 \xi$$

is replaced by:

$$M \models_0 \diamond \xi \text{ iff } \exists \{\mu_i \mid i \in I\} \subseteq M. \exists \{\lambda_i \mid i \in I\} \subseteq [0, 1]. \\ \sum_i \lambda_i = 1 \text{ and } \sum_i \lambda_i \mu_i[X] \models_1 \xi$$

where $M \in \mathcal{S}_\omega C$ and (C, γ) is a $\mathcal{P}_\omega \circ \mathcal{S}_\omega$ -coalgebra. The expressiveness of this logic w.r.t. $(\Gamma_{PA})^A$ -simulation is proved by first showing that the $\mathcal{P}_\omega \circ \mathcal{S}_\omega$ -semantics for $\mathcal{S}_{\supseteq} \circ \mathcal{S}_P$ described above preserves expressiveness w.r.t. Γ_{PA} , and subsequently using the modular techniques described at the beginning of this section to move to the labelled case.

7 Summary

We have presented a modular approach to defining notions of simulation, and logics which characterise them, by modelling systems as coalgebras of endofunctors. Our approach was based on the coalgebraic approach to defining notions of simulation [28,1,17], and on an inductive technique for defining logics for coalgebras [5]. We have shown that the expressiveness w.r.t. simulation of an inductively-defined logic for coalgebras can be inferred from an expressivity condition involving one step in the definition of the logic, and the relator inducing that notion of simulation. We have also proposed modular techniques for deriving notions of simulation and associated characterising logics for increasingly complex coalgebraic types.

We have applied these results to derive Baltag's logic for coalgebraic simulation, and to obtain an alternative proof of the expressiveness of this logic w.r.t. simulation. We have also derived notions of simulation and associated characterising logics for several kinds of non-deterministic and probabilistic systems, including (probabilistic) transition systems and probabilistic automata. We have, as a result, obtained both coalgebraic and logical characterisations of several existing notions of simulation, including standard, complete and ready simulation on labelled transition systems, simulation on probabilistic transition systems, and strong (probabilistic) simulation on probabilistic automata.

Our approach applies to a large class of state-based systems, the only requirement on these systems being that they can be modelled as coalgebras. In particular, the techniques described in Sections 4 and 6 can be applied to derive notions of simulation and characterising logics for all coalgebraic types identified in [2] as relevant to the modelling of probabilistic systems. Only two of these types, namely probabilistic transition systems and simple probabilistic automata, have been considered here. It would also be interesting to investigate the notions of simulation and associated logics that are obtained by applying our approach to the alternating probabilistic systems of [11], or the general probabilistic automata of [27].

References

- [1] A. Baltag. A logic for coalgebraic simulation. In H. Reichel, editor, *Coalgebraic Methods in Computer Science*, volume 33 of *Electronic Notes in Theoretical Computer Science*. Elsevier, 2000.
- [2] F. Bartels, A. Sokolova, and E. de Vink. A hierarchy of probabilistic system types. In H.P. Gumm, editor, *Coalgebraic Methods in Computer Science*, volume

82.1 of *Electronic Notes in Theoretical Computer Science*, 2003.

- [3] F. Borceux. *Handbook of Categorical Algebra*, volume II. Cambridge University Press, Cambridge, 1994.
- [4] C. Cirstea. On expressivity and compositionality in logics for coalgebras. In H.P. Gumm, editor, *Coalgebraic Methods in Computer Science*, volume 82.1 of *Electronic Notes in Theoretical Computer Science*. Elsevier, 2003.
- [5] C. Cirstea. A compositional approach to defining logics for coalgebras. *Theoretical Computer Science*, 327:45–69, 2004.
- [6] C. Cirstea. On logics for coalgebraic simulation. In J. Adámek and S. Milius, editors, *Coalgebraic Methods in Computer Science*, volume 106 of *Electronic Notes in Theoretical Computer Science*. Elsevier, 2004.
- [7] C. Cirstea and D. Pattinson. Modular construction of modal logics. In *Proceedings of CONCUR 2004*, volume 3170 of *Lecture Notes in Computer Science*, pages 258–275. Springer, 2004.
- [8] E.P. de Vink and J.J.M.M. Rutten. Bisimulation for probabilistic transition systems: a coalgebraic approach. *Theoretical Computer Science*, 221:271–293, 1999.
- [9] J. Desharnais. Logical characterisation of simulation for Markov chains. In *Proceedings of PROBMIV'99*, pages 33–48. University of Birmingham, 1999.
- [10] E.A. Emerson and E.M. Clarke. Using branching time temporal logic to synthesize synchronization skeletons. *Science of Computer Programming*, 2(3):241–266, 1982.
- [11] H. Hansson. Time and probability in formal design of distributed systems. In *Real-Time Safety Critical Systems*, volume 1. Elsevier, 1994.
- [12] H. Hansson and B. Jonsson. A logic for reasoning about time and reliability. *Formal Aspects of Computing*, 6(5):512–535, 1994.
- [13] C. Hermida and B. Jacobs. Structural induction and coinduction in a fibrational setting. *Information and Computation*, 145(2):107–152, 1998.
- [14] W.H. Hesselink and A. Thijs. Fixpoint semantics and simulation. *Theoretical Computer Science*, 238:275–311, 2000.
- [15] B. Jacobs. *Categorical Logic and Type Theory*. Number 141 in Studies in Logic and the Foundations of Mathematics. North Holland, Amsterdam, 1999.
- [16] B. Jacobs. Many-sorted coalgebraic modal logic: a model-theoretic study. *Theoretical Informatics and Applications*, 35:31–59, 2001.
- [17] B. Jacobs and J. Hughes. Simulations in coalgebra. In H.P. Gumm, editor, *Coalgebraic Methods in Computer Science*, volume 82.1 of *Electronic Notes in Theoretical Computer Science*. Elsevier, 2003.

- [18] B. Jonsson, K.G. Larsen, and W. Yi. Probabilistic extensions of process algebras. In J.A. Bergstra, A. Ponse, and S.A. Smolka, editors, *Handbook of Process Algebra*, chapter 11, pages 685–710. Elsevier, 2001.
- [19] D. Kozen. Results on the propositional mu-calculus. *Theoretical Computer Science*, 27:333–354, 1983.
- [20] K.G. Larsen and A. Skou. Bisimulation through probabilistic testing. *Information and Computation*, 94:1–28, 1991.
- [21] R. Milner. *Communication and Concurrency*. International series in computer science. Prentice Hall, 1989.
- [22] L.S. Moss. Coalgebraic logic. *Annals of Pure and Applied Logic*, 96:277–317, 1999.
- [23] D. Park. Concurrency and automata on infinite sequences. In *Proceedings of 5th GI-Conference on Theoretical Computer Science*, volume 104 of *Lecture Notes in Computer Science*, pages 167–183, 1981.
- [24] A. Pnueli. A temporal logic of concurrent programs. *Theoretical Computer Science*, 13(1):45–60, 1981.
- [25] J.J.M.M. Rutten. Universal coalgebra: a theory of systems. *Theoretical Computer Science*, 249(1):3–80, 2000.
- [26] R. Segala. *Modelling and Verification of Randomized Distributed Real-Time Systems*. PhD thesis, Massachusetts Institute of Technology, 1995.
- [27] R. Segala and N. Lynch. Probabilistic simulations for probabilistic processes. In B. Jonsson and J. Parrow, editors, *Proceedings of CONCUR 94*, volume 836 of *Lecture Notes in Computer Science*, pages 481–496. Springer, 1994.
- [28] A. Thijs. *Simulation and Fixpoint Semantics*. PhD thesis, Rijksuniversiteit Groningen, 1996.
- [29] R.J. van Glabbeek. The linear time – branching time spectrum I; the semantics of concrete, sequential processes. In J.A. Bergstra, A. Ponse, and S.A. Smolka, editors, *Handbook of Process Algebra*, chapter 1, pages 3–99. Elsevier, 2001.
- [30] R.J. van Glabbeek, S.A. Smolka, and B. Steffen. Reactive, generative and stratified models of probabilistic processes. *Information and Computation*, 121(1):59–80, 1995.
- [31] J. Worrell. Terminal sequences for accessible endofunctors. In B. Jacobs and J. Rutten, editors, *Coalgebraic Methods in Computer Science*, volume 19 of *Electronic Notes in Theoretical Computer Science*. Elsevier, 1999.