Math. Struct. in Comp. Science (2016), *vol.* 26, *pp.* 1234–1268. © Cambridge University Press 2014
doi:10.1017/S0960129514000449 First published online 3 December 2014

A coalgebraic view on decorated traces

F. BONCHI[†], M. BONSANGUE[‡,§], G. CALTAIS[§,¶],
J. RUTTEN[§,‖] and A. SILVA[§,‖,††]

[†]*ENS Lyon, Université de Lyon, LIP (UMR 5668 CNRS ENS Lyon UCBL INRIA), Lyon, France*
[‡]*LIACS - Leiden University, Leiden, The Netherlands*
[§]*Centrum voor Wiskunde en Informatica, Amsterdam, The Netherlands*
[¶]*School of Computer Science - Reykjavik University, Reykjavik, Iceland*
[‖]*Radboud University, Nijmegen, The Netherlands*
[††]*HASLab/INESC TEC, Universidade do Minho, Braga, Portugal*

Received 23 April 2014

In the concurrency theory, various semantic equivalences on transition systems are based on traces *decorated* with some additional observations, generally referred to as *decorated traces*. Using the generalized powerset construction, recently introduced by a subset of the authors (Silva *et al.* 2010 FSTTCS. *LIPIcs* **8** 272–283), we give a coalgebraic presentation of decorated trace semantics. The latter include ready, failure, (complete) trace, possible futures, ready trace and failure trace semantics for labelled transition systems, and ready, (maximal) failure and (maximal) trace semantics for generative probabilistic systems. This yields a uniform notion of minimal representatives for the various decorated trace equivalences, in terms of final Moore automata. As a consequence, proofs of decorated trace equivalence can be given by coinduction, using different types of (Moore-) bisimulation (up-to context).

1. Introduction

The study of behavioural equivalence of systems has been a research topic in concurrency for many years now. For different types of systems, several equivalences have been proposed throughout the years, each of which is suitable for use in different contexts of application.

The focus of this paper is on labelled transition systems (LTSs) and GPSs and a suite of corresponding equivalences usually referred to as *decorated trace semantics*. More explicitly, we consider ready, failure, (complete) trace, possible futures, ready trace and failure trace semantics for LTSs, as described in van Glabbeek (2001) and ready, (maximal) failure and (maximal) trace semantics for GPSs, as introduced in Jou and Smolka (1990).

Proof methods for the different equivalences are an important part of this research enterprise. In this paper, we propose *coinduction* as a general proof method for the aforementioned decorated trace semantics of LTSs and GPSs.

Coinduction is a general proof principle which has been uniformly defined in the theory of coalgebras for different types of state-based systems and infinite data types. Given a functor $\mathcal{F}\colon \mathbf{Set} \to \mathbf{Set}$, an $\mathcal{F}$-coalgebra is a pair (X, f) consisting of a set of states X and a function $f : X \to \mathcal{F}(X)$ defining the dynamics of the system. The functor $\mathcal{F}$ determines the type of the transition system or data type under study. For a large class of functors $\mathcal{F}$, there exists a *final coalgebra* into which every $\mathcal{F}$-coalgebra

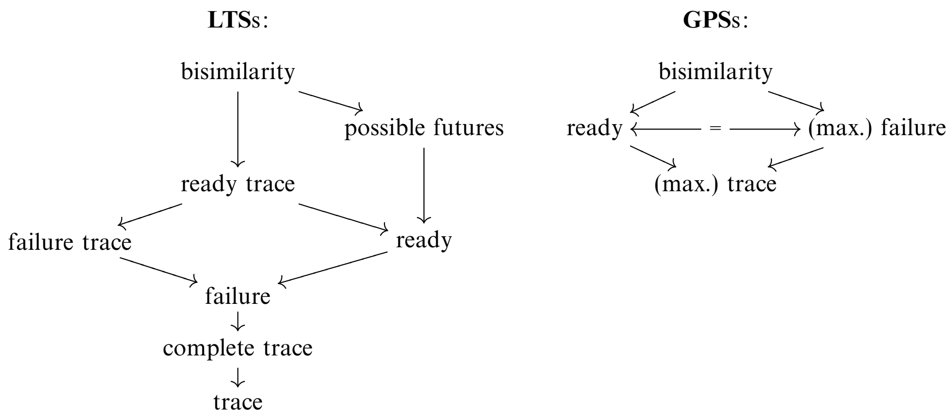


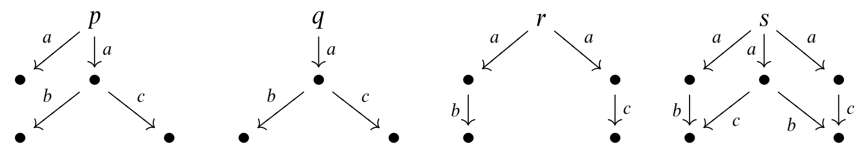
Fig. 1. Lattices of semantic equivalences for LTSs and GPSs.

is mapped by a unique homomorphism. Intuitively, one can see the final coalgebra as the universe of all behaviours of systems and the unique morphism as the map assigning to each system its behaviour. This provides a standard notion of equivalence called $\mathcal{F}$-*behavioural equivalence*. Moreover, these canonical behaviours are minimal, by general coalgebraic considerations (Rutten 2000), in that no two different states are equivalent.

LTSs can be modelled as coalgebras for the functor $\mathcal{L}(X) = (\mathcal{P}_\omega X)^A$ and the canonical behavioural equivalence associated with $\mathcal{L}$ is precisely the finest equivalence of the spectrum in van Glabbeek (2001). Orthogonally, GPSs are coalgebras for $\mathcal{G}(X) = \mathcal{D}_\omega(A \times X)$, where $\mathcal{D}_\omega$ is the (sub)probability functor. The behavioural equivalence associated to $\mathcal{G}$ is the probabilistic bisimilarity equivalence in Jou and Smolka (1990).

In the recent past, other equivalences of the spectrum have been also cast in the coalgebraic framework. Notably, trace semantics of LTSs was widely studied (Hasuo *et al.* 2007; Lenisa 1999; Lenisa *et al.* 2000; Silva *et al.* 2010) and, more recently, decorated trace semantics was recovered in Silva *et al.* (2013) via a coalgebraic generalization of the classical powerset construction (Cancila *et al.* 2003; Lenisa 1999; Silva *et al.* 2010). This paved the way to a coalgebraic modelling of a series of 'twin' semantics in the context of GPSs, which we provide in this paper.

In the right hand side of Figure 1, we illustrate the hierarchy (based on the coarseness level) among bisimilarity, ready, failure, (complete) trace, possible futures, ready trace and failure trace semantics for LTSs, as introduced in van Glabbeek (2001). In the left hand side, a similar hierarchy is depicted for bisimilarity, ready, (maximal) failure and (maximal) trace semantics for GPSs, as in Jou and Smolka (1990). For example, for both types of systems, bisimilarity (the standard behavioural equivalence on $\mathcal{F}$-coalgebras) is the finest of the semantics, whereas trace is the coarsest one. Moreover, note that for the case of GPSs, maximality does not bring more distinguishing power and, ready and failure semantics are equivalent. In order to get some intuition on the type of distinctions the equivalences above encompass, consider the following LTSs:



None of the top states of the systems above are bisimilar. The state p is the only among the four in which an action a can lead to a deadlock state, whereas q, r and s have a different branching structures.

The traces of the states p, q, r and s are $\{a, ab, ac\}$, and therefore they are all trace equivalent. Of the four states above, q and r and s are complete trace equivalent as they can execute the same traces that lead to states where no further action are possible, whereas p is the only state that can trigger a and terminate.

Ready (respectively, failure) semantics identifies states according to the set of actions they can (respectively, fail to) trigger immediately after a certain trace has been executed. None of the states above are ready equivalent; for example, after the execution of action a, process p can reach a deadlock state whereas q has always to choose between actions b and c. Orthogonally, only r and s are failure equivalent.

Possible-futures semantics identifies states that can perform the same traces w and, moreover, the states reached by executing such w's are trace equivalent. None of the states above are possible-futures equivalent. For example, after triggering action a, p can reach a deadlock state (with no further behaviour) whereas q can execute the set of traces $\{b, c\}$.

Ready (respectively failure) trace semantics identifies states that can trigger the same traces w and the (pairwise-taken) intermediate states determined by such w's are ready (respectively refuse) to trigger the same sets of actions. None of the systems above is ready trace equivalent. For example, after performing action a, process q reaches a state that is ready to trigger both b and c, whereas r cannot. The analysis on failure trace equivalence follows a similar reasoning, but different results.

The corresponding semantic equivalences in Figure 1 distinguish between p, q, r and s as summarized in the table below:

	p,q	p,r	p,s	q,r	q,s	r,s
bisimilarity	×	×	×	×	×	×
trace	✓	✓	✓	✓	✓	✓
complete trace	×	×	×	✓	✓	✓
ready	×	×	×	×	×	×

	p,q	p,r	p,s	q,r	q,s	r,s
failure	×	×	×	×	×	✓
possible futures	×	×	×	×	×	×
ready trace	×	×	×	×	×	×
failure trace	×	×	×	×	×	✓

where ✓ to stands for an 'yes' answer w.r.t. the behavioural equivalence of two of the states p, q, r and s, whereas × represents a 'no' answer.

Intuitively, GPSs resemble LTSs, with the difference that each transition is labelled by both an action and the probability of that action being executed. For more insight on decorated trace semantics for GPSs, consider the following systems:

In the setting of GPSs, decorated trace semantics take into consideration paths w which can be executed by a probabilistic process p. Reasoning on the corresponding equivalences is based on the sum of probabilities of occurrence of such w's that, for example, lead p to a set of processes, for the case of trace semantics, or to a set of processes that (fail to) trigger the same sets of actions as a first step, for ready (respectively, failure) semantics.

In Jou and Smolka (1990), a notion of *maximality* was introduced for the case of trace and failure semantics. Intuitively, the former takes into consideration the probability of a process p to execute a certain trace w and terminate, whereas the latter takes into consideration the largest set of actions p fails to trigger as a first step after the execution of w. However, it has been proven in Jou and Smolka (1990) that maximality does not increase the distinguishing power of decorated trace semantics and, moreover, ready and failure equivalence of GPSs coincide.

With respect to (maximal) trace semantics, amongst the systems above, p' and q' are equivalent: they have the same probability of executing traces $w \in \{\varepsilon, a, ab, abc, abd\}$. Moreover, each such w leads p' and q' to sets of processes S_1, S_2 ready to fire the same actions. Consequently, S_1 and S_2 fail to trigger the same sets of actions as a first step. Hence, p' and q' are ready and (maximal) failure equivalent as well. None of the processes above are bisimilar: the corresponding states reached via transitions labelled a (with total probability (1) display different behaviour as they either have different branching structure, or can trigger different actions.

This paper is an extended version of the conference paper (Bonchi *et al.* 2012) where we (a) proved that the coalgebraic ready, failure and (complete) trace semantics for LTSs are equivalent to the corresponding set-theoretic notions from van Glabbeek (2001), (b) showed how the coalgebraic semantics lead to canonical representatives for the aforementioned decorated traces, and (c) showed how to prove decorated trace equivalence of LTSs using coinduction, by constructing bisimulations (up-to context) that witness the desired equivalence. The latter is interesting also from the point of view of tool development: construction of bisimulations is known to be particularly suitable for automation. Moreover, the up-to context technique also increases the efficiency of reasoning, as verifications are performed under certain closure properties, which means that the bisimulations which are built are smaller (see Section 7 for an example). The techniques we used for up-to context reasoning on LTSs are an extension of the recent work in Bonchi and Pous (2013).

In this paper we extend (a)–(c) above also for the case of possible futures, ready trace and failure trace semantics for LTSs and for several equivalences on GPSs. We provide (more) details, proofs and examples on how to use the coalgebraic framework (summarized in Figure 10) for reasoning on decorated trace equivalences for both the case of LTSs and GPSs. We also show that the spectrum of decorated trace semantics in Figure 1 can be recovered from the coalgebraic modelling.

The paper is organized as follows. In Section 2, we provide the basic notions from coalgebra and recall the generalized powerset construction. In Sections 3 and 4, we show how the powerset construction can be applied for determinizing LTSs and GPSs, respectively, in terms of Moore automata ($X, f : X \to B \times X^A$), in order to coalgebraically characterize the corresponding decorated trace semantics. Here we also prove that the obtained coalgebraic models are equivalent to the original definitions, and illustrate how one can reason about decorated trace equivalence by constructing (Moore) bisimulations. A compact overview on the uniform coalgebraic framework is given in Section 5. Section 6 discusses that the canonical representatives of LTSs and GPSs we obtain coalgebraically coincide with the corresponding minimal automata one would obtain by identifying all states equivalent w.r.t. a particular decorated trace semantics. In Section 7, we introduce bisimulations up-to context and emphasize on their efficiency by means of an example for LTSs. Finally, Section 8 contains concluding remarks and discusses future work.

2. Preliminaries

In this section, we briefly recall basic notions from coalgebra and the generalized powerset construction (Silva *et al.* 2010; Lenisa 1999; Cancila *et al.* 2003). We first introduce some notation on sets.

We denote sets by capital letters $X, Y, \ldots$ and functions by lower case letters $f, g, \ldots$. The *cartesian product* of two sets X and Y is denoted by $X \times Y$, and has the projection maps $X \stackrel{\pi_1}{\longleftarrow} X \times Y \stackrel{\pi_2}{\longrightarrow} Y$. By X^Y we represent the family of *functions* $f : Y \to X$, whereas the collection of *finite subsets* of X is denoted by $\mathcal{P}_\omega X$. The collection of all subsets of X is denoted by $\mathcal{P}(X)$. For each of these operations defined on sets, there is an analogous one on functions (for details see for example Awodey (2010)). This turns the operations above into (bi)functors, which we shall use throughout this paper.

We recall the (finitely supported sub)probability distribution functor $\mathcal{D}_\omega$ defined on **Set** – the category of sets and functions. $\mathcal{D}_\omega$ maps a set X to

$$\mathcal{D}_\omega(X) = \{\varphi : X \to [0,1] \mid supp(\varphi) \text{ is finite and } \sum_{x \in X} \varphi(x) \leqslant 1\},$$

where $supp(\varphi) = \{x \in X \mid \varphi(x) > 0\}$ is the *support* of φ. Given a function $g : X \to Y$, $\mathcal{D}_\omega(g) : \mathcal{D}_\omega(X) \to \mathcal{D}_\omega(Y)$ is defined as

$$\mathcal{D}_\omega(g)(\varphi) = \lambda y \,.\, \sum_{g(x)=y} \varphi(x).$$

For an alphabet A, we denote by A^* the set of all *words* over A and by ε the *empty word*. The *concatenation* of words $w_1, w_2 \in A^*$ is written $w_1 w_2$.

2.1. *Coalgebra and bisimulation*

We consider coalgebras of set functors $\mathcal{F} : \mathbf{Set} \to \mathbf{Set}$. An $\mathcal{F}$-*coalgebra* (or *coalgebra*, when $\mathcal{F}$ is understood) is a pair $(X, c : X \to \mathcal{F}X)$. We call X the state space, and we say that $\mathcal{F}$ together with c determine the dynamics, or the transition structure of the $\mathcal{F}$-coalgebra.

An $\mathcal{F}$-*homomorphism* between two $\mathcal{F}$-coalgebras (X, f) and (Y, g), is a function $h : X \to Y$ preserving the transition structure, *i.e.*, $g \circ h = \mathcal{F}(h) \circ f$. $\mathcal{F}$-coalgebras and $\mathcal{F}$-homomorphisms form a category denoted by $\mathbf{Coalg}(\mathcal{F})$.

An $\mathcal{F}$-coalgebra (Ω, ω) is *final* if for any $\mathcal{F}$-coalgebra (X, f) there exists a unique $\mathcal{F}$-homomorphism $[\![-]\!]_X : X \to \Omega$. A final coalgebra represents the universe of all possible *behaviours* of $\mathcal{F}$-coalgebras. The unique morphism $[\![-]\!]_X : X \to \Omega$ maps each state in X to its behaviour. Using this mapping, behavioural equivalence can be defined as follows: for any two coalgebras (X, f) and (Y, g), the states $x \in X$ and $y \in Y$ are *behaviourally equivalent*, written $x \sim_{\mathcal{F}} y$, if and only if they have the same behaviour, that is

$$x \sim_{\mathcal{F}} y \text{ iff } [\![x]\!]_X = [\![y]\!]_Y. \tag{1}$$

We think of $[\![x]\!]_X$ as the *canonical representative* of the behaviour of x. The image of X under $[\![-]\!]_X$ can be viewed as the minimization of (X, f), since the final coalgebra contains no pairs of equivalent states.

For an example, we consider deterministic automata (DA's). A deterministic automaton over the input alphabet A is a pair $(X, \langle o, t \rangle)$, where X is a set of states and $\langle o, t \rangle : X \to 2 \times X^A$ is a function with two components: o, the output function, determines if a state x is final ($o(x) = 1$) or not ($o(x) = 0$); and t, the transition function, returns for each input letter a the next state. DA's are coalgebras for the functor $\mathcal{D}(X) = 2 \times X^A$. The final coalgebra of this functor is $(2^{A^*}, \langle \epsilon, (-)_a \rangle)$ where 2^{A^*} is the set of languages over A and $\langle \epsilon, (-)_a \rangle$, given a language L, determines whether or not the empty word ε is in the language ($\epsilon(L) = 1$ or $\epsilon(L) = 0$, resp.) and, for each input letter a, returns the *derivative* of L: $L_a = \{w \in A^* \mid aw \in L\}$. From any DA, there is a unique map $[\![-]\!]$ into 2^{A^*} which assigns to each state its behaviour (that is, the language that the state recognizes).

$$\begin{array}{ccc} X & \overset{[\![-]\!]_X}{\dashrightarrow} & 2^{A^*} \\ {\scriptstyle \langle o,t \rangle}\downarrow & & \downarrow{\scriptstyle \langle \epsilon,(-)_a \rangle} \\ 2 \times X^A & \underset{id \times [\![-]\!]_X^A}{\dashrightarrow} & 2 \times (2^{A^*})^A \end{array} \qquad \begin{array}{l} [\![x]\!]_X(\varepsilon) = o(x) \\ [\![x]\!]_X(aw) = [\![t(x)(a)]\!]_X(w) \end{array}$$

Therefore, behavioural equivalence for the functor $\mathcal{D}$ coincides with the classical language equivalence of automata.

Another example (fundamental for the rest of the paper) is given by Moore automata. Moore automata with inputs in A and outputs in B are coalgebras for the functor $\mathcal{M}(X) = B \times X^A$, that is pairs $(X, \langle o, t \rangle)$ where X is a set, $t : X \to X^A$ is the transition function (like for DA) and $o : X \to B$ is the output function which maps every state in its output. Thus DA can be seen as a special case of Moore automata where $B = 2$. The final coalgebra for $\mathcal{M}$ is $(B^{A^*}, \langle \epsilon, (-)_a \rangle)$ where B^{A^*} is the set of all functions $\varphi : A^* \to B$, $\epsilon : B^{A^*} \to B$ maps each φ into $\varphi(\varepsilon)$ and $(-)_a : B^{A^*} \to (B^{A^*})^A$ is defined for all $\varphi \in B^{A^*}$,

$a \in A$ and $w \in A^*$ as $(\varphi)_a(w) = \varphi(aw)$.

$$\begin{array}{ccc} X & \overset{[\![-]\!]_X}{\dashrightarrow} & B^{A^*} \\ {\scriptstyle \langle o,t\rangle}\downarrow & & \downarrow{\scriptstyle \langle \epsilon,(-)_a\rangle} \\ B \times X^A & \underset{id \times [\![-]\!]_X^A}{\dashrightarrow} & B \times (B^{A^*})^A \end{array} \qquad \begin{array}{l} [\![x]\!]_X(\varepsilon) = o(x) \\ [\![x]\!]_X(aw) = [\![t(x)(a)]\!]_X(w) \end{array}$$

Coalgebras provide a useful technique for proving behavioural equivalence, namely, *bisimulation*. Let (X, f) and (Y, g) be two $\mathcal{F}$-coalgebras. A relation $R \subseteq X \times Y$ is a *bisimulation* if there exists a function $\alpha_R : R \to \mathcal{F}R$ such that $\pi_1 : R \to X$ and $\pi_2 : R \to Y$ are coalgebra homomorphisms. In Rutten (2000), it is shown that under certain conditions on $\mathcal{F}$ (which are met by all the functors considered in this paper), bisimulations are a sound and complete proof technique for behavioural equivalence, namely,

$$x \sim_{\mathcal{F}} y \text{ iff there exists a bisimulation } R \text{ such that } xRy. \tag{2}$$

2.2. *The generalized powerset construction*

As shown above, every functor $\mathcal{F}$ induces both a notion of $\mathcal{F}$-coalgebra and a notion of behavioural equivalence $\sim_{\mathcal{F}}$. Sometimes, it is interesting to consider different equivalences than $\sim_{\mathcal{F}}$ for reasoning about $\mathcal{F}$-coalgebras. This is the case of LTSs and GPSs which can be modelled as coalgebras for the functor $\mathcal{L}(X) = (\mathcal{P}_\omega X)^A$ and $\mathcal{G}(X) = \mathcal{D}_\omega(A \times X)$, respectively. The corresponding induced behavioural equivalences $\sim_{\mathcal{L}}$ and $\sim_{\mathcal{G}}$ coincide with the standard notion of bisimilarity (Milner 1989; Park 1981) and probabilistic bisimilarity (Jou and Smolka 1990), respectively. However, in concurrency theory, many other equivalences have been studied, notably, *decorated trace equivalences* (van Glabbeek 2001; Jou and Smolka 1990). Another example is given by non-deterministic automata (NDA's) which are coalgebras for the functor $\mathcal{N}(X) = 2 \times (\mathcal{P}_\omega X)^A$. The associated equivalence $\sim_{\mathcal{N}}$ strictly implies language equivalence, which is often the intended semantics.

With this intuition in mind, we refer to the *generalized powerset construction* (Cancila *et al.* 2003; Lenisa 1999; Silva *et al.* 2010) for coalgebras $f : X \to \mathcal{F}T(X)$ for a functor $\mathcal{F}$ and a monad (T, η, μ), with the proviso that that $\mathcal{F}T(X)$ is an algebra for T. Recall that a T-*algebra* for a monad $(T(X), \eta, \mu)$ is a pair $(X, h\colon T(X) \to X)$ satisfying the laws $h \circ \eta = id$ and $h \circ \mu = h \circ Th$. For the case $T = \mathcal{P}_\omega$, T-algebras are semilattices (with bottom).

We briefly summarize the aforementioned construction, for the case when $\mathcal{F}$ has a final coalgebra (Ω, ω), as in the following commuting diagram:

$$\begin{array}{ccccc} X & \xrightarrow{\eta} & T(X) & \overset{[\![-]\!]}{\dashrightarrow} & \Omega \\ {\scriptstyle f}\downarrow & \swarrow{\scriptstyle f^\sharp} & & & \downarrow{\scriptstyle \omega} \\ \mathcal{F}T(X) & & \underset{\mathcal{F}[\![-]\!]}{\dashrightarrow} & & \mathcal{F}(\Omega) \end{array} \tag{3}$$

(We refer the interested reader to Silva *et al.* (2013) where all the technical details are explored and many instances of the construction are shown.)

Intuitively, the coalgebra $f : X \to \mathcal{F}T(X)$ is extended to $f^\sharp : T(X) \to \mathcal{F}T(X)$ which, for two elements $x_1, x_2 \in X$, enables checking their '$\mathcal{F}$-equivalence with respect to the monad T' ($\eta(x_1) \sim_\mathcal{F} \eta(x_2)$) rather than checking their $\mathcal{F}T$-equivalence.

Formally, $f^\sharp$ is the unique algebra map between $(T(X), \mu)$ and $(\mathcal{F}TX, h)$ (where h is a given algebra structure on $\mathcal{F}TX$) such that $f^\sharp \circ \eta = f$. Moreover, one can show that, under certain additional conditions, also Ω has an algebra structure and that $[\![-]\!]$ is also an algebra map (Silva *et al.* 2013).

Remark 2.1. Based on (1) and (2), verifying $\mathcal{F}$-behavioural equivalence of two states x_1, x_2 in a coalgebra $(T(X), f^\sharp)$ consists in identifying a bisimulation R relating $\eta(x_1)$ and $\eta(x_2)$:

$$[\![\eta(x_1)]\!] = [\![\eta(x_2)]\!] \text{ iff } \eta(x_1)\, R\, \eta(x_2). \tag{4}$$

Take, for example, the case of NDA's which are $\mathcal{F}T$-coalgebras for $\mathcal{F}(X) = 2 \times X^A$ and the monad $(T(X) = (\mathcal{P}_\omega(X), \eta, \mu)$, where

$$\begin{array}{ll} \eta : X \to \mathcal{P}_\omega X & \qquad \mu : \mathcal{P}_\omega(\mathcal{P}_\omega X) \to \mathcal{P}_\omega X \\ \eta(x) = \{x\} & \qquad \mu(U) = \bigcup_{S \in U} S. \end{array}$$

Note that $\mathcal{F}T(X)$ is a T-algebra, that is a semilattice, since $2 \cong \mathcal{P}(1)$ is a semilattice and, moreover, product and exponentiation preserve the algebra structure. Therefore, according to the diagram above, every NDA (X, f) is transformed into $(\mathcal{P}_\omega X, f^\sharp)$ which is a DA. This corresponds to the classical powerset construction for determinizing NDA's. The language recognized by a state x can be defined by precomposing the unique morphism $[\![-]\!] : \mathcal{P}_\omega X \to 2^{A^*}$ with the unit of $\mathcal{P}_\omega$. Consequently, this enables reasoning on language equivalence of states of NDA's, in terms of bisimulations.

In this paper, we exploit the coalgebraic modelling of the powerset construction and derive a framework for handling decorated trace semantics of both LTSs and GPSs in terms of (final) Moore coalgebras, in a uniform fashion. We will only be interested in the case $\mathcal{F}(X) = \mathcal{M}(X) = B \times X^A$, for A an action alphabet and B a T-algebra. (Intuitively, B captures the decorations of interest for each of the semantics under consideration.)

To model GPSs, we consider the (sub)probability distribution monad $(\mathcal{D}_\omega(X), \eta, \mu)$ where

$$\begin{array}{ll} \eta : X \to \mathcal{D}_\omega(X) & \qquad \mu : \mathcal{D}_\omega(\mathcal{D}_\omega(X)) \to \mathcal{D}_\omega(X) \\ \eta(x) = \lambda y \,.\, \begin{cases} 1 & \text{if } x = y \\ 0 & \text{otherwise} \end{cases} & \qquad \mu(\psi) = \lambda x \,.\, \sum\limits_{\varphi \in supp(\psi)} \varphi(x) \times \psi(\varphi) \end{array}$$

The algebras for this monad are the so-called positive convex structures (Doberkat 2008).

In Silva *et al.* (2013), it is shown that the function mapping a $\mathcal{F}T$-coalgebra f to the $\mathcal{F}$-colagebra $f^\sharp$ extends to a functor $D : \mathbf{Coalg}(\mathcal{F}T) \to \mathbf{Coalg}(\mathcal{F})$ assigning to each $\mathcal{F}T$-homomorphism h the $\mathcal{F}$-homomorphism $T(h)$. For later use, we fix $\mathbf{Det}(\mathcal{F}T)$ to be the image of $\mathbf{Coalg}(\mathcal{F}T)$ through D and we prove the following lemma.

Lemma 2.1. Let $(TX, f^\sharp)$ and $(TY, g^\sharp)$ be coalgebras in $\mathbf{Det}(\mathcal{F}T)$ and let $\approx_\mathcal{F}$ be the largest bisimulation on $\mathbf{Det}(\mathcal{F}T)$. Then, for all $x \in TX$, $y \in TY$, $x \approx_\mathcal{F} y = x \sim_\mathcal{F} y$.

Proof. Since $\mathbf{Det}(\mathcal{F}T)$ is a subcategory of $\mathbf{Coalg}(\mathcal{F})$, then every bisimulation in $\mathbf{Det}(\mathcal{F}T)$ is also a bisimulation in $\mathbf{Coalg}(\mathcal{F})$ and therefore $\approx_{\mathcal{F}} \subseteq \sim_{\mathcal{F}}$.

For the other direction, take a bisimilation $R \subseteq TX \times TY$, $\pi_1 : R \to TX$, $\pi_2 : R \to TY$ and an $\mathcal{F}$-coalgebra structure $r : R \to \mathcal{F}R$. The latter $f^\sharp$ and $g^\sharp$ can be post-composed with $\mathcal{F}\eta$ and, in this way, both π_1 and π_2 are $\mathcal{F}T$-homomorphisms. As a consequence $(TTX, (\mathcal{F}(\eta) \circ f^\sharp)^\sharp) \stackrel{T(\pi_1)}{\leftarrow} (TR, (\mathcal{F}(\eta) \circ r)^\sharp) \stackrel{T(\pi_2)}{\rightarrow} (TTX, (\mathcal{F}(\eta) \circ f^\sharp)^\sharp)$ is a span in $\mathbf{Det}(\mathcal{F}T)$. By routine calculation (???), one can show that $f^\sharp \circ \mu = (\mathcal{F}(\eta) \circ f^\sharp)^\sharp)$ and $g^\sharp \circ \mu = (\mathcal{F}(\eta) \circ g^\sharp)^\sharp)$ and thus $(TX, f^\sharp) \stackrel{\mu \circ T(\pi_1)}{\leftarrow} (TR, (\mathcal{F}(\eta) \circ r)^\sharp) \stackrel{\mu \circ T(\pi_2)}{\rightarrow} (TX, f^\sharp)$ is a span in $\mathbf{Coalg}(\mathcal{F})$. □

3. Decorated trace semantics of LTSs via determinization

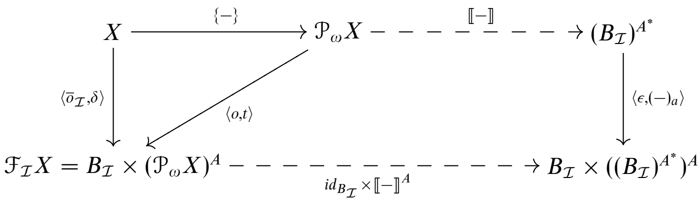
In this section, our aim is to provide a coalgebraic view on decorated trace equivalences of LTSs. We use the generalized powerset construction and show how one can determinize arbitrary labelled transition systems obtaining particular instances of Moore automata (with different output sets) in order to model ready, failure, (complete) trace, possible futures, ready trace and failure trace equivalences. This paves the way to building a general framework for reasoning on decorated trace equivalences in a uniform fashion, in terms of bisimulations (up-to context).

An LTS is a pair (X, δ) where X is a set of states and $\delta : X \to (\mathcal{P}_\omega X)^A$ is a function assigning to each state $x \in X$ and to each label $a \in A$ a finite set of possible successors states. We write $x \stackrel{a}{\to} y$ whenever $y \in \delta(x)(a)$. We extend the notion of transition to words $w = a_1 \ldots a_n \in A^*$ as follows: $x \stackrel{w}{\to} y$ if and only if $x \stackrel{a_1}{\to} \ldots \stackrel{a_n}{\to} y$. For $w = \varepsilon$, we have $x \stackrel{\varepsilon}{\to} y$ if and only if $y = x$.

The coalgebraic characterization of ready, failure and (complete) trace was originally obtained in Silva *et al.* (2013). We recall it here, with a slight adaptation which will be useful for the generalizations we will explore. Given an arbitrary LTS $(X, \delta : X \to (\mathcal{P}_\omega X)^A)$, one constructs a *decorated* LTS, which is a coalgebra of the functor $\mathcal{F}_\mathcal{I}(X) = B_\mathcal{I} \times (\mathcal{P}_\omega X)^A$. More precisely, we construct $(X, \langle \overline{o}_\mathcal{I}, \delta \rangle : X \to B_\mathcal{I} \times (\mathcal{P}_\omega X)^A)$, where the output operation $\overline{o}_\mathcal{I} : X \to B_\mathcal{I}$ provides the observations of interest corresponding to the original LTS and depending on the equivalence we want to study. (Here, $B_\mathcal{I}$ represents an arbitrary semilattice with a $\vee$ operation, instantiated for each of the semantics under consideration as in Silva *et al.* (2013).) Then, the decorated LTS is determinized, as depicted in Figure 2.

Note that both the output operation and its image are parameterized by $\mathcal{I}$, which will vary depending on the type of decorated trace semantics under consideration.

The coalgebraic modelling of possible-futures semantics could easily be recovered by following a similar approach. However, for the case of ready and failure trace semantics the transition structure of the LTS also needs to be slightly modified before the determinization. This consists in changing the alphabet A to include additional information represented by sets of actions ready to be triggered as a first step. Consequently, each LTS $(X, \delta : X \to (\mathcal{P}_\omega X)^A)$ is uniquely associated a coalgebra $(X, \langle \overline{o}_\mathcal{I}, \overline{\delta} : X \to (\mathcal{P}_\omega X)^{\overline{A}} \rangle)$, defined in a natural fashion, as we shall see later on. The construction in Figure 2 is then applied on $(X, \langle \overline{o}_\mathcal{I}, \overline{\delta} \rangle)$.



$$o(Y) = \bigvee_{y \in Y} \overline{o}_\mathcal{I}(y) \qquad [\![Y]\!](\varepsilon) = o(Y)$$
$$t(Y)(a) = \bigcup_{y \in Y} \delta(y)(a) \qquad [\![Y]\!](aw) = [\![\bigcup_{y \in Y} \delta(y)(a)]\!](w)$$

Fig. 2. The powerset construction for decorated LTSs.

The explicit instantiations of $\overline{o}_\mathcal{I}$ and $B_\mathcal{I}$ are provided later in this section, where we will also show that the coalgebraic modelling in fact coincides with the original definitions of the corresponding equivalences. This was not formally shown in Silva *et al.* (2013), for none of the aforementioned semantics.

Our coalgebraic modelling of decorated trace semantics enables the definition of the corresponding equivalences as Moore bisimulations (Rutten 2000) (i.e., bisimulations for a functor $\mathcal{M} = B_\mathcal{I} \times X^A$). This way, checking behavioural equivalence of x_1 and x_2 reduces to checking the equality of their unique representatives in the final coalgebra: $[\![\{x_1\}]\!]$ and $[\![\{x_2\}]\!]$.

In the subsequent sections we (a) provide the details on the coalgebraic modelling of ready, failure, (complete) trace, possible futures, ready trace and failure trace semantics, (b) show that the corresponding representations coincide with their original definitions in van Glabbeek (2001) and (c) show, by means of examples, how the associated coalgebraic framework can be used in order to reason on (some of) the aforementioned equivalences in terms of Moore bisimulations.

3.1. *Ready and failure semantics*

In this section, we show how the ingredients of Figure 2 can be instantiated in order to provide a coalgebraic modelling of ready and failure semantics. Moreover, we prove that the resulting coalgebraic characterizations of these semantics are equivalent to their original definitions in van Glabbeek (2001).

Consider an LTS $(X, \delta : X \to (\mathcal{P}_\omega X)^A)$ and define, for a function $\varphi : A \to \mathcal{P}_\omega X$, the set of *actions enabled by* φ:

$$I(\varphi) = \{a \in A \mid \varphi(a) \neq \emptyset\}, \tag{5}$$

and the set of *actions* φ *fails to enable*:

$$\mathit{Fail}(\varphi) = \{Z \subseteq A \mid Z \cap I(\varphi) = \emptyset\}.$$

For the particular case $\varphi = \delta(x)$, $I(\delta(x))$ denotes the set of all (initial) actions ready to be fired by $x \in X$, and $\mathit{Fail}(\delta(x))$ represents the set of subsets of all (initial) actions that cannot be triggered by such x.

A *ready pair* of x is a pair $(w,Z) \in A^* \times \mathcal{P}_\omega A$ such that $x \xrightarrow{w} y$ and $Z = I(\delta(y))$. A *failure pair* of x is a pair $(w,Z) \in A^* \times \mathcal{P}_\omega A$ such that $x \xrightarrow{w} y$ and $Z \in \mathit{Fail}(\delta(y))$. We denote by $\mathcal{R}(x)$ and $\mathcal{F}(x)$, respectively, the sets of *all ready pairs* and *failure pairs*, respectively, associated to x.

Intuitively, ready semantics identifies states in X based on the actions $a \in A$ they can immediately trigger after performing a certain action sequence $w \in A^*$, i.e., based on their ready pairs. It was originally defined as follows:

Definition 3.1 ($\mathcal{R}$-equivalence (van Glabbeek 2001)). Let $(X, \delta : X \to (\mathcal{P}_\omega X)^A)$ be an LTS and $x, y \in X$ two states. States x and y are *ready equivalent* ($\mathcal{R}$-equivalent) if and only if they have the same set of ready pairs, that is $\mathcal{R}(x) = \mathcal{R}(y)$.

Failure semantics identifies behaviours of states in X according to their failure pairs.

Definition 3.2 ($\mathcal{F}$-equivalence (van Glabbeek 2001)). Let $(X, \delta : X \to (\mathcal{P}_\omega X)^A)$ be an LTS and $x, y \in X$ two states. States x and y are *failure equivalent* ($\mathcal{F}$-equivalent) if and only if $\mathcal{F}(x) = \mathcal{F}(y)$, where

$$\mathcal{F}(x) = \{(w,Z) \in A^* \times \mathcal{P}_\omega A \mid \exists x' \in X.\, x \xrightarrow{w} x' \wedge Z \in \mathit{Fail}(\delta(x'))\}.$$

The coalgebraic modelling of ready, respectively, failure semantics is obtained in a uniform fashion, by instantiating the ingredients of Figure 2 as follows. For $\mathcal{I} \in \{\mathcal{R}, \mathcal{F}\}$, $\overline{o}_\mathcal{I} : X \to \mathcal{P}_\omega(\mathcal{P}_\omega A)$ is defined as:

$$\overline{o}_\mathcal{R}(x) = \{I(\delta(x))\} \qquad\qquad \overline{o}_\mathcal{F}(x) = \mathit{Fail}(\delta(x)).$$

Intuitively, in the setting of ready semantics, the observations provided by the output operation refer to the sets of actions ready to be executed by the states of the LTS. Similarly, for failure semantics, the output operation refers to the sets of actions the states of the LTS cannot immediately fire.

Remark 3.1. Observe that the codomain of $\bar{o}_\mathcal{R}$ is $\mathcal{P}_\omega(\mathcal{P}_\omega A)$, and not $\mathcal{P}_\omega A$, as one might expect. However, this is consistent with the intended semantics. For $B_\mathcal{I} = B_\mathcal{R} = B_\mathcal{F} = \mathcal{P}_\omega(\mathcal{P}_\omega A)$, the final Moore coalgebra has carrier $\mathcal{P}_\omega(\mathcal{P}_\omega A))^{A^*}$ which is isomorphic to $\mathcal{P}(A^* \times \mathcal{P}_\omega(A))$ the type of $\mathcal{R}(x)$ and $\mathcal{F}(x)$. The unique homomorphism into the final coalgebra will associate to each state $\{x\}$ a function that for each $w \in A^*$ returns a set containing all sets $R_{x'}$ of ready (resp. failed) actions triggered by all x' such that $x \xrightarrow{w} x'$, for $x, x' \in X$.

Next, we will prove the equivalence between the coalgebraic modelling of ready and failure semantics and their original definitions, presented above. More explicitly, given an arbitrary LTS $(X, \delta : X \to (\mathcal{P}_\omega X)^A)$ and a state $x \in X$, we want to show that $[\![\{x\}]\!]$ is equal to $\mathcal{I}(x)$, for $\mathcal{I} \in \{\mathcal{R}, \mathcal{F}\}$, depending on the semantics of interest.

The behaviour of a state $x \in X$ is a function $[\![\{x\}]\!] : A^* \to \mathcal{P}_\omega(\mathcal{P}_\omega A)$, whereas $\mathcal{I}(x)$ is defined as a set of pairs in $A^* \times \mathcal{P}_\omega A$. We represent the set $\mathcal{I}(x) \in \mathcal{P}(A^* \times \mathcal{P}_\omega A)$ by a function $\varphi_x^{\mathcal{I}} : \mathcal{P}_\omega(\mathcal{P}_\omega A)^{A^*}$, where, for $w \in A^*$,

$$\varphi_x^{\mathcal{R}}(w) = \{Z \subseteq A \mid x \xrightarrow{w} y \wedge Z = I(\delta(y))\}$$
$$\varphi_x^{\mathcal{F}}(w) = \{Z \subseteq A \mid x \xrightarrow{w} y \wedge Z \in Fail(\delta(y))\}.$$

Showing the equivalence between the coalgebraic and the original definitions of ready, respectively, failure semantics reduces to proving that

$$(\forall x \in X) \,.\, [\![\{x\}]\!] = \varphi_x^{\mathcal{I}}. \tag{6}$$

Theorem 3.1. Let $(X, \delta : X \to (\mathcal{P}_\omega X)^A)$ be an LTS. Then for all $x \in X$ and $w \in A^*$, $[\![\{x\}]\!](w) = \varphi_x^{\mathcal{I}}(w)$.

Proof. For $\mathcal{I}$ ranging over $\{\mathcal{R}, \mathcal{F}\}$, the proof is by induction on words $w \in A^*$. We provide the details for the case of ready semantics. A similar reasoning can be applied for failure semantics.

— *Base case.* $w = \varepsilon$. We have:

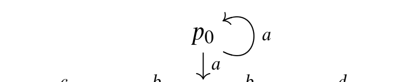
$$\begin{aligned} [\![\{x\}]\!](\varepsilon) &= o(\{x\}) = \{I(\delta(x))\} \\ \varphi_x^{\mathcal{R}}(\varepsilon) &= \{Z \subseteq A \mid x \xrightarrow{\varepsilon} y \wedge Z = I(\delta(y))\} = \{I(\delta(x))\}. \end{aligned}$$

— *Induction step.* Consider $w \in A^*$ and assume, for all $x \in X$, $[\![\{x\}]\!](w) = \varphi_x^{\mathcal{R}}(w)$. We want to prove that $[\![\{x\}]\!](aw) = \varphi_x^{\mathcal{R}}(aw)$, where $a \in A$.

$$\begin{aligned} [\![\{x\}]\!](aw) &= [\![t(\{x\})(a)]\!](w) = \bigcup_{x \xrightarrow{a} z} [\![\{z\}]\!](w) \stackrel{\text{IH}}{=} \bigcup_{x \xrightarrow{a} z} \varphi_z^{\mathcal{R}}(w) \\ \varphi_x^{\mathcal{R}}(aw) &= \{Z \mid x \xrightarrow{aw} y \wedge Z = I(\delta(y))\} \\ &= \{Z \mid x \xrightarrow{a} z \wedge z \xrightarrow{w} y \wedge Z = I(\delta(y))\} \\ &= \bigcup_{x \xrightarrow{a} z} \varphi_z^{\mathcal{R}}(w). \end{aligned}$$

□

Example 3.1. In what follows we illustrate the equivalence between the coalgebraic and the original definitions of ready semantics by means of an example. Consider the following LTS.



We write a^n to represent the action sequence $aa\ldots a$ of length $n \geqslant 1$, with $n \in \mathbb{N}$. The set of all ready pairs associated to p_0 is:

$$\mathcal{R}(p_0) = \{(\varepsilon, \{a\}), (a^n, \{a\}), (a^n, \{b\}), (a^n b, \{c\}), (a^n b, \{d\}), (a^n bc, \emptyset), (a^n bd, \emptyset) \mid n \geqslant 1\}.$$

We can construct a Moore automaton, for $S = \{p_0, p_1, \ldots, p_5\}$,

$$(\mathcal{P}_\omega S, \langle o, t\rangle : \mathcal{P}_\omega S \to \mathcal{P}_\omega(\mathcal{P}_\omega A) \times (\mathcal{P}_\omega S)^A)$$

by applying the generalized powerset construction on the LTS above. The automaton will have $2^6 = 64$ states. We depict the accessible part from state $\{p_0\}$, where the output sets are indicated by double arrows. The output sets of a state Y of the Moore automaton in

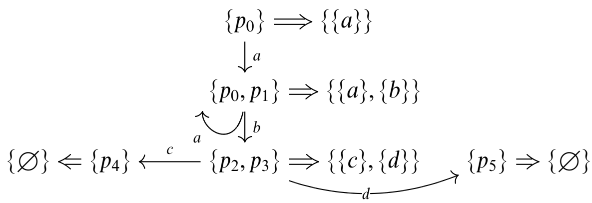
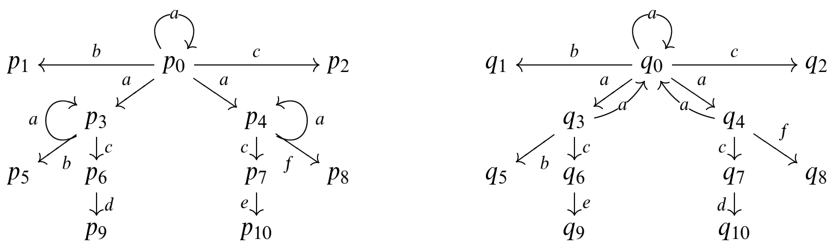


Fig. 3. Ready determinization when starting from $\{p_0\}$.

Figure 3 is the set of actions associated to a certain state $y \in Y$ which can immediately be performed. For example, process p_0 in the original LTS above is ready to perform action a, whereas p_1 can immediately perform b. Therefore, it holds that $o(\{p_0\}) = \{\{a\}\}$ and $o(\{p_0, p_1\}) = \{\{a\}, \{b\}\}$.

By simply looking at the automaton in Figure 3, one can easily see that the set of action sequences $w \in A^*$ the state $\{p_0\}$ can execute, together with the corresponding possible next actions equals $\mathcal{R}(p_0)$. Therefore, the automaton generated according to the generalized powerset construction captures the set of all ready pairs of the initial LTS.

Example 3.2. The last example considered in this section shows how the coalgebraic framework can be applied in order to reason on failure equivalence of LTSs. (Checking ready equivalence complies to a similar approach.) Consider the following two systems.



Let $Z = \{a_1, a_2, \ldots, a_n\}$ be the set of actions a process fails executing as a first step. For the simplicity of notation, we write $[a_1 a_2 \ldots a_n]$ to denote the set of all non-empty subsets $Z' \subseteq Z$. For example, if $Z = \{a_1, a_2\}$, then $[a_1 a_2]$ stands for $\{\{a_1\}, \{a_2\}, \{a_1, a_2\}\}$.

Note that p_0 and q_0 are $\mathcal{F}$-equivalent, according to Definition 3.2, as they have the same sets of failure pairs:

$$\begin{aligned}\mathcal{F}(p_0) = \mathcal{F}(q_0) = \{&(\varepsilon, [def]), (b, [abcdef]), (c, [abcdef])\} \cup \{(a^n, [def]), (a^n, [bde]),\\ &(a^n b, [abcdef]), (a^n c, [abcdef]), (a^n c, [abcef]), (a^n c, [abcdf]),\\ &(a^n f, [abcdef]), (a^n cd, [abcdef]), (a^n ce, [abcdef]) \mid n \in \mathbb{N}, n \geqslant 1\}.\end{aligned}$$

The same conclusion can be reached by checking behavioural equivalence of the two Moore automata generated according to the powerset construction, starting with $\{p_0\}$ and $\{q_0\}$. The fragments of the two automata starting from the states $\{p_0\}$ and $\{q_0\}$ are depicted in Figure 4. The states $\{p_0\}$ and $\{q_0\}$ are Moore bisimilar, since the automata above are isomorphic.

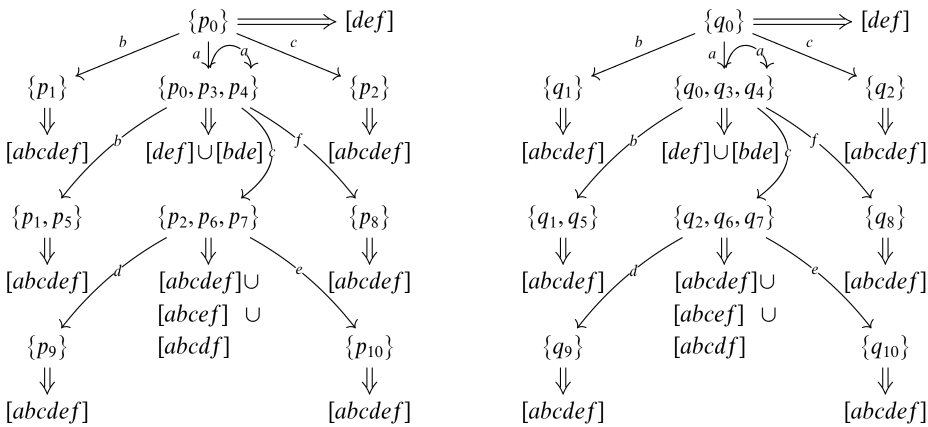


Fig. 4. Failure determinization when starting from $\{p_0\}$ and $\{q_0\}$.

3.2. *(Complete) trace semantics*

In this section, we model coalgebraically trace and complete trace semantics. Similar to the previous section, we also show that the corresponding coalgebraic representations of these semantics are equivalent to their original definitions.

Consider an LTS $(X, \delta : X \to (\mathcal{P}_\omega X)^A)$. Trace semantics identifies states in X according to the set of words $w \in A^*$ they can execute, whereas complete trace semantics identifies states $x \in X$ based on their set of complete traces. A trace $w \in A^*$ of x is complete if and only if x can perform w and reach a deadlock state y or, equivalently,

$$(\exists y \in X) . x \xrightarrow{w} y \wedge I(\delta(y)) = \emptyset.$$

The difference between trace and complete semantics is that the latter enables an external observer to detect stagnation, or deadlock states of a system.

Formally, trace and complete trace equivalences are defined as follows.

Definition 3.3 ($\mathcal{T}$-equivalence (van Glabbeek 2001)). Let $(X, \delta : X \to (\mathcal{P}_\omega X)^A)$ be an LTS and $x, y \in X$ two states. States x and y are *trace equivalent* ($\mathcal{T}$-equivalent) if and only if $\mathcal{T}(x) = \mathcal{T}(y)$, where

$$\mathcal{T}(x) = \{w \in A^* \mid \exists x' \in X. x \xrightarrow{w} x'\}. \tag{7}$$

Definition 3.4 ($\mathcal{CT}$-equivalence (Aceto *et al.* 1999)). Let $(X, \delta : X \to (\mathcal{P}_\omega X)^A)$ be an LTS and $x, y \in X$ two states. States x and y are *complete trace equivalent* ($\mathcal{CT}$-equivalent) if and only if $\mathcal{CT}(x) = \mathcal{CT}(y)$, where

$$\mathcal{CT}(x) = \{w \in A^* \mid \exists x' \in X. x \xrightarrow{w} x' \wedge I(\delta(x')) = \emptyset\}.$$

In what follows we instantiate the constituents of Figure 2 in order to provide the associated coalgebraic modellings.

For $\mathcal{I} \in \{\mathcal{T}, \mathcal{CT}\}$, the output function $\overline{o}_{\mathcal{I}} : X \to 2$ is:

$$\overline{o}_{\mathcal{T}}(x) = 1 \qquad \overline{o}_{\mathcal{CT}}(x) = \begin{cases} 1 & \text{if } I(\delta(x)) = \emptyset \\ 0 & \text{otherwise.} \end{cases}$$

Note that, for trace semantics, one does not distinguish between traces and complete traces. Intuitively, all states are accepting, so they have the same observable behaviour (*i.e.*, $\overline{o}_{\mathcal{T}}(\varphi) = 1$), no matter the transitions they perform. On the other hand, complete trace semantics distinguishes between deadlock states and states that can still execute actions $a \in A$.

Consider, for example, the following LTS:

$$p_1 \xleftarrow{a} p_0 \underset{b}{\overset{a}{\rightleftarrows}} p_2.$$

Observe that $(ab)^*a$ is a complete trace of p_0, as

$$p_0 \xrightarrow{a} p_2 \xrightarrow{b} p_0 \xrightarrow{a} p_2 \xrightarrow{b} \ldots \xrightarrow{b} p_0 \xrightarrow{a} p_1 \tag{8}$$

where p_1 cannot perform any further action.

The above behaviour, described in terms of transitions between states of the Moore automaton derived according to the generalized powerset construction, can be depicted as follows:

$$\{p_0\} \xrightarrow{a} \{p_1, p_2\} \xrightarrow{b} \{p_0\} \xrightarrow{a} \{p_1, p_2\} \xrightarrow{b} \ldots \xrightarrow{b} \{p_0\} \xrightarrow{a} \{p_1, p_2\}$$

where p_1 is a deadlock state and p_2 is not.

Intuitively, we can state that $(ab)^*a$ is a complete trace of $\{p_0\}$, as the deadlock state $p_2 \in \{p_1, p_2\}$ can be reached from $\{p_0\}$ by performing $(ab)^*a$ (see (8)).

Therefore, given $Y_1, Y_2 \subseteq X$ and $w \in A^*$ such that $Y_1 \xrightarrow{w} Y_2$, we observe that w is a complete trace of Y_1 whenever there exists a deadlock state $y \in Y_2$. Otherwise, w is not a complete trace of Y_1.

In the coalgebraic modelling, the above observations regarding (non)stagnating states appear in the definition of the output function $o : \mathcal{P}_\omega(X) \to 2$. Note that, for example, $o(\{p_1, p_2\}) = 1$ and $o(\{p_0\}) = 0$ for the case of complete trace equivalence, as p_1 is a deadlock state and p_0 is not. For trace semantics we have $o(\{p_1, p_2\}) = o(\{p_0\}) = 1$.

Here, $B_{\mathcal{I}} = 2$ and the final Moore coalgebra in Figure 2 is the set of languages 2^{A^*} over A (and the transition structure $\langle \epsilon, (-)_a \rangle$ is simply given by Brzozowski derivatives). Therefore, we can state that the map into the final coalgebra associates to each state $Y \in \mathcal{P}_\omega X$ the set of all traces corresponding to states $y \in Y$, namely, the language:

$$L = \bigcup_{y \in Y} \{w \in A^* \mid (\exists y' \in X) \,.\, y \xrightarrow{w} y'\}.$$

The set $\mathcal{P}(A^*)$ is isomorphic to the set of functions 2^{A^*} which enables us to represent the set $\mathcal{I}(x)$ in terms its characteristic function $\varphi_x^{\mathcal{I}} : A^* \to 2$ defined, for $\mathcal{I} \in \{\mathcal{T}, \mathcal{CT}\}$, $w \in A^*$, as follows:

$$\varphi_x^{\mathcal{T}}(w) = 1 \text{ if } \exists y \in X \,.\, x \xrightarrow{w} y \qquad \varphi_x^{\mathcal{CT}}(w) = \begin{cases} 1 & \text{if } \exists y \in X \,.\, x \xrightarrow{w} y \wedge I(\delta(y)) = \emptyset \\ 0 & \text{otherwise.} \end{cases}$$

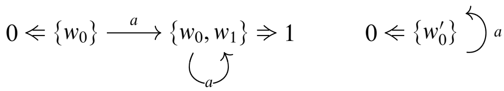


Fig. 5. Complete trace determinization when starting from $\{w_0\}, \{w_0'\}$.

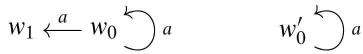
Proving the equivalence between the coalgebraic and the classic definition of (complete) trace semantics reduces to showing that

$$(\forall x \in X) . [\![\{x\}]\!] = \varphi_x^{\mathcal{I}}. \tag{9}$$

Theorem 3.2. Let $(X, \delta : X \to (\mathcal{P}_\omega X)^A)$ be an LTS. Then for all $x \in X$ and $w \in A^*$, $[\![\{x\}]\!](w) = \varphi_x^{\mathcal{I}}(w)$.

Proof. The proof is by induction on words $w \in A^*$ (similar to the proof of Theorem 3.1). □

Example 3.3. Consider the following two LTSs:



Observe that w_0 and w_0' are trace equivalent (according to Definition 3.3), as they output the same sets of traces

$$\mathcal{T}(w_0) = \mathcal{T}(w_0') = \{\varepsilon\} \cup \{a^n \mid n \in \mathbb{N}, n \geqslant 1\}$$

but they are not complete trace equivalent (according to Definition 3.4), as w_0' can never reach a deadlock state, whereas w_0 can reach the stagnating state w_1.

The complete trace determinization contains the sub-automata starting from states $\{w_0\}$ and $\{w_0'\}$ depicted in Figure 5. States $\{w_0\}$ and $\{w_0'\}$ are not behaviourally equivalent, since $\{w_0, w_1\}$ outputs 1, whereas $\{w_0'\}$ never reaches a state with this output. Hence, as expected, we will never be able to build a bisimulation containing states $\{w_0\}$ and $\{w_0'\}$.

On the other hand, in the setting of trace semantics, the determinized (Moore) automata associated to w_0 and w_0', respectively, are similar to those depicted in Figure 5, with the difference that now all their states output value 1. This makes the aforementioned automata bisimilar, hence providing a 'yes' answer w.r.t. $\mathcal{T}$-equivalence of w_0 and w_0', as anticipated.

3.3. *Possible-futures semantics*

In what follows we provide a coalgebraic modelling of possible-futures semantics and show that it coincides with the original definition in van Glabbeek (2001). We also give an example on how the generalized powerset construction and Moore bisimulations can be used in order to reason on possible-futures equivalence.

Let $(X, \delta : X \to (\mathcal{P}_\omega X)^A)$ be an LTS. A *possible future* of $x \in X$ is a pair $\langle w, T \rangle \in A^* \times \mathcal{P}(A^*)$ such that $x \xrightarrow{w} y$ and $T = \mathcal{T}(y)$ (where $\mathcal{T}(y)$ is the set of traces of y, as in Section 3.2).

Possible-futures semantics identifies states that can trigger the same sets of traces $w \in A^*$ and moreover, by executing such w, they reach trace-equivalent states.

Definition 3.5 ($\mathcal{PF}$-equivalence (van Glabbeek 2001)). Let $(X, \delta : X \to (\mathcal{P}_\omega X)^A)$ be an LTS and $x, y \in X$ two states. States x and y are *possible-futures equivalent* ($\mathcal{PF}$-equivalent) if and only if $\mathcal{PF}(x) = \mathcal{PF}(y)$, where

$$\mathcal{PF}(x) = \{\langle w, T\rangle \in A^* \times \mathcal{P}(A^*) \mid \exists x' \in X . x \xrightarrow{w} x' \wedge T = \mathcal{T}(x')\}.$$

The ingredients of Figure 2 are instantiated as follows.

The output function $\bar{o}_{\mathcal{PF}} : X \to \mathcal{P}(\mathcal{P}A^*)$, which refers to the set of traces enabled by states $x \in X$ of the LTS, is defined as

$$\bar{o}_{\mathcal{PF}}(x) = \{\mathcal{T}(x)\}.$$

Here, $B_{\mathcal{I}} = B_{\mathcal{PF}} = \mathcal{P}(\mathcal{P}A^*)$ and the behaviour of a state $x \in X$ in the final coalgebra is given in terms of a function $[\![\{x\}]\!] : A^* \to \mathcal{P}(\mathcal{P}A^*)^{A^*}$, which, intuitively, for each $w \in A^*$ returns the set of sets T_y of traces corresponding to states $y \in X$ such that $x \xrightarrow{w} y$.

Next we want to show that for each $x \in X$, $[\![\{x\}]\!]$ and $\mathcal{PF}(x)$ coincide.

First we choose to equivalently represent $\mathcal{PF}(x) \in \mathcal{P}(A^* \times \mathcal{P}(A^*))$ – the set of all possible futures of a state $x \in X$ – in terms of $\varphi_x^{\mathcal{PF}} \in (\mathcal{P}(\mathcal{P}A^*))^{A^*}$, where

$$\varphi_x^{\mathcal{PF}}(w) = \{\mathcal{T}(y) \mid x \xrightarrow{w} y\},$$

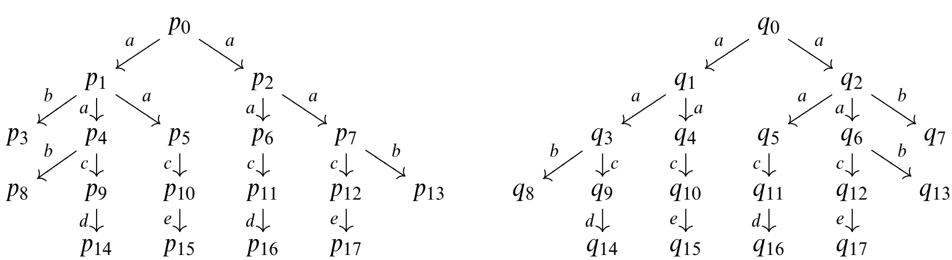
Showing the equivalence between the coalgebraic and the original definition of possible-futures semantics reduces to proving that

$$(\forall x \in X) . [\![\{x\}]\!] = \varphi_x^{\mathcal{PF}}. \tag{10}$$

Theorem 3.3. Let $(X, \delta : X \to (\mathcal{P}_\omega X)^A)$ be an LTS. Then for all $x \in X$ and $w \in A^*$, $[\![\{x\}]\!](w) = \varphi_x^{\mathcal{PF}}(w)$.

Proof. The proof is by induction on $w \in A^*$ (similar to the proof of Theorem 3.1). □

Example 3.4. Consider the following LTSs.



Note that p_0 and q_0 are possible-futures equivalent, as the traces both can follow are sequences $w \in \{a, ab, aa, aab, aac, aacd, aace\}$ and moreover, by triggering the same w they reach states with equal sets of traces. The equivalence between p_0 and q_0 can be formally captured in terms of a bisimulation relation R on the associated Moore automata (generated according to the generalized powerset construction) depicted in Figure 6, where

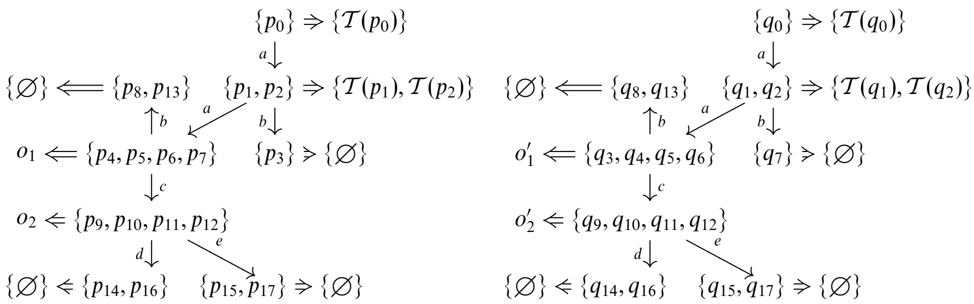


Fig. 6. Possible-futures determinization when starting from $\{p_0\}, \{q_0\}$.
$o_1 = \{\mathcal{T}(p_4), \mathcal{T}(p_5), \mathcal{T}(p_6), \mathcal{T}(p_7)\}, o_2 = \{\mathcal{T}(p_9), \mathcal{T}(p_{10}), \mathcal{T}(p_{11}), \mathcal{T}(p_{12})\}$,
$o'_1 = \{\mathcal{T}(q_3), \mathcal{T}(q_4), \mathcal{T}(q_5), \mathcal{T}(q_6)\}$, $o'_2 = \{\mathcal{T}(q_9), \mathcal{T}(q_{10}), \mathcal{T}(q_{11}), \mathcal{T}(q_{12})\}$.

$$R = \{(\{p_0\}, \{q_0\}), (\{p_1, p_2\}, \{q_1, q_2\}), (\{p_3\}, \{q_7\}), (\{p_8, p_{13}\}, \{q_8, q_{13}\}),$$
$$(\{p_5, p_5, p_6, p_7\}, \{q_3, q_4, q_5, q_6\}), (\{p_9, p_{10}, p_{11}, p_{12}\}, \{q_9, q_{10}, q_{11}, q_{12}\}),$$
$$(\{p_{14}, p_{16}\}, \{q_{14}, q_{16}\}), (\{p_{15}, p_{17}\}, \{q_{15}, q_{17}\})\ \}.$$

It is easy to check that R is a bisimulation, since both automata in Figure 6 are isomorphic. (Note that equality of the outputs – which are sets of traces – can be established using the framework introduced in Section 3.2.)

3.4. *Ready and failure trace semantics*

In this section, we provide a coalgebraic modelling of ready and failure trace semantics by employing the generalized powerset construction. Similarly to the other semantics tackled so far, we show (a) that the coalgebraic representation coincides with the original definition in van Glabbeek (2001) and (b) how to apply the coalgebraic machinery in order to reason on the corresponding equivalences.

Intuitively, ready trace semantics identifies two states if and only if they can follow the same traces w, and moreover, the corresponding (pairwise-taken) states determined by such w's have equivalent one-step behaviours. Failure trace semantics identifies states that can trigger the same traces w, and moreover, the (pairwise-taken) intermediate states occurring during the execution of a such w fail triggering the same (sets of) actions. Formally, the associated definitions are as follows:

Definition 3.6 ($\mathcal{RT}$-equivalence (van Glabbeek 2001)). Let $(X, \delta : X \to (\mathcal{P}_\omega X)^A)$ be an LTS and $x, y \in X$ two states. States x and y are *ready trace equivalent* ($\mathcal{RT}$-equivalent) if and only if $\mathcal{RT}(x) = \mathcal{RT}(y)$, where

$$\begin{aligned} \mathcal{RT}(x) = \{ \;& I_0 a_1 I_1 a_2 \ldots a_n I_n \in \mathcal{P}_\omega(A) \times (A \times \mathcal{P}_\omega(A))^* \mid \\ & (\exists x_1, \ldots, x_n \in X) \,.\, x = x_0 \xrightarrow{a_1} x_1 \xrightarrow{a_2} \cdots \xrightarrow{a_n} x_n \wedge \\ & (\forall i = 0, \ldots, n) \,.\, I_i = I(\delta(x_i)) \;\}. \end{aligned}$$

We call an element of $\mathcal{RT}(x)$ a *ready trace* of x.

Definition 3.7 (**$\mathcal{FT}$-equivalence**). Let $(X, \delta : X \to (\mathcal{P}_\omega X)^A)$ be an LTS and $x, y \in X$ two states. States x and y are *failure trace equivalent* ($\mathcal{FT}$-equivalent) if and only if $\mathcal{FT}(x) = \mathcal{FT}(y)$, where

$$\mathcal{FT}(x) = \{ \ F_0 a_1 F_1 a_2 \dots a_n F_n \in \mathcal{P}_\omega(A) \times (A \times \mathcal{P}_\omega(A))^* \mid (\exists x_1, \dots, x_n \in X) \,.\, x = x_0 \xrightarrow{a_1} x_1 \xrightarrow{a_2} \dots \xrightarrow{a_n} x_n \wedge F_i \in \mathit{Fail}(\delta(x_i)) \ \}.$$

We call an element of $\mathcal{FT}(x)$ a *failure trace* of x.

In order to model these two equivalences coalgebraically we will have to apply the generalized powerset construction, from Figure 2, not only by adding the output function but also by changing the transitions of the LTS.

In particular, we have to add to transitions of shape $x \xrightarrow{a} y$ information regarding the sets of actions ready to be triggered by x. In the new LTS, we consider transitions of shape $x \xrightarrow{\langle a, I(\delta(x))\rangle} y$ therefore enabling the construction of Moore automata 'collecting' states that have been reached not only via one-step transitions labelled the same, but also from processes sharing the same initial behaviour. (Note that $F \in \mathit{Fail}(\delta(x))$ whenever $F \subseteq A - I(\delta(x))$.)

We apply the generalized powerset construction to the decorated LTS:

$$X \xrightarrow{\langle \bar{o}_{\mathcal{I}}, \overline{\delta} \rangle} \mathcal{P}_\omega(\mathcal{P}_\omega(A)) \times \mathcal{P}_\omega(X)^{A \times \mathcal{P}_\omega(A)}$$

where $\overline{\delta}$ is defined by first computing the set I and then appending it to every successor of a state by using the strength of powerset:

$$\overline{\delta} = \ X \xrightarrow{\delta} \mathcal{P}_\omega(X)^A \xrightarrow{\langle I, id\rangle} \mathcal{P}_\omega(A) \times \mathcal{P}_\omega(X)^A \xrightarrow{st} \mathcal{P}_\omega(\mathcal{P}_\omega(A) \times X)^A \to \mathcal{P}_\omega(X)^{A \times \mathcal{P}_\omega(A)}.$$

For $\mathcal{I} \in \{\mathcal{RT}, \mathcal{FT}\}$, the output function $\bar{o}_{\mathcal{I}}$ provides information with respect to the actions ready, respectively, failed to be triggered by a state $x \in X$ as a first step:

$$\bar{o}_{\mathcal{RT}}(x) = \{I(\delta(x))\} \qquad\qquad \bar{o}_{\mathcal{FT}}(x) = \mathit{Fail}(\delta(x)).$$

We need to show that for $x_0 \in X$, there is a one-to-one correspondence between $[\![\{x_0\}]\!]$ and $\mathcal{I}(x_0)$. Intuitively, for ready trace semantics, for example, each behaviour

$$[\![\{x_0\}]\!](\bar{w}) = \{Z_n^j \mid x_a \xrightarrow{w} x_j\}, \ \ \text{with } \bar{w} = \langle a_1, Z_0\rangle \cdots \langle a_n, Z_{n-1}\rangle \in (A \times \mathcal{P}_\omega(A))^* \text{ and } w = a_1 \dots a_n \in A^*$$

corresponds to a set of sequences of shape

$$Z_0 a_1 Z_1 a_2 \dots Z_{n-1} a_n Z_n^j \in \mathcal{I}(x_0).$$

Given $x \in X$, for $\mathcal{I} \in \{\mathcal{RT}, \mathcal{FT}\}$, we again represent $\mathcal{I}(x) \in \mathcal{P}(\mathcal{P}_\omega(A) \times (A \times \mathcal{P}_\omega(A))^*)$ by a function $\varphi_x^{\mathcal{I}}$:

$$\varphi_x^{\mathcal{RT}}(\bar{w}) = \{Z \subseteq A \mid x \xrightarrow{\bar{w}} y \wedge Z = I(\delta(y))\}$$

$$\varphi_x^{\mathcal{FT}}(\bar{w}) = \{Z \subseteq A \mid x \xrightarrow{\bar{w}} y \wedge Z \in \mathit{Fail}(\delta(y))\}.$$

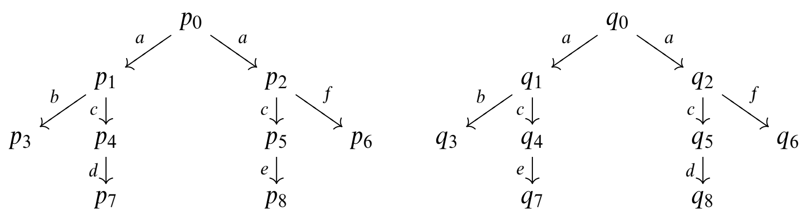
Showing the equivalence between the coalgebraic and the original definition of ready and failure trace semantics consists in proving that

$$(\forall x \in X).\, [\![\{x\}]\!] = \varphi_x^{\mathcal{I}}. \tag{11}$$

Theorem 3.4. Let $(X, \delta : X \to (\mathcal{P}_\omega X)^A)$ be an LTS. Then for all $x \in X$ and $\bar{w} \in (A \times \mathcal{P}_\omega(A))^*$, $[\![\{x\}]\!](\bar{w}) = \varphi_x^{\mathcal{I}}(\bar{w})$.

Proof. The proof follows by induction on words $w \in (A \times \mathcal{P}_\omega(A))^*$ (similar to the proof of Theorem 3.1). □

Example 3.5. Consider the following two systems:



Note that they are not ready trace equivalent as, for example, $\{a\}a\{c,f\}c\{e\}$ is a ready trace of p_0 but not of q_0. Moreover, they are not failure trace equivalent as, for example, $\{b,c,d,e,f\}a\{a,d,e,f\}c\{a,b,c,e,f\}d\{a,b,c,d,e,f\}$ is a failure trace of p_0 but not of q_0.

It is easy to check that by taking exactly the generalized powerset construction (starting with $\{p_0\},\{q_0\}$) without changing the transition function, as in Section 3.1, one gets two bisimilar Moore automata (for both the case of ready and failure trace equivalence). This would indicate that the initial LTSs are behavioural equivalent (which is not the case for ready and failure trace!).

The change in the transition function generates the automata (with labels in $A \times \mathcal{P}_\omega(A)$) in Figure 7. Then, for both semantics studied in this section, the determinization derives the two Moore automata in Figure 8.

For ready trace semantics it holds that:

$$\begin{array}{llll} o_0 = \bar{o}_0 = \{\{a\}\} & o_{12} = \bar{o}_{12} = \{\{b,c\},\{c,f\}\} & o_4 = \bar{o}_5 = \{\{d\}\} & o_5 = \bar{o}_4 = \{\{e\}\} \\ \multicolumn{4}{l}{o_3 = o_6 = o_7 = o_8 = \bar{o}_3 = \bar{o}_6 = \bar{o}_7 = \bar{o}_8 = \{\emptyset\}.} \end{array}$$

Hence, the systems in Figure 8 are not bisimilar as, for example, both states $\{p_4\}$ and $\{q_4\}$ can be reached via transitions labelled the same, but they output different sets of ready actions – namely $\{\{d\}\}$ and $\{\{e\}\}$, respectively. Therefore, we conclude that p_0 and q_0 are not ready trace equivalent.

Similarly, for failure trace we have:

$$\begin{array}{llll} o_0 = \bar{o}_0 = [bcdef] & o_{12} = \bar{o}_{12} = [adef] \cup [abde] & o_4 = \bar{o}_5 = [abcef] & o_5 = \bar{o}_4 = [abcdf] \\ \multicolumn{4}{l}{o_3 = o_6 = o_7 = o_8 = \bar{o}_3 = \bar{o}_6 = \bar{o}_7 = \bar{o}_8 = [abcdef].} \end{array}$$

As before, the automata in Figure 8 are not bisimilar as, for example, both $\{p_4\}$ and $\{q_4\}$ are reached via transitions labelled the same, but have different outputs. Therefore, we conclude that p_0 and q_0 are not failure trace equivalent.

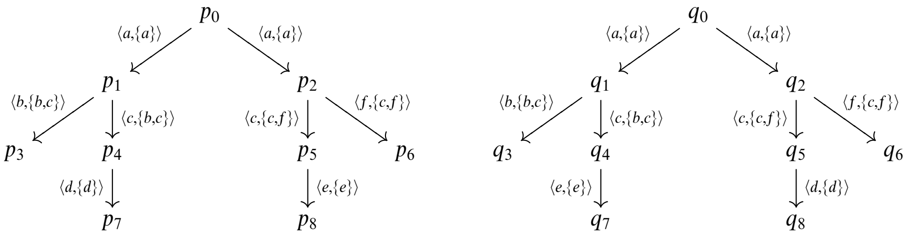


Fig. 7. Altered transition function before determinization.

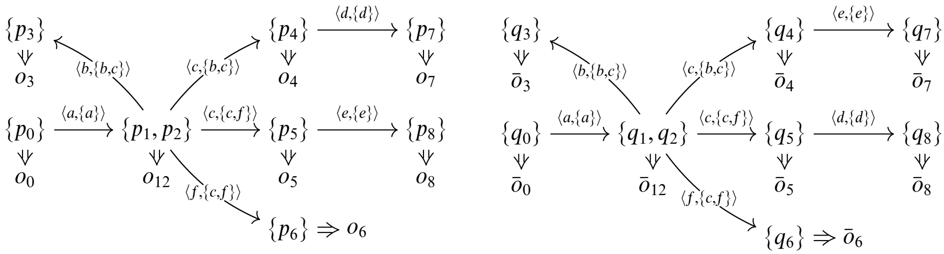


Fig. 8. Determinization starting from $\{p_0\}, \{q_0\}$.

The purpose of changing the transition labels with sets of ready actions is to collect in a Moore state only states of the initial LTSs that have been reached from 'parents' with the same one-step (initial) behaviour. Or dually, to distinguish between states that have 'parents' ready, respectively, failing to trigger different sets of actions. This way one avoids the unfortunate situation of encapsulating, for example, the states p_4, p_5, respectively q_4, q_5, fact which eventually would lead to providing a positive answer with respect to both ready and failure trace equivalence of p_0 and q_0.

In other words, the change in the transition function is needed in order to guarantee that whenever two states of an LTS are ready/failure trace equivalent, the (pairwise-taken) states determined by the executions of a given trace have the same initial behaviour.

4. Decorated trace semantics for GPSs via determinization

In this section, we show how the generalized powerset construction for coalgebras $f : X \to \mathcal{F}T(X)$ for a functor $\mathcal{F}$ and a monad T in (3) can be instantiated in order to provide coalgebraic modellings of decorated trace semantics for GPSs. More explicitly, we show how the determinization procedure can be applied in order to derive coalgebraic representations of ready, (maximal) failure and (maximal) trace semantics, equivalent to their standard definitions in Jou and Smolka (1990).

A GPS is similar to an LTS, but each transition is labelled by both an action and a probability p. More precisely, the transition dynamics is given by a *probabilistic transition* *function* $\mu : X \times A \times X \to [0,1]$ satisfying for all $x \in X$

$$\sum_{\substack{a \in A \\ y \in X}} \mu(x,a,y) \leqslant 1, \tag{12}$$

where X is the state space and A is the alphabet of actions. For simplicity, we write $\mu_a(x,y)$ in lieu of $\mu(x,a,y)$ and we will use the notation $x \xrightarrow{a[v]} y$ for $\mu_a(x,y) = v$. We extend μ to words $w \in A^*$:

$$\mu_\varepsilon(x,y) = \begin{cases} 1 & \text{if } x = y \\ 0 & \text{if } x \neq y \end{cases} \qquad \mu_{aw}(x,y) = \sum_{x' \in X} \mu_a(x,x') \times \mu_w(x',y).$$

Intuitively, $\mu_w(x,y)$ represents the sum of the probabilities associated to all traces w from x to y. Moreover, we write

$$\mu_0(x,\mathbf{0}) = 1 - \sum_{\substack{a \in A \\ y \in X}} \mu(x,a,y)$$

for the probability of x to *terminate*, where 0 is a special symbol not in A, called the *zero action*, and $\mathbf{0}$ is the (deadlock-like) *zero process* whose only transition is $\mu_0(\mathbf{0},\mathbf{0}) = 1$.

Similarly to the case of LTSs, the set of initial actions that can be triggered (with a probability greater than 0) from $x \in X$ is given by

$$I(x) = \{a \in A \mid (\exists y \in X) \,.\, \mu_a(x,y) > 0\},$$

whereas failure sets $Z \in \mathcal{P}_\omega A$ satisfy the condition $Z \cap I(x) = \emptyset$. We write $\mathit{Fail}(x)$ to represent the set of all failure sets of x.

The decorated trace semantics for GPSs considered in this paper can be intuitively described as follows. Given two states $x, y \in X$, we say that x and y are equivalent whenever traces $w \in A^*$

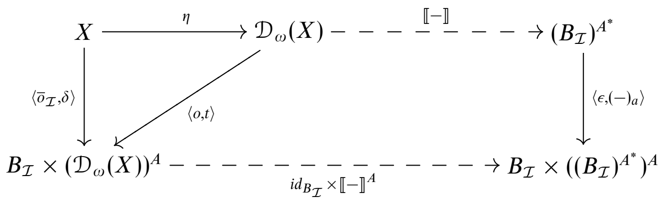
- lead, with the same probability, x and y to processes that trigger (respectively, fail to execute) as a first step the same sets of actions, for the case of ready (respectively, failure) semantics. Note that maximal failure semantics takes into consideration only the largest sets of failure actions (i.e., $A - I(x)$, $A - I(y)$).
- can be executed with the same probability from both x and y, for the case of trace semantics and, moreover, lead x and y to processes that have the same probability to terminate, for the case of maximal trace semantics.

For the coalgebraic modelling of the aforementioned semantics, we will model GPSs as coalgebras $(X, \delta : X \to (\mathcal{D}_\omega(X))^A)$ by setting $\delta(x)(a)(y) = \mu_a(x,y)$.† To these, we associate *decorated* GPSs

$$(X, \langle \overline{o}_{\mathcal{I}}, \delta \rangle : X \to B_{\mathcal{I}} \times (\mathcal{D}_\omega(X))^A)$$

'parameterized' by $\mathcal{I}$, depending on the semantics under consideration.

† Note that the coalgebraic type directly corresponds to reactive systems (Bartels *et al.* 2004). The embedding of generative into reactive is injective and poses no problems semantic-wise. In the sequel, when we write 'Let $(X, \delta : X \to (\mathcal{D}_\omega(X))^A)$ be a GPS' we implicitly mean a coalgebra of this type originating from a GPS defined by a probabilistic function $\mu : X \times A \times X \to [0,1]$ as in (12).



$$o = h_\mathcal{I} \circ \mathcal{D}_\omega(\overline{o}_\mathcal{I}) \qquad [\![\varphi]\!](\varepsilon) = o(\varphi)$$

$$t(\varphi)(a)(y) = \sum_{x \in supp(\varphi)} \delta(x)(a)(y) \times \varphi(x) \qquad [\![\varphi]\!](aw) = [\![t(\varphi)(a)]\!](w)$$

Fig. 9. The powerset construction for decorated GPSs.

Decorated GPSs can be determinized according to the generalized powerset construction as illustrated in Figure 9, where T is instantiated with the probability distribution monad $(\mathcal{D}_\omega, \mu, \eta)$, as defined in Section 2, and $\mathcal{F}$ is $B_\mathcal{I} \times (-)^A$. Moreover, for each of the semantics of interest the observations set $B_\mathcal{I}$ has to carry a $\mathcal{D}_\omega$-algebra structure, or, equivalently, there has to exist a morphism $h_\mathcal{I}$ such that $(B_\mathcal{I}, h_\mathcal{I} : \mathcal{D}_\omega(B_\mathcal{I}) \to B_\mathcal{I})$ is a $\mathcal{D}_\omega$-algebra.

The ingredients $\overline{o}_\mathcal{I}, B_\mathcal{I}$ and $h_\mathcal{I}$ of Figure 9 are explicitly defined in the subsequent sections for each of the coalgebraic decorated trace semantics. The latter are also proven to be equivalent with their corresponding definitions in Jou and Smolka (1990).

4.1. *Ready and (maximal) failure semantics*

In this section, we provide the detailed coalgebraic modelling of ready and (maximal) failure semantics and show the equivalence with their counterparts defined in Jou and Smolka (1990), as follows:

Definition 4.1 (ready equivalence (Jou and Smolka 1990)). The *ready function* $\mathcal{R}_p : X \to ((A^* \times \mathcal{P}_\omega A) \to [0,1])$ is given by

$$\mathcal{R}_p(x)((w, I)) = \sum_{I = I(y)} \mu_w(x, y).$$

We say that $x, x' \in X$ are *ready equivalent* whenever $\mathcal{R}_p(x) = \mathcal{R}_p(x')$.

Definition 4.2 (failure equivalence (Jou and Smolka 1990)). The *failure function* $\mathcal{F}_p : X \to ((A^* \times \mathcal{P}_\omega A) \to [0,1])$ is given by

$$\mathcal{F}_p(x)((w, Z)) = \sum_{Z \cap I(y) = \emptyset} \mu_w(x, y).$$

We say that $x, x' \in X$ are *failure equivalent* whenever $\mathcal{F}_p(x) = \mathcal{F}_p(x')$.

Definition 4.3 (maximal failure equivalence (Jou and Smolka 1990)). The *maximal failure function* $\mathcal{MF}_p : X \to ((A^* \times \mathcal{P}_\omega A) \to [0,1])$ is given by

$$\mathcal{MF}_p(x)((w, Z)) = \sum_{Z = A - I(y)} \mu_w(x, y).$$

We say that $x, x' \in X$ are *maximal failure equivalent* whenever $\mathcal{MF}_p(x) = \mathcal{MF}_p(x')$.

Intuition: *ready* and *(maximal) failure semantics*, respectively, identify states which have the same probability of reaching processes sharing the same sets of ready actions I, or (maximal) sets of failure actions Z, respectively, by executing the same traces $w \in A^*$. Consequently, appropriate modellings in the coalgebraic setting should capture sets of traces w, together with some notion of observations based on execution probabilities of such w's and sets of ready/(maximal) failure actions.

As a first step we define $B_{\mathcal{I}}$, the observation set in Figure 9, as $[0,1]^{\mathcal{P}_\omega(A)}$, for ready, failure and maximal failure semantics (for which, for consistency of notation, $\mathcal{I}$ will be instantiated with $\mathcal{R}_p$, $\mathcal{F}_p$ and $\mathcal{MF}_p$, respectively).

The associated 'decorating' functions $\overline{o}_{\mathcal{I}} : X \to [0,1]^{\mathcal{P}_\omega(A)}$ are defined for $x \in X$ as:

$$\overline{o}_{\mathcal{R}_p}(x)(I) = \begin{cases} 1 & \text{if } I = I(x) \\ 0 & \text{otherwise.} \end{cases} \qquad \overline{o}_{\mathcal{F}_p}(x)(Z) = \begin{cases} 1 & \text{if } Z \cap I(x) = \varnothing \\ 0 & \text{otherwise.} \end{cases}$$

$$\overline{o}_{\mathcal{MF}_p}(x)(Z) = \begin{cases} 1 & \text{if } Z = A - I(x) \\ 0 & \text{otherwise.} \end{cases}$$

For the generalized powerset construction for GPSs, $B_{\mathcal{I}} = [0,1]^{\mathcal{P}_\omega(A)}$ is required to carry a $\mathcal{D}_\omega$-algebra structure. This structure is given by the pointwise extension of the free algebra structure in $[0,1] = \mathcal{D}_\omega(1)$:

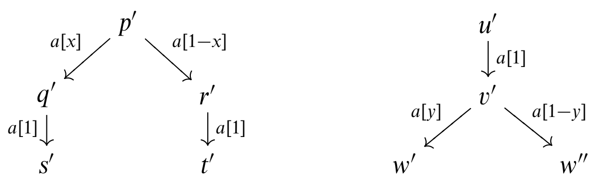
$$h_{\mathcal{I}} : \mathcal{D}_\omega([0,1]^{\mathcal{P}_\omega(A)}) \to [0,1]^{\mathcal{P}_\omega(A)}$$
$$h_{\mathcal{I}}(\varphi)(Z) = \sum_{f \in supp(\varphi)} \varphi(f) \times f(Z).$$

It is easy to check that, for $\mathcal{I} \in \{\mathcal{R}_p, \mathcal{F}_p, \mathcal{MF}_p\}$, the output function $o = h_{\mathcal{I}} \circ \mathcal{D}_\omega(\overline{o}_{\mathcal{I}})$ is explicitly defined, for $\varphi \in \mathcal{D}_\omega(X)$, as:

$$o(\varphi)(S) = \sum_{x \in supp(\varphi)} \varphi(x) \times \overline{o}_{\mathcal{I}}(x)(S).$$

This enables the modelling of the behaviour of GPSs in terms of (final) Moore machines with state space in $(B_{\mathcal{I}})^{A^*}$ and observations in $B_{\mathcal{I}}$. More explicitly, given a GPS (X, δ), the decorated trace behaviour of $x \in X$ is represented in the coalgebraic setting by $[\![\eta(x)]\!] \in (B_{\mathcal{I}})^{A^*} = ([0,1]^{\mathcal{P}_\omega(A)})^{A^*} \cong [0,1]^{A^* \times \mathcal{P}_\omega(A)}$, precisely the type of the functions in Definitions 4.1–4.3. This paves the way for reasoning on ready and (maximal) failure equivalence by coinduction, in terms of Moore bisimulations.

Example 4.1. Consider, for example, the following GPSs:



States p' and u' are ready equivalent, as their corresponding ready functions in Definition 4.1 are equal:

$$\begin{array}{llll} \mathcal{R}_p(p')(\varepsilon,\emptyset) = 0 & \mathcal{R}_p(p')(\varepsilon,\{a\}) = 1 & \mathcal{R}_p(p')(a,\emptyset) = 0 & \mathcal{R}_p(p')(aa,\{a\}) = 0 \\ \mathcal{R}_p(u')(\varepsilon,\emptyset) = 0 & \mathcal{R}_p(u')(\varepsilon,\{a\}) = 1 & \mathcal{R}_p(u')(a,\emptyset) = 0 & \mathcal{R}_p(u')(aa,\{a\}) = 0 \end{array}$$

$$\begin{aligned} \mathcal{R}_p(p')(a,\{a\}) &= \mu_a(p',q') + \mu_a(p',r') = x + (1-x) = 1 \\ \mathcal{R}_p(p')(aa,\emptyset) &= \mu_{aa}(p',s') + \mu_{aa}(p',t') = x \times 1 + (1-x) \times 1 = 1 \\ \mathcal{R}_p(u')(a,\{a\}) &= \mu_a(u',v') = 1 \\ \mathcal{R}_p(u')(aa,\emptyset) &= \mu_{aa}(u',w') + \mu_{aa}(u',w'') = 1 \times y + 1 \times (1-y) = 1. \end{aligned}$$

Intuitively, $\mathcal{R}_p(p')(\varepsilon,\emptyset) = 0$ states that from p', by executing the empty trace ε, the probability to reach states that cannot further trigger any action is 0. This is indeed the case, as p' can always fire a as a first step. Similarly, $\mathcal{R}_p(u')(a,\{a\}) = 1$ states that the probability of performing a from u' and reaching states with the ready set $\{a\}$ is 1. This because $u' \xrightarrow{a[1]} v'$ and $I(v') = \{a\}$.

The same answer w.r.t. the ready equivalence of p' and u' is obtained by applying the coalgebraic framework. As illustrated below, the corresponding Moore automata derived starting from p' and u', respectively, are bisimilar; they have the same branching structure and equal outputs:

$$p': \quad \begin{array}{ccccc} \varphi_1 & \xrightarrow{a} & \varphi_2 & \xrightarrow{a} & \varphi_3 \\ \Downarrow & & \Downarrow & & \Downarrow \\ o_{\varphi_1} & & o_{\varphi_2} & & o_{\varphi_3} \end{array} \qquad u': \quad \begin{array}{ccccc} \alpha_1 & \xrightarrow{a} & \alpha_2 & \xrightarrow{a} & \alpha_3 \\ \Downarrow & & \Downarrow & & \Downarrow \\ o_{\alpha_1} & & o_{\alpha_2} & & o_{\alpha_3} \end{array}$$

The state spaces of the aforementioned Moore machines consist of the functions:

$$\begin{aligned} \varphi_1 &= \eta(p') = \{p' \to 1,\ q' \to 0,\ r' \to 0,\ s' \to 0,\ t' \to 0\} \\ \varphi_2 &= \{p' \to 0,\ q' \to x,\ r' \to 1-x,\ s' \to 0,\ t' \to 0\} \\ \varphi_3 &= \{p' \to 0,\ q' \to 0,\ r' \to 0,\ s' \to 1,\ t' \to 1\} \\ \alpha_1 &= \eta(u') = \{u' \to 1,\ v' \to 0,\ w' \to 0,\ w'' \to 0\} \\ \alpha_2 &= \{u' \to 0,\ v' \to 1,\ w' \to 0,\ w'' \to 0\} \\ \alpha_3 &= \{u' \to 0,\ v' \to 0,\ w' \to y,\ w'' \to 1-y\}. \end{aligned}$$

The associated observations are:

$$o_{\varphi_1} = o_{\alpha_1} = o_{\varphi_2} = o_{\alpha_2} = (\emptyset \mapsto 0, \{a\} \mapsto 1), o_{\varphi_3} = o_{\alpha_3} = (\emptyset \mapsto 1, \{a\} \mapsto 0.)$$

The functions φ_2, φ_3, α_2 and α_3 together with their outputs are easily determined based on the operations of the corresponding Moore coalgebra (as depicted in Figure 9).

The connection between the behaviour, i.e., ready function of p' (respectively, u') and φ_i (respectively, α_i), for $i \in \{1,2,3\}$, is straightforward. Each of the functions φ_1, φ_2 and φ_3 captures the behaviour of the system starting from p', after executing the traces ε, a and aa, respectively. Note that, for example, the values of the ready function for trace ε and ready sets $\emptyset$ and $\{a\}$, respectively, are in one to one correspondence with the assignments in o_{φ_1}. Similarly for the case of u'.

By following the same approach, the coalgebraic machinery provides an "yes' answer w.r.t. (maximal) failure equivalence of p' and u' as well. This is also in agreement with the results in Jou and Smolka (1990) stating that ready and (maximal) failure equivalence for GPSs have the same distinguishing power.

The equivalence between the coalgebraic and the original definitions of the decorated trace semantics $\mathcal{I} \in \{\mathcal{R}_p, \mathcal{F}_p, \mathcal{MF}_p\}$ in Jou and Smolka (1990) consists in showing that, given a GPS (X, δ), $x \in X$, $w \in A^*$ and $S \subseteq A$, it holds that $[\![\eta(x)]\!](w)(S) = I(x)(w, S)$.

Theorem 4.1. Let $(X, \delta : X \to (\mathcal{D}_\omega(X))^A)$ be a GPS and $(\mathcal{D}_\omega(X), \langle o, t\rangle)$ be its associated determinization as in Figure 9. Then, for all $x \in X$, $w \in A^*$ and $S \subseteq A$, it holds

$$[\![\eta(x)]\!](w)(S) = \mathcal{I}(x)(w, S).$$

Proof. The proof is similar for all $\mathcal{I}$ in $\{\mathcal{R}_p, \mathcal{F}_p, \mathcal{MF}_p\}$, by induction on $w \in A^*$.

— *Base case* – $w = \varepsilon$: $[\![\eta(x)]\!](\varepsilon)(S) = \overline{o}_{\mathcal{I}}(x)(S) = \mathcal{I}(x)(\varepsilon, S)$.

— *Induction step.* Here, we will use the fact that the map into the final coalgebra is also an algebra map and the equality

$$\mathcal{I}(x)(aw, S) = \sum_{y \in Y} \mu_a(x, y) \times \mathcal{I}(x)(w)(S).$$

Consider $aw \in A^*$ and assume $[\![\eta(y)]\!](w)(S) = \mathcal{I}(y)(w, S)$, for all $y \in X$. We want to prove that $[\![\eta(x)]\!](aw)(S) = \mathcal{I}(x)(aw)(S)$, for $a \in A$.

$$\begin{aligned}
[\![\eta(x)]\!](aw)(S) &= [\![\delta(x)(a)]\!](w)(S) \\
&= \sum_{y \in Y} \delta(x)(a)(y) \times [\![\eta(y)]\!](w)(S) && ([\![\text{ - }]\!] \text{ is an algebra map}) \\
&= \sum_{y \in Y} \delta(x)(a)(y) \times \mathcal{I}(x)(w)(S) && \text{(IH)} \\
&= \sum_{y \in Y} \mu_a(x, y) \times \mathcal{I}(x)(w)(S) && (\mu_a(x, x') = \delta(x)(a)(x')) \\
&= \mathcal{I}(x)(aw)(S).
\end{aligned}$$

□

4.2. *(Maximal) trace semantics*

In this section, we provide the coalgebraic modelling of (maximal) trace semantics for GPSs. The approach resembles the one in the previous section: we first recall the aforementioned semantics as introduced in Jou and Smolka (1990), and then show how to instantiate the ingredients of Figure 9 in order to capture the corresponding behaviours in terms of (final) Moore coalgebras. As a last step, we prove the equivalence between the coalgebraic modellings and the original definitions in Jou and Smolka (1990).

Definition 4.4 ((Maximal) trace equivalence (Jou and Smolka 1990)).
The *trace function* $\mathcal{T}_p : X \to (A^* \to [0, 1])$ is given by

$$\mathcal{T}_p(x)(w) = \sum_{y \in X} \mu_w(x, y).$$

The *maximal trace function* $\mathcal{MT}_p : X \to [0,1]^{A^*}$ is given by $\mathcal{MT}_p(x)(w) = \mu_{w0}(x, \mathbf{0})$.

We say that $x, x' \in X$ are *trace (resp. maximal) equivalent* whenever $\mathcal{T}_p(x) = \mathcal{T}_p(x')$ (resp. $\mathcal{MT}_p(x) = \mathcal{MT}_p(x')$).

From the definition above, it can be easily seen at an intuitive level that trace equivalence identifies processes that can execute with the same probability the same sets of traces $w \in A^*$. Moreover, maximal trace equivalence takes into consideration the probability of not triggering any action after the performance of such w's.

Therefore, we choose the set of observations $B_{\mathcal{I}}$ (where $\mathcal{I} = \mathcal{T}_p$ for trace and $\mathcal{I} = \mathcal{MT}_p$ for maximal trace semantics) to denote probabilities (of processes to execute $w \in A^*$, or stagnate after triggering such w's) ranging over $[0,1]$.

We define the 'decorating' functions, for $\mathcal{I} \in \{\mathcal{T}_p, \mathcal{MT}_p\}$, $\overline{o}_{\mathcal{I}} : X \to [0,1]$ by

$$\overline{o}_{\mathcal{T}_p}(x) = 1 \qquad\qquad \overline{o}_{\mathcal{MT}_p}(x) = \mu_0(x, \mathbf{0}).$$

The (Moore) output function o is given by, for all $\varphi \in \mathcal{D}_\omega(X)$,

$$o(\varphi) = \sum_{x \in supp(\varphi)} \varphi(x) \times \overline{o}_{\mathcal{I}}(x).$$

We can now show the equivalence between the coalgebraic and the original definition of (maximal) trace semantics.

Theorem 4.2. Let $(X, \delta : X \to (\mathcal{D}_\omega(X))^A)$ be a GPS and $(\mathcal{D}_\omega(X), \langle o, t\rangle)$ be its associated determinization as in Figure 9. Then, for all $x \in X$ and $w \in A^*$:

$$[\![\eta(x)]\!](w) = \mathcal{I}(x)(w).$$

Proof. By induction on $w \in A^*$, similar to Theorem 4.1. □

Consider, for instance, the systems p' and u' in Example 4.1. They are trace equivalent as they both can execute traces ε, a and aa with total probability 1. Consequently, they are maximal trace equivalent as well: for sequences ε and a, their associated maximal trace functions compute value 0, whereas for aa the latter return value 1.

The same answer w.r.t. (maximal) trace equivalence of p' and u' is obtained by reasoning on bisimilarity of their associated determinizations derived according to the powerset construction. It is easy to check that in the current setting, the Moore automata corresponding to φ_1 and α_1 in Example 4.1 output

— in the case of trace: $o_{\varphi_i} = o_{\alpha_i} = 1$, for all $i \in \{1,2,3\}$;
— in the case of maximal trace: $o_{\varphi_i} = o_{\alpha_i} = 0$, for $i \in \{1,2\}$ and $o_{\varphi_3} = o_{\alpha_3} = 1$.

Therefore φ_1 and α_1 are bisimilar. Hence, p' and u' are (maximal) trace equivalent.

5. In a nutshell: decorated trace equivalences for LTSs and GPSs

Next we provide a more compact overview on the coalgebraic machineries introduced in Sections 3 and 4. This also in order to emphasize on the generality and uniformity of our coalgebraic framework.

Recall that for each of the decorated trace semantics we first instantiate the constituents of Figure 2 (summarizing the generalized powerset construction). Moreover, for the case of LTSs, the original definitions of the semantics under consideration are provided with equivalent representations in terms of functions $\varphi_Y^{\mathcal{I}}$, paving the way to their interpretation in terms of final Moore coaglebras.

All these are summarized in Figure 10, for an arbitrary LTS $(X, \delta : X \to (\mathcal{P}_\omega X)^A)$ and an arbitrary GPS $(X, \delta : X \to (\mathcal{D}_\omega X)^A)$.

Once the ingredients of Figure 2 and, for LTSs, functions $\varphi_Y^{\mathcal{I}}$ are defined, we formalize the equivalence between the coalgebraic modelling of $\mathcal{I}$-semantics and its original definition.

For the case of LTSs, for $\mathcal{I}$ ranging over $\mathcal{T}, \mathcal{CT}, \mathcal{F}, \mathcal{R}, \mathcal{PF}, \mathcal{RT}$ and $\mathcal{FT}$, we show that the following result holds:

Theorem 5.1. Let $(X, \delta : X \to (\mathcal{P}_\omega X)^A)$ be an LTS. For all $x \in X$, $[\![\{x\}]\!] = \varphi_x^{\mathcal{I}} \cong \mathcal{I}(x)$.

Orthogonally, for the case of GPSs, for $\mathcal{I}$ ranging over $\mathcal{R}_p, \mathcal{F}_p, \mathcal{MF}_p, \mathcal{T}_p$ and $\mathcal{MT}_p$, we prove the following:

Theorem 5.2. Let $(X, \delta : X \to (\mathcal{D}_\omega X)^A)$ be a GPS. For all $x \in X$, $[\![\eta(x)]\!] = \mathcal{I}(x)$.

For each of the semantics under consideration, the proofs of Theorems 5.1 and 5.2, follow by induction on words over the corresponding action alphabet. For more details see the proof of Theorem 3.1 in Section 3.1 (for LTSs) and Theorem 4.1 in Section 4.1 (for GPSs), respectively.

Remark 5.1. It is worth observing that by instantiating T with the identity functor, $\mathcal{F}$ with $\mathcal{P}_\omega(-)^A$ and, respectively, $\mathcal{D}_\omega(-)^A$ in (3) one gets the coalgebraic modelling of the standard notion of bisimilarity for LTSs and, respectively, GPSs.

Concrete examples on how to use the coalgebraic frameworks are provided for each of the decorated trace semantics. We show how to derive determinizations of LTSs and GPSs in terms of Moore automata, which eventually are used to reason on the corresponding equivalences in terms of Moore bisimulations.

6. Canonical representatives

Given a *decorated* system $(X, \langle \overline{o}_{\mathcal{I}}, \delta \rangle)$, we showed in the previous sections how to construct a determinization $(T(X), \langle o, t \rangle)$, with $T = \mathcal{P}_\omega$ for the case of LTSs, and $T = \mathcal{D}_\omega$ for GPSs, respectively. The map $[\![-]\!] : TX \to B_{\mathcal{I}}^{A^*}$ provides us with a *canonical representative* of the behaviour of each state in TX. The image (C, δ') of $(TX, \langle o, t \rangle)$, via the map $[\![-]\!]$, can be viewed as the minimization w.r.t. the equivalence $\mathcal{I}$.

Recall that the states of the final coalgebra $(B_{\mathcal{I}}^{A^*}, \langle \epsilon, (-)_a \rangle)$ are functions $\varphi : A^* \to B_{\mathcal{I}}$ and that their decorations and transitions are given by the functions $\epsilon : B_{\mathcal{I}}^{A^*} \to B_{\mathcal{I}}$ and $(-)_a : B_{\mathcal{I}}^{A^*} \to (B_{\mathcal{I}}^{A^*})^A$, defined in Section 2. The states of the canonical representative (C, δ') are also functions $\varphi : A^* \to B_{\mathcal{I}}$, i.e., $C \subseteq B_{\mathcal{I}}^{A^*}$. Moreover, the function $\delta' : C \to B_{\mathcal{I}} \times C^A$ is simply the restriction of $\langle \epsilon, (-)_a \rangle$ to C, that means $\delta'(\varphi) = \langle \varphi(\epsilon), (\varphi)_a \rangle$ for all $\varphi \in C$.

$\mathcal{I}$	$B_{\mathcal{I}}$	$\bar{o}_{\mathcal{I}} : X \to B_{\mathcal{I}}$	$\mathcal{I}$	$B_{\mathcal{I}}$	$\bar{o}_{\mathcal{I}} : X \to B_{\mathcal{I}}$
$\mathcal{R}$	$\mathcal{P}_\omega(\mathcal{P}_\omega A)$	$\bar{o}_{\mathcal{R}}(x) = \{I(\delta(x))\}$	$\mathcal{FT}$	$\mathcal{P}_\omega(\mathcal{P}_\omega A)$	$\bar{o}_{\mathcal{FT}}(x) = Fail(\delta(x))$
$\mathcal{F}$	$\mathcal{P}_\omega(\mathcal{P}_\omega A)$	$\bar{o}_{\mathcal{F}}(x) = Fail(\delta(x))$	$\mathcal{R}_p$	$[0,1]^{\mathcal{P}_\omega(A)}$	$\bar{o}_{\mathcal{R}_p}(x)(I) = \begin{cases} 1 & \text{if } I = I(x) \\ 0 & \text{otherwise} \end{cases}$
$\mathcal{T}$	2	$\bar{o}_{\mathcal{T}}(x) = 1$	$\mathcal{F}_p$	$[0,1]^{\mathcal{P}_\omega(A)}$	$\bar{o}_{\mathcal{F}_p}(x)(Z) = \begin{cases} 1 & \text{if } Z \cap I(x) = \varnothing \\ 0 & \text{otherwise} \end{cases}$
$\mathcal{CT}$	2	$\bar{o}_{\mathcal{CT}}(x) = \begin{cases} 1 & \text{if } I(\delta(x)) = \varnothing \\ 0 & \text{otherwise} \end{cases}$	$\mathcal{MF}_p$	$[0,1]^{\mathcal{P}_\omega(A)}$	$\bar{o}_{\mathcal{MF}_p}(x)(Z) = \begin{cases} 1 & \text{if } Z = A - I(x) \\ 0 & \text{otherwise} \end{cases}$
$\mathcal{PF}$	$\mathcal{P}(\mathcal{P}A^*)$	$\bar{o}_{\mathcal{PF}}(x) = \{\mathcal{T}(x)\}$	$\mathcal{T}_p$	$[0,1]$	$\bar{o}_{\mathcal{T}_p}(x) = 1$
$\mathcal{RT}$	$\mathcal{P}_\omega(\mathcal{P}_\omega A)$	$\bar{o}_{\mathcal{RT}}(x) = \{I(\delta(x))\}$	$\mathcal{MT}_p$	$[0,1]$	$\bar{o}_{\mathcal{MT}_p}(x) = \mu_0(x, \mathbf{0})$

Fig. 10. The coalgebraic framework in a nutshell.

Finally, it is interesting to observe that for LTS $B_{\mathcal{I}}^{A^*}$ carries a semilattice structure (inherited from $B_{\mathcal{I}}$) and that $[\![-]\!] : \mathcal{P}_\omega X \to B_{\mathcal{I}}^{A^*}$ is a semilattice homomorphism. From this observation, it is immediate to conclude that also C is a semilattice, but it is not necessarily freely generated, i.e., it is not necessarily a powerset. Similarly, for GPS $B_{\mathcal{I}}^{A^*}$ carries a positive convex algebra structure (these are the $\mathcal{D}_\omega$-algebras) and $[\![-]\!] : \mathcal{D}_\omega X \to B_{\mathcal{I}}^{A^*}$ is a positive convex algebra homomorphism. Again, from this observation, we know that also C is a positive convex algebra (not necessarily freely generated).

7. Bisimulation up-to

As previously stated in the beginning of this paper, when reasoning on behavioural equivalence it is preferable to use relations as small as possible, that are not necessarily bisimulations, but contained in a bisimulation relation. These relations are referred to as *bisimulations up-to* (Sangiorgi and Rutten 2011).

In what follows, we exploit the generalized powerset construction summarized in Figure 2 and define bisimulation up-to context in the setting of decorated LTSs determinized in terms of Moore automata. This comes as an extension of the recent work in Bonchi and Pous (2013). Similar observations hold also for GPSs, but we do not exploit them here.

Let $L_{dec} = (X, \langle \bar{o}_{\mathcal{I}}, id \rangle \circ \delta : X \to B_{\mathcal{I}} \times (\mathcal{P}_\omega X)^A)$ be a decorated (possibly 'preprocessed') LTS and $(\mathcal{P}_\omega X, \langle o, t \rangle : \mathcal{P}_\omega X \to B_{\mathcal{I}} \times (\mathcal{P}_\omega X)^A)$ its associated Moore automaton, as in Figure 2. A *bisimulation up-to context* for L_{dec} is a relation $R \subseteq (\mathcal{P}_\omega X) \times (\mathcal{P}_\omega X)$ such that:

$$X_1 \; R \; X_2 \Rightarrow \begin{cases} o(X_1) = o(X_2) \\ (\forall a \in A) \, . \, t(X_1)(a) \; c(R) \; t(X_2)(a) \end{cases} \tag{13}$$

where $c(R)$ is the smallest relation which is closed with respect to set union and which includes R, inductively defined by the following inference rules:

$$\frac{}{\varnothing\ c(R)\ \varnothing} \qquad \frac{X\,R\,Y}{X\,c(R)\,Y} \qquad \frac{X_1\,c(R)\,Y_1 \quad X_2\,c(R)\,Y_2}{X_1 \cup X_2\ c(R)\ Y_1 \cup Y_2} \tag{14}$$

Remark 7.1. Observe that by replacing $c(R)$ with R in (13) one gets the definition of *Moore bisimulation.*

Theorem 7.1. Any bisimulation up-to context for decorated LTSs is included in a bisimulation relation.

Proof. The proof consists in showing that for any bisimulation up-to context R, $c(R)$ is a bisimulation relation (recall that $R \subseteq c(R)$). The result follows by structural induction, as shown below.

Let $L_{dec} = (X, \delta^{\sharp} : X \to B_{\mathcal{I}} \times (\mathcal{P}_{\omega}X)^A)$ be a decorated LTS and $(\mathcal{P}_{\omega}X, \langle o, t\rangle : \mathcal{P}_{\omega}X \to B_{\mathcal{I}} \times (\mathcal{P}_{\omega}X)^A)$ be its associated Moore automaton, derived according to the powerset construction. Let R be a bisimulation up-to context for L_{dec}.

In what follows we want to prove that $c(R)$ is a bisimulation relation (that includes R, by (14)).

We have to show that

$$X\ c(R)\ Y \Rightarrow \begin{cases} o(X) = o(Y) \\ (\forall a \in A)\,.\,t(X)(a)\ c(R)\ t(Y)(a). \end{cases} \tag{15}$$

We proceed by structural induction.

1. Let $X\ R\ Y$. Then (15) holds by definition.
2. Let $X = X_1 \cup X_2$ and $Y = Y_1 \cup Y_2$ such that $X_1\,c(R)\ Y_1$ and $X_2\,c(R)\ Y_2$. By induction, we have that $o(X_1) = o(Y_1)$ and $o(X_2) = o(Y_2)$. We now need to prove that $o(X) = o(Y)$.

$$o(X) = o(X_1 \cup X_2) = o(X_1) \cup o(X_2) \stackrel{\text{IH}}{=} o(Y_1) \cup o(Y_2) = o(Y_1 \cup Y_2) = o(Y).$$

We also have, by induction, that, for all $a \in A$

$$t(X_1)(a)\ c(R)\ t(Y_1)(a) \quad \text{and} \quad t(X_2)(a)\ c(R)\ t(Y_2)(a).$$

Hence, for all $a \in A$, we can easily prove that $t(X)(a)\ c(R)\ t(Y)(a)$:

$$\begin{array}{rcl} t(X)(a) = t(X_1 \cup X_2)(a) & = & t(X_1)(a) \cup t(X_2)(a) \quad \text{(IH)} \\ & c(R) & t(Y_1)(a) \cup t(Y_2)(a) \\ & = & t(Y_1 \cup Y_2)(a) = t(Y)(a). \end{array}$$

Hence, $c(R) \supseteq R$ is a bisimulation relation, as (15) holds for all $(X, Y) \in c(R)$.

□

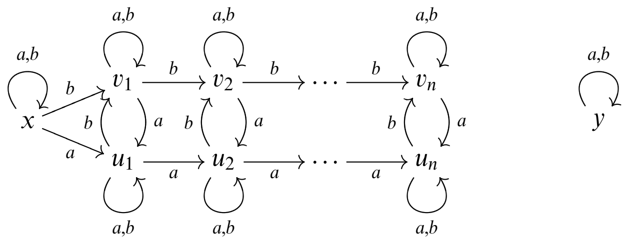
Remark 7.2. Based on (1), (2) and Theorem 7.1, verifying behavioural equivalence of two states x_1, x_2 in a decorated LTS consists in identifying a bisimulation up-to context R^c relating $\{x_1\}$ and $\{x_2\}$:

$$[\![\{x_1\}]\!] = [\![\{x_2\}]\!] \text{ iff } \{x_1\}\ R^c\ \{x_2\}. \tag{16}$$

Also note that Theorem 7.1 is not a very different, but useful generalization of Theorem 2 in Bonchi and Pous (2013) to the context of decorated LTSs.

Example 7.1. We provide an example of applying the generalized powerset construction and bisimulation up-to context for reasoning on decorated trace equivalence of LTSs.

Consider the following systems, where n is an arbitrary natural number:



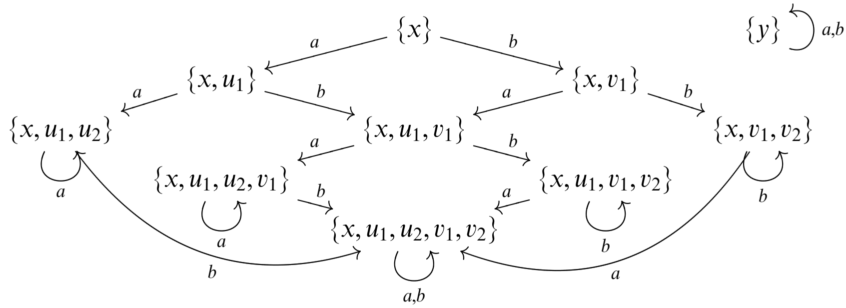
It is easy to see that x and y are bisimilar: intuitively, all the states of the automata depicted above can trigger actions a and b as a first step and, moreover, all their subsequent transitions lead to states with the same behaviour. Therefore x and y are also $\mathcal{I}$-equivalent for $\mathcal{I}$ ranging over $\mathcal{T}, \mathcal{CT}, \mathcal{F}, \mathcal{R}, \mathcal{PF}, \mathcal{RT}$ and $\mathcal{FT}$, according to the lattice of semantic equivalences in Figure 1.

The coalgebraic machinery provides an 'yes' answer w.r.t. $\mathcal{I}$-equivalence of the two LTSs as well. After determinization, $\{x\}$ can reach all states of shape: $\{x\} \cup \overline{u}_i$, $\{x\} \cup \overline{v}_i$, $\{x\} \cup \overline{u}_i \cup \overline{v}_i$, for $i \in \{1, \dots, n\}$ and $\{x\} \cup \overline{u}_j \cup \{v_1\}$, $\{x\} \cup \overline{v}_j \cup \{u_1\}$, respectively, for $j \in \{2, \dots, n\}$. (We write, for example, $\overline{u}_i$ in order to represent the set $\{u_1, u_2, \dots, u_i\}$.)

Consequently, the generalized powerset construction associates to x a Moore automaton consisting of $5n-1$ states, whereas the determinization of y has only one state. Hence, the (Moore) bisimulation relation R including $(\{x\}, \{y\})$ consists of $5n-1$ pairs as follows:

$$R = \{(\{x\}, \{y\})\} \cup \{(\{x\} \cup \overline{u}_i \cup \{v_1\}, \{y\}), (\{x\} \cup \overline{v}_i \cup \{u_1\}, \{y\}) \mid i \in \{2, \dots, n\}\} \cup \\ \{(\{x\} \cup \overline{u}_i, \{y\}), (\{x\} \cup \overline{v}_i, \{y\}), (\{x\} \cup \overline{u}_i \cup \overline{v}_i, \{y\}) \mid i \in \{1, \dots, n\}\}. \quad (17)$$

For a better intuition, we illustrate below the determinizations starting from x and y, for the case $n = 3$:



It is easy to see that the bisimulation relating $\{x\}$ and $\{y\}$ consists of all pairs $(X, \{y\})$, with X ranging over the state space of the Moore automaton derived according to the generalized powerset construction, starting with $\{x\}$.

Observe that all the pairs in R in (17) can be 'generated' from $(\{x\},\{y\})$, $(\{x\}\cup\overline{u}_i,\{y\})$ and $(\{x\}\cup\overline{v}_i,\{y\})$ by iteratively applying the rules in (14). Therefore, for an arbitrary natural number n, the bisimulation up-to context stating the equivalence of x and y is:

$$R^c = \{(\{x\},\{y\})\} \cup \{(\{x\}\cup\overline{u}_i,\{y\}),\, (\{x\}\cup\overline{v}_i,\{y\}) \mid i \in \{1,\ldots,n\}\}$$

and consists of only $2n+1$ pairs.

8. Conclusions and future work

In this paper, we have proved that the coalgebraic characterizations of decorated trace semantics for labelled transition systems and GPSs, respectively, are equivalent with the corresponding standard definitions in van Glabbeek (2001) and Jou and Smolka (1990). More precisely, we have shown that for a state x, the coalgebraic canonical representative $[\![\{x\}]\!]$, given by determinization and finality, coincides with the classical semantics $\mathcal{I}(x)$, for $\mathcal{I}$ ranging over $\mathcal{T},\mathcal{CT},\mathcal{R},\mathcal{F},\mathcal{PF},\mathcal{RT}$ and $\mathcal{FT}$, representing the traces, complete traces, ready pairs, failure pairs, possible futures, ready traces and respectively failure traces of x in a labelled transition system. Similar equivalences have been proven for $\mathcal{I}$ ranging over $\mathcal{R}_p,\mathcal{F}_p,\mathcal{MF}_p,\mathcal{T}_p$ and $\mathcal{MT}_p$ representing the ready, failure, maximal failure, trace and maximal trace functions for the case of probabilistic systems.

In addition, we have illustrated how to reason about decorated trace equivalence using coinduction, by constructing suitable bisimulations up-to context. This is a very efficient sound and complete proof technique, and represents an important step towards automated reasoning, as it opens the way for the use of, for instance, coinductive theorem provers such as CIRC (Rosu and Lucanu 2009). Last, but not least, we showed that the spectrum of decorated trace semantics can be recovered from the coalgebraic modelling.

Bisimulation up-to is a technique that has recently received renewed attention (Bonchi and Pous 2013; Rot 2013). The coalgebraic treatment thereof was originally studied by Lenisa (Cancila *et al.* 2003; Lenisa 1999) and further explored by Bartels (2004).

A coalgebraic characterization of the spectrum, not based on the powerset construction, was attempted in Monteiro (2008). The approach in Monteiro (2008) is based on an abstract notion of 'behaviour object' that has similar properties with final objects. It is not clear, however, how this approach could be modularly extended so to treat probabilistic decorated traces.

A similar idea of system determinization was also applied in Cleaveland and Hennessy (1993), in a non-coalgebraic setting, for the case of testing semantics where *must testing* coincides with failure semantics in the absence of divergence. The approach in Cleaveland and Hennessy (1993) is very similar to ours but it is restricted only to the case of testing semantics. Our use of coalgebraic techniques allows us to treat more decorated traces and also decorated probabilistic traces in essentially the same manner. Still in the context of must testing, a coalgebraic outlook is presented in Boreale and Gadducci (2006) which introduces a fully abstract semantics for CSP. The main difference with our work consists in the fact that Boreale and Gadducci (2006) build a coalgebra from the syntactic terms of CSP, while here we build a coalgebra starting from LTSs via the generalized powerset construction (Silva *et al.* 2010). Moreover, they only consider must testing and leave as future work capturing other decorated traces. In another paper (Bonchi *et al.* 2013), we have shown that must testing can also be captured using the generalized powerset construction. An important point is that our approach puts in evidence the underlying semilattice structure which is needed for defining bisimulations up-to whereas this is not at all considered in their paper. An interesting direction for future work would be to explore combinations of both approaches: on the one hand, apply up-to techniques to the their work; on the other hand, consider in our setting processes specified by a certain syntax and generate the (determinized) LTS directly from the expression specifying the process' behaviour. This would yield a coinductive approach to denotational (linear-time) semantics of different kinds of processes calculi.

There are several other possible directions for future work. One option is to investigate whether we can derive efficient algorithms implementing the proof techniques for reasoning on decorated trace equivalences of labelled transition systems and GPSs, in an uniform fashion.

Orthogonally, it would be worth investigating whether there exists a coalgebraic representation of system equivalences characterized by testing scenarios, or temporal logics, along the lines in van Glabbeek (2001).

Moreover, we aim at providing coalgebraic modellings for the remaining semantics of the spectrum in van Glabbeek (2001), and come up with a new representation of possible-futures semantics. The latter is motivated by the current drawback of storing for each state of the LTSs the corresponding set of traces. In this context it might be more appropriate considering the definition of possible-futures semantics given in terms of nested bisimulations (Hennessy and Milner 1985), rather than the set-theoretic one in van Glabbeek (2001).

Acknowledgments

We thank the anonymous reviewers, Luca Aceto and Anna Ingólfsdóttir for their constructive comments and references to the literature. The work of Georgiana Caltais has been partially supported by a CWI Internship and by the project 'Meta-theory of Algebraic Process Theories' (nr. 100014021) of the Icelandic Research Fund. The work of Alexandra Silva is partially funded by the ERDF through the Programme COMPETE and by the Portuguese Government through FCT – Foundation for Science and Technology, project ref. `PTDC/EIA-CCO/122240/2010` and `SFRH/BPD/71956/2010`.

References

Aceto, L., Fokkink, W. and Verhoef, C. (1999) Structural operational semantics. In: *Handbook of Process Algebra*, Elsevier 197–292.

Awodey, S. (2010) *Category theory*, Oxford Logic Guides, Oxford University Press.

Bartels, F. (2004) *On Generalised Coinduction and Probabilistic Specification Formats*, Ph.D. thesis, CWI, Amsterdam.

Bartels, F., Sokolova, A. and de Vink, E. P. (2004) A hierarchy of probabilistic system types. *Theoretical Computer Science* **327** (1–2) 3–22.

Bonchi, F., Bonsangue, M. M., Caltais, G., Rutten, J. J. M. M. and Silva, A. (2012) Final semantics for decorated traces. *Electronic Notes in Theoretical Computer Science* **286** 73–86.

Bonchi, F., Caltais, G., Pous, D. and Silva, A. (2013) Brzozowski's and up-to algorithms for must testing. In: Chieh Shan, C. (ed.) Asian Symposium on Programming Languages and Systems. *Springer Lecture Notes in Computer Science* **8301** 1–16.

Bonchi, F. and Pous, D. (2013) Checking NFA equivalence with bisimulations up to congruence. In: Giacobazzi, R. and Cousot, R. (eds.) *Principles of Programming Languages*, ACM 457–468.

Boreale, M. and Gadducci, F. (2006) Processes as formal power series: A coinductive approach to denotational semantics. *Theoretical Computer Science* **360** (1–3) 440–458.

Cancila, D., Honsell, F. and Lenisa, M. (2003) Generalized coiteration schemata. *Electronic Notes in Theoretical Computer Science* **82** (1) 79–93.

Cleaveland, R. and Hennessy, M. (1993) Testing equivalence as a bisimulation equivalence. *Formal Aspects of Computing* **5** (1) 1–20.

Doberkat, E.-E. (2008) Erratum and addendum: Eilenberg–Moore algebras for stochastic relations. *Information and Computation* **206** (12) 1476–1484.

Hasuo, I., Jacobs, B. and Sokolova, A. (2007) Generic trace semantics via coinduction. *Logical Methods in Computer Science* **3** (4).

Hennessy, M. and Milner, R. (1985) Algebraic laws for nondeterminism and concurrency. *Journal of the ACM* **32** (1) 137–161.

Jou, C.-C. and Smolka, S. (1990) Equivalences, congruences, and complete axiomatizations for probabilistic processes. In: Baeten J. and Klop, J. (eds.) International conference on concurrency theory Theories of Concurrency: Unification and Extension. *Springer Lecture Notes in Computer Science* **458**, 367–383.

Lenisa, M. (1999) From set-theoretic coinduction to coalgebraic coinduction: Some results, some problems. *Electronic Notes in Theoretical Computer Science* **19** 2–22.

Lenisa, M., Power, J. and Watanabe, H. (2000) Distributivity for endofunctors, pointed and co-pointed endofunctors, monads and comonads. *Electronic Notes in Theoretical Computer Science* **33** 230–260.

Milner, R. (1989) *Communication and Concurrency*, Prentice Hall.

Monteiro, L. (2008) A coalgebraic characterization of behaviours in the linear time – branching time spectrum. In: Corradini, A. and Montanari, U. (eds.) Workshop on Algebraic Development Techniques. *Springer Lecture Notes in Computer Science* **5486** 251–265.

Park, D. M. R. (1981) Concurrency and automata on infinite sequences. In: Deussen, P. (ed.) *Theoretical Computer Science*. *Springer Lecture Notes in Computer Science* **104** 167–183.

Rot, J., Bonsangue, M. M. and Rutten, J. J. M. M. (2013) Coalgebraic bisimulation-up-to. In: van Emde Boas, P., Groen, F. C. A., Italiano, G. F., Nawrocki, J. R. and Sack, H. (eds.) SOFtware SEMinar. *Springer Lecture Notes in Computer Science* **7741** 369–381.

Rosu, G. and Lucanu, D. (2009) Circular coinduction: A proof theoretical foundation. In: *Conference on Algebra and Coalgebra in Computer Science*, 127–144.

Rutten, J. J. M. M. (2000) Universal coalgebra: A theory of systems. *Theoretical Computer Science* **249** (1) 3–80.

Sangiorgi, D. and Rutten, J. (2011) *Advanced Topics in Bisimulation and Coinduction*, Cambridge Tracts in Theoretical Computer Science, Cambridge University Press.

Silva, A., Bonchi, F., Bonsangue, M. M. and Rutten, J. J. M. M. (2010) Generalizing the powerset construction, coalgebraically. In: Lodaya, K. and Mahajan, M. (eds.), Conference of Foundations of Software Technology and Theoretical Computer Science. *Leibniz International Proceedings in Informatics* **8** 272–283.

Silva, A., Bonchi, F., Bonsangue, M. M. and Rutten, J. J. M. M. (2013) Generalizing determinization from automata to coalgebras. *Logical Methods in Computer Science* **9** (1).

van Glabbeek, R. (2001) The linear time – branching time spectrum I. The semantics of concrete, sequential processes. In: Bergstra, J., Ponse, A. and Smolka, S. (eds.) *Handbook of Process Algebra*, Elsevier 3–99.