Erschienen in: Electronic Notes in Theoretical Computer Science ; 286 (2012). - S. 73-86
https://dx.doi.org/10.1016/j.entcs.2012.08.006

Available online at www.sciencedirect.com

SciVerse ScienceDirect

Electronic Notes in Theoretical Computer Science 286 (2012) 73–86

Electronic Notes in Theoretical Computer Science

www.elsevier.com/locate/entcs

Final Semantics for Decorated Traces

Filippo Bonchi[a], Marcello Bonsangue[b,c], Georgiana Caltais[d,c], Jan Rutten[e,c], Alexandra Silva[e,c,f]

[a] *ENS Lyon, Université de Lyon, LIP (UMR 5668 CNRS ENS Lyon UCBL INRIA)*

[b] *LIACS - Leiden University, The Netherlands*

[c] *Centrum voor Wiskunde en Informatica, The Netherlands*

[d] *School of Computer Science - Reykjavik University, Iceland*

[e] *Radboud University Nijmegen, The Netherlands*

[f] *HASLab / INESC TEC, Universidade do Minho, Braga, Portugal*

Abstract

In concurrency theory, various semantic equivalences on labelled transition systems are based on traces enriched or *decorated* with some additional observations. They are generally referred to as *decorated traces*, and examples include ready, failure, trace and complete trace equivalence. Using the generalized powerset construction, recently introduced by a subset of the authors [13], we give a coalgebraic presentation of decorated trace semantics. This yields a uniform notion of canonical, 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), which is helpful for automation.

Keywords: Labelled transition systems, decorated traces, coalgebras, final Moore automata

1 Introduction

The study of systems equivalence has been an interesting research topic for many years now. Several equivalences have been proposed throughout the years, each of which suitable for use in different contexts of application. Many of the equivalences that are important in the theory of concurrency were described in the well-known paper by van Glabbeek [14].

[1] 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. We thank Luca Aceto and Anna Ingólfsdóttir for comments and references to the literature.

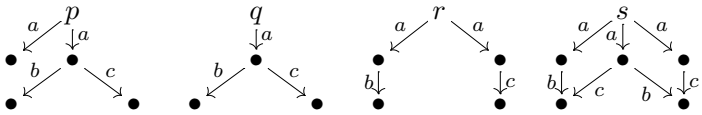
doi:10.1016/j.entcs.2012.08.006

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 what van Glabbeek calls *decorated trace semantics*, which includes (complete) trace, ready and failure semantics.

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 $f\colon X \to \mathcal{F}(X)$ 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 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 [10], in that no two different states are equivalent.

Labelled transition systems (LTS's) can be modelled as coalgebras for the functor $\mathcal{F}(X) = (\mathcal{P}_\omega X)^A$ and the canonical behavioural equivalence associated with $\mathcal{F}$ is precisely the finest equivalence of the spectrum in [14]. In the recent past, other equivalences of the spectrum have been also cast in the coalgebraic framework. Notably, trace semantics was widely studied [5,13] and, more recently, decorated trace semantics was recovered via a coalgebraic generalization of the classical powerset construction [12].

To get some intuition on the type of distinctions the equivalences above encompass, consider the following labelled transition systems over the alphabet $A = \{a, b, c\}$:



The traces of the states p, q, r and s are $\{a, ab, ac\}$, and therefore they are all trace equivalent. Complete trace semantics identifies states that have the same set of complete traces, that is, traces that lead to states where no further action are possible. Of the four states above, q and r and s are complete trace equivalent, but not p since it is the only state that has a as a complete trace. Failure semantics takes into account the set of actions that cannot be fired immediately after the execution of a certain trace. Only r and s are failure equivalent, since after a, state p might not be able to fire actions b and c, whereas p, r and s might not be able to fire only one of b or c and q never fails with those two actions. Ready semantics identifies states according to the set of actions they can trigger immediately after a certain trace has been executed. None of the states above are ready equivalent, since after a only p has the option of not executing any action, q and s can choose from b or c but r cannot and q always has two options b, c whereas s can end in a state where only b or c can be taken.

The contributions of the paper are three-fold. First, we prove that the coalgebraic decorated trace semantics, which are mentioned without proof in [12] as examples, are equivalent to the corresponding set-theoretic notions from [14]. Second, we show how the coalgebraic semantic leads to canonical representatives for the various decorated trace equivalences. Third, we show how to prove decorated trace equivalence using coinduction, by constructing bisimulations (up-to congruence) 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 congruence technique also increases the efficiency of reasoning, as verifications are performed under certain closure properties, which means the bisimulations that are built are smaller (see Section 3, and Section 4 for an example). The techniques we use here for up-to reasoning are an extension of the recent work by the first author [2].

The paper is organized as follows. In Section 2, we provide the basic notions from coalgebra and recall the generalized powerset construction. In Section 3, we show how the powerset construction can be applied for determinizing LTS's in terms of Moore automata $(X, f\colon X \to B \times X^A)$, in order to coalgebraically characterize decorated trace semantics. A detailed description of coalgebraic ready semantics is provided in Section 4. Here we also prove that the obtained coalgebraic model is equivalent to the original definition, and illustrate how one can reason about ready equivalence by constructing bisimulations up-to congruence. By following the approach in Section 4, similar results can be easily shown for (complete) trace and failure semantics coalgebraically modelled as in [12]. Section 5 discusses that the canonical representatives of LTS's we obtain coalgebraically coincide with the minimal LTS's one would obtain by identifying all states equivalent w.r.t. a particular decorated trace semantics. Section 6 contains concluding remarks and discusses future work.

2 Preliminaries

In this section, we briefly recall basic notions from coalgebra and the generalized powerset construction [5,13]. 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 $X \xleftarrow{\pi_1} X \times Y \xrightarrow{\pi_2} Y$. The *disjoint union* of X and Y is written $X + Y$ and has the injection maps $X \xrightarrow{k_1} X + Y \xleftarrow{k_2} Y$. By X^Y we represent the family of *functions* $f\colon Y \to X$, whereas the collection of *finite subsets* of X is denoted by $\mathcal{P}_\omega X$. For each of these operations defined on sets, there is an analogous one on functions (for details see for example [1]). This turns the operations above into (bi)functors, which we shall use throughout this paper.

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 $w_1 w_2$.

Coalgebras: We consider coalgebras of functors $\mathcal{F}$ defined on **Set** – the category of sets and functions. An $\mathcal{F}$-*coalgebra* (or *coalgebra*, when $\mathcal{F}$ is understood) is a

pair $(X, c\colon X \to \mathcal{F}X)$, where $X \in \mathbf{Set}$. 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 $h\colon X \to Y$ preserving the transition structure, *i.e.*, $g \circ h = \mathcal{F}(h) \circ f$.

An $\mathcal{F}$-coalgebra (Ω, ω) is *final* if for any $\mathcal{F}$-coalgebra (X, f) there exists a unique $\mathcal{F}$-homomorphism $[\![-]\!]_X\colon X \to \Omega$. A final coalgebra represents the universe of all possible *behaviours* of $\mathcal{F}$-coalgebras. The unique morphism $[\![-]\!]_X\colon X \to \Omega$ 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. Also it 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). A deterministic automaton over the input alphabet A is a pair $(X, \langle o, t\rangle)$, where X is a set of states and $\langle o, t\rangle\colon X \to 2 \times X^A$ 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 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).

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

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, $t\colon X \to X^A$ is the transition function (like for DA) and $o\colon X \to B$ 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 $\varphi\colon A^* \to B$, $\epsilon\colon B^{A^*} \to B$ maps each φ into $\varphi(\epsilon)$ and $(-)_a\colon B^{A^*} \to (B^{A^*})^A$

is defined for all $\varphi \in B^{A^*}$, $a \in A$ and $w \in A^*$ as $(\varphi)_a(w) = \varphi(aw)$.

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

Coalgebras provide a useful technique for proving behavioural equivalence: *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 $\alpha_R \colon R \to \mathcal{F}R$ such that $\pi_1 \colon R \to X$ and $\pi_2 \colon R \to Y$ are coalgebra homomorphisms. In [10], it is shown that under certain conditions on $\mathcal{F}$ (which are met by all the functors in this paper), bisimulations are a sound and complete proof technique for behavioural equivalence, namely,

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

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 labeled transition systems which are coalgebras for the functor $\mathcal{L}(X) = (\mathcal{P}_\omega X)^A$. The induced behavioural equivalence $\sim_{\mathcal{L}}$ coincides with the standard notion of bisimilarity by Milner and Park [8,6]. However, in concurrency theory, many other equivalences have been studied, notably, *decorated trace equivalences* [14]. Another example is given by non-deterministic automata 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 taken as an intended semantics.

For this reason, a subset of the authors has introduced in [12] the *generalized powerset construction*, for coalgebras $f \colon X \to \mathcal{F}T(X)$ for a functor $\mathcal{F}$ and a monad T, with the proviso that that $\mathcal{F}T(X)$ is an algebra for the monad T. In [12], all the technical details are explored and many interesting instances of the construction are shown. In this paper, we will only be interested in the case where $T = \mathcal{P}_\omega$ and $\mathcal{M}(X) = B \times X^A$, for B a semilattice, and we will therefore only explain the concrete picture for the functor and monad of interest. The fact that we take B to be a semilattice is enough to guarantee that $\mathcal{M}T(X) = B \times (\mathcal{P}_\omega X)^A$ is a semilattice. This fulfills then the proviso above, since semilattices are precisely the algebras of the monad $\mathcal{P}_\omega$.

Given a coalgebra $f \colon X \to \mathcal{M}\mathcal{P}_\omega X$, and because $\mathcal{M}$ has a final coalgebra, we can extend it uniquely to $f^\sharp \colon \mathcal{P}_\omega X \to \mathcal{M}\mathcal{P}_\omega X$ and consider the unique coalgebra homomorphism into the final coalgebra, as summarised by the following diagram:

$$\begin{array}{ccccc} X & \xrightarrow{\{\cdot\}} & \mathcal{P}_\omega X & \xrightarrow{[\![-]\!]} & B^{A^*} \\ {\scriptstyle f}\downarrow & \swarrow_{f^\sharp} & & & \downarrow{\scriptstyle\langle \epsilon,(-)_a\rangle} \\ B \times (\mathcal{P}_\omega X)^A & & \xrightarrow[id_B\times[\![-]\!]^A]{} & & B \times (B^{A^*})^A \end{array} \tag{3}$$

With this construction, one can coalgebraically characterize language equivalence for Moore automata and, in particular, for non-deterministic automata. Take $T = \mathcal{P}_\omega$ and $\mathcal{F} = \mathcal{D}$, which is an instance of $\mathcal{M}$ for $B = 2$, the two-element semilattice. An $\mathcal{M}T$-coalgebra is a pair (X, f) with $f\colon X \to 2 \times (\mathcal{P}_\omega X)^A$, *i.e.*, an NDA. Therefore 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 non-deterministic automata. The language recognized by a state x can be defined by precomposing the unique morphism $[\![-]\!]\colon \mathcal{P}_\omega X \to 2^{A^*}$ with the unit of $\mathcal{P}_\omega$, which is the function $\{-\}\colon X \to \mathcal{P}_\omega X$ mapping each $x \in X$ into the singleton set $\{x\} \in \mathcal{P}_\omega X$.

3 Decorated trace semantics via determinization

Our aim is to reason about decorated trace equivalences of labelled transition systems. In this section, 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, trace and complete 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 congruence.

A *labeled transition system* is a pair (X, δ) where X is a set of states and $\delta\colon X \to (\mathcal{P}_\omega X)^A$ 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 $x \xrightarrow{a} y$ whenever $y \in \delta(x)(a)$. We extend the notion of transition to words $w = a_1 \dots a_n \in A^*$ as follows: $x \xrightarrow{w} y$ if and only if $x \xrightarrow{a_1} \dots \xrightarrow{a_n} y$. For $w = \varepsilon$, we have $x \xrightarrow{\varepsilon} y$ if and only if $y = x$.

We now define in a nutshell the equivalences we will be dealing with in this paper. For a function $\varphi \in (\mathcal{P}_\omega X)^A$, $I(\varphi)$ denotes the set of all labels "enabled" by φ, given by $I(\varphi) = \{a \in A \mid \varphi(a) \neq \emptyset\}$, while $Fail(\varphi)$ denotes the set $\{Z \subseteq A \mid Z \cap I(\varphi) = \emptyset\}$. Let (X, δ) be a LTS and $x \in X$ be a state. A *trace* of x is a word $w \in A^*$ such that $x \xrightarrow{w} y$ for some y. A trace w of x is *complete* if $x \xrightarrow{w} y$ and y stops, i.e., $I(\delta(y)) = \emptyset$. A *failure pair* of x is a pair $(w, Z) \in A^* \times \mathcal{P}_\omega A$ such that $x \xrightarrow{w} y$ and $Z \in Fail(\delta(y))$. A *ready pair* of x is a pair $(w, Z) \in A^* \times \mathcal{P}_\omega A$ such that $x \xrightarrow{w} y$ and $Z = I(\delta(y))$. (See [14] for more details on the classical definition of traces, complete traces, ready and failure pairs.) We use $\mathcal{T}(x)$, $\mathcal{CT}(x)$, $\mathcal{F}(x)$ and $\mathcal{R}(x)$ to denote, respectively, the set of all traces, complete traces, failure pairs and ready pairs of x.

For $\mathcal{I}$ ranging over $\mathcal{T}, \mathcal{CT}, \mathcal{F}$ and $\mathcal{R}$, two states x and y are $\mathcal{I}$-equivalent iff $\mathcal{I}(x) = \mathcal{I}(y)$ [14].

Intuitively, these equivalences can be described as follows:

- *ready semantics* identifies states of LTS's according to the set Z of actions they can trigger immediately after a certain action sequence w has been "consumed"; we call a pair (w, Z) a *ready pair*,
- *failure semantics* takes into account the set Z of actions that cannot be fired immediately after the execution of sequences w; we call a pair (w, Z) a *failure pair*,
- *trace semantics* identifies system states if and only if they can execute the same

sets of action sequences w,

- *complete trace semantics* identifies system states that perform the same sets of "complete" traces w; we call an action sequence w a *complete trace* of a state p if and only if $p \xrightarrow{w} q$ and q cannot execute any further action.

  The slight difference between trace and complete trace semantics consists in the fact that trace semantics does not detect stagnation, whereas the latter semantics takes into consideration deadlock states.

The coalgebraic characterization of the equivalences above was obtained in [12] in the following way. Given an arbitrary LTS $(X, \delta\colon X \to (\mathcal{P}_\omega X)^A)$, we associate a *decorated* LTS represented by a coalgebra of the functor $\mathcal{F}_\mathcal{I}(X) = B_\mathcal{I} \times (\mathcal{P}_\omega X)^A$, namely $(X, \langle \overline{o}_\mathcal{I}, id\rangle \circ \delta\colon X \to B_\mathcal{I} \times (\mathcal{P}_\omega X)^A)$, where the output operation $\overline{o}_\mathcal{I}\colon (\mathcal{P}_\omega X)^A \to B_\mathcal{I}$ provides the observations of interest corresponding to the original LTS and depending on the equivalence we want to study. (At this point, $B_\mathcal{I}$ represents an arbitrary semilattice with a $\vee$ operation, instantiated for each of the semantics under consideration as in [12].) Then, we determinize the decorated LTS, as depicted in Figure 1.

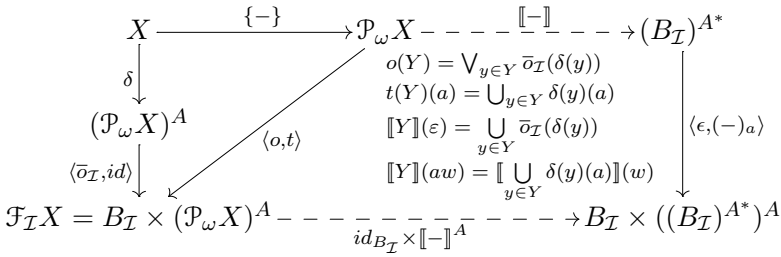


Fig. 1. The powerset construction for decorated LTSs.

Note that both the output operation and its image are parameterized by $\mathcal{I} \in \{\mathcal{R}, \mathcal{F}, \mathcal{T}, \mathcal{CT}\}$, depending on the type of decorated trace semantics under consideration. The explicit instantiations of $\overline{o}_\mathcal{I}$ and $B_\mathcal{I}$ for ready semantics are provided in Section 4, where we will also show that the coalgebraic modelling in fact coincides with the original definition of ready equivalence. (Note that the same result can easily be derived in the same style also for the case of trace, complete trace and failure semantics.) A fact that was not formally shown in [12].

The coalgebraic representation of ready, failure, trace and complete trace models as illustrated in Fig. 1 enables the definition of the corresponding equivalences as Moore bisimulations (*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 $[\![\{x_2\}]\!]$.

Moreover, it is worth observing that 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* [11].

In what follows we exploit the generalized powerset construction summarized in Fig. 1 and get an extension of bisimulation up-to congruence in [2] to the context of decorated LTS's determinized in terms of Moore automata.

Let $L_{dec} = (X, \langle \overline{o}_{\mathcal{I}}, id \rangle \circ \delta \colon X \to B_{\mathcal{I}} \times (\mathcal{P}_\omega X)^A)$ be a decorated LTS and $(\mathcal{P}_\omega X, \langle o, t \rangle : \mathcal{P}_\omega X \to B_{\mathcal{I}} \times (\mathcal{P}_\omega X)^A)$ its associated Moore automaton, as in Fig. 1. A *bisimulation up-to congruence* for L_{dec} is a relation $R \subseteq (\mathcal{P}_\omega X) \times (\mathcal{P}_\omega X)$ such that:

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

where $c(R)$ is the smallest equivalence relation which is closed with respect to set union and which includes R, defined as in [2].

Remark 3.1 Observe that by replacing $c(R)$ with R in ($\spadesuit$) one gets the definition of *Moore bisimulation.*

Theorem 3.2 *Any bisimulation up-to congruence for decorated LTS's is included in a bisimulation relation.*

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

Remark 3.3 Based on (1), (2) and Theorem 3.2, verifying behavioural equivalence of two states x_1, x_2 in a decorated LTS consists in identifying a bisimulation up-to congruence R^c relating $\{x_1\}$ and $\{x_2\}$:

$$[\![\{x_1\}]\!] = [\![\{x_2\}]\!] \text{ iff } \{x_1\} \, R^c \, \{x_2\}. \tag{4}$$

Also note that Theorem 3.2 is not a very different, but useful generalization of Theorem 2 in [2] to the context of decorated LTS's.

More insight on how to derive canonical representatives of decorated trace semantics and how to apply the bisimulation up-to congruence proof technique is provided in Section 4, for the case of ready semantics.

4 Ready semantics

In this section we show how the ingredients of Fig. 1 in Section 3 can be instantiated in order to provide a coalgebraic modelling of ready semantics, as introduced in [12]. Moreover, we prove that the resulting coalgebraic characterization of this semantics is equivalent to the original definition.

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

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

For the particular case $\varphi = \delta(x)$, $I(\delta(x))$ denotes the set of all (initial) actions ready to be fired by $x \in X$.

Recall also that a *ready pair* of x is a pair $(w, Z) \in A^* \times \mathcal{P}_\omega A$ such that $x \xrightarrow{w} y$ and $Z = I(\delta(y))$. We denote by $\mathcal{R}(x)$ the set of *all ready pairs* of 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 4.1 [$\mathcal{R}$-equivalence [14]] 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 $\mathcal{R}(x) = \mathcal{R}(y)$.

Next, we instantiate $\overline{o}_{\mathcal{I}}$ of Fig. 1 to ready semantics, where $\mathcal{I} = \mathcal{R}$.

First note that in the setting of ready semantics, the observations provided by the output operation, which we denote by $\overline{o}_{\mathcal{R}}$, refer to the sets of actions ready to be executed by the states of the LTS. Therefore, $\overline{o}_{\mathcal{R}}$ is defined as follows:

$$\begin{aligned} &\overline{o}_{\mathcal{R}} \colon (\mathcal{P}_\omega X)^A \to \mathcal{P}_\omega(\mathcal{P}_\omega A) \\ &\overline{o}_{\mathcal{R}}(\varphi) = \{I(\varphi)\}. \end{aligned}$$

For the case $\varphi = \delta(x)$, where $x \in X$, it holds that:

$$\overline{o}_{\mathcal{R}}(\delta(x)) = \{I(\delta(x))\} = \{\{a \in A \mid \delta(x)(a) \neq \emptyset\}\}.$$

In this particular instance, $B_{\mathcal{I}} = B_{\mathcal{R}} = \mathcal{P}_\omega(\mathcal{P}_\omega A)$ and the final Moore coalgebra

$$((\mathcal{P}_\omega(\mathcal{P}_\omega A))^{A^*}, \langle \epsilon, (-)_a \rangle)$$

associates to each state $\{x\}$ the set of action sequences $w \in A^*$ such that $x \xrightarrow{w} x'$, together with the sets of actions ready to be triggered by (all such) x', for $x, x' \in X$.

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

The first remark is that the behaviour of a state $x \in X$ is a function $[\![\{x\}]\!] \colon A^* \to \mathcal{P}_\omega(\mathcal{P}_\omega A)$, whereas $\mathcal{R}(x)$ is defined as a set of pairs in $A^* \times \mathcal{P}_\omega A$. However, this is no problem since the set of functions $A^* \to \mathcal{P}_\omega(\mathcal{P}_\omega A)$ and $\mathcal{P}(A^* \times \mathcal{P}_\omega A)$ are isomorphic. The set of all ready pairs $\mathcal{R}(x)$ associated to $x \in X$ is equivalently represented by $\varphi^{\mathcal{R}}_{\{x\}}$, where, for $w \in A^*$ and $Y \subseteq X$,

$$\begin{aligned} &\varphi^{\mathcal{R}}_Y \colon A^* \to \mathcal{P}_\omega(\mathcal{P}_\omega A) \\ &\varphi^{\mathcal{R}}_Y(w) = \{Z \subseteq A \mid \exists y \in t(Y)(w) \wedge Z = I(\delta(y))\} \end{aligned}$$

At this point, showing the equivalence between the coalgebraic and the original

definition of ready semantics reduces to proving that

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

Equality (6) is a direct consequence of the following theorem:

Theorem 4.2 *Let $(X, \delta\colon X \to (\mathcal{P}_\omega X)^A)$ be an LTS. Then for all $Y \subseteq X$ and $w \in A^*$, $[\![Y]\!](w) = \varphi^{\mathcal{R}}_Y(w)$.*

Proof. We proceed by induction on words $w \in A^*$.

- *Base case.* $w = \varepsilon$. Consider an arbitrary set $Y \subseteq X$. We have:

$$\begin{aligned}
[\![Y]\!](\varepsilon) &= o(Y) = \bigcup_{y \in Y} \{I(\delta(y))\} \\
\varphi^{\mathcal{R}}_Y(\varepsilon) &= \{Z \subseteq A \mid \exists y \in Y \wedge Z = I(\delta(y))\} \text{ (by def., } (\forall y \in Y)\,.\, y \xrightarrow{\varepsilon} y) \\
&= \bigcup_{y \in Y} \{I(\delta(y))\}
\end{aligned}$$

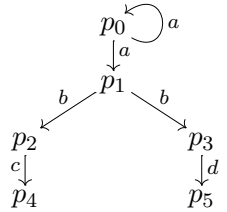
  Hence, $[\![Y]\!](\varepsilon) = \varphi^{\mathcal{R}}_Y(\varepsilon)$, for all $Y \subseteq X$.
- *Induction step.*
  Consider $w \in A^*$ and assume $[\![Y]\!](w) = \varphi^{\mathcal{R}}_Y(w)$, for all $Y \subseteq X$. We want to prove that $[\![Y]\!](aw) = \varphi^{\mathcal{R}}_Y(aw)$, where $a \in A$.

$$\begin{aligned}
[\![Y]\!](aw) &= [\![t(Y)(a)]\!](w) \\
\varphi^{\mathcal{R}}_Y(aw) &= \{Z \mid \exists y \in t(Y)(aw) \wedge Z = I(\delta(y))\} \\
&= \{Z \mid \exists y \in t(t(Y)(a))(w) \wedge Z = I(\delta(y))\} \\
&= \varphi^{\mathcal{R}}_{t(Y)(a)}(w)
\end{aligned}$$

  By the induction hypothesis, it follows that $[\![Y]\!](aw) = \varphi^{\mathcal{R}}_Y(aw)$, for all $Y \subseteq X$.

We have that $[\![Y]\!](w) = \varphi^{\mathcal{R}}_Y(w)$, for all $Y \subseteq X$ and $w \in A^*$. □

Example 4.3 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 $n \geq 1$, with

$n \in \mathbb{N}$. The set of all ready pairs associated to p_0 is:

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

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

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

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:

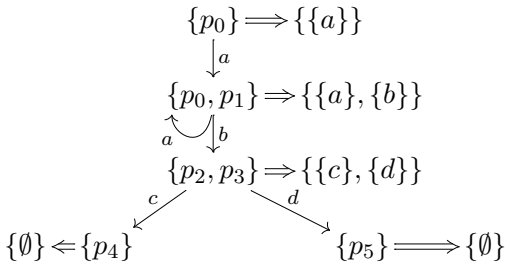


Fig. 2. Ready determinization when starting from $\{p_0\}$.

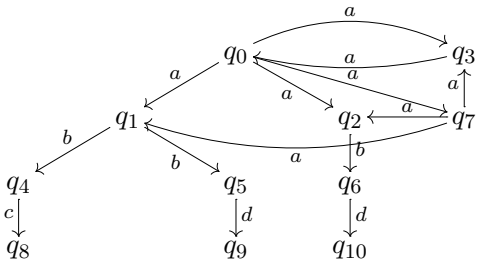
The output sets of a state Y of the Moore automaton in Fig. 2 is the set of actions associated to a certain state $y \in Y$ which can immediately be performed. For example, process p_0 in the original LTS above is ready to perform action a, whereas p_1 can immediately perform b. Therefore it holds that $o(\{p_0\}) = \{\{a\}\}$ and $o(\{p_0, p_1\}) = \{\{a\}, \{b\}\}$.

At this point, by simply looking at the automaton in Fig. 2, 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.

As we remarked in Section 3, ready equivalence of LTS's can be established in terms of bisimulation up-to congruence on Moore automata with output in $\mathcal{P}_\omega(\mathcal{P}_\omega A)$, representing the sets of actions ready to be triggered.

Next, we will explain how one can reason on ready equivalence of two LTS's, by constructing bisimulations up-to congruence on the associated Moore automata generated according to the powerset construction in Fig. 1.

Example 4.4 Consider the following LTS.



It is easy to check that q_0 and p_0 have the same ready pairs, that is $\mathcal{R}(q_0) = \mathcal{R}(p_0)$, where p_0 is the state in the LTS of the previous example.

Since we have shown the coincidence between the original definition involving equality of ready pairs and the coalgebraic representation, we can now prove that q_0 and p_0 are ready equivalent by building a bisimulation up-to congruence relating $\{p_0\}$ and $\{q_0\}$.

First, we have to determinize the LTS above. We show below the accessible part of the determinized automaton starting from state $\{q_0\}$:

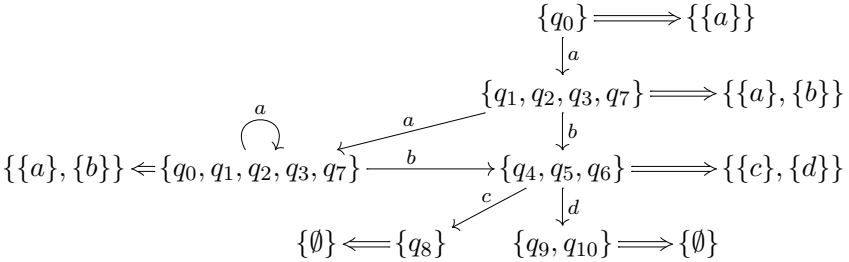


Fig. 3. Ready determinization when starting from $\{q_0\}$.

The next step is to build a bisimulation up-to congruence R on the sets of states of the generated Moore automata in Fig. 2 and Fig. 3, such that $(\{p_0\}, \{q_0\}) \in R$.

We start by taking $R = \{(\{p_0\}, \{q_0\})\}$ and check whether this is already a bisimulation up-to congruence, by considering the output values and transitions, and check whether no new states appear in $c(R)$ in the process. If new pairs of states appear, we add them to R and repeat the process.

Eventually, we end-up with a bisimulation up-to congruence

$$R = \{(\{p_0\}, \{q_0\}), (\{p_0, p_1\}, \{q_1, q_2, q_3, q_7\}),$$
$$(\{p_2, p_3\}, \{q_4, q_5, q_6\}), (\{p_4\}, \{q_8\}), (\{p_5\}, \{q_9, q_{10}\})\}$$

By construction $(\{p_0\}, \{q_0\}) \in R$, so by (4) it follows that $[\![\{p_0\}]\!] = [\![\{q_0\}]\!]$.

Note that R is not a bisimulation relation since $\{p_0, p_1\} \xrightarrow{a} \{p_0, p_1\}$ and $\{q_1, q_2, q_3, q_7\} \xrightarrow{a} \{q_0, q_1, q_2, q_3, q_7\}$ but $(\{p_0, p_1\}, \{q_0, q_1, q_2, q_3, q_7\}) \notin R$. Nevertheless, observe that R is a bisimulation up-to congruence since

$(\{p_0, p_1\}, \{q_0, q_1, q_2, q_3, q_7\}) \in c(R)$:

$$\begin{array}{rcll} \{p_0, p_1\} & = & \{p_0\} \cup \{p_0, p_1\} & \\ & c(R) & \{q_0\} \cup \{p_0, p_1\} & (\{\{p_0\}, \{q_0\}\}) \in R) \\ & c(R) & \{q_0\} \cup \{q_1, q_2, q_3, q_7\} & (\{\{p_0, p_1\}, \{q_1, q_2, q_3, q_7\}) \in R) \\ & = & \{q_0, q_1, q_2, q_3, q_7\} & \end{array}$$

Also observe that the bisimulation up-to congruence given above is one pair smaller than the Moore bisimulation relating the automata in Fig. 2 and Fig. 3, which would also include $(\{p_0, p_1\}, \{q_0, q_1, q_2, q_3, q_7\})$.

5 Canonical representatives

Given a *decorated* LTS $(X, \langle \overline{o}_{\mathcal{I}}, id \rangle \circ \delta)$, we showed in the previous section how to construct a *determinized decorated* LTS $(\mathcal{P}_\omega X, \langle o, t \rangle)$. The map $[\![-]\!] \colon \mathcal{P}_\omega X \to B_{\mathcal{I}}^{A^*}$ provides us with a *canonical representative* of the behaviour of each state in $\mathcal{P}_\omega X$. The image $(C, \overline{\delta})$ of $(\mathcal{P}_\omega X, \langle o, t \rangle)$, 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 $\varphi \colon A^* \to B_{\mathcal{I}}$. and that their decorations and transitions are given by the functions $\epsilon \colon B_{\mathcal{I}}^{A^*} \to B_{\mathcal{I}}$ and $(-)_a \colon B_{\mathcal{I}}^{A^*} \to (B_{\mathcal{I}}^{A^*})^A$, defined in Section 2. The states of the canonical representative $(C, \overline{\delta})$ are also functions $\varphi \colon A^* \to B_{\mathcal{I}}$, i.e., $C \subseteq B_{\mathcal{I}}^{A^*}$. Moreover, the function $\overline{\delta} \colon C \to B_{\mathcal{I}} \times C^A$ is simply the restriction of $\langle \epsilon, (-)_a \rangle$ to C, that means $\overline{\delta}(\varphi) = \langle \varphi(\epsilon), (\varphi)_a \rangle$ for all $\varphi \in C$.

Finally, it is interesting to observe that $B_{\mathcal{I}}^{A^*}$ carries a semilattice structure (inherited by $B_{\mathcal{I}}$) and that $[\![-]\!] \colon \mathcal{P}_\omega X \to B_{\mathcal{I}}^{A^*}$ 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.

6 Conclusions and future work

In this paper, we have proved that the coalgebraic characterizations of decorated trace semantics in [12] are equivalent with the corresponding standard definitions. More precisely, for the case of ready equivalence, we have shown that for a state x in a labelled transition system, the coalgebraic canonical representative $[\![\{x\}]\!]$, given by determinization and finality, coincides with the classical semantics $\mathcal{R}(x)$ representing the ready pairs of x. In addition, we have illustrated how to reason about decorated trace equivalence using coinduction, by constructing suitable bisimulations up-to congruence. 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 [9]. Note that even though in this paper we provided explicit proofs and examples only for the case of ready equivalence, similar results can be easily derived

in the same style for (complete) trace and failure semantics.

A similar idea of system determinization was also applied in [4], in a non-coalgebraic setting, for the case of testing semantics where *must testing* coincides with failure semantics in the absence of divergence. A coalgebraic characterization of the spectrum was also attempted in [7], in a somewhat *ad hoc* fashion. Connections with these works are still to be explored.

There are two possible directions for future works. On the one hand, we would like to investigate to what extent the coalgebraic treatment of decorated trace semantics can be applied in the context of probabilistic systems. On the other hand, we would like to understand how our approach can be combined with [3] to obtain a coinductive approach to denotational (linear-time) semantics of different kinds of processes calculi.

References

[1] Awodey, S., "Category theory," Oxford Logic Guides, Oxford University Press, 2010.

[2] Bonchi, F. and D. Pous, *Checking NFA equivalence with bisimulations up to congruence*, Technical report (2012), 13p.
URL http://hal.archives-ouvertes.fr/hal-00639716

[3] Boreale, M. and F. Gadducci, *Processes as formal power series: A coinductive approach to denotational semantics*, Theor. Comput. Sci. **360** (2006), pp. 440–458.

[4] Cleaveland, R. and M. Hennessy, *Testing equivalence as a bisimulation equivalence*, Formal Asp. Comput. **5** (1993), pp. 1–20.

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

[6] Milner, R., "Communication and concurrency," Prentice-Hall international series in computer science, Prentice Hall, 1989.

[7] Monteiro, L., *A coalgebraic characterization of behaviours in the linear time - branching time spectrum*, in: A. Corradini and U. Montanari, editors, *WADT*, Lecture Notes in Computer Science **5486** (2008), pp. 251–265.

[8] Park, D. M. R., *Concurrency and automata on infinite sequences*, in: P. Deussen, editor, *Theoretical Computer Science*, Lecture Notes in Computer Science **104** (1981), pp. 167–183.

[9] Roşu, G. and D. Lucanu, *Circular Coinduction – A Proof Theoretical Foundation*, in: *CALCO'09*, LNCS, 2009.

[10] Rutten, J. J. M. M., *Universal coalgebra: a theory of systems*, Theor. Comput. Sci. **249** (2000), pp. 3–80.

[11] Sangiorgi, D. and J. Rutten, "Advanced Topics in Bisimulation and Coinduction," Cambridge Tracts in Theoretical Computer Science, Cambridge University Press, 2011.

[12] Silva, A., F. Bonchi, M. Bonsangue and J. Rutten, *Generalizing determinization from automata to coalgebras*, submitted.

[13] Silva, A., F. Bonchi, M. M. Bonsangue and J. J. M. M. Rutten, *Generalizing the powerset construction, coalgebraically*, in: K. Lodaya and M. Mahajan, editors, *FSTTCS 2010*, LIPIcs **8**, 2010, pp. 272–283.
URL http://drops.dagstuhl.de/opus/volltexte/2010/2870

[14] van Glabbeek, R., *The linear time - branching time spectrum I. The semantics of concrete, sequential processes*, in: J. Bergstra, A. Ponse and S. Smolka, editors, *Handbook of Process Algebra* (2001), pp. 3–99.