# Coinductive Definition of Distances between Processes: Beyond Bisimulation Distances

David Romero-Hernández, David Frutos Escrig



# Coinductive Definition of Distances Between Processes: Beyond Bisimulation Distances*

David Romero Hernández, David de Frutos Escrig

Dpto. Sistemas Informáticos y Computación
Facultad CC. Matemáticas, Universidad Complutense de Madrid, Spain
dromeroh@pdi.ucm.es, defrutos@sip.ucm.es

**Abstract.** Bisimulation captures in a coinductive way the equivalence between processes, or trees. Several authors have defined bisimulation distances based on the bisimulation game. However, this approach becomes too local: whenever we have in one of the compared processes a large collection of branches different from those of the other, only the farthest away is taken into account to define the distance. Alternatively, we have developed a more global approach to define these distances, based on the idea of how much we need to modify one of the compared processes to obtain the other. Our original definition only covered finite processes. Instead, now we present here a coinductive approach that extends our distance to infinite but finitary trees, without needing to consider any kind of approximation of infinite trees by their finite projections.

## 1 Introduction

Bisimulation [16,14,20] is a popular way to define the semantics of processes. Starting from their operational semantics, defined by a transition system, it captures the "natural" behavior of the processes, paying attention to the branching in them, but abstracting away from possible repetitions of equivalent behaviors. Bisimulations are just coinductive proofs of the equivalence between processes, and in fact they became one of the main causes of the popularization of the study of coinduction [20] and coalgebras [19,11,12] in the last years. They can be established in many different ways, in particular by means of the bisimulation game [21], that enlightens the co-character of bisimilarity.

When comparing two processes, the proof of their bisimilarity certainly indicates us that they are equivalent. The problem comes if we receive the information that they are not bisimilar. Then, if we substitute one component by the other, it is expected that the behavior of the full system will change "at least a bit". We want to quantify those deviations; they are formalized by our (new) distance between processes with respect to the bisimulation equivalence.

Recently, several variants of the bisimulation game have been used to define "bisimulation distances" [4,6,8,1]. They develop the seminal ideas in several previous works, such as [5,9,23]. However, as we have already illustrated in [17,18]

* Partially supported by the Spanish projects STRONGSOFT (TIN2012-39391-C04-04) and PROMETIDOS S2009/TIC-1465.

by means of several examples, these distances have some "limitations", that we try to remove by means of our new bisimulation distance. We also include in this paper some new examples enlightening the difference between our approach and those based in the bisimulation game.

Whenever we formalize the family of computations of a process we obtain a tree. Therefore, any distance between processes induces a distance between those trees. We have followed this path in the opposite way: let us look for a "natural" notion of distance between trees, and we will turn it into a distance between processes. In [17] we have presented an operational definition of our new *global bisimulation distance* for the particular case of finite trees. Roughly speaking, we define our distance between trees "computing" the costs in order to transform one of the trees into the other. We consider a given distance $\mathbf{d}$ on the alphabet of actions, so that the cost of substituting an action $a$ by another $b$ is given by $\mathbf{d}(a, b)$.

In this paper, we use coinduction to define the distance between processes in which we are interested. Of course, an alternative way to define the distance between infinite trees is to approximate them by their finite projections, and then taking limits. Looking for an "homogeneous" procedure that could capture all these approximations in a compact way, we introduce our coinductive distance as the coinductive "closure" of the finite transformations by means of which we defined our distance between finite processes in [17]. Once we have it, we get all the machinery of coinductive proofs in order to study our distance.

Even if the notion of tree is omnipresent in the field of semantics of processes, there is not a clearly standardized presentation of the different classes of trees in the literature. This is why we start the paper by reminding in Section 2 the definitions on trees and labelled transition systems that we use in the following. In Section 3, we recall the previous work on bisimulation distances and our alternative operational proposal covering mainly finite trees. Section 4 is the core of the paper and presents the coinductive extension of this approach covering also infinite trees. Finally, we conclude with a short section devoted to a discussion on the continuity of the coinductive distance, and the conclusions of the paper.

We strongly acknowledge the detailed reading and the suggestions of the referees of this paper, that have contributed to improve the presentation of this work.

## 2 Preliminaries

Let us start by recalling the coalgebraic definition of labelled transition systems (lts). As usual, we use them to represent the operational semantics of processes.

**Definition 1.** *Labelled Transition Systems (lts)*[1] *on a set of actions* $\mathbb{A}$ *and a set of states* $N$*, are given by a function* $succ : N \to \mathcal{P}(\mathbb{A} \times N)$*. We denote each*

[1] Therefore, lts's are just arc-labelled graphs, or more formally coalgebras $succ : N \to LTS(N, \mathbb{A})$ of the functor $LTS(N, \mathbb{A}) := \mathcal{P}(\mathbb{A} \times N)$ on the plain category of sets, *Set*. See for instance [12,20] for much more on coalgebras.

*lts by the corresponding pair* $(N, succ)$*. A lts with initial state* $(N, succ, n_0)$*, is just a lts* $(N, succ)$ *where some distinguished (initial) state* $n_0 \in N$ *is fixed. To simplify our notation, we usually remove the succ component from lts's.*

We say that any sequence $n_0 a_1 n_1 \dots a_k n_k$ with $(a_{i+1}, n_{i+1}) \in succ(n_i)\ \forall i \in \{0 \dots k-1\}$, is a path in $(N, n_0)$. We denote the set of paths (or computations) by $Path(N, n_0)$. We say that the system $N$ is *finite state*, if $|N| < \infty$; $(N, n_0)$ is *finite*, if $|Path(N, n_0)| < \infty$; we say that $(N, n_0)$ has only *finite computations*, if there is no infinite path $n_0 a_1 n_1 a_2 n_2 \dots$. We say that a system $N$ is *finitely branching*, if for all $n \in N$ we have $|succ(n)| < \infty$.

*Example 1.* (See Fig.1) Two simple finite-state systems that however have infinitely many computations are the following: $N_{1,\infty} = \{n_0\}$, with $succ(n_0) = \{(a, n_0)\}$; $N_{2,\infty} = \{n_0, n_1\}$, with $succ(n_0) = succ(n_1) = \{(a, n_0), (a, n_1)\}$.

*Example 2.* (See Fig.1) Next three interesting non-finitely branching systems:

1. $N_{\mathbb{N}} = (\mathbb{N}, succ, 0)$ with $\mathbb{A} = \mathbb{N}$, $succ(0) = \{(k, k) \mid k \in \mathbb{N}\}$, $succ(k) = \emptyset$, $\forall k > 0$.
2. $N_2 = \{0\} \cup \{(i, j) \mid i, j \in \mathbb{N}, 1 \leq j \leq i\}$, taking $n_0 = 0$ with $succ(0) = \{(a, (n, 1)) \mid n \in \mathbb{N}\}$ and $succ((i, j)) = \{(a, (i, j+1))\}$ if $j < i$, while $succ((i, i)) = \emptyset$.
3. $N_2^+ = N_2 \cup \{(\infty, n) \mid n \in \mathbb{N}\}$, changing also the definition of $succ$, taking $succ(0) = \{(a, (x, 1)) \mid x \in \mathbb{N} \cup \{\infty\}\}$ and $succ((\infty, j)) = \{(a, (\infty, j+1))\}$.

We can define (rooted) trees as a particular class of lts's:

**Definition 2.** *We say that a system* $(N, n_0)$ *is (or defines) a* tree *$t$ if for all* $n \in N$ *there is a single path* $n_0 a_1 n_1 \dots a_k n_k$ *with* $n_k = n$*. Then, we say that each node* $n_k$ *is at* level *$k$ in $t$, and define* $Level_k(t) = \{n \in N \mid n$ *is at level* $k$ *in* $t\}$*. We define the* depth *of $t$ as* $depth(t) = sup\{l \in \mathbb{N} \mid Level_l(t) \neq \emptyset\} \in \mathbb{N} \cup \{\infty\}$*. We denote by* $Trees(\mathbb{A})$ *the class of trees on the set* $\mathbb{A}$*, and by* $FTrees(\mathbb{A})$*, the subclass of finite state trees.*

Any node $n \in N$ of a tree $t = (N, succ, n_0)$ induces a subtree $t_n = (N_n, succ, n)$, where $N_n$ is the set of nodes $n'_k \in N$ such that there exists a path $n'_0 a_1 n'_1 \dots a_k n'_k$ with $n'_0 = n$. We decompose any tree $t$ into the formal sum $\sum_{n_{1j} \in Level_1(t)} a_j t_{n_{1j}}$. Since our trees are unordered, by definition, this formal sum is also unordered. The tree $\mathbf{0}$ corresponds to the system $(\{n_0\}, succ_0, n_0)$ with $succ_0(n_0) = \emptyset$, while if $|Level_1(t)| = 1$ we have $t = at'$, which can be reversed to define the tree $at'$ starting from $a \in \mathbb{A}$ and $t' \in Trees(\mathbb{A})$. In a similar way, whenever $Level_1(t) = N_1 \cup N_2$ is a disjoint decomposition of that set, we can write $t = \sum_{n_{1j} \in N_1} a_j t_{n_{1j}} + \sum_{n_{1k} \in N_2} a_k t_{n_{1k}}$, which can be also reversed to define the sum $(+)$ of trees. Note that $+$ becomes commutative by definition.

For any tree $t \in Trees(\mathbb{A})$, we define its *first-level width*, that we will represent by $||t||$, as $||t|| = |Level_1(t)|$. We also define the *first $k$-levels width of $t$*, denoted by $||t||_k$, as $||t||_k = max\{||t_n|| \mid n \in \bigcup_{l \leq k} Level_l(t)\}$. *Finitary trees* are just trees that are finitely branching systems, or equivalently, those such that

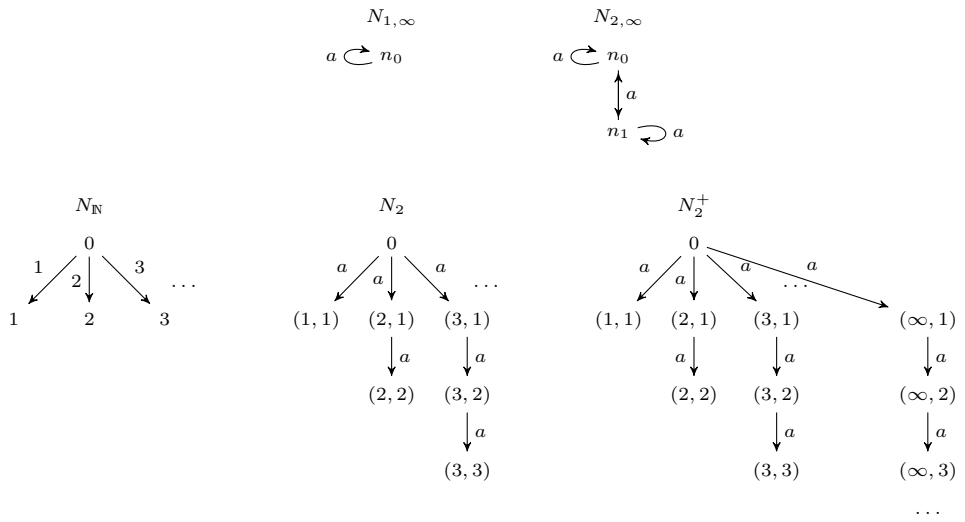



**Fig. 1.** Labelled Transitions Systems and Trees used in Examples 1-2

$||t||_k < \infty$, $\forall k \in \mathbb{N}$. We denote by $FyTrees(\mathbb{A})$ the collection of *finitary trees* in $Trees(\mathbb{A})$.

All the systems in Ex.2 are indeed trees. Instead, those in Ex.1 are not trees.

**Definition 3.** *Given a lts with initial state $(N, succ, n_0)$, we define its unfolding $unfold(N)$ as the tree $(\overline{N}, \overline{succ}, \overline{n_0})$, where $\overline{N} = Path(N, n_0)$, $\overline{succ}(n_0 a_1 \dots n_k) = \{(a, n_0 a_1 \dots n_k a n') \mid (a, n') \in succ(n_k)\}$, and $\overline{n_0} = n_0$.*

**Definition 4.** *Let $(N, succ)$ be a lts. We say that a relation $\mathcal{R}$ on $N$ is a bisimulation, if for all $(n, n') \in \mathcal{R}$ we have*

- $\forall\ (a, n_1) \in succ(n)\ \exists\ (a, n_1') \in succ(n')$, $(n_1, n_1') \in \mathcal{R}$.
- $\forall\ (a, n_2) \in succ(n')\ \exists\ (a, n_2') \in succ(n)$, $(n_2', n_2) \in \mathcal{R}$.

*We say that $n$ and $n'$ are bisimilar if there exists some bisimulation $\mathcal{R}$ such that $(n, n') \in \mathcal{R}$, and then we write $n \sim n'$.*

By considering the disjoint union of two systems, we can extend the definition above to relate states from two different systems. In particular, if we consider two *lts* with initial state (or equivalently two trees) we say that $(N, succ, n_0) \sim (N', succ', n_0')$ if and only if there is a bisimulation containing the pair $(n_0, n_0')$. Usually we will simply write $n_0 \sim n_0'$, and the same in the case of trees, as usual.

The fact that systems are represented by their unfolding is formalized by the following result.

**Proposition 1.** *For any lts with initial state $(N, succ, n_0)$, and its unfolding $(\overline{N}, \overline{succ}, \overline{n_0})$, we have $n_0 \sim \overline{n_0}$.*

**Definition 5.** *Given a tree $t = (N, succ, n_0)$ and $k \in \mathbb{N}$, we define its $k$-th cut or projection, $\pi_k(t)$, as the restriction of $t$ to the nodes in $\bigcup_{l \le k} Level_l(t)$:*

$\pi_k(t) = (\pi_k(N), succ_k, n_0)$, *where* $\pi_k(N) = \bigcup_{l \leq k} Level_l(t)$, $succ_k(n) = succ(n)$ *for* $n \in \bigcup_{l<k} Level_l(t)$, *and* $succ_k(n) = \emptyset$ *if* $n \in Level_k(t)$.

**Proposition 2.** *For any* $t \in Tree(\mathbb{A})$ *and* $l, k \in \mathbb{N}$ *with* $l \leq k$, *we have* $\pi_l(\pi_k(t)) = \pi_l(t)$. *Any finitary tree is unequivocally defined by its sequence of projections:* $\forall t, t' \in FyTree(\mathbb{A})$ $(\forall k \in \mathbb{N}\ \pi_k(t) \sim \pi_k(t')) \Rightarrow t \sim t'$.

*Example 3.* The result above becomes false if we consider infinitary trees. For the trees $N_2$ and $N_2^+$ in Ex.2, we have $\pi_k(N_2) \sim \pi_k(N_2^+)$ $\forall k \in \mathbb{N}$, since the "additional" branch executing $a^k$ provided by $N_2^+$ can be "absorbed" by the infinitely many such branches that we already have in $\pi_k(N_2)$. Therefore, this is a (well known) counterexample disproving the continuity of bisimilarity wrt the approximations, provided by the projections $\pi_k$, if we allow infinitary trees.

As a consequence, we will restrict ourselves to finitely branching processes all along the rest of the paper. It would not be enough to consider instead just image finiteness trees, because our approach considers all the successors of each node in an homogeneous way, without taking care of their labels. Then, problems can appear as soon as a node has infinitely many successors.

## 3 Classical and Global bisimulation distances

We consider domains of actions $(\mathbb{A}, \mathbf{d})$, where $\mathbf{d} : \mathbb{A} \times \mathbb{A} \to \mathbb{R}^+ \cup \{\infty\}$ is a distance between actions, with $\mathbf{d}(a, b) = \mathbf{d}(b, a)$, $\forall a, b \in \mathbb{A}$, and, as usual, $\mathbf{d}(a, b) = 0 \Leftrightarrow a = b$, and $\mathbf{d}(a, c) + \mathbf{d}(c, b) \geq \mathbf{d}(a, b)$, $\forall a, b, c \in \mathbb{A}$ where $+$ is extended to $\mathbb{R}^+ \cup \{\infty\}$ as usual. Intuitively $\mathbf{d}(a, b) = \infty$ expresses that two actions are absolutely not interchangeable. If the value of a distance $\mathbf{d}(a, b)$ is not specified in our examples, we will assume that $\mathbf{d}(a, b) = \infty$.

The well known bisimulation game [15,22], allows us to characterize the bisimilarity relation. It is played by two players: the attacker ($\mathcal{A}$) and the defender ($\mathcal{D}$). The former executes any fireable transition from one of the compared trees, and the second has to reply it in the other tree. The attacker wins if the defender cannot counteract one of his moves; while the defender wins if he can reply forever.

**Theorem 1.** *([15,22]) For any* $t$,$t' \in Trees(\mathbb{A})$ $t \sim t'$ *(resp.* $t \not\sim t'$*) if and only if* $\mathcal{D}$ *(resp.* $\mathcal{A}$*) has a winning strategy for the bisimulation game starting at* $(t, t')$.

Most of the recent approaches to define distances between processes –e.g. [7]– use quantitative versions of the bisimulation game. As in the plain bisimulation game, the defender has to simulate the action played by the attacker, but in this case he can fail to reply an $a$ transition, firing instead some $b$. However, whenever he cheats the attacker, he has to pay him for the distance $\mathbf{d}(b, a)$. Then, the distance between two trees $t$ and $t'$(equivalently, between two processes) is defined as the value of that game starting from the roots of $t$ and $t'$. In the following, we will call "classical" the distances defined following this approach.

Inspired by the notion of amortized bisimulation [13], we have developed a coinductive presentation of the classical bisimulation distances [17]. Instead of giving a definition of the distance between two trees, that requires the use of fixpoint theory, we state when an indexed family of relations between trees provides a collection of bounds on the distances between the pairs of trees in them.

As done for instance in [2,8], and thoroughly discussed in [3], when comparing pairs of processes, it is natural to introduce a "discount factor" $\alpha \in (0,1)$. Then, the differences in the $k$-th level of the compared trees are weighted by $\alpha^k$, following the idea that differences in the far future are less important than those in the near. As a consequence, it is possible to obtain finite distances when comparing two processes with "infinitely many differences" between them. However, we will also allow that $\alpha = 1$ to cover the case in which we are not interested in the weighting of those differences.

**Definition 6.** *(see [17]) Given a domain of actions* $(\mathbb{A}, \mathbf{d})$ *and a discount factor* $\alpha \in (0,1]$*, we say that a family of relations between trees,* $\mathcal{R} \subseteq Trees(\mathbb{A}) \times Trees(\mathbb{A}) \times \mathbb{R}^+$*, is a classical bisimulation distance family (cbdf) for* $\mathbf{d}$ *and* $\alpha$*, if it satisfies*

$$\begin{array}{ccccc} t & R_d & t' & & t & R_d & t' \\ \forall\, a \downarrow & \Longrightarrow & \downarrow \exists b & \wedge & \exists a \downarrow & \Longleftarrow & \downarrow \forall\, b \\ t_1 & R_{\underline{d-\mathbf{d}(b,a)}_{\alpha}} & t'_1 & & t_1 & R_{\underline{d-\mathbf{d}(b,a)}_{\alpha}} & t'_1 \end{array}$$

*where, we take* $tR_dt'$ *if and only if* $(t,t',d) \in \mathcal{R}$ *and implicitly, we are assuming that the values* $d - \mathbf{d}(b,a)$ *are nonnegative. We say that* $t$ *and* $t'$ *are at most at classical bisimulation distance* $d$ *for the factor* $\alpha$*, and then we write* $d^{\alpha}_{\mathbf{d}}(t,t') \leq d$*, if there is some* cbdf $\mathcal{R}$ *with* $tR_dt'$*.*

**Proposition 3.** *(see [17]) The value of the quantitative game –see [7]– defining the "classical" bisimulation distance* $dist^{\alpha}_{\mathbf{d}}(t,t')$ *is* $\inf(\{d \in \mathbb{R}^+ \mid d^{\alpha}_{\mathbf{d}}(t,t') \leq d\})$*.*

It is well known that, for finitary trees, this classical bisimulation distance is indeed a quantitative refinement of bisimilarity.

**Theorem 2.** *For all* $t$*,* $t' \in FyTrees(\mathbb{A})$ *and any discount factor* $\alpha \in (0,1]$*, we have* $t \sim t'$ *if and only if* $d^{\alpha}_{\mathbf{d}}(t,t') \leq 0$*, if and only if* $dist^{\alpha}_{\mathbf{d}}(t,t') = 0$*.*

In spite of this, we consider that in some cases this distance generates values that do not accurately reflect the differences between some pairs of trees.

*Example 4.* (see Fig. 2) We have a service that allows some access to the bits of our password, once we have identified ourselves in the appropriate way. Let us abstract this service as the tree $t = \sum_{i \in 1..64} a_i$. Now, let us assume that $a'_i$ represents a cracked access to the corresponding position. Then, for each $j \in \{1 \ldots 64\}$ the system represented by the tree $t'_j = (\sum_{i \neq j} a_i) + a'_j$ certainly is wrong, but does not compromise too much the security of the system. Instead, the totally cracked system represented by $t'_{1..64} = \sum_{i \in 1..64} a'_i$ corresponds to a disastrous situation. If we take $\mathbf{d}(a_i, a'_i) = 1$, we obtain $dist^1_{\mathbf{d}}(t'_j, t) = 1$, but also $dist^1_{\mathbf{d}}(t'_{1..64}, t) = 1$.

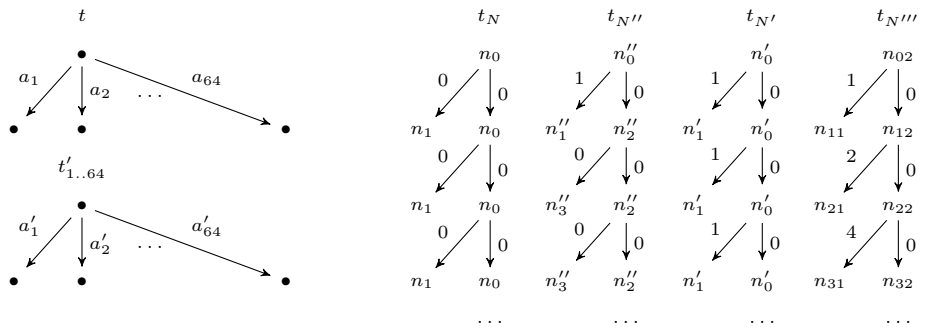



**Fig. 2.** Trees used in Ex. 4 - Ex.8

The bad behavior of the "classical" bisimulation distance stems from the fact that the quantitative bisimulation game only considers single computations of the compared trees. As a consequence, it cannot capture the differences "accumulated" by repeated use of a system, as illustrated in Ex.4. Later, we will see how our "global" approach copes with this feature in a more satisfactory manner.

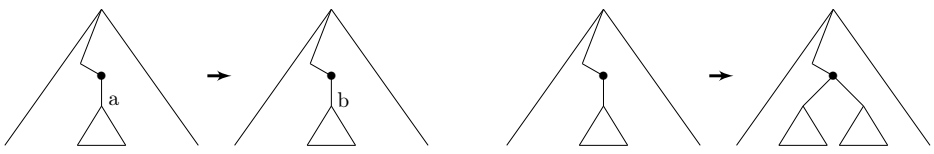
In [17], we have presented our operational definitions that allow us to obtain bounds for our new global distances between finite trees. These bounds are given by the cost of any transformation that turns one of the trees into the other. The following definition states which are the valid steps of those transformations and their costs. Roughly, any application of idempotency of $+$ has no cost, while the change of an action $a$ at level $k$ into another $b$ has as cost $\alpha^k \mathbf{d}(b,a)$.

**Definition 7.** *Given a domain of actions* $(\mathbb{A}, \mathbf{d})$ *and a discount factor* $\alpha \in (0,1]$*, we inductively define the distance steps on* $FTrees(\mathbb{A})$ *by*

1. $d \geq 0 \Rightarrow (t \rightsquigarrow^1_{\alpha,d} t + t \wedge t + t \rightsquigarrow^1_{\alpha,d} t)$.
2. $d \geq \mathbf{d}(a,b) \Rightarrow at \rightsquigarrow^1_{\alpha,d} bt$.
3. $t \rightsquigarrow^1_{\alpha,d} t' \Rightarrow t + t'' \rightsquigarrow^1_{\alpha,d} t' + t''$.
4. $t \rightsquigarrow^1_{\alpha,d} t' \Rightarrow at \rightsquigarrow^1_{\alpha,\alpha d} at'$.

*We associate to each distance step its level, that is a natural number. The level of any step generated by* 1. *or* 2. *is one; while if the level of the corresponding premise* $t \rightsquigarrow^1_{\alpha,d} t'$ *is* $k$*, then the level of a step generated by* 3. *(resp.* 4.*) is* $k$ *(resp.* $k+1$*). Finally, we define the family of global distance relations* $\langle \rightsquigarrow_{\alpha,d} |\ d \in \mathbb{R}^+ \rangle$*, taking* $t \rightsquigarrow_{\alpha,d} t'$ *if there exists a sequence* $\mathcal{S} := t = t^0 \rightsquigarrow^1_{\alpha,d_1} t^1 \rightsquigarrow^1_{\alpha,d_2} t^2 \rightsquigarrow^1_{\alpha,d_3} \cdots \rightsquigarrow^1_{\alpha,d_n} t^n = t'$*, with* $\sum_{i=1}^n d_i = d$.

Therefore, we can see any sequence of distance steps turning $t$ into $t'$ as a sequence of local transformations $t^i \rightsquigarrow^1_{\alpha,d_i} t^{i+1}$. Each one of them either changes a single label or duplicates a partial branch at a certain level of $t^i$.



*Remark 1.* For technical reasons we want that $t \rightsquigarrow_{\alpha,d} t$ for any $t$, and all $d \in \mathbb{R}^+$. This can be obtained by considering the sequence $\mathcal{S} := t \rightsquigarrow^1_{\alpha,d} t + t \rightsquigarrow^1_{\alpha,0} t$.

Although at the formal level we only work with the relations $\rightsquigarrow_{\alpha,d}$, sometimes we also talk about the (global) distance defined by these bounds.

*Example 5.* Let us consider again the systems in Ex.4. Now, it is immediate to check that $t \rightsquigarrow_{1,1} t'_j$ for all $j \in \{1 \dots 64\}$. Therefore, in this case, the global bisimulation distance between $t$ and any $t'_j$ coincides with the classical bisimulation distance. However $t \rightsquigarrow_{1,64} t'_{1..64}$, but we do not have $t \rightsquigarrow_{1,d} t'_{1..64}$ for any $d < 64$: in order to transform $t$ into $t'_{1..64}$, we need to change each $a_i$ into $a'_i$, paying one unit at each step. Instead, we had $dist^1_{\mathbf{d}}(t, t'_{1..64}) = 1$. We consider that our global distance reflects in a much more accurate way the "intuitive" distance between these trees.

*Example 6.* We have to pass an examination about a subject with $l$ lessons. A good student would study all of them, thus getting $S = \sum_{1..l} a_l$, which means that he totally knows the subject. At the exam the examiners choose somehow $k$ lessons, and then each student can select a single one to develop. This means that any student that ignores up to k-1 lessons could perfectly pass the exam. These students are represented by $S_I = \sum_{i \notin I} a_i$, where $I$ is the set of lessons that they did not study. Now, at which extend such an student is risky? What happens if the day of the exam he forgets some lesson?. If $|I| = k - 1$, then as soon as he forgets a single lesson he is in risk of failing; instead, if $|I| = 1$ he has definitely much more chances. This is again captured by our global bisimulation distance, but not by the classical one. The situation is similar to that studied in [10], where they wanted to capture how many failures are allowed before a system will fail to satisfy the requirements at its specification.

## 4 The coinductive global bisimulation distance

To get a general coinductive definition of our global distance for *FyTrees(𝔸)*, we keep the first three rules in Def.7, that allow us to make changes at the first level of the trees. But instead of rule 4, we introduce a coinductive rule that allows us to replace any non trivial subtree $t$ at depth one by another $t'$, getting a distance $\alpha d$, whenever $(t, t', d)$ is in the family that defines our global distance.

We formalize our definition in two steps. The first one, introduces the rules that produce the steps of the *coinductive transformations* between trees, starting from any family of triples $(t, t', d)$, with $t$, $t' \in FyTrees(\mathbb{A})$ and $d \in \mathbb{R}^+$.

**Definition 8.** *Given a domain of actions $(\mathbb{A}, \mathbf{d})$, a discount factor $\alpha \in (0, 1]$ and a family $\mathcal{D} \subseteq FyTrees(\mathbb{A}) \times FyTrees(\mathbb{A}) \times \mathbb{R}^+$, we define the family of relations $\equiv^{\mathcal{D},\alpha}_d$, by:*

1. *For all $d \geq 0$ we have (i) $(\sum_{j \in J} a_j t_j) + at + at \equiv^{\mathcal{D},\alpha}_d (\sum_{j \in J} a_j t_j) + at$ , and (ii) $(\sum_{j \in J} a_j t_j) + at \equiv^{\mathcal{D},\alpha}_d (\sum_{j \in J} a_j t_j) + at + at$.*
2. *$(\sum_{j \in J} a_j t_j) + at \equiv^{\mathcal{D},\alpha}_{\mathbf{d}(a,b)} (\sum_{j \in J} a_j t_j) + bt$.*

*3. For all* $(t, t', d) \in \mathcal{D}$ *we have* $(\sum_{j\in J} a_j t_j) + at \equiv^{\mathcal{D},\alpha}_{\alpha d} (\sum_{j\in J} a_j t_j) + at'$.

*Remark 2.* To simplify the notation, we will simply write $\equiv_d$ instead of $\equiv^{\mathcal{D},\alpha}_d$, whenever $\mathcal{D}$ and $\alpha$ will be clear from the context.

Next, the second one. Inspired by the conditions imposed to bisimulations –that can be seen as "circular proofs" of bisimilarity of all the pairs in them– we introduce the coinductive proof obligations imposed to the families of triples as above, in order to define satisfactory coinductive families of distances.

**Definition 9.** *Given a domain of actions* $(\mathbb{A}, \mathbf{d})$ *and a discount factor* $\alpha \in (0, 1]$*, we say that a family* $\mathcal{D}$ *is an* $\alpha$*-coinductive collection of distances (*$\alpha$*-ccd) between finitary trees, if for all* $(t, t', d) \in \mathcal{D}$ *there exists a* finite coinductive transformation sequence $\mathcal{C} := t = t^0 \equiv^{\mathcal{D},\alpha}_{d_1} t^1 \equiv^{\mathcal{D},\alpha}_{d_2} \ldots \equiv^{\mathcal{D},\alpha}_{d_n} t^n = t'$*, with* $d \geq \sum_{j=1}^{n} d_j$*. Then, when there exists an* $\alpha$*-ccd* $\mathcal{D}$ *with* $(t, t', d) \in \mathcal{D}$*, we will write* $t \equiv^{\alpha}_{d} t'$*, and say that tree* $t$ *is at most at distance* $d$ *from tree* $t'$ *wrt* $\alpha$*.*

Notation: We say that the steps generated by application of rules 1 and 2 in Def.8 are *first level steps*; while those generated by rule 3 are *coinductive steps*.

*Remark 3.* The reason because we have introduced the condition $d \geq \sum_{j=1}^{n} d_j$, and not just $d = \sum_{j=1}^{n} d_j$, is in order to guarantee that whenever we have $t \equiv^{\alpha}_{d} t'$ and $d \leq d'$ we also have $t \equiv^{\alpha}_{d'} t'$. In particular, using the trivial sequence $\mathcal{C} := \mathbf{0} = \mathbf{0}$, we can prove that $\mathbf{0} \equiv^{\alpha}_{d} \mathbf{0}$ for all $d \in \mathbb{R}^+$. This could not be inferred if we would impose instead the condition $d = \sum_{j=1}^{n} d_j$. In fact, the case of $\mathbf{0}$ is the only one in which we need the inequality in Def. 9, because for any other tree $t'$ we can apply Def. 8.1 twice, by considering any summand $at$ of $t'$. Instead, in Def. 7 we can apply 7.1 even to $t = \mathbf{0}$, thus we can indeed simply take $d = \sum_{i=1}^{n} d_i$ at the end of the definition.

*Remark 4.* In order to avoid technical difficulties, the authors defining the classical bisimulation distance usually consider processes without termination. Instead, since we have mainly consider finite trees in [17], we needed to take into account termination. Our Def. 7 does not allow any "unexpected" termination when comparing two trees. If we desire to allow some terminations without necessarily entailing an infinite distance, then two simple extensions are possible. We could either establish a fixed payment $f$ (that however will be weighted by the level at which it occurs), for any unexpected termination, including at any $\alpha$-ccd $\mathcal{D}$ all the pairs $(t, \mathbf{0}, f)$, and no proof obligation for them. Or instead, we could pay for any lost action, considering a function $lost : Act \to \mathbb{R}^+$. Then, we could introduce tuples $(at + t', \mathbf{0}, d)$ in the $\alpha$-ccd family $\mathcal{D}$, and for each one of them we need to check that there exist $(t, \mathbf{0}, d_1)$, $(t', \mathbf{0}, d_2) \in \mathcal{D}$ such that $\alpha d_1 + d_2 + lost(a) \leq d$. However, in order to make more understandable the paper, in the following we will not consider any of these extensions.

The next example presents a pair of trees with infinitely many differences, but a finite global bisimulation distance between them.

*Example 7.* (see Fig. 2) Let us consider the domain of actions $(\mathbb{N}, \mathbf{d})$, where $\mathbf{d}$ is the usual distance for numbers, and the trees $t_N = unfold(N)$ and $t_{N'} = unfold(N')$, with $N = \{n_0, n_1\}$, $succ(n_0) = \{(0, n_0), (0, n_1)\}$ and $succ(n_1) = \emptyset$; and $N' = \{n'_0, n'_1\}$, $succ'(n'_0) = \{(0, n'_0), (1, n'_1)\}$ and $succ'(n'_1) = \emptyset$. Then, we have $t_N \equiv_2^{\mathcal{D},1/2} t_{N'}$, using the family $\mathcal{D} = \{(t_N, t_{N'}, 2)\}$. We can prove that this is indeed a $\frac{1}{2}$-ccd, by considering the sequence: $\mathcal{C} := t_N \equiv_1^{\mathcal{D},1/2} t_{N''} \equiv_1^{\mathcal{D},1/2} t_{N'}$, where $t_{N''} = unfold(N'')$, with $N'' = \{n''_0, n''_1, n''_2, n''_3\}$, $succ''(n''_0) = \{(1, n''_1), (0, n''_2)\}$, $succ''(n''_1) = \emptyset$, $succ''(n''_2) = \{(0, n''_2), (0, n''_3)\}$ and $succ''(n''_3) = \emptyset$. The first step is obtained by application of rule 2 in Def.8, while the second one is obtained by application of rule 3, using the fact that $2\frac{1}{2} = 1$.

Note how the coinductive procedure "aggregates" the summands that produce the bound for the distance 2 in a single step. In fact, it is not necessary at all to sum any infinite series, as it would be the case if we would obtain that bound as the limit for the distances between the corresponding finite approximations of the two compared processes. Finally, we can observe that no bound $d < 2$ for the distance can be obtained in this way: $t_N \equiv_d^{\mathcal{D},1/2} t_{N'}$ does not hold for any $d < 2$, because for any such $d$ we have $d < 1 + d/2$; so that, the check for the condition in Def. 9 would fail.

But, making greater the differences in the example above, we can get pairs of trees that are infinitely far away each other, wrt our global bisimulation distance.

*Example 8.* (see Fig. 2) Let us consider the tree $t_N$ from Ex.7, and the tree $t_{N'''} = (N''', succ''', n_{0,2})$ with $N''' = \{n_{0,2}\} \cup \{n_{i,j} \mid i \in \mathbb{N} - \{0\}, j \in \{1, 2\}\}$, $succ'''(n_{i,1}) = \emptyset$ and $succ'''(n_{i,2}) = \{(2^i, n_{i+1,1}), (0, n_{i+1,2})\}$. We have $dist_{\mathbf{d}}^{1/2}(t_N, t_{N'''}) = 1$. Instead, $t_N \equiv_d^{1/2} t_{N'''}$ does not hold for any $d \in \mathbb{R}^+$. As a matter of fact, for the finite projections of these two trees, we have $\pi_k(t_N) \equiv_k^{1/2} \pi_k(t_{N'''})$, for all $k \in \mathbb{N}$, but we do not have $\pi_k(t_N) \equiv_d^{1/2} \pi_k(t_{N'''})$, for any $d < k$.

Based on the notion of bisimilarity, our coinductive global bisimulation distance, and the $\alpha$-ccd used to define it, inherit most of its basic properties, once quantified in the adequate way.

**Definition 10.** *1. We say that a family $\mathcal{D}$ is triangular-transitivity closed (ttc) (resp. + closed (+c)), if for all $(t, t', d), (t', t'', d') \in \mathcal{D}$, we have $(t, t'', d+d') \in \mathcal{D}$ (resp. $(t + t'', t' + t'', d) \in \mathcal{D}$).*

*2. Given a family $\mathcal{D}$, we define its tt-closure as the least family $\mathcal{D}^*$ defined by the clauses i) $\mathcal{D} \subseteq \mathcal{D}^*$; ii) If $(t, t', d), (t', t'', d') \in \mathcal{D}^*$ then $(t, t'', d+d') \in \mathcal{D}^*$.*

*3. Given a family $\mathcal{D}$, we define its +-closure as the family $\mathcal{D}^+ = \{(t + t'', t' + t'', d) \mid (t, t', d) \in \mathcal{D}\}$.*

**Proposition 4.** *If $\mathcal{D}$ is an $\alpha$-ccd, then $\mathcal{D}^*$ and $\mathcal{D}^+$ are too.*

As a consequence, we can assume that any ccd is ttc or +c, when convenient.

**Corollary 1 (triangular-transitivity).** *For any discount factor $\alpha \in (0, 1]$, whenever we have $t \equiv_d^\alpha t'$ and $t' \equiv_{d'}^\alpha t''$, we also have $t \equiv_{d+d'}^\alpha t''$.*

Next, we state the relationship between our global bisimulation distance, bisimilarity and the classical bisimulation distance.

**Proposition 5.** *1. For $t, t' \in FyTrees(\mathbb{A}), \alpha \in (0,1]$, we have $t \sim t' \Leftrightarrow t \equiv_0^\alpha t'$.*
*2. Our global bisimulation distance is greater or equal than the classical one.*

**Corollary 2.** *The topology induced by our global bisimulation distance is strictly finer than that induced by the classical bisimulation distance.*

*Proof.* It is an immediate consequence of Prop.5.2 and the (counter)Ex.8. Taking $\mathbb{R}^+$ as alphabet, and $2^i/k$ as labels of the edges of $t_{N'''}$, we obtain a family of trees $\{t^k_{N'''} \mid k \in \mathbb{N}\}$. Under the classical distance, any open ball centered in $t_N$, contains infinitely many trees $t^k_{N'''}$, but none of them is in any such ball for our global distance.

□

Our coinductive definition of the global bisimulation distance generalizes our operational definition for finite trees.

**Lemma 1.** *Any sequence $\mathcal{S}$ producing $t \rightsquigarrow_{\alpha,d} t'$ can be "factorized" into an "structured" sequence $\mathcal{T} := t = t^{0,2} \rightsquigarrow_{\alpha,d_{11}} t^{1,1} \rightsquigarrow^1_{\alpha,d_{12}} t^{1,2} \rightsquigarrow_{\alpha,d_{21}} \cdots \rightsquigarrow^1_{\alpha,d_{k2}} t^{k,2} \rightsquigarrow_{\alpha,d_{(k+1)1}} t^{k+1,1} = t'$, where the $\sum d_{1i} + \sum d_{2i} = d$, and the distance steps $t^{l,1} \rightsquigarrow^1_{\alpha,d_{l2}} t^{l,2}$ in it are exactly all the first level steps in $\mathcal{S}$. So that, no one of the subsequences producing $t^{l,2} \rightsquigarrow_{\alpha,d_{(l+1)1}} t^{l+1,1}$ contains any first level step.*

**Proposition 6.** *Any sequence $\mathcal{S}^l$ producing $t^{l,2} = \sum_{i=1}^m a_i t_i \rightsquigarrow_{\alpha,d_{(l+1)1}} t^{l+1,1} = \sum_{i=1}^m a_i t'_i$ can be reordered getting an "ordered" sequence $\mathcal{O} := t^{l,2} = \sum_{i=1}^m a_i t_i \rightsquigarrow_{\alpha,d^1_{(l+1)1}} \sum a_i t^1_i \rightsquigarrow_{\alpha,d^2_{(l+1)1}} \sum a_i t^2_i \rightsquigarrow_{\alpha,d^3_{(l+1)1}} \cdots \rightsquigarrow_{\alpha,d^m_{(l+1)1}} \sum a_i t^m_i = \sum_{i=1}^m a_i t'_i = t^{l+1,1}$, where $\sum_{j=1}^m d^j_{(l+1)1} = d$, $t^j_i = t'_i \ \forall j \leq i$, and $t^j_i = t_i \ \forall j > i$.*

*This means that for each $j \in \{1, \ldots, m\}$ $\sum a_i t^{j-1}_i \rightsquigarrow_{\alpha,d^j_{(l+1)1}} \sum a_i t^j_i$ corresponds to $a_j t_j \rightsquigarrow_{\alpha,d^j_{(l+1)1}} a_j t'_j$, so that the distance steps in the former are exactly those from $\mathcal{S}^l$ working at the corresponding summand $a_j t_j$ of $t$. As a consequence, for each $j \in \{1, \ldots, m\}$ we also have $t_j \rightsquigarrow_{\alpha,(d^j_{(l+1)1})/\alpha} t'_j$, which is obtained by removing the common prefix $a_j$ from the steps of the subsequence generating $a_j t_j \rightsquigarrow_{\alpha,d^j_{(l+1)1}} a_j t'_j$.*

**Proposition 7.** *For $t, t' \in FTrees(\mathbb{A})$, the operational (Def.7) and the coinductive definition of our distance between trees coincide, that means $t \equiv_d^\alpha t' \Leftrightarrow t \rightsquigarrow_{\alpha,d} t'$.*

*Proof.* $\underline{\Rightarrow |}$ Given an $\alpha$-ccd relating finite trees with $(t, t', d) \in \mathcal{D}$, we can "unfold" the corresponding sequence, $\mathcal{C}$, checking $t \equiv_d^{\mathcal{D},\alpha} t'$, into a sequence of distance steps, $\mathcal{S}$, proving that $t \rightsquigarrow_{\alpha,d} t'$. We proceed by induction on $depth(t)$, as follows.

Let $t^i \equiv_{d_i}^{\mathcal{D},\alpha} t^{i+1}$ be an intermediate step in the coinductive sequence $\mathcal{C}$. If $depth(t^i) = 0$, we will trivially get $t^i \rightsquigarrow_{\alpha,d_i} t^{i+1}$. For $depth(t) \geq 1$, we apply rule 3 in Def.8, getting $t^i = t^i_1 + at_1 \equiv_{d_i}^{\mathcal{D},\alpha} t^i_1 + at'_1 = t^{i+1}$ for $(t_1, t'_1, d_i/\alpha) \in \mathcal{D}$. By

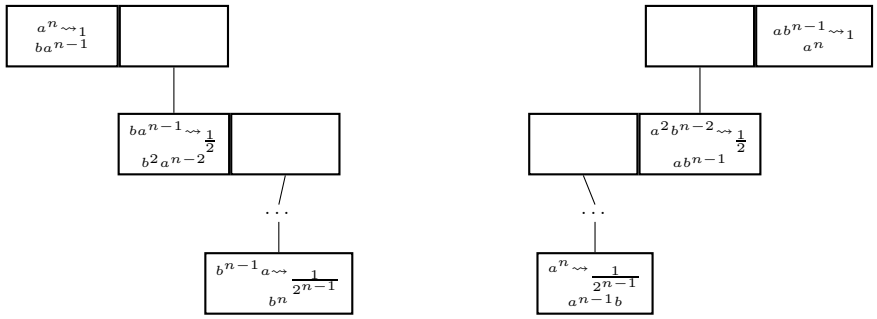



**Fig. 3.** Arborescent presentation of the operational sequences induced by an $\alpha$-ccd

applying the induction hypothesis, we get $t_1 \rightsquigarrow_{\alpha,d_i/\alpha} t'_1$, and using rules 4 and 3 in Def.7, we obtain the desired result $t^i = t_1^i + at_1 \rightsquigarrow_{\alpha,d_i} t_1^i + at'_1 = t^{i+1}$.

$\underline{\Leftarrow |}$ Given a sequence of distance steps, $\mathcal{S}$, proving that $t \rightsquigarrow_{\alpha,d} t'$, we can "fold" it into a coinductive sequence, $\mathcal{C}$, checking $t \equiv_d^{\mathcal{D},\alpha} t'$. For each $(t,t',d) \in \mathcal{D}$ we consider the factorization of the sequence $\mathcal{S}$ and its reordering as done in Prop.6. We get $t = \sum_{i\in I_0} a_i t_i \rightsquigarrow_{\alpha,d^i_{02}} \sum_{i\in I_0} a_i t'_i \rightsquigarrow^1_{\alpha,d^i_{11}} \sum_{i\in I_1} a_i t_i \rightsquigarrow_{\alpha,d^i_{12}} \sum_{i\in I_1} a_i t'_i \rightsquigarrow^1_{\alpha,d^i_{21}} \cdots \rightsquigarrow_{\alpha,d^i_{(k+1)2}} \sum_{i\in I_{k+1}} a_i t'_i = t'$, where for each sequence $\sum_{i\in I_j} a_i t_i \rightsquigarrow_{\alpha,d^i_{j2}} \sum_{i\in I_j} a_i t'_i$ and each $i \in I_j$, we have $t_i \rightsquigarrow_{\alpha,d^i_{j2}/\alpha} t'_i$ with $\sum_{i\in I_j} d^i_{j2} = d_{j2}$. Now, applying the induction hypothesis, we have $(t_i, t'_i, d^i_{j2}/\alpha) \in \mathcal{D}$, for all $i \in I_j$, so that $\sum a_i t_i \equiv^{\mathcal{D},\alpha}_{\alpha d_1} \sum a_i t^1_i \equiv^{\mathcal{D},\alpha}_{\alpha d_2} \sum a_i t^2_i \equiv^{\mathcal{D},\alpha}_{\alpha d_3} \cdots \equiv^{\mathcal{D},\alpha}_{\alpha d_{|I_j|}} \sum a_i t_i^{|I_j|} = \sum a_i t'_i$.

Therefore, each sequence $\sum_{i\in I_j} a_i t_i \rightsquigarrow_{\alpha,d^i_{j2}} \sum_{i\in I_j} a_i t'_i$ at the factorization above can be substituted by a sequence of $|I_j|$ valid coinductive steps, getting a total distance $\sum_{j=0}^{k+1} \sum_{i\in I_j} d^i_{j1} + \sum_{j=0}^{k+1} \sum_{k=1}^{|I_j|} \alpha d_k = d$.

□

Next, a pair of examples to illustrate the unfolding and folding procedures.

*Example 9.* Let us consider the family of trees $\{a^n \mid n \in \mathbb{N}\}$, defined by $a^0 = \mathbf{0}$ and $a^{n+1} = aa^n$. We define $b^n$ in an analogous way. Now, if $\mathbf{d}(a,b) = 1$, we have $a^n \equiv_2^{1/2} b^n \ \forall n \in \mathbb{N}$, using $\mathcal{D} = \{(a^n, b^n, 2) \mid n \in \mathbb{N}\}$, that is shown to be a $\frac{1}{2}$-ccd by considering $\mathbf{0}$=$\mathbf{0}$ and the sequences $\mathcal{C}^n := a^n = aa^{n-1} \equiv_1^{\mathcal{D},1/2} ba^{n-1} \equiv_{1/2\cdot 2}^{\mathcal{D},1/2} bb^{n-1} = b^n$. Using the notion of unfolding above, we get the operational sequences $\mathcal{S}^n := a^n \rightsquigarrow^1_{\frac{1}{2},1} ba^{n-1} \rightsquigarrow^1_{\frac{1}{2},1/2} b^2a^{n-2} \rightsquigarrow^1_{\frac{1}{2},1/4} \cdots \rightsquigarrow^1_{\frac{1}{2},\frac{1}{2^{n-1}}} b^n$. If we preserve the structure of the sequences $\mathcal{C}^n$, whose unfolding produce these operational sequences, we can visualize them in a arborescent way –see Fig.3–. The structure reminds that of *B-trees*, where we have nodes containing keys and pointers between them. The last give access to the elements in between the former that are located at nodes at "lower" levels. By means of the (inorder) traversing of the obtained tree we recover the original operational sequences.

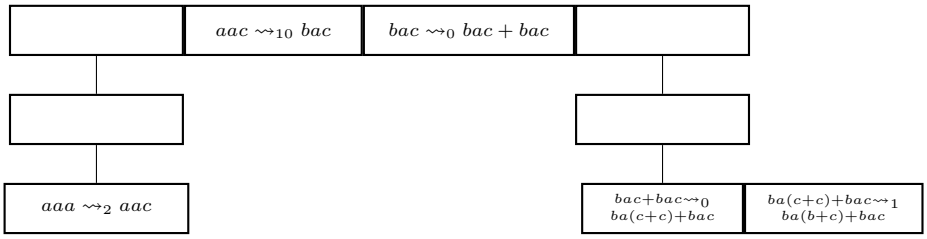



**Fig. 4.** Arborescent presentation of the operational sequence $\mathcal{S}$ in Ex.10

We are just "pushing the distance steps down" that correspond to "lower" levels, by introducing arcs that "move" the steps to the corresponding level. But whenever we have several steps in a row, that are not first level, then we group all of them introducing a single arc. We proceed in the same way down and down, introducing a "leaf" whenever we arrive to the level of a distance step, and new arcs going down, if there are other steps at the group at lower levels, either before or after the one which generated that leaf.

It is interesting to observe that we can also turn $a^n$ into $b^n$ in the opposite way, which means to use the same $\frac{1}{2}$-ccd, but a different sequence to check that it is indeed a $\frac{1}{2}$-ccd. We take now $\mathcal{C}'^n := a^n = aa^{n-1} \equiv^{\mathcal{D},1/2}_{1/2\cdot 2} ab^{n-1} \equiv^{\mathcal{D},1/2}_{1} bb^{n-1} = b^n$. Its unfolding produces the "symmetric" tree on the right of Fig.3. Certainly, you can also recognize the reversibility of the operational sequences: by reading $\mathcal{C}'^n$ from right to left we recover $\mathcal{C}^n$, simply interchanging the roles of $a$ and $b$.

*Example 10.* Taking $\mathbb{A} = \{a, b, c\}$ with $\mathbf{d}(a, c) = 8$, $\mathbf{d}(b, c) = 4$ and $\mathbf{d}(a, b) = 10$, we obtain $aaa \rightsquigarrow^1_{\frac{1}{2},13} ba(b + c) + bac$, by means of the sequence $\mathcal{S} := aaa \rightsquigarrow^1_{\frac{1}{2},2} aac \rightsquigarrow^1_{\frac{1}{2},10} bac \rightsquigarrow^1_{\frac{1}{2},0} bac + bac \rightsquigarrow^1_{\frac{1}{2},0} ba(c + c) + bac \rightsquigarrow^1_{\frac{1}{2},1} ba(b + c) + bac$. In Fig.4 we see its arborescent presentation whose folding generates the $\frac{1}{2}$-ccd $\mathcal{D} = \{(aaa, ba(b+c)+bac, 13), (aaa, aac, 2), (aa, ac, 4), (a, c, 8), (bac+bac, ba(b+c) + bac, 1), (ac, a(b + c), 2), (c, (b + c), 4)\}$.

This is a more illustrative example of the general form of these arborescent presentations: we can have several "leaves" together with no arc in between them, when they correspond to several consecutive steps of the sequence at the current level. We can also have "degenerated" nodes, with a single arc down the tree, which corresponds to a subsequence of steps with none at the current level.

Even if Prop.7 only concerns finite trees, it reveals the duality between induction and coinduction, which is particularly interesting in the infinite case.

*Example 11.* Let us consider the tree $a^\infty = unfold(N_{1,\infty})$, with $N_{1,\infty}$ as in Ex.1. In an analogous way, we obtain the tree $b^\infty$. We have $\pi_n(a^\infty) = a^n$, with $a^n$ as in Ex.9. Therefore, $a^\infty$ can be seen as the limit of its projections, and as we had $a^n \equiv^{1/2}_2 b^n$, we have also $a^\infty \equiv^{1/2}_2 b^\infty$. This can be proved by means of

the (trivial!) collection $\mathcal{D} = \{(a^\infty, b^\infty, 2)\}$. We can check that $\mathcal{D}$ is indeed an $\frac{1}{2}$-ccd using the sequence $\mathcal{C} := a^\infty = aa^\infty \equiv_1^{\mathcal{D},1/2} ba^\infty \equiv_{\frac{1}{2}\cdot 2}^{\mathcal{D},1/2} bb^\infty = b^\infty$.

Now, the (infinite!) "unfolding" of $\mathcal{C}$ would produce an infinite tree, that would "generate" an "infinite" operational sequence, which (intuitively) "converges" to $b^\infty$, and "gives" us the bound 2 for the distance between $a^\infty$ and $b^\infty$. But our coinductive approach avoids the consideration of these limits. Moreover, the "traversing" of the arborescent presentations of the sequences, needed in many of our coinductive proofs, would produce "nested" infinite sequences much more difficult to cover without the coinductive approach.

*Example 12.* Let us take $\mathbb{A} = \{a, b, c, d\}$ with $\mathbf{d}(a, b) = 4$, $\mathbf{d}(c, d) = 1$. We can prove $ac^\infty + ad^\infty \equiv_6^{1/2} bc^\infty + bd^\infty$, using $\mathcal{D} = \{(ac^\infty + ad^\infty, bc^\infty + bd^\infty, 6), (c^\infty, d^\infty, 2)\}$, where the second triple in $\mathcal{D}$ is checked as in Ex.11; while for the first one we consider the coinductive sequence $\mathcal{C} := ac^\infty + ad^\infty \equiv_1^{1/2} ac^\infty + ac^\infty \equiv_0^{1/2} ac^\infty \equiv_4^{1/2} bc^\infty \equiv_0^{1/2} bc^\infty + bc^\infty \equiv_1^{1/2} bc^\infty + bd^\infty$.

Anyway, out of the informal level (where it is quite useful!) and the finite case (where it is sound), we will avoid the use of this unfolding in our formal developments. However, the following definition formalizes the use of finite unfolding, getting a generalized characterization of the relations $\equiv_d^\alpha$. It combines our two approaches (inductive, Def.7, and coinductive, Def.8,9) in a more flexible way; now operational steps can be used, not only at the first level of the trees, but also at any lower level.

**Definition 11.** *We consider the extension of the family of relations $\langle \rightsquigarrow_{\alpha,d}^1 | \ d \in \mathbb{R}^+ \rangle$ in Def.7 to FyTrees(A). Now, given a family $\mathcal{D} = \{(t_i, t_i', d_i) \mid i \in I\}$ with $t_i, t_i' \in FyTrees(\mathbb{A})$ and $d_i \in \mathbb{R}^+$, we define the family of relations $\widehat{\equiv}_d^{\mathcal{D},\alpha}$, by:*

1. *$t \rightsquigarrow_{\alpha,d}^1 t'$ implies $t \,\widehat{\equiv}_d^{\mathcal{D},\alpha} t'$.*
2. *For all $(t, t', d) \in \mathcal{D}$ we have $(\sum_{j\in J} a_j t_j) + at \,\widehat{\equiv}_{\alpha\cdot d}^{\mathcal{D},\alpha} (\sum_{j\in J} a_j t_j) + at'$.*

*Now, we can proceed exactly as in Def.9, using the relations $\widehat{\equiv}_d^{\mathcal{D},\alpha}$ instead of $\equiv_d^{\mathcal{D},\alpha}$, getting the family of relations $\widehat{\equiv}_d^\alpha$.*

**Proposition 8.** *For all $d \in \mathbb{R}^+$, $\alpha \in (0, 1]$, the relations $\equiv_d^\alpha$ and $\widehat{\equiv}_d^\alpha$ are equal.*

The (simple) proof of this result uses the fact that operational steps not at the first level of the trees, can be "hidden" into nested coinductive steps. However, their explicit use will produce in some cases much shorter and clearer proofs.

## 5 On the continuity of the global bisimulations distance

We have proved in Prop.7 the consistency between our inductive and coinductive definitions for finite trees. This can be turned into the limit by considering the coinductive definition and the (finite) projections of infinite processes.

**Proposition 9.** *For any $\alpha$-ccd $\mathcal{D}$, the projected family $\pi(\mathcal{D}) = \{(\pi_n(t), \pi_n(t'), d) \mid (t, t', d) \in \mathcal{D},\ n \in \mathbb{N}\}$ is an $\alpha$-ccd that proves $t \equiv_d^\alpha t' \Rightarrow \forall n \in \mathbb{N}\ \pi_n(t) \equiv_d^\alpha \pi_n(t')$.*

*Proof.* Let $\mathcal{C} := t = t^0 \equiv_{d_1}^{\mathcal{D},\alpha} \ldots \equiv_{d_k}^{\mathcal{D},\alpha} t^k = t'$ be the sequence proving that $(t, t', d) \in \mathcal{D}$ satisfies the condition in order $\mathcal{D}$ to be an $\alpha$-ccd. Then each projected sequence $\pi_n(\mathcal{C}) := \pi_n(t) = \pi_n(t^0) \equiv_{d_1}^{\pi(\mathcal{D}),\alpha} \ldots \equiv_{d_k}^{\pi(\mathcal{D}),\alpha} \pi_n(t^k) = \pi_n(t')$ proves that $(\pi_n(t), \pi_n(t'), d) \in \pi(\mathcal{D})$ satisfies the condition in order $\pi(\mathcal{D})$ to be an $\alpha$-ccd. It is clear that the projection under $\pi_n$ of any first level step in $\mathcal{C}$, is also a valid step in $\pi_n(\mathcal{C})$. Moreover, any coinductive step in $\mathcal{C}$ using $(t_1, t_1', d) \in \mathcal{D}$, can be substituted by the corresponding projected step, that uses $(\pi_{n-1}(t_1), \pi_{n-1}(t_1'), d) \in \pi(\mathcal{D})$.

□

*Remark 5.* Alternatively, we can consider for each $n \in \mathbb{N}$ a family $\mathcal{D}_n = \pi_n(\mathcal{D}) = \{(\pi_m(t), \pi_m(t'), d) \mid (t, t', d) \in \mathcal{D},\ m \in \mathbb{N} \wedge m \leq n\}$, using the fact that the subtrees of a projection $\pi_n(t)$ are also projections $\pi_m(t'')$ of subtrees $t''$ of $t$, for some $m < n$. These families satisfy $\pi_m(\mathcal{D}) \subseteq \pi_n(\mathcal{D})$, whenever $m \leq n$.

*Example 13.* Let us consider the trees $a^\infty$ and $b^\infty$ in Ex.11 and the $\frac{1}{2}$-ccd $\mathcal{D} = \{(a^\infty, b^\infty, 2)\}$ that proves $a^\infty \equiv_2^{1/2} b^\infty$, by means of the sequence $\mathcal{C} := a^\infty = aa^\infty \equiv_1^{\mathcal{D},1/2} ba^\infty \equiv_{\frac{1}{2}\cdot 2}^{\mathcal{D},1/2} bb^\infty = b^\infty$. Now for the families $\mathcal{D}_n = \pi_n(\mathcal{D})$ in Remark 5, we have $\mathcal{D}_n = \{(a^m, b^m, 2) \mid m \leq n\}$, which gives us $a^n \equiv_2^{1/2} b^n$ by means of the sequence $\mathcal{C}^n = \pi_n(\mathcal{C}) := a^n = aa^{n-1} \equiv_1^{\mathcal{D}_n,1/2} ba^{n-1} \equiv_{\frac{1}{2}\cdot 2}^{\mathcal{D}_n,1/2} bb^{n-1} = b^n$.

We conjecture that the converse of Prop.9 asserting the continuity of our coinductive distance, is also true. Unfortunately, the proof of this result is being much more complicated than we expected. Our idea, is to use the reasoning in the proof of Prop.9 in the opposite direction and the correspondence between operational and coinductive sequences in the finite case. As far as we have a collection of "uniform"[2] operational sequences $\mathcal{S}^n := \pi_n(t) \rightsquigarrow_{\alpha,d} \pi_n(t')$, we could "overlap" all of them getting an infinite tree as that in Fig.3. By "folding" this tree we obtain the coinductive sequence $\mathcal{C}$, proving $t \equiv_d^\alpha t'$. Next we provide a simple example.

*Example 14.* Let us consider the trees $t = ac^\infty + ad^\infty$ and $t' = bc^\infty + bd^\infty$, as in Ex.12, and the same distance $\mathbf{d}$ as there. Then we have:

$$\pi_1(t) = a + a \rightsquigarrow_{\frac{1}{2},0}^{(1)} a \rightsquigarrow_{\frac{1}{2},4}^{(1)} b \rightsquigarrow_{\frac{1}{2},0}^{(1)} b + b = \pi_1(t'),$$

$$\pi_2(t) = ac + ad \rightsquigarrow_{\frac{1}{2},\frac{1}{2}}^{(2)} ac + ac \rightsquigarrow_{\frac{1}{2},0}^{(1)} ac \rightsquigarrow_{\frac{1}{2},4}^{(1)} bc \rightsquigarrow_{\frac{1}{2},0}^{(1)} bc + bc \rightsquigarrow_{\frac{1}{2},\frac{1}{2}}^{(2)} bc + bd = \pi_2(t'),$$

$$\pi_3(t) = acc + add \rightsquigarrow_{\frac{1}{2},\frac{1}{2}\cdot 1}^{(2)} acc + acd \rightsquigarrow_{\frac{1}{2},\frac{1}{4}\cdot 1}^{(3)} acc + acc \rightsquigarrow_{\frac{1}{2},0}^{(1)} acc \rightsquigarrow_{\frac{1}{2},4}^{(1)}$$
$$bcc \rightsquigarrow_{\frac{1}{2},0}^{(1)} bcc + bcc \rightsquigarrow_{\frac{1}{2},\frac{1}{2}\cdot 1}^{(2)} bcc + bdc \rightsquigarrow_{\frac{1}{2},\frac{1}{4}\cdot 1}^{(3)} bcc + bdd = \pi_3(t').$$

[2] Uniformity here means that for any $n, k \in \mathbb{N}$ with $k \geq n$ the steps of all the sequences $\mathcal{S}^k$ corresponding to the first $n$-levels are always the same.

We have included the superscripts $(k)$ to indicate at which level we apply each transformation step. Each of these sequences can be obtained from the following one by removing the steps marked with $(i+1)$ and applying $\pi_i$.

Now, if we consider the operational sequences, $\mathcal{S}^n$, relating $\pi_n(t)$ and $\pi_n(t')$, for any $n \in \mathbb{N}$, we obtain $\pi_n(t) \rightsquigarrow_{\frac{1}{2},d_n} \pi_n(t')$, for some $d_n < 6$. For instance, we get $\pi_1(t) \rightsquigarrow_{\frac{1}{2},4} \pi_1(t')$, $\pi_2(t) \rightsquigarrow_{\frac{1}{2},5} \pi_2(t')$ and $\pi_3(t) \rightsquigarrow_{\frac{1}{2},5.5} \pi_3(t')$. The obtained distances form an increasing (but bounded) sequence, since each of the operational sequences expand the former ones, adding new costs caused by the (new) differences at the bottom levels.

Turning these operational sequences into coinductive ones, $\mathcal{C}^n$, as in Prop.7 we obtain the proof of $\pi_n(t) \equiv_6^{1/2} \pi_n(t')$, for all $n \in \mathbb{N}$. Here, it is convenient to use the (same) value 6 at all the cases.

$$\mathcal{C}^1 := a + a \equiv_1^{1/2} a + a \equiv_0^{1/2} a \equiv_4^{1/2} b \equiv_0^{1/2} b + b \equiv_1^{1/2} b + b,$$
$$\mathcal{C}^2 := ac + ad \equiv_1^{1/2} ac + ac \equiv_0^{1/2} ac \equiv_4^{1/2} bc \equiv_0^{1/2} bc + bc \equiv_1^{1/2} bc + bd,$$
$$\mathcal{C}^3 := acc + add \equiv_1^{1/2} acc + acc \equiv_0^{1/2} acc \equiv_4^{1/2} bcc \equiv_0^{1/2} bcc + bcc \equiv_1^{1/2} bcc + bdd.$$

We expect that whenever we have $\pi_n(t) \equiv_d^{\alpha} \pi_n(t')\ \forall n \in \mathbb{N}$ there will be a collection of uniform sequences proving these facts. For $t, t' \in FTrees(\mathbb{A})$ with $||t||_n, ||t'||_n \leq l$ and $t \rightsquigarrow_{\alpha,d} t'$, we should prove this by means of a sequence $\mathcal{S}$ that only uses intermediate trees $t''$ with $||t''||_n \leq f(l,n)$, for a certain function $f$. But, the existence of such a uniform bound is still to be proved.

## 6 Conclusions and future work

We have presented a coinductive characterization of our global bisimulation distance, that previously we presented in an operational and an algebraic way. So, we extend our distance to the case of infinite trees without needing to introduce any complex notion of limit of our finite transformations generating the distances between finite trees. The coinductive approach makes the work in a much easier way. Our coinductive distances are always "sound" wrt the distances between their respective finite approximations. We expect that the "completeness" result, ending the proof of continuity, will also be true.

Besides the work devoted to complete the proof of the continuity theorem, now we are working in two complementary directions. On the one hand, we will try to apply our coinductive distance in order to define distances for testing, which should state how far away is a process to pass the tests imposed by any specification. On the other hand, we will continue the theoretical study of our coinductive distances. We consider that the results here are very promising, showing a new field of application of coinductive techniques into the study of the semantics of processes. We hope that much more will be shortly coming.

## References

1. P. Cerný, T. A. Henzinger, and A. Radhakrishna. Quantitative simulation games. In *EMAP 2010*, volume 6200 of *LNCS*, pages 42–60. Springer, 2010.

2. P. Cerný, T. A. Henzinger, and A. Radhakrishna. Simulation distances. In *CONCUR 2010*, volume 6269 of *LNCS*, pages 253–268. Springer, 2010.
3. L. de Alfaro, T. A. Henzinger, and R. Majumdar. Discounting the future in systems theory. In *ICALP 2003*, volume 2719 of *LNCS*, pages 1022–1037. Springer, 2003.
4. L. de Alfaro, R. Majumdar, V. Raman, and M. Stoelinga. Game relations and metrics. In *LICS 2007*, pages 99–108. IEEE Computer Society, 2007.
5. J. Desharnais, V. Gupta, R. Jagadeesan, and P. Panangaden. Metrics for labeled markov systems. In *CONCUR 1999*, volume 1664 of *LNCS*, pages 258–273. Springer, 1999.
6. J. Desharnais, F. Laviolette, and M. Tracol. Approximate analysis of probabilistic processes: Logic, simulation and games. In *QEST 2008*, pages 264–273. IEEE Computer Society, 2008.
7. U. Fahrenberg, A. Legay, and C. R. Thrane. The quantitative linear-time–branching-time spectrum. In *FSTTCS 2011*, volume 13 of *LIPIcs*, pages 103–114. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2011.
8. U. Fahrenberg, C. R. Thrane, and K. G. Larsen. Distances for weighted transition systems: Games and properties. In *QAPL 2011*, pages 134–147, 2011.
9. A. Giacalone, C. Jou, and S. A. Smolka. Algebraic reasoning for probabilistic concurrent systems. In *Proc. IFIP TC2 Working Conference on Programming Concepts and Methods*, pages 443–458. North-Holland, 1990.
10. Thomas A. Henzinger and Jan Otop. From model checking to model measuring. In *CONCUR 2013*, volume 8052 of *LNCS*, pages 273–287. Springer, 2013.
11. B. Jacobs. Exercises in coalgebraic specification. In *Algebraic and Coalgebraic Methods in the Mathematics of Program Construction 2000*, volume 2297 of *LNCS*, pages 237–280. Springer, 2000.
12. B. Jacobs. Introduction to coalgebra. towards mathematics of states and observations. In *http://www.cs.ru.nl/B.Jacobs/CLG/JacobsCoalgebraIntro.pdf*, 2012.
13. A. Kiehn and S. Arun-Kumar. Amortised bisimulations. In *FORTE 2005*, volume 3731 of *LNCS*, pages 320–334. Springer, 2005.
14. R. Milner. Communication and concurrency. Prentice Hall, 1989.
15. M. Nielsen and C. Clausen. Bisimulation, games, and logic. In *Results and Trends in Theoretical Computer Science*, Vol. 812 of *LNCS*, pages 289–306. Springer, 1994.
16. D.M.R. Park. Concurrency and automata on infinite sequences. In *Theoretical Computer Science, 5th Gl-Conf.*, Vol. 104 of *LNCS*, pages 167–183. Springer, 1981.
17. D. Romero-Hernández and D. de Frutos-Escrig. Defining distances for all process semantics. In *FMOODS/FORTE 2012*, volume 7273 of *LNCS*, pages 169–185. Springer, 2012.
18. D. Romero-Hernández and D. de Frutos-Escrig. Distances between processes: A pure algebraic approach. In *WADT 2012*, volume 7841 of *LNCS*, pages 265–282. Springer, 2012.
19. J. J. M. M. Rutten. Universal coalgebra: a theory of systems. *Theor. Comput. Sci.*, 249(1):3–80, 2000.
20. D. Sangiorgi. Advanced topics in bisimulation and coinduction. Cambridge Tracts in Theoretical Computer Science, 2011.
21. C. Stirling. The joys of bisimulation. In *MFCS 1998*, volume 1450 of *LNCS*, pages 142–151. Springer, 1998.
22. C. Stirling. Bisimulation, modal logic and model checking games. *Logic Journal of the IGPL*, 7(1):103–124, 1999.
23. M. Ying and M. Wirsing. Approximate bisimilarity. In *AMAST 2000*, volume 1816 of *LNCS*, pages 309–322. Springer, 2000.