Distances between Processes: A Pure Algebraic Approach

David Romero Hernández, David De Frutos Escrig

Distances Between Processes: a Pure Algebraic Approach *

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. Recently, we have presented operational and denotational definitions for distances between processes corresponding to any semantics in the ltbt-spectrum. In this paper, we develop a general algebraic framework to define distances between terms from any arbitrary signature. We apply this framework obtaining a new algebraic characterization of our previous distances. Moreover, we prove the generality of our approach developing an algebraic characterization of the distances based on the (bi)simulation game by other authors.

1 Introduction

In order to define an (abstract) semantics for processes, we need just to define an adequate equivalence relation, $\equiv$, relating the processes in some universe, *Proc*. Then, the values of this semantics are just the corresponding equivalence classes, and two processes have the same semantics if and only if they are equivalent. Once we have fixed such a semantics we can compare two processes, but the output of this comparison is just a Boolean value. In particular, when two processes are not equivalent, we do not have a general way to measure "how far away" they are from being equivalent.

There are several papers which introduce several distances between processes based on the (bi)simulation game –see e.g. [3, 5]–. Even before, Ying and Wirsing studied approximate bisimilarity, following similar but simpler ideas [12, 11]. Our work started by considering those distance games, where one essentially plays the (bi)simulation game, but with the "defender" having the possibility of replying a move of the "attacker" without matching exactly his move. In such a case, he should pay to the attacker some quantity, depending on the mismatch (distance) between the two involved actions.

It is well-known the use of equivalence relations to formalize the notion of implementation: a process implements some specification (given by another process) when they are equivalent w.r.t. the adequate semantics. But, if we follow this approach, we have no flexibility at all: our process has to satisfy in a precise

* Partially supported by the Spanish projects TESIS (TIN2009-14312-C02-01), DESAFIOS10 (TIN2009-14599-C03-01) and PROMETIDOS S2009/TIC-1465.

way all the constraints imposed by the specification, or it will not be a correct implementation. Instead, in the real world, we often find other more flexible quality requirements, where the specification establishes the ideal behavior of the system, but some (limited) deviations from it are allowed, without invalidating the adequateness of the implementation.

We need a notion of distance between processes to make precise how far away two processes are from being equivalent w.r.t. some given semantics. It is true that metrics have been used for a long time to formalize the semantics of infinite processes, by means of (the limit of) those of their finite approximations. But these metrics were just a very particular case that only cared for "the first" disagreement between the compared processes. Instead, now we look for more general distances, which moreover should be applicable to any syntactic process algebra (i.e., to any signature Σ) and any semantics (based, for instance, on the axiomatization of the desired semantics).

We have already introduced the basic operational ideas of our approach in [8]. It is true that the most flexible way to capture a semantics for processes, $\mathcal{L}$, is based on the use of an adequate preorder $\subseteq_{\mathcal{L}}$, and not just an equivalence relation $\equiv_{\mathcal{L}}$.

However, we prefer to start our presentation in Section 2 using just the better known *equational calculus*. Next, in Section 3 we will see that a simple modification of the proof system defining the classical equational calculus (see e.g. [7]), produces a general algebraic framework to define distances between processes w.r.t. any semantics. In particular, in Section 4 we study in detail the case of the bisimulation semantics. Later, in Section 5, we will see how we can easily generalize all our algebraic presentation to the inequational framework. Moreover, we define our distances for the rest of the semantics in the ltbt-spectrum. To show the flexibility of our approach, in Section 6 we see how the classic distances based on the (bi)simulation game, can be also defined in our algebraic framework. We conclude with our conclusions and some future work.

2 Preliminaries

A careful presentation of the equational calculus with application to the (testing) semantics of processes can be found in [7]. Next, we will only remind the definitions of the main concepts needed to develop that theory.

Definition 1. *1. A* signature*, Σ, is a set of formal* functional symbols*. Each functional symbol has associated a natural number which is its* arity*. We use Σ_n to denote the set of symbols in Σ of arity n.*

2. If Σ is a signature, a Σ-algebra is a pair $\langle A, \Sigma_A \rangle$ where A is a set, the domain or support of the algebra, and Σ_A is a set of internal functions. A Σ-algebra is simply an interpretation of each operation in Σ, respecting of course its arity.

3. There is a particularly important Σ-algebra, called the term algebra *for Σ, and denoted by T_Σ, whose support is the set of terms freely generated by Σ.*

Some particular collections of Σ-algebras can be singled out by means of equations. An equation is determined by a pair of terms which may contain variables. We consider an (arbitrary) set of variables, X, and the set of terms with variables $T_\Sigma(X)$. $T_\Sigma(X)$ can be obtained by extending the signature Σ adding these variables with null arity. In fact $T_\Sigma(X)$ is just an algebra, where $\Sigma(X)$ is the classic notation for the signature $\Sigma \cup X$ which add to Σ each $x \in X$ as a new function symbol of arity 0.

Given an equation $t \equiv t'$ with $t, t' \in T_\Sigma(X)$, we say that a Σ-algebra $\langle A, \Sigma_A \rangle$ satisfies it, when the values of both t and t' under any valuation, which assigns values in A to the variables $x \in X$, are the same. Given a set of equations E, $\mathcal{C}(E)$, is the class of Σ-algebras satisfying the equations E. The initial algebra of $\mathcal{C}(E)$ can be presented as a quotient algebra $T_\Sigma / \equiv_E$ for some particular congruence $\equiv_E$. We can obtain this congruence by means of the equational deduction system $\mathbf{DED}(E)$ in Fig. 1 whereby the equations in E may be used to derive statements of the form $t \equiv t'$, with $t, t' \in T_\Sigma$.

1. Reflexivity $\dfrac{}{t \equiv t}$ 2. Symmetry $\dfrac{t \equiv t'}{t' \equiv t}$ 3. Transitivity $\dfrac{t \equiv t' \quad t' \equiv t''}{t \equiv t''}$
4. Substitution $\dfrac{t_1 \equiv t'_1, \ldots, t_k \equiv t'_k}{f(t_1, \ldots, t_k) \equiv f(t'_1, \ldots, t'_k)}$ for every $f \in \Sigma$ of arity k.
5. Instantiation $\dfrac{t \equiv t'}{t\rho \equiv t'\rho}$ for every substitution ρ.
6. Equations $\dfrac{}{t \equiv t'}$ for every equation $\langle t, t' \rangle \in E$

Fig. 1. The proof system $\mathbf{DED}(E)$ in [7]

We write $\vdash_E t \equiv t'$ if we can derive $t \equiv t'$; and then, we say that $t \equiv t'$ is a theorem of $\mathbf{DED}(E)$. Obviously, we can see each derivable theorem as the pair of a relation $\equiv_E$, which due to 1-3 is an equivalence relation, and as a result of 4 a Σ-congruence.

As we have said, the idea in this paper is to define distance between processes. Then, we need to extend the concept of Σ-algebra with a notion of distance. This distance allows us to measure how far away is a process p of being equivalent to another process q.

In [8] we have considered as processes the terms generated by the free $(\mathbf{0}, Act, +)$-algebra, which correspond to the classic domain of $BCCSP(Act)$ processes.

Definition 2. *Given a set of actions Act, the set $BCCSP(Act)$ of processes is that defined by the BNF-grammar: $p ::= \mathbf{0} \mid ap \mid p + q$. The very well known operational semantics of BCCSP [10, 4] is defined by:*

$$(1)\ \frac{}{ap \xrightarrow{a} p} \qquad (2)\ \frac{p \xrightarrow{a} p'}{p + q \xrightarrow{a} p'} \qquad (3)\ \frac{q \xrightarrow{a} q'}{p + q \xrightarrow{a} q'}$$

Based on this operational semantics we can define all the semantics in the ltbt-spectrum [10]. In particular, we have studied the case of the finest of these semantics, that is the bisimulation semantics.

Definition 3 (Bisimulation). *A bisimulation between processes is a relation B such that, whenever pBq, we have:*

- *for every $a \in Act$, if $p \xrightarrow{a} p'$ there exists some q', with $q \xrightarrow{a} q'$ and $p'Bq'$.*
- *for every $a \in Act$, if $q \xrightarrow{a} q'$ there exists some p', with $p \xrightarrow{a} p'$ and $p'Bq'$.*

We have defined distances between processes corresponding to any semantics defined by a preorder $\subseteq_{\mathcal{L}}$, this covering in particular the case of equivalence relations, such as bisimulation . Both are defined in an operational way based on transformations between processes, and in a denotational way, using SOS-rules. Next, we recall this second definition

Definition 4 ([8]). *Given a semantics $\mathcal{L}$, defined by a preorder $\subseteq_{\mathcal{L}}$, coarser than bisimulation, we say that a process q is at distance at most $m \in \mathbb{N}$ of being better than some other p, w.r.t. the semantics $\mathcal{L}$ and the distance between actions $\overline{d}$, and then we write $d^{\mathcal{L}}_{\overline{d}}(p,q) \leq n$, if we can infer $p \sqsubseteq^{\mathcal{L}}_{n} q$, by applying the following rules:*

$$(1)\ \frac{p \subseteq_{\mathcal{L}} q}{p \sqsubseteq^{\mathcal{L}}_{n} q} \quad (2)\ \frac{p \sqsubseteq^{\mathcal{L}}_{n} q}{ap \sqsubseteq^{\mathcal{L}}_{n+\overline{d}(b,a)} bq} \quad (3)\ \frac{p \sqsubseteq^{\mathcal{L}}_{n} p'}{p+q \sqsubseteq^{\mathcal{L}}_{n} p'+q} \quad (4)\ \frac{p \sqsubseteq^{\mathcal{L}}_{n} q \quad q \sqsubseteq^{\mathcal{L}}_{n'} r}{p \sqsubseteq^{\mathcal{L}}_{n+n'} r}$$

For instance, for the bisimulation semantics, we have $\sim$ in the place of $\subseteq_{\mathcal{L}}$ and we simply write $=_d$ for the obtained collection of distance relations, that in this case are all symmetric.

The rest of the paper is devoted to the development of a pure algebraic framework that allows us to define those distances in an algebraic way. Once we have it, we could use all the algebraic techniques and general results from the area, on the study of those distance relations.

3 From algebraic semantics to algebraic distances

We will see in this section how the basic concepts appearing in the definition of algebraic semantics can be adequately modified in a natural way to obtain an algebraic theory of distances between processes. It can be used to get algebraic characterizations of all the distances previously presented in [8].

We start with the notion of Σ-algebra with distances, whose definition will be a resetting of the definition of quotient Σ-algebra.

Definition 5. *Let $\mathcal{D}$ be an adequate domain for distance values (e.g. $\mathbb{N}$, $\mathbb{R}^+$, $\mathbb{Q}^+$) and Σ a (classic) signature. A $(\mathcal{D}, \Sigma)$-algebra is a Σ-algebra $\langle \mathcal{A}, \Sigma_{\mathcal{A}} \rangle$ and a collection of relations $\langle \equiv_d, d \in \mathcal{D} \rangle$, $\equiv_d \subseteq \mathcal{D} \times \mathcal{D}$, such that:*

1. *$a \equiv_0 a$, for all $a \in \mathcal{A}$.*

2. $a \equiv_d b \Leftrightarrow b \equiv_d a$, *for all* $a, b \in \mathcal{A}$ *and all* $d \in \mathcal{D}$.
3. $d \leq d'$, $a \equiv_d b \Rightarrow a \equiv_{d'} b$, *for all* $a, b \in \mathcal{A}$ *and all* $d, d' \in \mathcal{D}$.
4. $(a \equiv_d b \wedge b \equiv_{d'} c) \Rightarrow a \equiv_{d+d'} c$, *for all* $a, b, c \in \mathcal{A}$ *and all* $d, d' \in \mathcal{D}$.
5. $f \in \Sigma$, $ar(f) = k$, $a_i \equiv_{d_i} b_i$ *for all* $i \in 1..k \Rightarrow f(\overline{a}) \equiv_{\widetilde{f}(d_1,\ldots,d_k)} f(\overline{b})$, *where* $\widetilde{f}$ *is a function associated to* f *that combines the distances between the components of its arguments and satisfies* $\widetilde{f}\,(\overline{0}) = 0$.

Remark 1. In the previous definition we have only directly stated reflexivity of $\equiv_0$ in (1). However, applying these rules we can infer

$$(1')\ a \equiv_d a \text{ for all } a \in \mathcal{A} \text{ and all } d \in \mathcal{D},$$

by combining (1) and (3). Another possibility is to take (1′) instead of (1), and then by combining it with (4), we get the monotonicity rule (3), which therefore could be removed.

Let us discuss and justify one by one all the ingredients of Def. 5. We have introduced a collection of relations $\equiv_d$, that intuitively describe the balls of radius d of the topology induced by our distance notion. The classical properties of distances correspond to 1-4. Note how the transitivity of equivalence relations is substituted by the triangular inequality 4.

We have said that we are generalizing the notion of quotient algebra, and not just that of (plain) algebra, because we allow that $\equiv_0$ will be any Σ-congruence, and not just the equality relation. Note that in that particular case the triangular inequality becomes transitivity, because $0 + 0 = 0$, and then $\equiv_0$ has to be first an equivalence relation, but also a Σ-congruence, by applying 5.

We have preferred to be quite general w.r.t. the constraints imposed to the combination functions $\widetilde{f}$. Certainly $\widetilde{f} = +$ for all $f \in \Sigma$, will be the most interesting case. In Section 6 we will see how taking $\widetilde{f} = max$, we obtain the algebraic characterization of other noticeable distances.

Now we can get our deduction systems for distances, $\mathbf{dDED}(E_{\mathcal{D}})$, by resetting the clauses that define $\mathbf{DED}(E)$, again in a very natural way.

Definition 6. *A deduction system for distances,* ***dDED****$(E_{\mathcal{D}})$, between terms in* $\mathcal{T}_{\Sigma}(X)$*, is a collection of rules including:*

1. *Reflexivity:* $t \equiv_d t$.
2. *Symmetry:* $t \equiv_d t' \Rightarrow t' \equiv_d t$.
3. *Triangular transitivity:* $t \equiv_d t'$, $t' \equiv_{d'} t'' \Rightarrow t \equiv_{d+d'} t''$.
4. *Substitution:* $t_1 \equiv_{d_1} t'_1, \ldots, t_k \equiv_{d_k} t'_k \Rightarrow f(t_1, \ldots, t_k) \equiv_{\widetilde{f}(d_1,\ldots,d_k)} f(t'_1, \ldots, t'_k)$ *for every* $f \in \Sigma$ *of arity* k *and the corresponding* $\widetilde{f}$ *composing distances.*
5. *Instantiation:* $t \equiv_d t' \Rightarrow t\rho \equiv_d t'\rho$, *for every substitution* ρ.
6. *A set of distance equations* $E_{\mathcal{D}} = \{t_i \equiv_{d_i} t'_i \mid i \in I\}$.

Since variables are useful to get compact (possibly finite) axiomatizations, they are typically used when defining deduction systems. We have followed the same idea when defining $\mathbf{dDED}(E_{\mathcal{D}})$, that gives us distance pairs not only between closed processes, but also between open processes. Of course, we expect that any such derivable pair will reflect a universal information, which is formalized by the instantiation rule.

The roles of reflexivity, symmetry and that of the triangular transitivity, were already commented when defining the algebras with distances. Moreover, the substitution rule states the homomorphic character of the obtained distances. Of course, we have a different deduction system for each collection of composing functions $\langle \widetilde{f} \mid f \in \Sigma \rangle$, however, we prefer to maintain this small abuse of notation.

Finally, we have again adopted quite a general point of view when defining $\mathbf{dDED}(E_{\mathcal{D}})$ based on an arbitrary set of distance equations $E_{\mathcal{D}}$. But, once more it is interesting to discuss which are the most natural sets of equations, in which we are specially interested. The role of $\mathbf{DED}(E)$ is to generate the induced set of derived equations from the set E. When we start from the axiomatization of any semantics (e.g. the bisimulation semantics), the related closed processes are exactly those having the same semantics (e.g. those bisimilar).

As explained above, $\equiv_0$ just reflects the quotient algebra on top of which we will define our distance relations. As a matter of fact the following result tells us that, when E does only contain equations on $\equiv_0$, we just obtain a system totally equivalent to an ordinary deduction system.

Proposition 1. *If $E_{\mathcal{D}}$ is a system that only contains equations on $\equiv_0$, then the system **dDED**$(E_{\mathcal{D}})$ is "essentially" equivalent to **DED**(E), where $E = \{t \equiv t' \mid t \equiv_0 t' \in E_{\mathcal{D}}\}$. This means that $\vdash_{E_{\mathcal{D}}} t \equiv_d t' \Leftrightarrow \vdash_E t \equiv t', \ \forall d \in \mathcal{D}$.*

Proof. This is an immediate consequence of the following facts: (1) The triangular transitivity becomes plain transitivity when $d_1 + d_2 = 0$; (2) The impossibly to infer facts about $\equiv_0$ using other ones $\equiv_d$, with $d \neq 0$.

□

Therefore, in order to have a useful deduction system for distances we need to start with a collection of equations $E_{\mathcal{D}}$ containing a set of non-trivial non-zero distance axioms $t_0 \equiv_d t'_0$ with $d > 0$. This subset is the "algebraic basis" on top of which $E_{\mathcal{D}}$ will derive the induced distance pairs in $\vdash_{E_{\mathcal{D}}} t \equiv_d t'$.

Proposition 2. *Given a system of distance equations $E_{\mathcal{D}}$, if we define $E^0_{\mathcal{D}} = \{t \equiv_d t' \mid d = 0\}$ and we consider the set of ordinary equations $E = \{t \equiv t' \mid t \equiv_0 t' \in E^0_{\mathcal{D}}\}$, then we can see the family of distance relations induced by $\vdash_{E_{\mathcal{D}}}$ as a family of distance relations between the equivalence classes induced by $\vdash_E$,*

$$[t] \equiv_d [t'] ::= \vdash_{E_{\mathcal{D}}} t \equiv_d t'$$

Proof. We only need to apply the triangular transitivity rule.

□

In the following section we apply this algebraic approach, defining a distance for the bisimulation semantics.

4 An algebraic distance for bisimulation

As stated above, to define our processes, we use the signature including the choice operation with arity 2, and the prefix operators $a\cdot \in Act$ with arity 1, together with the constant null, $\mathbf{0}$. We expand this signature including variables in a set X to obtain the $BCCSP(Act, X)$ syntactic algebra. Then, the corresponding compositional approach to the definition of distances between processes in $BCCSP(Act, X)$ includes the rules:

- If we have $p \equiv_d p'$ and $q \equiv_d q'$, then $p + q \equiv_d p' + q'$.
- If we have $p \equiv_d q$, then $ap \equiv_d aq$.

Moreover, the equations characterizing the bisimulation axioms are turned into distance equations getting:

$$(B1)\ x + y \equiv_0 y + x \qquad (B2)\ x + x \equiv_0 x$$

$$(B3)\ (x + y) + z \equiv_0 x + (y + z) \qquad (B4)\ z + \mathbf{0} \equiv_0 z$$

Finally, we need to add the collection of equations that will work as seed for the computation of distances in an algebraic way. The idea is that we want to pay a tax for each punctual change. Since we are working under a function $\overline{d}$ defining the distance between actions in Act, we introduce the family of axioms with $a, b \in Act$, which can be considered as a single axiom scheme if we see a and b as generic action: $ax \equiv_{d(a,b)} bx$. Putting everything together we obtain the following algebraic definition of the bisimulation distance.

Definition 7. *Given a function distance $\overline{d}$ between the actions in Act, producing values in $\mathcal{D}$, and two processes $p, q \in BCCSP(Act, X)$, we can say that p is at most $d \geq 0$ far away of being bisimilar to q, if and only if $p \equiv_d q$ can be derived using the set of rules:*

1. *$p \equiv_d p$ for all $d \in \mathcal{D}$ and for all $p \in Proc$.*
2. *$p \equiv_d p' \Rightarrow p' \equiv_d p$ for all $d \in \mathcal{D}$ and for all $p, p' \in Proc$.*
3. *$p \equiv_{d_1} p'$ and $p' \equiv_{d_2} p'' \Rightarrow p \equiv_{d_1+d_2} p''$ for all $d_1, d_2 \in \mathcal{D}$ and $p, p', p'' \in Proc$.*
4. *(i) $p \equiv_{d_1} p'$, $q \equiv_{d_2} q' \Rightarrow p + q \equiv_{d_1+d_2} p' + q'$.*
   (ii) $p \equiv_d q \Rightarrow ap \equiv_d aq$ for all $a \in Act$, $d, d_1, d_2 \in \mathcal{D}$ and $p, p', q, q' \in Proc$.
5. *$p \equiv_d p' \Rightarrow p\rho \equiv_d p'\rho$, for every substitution ρ.*
6. *(i) $ax \equiv_{\overline{d}(a,b)} bx$ for all $a, b \in Act$.* *(ii) $x + y \equiv_0 y + x$.*
   (iii) $x + x \equiv_0 x$. *(iv) $(x + y) + z \equiv_0 x + (y + z)$.* *(v) $z + \mathbf{0} \equiv_0 z$.*

Remark 2. 1. The definition above only considers finite terms in $BCCSP(Act, X)$, but we can extend the application of these rules to infinite processes. However, this extension will only produce distances for the case of pairs of processes that are bisimilar up to the change of finitely many actions (e.g., $a^\omega \equiv_{d(a,b)} ba^\omega$). In Section 7 we will discuss how we can get other more interesting distances in the infinite case.

2. Once we use addition as the composition function of distance for the choice operator, we could substitute rule $4(i)$ in Def. 7 by the simpler rule $p \equiv_d p' \Rightarrow p + q \equiv_d p' + q$. We immediately obtain the original rule by combining this with the triangular transitivity rule 3.
   By combining that simplified rule with rule $4(ii)$ is easy to prove that for any linear context $\mathcal{C}(x)$ we have the preservation rule

$$p \equiv_d p' \;\Rightarrow\; \mathcal{C}(p) \equiv_d \mathcal{C}(p').$$

   This will not be true for any arbitrary non-linear $\mathcal{C}$, where in principle if x appears k times in $\mathcal{C}$, then we will have $\mathcal{C}(p) \equiv_{k*d} \mathcal{C}(p')$ but not $\mathcal{C}(p) \equiv_d \mathcal{C}(p')$. Obviously, this is an important difference to what happens in $\mathbf{DED}(E)$. There we can modify the global substitution rule 4 in Fig. 1 by a local substitution, where only an argument of f is substituted. It is possibly to do it without obtaining nothing new due to the presence of the transitivity rule.

Example 1. For instance, let us take $Act = \{a, b, c\}$ and define $\overline{d}(a, b) = 1$, $\overline{d}(a, c) = 2$ and $\overline{d}(b, c) = 1$. Now we will show that $ab\mathbf{0} + bb\mathbf{0} \equiv_3 ac\mathbf{0} + cc\mathbf{0}$:

$$b\mathbf{0} \equiv_1 c\mathbf{0} \;\; (Def.7.5,\; 7.6) \;\Rightarrow \left\{ \begin{array}{l} ab\mathbf{0} \equiv_1 ac\mathbf{0} \;\; (Def.7.4) \\ \qquad \wedge \\ bb\mathbf{0} \equiv_1 bc\mathbf{0} \;\; (Def.7.4) \\ \qquad \wedge \\ bc\mathbf{0} \equiv_1 cc\mathbf{0} \;\; (by\; Def.7.5,\; 7.6) \end{array} \right\} \stackrel{(Def.7.3)}{\Rightarrow} bb\mathbf{0} \equiv_2 cc\mathbf{0}$$

and finally applying Def. 7.4 we can conclude $ab\mathbf{0} + bb\mathbf{0} \equiv_3 ac\mathbf{0} + cc\mathbf{0}$.

We developed in [8] our operational and our denotational approaches to the definition of distance relations without considering variables. However we can easily extend both of them to cover the full set $BCCSP(Act, X)$. In particular, for the denotational approach that we are using here, we can extend Def. 4 by first extending the preorder $\subseteq_{\mathcal{L}}$ to open terms in the classic way, and simply applying the rest of the rules also to these open terms.

Now it is immediate to check that if we can derive $p(\overline{x}) =_d q(\overline{x})$, using this extension of Def. 4, where the variables in $\overline{x}$ are those appearing in either p or q, then we can also derive any instance $p(\overline{r}) =_d q(\overline{r})$. Here we have used the classical notation $p(\overline{r})$, to denote the application of the instantiation of each variable x_i in $p(\overline{x})$ by the corresponding term, r_i.

Lemma 1. *If we can derive $p(\overline{x}) =_d q(\overline{x})$, then we can also derive any instance of it, $p(\overline{r}) =_d q(\overline{r})$.*

Proof. By induction on the derivations of $p(\overline{x}) =_d q(\overline{x})$

1. $p(\overline{x}) \sim q(\overline{x}) \Rightarrow p(\overline{r}) \sim q(\overline{r})$, by definition of bisimulation.
2. and 3. The application of (2) and (3) in Def. 4, is immediate because instantiation satisfies the homomorphic rules:

$$(ap(\overline{r})) = a(p(\overline{r})) \text{ and } (p+q)(\overline{r}) = p(\overline{r}) + q(\overline{r}).$$

4. Finally, by a direct application of (4) in Def. 4, from $p(\overline{r}) =_n s(\overline{r})$ and $s(\overline{r}) =_{n'} q(\overline{r})$ it produces $p(\overline{r}) =_{n+n'} q(\overline{r})$.

□

Next we prove the equivalence between this algebraic definition of the distance relations and the extension of the denotational definition in Def. 4.

Theorem 1. *For all p,q* $\in BCCSP(Act, X)$ *we have* $p \equiv_d q \Leftrightarrow p =_d q$

Proof. $\underline{\Leftarrow |}$ We want to show that if $p =_d q$ then $p \equiv_d q$. We use induction over the depth of derivations.

1. $\frac{p \sim q}{p =_d q}$. If $p \sim q$, then we can prove it using the set of axioms for bisimulation that are mimicked by the last four axioms in Def. 7.6. So that we have $p \equiv_0 q$, and then by Def. 7.1 we have $q \equiv_d q$, and finally applying Def. 7.3 we get $p \equiv_d q$ as we wanted to show.
2. $\frac{p =_d q}{ap =_{d+\overline{d}(b,a)} bq}$. By the i.h. we have $p \equiv_d q$. We use the equation $ax \equiv_{\overline{d}(a,b)} bx$, Def. 7.1 and 7.5 to get $ap \equiv_{\overline{d}(a,b)} bp$. Moreover, applying Def. 7.4 (ii) and the i.h. we get $bp \equiv_d bq$; and finally by the triangular transitivity rule (7.3) we can conclude $ap \equiv_{d+\overline{d}(a,b)} bq$.
3. $\frac{p =_d p'}{p+q =_d p'+q}$. We have, by the i.h., that $p \equiv_d p'$. Trivially $q \equiv_0 q$ using Def. 7.1. Then, we can conclude applying 7.4 (i) that $p + q \equiv_d p' + q$.
4. $\frac{p =_d r \quad r =_{d'} q}{p =_{d+d'} q}$. Immediate, using the i.h. and Def. 7.3.

$\underline{\Rightarrow |}$ By induction on the derivation of $p \equiv_d q$.

1. $p \equiv_d p$. Obviously, we have $p \sim p$ and then by Def. 4.1 we get $p =_d p$.
2. $p \equiv_d p' \Rightarrow p' \equiv_d p$. By the i.h. we get $p =_d p'$, and it is immediate to check, by induction on the derivation of $p =_d p'$ that we can generate a symmetric derivation concluding $p' =_d p$. We use that $\sim$ is an equivalence relation in Def. 4.1, and $\overline{d}$ is a symmetric relation in Def. 4.2.
3. $p \equiv_{d_1} p'$ and $p' \equiv_{d_2} p'' \Rightarrow p \equiv_{d_1+d_2} p''$. By the i.h. we get $p =_{d_1} p'$, and $p' =_{d_2} p''$, and applying Def. 4.4 we conclude $p =_{d_1+d_2} p''$.
4. (i)$p \equiv_{d_1} p'$, $q \equiv_{d_2} q' \Rightarrow p + q \equiv_{d_1+d_2} p' + q'$. By the i.h. we get $p =_{d_1} p'$ and $q =_{d_2} q'$ using Def. 4.3 and 4.4 we get $p + q =_{d_1+d_2} p' + q'$.
   (ii)$p \equiv_d q \Rightarrow ap \equiv_d aq$. Immediate, applying the i.h. and Def. 4.2.
5. $p \equiv_d p' \Rightarrow p\rho \equiv_d p'\rho$, for every substitution ρ. Trivially, applying Lemma 1.
6. (i) $ax \equiv_{\overline{d}(a,b)} bx$, applying 4.2 and 4.1.
   (ii)-(iv). Immediate, applying 4.1.

□

Remark 3. Certainly, we could also remove variables from the algebraic presentation simply expanding every axiom, including instead of it, all its closed instances. Obviously the instance rules will not be necessary after that. But, of course the role of variables in any algebraic presentation is crucial in order to obtain finite axiomatizations where we reflect by a single action an infinity of facts.

5 Algebraic distances for other semantics

Once we have studied in detail the algebraic definition of the distance for the case of bisimulation, we will briefly discuss the case of the rest of the semantics which are collected in the ltbt-spectrum [10]. These semantics are not induced by an equivalence relation but by a preorder. Hennessy also presented in [7] a theory of Σ-po algebras, $\langle A, \leq_A, \Sigma_A \rangle$, which are endowed with a partial order. Now for each f in Σ of arity k, there is a monotonic function f_A: $A^k \to A$. You can find in [7] all the details about this theory of ordered algebras which smoothly extends that of plain algebras. In particular, Hennessy purposes the proof system $\mathbf{DED}(I)$, in Fig. 2, where I is now a set of inequations.

1. Reflexivity $\frac{}{t \leq t}$
2. Transitivity $\frac{t \leq t' \quad t' \leq t''}{t \leq t''}$
3. Substitution $\frac{t_1 \leq t'_1, \ldots, t_k \leq t'_k}{f(t_1, \ldots, t_k) \leq f(t'_1, \ldots, t'_k)}$, for every $f \in \Sigma$ of arity k.
4. Instantiation $\frac{t \leq t'}{t\rho \leq t'\rho}$, for every substitution ρ.
5. Equations $\frac{}{t \leq t'}$, for every inequation $t \leq t' \in I$

Fig. 2. The proof system $\mathbf{DED}(I)$

As in Section 4 we can adapt this theory of ordered algebras, obtaining an algebraic theory which allows us to measure the distance between processes w.r.t. a given semantics $\mathcal{L}$ (in)equationally defined by axioms on an order $\subseteq^{\mathcal{L}}$. So, we define the following ordered deduction system with distances $\mathbf{dDED}(I)$.

Definition 8. *Given a semantics $\mathcal{L}$ algebraically defined by means of an axiomatization I on $\leq^{\mathcal{L}}$, and a distance $\overline{d}$ over the set of actions Act, we will say that a process p is at most $d \geq 0$ far away of being better than another process q, w.r.t. the preorder $\leq^{\mathcal{L}}$, if and only if we have $\vdash_{\boldsymbol{dDED}(I)} p \leq^{\mathcal{L}}_d q$. **dDED**(I) is the following deduction system:*

1. *$p \leq^{\mathcal{L}}_d p$, for all $d \in \mathcal{D}$.*
2. *$p \leq^{\mathcal{L}}_{d_1} p'$ and $p' \leq^{\mathcal{L}}_{d_2} p'' \Rightarrow p \leq^{\mathcal{L}}_{d_1+d_2} p''$.*
3. *(i) $p \leq^{\mathcal{L}}_{d_1} p'$ $q \leq^{\mathcal{L}}_{d_2} q' \Rightarrow p + q \leq^{\mathcal{L}}_{d_1+d_2} p' + q'$.*
   (ii) $p \leq^{\mathcal{L}}_d q \Rightarrow ap \leq^{\mathcal{L}}_d aq$, for all $a \in Act$.
4. *$p \leq^{\mathcal{L}}_d p'$ then $p\rho \leq^{\mathcal{L}}_d p'\rho$, for every substitution ρ.*
5. *$ax \leq^{\mathcal{L}}_{\overline{d}(a,b)} bx$.*
   $t \leq^{\mathcal{L}}_d t'$, for every inequation $t \leq^{\mathcal{L}} t' \in I$.

Remark 4. 1. By the way, the only difference between the proof systems $\mathbf{DED}(E)$ and $\mathbf{DED}(I)$ is that in this second case we are defining a preorder and not

an equivalence relation, therefore symmetry is lost. The same happens in **dDED**(I) by means of which we measure "how far away" a process is to be better than another process w.r.t. the corresponding order $\subseteq^{\mathcal{L}}$.

2. Once we are defining an order and not an equivalence it is natural to consider asymmetric distances d, where $d(b, a)$ denotes what we have to add to b to obtain (at least) a. For instance if $a = 1€$ and $b = 2€$ we could have $d(a, b) = 1$ but $d(b, a) = 0$. We have developed our operational distances in [8] including this generalization, and it can be also introduced here simply changing the symmetric distance $\overline{d}$ by an asymmetric distance d.

It is easy to translate the results in Section 4 to this more general framework. In particular, we can prove that each denotational distance, obtained by application of Def.4, is equivalent to the corresponding algebraic distance, obtained by application of Def. 8.

6 Characterizing the bisimulation distance game

The classic approaches to the definition of distances between processes are based on valued versions of the (bi)simulation game [9, 6, 2].

Definition 9. *(Bisimulation game) Given two LTSs, L_1 and L_2, we call configurations the pairs (p, q), with $p \in L_1$ and $q \in L_2$. The bisimulation game is played by two players: the attacker $\mathbb{A}$ and the defender $\mathbb{D}$. The initial configuration of the game deciding if $p_0 \sim q_0$, is just the pair (p_0, q_0). A round of the game, when the current configuration is (p, q), proceeds as follows:*

1. *$\mathbb{A}$ chooses either p or q.*
2. *Assuming it chooses p, he next executes a transition in L_1: $p \xrightarrow{a} p'$.*
3. *$\mathbb{D}$ must choose a transition with the same label at the other side of the board, i.e., it must execute an a-move in L_2: $q \xrightarrow{a} q'$. If $\mathbb{A}$ plays at L_2, then $\mathbb{D}$ replies at L_1.*
4. *The game proceeds in the same way from the new configuration (p', q').*

The bisimulation game can be turned into the "classical" bisimulation distance game [1], by allowing to reply any a-move by some b-move with $b \neq a$. However, in this case the defender should pay $\overline{d}(b, a)$ to the attacker for the mismatch. The value of the game, V(p,q), provides the "classical" bisimulation distance between p and q, $d_{\sim}(p, q)$.

In order to illustrate the generality of our algebraic approach to the definition of distances between processes, next we present an algebraic characterization of that bisimulation game distance.

Definition 10. *We define the algebraic bisimulation game collection of relations, $\{\equiv^g_d, d \in \mathcal{D}\}$, as the set of tuples $p \equiv^g_d q$ which can be derived by applying the following set of rules:*

1. $p \equiv^g_d p$ *for all* $d \in \mathcal{D}$ *and* $p \in Proc.$
2. $p \equiv^g_d p' \Rightarrow p' \equiv^g_d p$ *for all* $d \in \mathcal{D}$ *and* $p, p' \in Proc.$
3. $p \equiv^g_{d_1} p'$ *and* $p' \equiv^g_{d_2} p'' \Rightarrow p \equiv^g_{d_1+d_2} p''$ *for all* $d_1, d_2 \in \mathcal{D}$ *and* $p, p', p'' \in Proc.$
4. *(i)* $p \equiv^g_{d_1} p'$, $q \equiv^g_{d_2} q' \Rightarrow p + q \equiv^g_{max\{d_1,d_2\}} p' + q'$.
   (ii) $p \equiv^g_d q \Rightarrow ap \equiv^g_d aq$ *for all* $a \in Act$, $d, d_1, d_2 \in \mathcal{D}$ *and* $p, p', q, q' \in Proc.$
5. $p \equiv^g_d p' \Rightarrow p\rho \equiv^g_d p'\rho$, *for every substitution* ρ.
6. *(i)* $ax \equiv^g_{\overline{d}(a,b)} bx$ *for all* $a, b \in Act.$ *(ii)* $x + y \equiv^g_0 y + x.$
   (iii) $x + x \equiv^g_0 x.$ *(iv)* $(x + y) + z \equiv^g_0 x + (y + z).$ *(v)* $z + \mathbf{0} \equiv^g_0 z.$

Remark 5. Note that the definition of the algebraic bisimulation distance by means of the collection $\{\equiv_d, d \in \mathcal{D}\}$ in Def. 7, and that of the algebraic bisimulation game distance by means of $\{\equiv^g_d, d \in \mathcal{D}\}$, are nearly the same. We only modify the composition rule 4 (i), where the composition function $\widetilde{f}$–see Def. 5.5– was initially + and now it becomes max.

As we did for our bisimulation distance, there is still a third equivalent denotational characterization of the bisimulation game distance.

Definition 11. *We define the denotational bisimulation game collection of relations,* $\{=^g_d, d \in \mathcal{D}\}$, *as the set of tuples* $p =^g_d q$ *which can be derived by applying the following set of rules:*

$$(1)\ \frac{p \sim q}{p =^g_n q} \qquad (2)\ \frac{p =^g_d q}{ap =^g_{d+\overline{d}(b,a)} bq} \qquad (3)\ \frac{p =^g_{d_1} p'\ q =^g_{d_2} q'}{p + q =^g_{max\{d_1,d_2\}} p' + q'}$$

We have proved in [8] that the relations defined by Def. 11 remain the same if we add the transitivity rule

$$(4)\ \frac{p =^g_{d_1} q\ q =^g_{d_2} r}{p =^g_{d_1+d_2} r}$$

so further in this paper we consider Def. 11 including this rule. Then, it is easy to get the following theorem based on the proof of Th. 1

Theorem 2. $p \equiv^g_d q \Leftrightarrow p =^g_d q$.

Proof. Immediate, just substituting + by max in the reasoning related to the application of Def. 4.3 and Def. 7.4.

□

Remark 6. Although rule (3) in Def. 4 has not +, we can see applying transitivity that this rule is equivalent to

$$(3')\ \frac{p =^{\mathcal{L}}_{n_1} p' \quad q =^{\mathcal{L}}_{n_2} q'}{p + q =^{\mathcal{L}}_{n_1+n_2} p' + q'}$$

which produce a simpler proof than the given in Th. 1 when we use 4.3.

Next, we see that the original definition of the distances by means of the distance bisimulation game is also equivalent to these. We start proving two lemmas that provide us some properties of the denotational characterization and of the values of the distance bisimulation game.

Lemma 2 (Prefix lemma). *Given two processes $P = ap$ and $Q = bq$, we have $ap =^g_d bq$ if and only if there exists some d' such that $d = d' + \overline{d}(a,b)$ with $p =^g_{d'} q$.*

Proof. $\Leftarrow|$ Immediate, since if we have Def. 11.2 to $p =^g_{d'} q$ we get the result.
$\Rightarrow|$ We will prove a more general result. It says that $\sum_{i\in I} a_i p_i =^g_d \sum_{j\in J} b_j q_j$ implies $(\forall\, i \in I\, \exists\, j \in J \quad p_i =^g_{d-\overline{d}(a_i,b_j)} q_j)$. So, the result of this theorem will be just the particular case of this result when we have only one element in the sum. We use induction over the derivation of $p =^g_d q$ taking $p = \sum_{i\in I} a_i p_i$ and $q = \sum_{j\in J} b_j q_j$.

1. $\dfrac{p \sim q}{p =^g_d q}$. If $p \sim q$ then we have for each $i \in I$ such that $a_i p_i \xrightarrow{a_i} p_i$ there exists some $j \in J$ with $b_j q_j \xrightarrow{b_j} q_j$ where $b_j = a_i$ and $p_i \sim q_j$. Then we have $p_i =^g_d q_j$ and using the rule (2) we can conclude that $a_i p_i =^g_0 b_j q_j$ as we wanted to show, since $\overline{d}(a_i, b_j) = 0$.

2. $\dfrac{p =^g_{d-\overline{d}(b,a)} q}{ap =^g_{d-\overline{d}(b,a)+\overline{d}(b,a)} bq}$. Then we have a single summand at both sides, and the premise of the last set of the derivations exactly expresses the thesis to be proved.

3. If we have $p = p' + p''$, $q = q' + q''$ and $\dfrac{p' =^g_{d'} q' \quad p'' =^g_{d''} q''}{p =^g_{d=max\{d',d''\}} q}$. We will have $p' = \sum_{i\in I'} a_i p_i$, $p'' = \sum_{i\in I''} a_i p_i$, $q' = \sum_{j\in J'} b_j q_j$, and $q'' = \sum_{j\in J''} b_j q_j$ with $I = I' \cup I''$ and $J = J' \cup J''$. Then, by applying the i.h. we have $\forall i \in I'$ $\exists j \in J'$ with $p_i =^g_{d'-\overline{d}(a_i,b_j)} q_j$ and $\forall i \in I''\, \exists j \in J''$ with $p_i =^g_{d''-\overline{d}(a_i,b_j)} q_j$. As $d = max\{d', d''\}$ we immediately conclude the result applying the triangular transitivity(rule (3)).

4. $\dfrac{p =^g_d r \quad r =^g_{d'} q}{p =^g_{d+d'} q}$. We take $r = \sum_{k\in K} c_k r_k$ and $r_k =^g_{d'-\overline{d}(c_k,b_j)} q_j$, so applying the i.h. we get $\forall i \in I\ \exists k \in K\ p_i =^g_{d-\overline{d}(a_i,c_k)} r_k$ (and $\forall k \in K\ \exists j \in J$ $r_k =^g_{d'-\overline{d}(c_k,b_j)} q_j$). Then, applying the triangular inequality ($\overline{d}(a_i,b_j) \leq \overline{d}(a_i,c_k) + \overline{d}(c_k,b_j)$) we get that $\forall i \in I\ \exists j \in J$ such that $p_i =^g_{d+d'-\overline{d}(a_i,b_j)} q_j$. □

Corollary 1. *$d(\alpha,\beta) = \sum \overline{d}(a_i,b_i)$ where $\alpha = a_1 \ldots a_n$ and $\beta = b_1 \ldots b_n$. This means that we have $\alpha =^g_d \beta$ with $d = \sum_{i=1}^n \overline{d}(a_i,b_i)$, and for all $d' < d$ we have $\alpha \neq^g_{d'} \beta$.*

Proof. That $\alpha =^g_d \beta$ is immediate by iterated application of the triangular transitivity rule. We prove the second negative result by induction on n.

$\underline{n = 0|}$ We have $d = 0$ and then the result is trivial.

$\underline{n > 0|}$ Applying the Prefix Lemma –Lemma 2– we should have $\alpha' =^{g}_{d'-\overline{d}(a_1,b_1)} \beta'$ with $\alpha' = a_2 \ldots a_n$ and $\beta' = b_2 \ldots b_n$. But, if $d' < \sum_{i=1}^{n} \overline{d}(a_i, b_i)$ then we have $d' - \overline{d}(a_1, b_1) < \sum_{i=2}^{n} \overline{d}(a_i, b_i)$ contradicting the i.h. for the shorter sequences α' and β'

□

Lemma 3. *$V(p, q) \leq V(p, r) + V(r, q)$ for all processes p,q and r.*

Proof. We use induction over the depth of processes.

For all $i \in I$ there exists some $k \in K$ with $V(p_i, r_k) = V(p, r) - \overline{d}(a_i, c_k)$. For all $k \in K$ there exists some $j \in J$ with $V(r_k, q_j) = V(r, q) - \overline{d}(c_k, b_j)$. Combining both, we obtain:

$$\begin{array}{c}\text{For all } i \in I \text{ there exists some } j \in J \text{ and } k \in K \text{ with} \\ V(p_i, r_k) + V(r_k, q_j) = V(p, r) + V(r, q) - \overline{d}(a_i, c_k) - \overline{d}(c_k, b_j).\end{array}$$

and then by applying the i.h. and the triangular inequality for d, we obtain:

$$\forall i \in I \;\; \exists j \in J \text{ with } V(p_i, q_j) \leq V(p, r) + V(r, q) - \overline{d}(a_i, b_j).$$

In a symmetric way, we prove that For all $j \in J$ there exists some $i \in I$ with $V(p_i, q_j) \leq V(p, r) + V(r, q) - \overline{d}(a_i, b_j)$. Applying the definition of the value of the distance game for bisimulation we conclude: $V(p, q) \leq V(p, r) + V(r, q)$.

□

The value of the bisimulation game between two processes p and q, $V(p, q)$, gives us "the" distance between them $d_{\sim}(p, q)$. Next we will see that our denotational definition gives us all the bounds of this distance.

Theorem 3. *$d_{\sim}(p, q) = \min_d\{p =^{g}_{d} q\}$; i.e. $p =^{g}_{d'} q$ iff $d_{\sim}(p, q) \leq d'$.*

Proof. In order to simplify our notation we denote simply by $d(p, q)$ the distance between these two processes. We use induction on the depth of p and q.

$\underline{\supseteq |}$ We have that $p =^{g}_{v} q$ and we want to check that $d(p, q) \leq v$. We prove it by induction on the derivation of $p =^{g}_{v} q$.

1. $\dfrac{p \sim q}{p =^{g}_{v} q}$. If $p \sim q$ then the value of the bisimulation game is 0, because all along a game we can reply an a-move of p by an a-move of q, and $\overline{d}(a, a) = 0$, and conversely.
2. $\dfrac{p =^{g}_{v} q}{ap =^{g}_{v+\overline{d}(b,a)} bq}$. By applying the induction hypothesis we have $d(p, q) \leq v$. Then, the definition of the bisimulation game, produces $d(ap, bq) = d(p, q) + \overline{d}(a, b) \leq v + \overline{d}(b, a)$ as we wanted to see.
3. $\dfrac{p =^{g}_{v} q \quad p' =^{g}_{v'} q'}{p + p' =^{g}_{max\{v,v'\}} q + q'}$. By applying the induction hypothesis we have $d(p, q) \leq v$ and $d(p', q') \leq v'$. Now, any a-move on the p-side (resp. p'-side) of $p + p'$ can be replied by some b-move on the q side (resp. q'-side) of

$q+q'$ guaranteeing a payment less or equal than v (resp. v'), and conversely. Thus concluding that $d(p+p', q+q') \leq max\{v, v'\}$.

4. $\dfrac{p =^g_v r \quad r =^g_{v'} q}{p =^g_{v+v'} q}$. By applying the induction hypothesis we have $d(p,r) \leq v$ and $d(r,q) \leq v'$. Now, applying Lemma 3 we conclude $d(p,q) \leq v + v'$.

$\underline{\subseteq}|$ The proof is by induction on the depth of the processes. If it is 0 the result is trivial. Therefore, let us consider decomposition of the two involved processes $p = \sum a_i p_i$ and $q = \sum b_j q_j$.

Applying the definition of the bisimulation game, from $d(p,q) = v$ we obtain that for all $i \in I$ there exists some $j_i \in J$ such that $d(p_i, q_{j_i}) + \overline{d}(a_i, b_{j_i}) \leq d(p,q)$. Reciprocally, for all $j \in J$ there exists some $i_j \in I$ such that $d(p_{i_j}, q_j) + \overline{d}(a_{i_j}, b_j) \leq d(p,q)$. Now, by applying the induction hypothesis we have

$$\forall i \in I\ p_i =_{d(p_i, q_{j_i})}\ q_{j_i} \text{ and } \forall j \in J\ p_{i_j} =_{d(p_{i_j}, q_j)}\ q_j.$$

From which applying Def. 11.2 we obtain $\forall i \in I\ a_i p_i =_{d(p_i, q_{i_j}) + \overline{d}(a_i, b_{j_i})}\ b_{j_i} q_{j_i}$ and $\forall j \in J\ a_{i_j} p_{i_j} =_{d(p_{i_j}, q_j) + \overline{d}(a_{i_j}, b_j)}\ b_j q_j$, that applying the monotonicity of the bounds computed by Def. 11 produces

$$\forall i \in I\ a_i p_i =_{d(p,q)}\ b_{j_i} q_{j_i} \quad \text{and} \quad \forall j \in J\ a_{i_j} p_{i_j} =_{d(p,q)}\ b_j q_j.$$

These two can be combined applying Def. 11.3 to produce $p = \sum a_i p_i =_{d(p,q)} \sum_{i \in I} b_{j_i} q_{j_i}$ and $\sum_{j \in J} a_{i_j} p_{i_j} =_{d(p,q)} \sum b_j q_j = q$.

We only need to combine these two again using Def. 11.3 and the idempotency of bisimilarity, to conclude $p =_{d(p,q)} q$, as we wanted to show.

□

7 Conclusions and future work

We have presented an algebraic framework to define distances between processes. In particular those associated to the semantics that are axiomatizable. Although a part of our definitions and properties could be applied to arbitrary processes, most of them are based on the consideration of finite image processes. It can be syntactically represented by a (finite) term of a certain signature.

Currently we are working on the extension of our results to the infinite case. Following [7] the idea is to approximate processes by their finite approximations and compare them level by level. Then, we would state that $p \equiv_d q$ if and only if we have $p_n \equiv_d q_n\ \forall n \in \mathbb{N}$. But whenever Act is finite, or the non-null values of $\overline{d}(b,a)$ are low bounded, we could only obtain a finite distance $p \equiv_d q$ for some $d \in \mathcal{D}$, when q can be obtained from p by a finite number of applications of the rules in Def. 7; that is, whenever p and q are bisimilar up to a finite number of changes of the actions occurring in them.

In order to obtain a more general distance that also produces finite values for pairs of processes which cannot be transformed one into the other by a finite number of transformations, we would need to adopt a discounting function. The

idea is that the weights of the disagreements between the two compared processes decrease with the depth they occur. This is easily formalized in our algebraic framework, simply changing our rule 4 ii) in Def. 7 by a discounted rule

$$p \equiv_d q \;\Rightarrow\; ap \equiv_{\alpha d} aq$$

where $\alpha > 1$. As a matter of fact this is another instantiation of our Def. 5. In such a case it would be immediate to check that when $\overline{d}(b, a) = 1$ and $\alpha = \frac{1}{2}$, we would have $a^\infty \equiv_2 b^\infty$.

We are also working on the definition of these distances by applying a coinductive approach that avoids the consideration of finite approximations to obtain the distances between infinite processes.

We have used the algebraic developments in [7] to base our algebraic theory on distances. We did that, not only due to the simplicity and clarity of its presentation of the theory, but also because of its detailed study of the testing semantics. We hope indeed that most, if not all, of the concepts and results on this semantics will be transferable to the distance scenario. So, we will obtain a nice theory of approximated pass of test both producing a distance for the induced semantics, and an interesting new concept to be applicable in practice.

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] X. Chen and Y. Deng. Game characterizations of process equivalences. In *APLAS 2008*, volume 5356 of *LNCS*, pages 107–121. Springer, 2008.

[4] D. de Frutos-Escrig, C. Gregorio-Rodríguez, and M. Palomino. On the unification of process semantics: equational semantics. *ENTCS*, 249:243–267, 2009.

[5] 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 fuer Informatik, 2011.

[6] U. Fahrenberg, C. R. Thrane, and K. G. Larsen. Distances for weighted transition systems: Games and properties. In *QAPL 2011*, volume 57 of *EPTCS*, pages 134–147, 2011.

[7] M. Hennessy. Algebraic theory of processes. MIT Press, 1988.

[8] 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.

[9] C. Stirling. Modal and temporal logics for processes. In *BHOW 1995*, volume 1043 of *LNCS*, pages 149–237. Springer, 1995.

[10] R. van Glabbeek. The linear time-branching time spectrum I: the semantics of concrete, sequential processes. In *Handbook of Process Algebra, chapter 1*, pages 3–99. Elsevier, 2001.

[11] M. Ying. *Topology in process calculus - approximate correctness and infinite evolution of concurrent programs*. Springer, 2001.

[12] M. Ying and M. Wirsing. Approximate bisimilarity. In *AMAST 2000*, volume 1816 of *LNCS*, pages 309–322. Springer, 2000.

8 Appendix: On the operational definition of the distances between processes

It is well known that the set of (unordered) finite trees labelled in their arcs by actions in Act: $FTree(Act)$, constitute an initial model for the theory of bisimulation. A similar result can be obtained, adding variables in X, for the set $FTree(Act, X)$ of trees which besides the constants in Act also can have variables in X labeling their leaves.
By applying Def. 7 and Prop. 2, we immediately obtain a family of distance relations on both $FTree(Act)$ and $FTree(Act, X)$. In other words these two algebras become a (D, Σ)-algebra and a $(D, \Sigma(X))$-algebra, respectively, when considering these two families of distance relations.

Next, we present in detail the corresponding operational definition, already studied in [8]. Later, we develop the proof of its equivalence with the denotational and algebraic characterizations, that were proved to be equivalent each other at Section 4.

Definition 12. *We say that an unordered tree p is at most at distance d from another tree q, w.r.t. the symmetric distance between actions $\overline{d}$, and then we write $d_{\overline{d}}(p, q) \leq d$, if and only if:*

- *(C1) $p = ap'$, $q = bp'$, and $d \geq \overline{d}(a, b)$, or*
- *(C2) $p = p' + r$, $q = q' + r$, and $d \geq d_{\overline{d}}(p', q')$, or*
- *(C3) $p = ap'$, $q = aq'$, and $d \geq d_{\overline{d}}(p', q')$, or*
- *(C4) $d \geq 0$ and q can be obtained from p by application of (B1)-(B4), or*
- *(C5) There exist r, d' and d'' s.t. $d' \geq d_{\overline{d}}(p, r)$, $d'' \geq d_{\overline{d}}(r, q)$ and $d \geq d' + d''$.*

Definition 13. *We define $p \rightsquigarrow^1_d q$ if and only if*

1. *$ap' \rightsquigarrow^1_d bp'$ and $d \geq d(a, b)$, or*
2. *$p' + r \rightsquigarrow^1_d q' + r$ and $p' \rightsquigarrow^1_d q'$, or*
3. *$ap' \rightsquigarrow^1_d aq'$ and $p' \rightsquigarrow^1_d q'$, or*
4. *$d \geq 0$ and q can be obtained from p by application of (B1)-(B4).*

Definition 14. *We define $p \rightsquigarrow_d p'$ if and only if there exist $p_1, \ldots, p_n$ such that $p \rightsquigarrow^1_{d_1} p_1 \rightsquigarrow^1_{d_2} p_2 \rightsquigarrow^1_{d_3} \cdots \rightsquigarrow^1_{d_n} p_n = p'$, where $\sum d_i = d$.*

Definition 15. *We define $p \rightsquigarrow^*_d p'$ if and only if we have $p \rightsquigarrow^1_d p'$, or there exists some p'' such that $p \rightsquigarrow^1_{d_1} p''$ and $p'' \rightsquigarrow^*_{d_2} p'$, where $d = d_1 + d_2$.*

Definition 16. *We define $p \mid\!\rightsquigarrow_d p'$ if and only if we have $p \rightsquigarrow^1_d p'$, or there exists some q such that $p \mid\!\rightsquigarrow_{d_1} q \quad q \mid\!\rightsquigarrow_{d_2} p'$, where $d_1 + d_2 = d$.*

Proposition 3. *Def. 14,15, 16 are obviously equivalent.*

Proof. Routine well known induction. □

Lemma 4 (Structural lemma). *If $p \rightsquigarrow_d q$ then (1) $ap \rightsquigarrow_d aq$ and (2) $p+r \rightsquigarrow_d q + r$.*

Proof. (1)| If $p \rightsquigarrow_d^1 q$ then trivially we have $ap \rightsquigarrow_d^1 aq$ applying Def. 13.3.
If $p \rightsquigarrow_d q$ then we use induction over the path length. Similarly, if we have $p = p_0 \rightsquigarrow_{d_1}^1 p_1 \rightsquigarrow_{d_2}^1 p_2 \rightsquigarrow_{d_3}^1 \cdots \rightsquigarrow_{d_n}^1 p_n = p'$ with $d = \sum d_i$, we will get $ap_i \rightsquigarrow_{d_{i+1}}^1 ap_{i+1}$ for all $i < n$. So that, we have

$$ap = ap_0 \rightsquigarrow_{d_1}^1 ap_1 \rightsquigarrow_{d_2}^1 ap_2 \rightsquigarrow_{d_3}^1 \cdots \rightsquigarrow_{d_n}^1 ap_n = ap' \text{ with } d = \sum d_i$$

thus proving $ap \rightsquigarrow_{d_1} ap'$.

(2)| Analogous to (1). □

Theorem 4. *The operational definition of distance relations, $\rightsquigarrow_d$, and the denotational one, defining the family $=_d$, are equivalent.*

Proof. We want to prove that $p \rightsquigarrow_d q \leftrightarrow p =_d q$.

$\Rightarrow$ | We use induction over the length of the path generating $p \rightsquigarrow_d q$. We also use induction on the derivation of $p \rightsquigarrow_d^1$ to prove this basic case.

1. $p \rightsquigarrow_d^1 p'$ with $p = ap''$ and $p' = bp''$ and $d \geq d(a,b)$. We only need to apply rule (2) in Def. 4 to get that $ap'' =_d bp''$.
2. $p' + r \rightsquigarrow_d^1 q' + r$ with $p \rightsquigarrow_d^1 p'$. The i.h. now produces that $p' =_d q'$, and then applying rule (3) in Def. 4 we get that $p' + r =_d q' + r$.
3. $ap \rightsquigarrow_d^1 ap'$ with $p \rightsquigarrow_d^1 p'$. One more time, the i.h. produces $p =_d p'$, and applying rule (2) in Def. 4 we get $ap =_d ap'$.
4. $p \rightsquigarrow_d^1 p'$ with $d \geq 0$ and p' can be obtained from p by application of (B1)-(B4). As $p \sim p'$ trivially we have using rule (1) in Def. 4 that $p =_d p'$.
5. $p \rightsquigarrow_d p'$ if and only if $\exists p''\ p \rightsquigarrow_{d_1} p''\ p'' \rightsquigarrow_{d_2} p'$ with $d = d_1 + d_2$. We can apply the i.h. obtaining $p =_{d_1} p''$ and $p'' =_{d_2} p'$, and now applying rule (4) in Def. 4 we get $p =_{d_1+d_2} p'$, i.e., $p =_d p'$, as we wanted to show.

$\Leftarrow$ | Now we proof $p =_d q \Rightarrow p \rightsquigarrow_d q$, by induction on the derivation of $p =_d q$.

1. $\dfrac{p \sim q}{p =_d q}$. Trivially if $p \sim q$ then $p \rightsquigarrow_d q$ with $d \geq 0$, applying Def. 13.4.
2. $\dfrac{p =_{d-\overline{d}(b,a)} q}{ap =_d bq}$. By applying the induction hypothesis we have $p \rightsquigarrow_{d-\overline{d}(a,b)} q$ and applying the structural lemma we obtain $bq \rightsquigarrow_{d-\overline{d}(a,b)} bq$, from where using Def. 13.1 and Def. 15 we can conclude that $ap \rightsquigarrow_d bq$.
3. $\dfrac{p =_d p'}{p+q =_d p'+q}$. Again, by applying the induction hypothesis we have $p \rightsquigarrow_d p'$, and applying the structural lemma we conclude $p + q \rightsquigarrow_d p' + q$.
4. $\dfrac{p =_d q \quad q =_{d'} r}{p =_{d+d'} r}$. By the i.h. we have $p \rightsquigarrow_d q$ and $q \rightsquigarrow_{d'} r$. So, applying Def. 15 and Prop. 3 we get $p \rightsquigarrow_{d+d'} r$.

□