Generalized Quantitative Analysis of Metric Transition Systems

Uli Fahrenberg, Axel Legay

HAL Id: hal-01087911

https://hal.inria.fr/hal-01087911

Submitted on 27 Nov 2014

Generalized Quantitative Analysis
of Metric Transition Systems

Uli Fahrenberg and Axel Legay

Irisa / INRIA Rennes, France

Abstract. The formalism of metric transition systems, as introduced by de Alfaro, Faella and Stoelinga, is convenient for modeling systems and properties with quantitative information, such as probabilities or time. For a number of applications however, one needs other distances than the point-wise (and possibly discounted) linear and branching distances introduced by de Alfaro *et.al.* for analyzing quantitative behavior.

In this paper, we show a vast generalization of the setting of de Alfaro *et.al.*, to a framework where any of a large number of other useful distances can be applied. Concrete instantiations of our framework hence give *e.g.* limit-average, discounted-sum, or maximum-lead linear and branching distances; in each instantiation, properties similar to the ones of de Alfaro *et.al.* hold.

In the end, we achieve a framework which is not only suitable for modeling different kinds of quantitative systems and properties, but also for analyzing these by using different application-determined ways of measuring quantitative behavior.

1 Introduction

During the last decade, formal verification has seen a trend towards modeling and analyzing systems which contain quantitative information. This is motivated by applications in real-time systems, hybrid systems, embedded systems and others. Quantitative information can thus be a variety of things: probabilities, time, tank pressure, energy intake, *etc.*

A number of quantitative models have hence been developed: probabilistic automata [39], stochastic process algebras [30], timed automata [2], hybrid automata [1], continuous-time Markov chains [40], *etc.* Similarly, there is a number of specification formalisms for expressing quantitative properties: timed computation tree logic [29], probabilistic computation tree logic [26], metric temporal logic [31], stochastic continuous logic [3], *etc.*

Quantitative model checking, the verification of quantitative properties for quantitative systems, has also seen rapid development: for probabilistic systems in PRISM [32] and PEPA [23], for real-time systems in UPPAAL [35] and RED [46], and for hybrid systems in HyTech [27] and SpaceEx [22], to name but a few.

Quantitative model checking has, however, a problem of *robustness*. When the answers to model checking problems are Boolean—either a system meets its
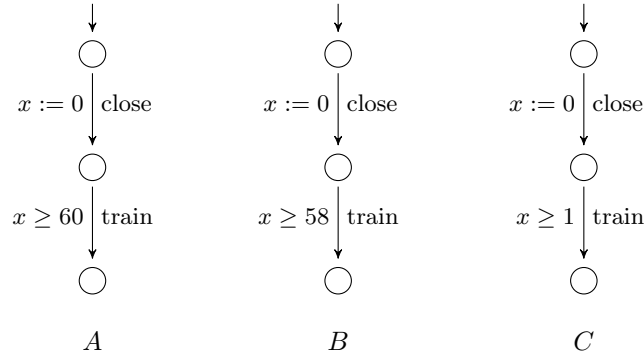
Fig. 1. Three timed automata modeling a train crossing.

specification or it does not—then small perturbations in the system's parameters may invalidate the result. This means that, from a model checking point of view, small, perhaps unimportant, deviations in quantities are indistinguishable from larger ones which may be critical.

As an example, Fig. 1 shows three simple timed-automata models of a train crossing, each modeling that once the gates are closed, some time will pass before the train arrives. Now if the specification of the system is "The gates have to be closed 60 seconds before the train arrives", then model A does satisfy the specification, and models B and C do not. What this does not tell us, however, is that model C is dangerously far away from the specification, whereas model B only violates it slightly (and may be acceptable from a practical point of view).

In order to address the robustness problem, one approach is to replace the Boolean yes-no answers of standard verification with distances. That is, the Boolean co-domain of model checking is replaced by the non-negative real numbers. In this setting, the Boolean `true` corresponds to a distance of zero and `false` to the non-zero numbers, so that quantitative model checking can now tell us not only that a specification is violated, but also *how much* it is violated, or *how far* the system is from corresponding to its specification.

In our example, and depending on precisely how one wishes to measure distances, the distance from A to our specification is 0, whereas the distances from B and C to the specification may be 2 and 59, respectively. Note that the precise interpretation of distance values will be application-dependent; but in any case, it is clear that C is much further away from the specification than B is.

The distance-based approach to quantitative verification has been developed the furthest for probabilistic and stochastic systems, perhaps akin to the fact that for these systems, the need for a truly quantitative verification is felt the most urgent. Panangaden and Desharnais *et.al.* have worked with distances for Markov processes *e.g.* in [4, 16, 34, 37], and van Breugel and Worrell *et.al.* have developed distances for probabilistic transition systems *e.g.* in [44, 45]. De Alfaro

and Stoelinga *et.al.* have worked on distances between probabilistic systems and specifications in [13, 14] and other papers.

For real-time and hybrid systems, some explicit work on distances is available in [28, 38]. Otherwise, distances have been used in approaches to robust verification [8, 33], and Girard *et.al.* have developed a theory of approximate bisimulation for robust control [25, 47].

Also general work on distances for quantitative systems where the precise meaning of the quantities remains unspecified has been done. Van Breugel has developed a general theory of behavioral pseudometrics, see *e.g.* [43], and Fahrenberg and Legay *et.al.* have introduced linear and branching distances for such systems in [5, 6, 21, 42]. Henzinger *et.al.* have employed distances in a software engineering context in [9] and for abstraction refinement and synthesis in [10,11].

A different but related approach to quantitative verification is the theory of weighted automata and quantitative languages developed by Droste *et.al.* in [17–19] and by Henzinger and Chatterjee *et.al.* in [7, 12].

Common to all the above distance-based approaches is that they introduce distances between systems, or between systems and specifications, and then employ these for approximate or quantitative verification. However, depending on the application context, a plethora of different distances are being used. Consequently, there is a need for a general theory of quantitative verification which depends as little as possible on the concrete distances being used. This is a point of view which is argued in [6, 10, 11, 21], and a number of the above papers [7, 17, 19, 37, 43] attempt to develop the theory at this general level.

To be more specific, most of the above approaches can be classified according to the way they measure distances between *executions*, or system traces. The perhaps easiest such way is the *point-wise* distance, which measures the greatest individual distance between corresponding points in the traces. Theory for this specific distance has been developed *e.g.* in [8, 13–15, 42]. Sometimes *discounting* is applied to diminish the influence of individual distances far in the future, *e.g.* in [13–15].

Another distance which has been used is the *accumulating* one, which sums individual distances along executions. Two major types have been considered here: the *discounted* accumulating distance *e.g.* in [5, 9, 42] and the *limit-average* accumulating distance *e.g.* in [9]. For real-time systems, a useful distance is the *maximum-lead* one of [28, 42] which measures the maximum difference between accumulated time delays along traces. For hybrid systems, things are more complicated, as distances between hybrid traces have to take into account both spatial and timing differences, see *e.g.* [24, 25, 38, 47].

It is our point of view that the differences between measuring distances between system traces are fundamental, in the sense that specifying one concrete way of measuring such trace distances fixes the quantitative semantics to a concrete application. Any general theory of quantitative verification should, then, be independent of the way one measures distances between traces.

In this paper we show how such a distance-independent theory of quantitative verification may be attempted. Taking as our model of quantitative systems

the *metric transition systems* of [15] and starting out with an abstract distance on traces, we define linear and branching distances and show that they have the expected properties. Our linear distances generalize trace inclusion and equivalence to the general quantitative setting, and the branching distances generalize simulation and bisimulation.

As a central technical tool in our developments, we assume that the trace distance factors through a complete lattice, and that this lifted trace distance has a recursive characterization. We show that this assumption holds for most of the trace distances considered in the above-mentioned papers; specifically, our theory can be instantiated with the point-wise (and possibly discounted) distance, the accumulating distance (both discounted and limit-average), and the maximum-lead distance.

This paper follows up on work in [21], where we develop a general quantitative theory for weighted transition systems, using the theory of quantitative games. Compared to this work, the present paper uses a different model for quantitative systems (namely, the one from [15]), hence the linear and branching distances have to be defined differently; also, no game theory is necessary for us here.

2   Metric Transition Systems

We recapitulate the setting and terminology of [15], adapting it slightly to our needs.

A *hemimetric* on a set X is a function $d : X \times X \to \mathbb{R}_{\geq 0} \cup \{\infty\}$ which satisfies $d(x, x) = 0$ and $d(x, y) + d(y, z) \geq d(x, z)$ (the *triangle inequality*) for all $x, y, z \in X$. The hemimetric is said to be *symmetric* if also $d(x, y) = d(y, x)$ for all $x, y \in X$; it is said to be *separating* if $d(x, y) = 0$ implies $x = y$. The terms "pseudometric" for a symmetric hemimetric, "quasimetric" for a separating hemimetric, and "metric" for a hemimetric which is both symmetric and separating are also in use, but we will not use them here. The tuple (X, d) is called a *hemimetric space*.

In [15], hemimetrics are called "directed metrics", "undirected" is used instead of "symmetric", and "proper" instead of "separating". Our choice of jargon is driven by a wish to follow more-or-less established terminology; specifically, the term "metric" has a standard meaning in mathematics, so that we find the term "directed metric" for what should better be called a "directed *quasi*metric" (or, "pseudometric") unfortunate. Similarly, our use of "separating" instead of "proper" is motivated by the use of this term in topology: the topology induced by a (separating) metric has the T_2 Hausdorff separation property, whereas the one induced by a pseudometric does not.

Note that our hemimetrics are *extended* in that they can take the value ∞. This is convenient for several reasons, *cf.* [36], one of them being that it allows for a disjoint union, or coproduct, of hemimetric spaces: the disjoint union of (X_1, d_1) and (X_2, d_2) is the hemimetric space $(X_1, d_1) \uplus (X_1, d_1) = (X_1 \uplus X_2, d)$ where points from different components are infinitely far away from each other,
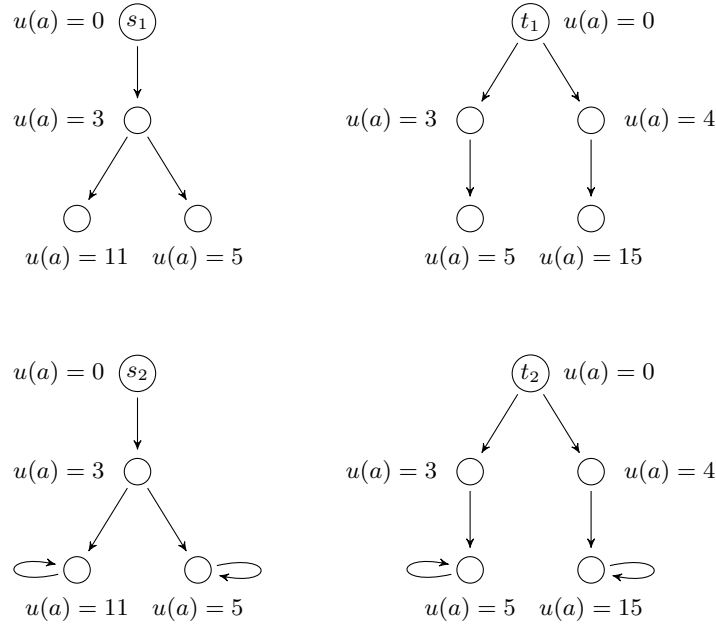
Fig. 2. Example of a metric transition system

i.e. with d defined by

$$d(x, y) = \begin{cases} d_1(x, y) & \text{if } x, y \in X_1, \\ d_2(x, y) & \text{if } x, y \in X_2, \\ \infty & \text{otherwise.} \end{cases}$$

We will need to generalize hemimetrics to codomains other than $\mathbb{R}_{\geq 0} \cup \{\infty\}$. For a partially ordered monoid $(M, \sqsubseteq, \oplus, \nvdash)$, an M-*hemimetric* on X is a function $d : X \times X \to M$ which satisfies $d(x, x) = \nvdash$ and $d(x, y) \oplus d(y, z) \sqsupseteq d(x, z)$ for all $x, y, z \in X$; symmetry and separation are generalized in similar ways.

Let Σ be a set of atomic propositions and (X, d) a hemimetric space; these will be fixed throughout this paper. A *valuation* on Σ is a mapping $u : \Sigma \to X$; the set of all valuations on Σ is denoted $\mathcal{U}[\Sigma]$. Note that in the setting of [15], each proposition a takes values in a separate hemimetric space X_a. Using extended hemimetrics allows us to unite all these spaces into one. The *propositional distance* [15] is the mapping $pd : \mathcal{U}[\Sigma] \times \mathcal{U}[\Sigma] \to \mathbb{R}_{\geq 0}$ defined by $pd(u, v) = \sup_{a \in \Sigma} d(u(a), v(a))$.

A *metric transition system* (MTS) $S = (S, T, [\cdot])$ consists of sets S of states and $T \subseteq S \times S$ of transitions, together with a state valuation mapping $[\cdot] : S \to \mathcal{U}[\Sigma]$. For $s, t \in S$, we write $s \to t$ iff $(s, t) \in T$. Fig. 2 shows a simple example of a MTS over $\Sigma = \{a\}$ which we will use later.

A *path* in S is a (finite or infinite) sequence $\pi = (\pi_0, \pi_1, \dots)$ of states $\pi_i \in S$ in which $\pi_i \to \pi_{i+1}$ for all i. Note that, extending [15], we also handle finite paths instead of only infinite ones.

A *trace* in Σ is a (finite or infinite) sequence $\sigma = (\sigma_0, \sigma_1, \dots)$ of valuations $\sigma_i \in \mathcal{U}[\Sigma]$. The set of such traces is denoted $\mathcal{U}[\Sigma]^\infty$. For a path π in S, its *induced trace* is $[\pi] = ([\pi_0], [\pi_1], \dots)$. For a state $s \in S$, we let $\mathrm{Tr}(s) = \{[\pi] \mid \pi_0 = s\}$ denote the set of traces emanating from s. We introduce some convenient notation for traces: ϵ denotes the empty trace, $u.\sigma$ the concatenation of $u \in \mathcal{U}[\Sigma]$ with $\sigma \in \mathcal{U}[\Sigma]^\infty$, and $\mathsf{len}(\sigma)$ the length (finite or ∞) of σ.

As usual, a relation $R \subseteq S \times S$ is called a *simulation* (on S) if it holds that $[s] = [t]$ for all $(s, t) \in S$ and

 – for all $s \to s'$ there is $t \to t'$ such that $(s', t') \in R$.

R is called a *bisimulation* if, additionally,

 – for all $t \to t'$ there is $s \to s'$ such that $(s', t') \in R$.

We write $s \preceq t$ if there is a simulation R with $(s, t) \in R$, and $s \approx t$ if there is a bisimulation R with $(s, t) \in R$.

3   Examples of Trace Distances

We can now give concrete examples of trace distances which have been used in the literature.

The *point-wise* trace distance is $td_{\mathsf{pw}} : \mathcal{U}[\Sigma]^\infty \times \mathcal{U}[\Sigma]^\infty \to \mathbb{R}_{\geq 0} \cup \{\infty\}$ defined by

$$td_{\mathsf{pw}}(\sigma, \tau) = \begin{cases} \infty & \text{if } \mathsf{len}(\sigma) \neq \mathsf{len}(\tau), \\ \sup_i pd(\sigma_i, \tau_i) & \text{otherwise.} \end{cases}$$

This distance has been employed in [8, 13–15, 42].

Using a *discount factor* $\lambda \in \mathbb{R}_{\geq 0}$ with $\lambda < 1$, one may discount the influence of individual distances which occur further along the traces. The *discounted point-wise* trace distance, which has been used in [13–15], is thus $td_{\mathsf{pw}, \lambda} : \mathcal{U}[\Sigma]^\infty \times \mathcal{U}[\Sigma]^\infty \to \mathbb{R}_{\geq 0} \cup \{\infty\}$ defined by

$$td_{\mathsf{pw}, \lambda}(\sigma, \tau) = \begin{cases} \infty & \text{if } \mathsf{len}(\sigma) \neq \mathsf{len}(\tau), \\ \sup_i \lambda^i pd(\sigma_i, \tau_i) & \text{otherwise.} \end{cases}$$

The *accumulating* trace distance is $td_{\mathsf{acc}} : \mathcal{U}[\Sigma]^\infty \times \mathcal{U}[\Sigma]^\infty \to \mathbb{R}_{\geq 0} \cup \{\infty\}$ defined by

$$td_{\mathsf{acc}}(\sigma, \tau) = \begin{cases} \infty & \text{if } \mathsf{len}(\sigma) \neq \mathsf{len}(\tau), \\ \sum_i pd(\sigma_i, \tau_i) & \text{otherwise.} \end{cases}$$

This distance is typically used with discounting or limit-averaging:

Using again a discount factor $\lambda \in \mathbb{R}_{\geq 0}$ with $\lambda < 1$, the *discounted accumulating* trace distance is $td_{\mathsf{acc}, \lambda} : \mathcal{U}[\Sigma]^\infty \times \mathcal{U}[\Sigma]^\infty \to \mathbb{R}_{\geq 0} \cup \{\infty\}$ defined by

$$td_{\mathsf{acc}, \lambda}(\sigma, \tau) = \begin{cases} \infty & \text{if } \mathsf{len}(\sigma) \neq \mathsf{len}(\tau), \\ \sum_i \lambda^i pd(\sigma_i, \tau_i) & \text{otherwise.} \end{cases}$$

This distance has been used in [5, 9, 42].

The *limit-average* trace distance, which has been used in [9], is td_{limavg} : $\mathcal{U}[\Sigma]^\infty \times \mathcal{U}[\Sigma]^\infty \to \mathbb{R}_{\geq 0} \cup \{\infty\}$ defined by

$$td_{\mathsf{limavg}}(\sigma, \tau) = \begin{cases} \infty & \text{if } \mathsf{len}(\sigma) \neq \mathsf{len}(\tau), \\ \liminf_j \frac{1}{j+1} \sum_{i=0}^{j} pd(\sigma_i, \tau_i) & \text{otherwise.} \end{cases}$$

This is generally defined only for infinite traces. If one wants it defined also for finite traces σ, τ (of equal length N), one can patch σ and τ so that $pd(\sigma_i, \tau_i) = 0$ for $i > N$; then $td_{\mathsf{limavg}}(\sigma, \tau) = 0$ in this case.

Both the discounted accumulating and limit-average trace distances are well-known from the theory of discounted and mean-payoff games [20, 48].

The *maximum-lead* trace distance $td_{\mathsf{maxlead}} : \mathcal{U}[\Sigma]^\infty \times \mathcal{U}[\Sigma]^\infty \to \mathbb{R}_{\geq 0} \cup \{\infty\}$ from [28, 42] is only defined for the case where the valuation space is $X = \mathbb{R}$. It is given by

$$td_{\mathsf{maxlead}}(\sigma, \tau) = \begin{cases} \infty & \text{if } \mathsf{len}(\sigma) \neq \mathsf{len}(\tau), \\ \sup_j \sup_{a \in \Sigma} \left| \sum_{i=0}^{j} \sigma_i(a) - \sum_{i=0}^{j} \tau_i(a) \right| & \text{otherwise.} \end{cases}$$

4   Linear Distances

To generalize the examples of the previous section, we define a trace distance to be a general hemimetric on traces which specializes to the propositional distance on individual valuations and is finite only for traces of equal length:

Definition 1. *A* trace distance *is a hemimetric* $td : \mathcal{U}[\Sigma]^\infty \times \mathcal{U}[\Sigma]^\infty \to \mathbb{R}_{\geq 0} \cup \{\infty\}$ *for which* $td(u, v) = pd(u, v)$ *for all* $u, v \in \mathcal{U}[\Sigma]$ *and* $td(\sigma, \tau) = \infty$ *whenever* $\mathsf{len}(\sigma) \neq \mathsf{len}(\tau)$.

We note that in case pd is separating, then also all example trace distances from Section 3 are separating, except for the limit-average distance. For the latter, $td_{\mathsf{limavg}}(\sigma, \tau) = 0$ iff either $\sigma = \tau$ are finite traces, or there exists an index k such that $\sigma_i = \tau_i$ for all $i \geq k$. Also, the maximum-lead trace distance is symmetric; the others are symmetric iff pd is symmetric.

For any given trace distance, we can define the linear distance between states in S as the (asymmetric) *Hausdorff distance* between the corresponding sets of traces:

Definition 2. *For a given trace distance td, the* linear distance *induced by td is* $ld : S \times S \to \mathbb{R}_{\geq 0} \cup \{\infty\}$ *given by*

$$ld(s, t) = \sup_{\sigma \in \mathrm{Tr}(s)} \inf_{\tau \in \mathrm{Tr}(t)} td(\sigma, \tau).$$

The symmetric linear distance *induced by td is* $\overline{ld} : S \times S \to \mathbb{R}_{\geq 0} \cup \{\infty\}$ *given by* $\overline{ld}(s, t) = \max(ld(s, t), ld(t, s))$.

Continuing the example from Fig. 2, we compute the linear distances $ld(s_1, t_1)$ and $ld(s_2, t_2)$ induced by all our example trace distances, using the usual metric on $\mathbb{R}$ for valuations and a discount factor of $\lambda = .9$ where applicable:

$$ld_{\mathsf{pw}}(s_1, t_1) = 4 \qquad\qquad ld_{\mathsf{pw}}(s_2, t_2) = 4$$
$$ld_{\mathsf{pw},\lambda}(s_1, t_1) = 3.24 \qquad\qquad ld_{\mathsf{pw}}(s_2, t_2) = 3.24$$
$$ld_{\mathsf{acc}}(s_1, t_1) = 5 \qquad\qquad ld_{\mathsf{acc}}(s_2, t_2) = \infty$$
$$ld_{\mathsf{acc},\lambda}(s_1, t_1) = 4.14 \qquad\qquad ld_{\mathsf{acc},\lambda}(s_2, t_2) = 33.3$$
$$ld_{\mathsf{limavg}}(s_1, t_2) = 0 \qquad\qquad ld_{\mathsf{limavg}}(s_2, t_2) = 4$$
$$ld_{\mathsf{maxlead}}(s_1, t_1) = 5 \qquad\qquad ld_{\mathsf{maxlead}}(s_2, t_2) = \infty$$

Our first theorem shows that for separating trace distances on finite transition systems, trace inclusion is the kernel of ld and trace equivalence the kernel of $\overline{ld}$:

Theorem 1. *Let S be finite and td separating. For all $s, t \in S$, $\mathrm{Tr}(s) \subseteq \mathrm{Tr}(t)$ iff $ld(s, t) = 0$ and $\mathrm{Tr}(s) = \mathrm{Tr}(t)$ iff $\overline{ld}(s, t) = 0$.*

Proof. It is clear that $\mathrm{Tr}(s) \subseteq \mathrm{Tr}(t)$ implies $ld(s, t) = 0$. For the opposite direction, assume $ld(s, t) = 0$ and let $\sigma \in \mathrm{Tr}(s)$. For every $i \in \mathbb{N}_+$, there exists $\tau_i \in \mathrm{Tr}(t)$ for which $td(\sigma, \tau_i) < \frac{1}{i}$. Because S is finite, there is an index N such that $\tau_i = \tau_N$ for all $i \geq N$. Then $td(\sigma, \tau_N) = 0$ and thus, as td is separating, $\sigma = \tau_N$. The second bi-implication is now clear. □

Note that we have also shown the statement that, whether td is separating or not, $ld(s, t) = 0$ implies that for every $\sigma \in \mathrm{Tr}(s)$, there exists $\tau \in \mathrm{Tr}(t)$ with $td(\sigma, \tau) = 0$. An example in [15] shows that precisely this statement may fail in case S is not finite.

5   Branching Distances

We have seen in Theorem 1 that the linear distances of the previous section are generalizations of trace inclusion and trace equivalence. In order to generalize simulation and bisimulation in a similar manner, we define branching distances.

To be able to introduce these branching distances, we need to assume that our trace distance td factors through a complete lattice, and that the lifted trace distance has a recursive characterization as given below. We will see in Section 6 that this is the case for all the example trace distances of Section 3.

For any set M, let $\mathbb{L}M = (\mathbb{R}_{\geq 0} \cup \{\infty\})^M$ be the set of functions from M to $\mathbb{R}_{\geq 0} \cup \{\infty\}$. Then $\mathbb{L}M$ is a complete lattice with partial order $\sqsubseteq$ given by $\alpha \sqsubseteq \beta$ iff $\alpha(x) \leq \beta(x)$ for all $x \in M$, and with an addition $\oplus$ given by $(\alpha \oplus \beta)(x) = \alpha(x) + \beta(x)$. The bottom element of $\mathbb{L}M$ is also the zero of $\oplus$ and given by $\bot(x) = 0$, and the top element is $\top(x) = \infty$.

Definition 3. *A recursive specification of a trace distance td consists of a set M, a lattice homomorphism $\mathsf{eval} : \mathbb{L}M \to \mathbb{R}_{\geq 0} \cup \{\infty\}$ and an $\mathbb{L}M$-hemimetric*

$td^{\mathbb{L}} : \mathcal{U}[\Sigma]^{\infty} \times \mathcal{U}[\Sigma]^{\infty} \to \mathbb{L}M$ *which together satisfy* $td = \mathsf{eval} \circ td^{\mathbb{L}}$, *and a function* $F : \mathcal{U}[\Sigma] \times \mathcal{U}[\Sigma] \times \mathbb{L}M \to \mathbb{L}M$. F *must be monotone in the third coordinate, i.e.* $F(u, v, \cdot) : \mathbb{L}M \to \mathbb{L}M$ *is monotone for all* $u, v \in \mathcal{U}[\Sigma]$, *have* $F(u, u, \bot) = \bot$ *for all* $u \in \mathcal{U}[\Sigma]$, *and satisfy, for all* $u, v \in \mathcal{U}[\Sigma]$ *and* $\sigma, \tau \in \mathcal{U}[\Sigma]^{\infty}$, *that* $td^{\mathbb{L}}(u.\sigma, v.\tau) = F(u, v, td^{\mathbb{L}}(\sigma, \tau))$.

Now if td is recursively specified as above, then we can use the recursion to introduce branching distances sd and bd which generalize simulation and bisimulation:

Definition 4. *For a recursively specified trace distance td, let* $sd^{\mathbb{L}}, bd^{\mathbb{L}} : S \times S \to \mathbb{L}M$ *be the respective least fixed points to the equations*

$$sd^{\mathbb{L}}(s, t) = \sup_{s \to s'} \inf_{t \to t'} F([s], [t], sd^{\mathbb{L}}(s', t')), \tag{1}$$

$$bd^{\mathbb{L}}(s, t) = \max \begin{cases} \sup_{s \to s'} \inf_{t \to t'} F([s], [t], bd^{\mathbb{L}}(s', t')), \\ \sup_{t \to t'} \inf_{s \to s'} F([s], [t], bd^{\mathbb{L}}(s', t')). \end{cases} \tag{2}$$

The simulation distance *induced by td is* $sd : S \times S \to \mathbb{R}_{\geq 0} \cup \{\infty\}$ *given by* $sd = \mathsf{eval} \circ sd^{\mathbb{L}}$; *the* bisimulation distance *induced by td is* $bd : S \times S \to \mathbb{R}_{\geq 0} \cup \{\infty\}$ *given by* $bd = \mathsf{eval} \circ bd^{\mathbb{L}}$.

Note that we define the distances using *least* fixed points, as opposed to the *greatest* fixed point definition of standard (bi)simulation. Informally, this is because our order is reversed: we are not interested in maximizing (bi)simulation relations, but in *minimizing* (bi)simulation distance.

Lemma 1. *The mappings sd and bd are well-defined hemimetrics on* S.

Proof. We show the proof for sd; for bd it is similar. Let $\mathbb{F}S = \mathbb{L}M^{S \times S}$ be the lattice of functions from $S \times S$ to $\mathbb{L}M$, then $\mathbb{F}S$ is complete because $\mathbb{L}M$ is. Let $I : \mathbb{F}S \to \mathbb{F}S$ be defined by

$$I(f)(s, t) = \sup_{s \to s'} \inf_{t \to t'} F([s], [t], f(s', t')),$$

similarly to (1). Because $F([s], [t], \cdot) : \mathbb{L}M \to \mathbb{L}M$ is monotone for all $s, t \in S$, I is monotone. Using Tarski's fixed-point theorem, we can hence conclude that I has a unique minimal fixed point, which is $sd^{\mathbb{L}}$. Clearly $sd^{\mathbb{L}}(s, s) = \bot$ for all $s \in S$, and by induction one can show that $sd^{\mathbb{L}}(s, t) \oplus sd^{\mathbb{L}}(t, u) \sqsupseteq sd^{\mathbb{L}}(s, u)$ for all $s, t, u \in S$. Hence $sd^{\mathbb{L}}$ is an $\mathbb{L}M$-hemimetric and sd is a hemimetric. □

In order to show, similarly to Theorem 1, that simulation is the kernel of simulation distance, we need a condition on the recursive F which mimics the separation condition for hemimetrics. We say that a recursively specified trace distance td is *recursively separating* if $F : \mathcal{U}[\Sigma] \times \mathcal{U}[\Sigma] \times \mathbb{L}M \to \mathbb{L}M$ satisfies the condition that whenever $F(u, v, x) = \bot$, then $u = v$ and $x = \bot$. Note that this condition implies that $td^{\mathbb{L}}$ is separating.

Theorem 2. *Let S be finite and td recursively specified and recursively separating. For all $s, t \in S$, $s \preceq t$ iff $sd^{\mathbb{L}}(s,t) = \bot$ and $s \approx t$ iff $bd^{\mathbb{L}}(s,t) = \bot$.*

Proof. It is clear that $s \preceq t$ implies $sd^{\mathbb{L}}(s,t) = \bot$. For the other direction, let $R = \{(s', t') \mid sd^{\mathbb{L}}(s', t') = \bot\} \subseteq S \times S$. Then $(s, t) \in R$. Let $(s', t') \in R$, then $\sup_{s' \to s''} \inf_{t' \to t''} F([s'], [t'], sd^{\mathbb{L}}(s'', t'')) = \bot$. As S is finite, this implies that for all $s' \to s''$, there exists $t' \to t''$ with $F([s'], [t'], sd^{\mathbb{L}}(s'', t'')) = \bot$. By recursive separation, we hence have $[s'] = [t']$ and $sd^{\mathbb{L}}(s'', t'') = \bot$. We have shown that R is a simulation on S. The proof that $s \approx t$ iff $bd^{\mathbb{L}}(s,t) = \bot$ is similar.   □

The next theorem gives the relations between the different distances we have introduced. Also these relations generalize the situation in the Boolean setting: in light of Theorems 1 and 2, they are quantitative analogues to the facts that simulation implies trace inclusion and that bisimulation implies simulation and trace equivalence.

Theorem 3. *Let td be recursively specified. For all $s, t \in S$, $ld(s,t) \le sd(s,t) \le bd(s,t)$ and $\overline{ld}(s,t) \le bd(s,t)$.*

Proof. The proof is best understood in a setting of *quantitative games*, *cf.* [21]. In this setting, the standard simulation and bisimulation games [41] are generalized to games with quantitative objectives. One can then see that the linear distances can be computed by similar games, and that the only differences between these games are given by certain restrictions on the strategies available to the first player. The result follows from inclusions on these sets of restricted strategies.

We can, however, also give a direct proof of the fact that $ld(s,t) \le sd(s,t)$ without resorting to games (and similar proofs may be given for the other inequalities). To do so, we need to lift ld to the lattice $\mathbb{L}M$: for $s, t \in S$, define

$$ld^{\mathbb{L}}(s,t) = \sup_{\sigma \in \mathrm{Tr}(s)} \inf_{\tau \in \mathrm{Tr}(t)} td^{\mathbb{L}}(\sigma, \tau),$$

then $ld = \mathsf{eval} \circ ld^{\mathbb{L}}$ because eval is monotone. We show that $ld^{\mathbb{L}}(s,t) \sqsubseteq sd^{\mathbb{L}}(s,t)$ for all $s, t \in S$, which will imply the result.

Let $s, t \in S$. We have

$$
\begin{aligned}
ld^{\mathbb{L}}(s,t) &= \sup_{\sigma \in \mathrm{Tr}(s)} \inf_{\tau \in \mathrm{Tr}(t)} td^{\mathbb{L}}(\sigma, \tau) \\
&= \sup_{s \to s'} \sup_{\sigma' \in \mathrm{Tr}(s')} \inf_{t \to t'} \inf_{\tau' \in \mathrm{Tr}(t')} td^{\mathbb{L}}([s].\sigma', [t].\tau') \\
&= \sup_{s \to s'} \sup_{\sigma' \in \mathrm{Tr}(s')} \inf_{t \to t'} \inf_{\tau' \in \mathrm{Tr}(t')} F([s], [t], td^{\mathbb{L}}(\sigma', \tau')) \\
&\sqsubseteq \sup_{s \to s'} \inf_{t \to t'} \sup_{\sigma' \in \mathrm{Tr}(s')} \inf_{\tau' \in \mathrm{Tr}(t')} F([s], [t], td^{\mathbb{L}}(\sigma', \tau')) \\
&= \sup_{s \to s'} \inf_{t \to t'} F([s], [t], \sup_{\sigma' \in \mathrm{Tr}(s')} \inf_{\tau' \in \mathrm{Tr}(t')} td^{\mathbb{L}}(\sigma', \tau')) \\
&= \sup_{s \to s'} \inf_{t \to t'} F([s], [t], ld^{\mathbb{L}}(s', t')),
\end{aligned}
$$

and the statement now follows by induction.   □

6   Examples Revisited

We can now apply the constructions of Section 5 to the example trace distances from Section 3. We give recursive specifications for all distances and deduce the corresponding branching distances. For ease of exposition, we will only consider ourselves with the simulation distances here, but similar things can be said about the bisimulation distances.

For the *point-wise* trace distance td_{pw}, a recursive specification is given as follows (where $\{*\}$ denotes the one-point set; hence $\mathbb{L}M$ is isomorphic to $\mathbb{R}_{\geq 0} \cup \{\infty\}$):

$$M = \{*\} \qquad \mathsf{eval}(x) = x$$
$$td^{\mathbb{L}}(\sigma, \tau) = \sup_i pd(\sigma_i, \tau_i)$$
$$F(u, v, x) = \max(pd(u, v), x)$$

Using Definition 4, the corresponding simulation distance $sd_{\mathsf{pw}} = sd^{\mathbb{L}}_{\mathsf{pw}}$ is the least fixed point to the equations

$$sd_{\mathsf{pw}}(s, t) = \sup_{s \to s'} \inf_{t \to t'} \max(pd([s], [t]), sd_{\mathsf{pw}}(s', t')).$$

This is similar to the formulation given in [14, 15]. Note that if pd is separating, then F is recursively separating, hence Theorem 2 applies.

For the *discounted point-wise* trace distance $td_{\mathsf{pw}, \lambda}$ the recursive specification is similar:

$$M = \{*\} \qquad \mathsf{eval}(x) = x$$
$$td^{\mathbb{L}}(\sigma, \tau) = \sup_i \lambda^i pd(\sigma_i, \tau_i)$$
$$F(u, v, x) = \max(pd(u, v), \lambda x)$$

The corresponding simulation distance is hence the least fixed point to the equations

$$sd_{\mathsf{pw}, \lambda}(s, t) = \sup_{s \to s'} \inf_{t \to t'} \max(pd([s], [t]), \lambda sd_{\mathsf{pw}, \lambda}(s', t')),$$

which is similar to what is in [14, 15]. Note again that if pd is separating, then F is recursively separating, hence also here Theorem 2 applies.

Also the *accumulating* trace distance td_{acc} has a simple recursive specification with $\mathbb{L}M$ isomorphic to $\mathbb{R}_{\geq 0} \cup \{\infty\}$:

$$M = \{*\} \qquad \mathsf{eval}(x) = x$$
$$td^{\mathbb{L}}(\sigma, \tau) = \sum_i pd(\sigma_i, \tau_i)$$
$$F(u, v, x) = pd(u, v) + x$$

The corresponding simulation distance is hence the least fixed point to the equations

$$sd_{\mathsf{acc}}(s, t) = \sup_{s \to s'} \inf_{t \to t'} (pd([s], [t]) + sd_{\mathsf{acc}}(s', t')).$$

Again, if pd is separating, then F is recursively separating, hence Theorem 2 applies.

A recursive specification for the *discounted accumulating* trace distance is given as follows:

$$M = \{*\} \qquad \mathsf{eval}(x) = x$$
$$td^{\mathbb{L}}(\sigma, \tau) = \sum_i \lambda^i pd(\sigma_i, \tau_i)$$
$$F(u, v, x) = pd(u, v) + \lambda x$$

The corresponding simulation distance is then the least fixed point to the equations

$$sd_{\mathsf{acc}, \lambda}(s, t) = \sup_{s \to s'} \inf_{t \to t'} pd([s], [t]) + \lambda sd_{\mathsf{acc}, \lambda}(s', t'),$$

similarly to what is in [9]. If pd is separating, then F is recursively separating, hence Theorem 2 applies also in this case.

To obtain a recursive specification of the *limit-average* trace distance, we need a richer lattice:

$$M = \mathbb{N} \qquad \mathsf{eval}(x) = \liminf_j x(j)$$
$$td^{\mathbb{L}}(\sigma, \tau)(j) = \frac{1}{j+1} \sum_{i=0}^{j} pd(\sigma_i, \tau_i)$$
$$F(u, v, x)(j) = \frac{1}{j+1} pd(u, v) + \frac{j}{j+1} x(j-1)$$

Using Definition 4, we obtain the corresponding lifted simulation distance as the least fixed point to the equations

$$sd_{\mathsf{limavg}}^{\mathbb{L}}(s, t)(j) = \sup_{s \to s'} \inf_{t \to t'} \left(\frac{1}{j+1} pd([s], [t]) + \frac{j}{j+1} sd_{\mathsf{limavg}}^{\mathbb{L}}(s', t')(j-1) \right).$$

The limit-average simulation distance is then

$$sd_{\mathsf{limavg}}(s, t) = \liminf_j sd_{\mathsf{limavg}}^{\mathbb{L}}(s, t)(j).$$

To the best of our knowledge, this formulation of limit-average simulation distance is new. We again remark that if pd is separating, then F is recursively separating, hence Theorem 2 applies.

For the *maximum-lead* trace distance, we need a lattice which maps leads to maximum leads. A recursive specification is as follows:

$$M = \mathbb{R} \qquad \mathsf{eval}(x) = x(0)$$
$$td^{\mathbb{L}}(\sigma, \tau)(\delta) = \max(|\delta|, \sup_j \sup_{a \in \Sigma} |\delta + \sum_{i=0}^{j} \sigma_i(a) - \sum_{i=0}^{j} \tau_i(a)|)$$
$$F(u, v, x)(\delta) = \sup_{a \in \Sigma} \max(|\delta + u(a) - v(a)|, x(\delta + u(a) - v(a)))$$

The lifted simulation distance is then the least fixed point to the equations

$$sd_{\mathsf{maxlead}}^{\mathbb{L}}(s, t)(\delta) = \sup_{s \to s'} \inf_{t \to t'} \sup_{a \in \Sigma} \max \begin{cases} |\delta + [s](a) - [t](a)|, \\ sd_{\mathsf{maxlead}}^{\mathbb{L}}(s', t')(\delta + [s](a) - [t](a)), \end{cases}$$

cf. [28], and maximum-lead simulation distance is

$$sd_{\mathsf{maxlead}}(s,t) = sd^{\mathbb{L}}_{\mathsf{maxlead}}(s,t)(0).$$

Also here it holds that if pd is separating, then F is recursively separating, hence Theorem 2 applies.

Finishing the example from Fig. 2, we compute the simulation distances $sd(s_1,t_1)$ and $sd(s_2,t_2)$ induced by all our example trace distances, using the usual metric on $\mathbb{R}$ for valuations and a discount factor of $\lambda = .9$ where applicable:

$$sd_{\mathsf{pw}}(s_1,t_1) = 6 \qquad\qquad sd_{\mathsf{pw}}(s_2,t_2) = 6$$
$$sd_{\mathsf{pw},\lambda}(s_1,t_1) = 5.46 \qquad\qquad sd_{\mathsf{pw},\lambda}(s_2,t_2) = 5.46$$
$$sd_{\mathsf{acc}}(s_1,t_1) = 6 \qquad\qquad sd_{\mathsf{acc}}(s_2,t_2) = \infty$$
$$sd_{\mathsf{acc},\lambda}(s_1,t_1) = 5.46 \qquad\qquad sd_{\mathsf{acc},\lambda}(s_2,t_2) = 54$$
$$sd_{\mathsf{limavg}}(s_1,t_2) = 0 \qquad\qquad sd_{\mathsf{limavg}}(s_2,t_2) = 6$$
$$sd_{\mathsf{maxlead}}(s_1,t_1) = 6 \qquad\qquad sd_{\mathsf{maxlead}}(s_2,t_2) = \infty$$

7   A Note on Robustness

In [15] it is shown that with respect to the *point-wise* linear and branching distances, metric transition systems are *robust* to perturbations in the state valuations. To be precise, let $[\cdot]_1, [\cdot]_2 : S \to \mathcal{U}[\Sigma]$ be two different state valuations on a MTS S and define their *valuation distance* by $vd([\cdot]_1, [\cdot]_2) = \sup_{s \in S} pd([s]_1, [s]_2)$. This measures how close the state valuations are to each other.

Now write ld^i_{pw} and sd^i_{pw} (with $i \in \{1,2\}$) for the point-wise linear and simulation distances with respect to the valuation $[\cdot]_i$. It is shown in [15] that for all $s,t \in S$,

$$|ld^1_{\mathsf{pw}}(s,t) - ld^2_{\mathsf{pw}}(s,t)| \le vd([\cdot]_1, [\cdot]_2) + vd([\cdot]_2, [\cdot]_1),$$
$$|sd^1_{\mathsf{pw}}(s,t) - sd^2_{\mathsf{pw}}(s,t)| \le vd([\cdot]_1, [\cdot]_2) + vd([\cdot]_2, [\cdot]_1).$$

This is, hence, a *robustness* result: given that the two valuations are close to each other, also the linear and branching distances will be.

Similar results can easily be seen to hold also for the symmetric linear and the bisimulation distances, and also for the *discounted point-wise* versions of these distances. A result similar in spirit, also for the point-wise distance, is reported for robustness of timed automata in [8].

Using almost the same arguments as in [15], one can show that for the *discounted accumulating* distances,

$$|ld^1_{\mathsf{acc},\lambda}(s,t) - ld^2_{\mathsf{acc},\lambda}(s,t)| \le \tfrac{1}{1-\lambda}(vd([\cdot]_1, [\cdot]_2) + vd([\cdot]_2, [\cdot]_1)),$$
$$|sd^1_{\mathsf{acc},\lambda}(s,t) - sd^2_{\mathsf{acc},\lambda}(s,t)| \le \tfrac{1}{1-\lambda}(vd([\cdot]_1, [\cdot]_2) + vd([\cdot]_2, [\cdot]_1)).$$

Hence MTS are also robust with respect to the discounted accumulating distances. In the proof, one uses the convergence of the geometric series: after the ith

$$u_1(a) = 0 \quad u_1(a) = 0 \qquad\qquad u_2(a) = 0 \quad u_2(a) = 1$$

Fig. 3. Example of a MTS with two different state valuations.

step, distances are discounted by λ^i, so no more than $\lambda^i(vd([\cdot]_1, [\cdot]_2) + vd([\cdot]_2, [\cdot]_1))$ can be added to the total distance. Hence the distance is bounded above by $(vd([\cdot]_1, [\cdot]_2) + vd([\cdot]_2, [\cdot]_1)) \sum_{i=0}^{\infty} \lambda^i = \frac{1}{1-\lambda}(vd([\cdot]_1, [\cdot]_2) + vd([\cdot]_2, [\cdot]_1))$.

Unfortunately, no general robustness results are available, and our other example distances do not have similar properties. This is shown by the example in Fig. 3. Here, $vd([\cdot]_1, [\cdot]_2) = vd([\cdot]_2, [\cdot]_1) = 1$, and the linear distances are as follows, with the usual metric on $\mathbb{R}$ and $\lambda = .9$ where applicable:

$$
\begin{array}{llll}
ld^1_{\mathsf{pw}}(s,t) = 0 & ld^2_{\mathsf{pw}}(s,t) = 1 & ld^1_{\mathsf{pw}}(t,s) = 0 & ld^2_{\mathsf{pw}}(t,s) = 1 \\
ld^1_{\mathsf{pw},\lambda}(s,t) = 0 & ld^2_{\mathsf{pw},\lambda}(s,t) = 1 & ld^1_{\mathsf{pw},\lambda}(t,s) = 0 & ld^2_{\mathsf{pw},\lambda}(t,s) = 1 \\
ld^1_{\mathsf{acc}}(s,t) = 0 & ld^2_{\mathsf{acc}}(s,t) = \infty & ld^1_{\mathsf{acc}}(t,s) = 0 & ld^2_{\mathsf{acc}}(t,s) = \infty \\
ld^1_{\mathsf{acc},\lambda}(s,t) = 0 & ld^2_{\mathsf{acc},\lambda}(s,t) = 10 & ld^1_{\mathsf{acc},\lambda}(t,s) = 0 & ld^2_{\mathsf{acc},\lambda}(t,s) = 10 \\
ld^1_{\mathsf{limavg}}(s,t) = 0 & ld^2_{\mathsf{limavg}}(s,t) = \infty & ld^1_{\mathsf{limavg}}(t,s) = 0 & ld^2_{\mathsf{limavg}}(t,s) = \infty \\
ld^1_{\mathsf{maxlead}}(s,t) = 0 & ld^2_{\mathsf{maxlead}}(s,t) = \infty & ld^1_{\mathsf{maxlead}}(t,s) = 0 & ld^2_{\mathsf{maxlead}}(t,s) = \infty
\end{array}
$$

(The branching distances are equal to the linear distances in all cases.)

8   Conclusion

We have shown how the model of metric transition systems from [15] can be embedded in a general quantitative framework which allows quantitative verification using a large number of different system distances. As these distances are an essential part of quantitative verification and, at the same time, typically depend on what precise application one has in mind, it is important to develop a general quantitative theory of systems which is independent of the employed distances. This is what we have done here.

Assuming an abstract trace distance as input, we have developed corresponding linear and branching distances. What we have not done, however, is to compare linear and branching distances which arise from *different* trace distances. One important question is, for example, whether the linear and branching distances one obtains from two Lipschitz equivalent, or topologically equivalent, trace distances will again be Lipschitz or topologically equivalent. This would be a crucial step in a *classification* of system distances and is part of our ongoing research.

Another issue which we have not treated here is the *logical* side of quantitative verification. In [15], the authors introduce a quantitative variant of LTL which characterizes the point-wise linear distance, and a quantitative μ-calculus which

characterizes the point-wise simulation and bisimulation distances. We plan to work on similar logics for our general setting.

References

1. R. Alur, C. Courcoubetis, N. Halbwachs, T. A. Henzinger, P.-H. Ho, X. Nicollin, A. Olivero, J. Sifakis, and S. Yovine. The algorithmic analysis of hybrid systems. *TCS*, 138(1):3–34, 1995.
2. R. Alur and D. L. Dill. A theory of timed automata. *TCS*, 126(2):183–235, 1994.
3. A. Aziz, K. Sanwal, V. Singhal, and R. K. Brayton. Model-checking continous-time Markov chains. *ACM Trans. Comput. Log.*, 1(1):162–170, 2000.
4. G. Bacci, G. Bacci, K. G. Larsen, and R. Mardare. On-the-fly exact computation of bisimilarity distances. In *TACAS*, pp. 1–15, 2013.
5. S. S. Bauer, U. Fahrenberg, L. Juhl, K. G. Larsen, A. Legay, and C. Thrane. Weighted modal transition systems. *Form. Meth. Syst. Design*, 2013. Online first.
6. S. S. Bauer, U. Fahrenberg, A. Legay, and C. Thrane. General quantitative specification theories with modalities. In *CSR*, pp. 18–30, 2012.
7. U. Boker and T. A. Henzinger. Approximate determinization of quantitative automata. In *FSTTCS*, pp. 362–373, 2012.
8. P. Bouyer, K. G. Larsen, N. Markey, O. Sankur, and C. Thrane. Timed automata can always be made implementable. In *CONCUR*, pp. 76–91, 2011.
9. P. Černý, T. A. Henzinger, and A. Radhakrishna. Simulation distances. In *CONCUR*, pp. 253–268, 2010.
10. P. Černý, T. A. Henzinger, and A. Radhakrishna. Quantitative abstraction refinement. In *POPL*, pp. 115–128, 2013.
11. K. Chatterjee, L. de Alfaro, M. Faella, T. A. Henzinger, R. Majumdar, and M. Stoelinga. Compositional quantitative reasoning. In *QEST*, pp. 179–188, 2006.
12. K. Chatterjee, L. Doyen, and T. A. Henzinger. Quantitative languages. *ACM Trans. Comput. Log.*, 11(4), 2010.
13. L. de Alfaro, M. Faella, T. A. Henzinger, R. Majumdar, and M. Stoelinga. Model checking discounted temporal properties. *TCS*, 345(1):139–170, 2005.
14. L. de Alfaro, M. Faella, and M. Stoelinga. Linear and branching metrics for quantitative transition systems. In *ICALP*, pp. 97–109, 2004.
15. L. de Alfaro, M. Faella, and M. Stoelinga. Linear and branching system metrics. *IEEE Trans. Softw. Eng.*, 35(2):258–273, 2009.
16. J. Desharnais, V. Gupta, R. Jagadeesan, and P. Panangaden. Metrics for labelled Markov processes. *TCS*, 318(3):323–354, 2004.
17. M. Droste and P. Gastin. Weighted automata and weighted logics. *TCS*, 380(1-2):69–86, 2007.
18. M. Droste and I. Meinecke. Weighted automata and weighted MSO logics for average and long-time behaviors. *Inf. Comp.*, 220:44–59, 2012.
19. M. Droste and G. Rahonis. Weighted automata and weighted logics with discounting. *TCS*, 410(37):3481–3494, 2009.
20. A. Ehrenfeucht and J. Mycielski. Positional strategies for mean payoff games. *International Journal of Game Theory*, 8:109–113, 1979.
21. U. Fahrenberg, A. Legay, and C. Thrane. The quantitative linear-time–branching-time spectrum. In *FSTTCS*, pp. 103–114, 2011.
22. G. Frehse, C. L. Guernic, A. Donzé, S. Cotton, R. Ray, O. Lebeltel, R. Ripado, A. Girard, T. Dang, and O. Maler. SpaceEx: Scalable verification of hybrid systems. In *CAV*, pp. 379–395, 2011.

23. S. Gilmore and J. Hillston. The PEPA workbench: A tool to support a process algebra-based approach to performance modelling. In *CPE*, pp. 353–368, 1994.
24. A. Girard. Synthesis using approximately bisimilar abstractions: Time-optimal control problems. In *CDC*, pp. 5893–5898, 2010.
25. A. Girard and G. J. Pappas. Approximation metrics for discrete and continuous systems. *IEEE Trans. Automat. Contr.*, 52(5):782–798, 2007.
26. H. Hansson and B. Jonsson. A logic for reasoning about time and reliability. *Formal Asp. Comput.*, 6(5):512–535, 1994.
27. T. A. Henzinger, P.-H. Ho, and H. Wong-Toi. Hytech: A model checker for hybrid systems. *Int. J. Softw. Tools Techn. Trans.*, 1(1-2):110–122, 1997.
28. T. A. Henzinger, R. Majumdar, and V. S. Prabhu. Quantifying similarities between timed systems. In *FORMATS*, pp. 226–241, 2005.
29. T. A. Henzinger, X. Nicollin, J. Sifakis, and S. Yovine. Symbolic model checking for real-time systems. *Inf. Comp.*, 111(2):193–244, 1994.
30. J. Hillston. *A Compositional Approach to Performance Modelling.* Cambridge University Press, 1996.
31. R. Koymans. Specifying real-time properties with metric temporal logic. *Real-Time Systems*, 2(4):255–299, 1990.
32. M. Z. Kwiatkowska, G. Norman, and D. Parker. Probabilistic symbolic model checking with PRISM: A hybrid approach. In *TACAS*, pp. 52–66, 2002.
33. K. G. Larsen, A. Legay, L.-M. Traonouez, and A. Wąsowski. Robust specification of real time components. In *FORMATS*, pp. 129–144, 2011.
34. K. G. Larsen, R. Mardare, and P. Panangaden. Taking it to the limit: Approximate reasoning for Markov processes. In *MFCS*, pp. 681–692, 2012.
35. K. G. Larsen, P. Pettersson, and W. Yi. UPPAAL in a nutshell. *Int. J. Softw. Tools Techn. Trans.*, 1(1-2):134–152, 1997.
36. F. W. Lawvere. Metric spaces, generalized logic, and closed categories. *Rendiconti del seminario matématico e fisico di Milano*, XLIII:135–166, 1973.
37. P. Panangaden. *Labelled Markov Processes.* Imperial College Press, 2009.
38. J.-D. Quesel, M. Fränzle, and W. Damm. Crossing the bridge between similar games. In *FORMATS*, pp. 160–176, 2011.
39. R. Segala and N. A. Lynch. Probabilistic simulations for probabilistic processes. In *CONCUR*, pp. 481–496, 1994.
40. W. J. Stewart. *Introduction to the Numerical Solution of Markov Chains.* Princeton University Press, 1994.
41. C. Stirling. Modal and temporal logics for processes. In *Banff Higher Order Workshop*, vol. 1043 of *LNCS*, pp. 149–237. Springer, 1995.
42. C. Thrane, U. Fahrenberg, and K. G. Larsen. Quantitative analysis of weighted transition systems. *J. Log. Alg. Prog.*, 79(7):689–703, 2010.
43. F. van Breugel. An introduction to metric semantics: operational and denotational models for programming and specification languages. *TCS*, 258(1-2):1–98, 2001.
44. F. van Breugel and J. Worrell. A behavioural pseudometric for probabilistic transition systems. *TCS*, 331(1):115–142, 2005.
45. F. van Breugel and J. Worrell. Approximating and computing behavioural distances in probabilistic transition systems. *TCS*, 360(1-3):373–385, 2006.
46. F. Wang, A. K. Mok, and E. A. Emerson. Symbolic model checking for distributed real-time systems. In *FME*, pp. 632–651, 1993.
47. G. Zheng and A. Girard. Bounded and unbounded safety verification using bisimulation metrics. In *HSCC*, pp. 426–440, 2009.
48. U. Zwick and M. Paterson. The complexity of mean payoff games. In *COCOON*, pp. 1–10, 1995.