Weighted modal transition systems

Sebastian S. Bauer, Uli Fahrenberg, Line Juhl, Kim Guldstrand Larsen, Axel Legay, Claus Thrane

Noname manuscript No.
(will be inserted by the editor)

Weighted Modal Transition Systems

Sebastian S. Bauer · Uli Fahrenberg · Line Juhl · Kim G. Larsen · Axel Legay · Claus Thrane

the date of receipt and acceptance should be inserted later

Abstract Specification theories as a tool in model-driven development processes of component-based software systems have recently attracted a considerable attention. Current specification theories are however qualitative in nature, and therefore fragile in the sense that the inevitable approximation of systems by models, combined with the fundamental unpredictability of hardware platforms, makes it difficult to transfer conclusions about the behavior, based on models, to the actual system. Hence this approach is arguably unsuited for modern software systems. We propose here the first specification theory which allows to capture quantitative aspects during the refinement and implementation process, thus leveraging the problems of the qualitative setting.

Our proposed quantitative specification framework uses weighted modal transition systems as a formal model of specifications. These are labeled transition systems with the additional feature that they can model optional behavior which may or may not be implemented by the system. Satisfaction and refinement is lifted from the well-known qualitative to our quantitative setting, by introducing a notion of distances between weighted modal transition systems. We show that quantitative versions of parallel composition as well as quotient (the dual to parallel composition) inherit the properties from the Boolean setting.

Keywords reducing complexity of design, modal specification, quantitative reasoning

This paper is based on the conference contribution [6] which was presented at the 36th International Symposium on Mathematical Foundations of Computer Science, MFCS 2011, Warszawa, Poland.

Sebastian S. Bauer
Ludwig-Maximilians-Universität München, Germany

Uli Fahrenberg · Axel Legay
Irisa / INRIA Rennes, France

Kim G. Larsen · Line Juhl · Claus Thrane
Aalborg University, Denmark

1 Introduction

One of the major current challenges to rigorous design of software systems is that these systems are becoming increasingly complex and difficult to reason about [40]. As an example, an integrated communication system in a modern airplane can have more than 10^{900} distinct states [5], and state-of-the-art tools offer no possibility to reason about, and model check, the system as a whole.

One promising approach to overcome such problems is the one of *compositional and incremental design*. Here the reasoning is done as much as possible at higher *specification* levels rather than at *implementations*; partial specifications are proven correct and then composed and refined until one arrives at an implementation model. Practice has shown that this is indeed a viable approach [15,41].

Specifications of system requirements are high-level finite abstractions of possibly infinite sets of implementations. A model of a system is considered an implementation of a given specification if the behavior defined by the implementation is implied by the description provided by the specification.

Any practical specification formalism comes equipped with a number of operations which allow compositional and incremental reasoning. The first of these is a *refinement* relation which allows to successively distill specifications into more detailed ones and eventually into implementations. In an implementation, all optional behavior defined in the specification has been decided upon in compliance with the specification.

Also needed is an operation of *logical conjunction* which allows to combine specifications so that the systems which refine the conjunction of two specifications are precisely the ones which satisfy both partial specifications. Refinement and conjunction together allow for incremental reasoning as specifications are successively refined and composed.

For compositional reasoning, one needs an operation of *structural composition* which allows to infer specifications from sub-specifications of independent requirements, mimicking at the implementation level *e.g.* the interaction of components in a distributed system. A partial inverse of this operation is given by the *quotient* operation which allows to synthesize a specification of the missing components from an overall specification and an implementation which realizes a part of the overall specification.

Over the years, there have been a series of advances on specification theories [2,12, 17,21,35,37,42]. The predominant approaches are based on modal logics and process algebras but have the drawback that they cannot naturally embed both logical and structural composition within the same formalism [31]. Hence such formalisms do not permit to reason incrementally through refinement.

In order to leverage these problems, the concept of *modal transition systems* was introduced [31]. In short, modal transition systems are labeled transition systems equipped with two types of transitions: *must* transitions which are mandatory for any implementation, and *may* transitions which are optional for implementations. It is well established that modal transition systems match all the requirements of a reasonable specification theory (see also [38] for motivation), and much progress has been made using modal specifications, see *e.g.* [4] for an overview. Also, practical experience shows that the formalism is expressive enough to handle complex industrial problems [15,41].

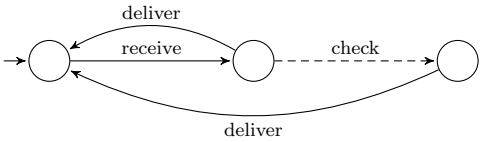


Fig. 1 Modal transition system modeling a simple email system, with an optional behavior: Once an email is received it may *e.g.* be scanned for containing viruses, or automatically decrypted, before it is delivered to the receiver.

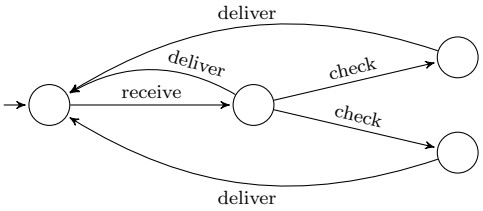


Fig. 2 An implementation of the simple email system in Figure 1 in which we explicitly model two distinct types of email pre-processing.

As an example, consider the modal transition system shown in Figure 1 which models the requirements of a simple email system in which emails are first received and then delivered. Before delivering the email, the system may check or process the email, *e.g.* for en- or decryption, filtering of spam emails, or generating automatic answers using has an auto-reply feature (see also [29]). *Must* transitions, representing obligatory behavior, are drawn as solid arrows, whereas *may* transitions, modeling optional behavior, are shown as dashed arrows; hence any implementation of this email system specification must be able to receive and deliver email, and it may also be able to check arriving email before delivering it. No other behavior is allowed.

Implementations can also be represented within the modal transition system formalism, simply as specifications without *may* transitions. Hence any implementation choice has been resolved, and implementations are plain labeled transition systems. Formally, for a labeled transition system to be an implementation of a given specification, we require that the states of the two objects are related by a refinement relation with the property that all behavior required (*must*) by the specification has been implemented, and that any implementation behavior was permitted (*may*) in the specification. Figure 2 shows an implementation of our email specification with two different checks, leading to distinct processing states. Note that a simple system without any check at all, hence only able to receive and deliver email, is also an implementation of the specification.

Motivated by applications to embedded, real-time and hybrid systems, the modal transition system framework has recently been extended in order to reason about *quantitative* aspects [7,30]. With these applications in mind, it is necessary not only to be able to *specify* quantitative aspects of systems, but also to formalize successive *refinement* of quantities. To illustrate this extension, consider again the modal transition system of Figure 1, but this time with quantities, see Figure 3: Every transition label is extended by integer intervals modeling upper and lower bounds on time required for performing the corresponding actions. For instance, the recep-

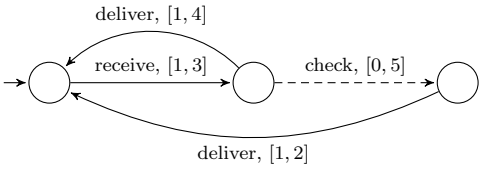


Fig. 3 Specification of a simple email system, similar to Figure 1, but extended by integer intervals modeling time units for performing the corresponding actions.

tion of a new email (action *receive*) must take between one and three time units, the checking of the email (action *check*) is allowed to take up to five time units.

In this quantitative setting, there is a problem with using a *Boolean* notion of refinement (as is done in [7,30]): If one only can decide *whether or not* an implementation refines a specification, then the quantitative aspects get lost in the refinement process. As an example, consider the email system implementations in Figure 4. Implementation (a) does not refine the specification, as there is an error in the discrete structure of actions: after receiving an email, the system can check it indefinitely without ever delivering it. Also implementations (b) and (c) do not refine the specification: (b) takes too long to receive email, (c) does not deliver email fast enough after checking it. Implementation (d) on the other hand is a perfect refinement of the specification.

Intuitively however, implementations (b) and (c) conform much better to the specification than implementation (a) in Figure 4: there are no discrepancies in the discrete structure, only the weights are off by 1. Additionally, the quantitative error in implementation (c) occurs later than the one in (b). Hence one may want to say that implementation (d) is in perfect refinement of the specification, (c) is slightly off, (b) is a bit more problematic, whereas implementation (a) is completely unacceptable. A Boolean notion of refinement does not allow to make such distinctions between different negative answers.

To sum up, a Boolean notion of refinement is too *fragile* for quantitative formalisms. Minor and major modifications in the implementation cannot be distinguished, as both of them may reverse the Boolean answer. As observed in [1], this view is obsolete; engineers need quantitative notions on how modified implementations differ. The introduction of such a quantitative notion of refinement, and its consequences for the specification theory, are the subject of this paper.

In the above examples, the transition weights have expressed the time used to perform the associated action. However our formalism is abstract enough to also model other quantitative aspects such as *e.g.* energy consumption or financial aspects. For instance, Figure 5 presents a simple electronic wiper control component for a car, with a normal mode and an optional fast mode. Integer intervals express the allowed energy consumption of each action (using abstract energy units).

Depending on the precise application of our quantitative formalism, there are a few choices which one has to make. One such choice is the precise definition of quantitative refinement, as the way quantitative discrepancies between specifications is measured *e.g.* depends on whether differences accumulate over time or the interest more lies in the maximal individual differences. Another choice is how to combine quantities during structural composition: when modeling *e.g.* energy consumption, they should be added; when modeling timing constraints, some form of conjunction

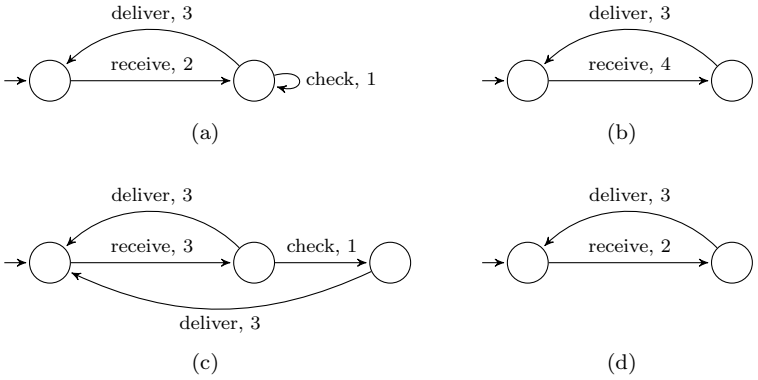


Fig. 4 Four implementations of the simple email system in Figure 3.

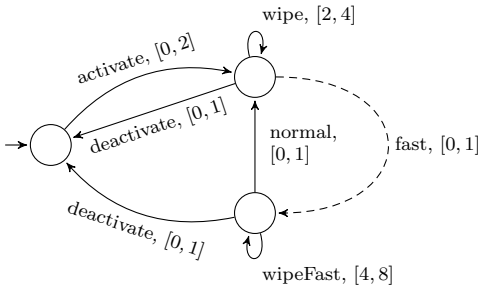


Fig. 5 Weighted modal transition system modeling a simple wiper control component of a car.

should be used. To simplify presentation, we develop the theory in this paper for one specific kind of quantitative refinement and one specific choice of composition; a more general treatment is deferred to future work.

To facilitate quantitative reasoning on specifications and implementations, we introduce a real-valued *distance* between specifications such that perfect refinement corresponds to distance 0, small quantitative discrepancies give rise to small distances, and differences in the discrete control structure correspond to distance ∞. For the examples in Figs. 3 and 4, we will hence deduce the following chain of decreasing distances:

$$\infty = d(I_1, S) > d(I_2, S) > d(I_3, S) > d(I_4, S) = 0$$

Our distance is *discounted* in the sense that behaviors which occur d steps in the future are discounted by a factor λ^d, where λ with $0 < \lambda < 1$ is a fixed discounting factor.

Using a reduction to discounted games [46], we show that this so-called *modal* distance is computable in NP $\cap$ CO-NP. As any specification can be seen as the (generally infinite) set of implementations which are in perfect refinement, we also have a natural notion of so-called *thorough* distance between specifications which is

given by the (Hausdorff) distance between their implementation sets; we show that computing through distances is EXPTIME-hard.

Replacing Boolean refinement by distances has an impact on operations between specifications. As a second contribution of this paper, we propose quantitative versions of structural composition and quotient which inherit the good properties from the Boolean setting. We also propose a new notion of *relaxation* which is inherent to the quantitative framework and allows *e.g.* to calibrate the quotient operator: If the overall specification is too restrictive with respect to a partial implementation to synthesize a meaningful specification of the missing components, the overall specification may be relaxed to facilitate a better quotient.

However, there is no free lunch, and working with distances has a price: some of the properties of logical conjunction and determinization are not preserved in our quantitative setting. More precisely, conjunction is not the greatest lower bound with respect to refinement distance as it is in the Boolean setting, and deterministic overapproximation is too coarse. In fact we show that this is a fundamental limitation of *any* reasonable quantitative specification formalism.

Our final contribution consists of showing that a quantitative interpretation of Hennessy-Milner logic provides a logical characterization which is sound with respect to refinement distance and complete for the disjunction-free fragment.

Related work. The objective of the paper is to propose a new complete quantitative modal specification theory, which exploits a notion of distance between specifications. This distance builds on previous work of some of the authors [26–28, 32, 42, 43]. For the sake of completeness, we briefly put it in perspective with other notions of distances proposed, particularly but not exclusively for probabilistic systems, in recent years. These include [44, 45] which develop a theory of *metric transition systems* and introduce the notion of compact branching, [18, 19, 22, 36] which introduce discounting distances for Markov decision processes, and [13, 20] which generalize these to a game setting.

For a non-probabilistic setting of metric transition systems (different from van Breugel's), notions of discounting linear and branching distances are developed in [1], and an important theoretical contribution is [10] which develops a theory of directed distances, or *hemimetrics* as they have come to be called, and relate completion of hemimetric spaces to Yoneda embeddings (see also [33, 34]). Another, language-based approach to quantitative verification, related to the theory of semiring-weighted automata [23–25], can be found in [11, 14].

Structure of the paper. The paper starts by introducing our quantitative formalism which has weighted transition systems as implementations and weighted modal transition systems as specifications. In Section 3 we introduce the distances we use for quantitative comparison of both implementations and specification, and Section 4 provides complexity results for the computation of these distances. Section 5 is devoted to a formalization of the notion of relaxation which is of great use in quantitative design. In Section 6 we see some inherent limitations of the quantitative approach, and Section 7 shows that structural composition works as expected in the quantitative framework and links relaxation to quotients. Section 8 finishes the paper by providing logical characterizations of refinement distance.

2 Weighted Modal Transition Systems

In this section we present the formalism we use for implementations and specifications. As implementations we choose the model of *weighted transition systems*, *i.e.* labeled transition systems with integer weights at transitions. Specifications both have a *modal* dimension, specifying discrete behavior which *must* be implemented and behavior which *may* be present in implementations, and a *quantitative* dimension, specifying intervals of weights on each transition within are permissible for an implementation.

Let $\mathbb{I} = \left\{[x,y] \;\middle|\; x \in \mathbb{Z} \cup \{-\infty\}, y \in \mathbb{Z} \cup \{\infty\}, x \leq y\right\}$ be the set of closed extended-integer intervals and let Σ be a finite set of actions. Our set of *specification labels* is $\mathrm{K} = \Sigma \times \mathbb{I}$, pairs of actions and intervals. The set of *implementation labels* is defined as $\mathsf{Imp} = \Sigma \times \left\{[x,x] \;\middle|\; x \in \mathbb{Z}\right\} \approx \Sigma \times \mathbb{Z}$. Hence a specification imposes labels and integer intervals which constrain the possible weights of an implementation.

We define a partial order on $\mathbb{I}$ (representing inclusion of intervals) by $[x,y] \sqsubseteq [x',y']$ if $x' \leq x$ and $y \leq y'$, and we extend this order to specification labels by $(a,I) \sqsubseteq (a',I')$ if $a = a'$ and $I \sqsubseteq I'$. The partial order on K is hence a *refinement* order; if $k_1 \sqsubseteq k_2$ for $k_1, k_2 \in \mathrm{K}$, then no more implementation labels are contained in k_1 than in k_2.

Specifications and implementations are defined as follows:

Definition 1 A *weighted modal transition system* (WMTS) is a quadruple $(S, s^0, \dashrightarrow, \longrightarrow)$ consisting of a set of states S with an initial state $s^0 \in S$ and *must* ($\longrightarrow$) and *may* ($\dashrightarrow$) transition relations $\longrightarrow, \dashrightarrow \subseteq S \times \mathrm{K} \times S$ such that for every $(s,k,s') \in \longrightarrow$ there is $(s,\ell,s') \in \dashrightarrow$ where $k \sqsubseteq \ell$. A WMTS is an *implementation* if $\longrightarrow \;=\; \dashrightarrow \subseteq S \times \mathsf{Imp} \times S$.

Note the natural requirement that any required (*must*) behavior is also allowed (*may*) above, and that implementations correspond to standard integer-weighted transition systems, where all optional behavior and positioning in the intervals has been decided on.

A WMTS is *finite* if S and $\dashrightarrow$ (and hence also $\longrightarrow$) are finite sets, and it is *deterministic* if it holds that for any $s \in S$ and $a \in \Sigma$, $\big(s,(a,I_1),t_1\big), \big(s,(a,I_2),t_2\big) \in \dashrightarrow$ imply $I_1 = I_2$ and $t_1 = t_2$. Hence a deterministic specification allows at most one transition under each discrete action from every state. In the rest of the paper we will write $s \overset{k}{\dashrightarrow} s'$ for $(s,k,s') \in \dashrightarrow$ and similarly for $\longrightarrow$, and we will always write $S = (S, s^0, \dashrightarrow, \longrightarrow)$ or $S_i = (S_i, s_i^0, \dashrightarrow_i, \longrightarrow_i)$ for WMTS and $I = (I, i^0, \longrightarrow)$ for implementations. Note that an implementation is just a usual integer-weighted transition system.

Our theory will work with infinite WMTS, though we will require them to be *compactly branching*. This is a natural generalization of the standard requirement on systems to be *finitely branching* which was first used in [45]; see Definition 7 below.

The implementation semantics of a specification is given through modal refinement, as follows:

Definition 2 A *modal refinement* of WMTS S_1, S_2 is a relation $R \subseteq S_1 \times S_2$ such that for any $(s_1, s_2) \in R$

- whenever $s_1 \overset{k_1}{\dashrightarrow}_1 t_1$ for some $k_1 \in \mathrm{K}$, $t_1 \in S_1$, then there exists $s_2 \overset{k_2}{\dashrightarrow}_2 t_2$ for some $k_2 \in \mathrm{K}$, $t_2 \in S_2$, such that $k_1 \sqsubseteq k_2$ and $(t_1, t_2) \in R$,

– whenever $s_2 \xrightarrow{k_2}_2 t_2$ for some $k_2 \in \mathrm{K}$, $t_2 \in S_2$, then there exists $s_1 \xrightarrow{k_1}_1 t_1$ for some $k_1 \in \mathrm{K}$, $t_1 \in S_1$, such that $k_1 \sqsubseteq k_2$ and $(t_1, t_2) \in R$.

We write $S_1 \leq_m S_2$ if there is a modal refinement relation R for which $(s_1^0, s_2^0) \in R$.

Hence in such a modal refinement, behavior which is required in S_2 is also required in S_1, no more behavior is allowed in S_1 than in S_2, and the quantitative requirements in S_1 are refinements of the ones in S_2. The implementation semantics of a specification can then be defined as the set of all implementations which are also refinements:

Definition 3 The *implementation semantics* of a WMTS S is the set $[\![S]\!] = \{I \mid I \leq_m S \text{ and } I \text{ is an implementation}\}$.

This conforms with the intuition developed in the introduction: if $I \in [\![S]\!]$, then any (reachable) behavior $i \xrightarrow{a,x} j$ in I must be allowed by a matching transition $s \xrightarrow{a,[l,r]} t$ in S with $l \leq x \leq r$; correspondingly, any (reachable) required behavior $s \xrightarrow{a,[l,r]} t$ in S must be implemented by a matching transition $i \xrightarrow{a,x} j$ in I with $l \leq x \leq r$.

3 Thorough and Modal Refinement Distances

For the quantitative specification formalism we have introduced in the last section, the standard Boolean notions of satisfaction and refinement are too fragile. To be able to reason not only whether a given quantitative implementation satisfies a given quantitative specification, but also to what extent, we introduce a notion of *distance* between both implementations and specifications.

We recall some terminology. Let $\mathbb{R}_{\geq 0} \cup \{\infty\}$ denote the extended positive reals, let X be a set and $d : X \times X \to \mathbb{R}_{\geq 0} \cup \{\infty\}$. Then d is called

– a *hemimetric* if $d(x,x) = 0$ for all $x \in X$ (indiscernibility of identicals) and $d(x,y) + d(y,z) \geq d(x,z)$ for all $x, y, z \in X$ (triangle inequality);
– a *pseudometric* if it is a hemimetric and additionally, $d(x,y) = d(y,x)$ for all $x, y \in X$ (symmetry);
– a *metric* if it is a pseudometric and additionally, $d(x,y) = 0$ implies $x = y$ for all $x, y \in X$ (identity of indiscernibles)

Note that as our (hemi-, pseudo-)metrics may take the values ∞, some authors will refer to them as *extended* (hemi-, pseudo-)metrics.

The *symmetrization* of a hemimetric d is the pseudometric $\bar{d} : X \times X \to \mathbb{R}_{\geq 0} \cup \{\infty\}$ given by $\bar{d}(x,y) = \max(d(x,y), d(y,x))$; this is the smallest of all pseudometrics d' on X for which $d \leq d'$. Given hemimetrics d on X and d' on another set X', the *product distance* D on $X \times X'$ is defined by $D((x,x'),(y,y')) = d(x,y) + d(x',y')$.

We first define the distance between *implementations*; for this we introduce a distance on implementation labels by

$$d_{\mathsf{Imp}}\big((a_1,x_1),(a_2,x_2)\big) = \begin{cases} \infty & \text{if } a_1 \neq a_2, \\ |x_1 - x_2| & \text{if } a_1 = a_2. \end{cases} \tag{1}$$

In the rest of the paper, let $\lambda \in \mathbb{R}$ with $0 < \lambda < 1$ be a *discounting factor*.

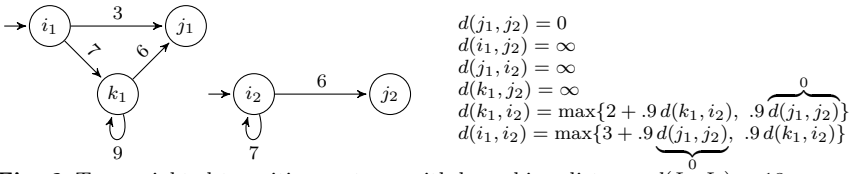


Fig. 6 Two weighted transition systems with branching distance $d(I_1, I_2) = 18$.

Definition 4 The *implementation distance* $d : I_1 \times I_2 \to \mathbb{R}_{\geq 0} \cup \{\infty\}$ between the states of implementations I_1 and I_2 is the least fixed point of the equations

$$d(i_1, i_2) = \max \begin{cases} \sup\limits_{i_1 \xrightarrow{k_1}_1 j_1} \inf\limits_{i_2 \xrightarrow{k_2}_2 j_2} d_{\mathsf{lmp}}(k_1, k_2) + \lambda d(j_1, j_2), \\ \sup\limits_{i_2 \xrightarrow{k_2}_2 j_2} \inf\limits_{i_1 \xrightarrow{k_1}_1 j_1} d_{\mathsf{lmp}}(k_1, k_2) + \lambda d(j_1, j_2). \end{cases}$$

We define $d(I_1, I_2) = d(i_1^0, i_2^0)$.

Lemma 1 *The implementation distance is well-defined, and is a pseudometric.*

Proof Except for the symmetrizing max operation, this is precisely the *accumulating branching distance* from [32,43]. Because of $\lambda < 1$, the equations above define a contraction (with Lipschitz constant λ), so the Banach fixed point theorem (for extended metric spaces) applies. Hence besides the fixed point $d(i_1, i_2) = \infty$, the contraction has at most one other fixed point, *i.e.* there exists indeed a unique least fixed point. We refer to [32] for a more detailed proof.

Symmetry of d is clear, and so is the property $d(i, i) = 0$. The triangle inequality can be shown inductively, *cf.* [32]. □

We remark that besides this accumulating distance, other interesting system distances may be defined depending on the application at hand, *cf.* [43,26,27], but we concentrate here on this distance and leave a generalization to other distances for future work.

Example 1 Consider the two implementations I_1 and I_2 in Figure 6 with a single action (elided for simplicity) and with discounting factor $\lambda = .9$. The equations in the illustration have already been simplified by removing all expressions that evaluate to ∞. What remains to be done is to compute the least fixed point of the equation $d(k_1, i_2) = \max\{2 + .9\, d(k_1, i_2), 0\}$. Clearly 0 is not a fixed point, and solving the equation $d(k_1, i_2) = 2 + .9\, d(k_1, i_2)$ gives $d(k_1, i_2) = 20$. Hence $d(i_1, i_2) = \max\{3, .9 \cdot 20\} = 18$.

Note that the interpretation of the distance between two implementations depends entirely on the application one has in mind; but it can easily be shown [43] that the distance between two implementations is zero if and only if they are *weighted bisimilar*. The intuition is then that the smaller the distance, the closer the implementations are to being bisimilar.

To lift the implementation distance to specifications, we need first to consider the distance between *sets* of implementations. Given implementation sets $\mathcal{I}_1, \mathcal{I}_2$, we define

$$d(\mathcal{I}_1, \mathcal{I}_2) = \sup_{I_1 \in \mathcal{I}_1} \inf_{I_2 \in \mathcal{I}_2} d(I_1, I_2)$$

Note that in case $\mathcal{I}_2$ is finite, we have that for all $\varepsilon \geq 0$, $d(\mathcal{I}_1, \mathcal{I}_2) \leq \varepsilon$ if and only if for each implementation $I_1 \in \mathcal{I}_1$ there exists $I_2 \in \mathcal{I}_2$ for which $d(I_1, I_2) \leq \varepsilon$, hence this is quite a natural notion of distance. Especially, $d(\mathcal{I}_1, \mathcal{I}_2) = 0$ if $\mathcal{I}_1$ is a subset of $\mathcal{I}_2$ up to bisimilarity. For infinite $\mathcal{I}_2$, we have the slightly more complicated property that $d(\mathcal{I}_1, \mathcal{I}_2) \leq \varepsilon$ if and only if for all $\delta > 0$ and any $I_1 \in \mathcal{I}_1$, there is $I_2 \in \mathcal{I}_2$ for which $d(I_1, I_2) \leq \varepsilon + \delta$.

Also remark the similarity of this definition to the one of *Hausdorff distance* between subsets of a metric space, see *e.g.* [3, Sect. 3.16]. Crucially however, our distance is missing the symmetrizing max operation of Hausdorff distance, hence it is *asymmetric*. We may well have $d(\mathcal{I}_1, \mathcal{I}_2) \neq d(\mathcal{I}_2, \mathcal{I}_1)$ and will thus prefer to speak of the distance *from* $\mathcal{I}_1$ *to* $\mathcal{I}_2$ rather than *between* $\mathcal{I}_1$ and $\mathcal{I}_2$. We lift this distance to specifications as follows:

Definition 5 The *thorough refinement distance* between WMTS S_1 and S_2 is defined as $d_t(S_1, S_2) = d\big([\![S_1]\!], [\![S_2]\!]\big)$. We write $S_1 \leq_t^\varepsilon S_2$ if $d_t(S_1, S_2) \leq \varepsilon$.

Lemma 2 *The thorough refinement distance is a hemimetric.*

Proof To show that $d_t(S, S) = 0$ is trivial, and the triangle inequality $d_t(S_1, S_2) + d_T(S_2, S_3) \geq d_t(S_1, S_3)$ follows like in the proof of [3, Lemma 3.72]. □

Indeed this permits us to measure incompatibility of specifications; intuitively, if two specifications have thorough distance ε, then any implementation of the first specification can be matched by an implementation of the second up to ε. Also observe the special case where $S_1 = I_1$ is an implementation: then $d_t(I_1, S_2) = \inf_{I_2 \in [\![S_2]\!]} d(I_1, I_2)$, which measures how close I_1 is to satisfy the specification S_2.

To facilitate computation and comparison of refinement distance, we introduce *modal* refinement distance as an overapproximation. We will show in Theorem 3 below that similarly to the Boolean setting [9], computation of thorough refinement distance is EXPTIME-hard, whereas modal refinement distance is computable in NP ∩ CO-NP.

First we generalize the distance on implementation labels from Equation (1) to specification labels, again using a Hausdorff-type construction. For $k, \ell \in \mathrm{K}$ we define

$$d_{\mathrm{K}}(k, \ell) = \sup_{k' \sqsubseteq k, k' \in \mathsf{Imp}} \inf_{\ell' \sqsubseteq \ell, \ell' \in \mathsf{Imp}} d_{\mathsf{Imp}}(k', \ell').$$

Note that d_{K} is asymmetric, and that $d_{\mathrm{K}}(k, \ell) = 0$ if and only if $k \sqsubseteq \ell$. Also, $d_{\mathrm{K}}(k, \ell) = d_{\mathsf{Imp}}(k, \ell)$ for all $k, \ell \in \mathsf{Imp}$. In more elementary terms, we can express d_{K} as follows:

$$d_{\mathrm{K}}\big((a_1, I_1), (a_2, I_2)\big) = \infty \quad \text{if } a_1 \neq a_2$$
$$d_{\mathrm{K}}\big((a, [x_1, y_1]), (a, [x_2, y_2])\big) = \max(x_2 - x_1, y_1 - y_2, 0)$$

Definition 6 Let S_1, S_2 be WMTS. The *modal refinement distance* $d_m : S_1 \times S_2 \to \mathbb{R}_{\geq 0} \cup \{\infty\}$ from states of S_1 to states of S_2 is the least fixed point of the equations

$$d_m(s_1, s_2) = \max \begin{cases} \sup\limits_{s_1 \xrightarrow{k_1}_1 t_1} \inf\limits_{s_2 \xrightarrow{k_2}_2 t_2} d_{\mathrm{K}}(k_1, k_2) + \lambda d_m(t_1, t_2), \\ \sup\limits_{s_2 \xrightarrow{k_2}_2 t_2} \inf\limits_{s_1 \xrightarrow{k_1}_1 t_1} d_{\mathrm{K}}(k_1, k_2) + \lambda d_m(t_1, t_2). \end{cases}$$

We define $d_m(S_1, S_2) = d_m(s_1^0, s_2^0)$, and we write $S_1 \leq_m^\varepsilon S_2$ if $d_m(S_1, S_2) \leq \varepsilon$.

Lemma 3 *The modal refinement distance is well-defined, and is a hemimetric.*

Proof Like in the proof of Lemma 1, the argument for existence of a unique least fixed point to the defining equations is that they define a contraction. The triangle inequality can again be shown inductively, and the property $d_m(s,s) = 0$ is clear. □

We can now give a precise definition of compact branching; for this we need the notions of symmetrization of a hemimetric and of product distance as defined on page 8.

Definition 7 A WMTS S is said to be *compactly branching* if the sets $\{(s',k) \mid s \overset{k}{\dashrightarrow} s'\}, \{(s',k) \mid s \overset{k}{\longrightarrow} s'\} \subseteq S \times \mathrm{K}$ are compact under the symmetrized product distance $\bar{d}_m \times \bar{d}_{\mathrm{K}}$ for every $s \in S$.

The notion of compact branching was first introduced, for a formalism of *metric transition systems*, in [45]. It is a natural generalization of the standard requirement on transition systems to be *finitely branching* to a distance setting; we will need it for the property that continuous functions defined on the sets $\{(s',k) \mid s \overset{k}{\dashrightarrow} s'\}, \{(s',k) \mid s \overset{k}{\longrightarrow} s'\} \subseteq S \times \mathrm{K}$, for some $s \in S$, attain their infimum and supremum, see Lemma 5 and its proof below.

Thus, we shall henceforth assume all our WMTS to be compactly branching. The following lemma sets up some sufficient conditions for this to be the case.

Lemma 4 *Let S be a WMTS and define the sets $L_i(s,a)$, $U_i(s,a)$ for all $s \in S$, $a \in \Sigma$ and $i \in \{1,2\}$ by*

$$L_1(s,a) = \{l \mid s \overset{a,[l,r]}{\dashrightarrow} s'\}, \qquad L_2(s,a) = \{l \mid s \overset{a,[l,r]}{\longrightarrow} s'\},$$
$$U_1(s,a) = \{r \mid s \overset{a,[l,r]}{\dashrightarrow} s'\}, \qquad U_2(s,a) = \{r \mid s \overset{a,[l,r]}{\longrightarrow} s'\}.$$

Then S is compactly branching if

- *for all $s \in S$, any Cauchy sequence $(s'_n)_{n\in\mathbb{N}}$ in $\{s' \mid s \dashrightarrow s'\}$ (with pseudometric $\bar{d}_m$) has $\lim_{n\to\infty} s_n \in \{s' \mid s \dashrightarrow s'\}$, and likewise, any Cauchy sequence $(s'_n)_{n\in\mathbb{N}}$ in $\{s' \mid s \longrightarrow s'\}$ has $\lim_{n\to\infty} s_n \in \{s' \mid s \longrightarrow s'\}$, and*
- *for all $s \in S$, $a \in \Sigma$ and $i \in \{1,2\}$, L_i is finite or $-\infty \in L_i$, and U_i is finite or $\infty \in U_i$.*

Note that the first property mimicks (and generalizes) standard properties of finite branching and *saturation*, *cf.* [39, Sect. 3.3]. The intuition is that if s has (either *may* or *must*) transitions to a converging sequence of states, then it also has a transition to the limit.

Proof The first condition implies that the sets $\{s' \in S \mid s \dashrightarrow s'\}$ and $\{s' \in S \mid s \longrightarrow s'\}$ are compact in the pseudometric $\bar{d}_m$ for all $s \in S$. By Tychonoff's theorem, products of compact sets are compact, so we need only show that the second condition implies that the sets $\{k \in \mathrm{K} \mid s \overset{k}{\dashrightarrow} s'\}$ and $\{k \in \mathrm{K} \mid s \overset{k}{\longrightarrow} s'\}$ are compact in the pseudometric $\bar{d}_{\mathrm{K}}$ for every $s \in S$.

Let $s \in S$. By definition of d_{K}, the sets $\{k \mid s \overset{k}{\dashrightarrow} s'\}$, $\{k \mid s \overset{k}{\longrightarrow} s'\}$ fall into connected components $\{I \mid s \overset{a,I}{\dashrightarrow} s'\}$, $\{I \mid s \overset{a,I}{\longrightarrow} s'\}$ for all $a \in \Sigma$, hence the former are compact if and only if all the latter are. These in turn are compact if and only

if the four sets L_i, U_i in the lemma, collecting lower and upper bounds of intervals, are compact. Now interval bounds are extended integers, so a sequence in L_i or U_i converges if and only if it is eventually stable or goes towards $-\infty$ or ∞. If the sets are finite, eventual stability is the only option; if they are infinite, they need to include the limit points $-\infty$ (for the lower interval bounds in L_i) or ∞ (for the upper interval bounds in U_i). □

There is a powerful proof technique introduced for branching distances between implementations in [43] that we here extend to modal refinement distance. We define a *modal refinement family* as an $\mathbb{R}_{\geq 0}$-indexed family of relations $R = \{R_\varepsilon \subseteq S_1 \times S_2 \mid \varepsilon \geq 0\}$ such that for any ε and any $(s_1, s_2) \in R_\varepsilon$,

- whenever $s_1 \xrightarrow{k_1} t_1$ for some $k_1 \in \mathrm{K}$, $t_1 \in S_1$, then there exists $s_2 \xrightarrow{k_2} t_2$ for some $k_2 \in \mathrm{K}$, $t_2 \in S_2$, such that $d_{\mathrm{K}}(k_1, k_2) \leq \varepsilon$ and $(t_1, t_2) \in R_{\varepsilon'}$ for some $\varepsilon' \leq \lambda^{-1}\big(\varepsilon - d_{\mathrm{K}}(k_1, k_2)\big)$,
- whenever $s_2 \xrightarrow{k_2} t_2$ for some $k_2 \in \mathrm{K}$, $t_2 \in S_2$, then there exists $s_1 \xrightarrow{k_1} t_1$ for some $k_1 \in \mathrm{K}$, $t_1 \in S_1$, such that $d_{\mathrm{K}}(k_1, k_2) \leq \varepsilon$ and $(t_1, t_2) \in R_{\varepsilon'}$ for some $\varepsilon' \leq \lambda^{-1}\big(\varepsilon - d_{\mathrm{K}}(k_1, k_2)\big)$.

Note that modal refinement families are

- *upward closed* in the sense that $(s_1, s_2) \in R_\varepsilon$ implies that $(s_1, s_2) \in R_{\varepsilon'}$ for all $\varepsilon' \geq \varepsilon$, and
- *downward closed* in the sense that for any set $E \subseteq \mathbb{R}_{\geq 0}$, if $(s_1, s_2) \in R_\varepsilon$ for all $\varepsilon \in E$, then also $(s_1, s_2) \in R_{\inf E}$. This property follows from the assumption that our WMTS are compactly branching.

Following the proof strategy developed in [43] for implementations, we can show the following characterization of modal refinement distance by modal refinement families:

Lemma 5 *$S_1 \leq_m^\varepsilon S_2$ if and only if there is a modal refinement family R with $(s_1^0, s_2^0) \in R_\varepsilon \in R$.*

Proof First, assume that $S_1 \leq_m^\varepsilon S_2$, *i.e.* $d_m(s_1^0, s_2^0) \leq \varepsilon$, and define a relation family $R = \{R_\delta \mid \delta \geq 0\}$ by $R_\delta = \{(s_1, s_2) \in S_1 \times S_2 \mid d_m(s_1, s_2) \leq \delta\}$ for all $\delta \geq 0$, then $(s_1^0, s_2^0) \in R_\varepsilon$ holds by assumption. We show that R is a modal refinement family. Let $(s_1, s_2) \in R_\delta$ for some $\delta \geq 0$, then by definition we know that $d_m(s_1, s_2) \leq \delta$. Assume $s_1 \xrightarrow{k_1}_1 t_1$. From $d_m(s_1, s_2) \leq \delta$ we can infer that

$$\inf_{s_2 \xrightarrow{k_2}_2 t_2} d_{\mathrm{K}}(k_1, k_2) + \lambda d_m(t_1, t_2) \leq \delta.$$

Hence, because S_2 is compactly branching, there exists a may-transition $s_2 \xrightarrow{k_2} t_2$ such that $d_{\mathrm{K}}(k_1, k_2) \leq \delta$ and $d_m(t_1, t_2) \leq \lambda^{-1}(\delta - d_{\mathrm{K}}(k_1, k_2))$. The latter implies that $(t_1, t_2) \in R_{\delta'}$ for some $\delta' \leq \lambda^{-1}(\delta - d_{\mathrm{K}}(k_1, k_2))$ which was to be shown. The argument for the other assertion for must-transitions is symmetric. This proves that there is a modal refinement family R such that $(s_1^0, s_2^0) \in R_\varepsilon \in R$.

For the reverse direction, assume that $(s_1^0, s_2^0) \in R_\varepsilon \in R$ for some modal refinement family $R = \{R_\varepsilon \mid \varepsilon \geq 0\}$. We prove that $(s_1, s_2) \in R_\delta$, for some $\delta \geq 0$, implies $d_m(s_1, s_2) \leq \delta$. The claim $S_1 \leq_m^\varepsilon S_2$ then follows from the assumption $(s_1^0, s_2^0) \in R_\varepsilon$.

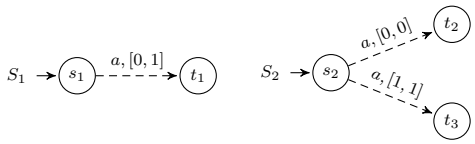


Fig. 7 Incompleteness of modal refinement distance: $d_t(S_1, S_2) = 0$, but $d_m(S_1, S_2) = \infty$.

To this end, observe that the space of functions $\Delta = [S_1 \times S_2 \to \mathbb{R}_{\geq 0} \cup \{\infty\}]$ forms a complete lattice, when the partial order $\leq_\Delta$ is defined such that for $f, f' \in \Delta$, $f \leq_\Delta f'$ iff $f(s_1, s_2) \leq f'(s_1, s_2)$ for all $s_1 \in S_1$, $s_2 \in S_2$. Moreover, since $\max, \sup, \inf$ and $+$ are monotone, the function D defined for all $f \in \Delta$ by

$$D(f) = \max \begin{cases} \sup\limits_{s_1 \xrightarrow{k_1}_1 t_1} \inf\limits_{s_2 \xrightarrow{k_2}_2 t_2} d_{\mathrm{K}}(k_1, k_2) + \lambda f(t_1, t_2), \\ \sup\limits_{s_2 \xrightarrow{k_2}_2 t_2} \inf\limits_{s_1 \xrightarrow{k_1}_1 t_1} d_{\mathrm{K}}(k_1, k_2) + \lambda f(t_1, t_2) \end{cases}$$

is a monotone endofunction on Δ, hence by Tarski's fixed point theorem, D has a least fixed point. Now let us define $h(s_1, s_2) = \inf\{\delta \mid (s_1, s_2) \in R_\delta \in R\}$, and since R_δ is downward closed, we have that $(s_1, s_2) \in R_{h(s_1,s_2)}$. By showing that h is a pre-fixed point of D, *i.e.* that $D(h) \leq_\Delta h$, we get that $(s_1, s_2) \in R_\delta$ implies that $d_m(s_1, s_2) \leq \delta$, since $h(s_1, s_2) \leq \delta$ and $d_m(s_1, s_2) \leq h(s_1, s_2)$.

Since $(s_1, s_2) \in R_{h(s_1,s_2)}$ every $s_1 \xrightarrow{k_1} s_1'$ can be matched by some $s_2 \xrightarrow{k_2} s_2'$ such that $d_{\mathrm{K}}(k_1, k_2) + \lambda\delta' \leq h(s_1, s_2)$ for some δ' where $(s_1', s_2') \in R_{\delta'}$, implying $h(s_1', s_2') \leq \delta'$, but then also $d_{\mathrm{K}}(k_1, k_2) + \lambda h(s_1', s_2') \leq h(s_1, s_2)$. Similarly, every $s_2 \xrightarrow{k_2} s_2'$ has a match $s_1 \xrightarrow{k_1} s_1'$ such that $d_{\mathrm{K}}(k_1, k_2) + \lambda h(s_1', s_2') \leq h(s_1, s_2)$. Hence we have $D(h) \leq_\Delta h$ which was to be shown. □

The next theorems show that modal refinement distance indeed overapproximates thorough refinement distance, and that it is exact for deterministic WMTS. Note that nothing general can be said about the precision of the overapproximation in the nondeterministic case; as an example observe the two specifications in Figure 7 for which $d_t(S_1, S_2) = 0$ but $d_m(S_1, S_2) = \infty$.

Theorem 1 *For WMTS S_1, S_2 we have $d_t(S_1, S_2) \leq d_m(S_1, S_2)$.*

Proof If $d_m(S_1, S_2) = \infty$, we have nothing to prove. Otherwise, let $R = \{R_\varepsilon \subseteq S_1 \times S_2 \mid \varepsilon \geq 0\}$ be a modal refinement family which witnesses $d_m(S_1, S_2)$, *i.e.* such that $(s_1^0, s_2^0) \in R_{d_m(S_1,S_2)}$, and let $I_1 \in [\![S_1]\!]$. We have to expose $I_2 \in [\![S_2]\!]$ for which $d(I_1, I_2) \leq d_m(S_1, S_2)$.

Let $R_1 \subseteq I_1 \times S_1$ be a witness for $I_1 \leq_m S_1$, define $R'_\varepsilon = R_1 \circ R_\varepsilon \subseteq I_1 \times S_2$ for all $\varepsilon \geq 0$, and let $R' = \{R'_\varepsilon \mid \varepsilon \geq 0\}$. The states of $I_2 = (I_2, i_2^0, \mathsf{Imp}, \longrightarrow_{I_2})$ are $I_2 = S_2$ with $i_2^0 = s_2^0$, and the transitions we define as follows:

For any $i_1 \xrightarrow{k_1'}_{I_1} j_1$ and any $s_2 \in S_2$ for which $(i_1, s_2) \in R'_\varepsilon \in R'$ for some ε, we have $s_2 \xrightarrow{k_2}_2 t_2$ in S_2 with $d_{\mathrm{K}}(k_1', k_2) \leq \varepsilon$ and $(j_1, t_2) \in R'_{\varepsilon'} \in R'$ for some $\varepsilon' \leq \lambda^{-1}\big(\varepsilon - d_{\mathrm{K}}(k_1', k_2)\big)$. Write $k_1' = (a_1', x_1')$ and $k_2 = \big(a_2, [x_2, y_2]\big)$, then we must have $a_1' = a_2$. Let

$$x_2' = \begin{cases} x_2 & \text{if } x_1' < x_2, \\ x_1' & \text{if } x_2 \leq x_1' \leq y_2, \\ y_2 & \text{if } x_1' > y_2 \end{cases} \tag{2}$$

and $k_2' = (a_2, x_2')$, and put $s_2 \xrightarrow{k_2'}_{I_2} t_2$ in I_2. Note that

$$d_{\mathrm{K}}(k_1', k_2') = d_{\mathrm{K}}(k_1', k_2). \tag{3}$$

Similarly, for any $s_2 \xrightarrow{k_2}_2 t_2$ in S_2 and any $i_1 \in I_1$ with $(i_1, s_2) \in R_\varepsilon' \in R'$ for some ε, we have $i_1 \xrightarrow{k_1'}_{I_1} j_1$ with $d_{\mathrm{K}}(k_1', k_2) \leq \varepsilon$ and $(j_1, t_2) \in R_{\varepsilon'}' \in R'$ for some $\varepsilon' \leq \lambda^{-1}\big(\varepsilon - d_{\mathrm{K}}(k_1', k_2)\big)$. Write $k_1' = (a_1', x_1')$ and $k_2 = (a_2, [x_2, y_2])$, define x_2' as in (2) and $k_2' = (a_2, x_2')$, and put $s_2 \xrightarrow{k_2'}_{I_2} t_2$ in I_2.

We show that the identity relation $\mathrm{id}_{S_2} = \{(s_2, s_2) \mid s_2 \in S_2\} \subseteq S_2 \times S_2$ witnesses $I_2 \leq_m S_2$. Let first $s_2 \xrightarrow{k_2'}_{I_2} t_2$; we must have used one of the two constructions above for creating this transition. In the first case, we have $s_2 \dashrightarrow_2 t_2$ with $k_2' \sqsubseteq k_2$, and in the second case, we have $s_2 \xrightarrow{k_2}_2 t_2$, hence also $s_2 \dashrightarrow_2 t_2$, with the same property. For a transition $s_2 \xrightarrow{k_2}_2 t_2$ on the other hand, we have introduced $s_2 \xrightarrow{k_2'}_{I_2} t_2$ in the second construction above, with $k_2' \sqsubseteq k_2$.

We also want to show that the family R' is a witness for $d(I_1, I_2) \leq d_m(S_1, S_2)$. We have $(i_1^0, s_2^0) \in R'_{d_m(S_1,S_2)} = R_1 \circ R_{d_m(S_1,S_2)}$, so let $(i_1, s_2) \in R_\varepsilon' \in R'$ for some $\varepsilon \geq 0$. For any $i_1 \xrightarrow{k_1'}_{I_1} j_1$ we have $s_2 \dashrightarrow_2 t_2$ and $s_2 \xrightarrow{k_2'}_{I_2} t_2$ by the first part of our construction above, with $d_{\mathrm{K}}(k_1', k_2') = d_{\mathrm{K}}(k_1', k_2) \leq \varepsilon$ because of (3), and also $(j_1, t_2) \in R_{\varepsilon'}' \in R'$ for some $\varepsilon' \leq \lambda^{-1}\big(\varepsilon - d_{\mathrm{K}}(k_1', k_2)\big)$. For any $s_2 \xrightarrow{k_2'}_{I_2} t_2$, we must have used one of the constructions above to introduce this transition, and both give us $i_1 \xrightarrow{k_1'}_{I_1} j_1$ with $d_{\mathrm{K}}(k_1', k_2') \leq \varepsilon$ and $(j_1, t_2) \in R_{\varepsilon'}' \in R'$ for some $\varepsilon' \leq \lambda^{-1}\big(\varepsilon - d_{\mathrm{K}}(k_1', k_2)\big)$. □

The fact that modal refinement only equals thorough refinement for deterministic specifications is well-known from the theory of modal transition systems [31], and the special case of S_2 deterministic is important, as it can be argued [31] that indeed, deterministic specifications are sufficient for applications.

Theorem 2 *If S_2 is deterministic, then $d_t(S_1, S_2) = d_m(S_1, S_2)$.*

Proof If $d_t(S_1, S_2) = \infty$, we are done by Theorem 1. Otherwise, let $R = \{R_\varepsilon \mid \varepsilon \geq 0\}$ be the smallest relation family for which

- $(s_1^0, s_2^0) \in R_{d_t(S_1,S_2)}$ and
- whenever we have $(s_1, s_2) \in R_\varepsilon \in R$, $s_1 \overset{a,I_1}{\dashrightarrow}_1 t_1$, and $s_2 \overset{a,I_2}{\dashrightarrow}_2 t_2$, then $(t_1, t_2) \in R_{\lambda^{-1}(\varepsilon - d_{\mathrm{K}}((a,I_1),(a,I_2)))}$.

We show below that this definition makes sense (also that $\varepsilon - d_{\mathrm{K}}\big((a, I_1), (a, I_2)\big) \geq 0$ in all cases), and that R is a modal refinement family. We will use the convenient notation (s_1, S_1) for the WMTS S_1 with initial state s_1^0 replaced by s_1, similarly for (s_2, S_2).

We first show inductively that for any pair of states $(s_1, s_2) \in R_\varepsilon \in R$ we have $d_t\big((s_1, S_1), (s_2, S_2)\big) \leq \varepsilon$. This is obviously the case for $s_1 = s_1^0$ and $s_1 = s_2^0$, so assume now that $(s_1, s_2) \in R_\varepsilon \in R$ is such that $d_t\big((s_1, S_1), (s_2, S_2)\big) \leq \varepsilon$ and let $s_1 \overset{a,I_1}{\dashrightarrow}_1 t_1$, $s_2 \overset{a,I_2}{\dashrightarrow}_2 t_2$. Let $(q_1', P_1') \in [\![(t_1, S_1)]\!]$ and $x_1 \in I_1$.

There is an implementation $(p_1, P_1) \in [\![(s_1, S_1)]\!]$ for which $p_1 \xrightarrow{a,x_1} q_1$ and such that $(q_1, P_1) \leq_m (q_1', P_1')$. Now

$$d_t\big((p_1, P_1), (s_2, S_2)\big) \leq d_t\big((p_1, P_1), (s_1, S_1)\big) + d_t\big((s_1, S_1), (s_2, S_2)\big) \leq \varepsilon,$$

hence we must have $s_2 \overset{a_2',I_2'}{\dashrightarrow}_2 t_2'$ with $d_{\mathrm{K}}\big((a,x_1),(a_2',I_2')\big) \leq \varepsilon$. But then $a_2' = a$, hence by determinism of S_2, $I_2 = I_2'$ and $t_2 = t_2'$.

The above considerations hold for any $x_1 \in I_1$, hence $d_{\mathrm{K}}\big((a,I_1),(a,I_2)\big) \leq \varepsilon$. Thus $\varepsilon - d_{\mathrm{K}}\big((a,I_1),(a,I_2)\big) \geq 0$, and the definition of R above is justified. Now let $x_2 \in I_2$ such that $d_{\mathrm{K}}\big((a,x_1),(a,x_2)\big) = d_{\mathrm{K}}\big((a,x_1),(a,I_2)\big)$, then there is an implementation $(p_2,P_2) \in [\![(s_2,S_2)]\!]$ for which $p_2 \overset{a,x_2}{\longrightarrow} q_2$, and

$$\begin{aligned} d\big((q_1',P_1'),(q_2,P_2)\big) &\leq \lambda^{-1}\big(\varepsilon - d_{\mathrm{K}}((a,x_1),(a,x_2))\big) \\ &= \lambda^{-1}\big(\varepsilon - d_{\mathrm{K}}((a,I_1),(a,I_2))\big), \end{aligned}$$

which, as $(q_1',P_1') \in [\![(t_1,S_1)]\!]$ was chosen arbitrarily, entails $d_t\big((s_1,S_1),(s_2,S_2)\big) \leq \lambda^{-1}\big(\varepsilon - d_{\mathrm{K}}((a,I_1),(a,I_2))\big)$.

We are ready to show that R is a refinement family. Let $(s_1,s_2) \in R_\varepsilon \in R$ for some ε, and assume $s_1 \overset{a,I_1}{\dashrightarrow}_1 t_1$. Let $x \in I_1$, then there is an implementation $(p,P^x) \in [\![(s_1,S_1)]\!]$ with a transition $p \overset{m}{\longrightarrow} q$. Now $d_t\big((p,P^x),(s_2,S_2)\big) \leq \varepsilon$, hence we have a transition $s_2 \overset{a,I_2^x}{\dashrightarrow}_2 t_2^x$ with $d_{\mathrm{K}}\big((a,x),(a,I_2^x)\big) \leq \varepsilon$. Also for any other $x' \in I_1$ we have a transition $s_2 \overset{a,I_2^{x'}}{\dashrightarrow}_2 t_2^{x'}$ with $d_{\mathrm{K}}\big((a,x'),(a,I_2^{x'})\big) \leq \varepsilon$, hence by determinism of S_2, $I_2^x = I_2^{x'}$ and $t_2^x = t_2^{x'}$. It follows that there is a unique transition $s_2 \overset{a,I_2}{\dashrightarrow} t_2$, and as $d_{\mathrm{K}}\big((a,x),(a,I_2)\big) \leq \varepsilon$ for all $x \in I_1$, we have $d_{\mathrm{K}}\big((a,I_1),(a,I_2)\big) \leq \varepsilon$, and $(t_1,t_2) \in R_{\lambda^{-1}(\varepsilon - d_{\mathrm{K}}((a,I_1),(a,I_2)))}$ by definition.

Now assume $s_2 \overset{a,I_2}{\longrightarrow}_2 t_2$. Let $(p_1,P_1) \in [\![(s_1,S_1)]\!]$, then we have $(p_2,P_2) \in [\![(s_2,S_2)]\!]$ with $d\big((p_1,P_1),(p_2,P_2)\big) \leq \varepsilon$. Now any $(p_2,P_2) \in [\![(s_2,S_2)]\!]$ has $p_2 \overset{a,x_2}{\longrightarrow} q_2$ with $x_2 \in I_2$, thus there is also $p_1 \overset{a,x_1}{\longrightarrow} q_1$ with $d_{\mathrm{K}}\big((a,x_1),(a,x_2)\big) \leq \varepsilon$ and $d\big((q_1,P_1),(q_2,P_2)\big) \leq \lambda^{-1}\big(\varepsilon - d_{\mathrm{K}}((a,x_1),(a,x_2))\big)$. This in turn implies that $s_1 \overset{a,I_1}{\longrightarrow}_1 t_1$ for some $x_1 \in I_1$. We will be done once we can show $d_{\mathrm{K}}\big((a,I_1),(a,I_2)\big) \leq \varepsilon$, so assume to the contrary that there is $x_1' \in I_1$ with $d_{\mathrm{K}}\big((a,x_1'),(a,I_2)\big) > \varepsilon$. Then there is an implementation $(p_1',P_1') \in [\![(s_1,S_1)]\!]$ with $p_1' \overset{a,x_1'}{\longrightarrow} q_1'$, hence a transition $s_2 \overset{a,I_2'}{\dashrightarrow}_2 t_2'$ with $d_{\mathrm{K}}\big((a,x_1'),(a,I_2')\big) \leq \varepsilon$. But $I_2' = I_2$ by determinism of S_2, a contradiction. □

4 Complexity of Computing Thorough and Modal Refinement Distances

The complexity results in the next theorem show that modal refinement distance can serve as a useful approximation of thorough refinement distance.

Theorem 3 *For finite WMTS S_1, S_2 and $\varepsilon \geq 0$, it is* EXPTIME*-hard to decide whether $S_1 \leq_t^\varepsilon S_2$. The problem whether $S_1 \leq_m^\varepsilon S_2$ is decidable in* NP ∩ CO-NP.

The fact that computing thorough refinement distance is EXPTIME-hard is easy. By [9], deciding thorough refinement for MTS (without weights) is EXPTIME-complete. By translating MTS to WMTS with weight 0 on all transitions, deciding thorough refinement for modal transition systems polynomial-time reduces to deciding whether thorough refinement distance is ≤ 0.

To show an upper bound on the complexity of computing modal refinement distance, we need to introduce *discounted values of weighted games*, *cf.* [46]. A weighted game graph is a finite real-weighted bipartite digraph $(V_1,V_2,\longrightarrow)$, *i.e.* with $V_1 \cap V_2 = \emptyset$ and $\longrightarrow \in (V_1 \times \mathbb{R} \times V_2) \cup (V_2 \times \mathbb{R} \times V_1)$ a finite set of edges. These

are assumed to be non-blocking in the sense that each $v \in V_1 \cup V_2$ has at least one outgoing edge $v \xrightarrow{r} w$ (which is the shorthand for $(v, r, w) \in \longrightarrow$).

A Player-1 strategy in such a weighted game graph is a mapping $\theta_1 : V_1 \to \mathbb{R} \times V_2$ for which $\big(v_1, \theta_1(v_1)\big) \in \longrightarrow$ for each $v_1 \in V_1$. Similarly, a Player-2 strategy is a mapping $\theta_2 : V_2 \to \mathbb{R} \times V_1$ such that $\big(v_2, \theta_2(v_2)\big) \in \longrightarrow$ for each $v_2 \in V_2$. The sets of all Player-1 and Player-2 strategies are denoted Θ_1 and Θ_2, respectively.

Denote by $tgt(e) = w$ the target of an edge $e = (v, r, w) \in \longrightarrow$ and by $wt(e) = r$ its weight. A vertex $v_0 \in V_1$ and a pair $(\theta_1, \theta_2) \in \Theta_1 \times \Theta_2$ of strategies determine a unique infinite sequence $\big(e_j(\theta_1, \theta_2)\big)_{j \geq 0}$ of edges $e_j(\theta_1, \theta_2) \in \longrightarrow$ for which

$$\begin{aligned} e_0(\theta_1, \theta_2) &= \big(v_0, \theta_1(v_0)\big), \\ e_{2j+1}(\theta_1, \theta_2) &= \big(tgt(e_{2j}), \theta_2(tgt(e_{2j}))\big), \\ e_{2j}(\theta_1, \theta_2) &= \big(tgt(e_{2j-1}, \theta_1(tgt(e_{2j-1}))\big). \end{aligned}$$

In other words, the two players alternate to pick edges in $\longrightarrow$ according to their strategies. The *discounted value* of the game $(V_1, V_2, \longrightarrow)$ played from $v_0 \in V_1$ with discounting factor λ, $0 \leq \lambda < 1$, is defined to be

$$p(v_0, \lambda) = \sup_{\theta_1 \in \Theta_1} \inf_{\theta_2 \in \Theta_2} \sum_{j=0}^{\infty} \lambda^j \, wt\big(e_j(\theta_1, \theta_2)\big).$$

We recall the following theorem from [46]; the complexity result is obtained by reduction to simple stochastic games [16].

Lemma 6 ([46]) *The discounted value $p(v_0, \lambda)$ may be computed as the unique fixed point to the equations*

$$p(v, \lambda) = \begin{cases} \max\limits_{v \xrightarrow{r} w} r + \lambda p(w, \lambda) & \text{if } v \in V_1, \\ \min\limits_{v \xrightarrow{r} w} r + \lambda p(w, \lambda) & \text{if } v \in V_2. \end{cases}$$

The decision problem corresponding to computing $p(v_0)$ is contained in NP $\cap$ co-NP.

Next we present a reduction from modal refinement distance of WMTS to discounted values of weighted games, *cf.* [32].

Lemma 7 *For WMTS S_1, S_2 one can construct in polynomial time a weighted game $(V_1, V_2, \longrightarrow)$ with a vertex $v_0 \in V_1$ such that $d_m(S_1, S_2) = p(v_0, \sqrt{\lambda})$.*

Proof Let $V_1 = S_1 \times S_2$, $V_2 = S_1 \times S_2 \times \mathrm{K} \times \{may, must\}$, and define the transitions as follows:

$$\begin{aligned} (s_1, s_2) &\xrightarrow{0} (t_1, s_2, k_1, may) && \text{if} \quad s_1 \xrightarrow{k_1}_1 t_1 \\ (s_1, s_2) &\xrightarrow{0} (s_1, t_2, k_2, must) && \text{if} \quad s_2 \xrightarrow{k_2}_2 t_2 \\ (t_1, s_2, k_1, may) &\xrightarrow{d_{\mathrm{K}}(k_1, k_2)} (t_1, t_2) && \text{if} \quad s_2 \xrightarrow{k_2}_2 t_2 \\ (s_1, t_2, k_2, must) &\xrightarrow{d_{\mathrm{K}}(k_1, k_2)} (t_1, t_2) && \text{if} \quad s_1 \xrightarrow{k_1}_1 t_1 \end{aligned}$$

Setting $v_0 = (s_1^0, s_2^0)$ finishes the construction. □

In [32] it is also shown that conversely, computing discounted values of weighted games may be polynomial-time reduced to computing simulation distance for weighted transition systems, hence we can conclude the following.

Lemma 8 *The decision problem corresponding to computing modal refinement distance for WMTS is polynomial-time equivalent to the decision problem corresponding to computing discounted values of weighted games.*

5 Relaxation

We introduce here a notion of *relaxation* which is specific to the quantitative setting. Intuitively, relaxing a specification means to weaken the quantitative constraints, while the discrete demands on which transitions may or must be present in implementations are kept. A similar notion of *strengthening* may be defined, but we do not use this here.

Definition 8 For WMTS S, S' and $\varepsilon \geq 0$, S' is an *ε-relaxation* of S if $S \leq_m S'$ and $S' \leq_m^{\varepsilon} S$.

Hence the quantitative constraints in S' may be more permissive than the ones in S, but no new discrete behavior may be introduced. Also note that any implementation of S is also an implementation of S', and no implementation of S' is further than ε away from an implementation of S. The following proposition relates specifications to relaxed specifications:

Proposition 1 *If S_1' and S_2' are ε-relaxations of S_1 and S_2, respectively, then $d_m(S_1, S_2) - \varepsilon \leq d_m(S_1, S_2') \leq d_m(S_1, S_2)$ and $d_m(S_1, S_2) \leq d_m(S_1', S_2) \leq d_m(S_1, S_2) + \varepsilon$.*

Proof By the triangle inequality we have

$$
\begin{aligned}
&d_m(S_1, S_2') \leq d_m(S_1, S_2) + d_m(S_2, S_2'),\\
&d_m(S_1, S_2) \leq d_m(S_1, S_2') + d_m(S_2', S_2),\\
&d_m(S_1, S_2) \leq d_m(S_1, S_1') + d_m(S_1', S_2),\\
&d_m(S_1', S_2) \leq d_m(S_1', S_1) + d_m(S_1, S_2).\square
\end{aligned}
$$

On the syntactic level, we can introduce the following *widening* operator which relaxes all quantitative constraints in a systematic manner. We write $I \pm \delta = [x - \delta, y + \delta]$ for an interval $I = [x, y]$ and $\delta \in \mathbb{N}$.

Definition 9 Given $\delta \in \mathbb{N}$, the *δ-widening* of a WMTS S is the WMTS $S^{+\delta}$ with transitions $s \overset{a, I \pm \delta}{\dashrightarrow} t$ in $S^{+\delta}$ for all $s \overset{a, I}{\dashrightarrow} t$ in S, and $s \overset{a, I \pm \delta}{\longrightarrow} t$ in $S^{+\delta}$ for all $s \overset{a, I}{\longrightarrow} t$ in S.

Widening and relaxation are related as follows; note also that as widening is a global operation whereas relaxation may be achieved entirely locally, not all relaxations may be obtained as widenings.

Proposition 2 *The δ-widening of any WMTS S is a $(1 - \lambda)^{-1}\delta$-relaxation.*

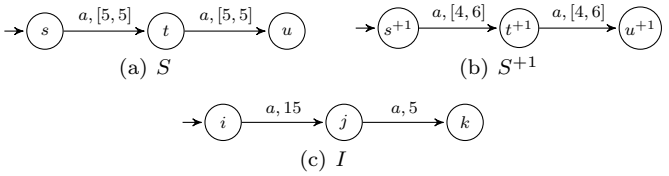


Fig. 8 WMTS S and implementation I for which $I \in [\![S]\!]^{+(1-\lambda)^{-1}\delta}$, for $\delta = 1$ and $\lambda = .9$ (thus $(1-\lambda)^{-1}\delta = 10$), but $I \notin [\![S^{+\delta}]\!]$, so that $[\![S^{+\delta}]\!] \subsetneq [\![S]\!]^{+(1-\lambda)^{-1}\delta}$, even though $S^{+\delta}$ is a $(1-\lambda)^{-1}\delta$-relaxation of S.

Proof For the first claim, the identity relation $\mathrm{id}_S = \{(s,s) \mid s \in S\} \subseteq S \times S$ is a witness for $S \leq_m S^{+\delta}$: if $s \dashrightarrow^{k} t$, then by construction $s \dashrightarrow^{k_2}_{+\delta} t$ with $k \sqsubseteq k_2$, and if $s \xrightarrow{k_2}_{+\delta} t$, then again by construction $s \xrightarrow{k} t$ for some $k \sqsubseteq k_2$.

Now to prove $d_m(S^{+\delta}, S) \leq (1-\lambda)^{-1}\delta$, we define a family of relations $R = \{R_\varepsilon \mid \varepsilon \geq 0\}$ by $R_\varepsilon = \emptyset$ for $\varepsilon < (1-\lambda)^{-1}\delta$ and $R_\varepsilon = \mathrm{id}_S$ for $\varepsilon \geq (1-\lambda)^{-1}\delta$. We show that R is a modal refinement family.

Let $(s,s) \in R_\varepsilon$ for some $\varepsilon \geq (1-\lambda)^{-1}\delta$, and assume $s \dashrightarrow^{k_2}_{+\delta} t$. By construction there is a transition $s \dashrightarrow^{k} t$ with $d_{\mathrm{K}}(k_2, k) \leq \delta \leq \varepsilon$. Now

$$\frac{1}{\lambda}\Big(\varepsilon - d_{\mathrm{K}}(k_2,k)\Big) \geq \frac{1}{\lambda}\Big(\frac{\delta}{1-\lambda} - \delta\Big) = \frac{\delta}{1-\lambda} \geq \varepsilon$$

and $(t,t) \in R_\varepsilon$, which settles this part of the proof. The other direction, starting with a transition $s \xrightarrow{k} t$, is similar. □

There is also an implementation-level notion which corresponds to relaxation:

Definition 10 The *ε-extended implementation semantics*, for $\varepsilon \geq 0$, of a WMTS S is $[\![S]\!]^{+\varepsilon} = \big\{I \;\big|\; I \leq^{\varepsilon}_m S, I \text{ implementation}\big\}$.

Proposition 3 *If S' is an ε-relaxation of S, then $[\![S']\!] \subseteq [\![S]\!]^{+\varepsilon}$.*

Proof If $I \in [\![S']\!]$, then $d_m(I, S') = 0$, hence $d_m(I,S) \leq \varepsilon$ by Proposition 1, which in turn implies that $I \in [\![S]\!]^{+\varepsilon}$. □

The example in Figure 8 shows that there are WMTS S, S' such that S' is an ε-relaxation of S but the inclusion $[\![S']\!] \subseteq [\![S]\!]^{+\varepsilon}$ is strict. Indeed, for $\delta = 1$ and $\lambda = .9$, we have $I \in [\![S]\!]^{+(1-\lambda)^{-1}\delta}$, but $I \notin [\![S^{+\delta}]\!]$.

6 Limitations of the Quantitative Approach

In this section we turn our attention towards some of the standard operators for specification theories; determinization and logical conjunction. In the standard Boolean setting, there is indeed a determinization operator which derives the smallest deterministic overapproximation of a specification, which is useful because it enables checking thorough refinement, *cf.* Theorem 2. Quite surprisingly, we show that in the quantitative setting, there are problems with these notions which do not appear in the Boolean theory. More specifically, we show that there is no determinization operator which always yields a smallest deterministic overapproximation, and there is no conjunction operator which acts as a greatest lower bound.

Theorem 4 *There is no unary operator $\mathcal{D}$ on WMTS for which it holds that*

(4.1) *$\mathcal{D}(S)$ is deterministic for any WMTS S,*
(4.2) *$S \leq_m \mathcal{D}(S)$ for any WMTS S,*
(4.3) *$S \leq_m^\varepsilon D$ implies $\mathcal{D}(S) \leq_m^\varepsilon D$ for any WMTS S, any deterministic WMTS D, and any $\varepsilon \geq 0$.*

Proof There is a determinization operator $\mathcal{D}'$ on WMTS which satisfies Properties (4.1) and (4.2) above and a weaker version of Property (4.3) with $\varepsilon = 0$:

(4.3′) $S \leq_m D$ implies $\mathcal{D}'(S) \leq_m D$ for any WMTS S and any deterministic WMTS D.

This $\mathcal{D}'$ can be defined as follows: For a WMTS $S = (S, s_0, \dashrightarrow, \longrightarrow)$,

$$\mathcal{D}'(S) = \big(\mathcal{P}(S) \setminus \{\emptyset\}, \{s_0\}, \dashrightarrow_d, \longrightarrow_d\big),$$

where $\mathcal{P}(S)$ is the power set of S and the transition relations $\dashrightarrow_d$ and $\longrightarrow_d$ are defined as follows: Let $\mathcal{T} \in (\mathcal{P}(S) \setminus \{\emptyset\})$ be a state in $\mathcal{D}'(S)$. For every maximal, nonempty set $L_a \subseteq \{I \mid \exists s \in \mathcal{T} : s \overset{a,I}{\dashrightarrow}\}$ for some $a \in \Sigma$, we have $\mathcal{T} \overset{a,\bigcup L_a}{\dashrightarrow}_d \mathcal{T}_a$ where $\mathcal{T}_a = \{s' \in S \mid \exists s \in \mathcal{T}, I \in L_a : s \overset{a,I}{\dashrightarrow} s'\}$ and $\bigcup L_a$ is the smallest interval containing all intervals from L_a. If, moreover, for each $s \in \mathcal{T}$ we have $s \overset{a,I}{\longrightarrow} s'$ for some $s' \in \mathcal{T}_a$ and some $I \in L_a$, then $\mathcal{T} \overset{a,\bigcup L_a}{\longrightarrow}_d \mathcal{T}_a$. It is straightforward to prove that $\mathcal{D}'$ satisfies the expected properties.

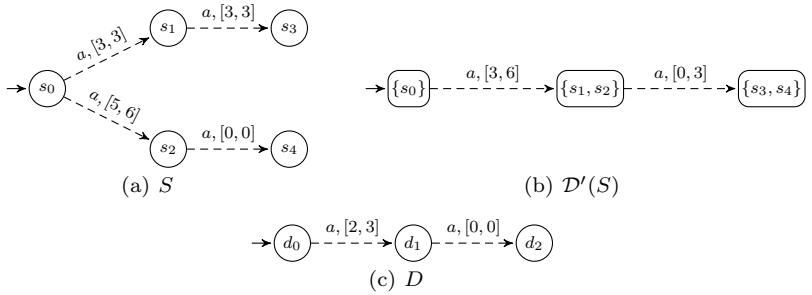
Assume now that there is an operator $\mathcal{D}$ as in the theorem. Then for any WMTS S, $S \leq_m \mathcal{D}'(S)$ and thus $\mathcal{D}(S) \leq_m \mathcal{D}'(S)$ by (4.3), and $S \leq_m \mathcal{D}(S)$ and hence $\mathcal{D}'(S) \leq_m \mathcal{D}(S)$ by (4.3′). We finish the proof by showing that the operator $\mathcal{D}'$ does not satisfy (4.3). The example in Figure 9 shows a WMTS S and a deterministic WMTS D for which $d_m\big(\mathcal{D}'(S), D\big) = 3 + 3\lambda$ and $d_m(S, D) = \max(3, 3\lambda) = 3$, hence $d_m\big(\mathcal{D}'(S), D\big) \not\leq d_m(S, D)$. □

Likewise, the greatest-lower-bound property of logical conjunction in the Boolean setting ensures that the set of implementations of a conjunction of specifications is precisely the intersection of the implementation sets of the two specifications. Conjoining two WMTS naturally involves a partial label conjunction operator $\owedge$. We let $(a_1, I_1) \owedge (a_2, I_2)$ be undefined if $a_1 \neq a_2$, and otherwise

$$\big(a, [x_1, y_1]\big) \owedge \big(a, [x_2, y_2]\big) = \begin{cases} \big(a, [\max(x_1, x_2), \min(y_1, y_2)]\big) & \\ & \text{if } \max(x_1, x_2) \leq \min(y_1, y_2), \\ \text{undefined} & \text{otherwise.} \end{cases}$$

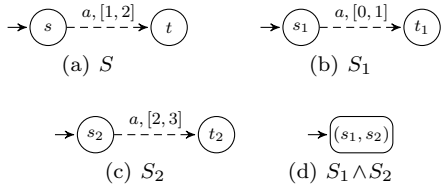
Before we show that such a conjunction operator for WMTS does not exist in general, we need to define a *pruning operator* which removes inconsistent states that naturally arise when conjoining two WMTS. The intuition is that if a WMTS S_1 requires a behavior $s_1 \overset{k_1}{\longrightarrow}_1$ for which there is no may transition $s_2 \overset{k_2}{\dashrightarrow}_2$ such that $k_1 \owedge k_2$ is defined, then the state (s_1, s_2) in the conjunction is *inconsistent* and will have to be pruned away, together with all *must* transitions leading to it. In the definition below, pre* denotes the reflexive, transitive closure of pre.

Definition 11 For a WMTS S, let pre : $2^S \to 2^S$ be given by pre$(B) = \{s \in S \mid s \overset{k}{\longrightarrow} t \in B \text{ for some } k\}$. Let $\lightning \subseteq S$ be a set of *inconsistent* states. If $s^0 \notin \text{pre}^*(\lightning)$, then the *pruning of S w.r.t. $\lightning$* is defined by $\rho^{\lightning}(S) = (S_\rho, s^0, \dashrightarrow_\rho, \longrightarrow_\rho)$ where $S_\rho = S \setminus \text{pre}^*(\lightning)$, $\dashrightarrow_\rho = \dashrightarrow \cap \big(S_\rho \times \mathrm{K} \times S_\rho\big)$ and $\longrightarrow_\rho = \longrightarrow \cap \big(S_\rho \times \mathrm{K} \times S_\rho\big)$.



(a) S (b) $\mathcal{D}'(S)$ (c) D

Fig. 9 Counter-example for Theorem 4: $d_m\big(\mathcal{D}'(S), D\big) = 3+3\lambda$ and $d_m(S, D) = \max(3, 3\lambda) = 3$, hence $d_m\big(\mathcal{D}'(S), D\big) \not\leq d_m(S, D)$.



(a) S (b) S_1 (c) S_2 (d) $S_1 \wedge S_2$

Fig. 10 Counter-example for Theorem 5: $d_m(S, S_1) = d_m(S, S_2) = 1$, but $d_m(S, S_1 \wedge S_2) = \infty$.

Theorem 5 *There is no partial binary operator $\wedge$ on WMTS for which it holds that, for all WMTS S, S_1, S_2 such that S_1 and S_2 are deterministic,*

(5.1) *whenever $S_1 \wedge S_2$ is defined, then $S_1 \wedge S_2 \leq_m S_1$ and $S_1 \wedge S_2 \leq_m S_2$,*
(5.2) *whenever $S \leq_m S_1$ and $S \leq_m S_2$, then $S_1 \wedge S_2$ is defined and $S \leq_m S_1 \wedge S_2$,*
(5.3) *for any $\varepsilon \geq 0$, there exist $\varepsilon_1 \geq 0$ and $\varepsilon_2 \geq 0$ such that if $S_1 \wedge S_2$ is defined, $S \leq_m^{\varepsilon_1} S_1$ and $S \leq_m^{\varepsilon_2} S_2$, then $S \leq_m^{\varepsilon} S_1 \wedge S_2$.*

Proof We follow the same strategy as in the proof of Theorem 4. One can define a partial conjunction operator $\wedge'$ defined for WMTS which satisfies Properties (5.1) and (5.2) as follows: For deterministic WMTS S_1 and S_2, $S_1 \wedge' S_2 = \rho^{\lightning}(S_1 \times S_2, (s_1^0, s_2^0), \dashrightarrow, \longrightarrow)$ where the transition relations $\dashrightarrow$ and $\longrightarrow$ and the set $\lightning \subseteq S_1 \times S_2$ of inconsistent states are defined by the following rules:

$$\frac{s_1 \xrightarrow{k_1} s_1' \quad s_2 \dashrightarrow{}^{k_2} s_2' \quad k_1 \mathbin{⍁} k_2 \text{ defined}}{(s,t) \xrightarrow{k_1 \mathbin{⍁} k_2} (s_1', s_2')} \qquad \frac{s_1 \dashrightarrow{}^{k_1} s_1' \quad s_2 \xrightarrow{k_2} s_2' \quad k_1 \mathbin{⍁} k_2 \text{ defined}}{(s_1,s_2) \xrightarrow{k_1 \mathbin{⍁} k_2} (s_1', s_2')}$$

$$\frac{s_1 \dashrightarrow{}^{k_1} s_1' \quad s_2 \dashrightarrow{}^{k_2} s_2' \quad k_1 \mathbin{⍁} k_2 \text{ defined}}{(s_1,s_2) \dashrightarrow{}^{k_1 \mathbin{⍁} k_2} (s_1', s_2')}$$

$$\frac{s_1 \xrightarrow{k_1} \quad \big(k_1 \mathbin{⍁} k_2 \text{ undefined for any } k_2 \text{ such that } s_2 \dashrightarrow{}^{k_2}\big)}{(s_1,s_2) \in \lightning}$$

$$\frac{s_2 \xrightarrow{k_2} \quad \big(k_1 \mathbin{⍁} k_2 \text{ undefined for any } k_1 \text{ such that } s_1 \dashrightarrow{}^{k_1}\big)}{(s_1,s_2) \in \lightning}$$

Using these properties, one can see that for all deterministic WMTS S_1 and S_2, $S_1 \wedge S_2 \leq_m S_1 \wedge' S_2$ and $S_1 \wedge' S_2 \leq_m S_1 \wedge S_2$. The WMTS depicted in Figure 10

then show that Property (5.3) cannot hold: here, $d_m(S, S_1) = d_m(S, S_2) = 1$, but $d_m(S, S_1 \wedge S_2) = \infty$. $\square$

The counterexamples used in the proofs of Theorems 4 and 5 are quite general and apply to a large class of distances, rather than only to the accumulating distance discussed in this paper. Hence it can be argued that what we have exposed here is a fundamental limitation of any quantitative approach to modal specifications.

7 Structural Composition and Quotient

In this section we show that in our quantitative setting, notions of structural composition and quotient can be defined which obey the properties expected of such operations. In particular, structural composition satisfies independent implementability [2], hence the refinement distance between structural composites can be bounded by the distances between their respective components.

First we define partial synchronization operators $\oplus$ and $\ominus$ on specification labels which will be used for synchronizing transitions. We let $(a_1, I_1) \oplus (a_2, I_2)$ and $(a_1, I_1) \ominus (a_2, I_2)$ be undefined if $a_1 \neq a_2$, and otherwise

$$\big(a, [x_1, y_1]\big) \oplus \big(a, [x_2, y_2]\big) = \big(a, [x_1 + x_2, y_1 + y_2]\big),$$
$$\big(a, [x_1, y_1]\big) \ominus \big(a, [x_2, y_2]\big) = \begin{cases} \text{undefined} & \text{if } x_1 - x_2 > y_1 - y_2, \\ \big(a, [x_1 - x_2, y_1 - y_2]\big) & \text{if } x_1 - x_2 \leq y_1 - y_2. \end{cases}$$

Note that we use CSP-style synchronization, but other types of synchronization can easily be defined. Also, defining $\oplus$ to add intervals (and $\ominus$ to subtract them) is only one particular choice; depending on the application, one can also *e.g.* let $\oplus$ be intersection of intervals or some other operation. It is not difficult to see that these alternative synchronization operators would lead to properties similar to those we show here.

Definition 12 Let S_1 and S_2 be WMTS. The *structural composition* of S_1 and S_2 is $S_1 \| S_2 = \big(S_1 \times S_2, (s_1^0, s_2^0), \mathrm{K}, \dashrightarrow, \longrightarrow\big)$ with transitions given as follows:

$$\frac{s_1 \overset{k_1}{\dashrightarrow}_1 t_1 \quad s_2 \overset{k_2}{\dashrightarrow}_2 t_2 \quad k_1 \oplus k_2 \text{ defined}}{(s_1, s_2) \overset{k_1 \oplus k_2}{\dashrightarrow} (t_1, t_2)} \qquad \frac{s_1 \overset{k_1}{\longrightarrow}_1 t_1 \quad s_2 \overset{k_2}{\longrightarrow}_2 t_2 \quad k_1 \oplus k_2 \text{ defined}}{(s_1, s_2) \overset{k_1 \oplus k_2}{\longrightarrow} (t_1, t_2)}$$

The *quotient* of S_1 by S_2 is $S_1 \setminus\!\setminus S_2 = \rho^{\not\downarrow}\big(S_1 \times S_2 \cup \{u\}, (s_1^0, s_2^0), \mathrm{K}, \dashrightarrow, \longrightarrow\big)$ with transitions and the set of inconsistent states given as follows:

$$\frac{s_1 \overset{k_1}{\dashrightarrow}_1 t_1 \quad s_2 \overset{k_2}{\dashrightarrow}_2 t_2 \quad k_1 \ominus k_2 \text{ defined}}{(s_1, s_2) \overset{k_1 \ominus k_2}{\dashrightarrow} (t_1, t_2)} \qquad \frac{s_1 \overset{k_1}{\longrightarrow}_1 t_1 \quad s_2 \overset{k_2}{\longrightarrow}_2 t_2 \quad k_1 \ominus k_2 \text{ defined}}{(s_1, s_2) \overset{k_1 \ominus k_2}{\longrightarrow} (t_1, t_2)}$$

$$\frac{s_1 \overset{k_1}{\longrightarrow}_1 t_1 \quad \forall s_2 \overset{k_2}{\longrightarrow}_2 t_2 : k_1 \ominus k_2 \text{ undefined}}{(s_1, s_2) \in \not\downarrow}$$

$$\frac{k \in \mathrm{K} \quad \forall s_2 \overset{k_2}{\dashrightarrow}_2 t_2 : k \oplus k_2 \text{ undefined}}{(s_1, s_2) \overset{k}{\dashrightarrow} u} \qquad \frac{k \in \mathrm{K}}{u \overset{k}{\dashrightarrow} u}$$

Note that during the quotient $S_1 \setminus\setminus S_2$ inconsistent states can arise which are then recursively removed using the pruning operator ρ, see Definition 11. After a technical lemma, the next theorem shows that structural composition is well-behaved with respect to modal refinement distance in the sense that the distance between the composed systems is bounded by the distances of the individual systems. Note also the special case in the theorem of $S_1 \leq_m S_2$ and $S_3 \leq_m S_4$ implying $S_1 \| S_3 \leq_m S_2 \| S_4$.

Lemma 9 *For $k_1, k_2, k_3, k_4 \in \mathrm{K}$ with $k_1 \oplus k_3$ and $k_2 \oplus k_4$ defined, we have $d_{\mathrm{K}}(k_1 \oplus k_3, k_2 \oplus k_4) \leq d_{\mathrm{K}}(k_1, k_2) + d_{\mathrm{K}}(k_3, k_4)$.*

Proof Let $k_i = \big(a, [x_i, y_i]\big)$ for all i. We have

$$\begin{aligned} d_{\mathrm{K}}(k_1, k_2) + d_{\mathrm{K}}(k_3, k_4) &= \max(x_2 - x_1, y_1 - y_2, 0) + \max(x_4 - x_3, y_3 - y_4, 0) \\ &\geq \max\big((x_2 - x_1) + (x_4 - x_3), (y_1 - y_2) + (y_3 - y_4), 0\big) \\ &= \max\big((x_2 + x_4) - (x_1 + x_3), (y_1 + y_3) - (y_2 + y_4), 0\big) \\ &= d_{\mathrm{K}}(k_1 \oplus k_3, k_2 \oplus k_4). \quad \square \end{aligned}$$

Theorem 6 (Independent implementability) *For WMTS S_1, S_2, S_3, S_4 we have $d_m\big(S_1 \| S_3, S_2 \| S_4\big) \leq d_m(S_1, S_2) + d_m(S_3, S_4)$.*

Proof If $d_m(S_1, S_2) = \infty$ or $d_m(S_3, S_4) = \infty$, we have nothing to prove. Otherwise, let $R^1 = \{R^1_\varepsilon \subseteq S_1 \times S_2 \mid \varepsilon \geq 0\}$, $R^2 = \{R^2_\varepsilon \subseteq S_3 \times S_4 \mid \varepsilon \geq 0\}$ be witnesses for $d_m(S_1, S_2)$ and $d_m(S_3, S_4)$, respectively; hence $(s^0_1, s^0_2) \in R^1_{d_m(S_1,S_2)} \in R^1$ and $(s^0_3, s^0_4) \in R^2_{d_m(S_3,S_4)} \in R^2$. Define

$$\begin{aligned} R_\varepsilon = \big\{\big((s_1, s_3), (s_2, s_4)\big) \in S_1 \times S_3 \times S_2 \times S_4 \;\big|\; \\ (s_1, s_2) \in R^1_{\varepsilon_1} \in R^1, (s_3, s_4) \in R^2_{\varepsilon_2} \in R^2, \varepsilon_1 + \varepsilon_2 \leq \varepsilon\big\} \end{aligned}$$

for all $\varepsilon \geq 0$ and let $R = \{R_\varepsilon \mid \varepsilon \geq 0\}$. We show that R witnesses $d_m\big(S_1 \| S_3, S_2 \| S_4\big) \leq d_m(S_1, S_2) + d_m(S_3, S_4)$.

We have $\big((s^0_1, s^0_3), (s^0_2, s^0_4)\big) \in R_{d_m(S_1,S_2)+d_m(S_3,S_4)} \in R$. Now let

$$\big((s_1, s_3), (s_2, s_4)\big) \in R_\varepsilon \in R$$

for some ε, then $(s_1, s_2) \in R^1_{\varepsilon_1} \in R^1$ and $(s_3, s_4) \in R^2_{\varepsilon_2} \in R^2$ for some $\varepsilon_1 + \varepsilon_2 \leq \varepsilon$.

Assume $(s_1, s_3) \xrightarrow{k_1 \oplus k_3} (t_1, t_3)$, then $s_1 \xrightarrow{k_1}_1 t_1$ and $s_3 \xrightarrow{k_3}_3 t_3$. By $(s_1, s_2) \in R^1_{\varepsilon_1} \in R^1$, we have $s_2 \xrightarrow{k_2}_2 t_2$ with $d_{\mathrm{K}}(k_1, k_2) \leq \varepsilon_1$ and $(t_1, t_2) \in R^1_{\varepsilon'_1} \in R^1$ for some $\varepsilon'_1 \leq \lambda^{-1}\big(\varepsilon_1 - d_{\mathrm{K}}(k_1, k_2)\big)$; similarly, $s_4 \xrightarrow{k_4}_4 t_4$ with $d_{\mathrm{K}}(k_3, k_4) \leq \varepsilon_2$ and $(t_3, t_4) \in R^2_{\varepsilon'_2} \in R^2$ for some $\varepsilon'_2 \leq \lambda^{-1}\big(\varepsilon_2 - d_{\mathrm{K}}(k_3, k_4)\big)$. Let $\varepsilon' = \varepsilon'_1 + \varepsilon'_2$, then the sum $k_2 \oplus k_4$ is defined, and

$$\begin{aligned} \varepsilon' &\leq \lambda^{-1}\big(\varepsilon_1 + \varepsilon_2 - (d_{\mathrm{K}}(k_1, k_2) + d_{\mathrm{K}}(k_3, k_4))\big) \\ &\leq \lambda^{-1}\big(\varepsilon - d_{\mathrm{K}}(k_1 \oplus k_3, k_2 \oplus k_4)\big) \end{aligned}$$

by Lemma 9. We have $(s_2, s_4) \xrightarrow{k_2 \oplus k_4} (t_2, t_4)$, $d_{\mathrm{K}}(k_1 \oplus k_3, k_2 \oplus k_4) \leq \varepsilon_1 + \varepsilon_2 \leq \varepsilon$ again by Lemma 9, and $\big((t_1, t_3), (t_2, t_4)\big) \in R_{\varepsilon'} \in R$. The reverse direction, starting with a transition $(s_2, s_4) \xrightarrow{k_2 \oplus k_4} (t_2, t_4)$, is similar. $\square$

Again after a technical lemma, the next theorem expresses the fact that quotient is a partial inverse to structural composition. Intuitively, the theorem shows that the quotient $S_1 \mathbin{\backslash\!\backslash} S_2$ is maximal among all WMTS S_3 with respect to any distance $S_2 \| S_3 \leq^{\varepsilon}_{m} S_1$; note the special case of $S_3 \leq_m S_1 \mathbin{\backslash\!\backslash} S_2$ if and only if $S_2 \| S_3 \leq_m S_1$.

Lemma 10 *If $k_1, k_2, k_3 \in \mathrm{K}$ are such that $k_1 \ominus k_2$ and $k_2 \oplus k_3$ are defined, then $d_{\mathrm{K}}(k_3, k_1 \ominus k_2) = d_{\mathrm{K}}(k_2 \oplus k_3, k_1)$.*

Proof We can write $k_i = \big(a, [x_i, y_i]\big)$ for some $a \in \Sigma$. Then

$$
\begin{aligned}
d_{\mathrm{K}}(k_3, k_1 \ominus k_2) &= \max\big((x_1 - x_2) - x_3, y_3 - (y_1 - y_2), 0\big) \\
&= \begin{cases}
x_1 - x_2 - x_3 & \text{if} \quad x_1 - x_2 - x_3 \geq 0, \\
& \qquad\; x_1 - x_2 - x_3 \geq y_3 - y_1 + y_2; \\
y_3 - y_1 + y_2 & \text{if} \quad y_3 - y_1 + y_2 \geq 0, \\
& \qquad\; y_3 - y_1 + y_2 \geq x_1 - x_2 - x_3; \\
0 & \text{if} \quad x_1 - x_2 - x_3 \leq 0, \\
& \qquad\; y_3 - y_1 + y_2 \leq 0.
\end{cases}
\end{aligned}
$$

Similarly,

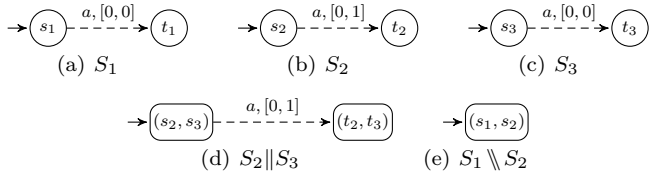
$$
\begin{aligned}
d_{\mathrm{K}}(k_2 \oplus k_3, k_1) &= \max\big(x_1 - (x_2 + x_3), (y_2 + y_3) - y_1, 0\big) \\
&= \begin{cases}
x_1 - x_2 - x_3 & \text{if} \quad x_1 - x_2 - x_3 \geq 0, \\
& \qquad\; x_1 - x_2 - x_3 \geq y_2 + y_3 - y_1; \\
y_2 + y_3 - y_1 & \text{if} \quad y_2 + y_3 - y_1 \geq 0, \\
& \qquad\; y_2 + y_3 - y_1 \geq x_1 - x_2 - x_3; \\
0 & \text{if} \quad x_1 - x_2 - x_3 \leq 0, \\
& \qquad\; y_2 + y_3 - y_1 \leq 0. \qquad \square
\end{cases}
\end{aligned}
$$

Theorem 7 (Soundness and maximality of quotient) *Let S_1, S_2 and S_3 be locally consistent WMTS such that S_2 is deterministic and $S_1 \mathbin{\backslash\!\backslash} S_2$ is defined. If $d_m(S_3, S_1 \mathbin{\backslash\!\backslash} S_2) < \infty$, then $d_m(S_3, S_1 \mathbin{\backslash\!\backslash} S_2) = d_m(S_2 \| S_3, S_1)$.*

Proof To avoid confusion, we write $\dashrightarrow_{\backslash\!\backslash}$ and $\longrightarrow_{\backslash\!\backslash}$ for transitions in $S_1 \mathbin{\backslash\!\backslash} S_2$ and $\dashrightarrow_{\|}$ and $\longrightarrow_{\|}$ for transitions in $S_2 \| S_3$. The inequality $d_m(S_3, S_1 \mathbin{\backslash\!\backslash} S_2) \geq d_m(S_2 \| S_3, S_1)$ is trivial if $d_m(S_2 \| S_3, S_1) = \infty$, so assume the opposite and let $R^1 = \big\{ R^1_\varepsilon \subseteq S_3 \times \big(S_1 \times S_2 \cup \{u\}\big) \;\big|\; \varepsilon \geq 0 \big\}$ be a witness for $d_m(S_3, S_1 \mathbin{\backslash\!\backslash} S_2)$. Define $R^2_\varepsilon = \big\{ \big((s_2, s_3), s_1\big) \;\big|\; \big(s_3, (s_1, s_2)\big) \in R^1_\varepsilon \big\} \subseteq S_2 \times S_3 \times S_1$ for all $\varepsilon \geq 0$, and let $R^2 = \{R^2_\varepsilon \mid \varepsilon \geq 0\}$. Certainly $\big((s^0_2, s^0_3), s^0_1\big) \in R^2_{d_m(S_3, S_1 \mathbin{\backslash\!\backslash} S_2)} \in R^2$, so let now $\big((s_2, s_3), s_1\big) \in R^2_\varepsilon \in R^2$ for some $\varepsilon \geq 0$.

Assume $(s_2, s_3) \overset{k_2 \oplus k_3}{\dashrightarrow}_{\|} (t_2, t_3)$, then also $s_2 \overset{k_2}{\dashrightarrow}_2 t_2$ and $s_3 \overset{k_3}{\dashrightarrow}_3 t_3$. We have $\big(s_3, (s_1, s_2)\big) \in R^1_\varepsilon$, so there is $(s_1, s_2) \overset{k_1 \ominus k'_2}{\dashrightarrow}_{\backslash\!\backslash} (t_1, t'_2)$ for which $d_{\mathrm{K}}(k_3, k_1 \ominus k'_2) = d_{\mathrm{K}}(k'_2 \oplus k_3, k_1) \leq \varepsilon$ and such that $\big(t_3, (t_1, t'_2)\big) \in R^1_{\varepsilon'} \in R^1$, hence $\big((t'_2, t_3), t_1\big) \in R^2_{\varepsilon'} \in R^2$, for some $\varepsilon' \leq \lambda^{-1}\big(\varepsilon - d_{\mathrm{K}}(k'_2 \oplus k_3, k_1)\big)$. By definition of quotient we must have $s_1 \overset{k_1}{\dashrightarrow}_1 t_1$ and $s_2 \overset{k'_2}{\dashrightarrow}_2 t'_2$, and by determinism of S_2, $k'_2 = k_2$ and $t'_2 = t_2$.

Assume $s_1 \overset{k_1}{\longrightarrow}_1 t_1$. We must have a transition $s_2 \overset{k_2}{\longrightarrow}_2 t_2$ for which $k_1 \ominus k_2$ is defined. Hence $(s_1, s_2) \overset{k_1 \ominus k_2}{\longrightarrow}_{\backslash\!\backslash} (t_1, t_2)$. This in turn implies that there is $s_3 \overset{k_3}{\longrightarrow}_3 t_3$ for which $d_{\mathrm{K}}(k_3, k_1 \ominus k_2) = d_{\mathrm{K}}(k_2 \oplus k_3, k_1) \leq \varepsilon$ and such that $\big(t_3, (t_1, t_2)\big) \in R^1_{\varepsilon'} \in R^1$, hence $\big((t_2, t_3), t_1\big) \in R^2_{\varepsilon'} \in R^2$, for some $\varepsilon' \leq \lambda^{-1}\big(\varepsilon - d_{\mathrm{K}}(k_2 \oplus k_3, k_1)\big)$, and by definition of parallel composition, $(s_2, s_3) \overset{k_2 \oplus k_3}{\longrightarrow}_{\|} (t_2, t_3)$.



(a) S_1 (b) S_2 (c) S_3

(d) $S_2 \| S_3$ (e) $S_1 \setminus\!\setminus S_2$

Fig. 11 WMTS for which $d_m(S_2 \| S_3, S_1) \neq d_m(S_3, S_1 \setminus\!\setminus S_2) = \infty$.

To show that $d_m(S_3, S_1 \setminus\!\setminus S_2) \leq d_m(S_2 \| S_3, S_1)$, let $R^2 = \{R^2_\varepsilon \subseteq S_2 \times S_3 \times S_1 \mid \varepsilon \geq 0\}$ be a witness for $d_m(S_2 \| S_3, S_1)$, define $R^1_\varepsilon = \big\{\big(s_3, (s_1, s_2)\big) \;\big|\; \big((s_2, s_3), s_1\big) \in R^2_\varepsilon\big\} \cup \big\{(s_3, u) \;\big|\; s_3 \in S_3\big\}$ for all $\varepsilon \geq 0$, and let $R^1 = \{R^1_\varepsilon \mid \varepsilon \geq 0\}$, then $\big(s^0_3, (s^0_1, s^0_2)\big) \in R^1_{d_m(S_2 \| S_3, S_1)} \in R^1$.

For any $(s_3, u) \in R^1_\varepsilon$ for some $\varepsilon \geq 0$, any transition $s_3 \xrightarrow{k_3}_3 t_3$ can be matched by $u \xrightarrow{k_3}_{\setminus\!\setminus} u$, and then $(t_3, u) \in R^1_0$. Let now $\big(s_3, (s_1, s_2)\big) \in R^1_\varepsilon$ for some $\varepsilon \geq 0$, and assume $s_3 \xrightarrow{k_3}_3 t_3$. If $k_2 \oplus k_3$ is undefined for all transitions $s_2 \xrightarrow{k_2}_2 t_2$, then by definition $(s_1, s_2) \xrightarrow{k_3} u$, and again $(t_3, u) \in R^1_0$. If there is a transition $s_2 \xrightarrow{k_2}_2 t_2$ such that $k_2 \oplus k_3$ is defined, then also $(s_2, s_3) \xrightarrow{k_2 \oplus k_3}_{\|} (t_2, t_3)$. Hence we have $s_1 \xrightarrow{k_1}_1 t_1$ with $d_{\mathrm{K}}(k_2 \oplus k_3, k_1) \leq \varepsilon$, implying that $(s_1, s_2) \xrightarrow{k_1 \ominus k_2}_{\setminus\!\setminus} (t_1, t_2)$. Hence $d_{\mathrm{K}}(k_3, k_1 \ominus k_2) = d_{\mathrm{K}}(k_2 \oplus k_3, k_1) \leq \varepsilon$. Also, $\big((t_2, t_3), t_1\big) \in R^2_{\varepsilon'} \in R^2$, hence $\big(t_3, (t_1, t_2)\big) \in R^1_{\varepsilon'} \in R^1$, for some $\varepsilon' \leq \lambda^{-1}\big(\varepsilon - d_{\mathrm{K}}(k_3, k_1 \ominus k_2)\big)$.

Assume $(s_1, s_2) \xrightarrow{k_1 \ominus k_2}_{\setminus\!\setminus} (t_1, t_2)$, hence we have $s_1 \xrightarrow{k_1}_1 t_1$ and $s_2 \xrightarrow{k_2}_2 t_2$. It follows that $(s_2, s_3) \xrightarrow{k'_2 \oplus k_3}_{\|} (t'_2, t_3)$ with $d_{\mathrm{K}}(k'_2 \oplus k_3, k_1) = d_{\mathrm{K}}(k_3, k_1 \ominus k'_2) \leq \varepsilon$ and such that $\big((t'_2, t_3), t_1\big) \in R^2_{\varepsilon'} \in R^2$, hence $\big(t_3, (t_1, t'_2)\big) \in R^1_{\varepsilon'} \in R^1$, for some $\varepsilon' \leq \lambda^{-1}\big(\varepsilon - d_{\mathrm{K}}(k_3, k_1 \ominus k'_2)\big)$. By definition of parallel composition we must have $s_2 \xrightarrow{k'_2}_2 t'_2$ and $s_3 \xrightarrow{k_3}_3 t_3$, and by determinism of S_2, $k'_2 = k_2$ and $t'_2 = t_2$. □

The example depicted in Figure 11 shows that the condition $d_m(S_3, S_1 \setminus\!\setminus S_2) < \infty$ in Theorem 7 is necessary. Here $d_m(S_2 \| S_3, S_1) = 1$, but $d_m(S_3, S_1 \setminus\!\setminus S_2) = \infty$ because of inconsistency between the transitions $s_1 \xrightarrow{a,[0,0]}_1 t_1$ and $s_2 \xrightarrow{a,[0,1]}_2 t_2$ for which $k_1 \ominus k_2$ is defined.

As a practical application, we notice that *relaxation* as defined in Section 5 can be useful when computing quotients. The quotient construction in Definition 12 introduces inconsistent states (which afterwards are pruned) whenever there is a *must* transition $s_1 \xrightarrow{k_1}_1 s'_1$ such that $k_1 \ominus k_2$ is undefined for all transitions $s_2 \xrightarrow{k_2}_2 s'_2$. Looking at the definition of $\ominus$, we see that this is the case if $k_1 = (a_1, [x_1, y_1])$ and $k_2 = (a_2, [x_2, y_2])$ are such that $a_1 \neq a_2$ or $x_1 - x_2 > y_1 - y_2$. In the first case, the inconsistency is of a *structural* nature and cannot be dealt with; but in the second case, it may be avoided by *enlarging* k_1: decreasing x_1 or increasing y_1 so that now, $x_1 - x_2 \leq y_1 - y_2$.

Enlarging quantitative constraints is exactly the intuition of relaxation, thus in practical cases where we get a quotient $S_1 \setminus\!\setminus S_2$ which is "too inconsistent", we may be able to solve this problem by constructing a suitable ε-relaxation S'_1 of S_1. Theorems 6 and 7 can then be used to ensure that also $S'_1 \setminus\!\setminus S_2$ is a relaxation of $S_1 \setminus\!\setminus S_2$.

8 Logical Characterizations

We now turn our attention to showing that quantitative refinement admits a logical characterization. Our results extend the logical characterization of modal transition systems in [31], by abandoning the usual Boolean interpretation of logical satisfaction, as we did for refinement, and instead interpreting each formula as a map assigning to states a real-valued number denoting the relationship between the property and the state. The logic $\mathcal{L}$ is the smallest set of expressions generated by the following abstract syntax:

$$\phi, \phi_1, \phi_2 := tt \mid ff \mid \langle \ell \rangle \phi \mid [\ell]\phi \mid \phi_1 \wedge \phi_2 \mid \phi_1 \vee \phi_2 \qquad (\ell \in \mathrm{K})$$

As usual, when $\ell = (a, [x_1, x_2])$, writing $\langle \ell \rangle \phi$ means that we insist on implementations exhibiting a transition which reaches a state having property ϕ and is labeled by a and an integer x for which $x_1 \leq x \leq x_2$. Dually, $[\ell]\phi$ restricts the set of implementations to those where every transition labeled with a and an integer in $[x_1, x_2]$ reaches a state with property ϕ.

With this standard (informal) interpretation of logical specifications, implementations which come close to matching the specification are rejected just as much as the truly wrong implementations. Analog to our refinement distance, a quantitative interpretation provides us with continuous judgments on the relationship between a specification S or implementation I and a logical specification ϕ. Defining the semantics of formulae as a map from states to reals, the value of any ϕ for the initial state of implementations determines an order on the applicability of the implementations for the given specification. The semantics of a formula $\phi \in \mathcal{L}$ is a mapping $[\![\phi]\!] : S \to \mathbb{R}_{\geq 0} \cup \{\infty\}$ given inductively, again relative to the discounting factor λ with $0 < \lambda < 1$, as follows:

$$[\![tt]\!]s = 0 \qquad [\![ff]\!]s = \infty$$
$$[\![(\phi_1 \wedge \phi_2)]\!]s = \max([\![\phi_1]\!]s, [\![\phi]\!]_2 s) \qquad [\![(\phi_1 \vee \phi_2)]\!]s = \min([\![\phi_1]\!]s, [\![\phi_2]\!]s)$$
$$[\![\langle \ell \rangle \phi]\!]s = \inf\{d_{\mathrm{K}}(k, \ell) + \lambda [\![\phi]\!]t) \mid s \xrightarrow{k} t, d_{\mathrm{K}}(k, \ell) \neq \infty\}$$
$$[\![[\ell]\phi]\!]s = \sup\{d_{\mathrm{K}}(k, \ell) + \lambda [\![\phi]\!]t \mid s \overset{k}{\dashrightarrow} t, d_{\mathrm{K}}(k, \ell) \neq \infty\}$$

Intuitively, $[\![[\ell]\phi]\!]s$ takes the value of the supremum over all outgoing $s \overset{a,x}{\dashrightarrow} t$ transitions and the respective match with $x \in [x_1, x_2]$ plus the discounted value of the property ϕ for t. Clearly if $[\![[\ell]\phi]\!]s = 0$ then every $s \overset{a,x}{\dashrightarrow} t$ satisfies the property exactly, recovering the standard interpretation. Notice that by evaluating a logical specification ϕ for a WMTS specification S, we get a measure on the set of implementations of S which are not shared by the specification ϕ. The value is 0 if and only if there is a thorough refinement from S to ϕ, *i.e.* if and only if any implementation of S satisfies ϕ.

For a SMTS S we write $[\![\phi]\!]S = [\![\phi]\!]s_0$. The first theorem below expresses the fact that $\mathcal{L}$ is *quantitatively sound* for refinement distance, *i.e.* the value of a formula in a specification is bounded by its value in any other specification together with their distance. Note the special case that $S \leq_m T$ implies $[\![\phi]\!]S \leq [\![\phi]\!]T$.

Theorem 8 *For all $\phi \in \mathcal{L}$ and WMTS S, T, $[\![\phi]\!]S \leq [\![\phi]\!]T + d_m(S, T)$.*

Proof By standard structural induction in ϕ. The claim obviously holds for $\phi = tt$ and $\phi = ff$.

For $\phi = \phi_1 \wedge \phi_2$, the induction hypothesis that $[\![\phi_i]\!]s_1 \leq [\![\phi_i]\!]s_2 + d_m(s_1, s_2)$ for $i = 1, 2$ implies that also $\max([\![\phi_1]\!]s_1, [\![\phi_2]\!]s_1) \leq \max([\![\phi_1]\!]s_2, [\![\phi_2]\!]s_2) + d_m(s_1, s_2)$. Similarly for $\phi = \phi_1 \vee \phi_2$.

For the case $\phi = \langle \ell \rangle \phi'$, if $d_m(s_1, s_2) = \infty$ or if there are no transitions $s_2 \longrightarrow$ the claim is trivial. Let thus $s_2 \xrightarrow{k_2} t_2$, then there exist $s_1 \xrightarrow{k_1} t_1$ with $d_{\mathrm{K}}(k_1, k_2) + \lambda d_m(t_1, t_2) \leq d_m(s_1, s_2)$ (by definition of d_m).

Then $d_{\mathrm{K}}(k_1, \ell) + \lambda[\![\phi']\!]t_1 \leq (d_{\mathrm{K}}(k_1, k_2) + \lambda d_m(t_1, t_2)) + (d_{\mathrm{K}}(k_2, \ell) + \lambda[\![\phi']\!]t_2)$ by induction hypothesis and the triangle inequality for d_{K}, hence $d_{\mathrm{K}}(k_1, \ell) + \lambda[\![\phi']\!]t_1 \leq d_m(s_1, s_2) + d_{\mathrm{K}}(k_2, \ell) + \lambda[\![\phi']\!]t_2$. As $s_2 \xrightarrow{k_2} t_2$ was arbitrary, this entails $\inf\{d_{\mathrm{K}}(k_1, \ell) + \lambda[\![\phi']\!]t_1 \mid s_1 \xrightarrow{k_1} t_1\} \leq \inf\{d_{\mathrm{K}}(k_2, \ell) + \lambda[\![\phi']\!]t_2 \mid s_1 \xrightarrow{k_2} t_2\} + d_m(s_1, s_2)$, which was to be shown.

For the case of $\phi = [\ell]\phi'$ the proof is similar: We have nothing to prove if $d_m(s_1, s_2) = \infty$ or if there are no transitions $s_1 \xrightarrow{k_1} t_1$ with $d_{\mathrm{K}}(k_1, \ell) \neq \infty$, so assume there is such a transition. Then we also have $s_2 \xrightarrow{k_2} t_2$ with $(d_{\mathrm{K}}(k_1, k_2) + \lambda d_m(t_1, t_2)) \leq d_m(s_1, s_2)$, and $d_{\mathrm{K}}(k_1, \ell) + \lambda[\![\phi']\!]t_1 \leq (d_{\mathrm{K}}(k_1, k_2) + \lambda d_m(t_1, t_2)) + d_{\mathrm{K}}(k_2, \ell) + \lambda[\![\phi']\!]t_2 \leq d_m(s_1, s_2) + d_{\mathrm{K}}(k_2, \ell) + \lambda[\![\phi']\!]t_2$. □

The next theorem shows that the disjunction-free fragment of $\mathcal{L}$ is also *quantitatively implementation complete*, *i.e.* the value of any disjunction-free formula in a specification S is bounded above by its value in any implementation of S. Note that disjunction-freeness is a common assumption in this context, *cf.* [31,8].

Theorem 9 *For all disjunction-free $\phi \in \mathcal{L}$ and locally consistent and compactly branching WMTS S, we have $[\![\phi]\!]S = \sup_{I \in [\![S]\!]} [\![\phi]\!]I$.*

Proof Since $d_m(I, S) = 0$ for all $I \in [\![S]\!]$, Theorem 8 entails $[\![\phi]\!]I \leq [\![\phi]\!]S$, hence also $\sup_{I \in [\![S]\!]} [\![\phi]\!]I \leq [\![\phi]\!]S$. To show that $[\![\phi]\!]S \leq \sup_{I \in [\![S]\!]} [\![\phi]\!]I$ we use structural induction on ϕ. If $\phi = tt$, both sides are 0, and if $\phi = ff$, both sides are ∞, so the induction base is clear.

The case $\phi = \phi_1 \wedge \phi_2$ is also clear: By hypothesis, $[\![\phi_1]\!]S \leq \sup_{I \in [\![S]\!]} [\![\phi_1]\!]I$ and similarly for ϕ_2, hence

$$\begin{aligned}[\![\phi]\!]S = \max([\![\phi_1]\!]S, [\![\phi_2]\!]S) &\leq \max(\sup_{I \in [\![S]\!]} [\![\phi_1]\!]I, \sup_{I \in [\![S]\!]} [\![\phi_2]\!]I) \\ &= \sup_{I \in [\![S]\!]} \max([\![\phi_1]\!]I, [\![\phi_2]\!]I).\end{aligned}$$

For the case $\phi = \langle \ell \rangle \phi'$, we are done if $[\![\phi]\!]S = 0$. Otherwise, to conclude that $\sup_{I \in [\![S]\!]} [\![\langle \ell \rangle \phi']\!]I \geq [\![\langle \ell \rangle \phi']\!]S$ we expose an $I \in [\![S]\!]$ for which $\alpha < [\![\phi]\!]I$ for any $\alpha < [\![\phi]\!]S$. For a fixed $\alpha < [\![\phi]\!]S$, start by letting $I = \{i_0\}$ and $\longrightarrow_I = \emptyset$.

Now for each transition $s_0 \xrightarrow{k}_S t$ we have $\alpha < d_{\mathrm{K}}(k, \ell) + \lambda[\![\phi']\!]t$, so (assuming for the moment that $[\![\phi']\!]t \neq 0$) by the density of the reals, there is a number $\alpha'_k < [\![\phi']\!]t$ for which $\alpha < d_{\mathrm{K}}(k, \ell) + \lambda\alpha'_k$. By induction hypothesis, the sub-formula ϕ' satisfies $\sup_{J \in [\![S']\!]} [\![\phi']\!]J = [\![\phi']\!]S'$ for any S', specifically when $S' = (t, S)$ is taken as S with initial state replaced by t. Therefore, and as $\alpha'_k < [\![\phi']\!]t$, there exists a $J \in [\![(t, S)]\!]$ with $\alpha'_k < [\![\phi']\!]J$. Now let $n \in \mathsf{Imp}$ with $n \sqsubseteq k$ be such that $d_{\mathrm{K}}(n, \ell) + \lambda[\![\phi']\!]J = d_{\mathrm{K}}(k, \ell) + \lambda[\![\phi']\!]J$, and add J together with a transition $i_0 \xrightarrow{n}_I j_0$ to I.

In case $[\![\phi']\!]t = 0$, we have $J \in [\![t, S]\!]$ with $[\![\phi']\!]J = 0$, and we can add J together with a transition $i_0 \xrightarrow{n}_I j_0$ to I as above.

For the so-constructed implementation I we have

$$\begin{aligned}[\![\phi]\!]I &= \inf\{d_{\mathrm{K}}(m,\ell) + \lambda[\![\phi']\!]j \mid i_0 \xrightarrow{m}_I j\} \\ &= \inf\{d_{\mathrm{K}}(k,\ell) + \lambda[\![\phi']\!]J \mid s_0 \xrightarrow{k}_S t, J \in [\![t,S]\!], [\![\phi']\!]t = \infty \text{ or } \alpha'_k < [\![\phi']\!]J\} \\ &> \inf(\{d_{\mathrm{K}}(k,\ell) + \lambda\alpha'_k \mid s_0 \xrightarrow{k}_S t\} \cup \{d_{\mathrm{K}}(k,\ell) + \lambda[\![\phi']\!]t\}) \geq \alpha, \end{aligned} \tag{4}$$

the strict inequality in (4) because S is compactly branching.

For the case $\phi = [\ell]\phi'$, let again $\alpha < [\![\phi]\!]S$, and let $I \in [\![S]\!]$ be any implementation. If $d_{\mathrm{K}}(k,\ell) + \lambda[\![\phi']\!]t = \infty$ for all $s_0 \xrightarrow{k}_S t$, then $[\![\phi]\!]S = \sup\emptyset = 0$ and we are done. Otherwise let $s_0 \xrightarrow{k}_S t$ be such that $[\![\phi]\!]S = d_{\mathrm{K}}(k,\ell) + \lambda[\![\phi']\!]t$, which exists because S is compactly branching. Then $\alpha < d_{\mathrm{K}}(k,\ell) + \lambda[\![\phi']\!]t$, so (assuming that $[\![\phi']\!]t \neq 0$) we have $\alpha'_k < [\![\phi']\!]t$ with $d_{\mathrm{K}}(k,\ell) + \lambda\alpha'_k > \alpha$.

Let $J \in [\![t,S]\!]$ such that $\alpha'_k < [\![\phi']\!]J$, let $n \in \mathsf{Imp}$ with $n \sqsubseteq k$ be such that $d_{\mathrm{K}}(n,\ell) + \lambda[\![\phi']\!]J = d_{\mathrm{K}}(k,\ell) + \lambda[\![\phi']\!]J$, and add J together with a transition $i_0 \xrightarrow{n}_I j_0$ to I. Then

$$\begin{aligned}[\![\phi]\!]I &= \sup\{d_{\mathrm{K}}(m,\ell) + \lambda[\![\phi']\!]n \mid i_0 \xrightarrow{m}_I j\} \\ &\geq d_{\mathrm{K}}(n,\ell) + \lambda[\![\phi']\!]J = d_{\mathrm{K}}(k,\ell) + \lambda[\![\phi']\!]J \geq F(k,\ell,\alpha'_k) > \alpha. \end{aligned}$$

In case $[\![\phi']\!]t = 0$ instead, we again take some $J \in [\![t,S]\!]$, and then $[\![\phi]\!]I \geq d_{\mathrm{K}}(k,\ell) + \lambda[\![\phi']\!]t > \alpha$. □

Other notions of completeness (see e.g. [7]) are subject of future work.

9 Conclusion and Further Work

We have shown in this paper that within the quantitative specification framework of weighted modal transition systems, refinement and implementation distances provide a useful tool for robust compositional reasoning. Note that these distances permit us not only to reason about differences between implementations and from implementations to specifications, but they also provide a means by which we can compare specifications directly at the abstract level.

We have shown that for some of the ingredients of our specification theory, namely structural composition and quotient, our formalism is a conservative extension of the standard Boolean notions. We have also noted however, that for determinization and logical conjunction, the properties of the Boolean notions are not preserved, and that this is a fundamental limitation of any reasonable quantitative specification theory. The precise practical implications of this for the applicability of our quantitative specification framework, and perhaps how to circumvent these limitations, are subject to future work.

References

1. Luca de Alfaro, Marco Faella, and Mariëlle Stoelinga. Linear and branching system metrics. *IEEE Transactions on Software Engineering*, 35(2):258–273, 2009.

2. Luca de Alfaro and Thomas Henzinger. Interface-based design. In Manfred Broy, Johannes Grünbauer, David Harel, and Tony Hoare, editors, *Engineering Theories of Software Intensive Systems*, volume 195 of *NATO Science Series II: Mathematics, Physics and Chemistry*, pages 83–104. Springer-Verlag, 2005.
3. Charalambos D. Aliprantis and Kim C. Border. *Infinite Dimensional Analysis: A Hitchhiker's Guide*. Springer-Verlag, 2007.
4. Adam Antonik, Michael Huth, Kim G. Larsen, Ulrik Nyman, and Andrzej Wąsowski. 20 years of modal and mixed specifications. *Bulletin of the EATCS*, 95:94–129, 2008.
5. Ananda Basu, Saddek Bensalem, Marius Bozga, Benoît Caillaud, Benoît Delahaye, and Axel Legay. Statistical abstraction and model-checking of large heterogeneous systems. In John Hatcliff and Elena Zucca, editors, *FMOODS/FORTE*, volume 6117 of *Lecture Notes in Computer Science*, pages 32–46. Springer-Verlag, 2010.
6. Sebastian S. Bauer, Uli Fahrenberg, Line Juhl, Kim G. Larsen, Axel Legay, and Claus Thrane. Quantitative refinement for weighted modal transition systems. In Filip Murlak and Piotr Sankowski, editors, *MFCS*, volume 6907 of *Lecture Notes in Computer Science*, pages 60–71. Springer-Verlag, 2011.
7. Sebastian S. Bauer, Line Juhl, Kim G. Larsen, Axel Legay, and Jiří Srba. Extending modal transition systems with structured labels. *Mathematical Structures in Computer Science*, 2012. To be published.
8. Nikola Beneš, Ivana Černá, and Jan Křetínský. Modal transition systems: Composition and LTL model checking. In Tevfik Bultan and Pao-Ann Hsiung, editors, *ATVA*, volume 6996 of *Lecture Notes in Computer Science*, pages 228–242. Springer-Verlag, 2011.
9. Nikola Beneš, Jan Křetínský, Kim G. Larsen, and Jiří Srba. Checking thorough refinement on modal transition systems is EXPTIME-complete. In Martin Leucker and Carroll Morgan, editors, *ICTAC*, volume 5684 of *Lecture Notes in Computer Science*, pages 112–126. Springer-Verlag, 2009.
10. Marcello M. Bonsangue, Franck van Breugel, and Jan J. M. M. Rutten. Generalized metric spaces: Completion, topology, and powerdomains via the Yoneda embedding. *Theoretical Computer Science*, 193(1-2):1–51, 1998.
11. Pavol Černý, Thomas A. Henzinger, and Arjun Radhakrishna. Simulation distances. *Theoretical Computer Science*, 413(1):21–35, 2012.
12. Arindam Chakrabarti, Luca de Alfaro, Thomas A. Henzinger, and Freddy Y. C. Mang. Synchronous and bidirectional component interfaces. In Ed Brinksma and Kim G. Larsen, editors, *CAV*, volume 2404 of *Lecture Notes in Computer Science*, pages 414–427. Springer-Verlag, 2002.
13. Krishnendu Chatterjee, Luca de Alfaro, Rupak Majumdar, and Vishwanath Raman. Algorithms for game metrics. *Logical Methods in Computer Science*, 6(3), 2010.
14. Krishnendu Chatterjee, Laurent Doyen, and Thomas A. Henzinger. Expressiveness and closure properties for quantitative languages. *Logical Methods in Computer Science*, 6(3), 2010.
15. STREP COMBEST (COMponent-Based Embedded Systems design Techniques). `http://www.combest.eu/home/`.
16. Anne Condon. The complexity of stochastic games. *Information and Computation*, 96(2):203–224, 1992.
17. Alexandre David, Kim G. Larsen, Axel Legay, Ulrik Nyman, and Andrzej Wąsowski. Timed I/O automata: A complete specification theory for real-time systems. In Karl Henrik Johansson and Wang Yi, editors, *HSCC*, pages 91–100. ACM, 2010.
18. Luca de Alfaro. Quantitative verification and control via the mu-calculus. In Roberto M. Amadio and Denis Lugiez, editors, *CONCUR*, volume 2761 of *Lecture Notes in Computer Science*, pages 102–126. Springer-Verlag, 2003.
19. Luca de Alfaro, Thomas A. Henzinger, and Rupak Majumdar. Discounting the future in systems theory. In Jos C. M. Baeten, Jan Karel Lenstra, Joachim Parrow, and Gerhard J. Woeginger, editors, *ICALP*, volume 2719 of *Lecture Notes in Computer Science*, pages 1022–1037. Springer-Verlag, 2003.
20. Luca de Alfaro, Rupak Majumdar, Vishwanath Raman, and Mariëlle Stoelinga. Game refinement relations and metrics. *Logical Methods in Computer Science*, 4(3), 2008.
21. Benoît Delahaye. *Modular Specification and Compositional Analysis of Stochastic Systems*. PhD thesis, Université de Rennes 1, 2010.
22. Josee Desharnais, Vineet Gupta, Radha Jagadeesan, and Prakash Panangaden. Metrics for labelled Markov processes. *Theoretical Computer Science*, 318(3):323–354, 2004.

23. Manfred Droste and Paul Gastin. Weighted automata and weighted logics. *Theoretical Computer Science*, 380(1-2):69–86, 2007.
24. Manfred Droste, Werner Kuich, and Heiko Vogler. *Handbook of Weighted Automata*. EATCS Monographs in Theoretical Computer Science. Springer, 2009.
25. Manfred Droste and George Rahonis. Weighted automata and weighted logics with discounting. *Theoretical Computer Science*, 410(37):3481–3494, 2009.
26. Uli Fahrenberg, Kim G. Larsen, and Claus Thrane. A quantitative characterization of weighted Kripke structures in temporal logic. *Computing and Informatics*, 29(6+):1311–1324, 2010.
27. Uli Fahrenberg, Axel Legay, and Claus Thrane. The quantitative linear-time–branching-time spectrum. In Supratik Chakraborty and Amit Kumar, editors, *FSTTCS*, volume 13 of *LIPIcs*, pages 103–114. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2011.
28. Uli Fahrenberg, Claus Thrane, and Kim G. Larsen. Distances for weighted transition systems: Games and properties. In Mieke Massink and Gethin Norman, editors, *QAPL*, volume 57 of *Electronic Proceedings in Theoretical Computer Science*, pages 134–147, 2011.
29. Robert J. Hall. Feature interactions in electronic mail. In Muffy Calder and Evan H. Magill, editors, *FIW*, pages 67–82. IOS Press, 2000.
30. Line Juhl, Kim G. Larsen, and Jiří Srba. Modal transition systems with weight intervals. *Journal of Logic and Algebraic Programming*, 81(4):408–421, 2012.
31. Kim G. Larsen. Modal specifications. In Joseph Sifakis, editor, *Automatic Verification Methods for Finite State Systems*, volume 407 of *Lecture Notes in Computer Science*, pages 232–246. Springer-Verlag, 1989.
32. Kim G. Larsen, Uli Fahrenberg, and Claus Thrane. Metrics for weighted transition systems: Axiomatization and complexity. *Theoretical Computer Science*, 412(28):3358–3369, 2011.
33. F. William Lawvere. Metric spaces, generalized logic, and closed categories. *Rendiconti del seminario matématico e fisico di Milano*, XLIII:135–166, 1973.
34. F. William Lawvere. Taking categories seriously. *Revista Colombiana de Matemâticas*, XX:147–178, 1986.
35. Nancy Lynch and Mark R. Tuttle. An introduction to input/output automata. *CWI-Quarterly*, 2(3), 1989.
36. Rupak Majumdar. *Symbolic Algorithms for Verification and Control*. PhD thesis, University of California, Berkeley, 2003.
37. Ulrik Nyman. *Modal Transition Systems as the Basis for Interface Theories and Product Lines*. PhD thesis, Aalborg University, 2008.
38. Jean-Baptiste Raclet. Residual for component specifications. *Electronic Notes in Theoretical Computer Science*, 215:93–110, 2008.
39. Davide Sangiorgi. On the origins of bisimulation and coinduction. *ACM Trans. Program. Lang. Syst.*, 31(4), 2009.
40. Joseph Sifakis. A vision for computer science – the system perspective. *Central European Journal of Computer Science*, 1(1):108–116, 2011.
41. SPEEDS (SPEculative and Exploratory Design in Systems Engineering). `http://www.speeds.eu.com`.
42. Claus Thrane. *Quantitative Models and Analysis For Reactive Systems*. PhD thesis, Aalborg University, 2011.
43. Claus Thrane, Uli Fahrenberg, and Kim G. Larsen. Quantitative simulations of weighted transition systems. *Journal of Logic and Algebraic Programming*, 79(7):689–703, 2010.
44. Franck van Breugel. *Topological Models in Comparative Semantics*. PhD thesis, Vrije Universiteit, Amsterdam, 1994.
45. Franck van Breugel. A theory of metric labelled transition systems. *Annals of the New York Academy of Sciences*, 806(1):69–87, 1996.
46. Uri Zwick and Mike Paterson. The complexity of mean payoff games. In Ding-Zhu Du and Ming Li, editors, *COCOON*, volume 959 of *Lecture Notes in Computer Science*, pages 1–10. Springer-Verlag, 1995.