Theoretical Computer Science 413 (2012) 73–86

Contents lists available at SciVerse ScienceDirect

Theoretical Computer Science

journal homepage: www.elsevier.com/locate/tcs

Approximating Markovian testing equivalence

Alessandro Aldini *

University of Urbino, Italy

ARTICLE INFO

Keywords:
Markovian testing equivalence
Process algebra
Approximate equivalence

ABSTRACT

Several approaches have been proposed to relax behavioral equivalences for fine-grain models including probabilities and time. All of them face two problems behind the notion of approximation, i.e., the lack of transitivity and the efficiency of the verification algorithm. While the typical equivalence under approximation is bisimulation, we present a relaxation of Markovian testing equivalence in a process algebraic framework. In this coarser setting, we show that it is particularly intuitive to manage separately three different dimensions of the approximation – execution time, event probability, and observed behavior – by illustrating in each case, results concerning the two problems mentioned above. Finally, a unified definition combining the three orthogonal aspects is provided in order to favor trade-off analyses.

1. Introduction

The comparison of system models through equivalence checking [7] is a well-known approach to the analysis of systems of practical interest, ranging from the model-based verification of software architectures (see, e.g., [6]) to the analysis of noninterference based dependability properties (see, e.g., [2]). However, in practice, perfect equivalence is usually hard to achieve, e.g., because the models to compare are specified at different abstraction levels, or else they describe alternative implementations of the same ideal system. In order to evaluate how much the behaviors of these models fit the same properties, quantitative aspects come into play in such a way that the comparison can result in numbers giving a flavor of the degree of similarity.

Very often, the considered quantitative aspects are expressed in terms of probability distributions and/or temporal durations. As a consequence, fine-grain notions of behavioral equivalences – typically bisimulation – are somehow relaxed in order to measure the similarity between models. It is also possible to compare quantitatively models that are purely functional. This can be done, e.g., through a benchmark of testing scenarios and some kind of mathematical function that estimates the capability of the models in fitting the behaviors offered by the tests forming the benchmark.

In this paper, we merge these different approaches in a formal framework where models are expressed in terms of Markovian process algebra and the comparison is based on the Markovian testing equivalence semantics. This is a novelty in the field of approximate behavioral equivalences. The reason for this choice is that this framework provides, in a natural and explicit way, ingredients for the definition of the degree of similarity with respect to three orthogonal aspects: time, probability, and observed behavior.

To give some intuitive insights, testing equivalence for Markovian processes is based on the comparison of the probabilities of observing successful test-driven computations (i.e., they somehow "pass" tests) that satisfy temporal constraints concerning the amount of time needed to pass these tests. Therefore, by relaxing, in turn, each of these parameters – durations associated with specific computations, probability distributions of these computations, and kind of tests elucidating them – we easily obtain different definitions of approximation under the three considered dimensions.

* Tel.: +39 07224475.
E-mail addresses: alessandro.aldini@uniurb.it, aldini@sti.uniurb.it.

0304-3975/$ – see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2011.07.019

Formally, we first observe that the temporal behavior of a test-driven computation is described in terms of average sojourn times in the states traversed by the computation. Hence, relaxing time in this setting amounts to matching computations with stepwise average durations that are similar up to a threshold ϵ.

Secondly, the probabilistic behavior of a test-driven computation is defined by the product of the execution probabilities of the transitions forming the computation. The relaxation of this aspect consists of checking whether the difference between the probabilities of corresponding computations is confined by a threshold ν.

Thirdly, the observed behavior of a test-driven computation is specified explicitly by the test that guides the system execution. The family of tests with respect to which the comparison is conducted represents the benchmark of observations parameterizing the analysis. For instance, a family of tests may be formally characterized in terms of a modal logic formula that must be satisfied by each test. Then, introducing approximation at this level consists of matching computations that pass different but behaviorally similar tests in a given family, i.e. tests are used as a benchmark of testing scenarios based on which we evaluate the similarity between different models. Such an estimation relies on two fitness functions, called precision and recall, that are used to measure the distance between behaviorally similar tests.

For each approach, we investigate separately the properties that are preserved by the distance function resulting from the approximation definition, e.g. conservativeness with respect to the equivalence relation, and we verify the complexity of the verification algorithm. This separation of concerns favors the comprehension of the theoretical constraints imposed by every kind of approximation. On the other hand, we also present a unifying definition joining the three levels of approximation. This is done in order to facilitate the study of possible trade-offs among them, as typically required, e.g., in performability analysis of real-world cases.

The remainder of the paper, which is a full and revised version of [1], is organized as follows. In Section 2, we set the background for the definition of Markovian process algebra, in which to formalize the different notions of approximation. In Section 3, we present the three relaxations surveyed above – which result in a unifying definition formalized in Section 3.4 – together with the related properties. In all these sections, we employ as a running example a client–server system based on a multi-core server. Finally, in Section 4 we draw some conclusions and a comparison with related work.

2. Markovian process algebra

In this section, we recall the framework in which to formalize the notion of similarity: first, we describe a Markovian process calculus that we call MPC; then we provide the definition of Markovian testing equivalence. For a complete survey of the main results concerning these topics, the interested reader is referred to [3].

2.1. Syntax and semantics

In MPC, timing aspects are represented through durational actions of the form $<a, \lambda>$, where $a \in \mathit{Name}$ denotes the action name, while the action duration is exponentially distributed with rate $\lambda \in \mathbb{R}_{>0}$, meaning that the average duration of the action is equal to the reciprocal of its rate, i.e. $1/\lambda$.

Moreover, we have actions of the form $<a, *_w>$, which are unspecified from the temporal standpoint, i.e. they are not associated with any duration. The parameter $w \in \mathbb{R}_{>0}$ is a weight that, as we will see, comes into play when a choice among different nondurational actions with the same name occurs.

The set of process terms of MPC is generated by the following syntax:

$$P ::= \underline{0} \mid <a, \tilde{\lambda}>.P \mid P + P \mid P \parallel_S P \mid A$$

where $a \in \mathit{Name}$ (among the action names we consider also the distinguished symbol τ denoting the internal, unobservable action), $\tilde{\lambda} \in \mathit{Rate} = \mathbb{R}_{>0} \cup \{*_w \mid w \in \mathbb{R}_{>0}\}$, $S \subseteq \mathit{Name}_{\mathrm{v}} = \mathit{Name} - \{\tau\}$ denotes the synchronization set, and A is a process constant defined by the possibly recursive equation $A \triangleq P$. We denote with $\mathit{Act} = \mathit{Name} \times \mathit{Rate}$ the set of actions of MPC and with $\mathbb{P}$ the set of closed and guarded process terms of MPC. We do not consider other static operators (hiding, restriction, and relabeling), which however would not alter the results of this paper. An informal presentation of the algebraic operators is as follows.

The inactive process $\underline{0}$ represents a terminated process.

The action prefix operator $<a, \_>.P$ represents a process performing the durational/nondurational action with name a and then behaving as P.

The alternative composition operator encodes choice. If several durational actions can be performed, then the race policy is adopted: the execution probability of each durational action is proportional to its rate and the average sojourn time associated with the related process term is exponentially distributed with rate given by the sum of the rates of the durational actions enabled by the term.

On the other hand, when several nondurational actions with the same name are enabled, the preselection policy is adopted: each such actions is given an execution probability equal to its action weight divided by the sum of the weights of all the actions with the same name enabled by the term.

All the other choices, among nondurational actions with different names or between nondurational actions and exponentially timed actions, are nondeterministic.

The parallel composition operator is inspired by CSP [14] and relies on the following asymmetric synchronization policy. A durational action can synchronize only with a nondurational action with the same visible name. The synchronization proceeds as follows. First, a durational action is chosen according to the race policy. Second, a nondurational action with the same name of the proposed durational action is chosen probabilistically on the basis of the preselection policy. Third, these two actions synchronize and the resulting rate is given by the rate of the durational action multiplied by the execution probability of the nondurational action.

Formally, the semantic rules for MPC are defined in the classical operational style as follows:

$$<a, \lambda>.P \xrightarrow{a,\lambda} P \qquad <a, *_w>.P \xrightarrow{a,*_w} P$$

$$\frac{P_1 \xrightarrow{a,\tilde{\lambda}} P'}{P_1 + P_2 \xrightarrow{a,\tilde{\lambda}} P'} \qquad \frac{P_2 \xrightarrow{a,\tilde{\lambda}} P'}{P_1 + P_2 \xrightarrow{a,\tilde{\lambda}} P'}$$

$$\frac{P_1 \xrightarrow{a,\tilde{\lambda}} P_1' \quad a \notin S}{P_1 \parallel_S P_2 \xrightarrow{a,\tilde{\lambda}} P_1' \parallel_S P_2} \qquad \frac{P_2 \xrightarrow{a,\tilde{\lambda}} P_2' \quad a \notin S}{P_1 \parallel_S P_2 \xrightarrow{a,\tilde{\lambda}} P_1 \parallel_S P_2'}$$

$$\frac{P_1 \xrightarrow{a,\lambda} P_1' \quad P_2 \xrightarrow{a,*_w} P_2' \quad a \in S}{P_1 \parallel_S P_2 \xrightarrow{a,\lambda \cdot \frac{w}{weight(P_2,a)}} P_1' \parallel_S P_2'}$$

$$\frac{P_1 \xrightarrow{a,*_w} P_1' \quad P_2 \xrightarrow{a,\lambda}_{\mathrm{M}} P_2' \quad a \in S}{P_1 \parallel_S P_2 \xrightarrow{a,\lambda \cdot \frac{w}{weight(P_1,a)}} P_1' \parallel_S P_2'}$$

$$\frac{P_1 \xrightarrow{a,*_{w_1}} P_1' \quad P_2 \xrightarrow{a,*_{w_2}} P_2' \quad a \in S}{P_1 \parallel_S P_2 \xrightarrow{a,*_{norm(w_1,w_2,a,P_1,P_2)}} P_1' \parallel_S P_2'}$$

$$\frac{A \triangleq P \qquad P \xrightarrow{a,\tilde{\lambda}} P'}{A \xrightarrow{a,\tilde{\lambda}} P'}$$

where: $weight(P, a) = \sum\{\!\!\{ w \in \mathbb{R}_{>0} \mid \exists P' \in \mathbb{P}.\, P \xrightarrow{a,*_w} P' \}\!\!\}$ and:

$$norm(w_1, w_2, a, P_1, P_2) = \frac{w_1}{weight(P_1, a)} \cdot \frac{w_2}{weight(P_2, a)} \cdot (weight(P_1, a) + weight(P_2, a)).$$

The semantic model of P is a labeled multitransition system $[\![P]\!]$ – i.e. a transition system taking into account the number of instances of each transition – whose multitransition relation is described by the elements of $\mathbb{P} \times Act \times \mathbb{P}$ satisfying the operational semantic rules above and such that the transition multiplicity denotes the number of different proofs for its derivation. This is necessary because the idempotent law does not hold in the stochastic setting (see, e.g., [16,8] for a treatment of this problem). As an example, $<a, \lambda>.P + <a, \lambda>.P$ is not the same as $<a, \lambda>.P$ because of the race policy. Instead, it is equated by $<a, 2 \cdot \lambda>.P$.

Finally, we denote with $\mathcal{P}$ the set of performance closed process terms of $\mathbb{P}$, i.e. they do not offer nondurational actions to the environment. Hence, from the semantic model of $P \in \mathcal{P}$ we can derive an action-labeled continuous-time Markov chain. This restriction will allow us to compare process terms from the probabilistic/temporal standpoint without resorting to schedulers that would be needed to solve the nondeterminism.

2.2. Markovian testing equivalence

Markovian testing equivalence requires to compare test-driven process computations. Hence, we start by introducing the notion of computation of a process term, which is a (possibly empty) sequence of transitions that can be executed starting from the initial state associated with the term. The length of a computation is the number n of its constituting transitions, where $n = 0$ whenever the computation is empty. We denote with $\mathcal{C}_f(P)$ the multiset of finite-length computations starting from $P \in \mathcal{P}$. Note that $\mathcal{C}_f(P)$ is a multiset because the semantics of process terms is given by labeled multitransition systems. Then, two distinct computations are independent of each other if neither is a proper prefix of the other one. In the remainder, we concentrate on finite sets of independent, finite-length computations. Before describing in detail the computation attributes, we recall the notion of exit rate of a process term.

Definition 2.1. Let $P \in \mathcal{P}$, $a \in \mathit{Name}$, and $C \subseteq \mathcal{P}$. The exit rate at which P executes actions of name a that lead to C is defined through the non-negative real function:

$$\mathit{rate}(P, a, C) = \sum \{\!|\, \lambda \in \mathbb{R}_{>0} \mid \exists P' \in C.\, P \xrightarrow{a,\lambda} P' \,|\!\}$$

where the summation is taken to be zero whenever its multiset is empty. □

If we sum up the rates of all the actions that a process term P can execute, we obtain the total exit rate of P as $\mathit{rate}_{\mathrm{t}}(P) = \sum_{a \in \mathit{Name}} \mathit{rate}(P, a, \mathcal{P})$.

We now define formally the concrete trace, the probability, and the duration of an element of $\mathcal{C}_{\mathrm{f}}(P)$, using $\_ \circ \_$ for sequence concatenation and $|\_|$ for sequence length.

Definition 2.2. Let $P \in \mathcal{P}$ and $c \in \mathcal{C}_{\mathrm{f}}(P)$. The concrete trace associated with c is the sequence of action names labeling the transitions of c, which is defined by induction on the length of c through the Name^*-valued function:

$$\mathit{trace}(c) = \begin{cases} \delta & \text{if } |c| = 0 \\ a \circ \mathit{trace}(c') & \text{if } c \equiv P \xrightarrow{a,\lambda} c' \end{cases}$$

where δ is the empty trace. □

Definition 2.3. Let $P \in \mathcal{P}$ and $c \in \mathcal{C}_{\mathrm{f}}(P)$. The probability of executing c is the product of the execution probabilities of the transitions of c, which is defined by induction on the length of c through the $\mathbb{R}_{]0,1]}$-valued function:

$$\mathit{prob}(c) = \begin{cases} 1 & \text{if } |c| = 0 \\ \dfrac{\lambda}{\mathit{rate}_{\mathrm{t}}(P)} \cdot \mathit{prob}(c') & \text{if } c \equiv P \xrightarrow{a,\lambda} c'. \end{cases}$$

We also define the probability of executing a computation in $C \subseteq \mathcal{C}_{\mathrm{f}}(P)$ as:

$$\mathit{prob}(C) = \sum_{c \in C} \mathit{prob}(c)$$

whenever C is finite and all of its computations are independent of each other. □

Definition 2.4. Let $P \in \mathcal{P}$ and $c \in \mathcal{C}_{\mathrm{f}}(P)$. The stepwise average duration of c is the sequence of average sojourn times in the states traversed by c, which is defined by induction on the length of c through the $(\mathbb{R}_{>0})^*$-valued function:

$$\mathit{time}(c) = \begin{cases} \delta & \text{if } |c| = 0 \\ \dfrac{1}{\mathit{rate}_{\mathrm{t}}(P)} \circ \mathit{time}(c') & \text{if } c \equiv P \xrightarrow{a,\lambda} c' \end{cases}$$

where δ is the empty stepwise average duration. The multiset of computations in $C \subseteq \mathcal{C}_{\mathrm{f}}(P)$ whose stepwise average duration is not greater than $\theta \in (\mathbb{R}_{>0})^*$ is defined as:

$$C_{\leq \theta} = \{\!|\, c \in C \mid |c| \leq |\theta| \wedge \forall i = 1, \ldots, |c|.\, \mathit{time}(c)[i] \leq \theta[i] \,|\!\}. \quad \square$$

We denote with C^l the multiset of computations in $C \subseteq \mathcal{C}_{\mathrm{f}}(P)$ whose length is equal to $l \in \mathbb{N}$. Moreover, we sometimes use shorthands of the form $\mathit{time}(c) \leq \theta$ to denote that $\mathit{time}(c)$ is stepwise less or equal to θ.

The other fundamental ingredient underlying Markovian testing equivalence is the notion of test. Process terms are observed by interacting with them by means of tests, which are represented as process terms that are composed in parallel with the process term under test by enforcing synchronization on all visible action names. Since the process term under test is performance closed, a test can interact by offering nondurational actions only. Intuitively, in any of its states the process term proposes the execution of a durational action and then, if such an action is visible, the test reacts either by enabling the interaction or by blocking it (note that tests cannot block the execution of τ actions). Then, the test is passed with success whenever a specific point during execution is reached with a certain probability and within an arbitrary sequence of average amounts of time. Thanks to the presence of these average time upper bounds, it is enough to consider acyclic finite-state labeled multitransition systems for the test representation. In other words, we can restrict ourselves to nonrecursive tests.

Definition 2.5. The set $\mathbb{T}_{\mathrm{R}}$ of reactive tests is generated by the syntax:

$$\begin{aligned} T &::= \mathrm{s} \mid T' \\ T' &::= <a, *_w>.T \mid T' + T' \end{aligned}$$

where $a \in \mathit{Name}_{\mathrm{v}}$, $w \in \mathbb{R}_{>0}$, and s is a zeroary operator standing for success. □

Given $P \in \mathcal{P}$ and $T \in \mathbb{T}_{\mathrm{R}}$, the interaction system of P and T is the process term $P \parallel_{Name_{\mathrm{v}}} T$, where each of its states is called a configuration. We say that a configuration is successful if its test part is s, and that a test-driven computation is successful if it traverses a successful configuration. We denote with $\mathcal{SC}(P, T)$ the multiset of successful computations of $P \parallel_{Name_{\mathrm{v}}} T$. Note that for any sequence $\theta \in (\mathbb{R}_{>0})^*$ of average amounts of time the multiset $\mathcal{SC}^{|\theta|}_{\leq\theta}(P, T)$ is finite and all of its computations have a finite length and are independent of each other. We are now ready to formalize the definition of Markovian testing equivalence, which is based on the comparison between the probabilities of successful test-driven time-constrained computations.

Definition 2.6. Let $P_1, P_2 \in \mathcal{P}$. We say that P_1 is Markovian testing equivalent to P_2, written $P_1 \sim_{\mathrm{MT}} P_2$, iff for all reactive tests $T \in \mathbb{T}_{\mathrm{R}}$ and sequences $\theta \in (\mathbb{R}_{>0})^*$ of average amounts of time:

$$prob(\mathcal{SC}^{|\theta|}_{\leq\theta}(P_1, T)) = prob(\mathcal{SC}^{|\theta|}_{\leq\theta}(P_2, T)). \quad \square$$

Actually, the set of tests respecting a canonical form is necessary and sufficient to decide whether two process terms are Markovian testing equivalent [5]. Each of these canonical tests allows for one computation leading to success, whose intermediate states can have alternative computations leading to failure in one step.

Definition 2.7. The set $\mathbb{T}_{\mathrm{R,c}}$ of canonical reactive tests is generated by the syntax:

$$T ::= \mathrm{s} \mid <a, *_1>.T + \sum_{b \in \mathcal{E}-\{a\}} <b, *_1>.\mathrm{f}$$

where $a \in \mathcal{E}$, $\mathcal{E} \subseteq Name_{\mathrm{v}}$ is finite, the summation is absent whenever $\mathcal{E} = \{a\}$, and f is a zeroary operator standing for failure. $\square$

Corollary 2.8. *Let $P_1, P_2 \in \mathcal{P}$. Then, $P_1 \sim_{\mathrm{MT}} P_2$ iff for all $T \in \mathbb{T}_{\mathrm{R,c}}$ and $\theta \in (\mathbb{R}_{>0})^*$ of average amounts of time:*

$$prob(\mathcal{SC}^{|\theta|}_{\leq\theta}(P_1, T)) = prob(\mathcal{SC}^{|\theta|}_{\leq\theta}(P_2, T)).$$

$\square$

Example 2.9. Let us consider a client–server system with a dual-core server. Requests arrive at the system with rate $\lambda \in \mathbb{R}_{>0}$. When a request finds both cores busy, it must immediately leave the system; i.e., no buffer is present. When a request finds both cores idle, it has the same probability to be accepted by the two cores. Both cores serve incoming requests at rate $\mu \in \mathbb{R}_{>0}$. They can also fail to process the request with rate $\varphi \in \mathbb{R}_{>0}$.

In MPC, we describe the client with the following arrival process:

$$Arrivals \triangleq <arrive, \lambda>.Arrivals$$

while the behavior of each core is modeled as follows:

$$Core \triangleq <arrive, *_1>.(<serve, \mu>.Core + <fail, \varphi>.Core).$$

Hence, the overall system is given by $CS \triangleq Arrivals \parallel_{\{arrive\}} (Core \parallel_{\emptyset} Core)$, which can be equated, through $\sim_{\mathrm{MT}}$, to the following process term representing a single-core server with a two-positions buffer:

$$P \triangleq <arrive, \lambda>.P'$$
$$P' \triangleq <serve, \mu>.P + <fail, \varphi>.P + <arrive, \lambda>.P''$$
$$P'' \triangleq <serve, 2 \cdot \mu>.P' + <fail, 2 \cdot \varphi>.P'$$

3. Approximating time, probability, and behavior

In this section, we consider approximate notions of $\sim_{\mathrm{MT}}$ based on the three following dimensions: time taken to pass a test (Section 3.1), probability with which tests are passed (Section 3.2), and syntactical form of the passed test (Section 3.3). The goal is to estimate from different perspectives how much a process term P_2 is similar to a given process term P_1, where we assume that P_1 is the original model to be approximated through an alternative model P_2. The three approximations are then merged into a general definition of similarity (Section 3.4).

Any notion of behavioral similarity should meet a number of good properties, among which we expect it to be a conservative extension of the behavioral equivalence that it intends to approximate. In essence, it should result into a distance function d satisfying (i) $d(P_1, P_2) = 0$ whenever $P_1 \sim_{\mathrm{MT}} P_2$ and (ii) the triangular inequality, i.e. it is worth establishing what can be "transitively" inferred about the distance $d(P_1, P_3)$ whenever the distances $d(P_1, P_2)$ and $d(P_2, P_3)$ are known.

On the other hand, if we intend to use safely the compositionality properties of $\sim_{\mathrm{MT}}$, another expected law is that $d(P, Q) = d(P', Q')$ whenever $P \sim_{\mathrm{MT}} P'$ and, likewise, $Q \sim_{\mathrm{MT}} Q'$. Finally, the similarity notion should be equipped with an efficient verification algorithm.

All these properties will be taken into account when defining relaxations of $\sim_{\mathrm{MT}}$ based on the three orthogonal aspects mentioned before.

3.1. Approximating time

In the setting of $\sim_{\mathrm{MT}}$, the time needed to pass a test with success is expressed as the sequence of average sojourn times in the states traversed by successful computations. Hence, introducing a tolerance at this level amounts to relaxing the condition that requires the process terms under test to respect the same temporal constraints.

We start with a notion of "slow approximation" with the aim of comparing a process term P_1 with a process term P_2 that is slightly slower than P_1 in the following sense. To explain the intuition, we point out that in the "exact" setting any computation of P_1 is matched by a computation of P_2 with the same concrete trace and stepwise average duration. In our "approximated" setting any computation c of P_1 is matched by all the computations of P_2 with the same concrete trace and with stepwise average duration that lies in a temporal interval whose lower bound is $time(c)$ and whose upper bound is described by $time(c)$ augmented with some negligible $\epsilon \in \mathbb{R}_{\geq 0}$. The intuition behind the temporal approximation above is captured through the following relaxation of the multiset $C_{\leq\theta}$, which is formalized with respect to the sequence θ of average amounts of time modeling the temporal constraint, the tolerance ϵ, and the multiset C' of finite-length computations to be approximated.

Definition 3.1. Let C, C' be multisets of finite-length computations, $\epsilon \in \mathbb{R}_{\geq 0}$, and $\theta \in (\mathbb{R}_{>0})^*$. The multiset of computations of $C_{\leq\theta}$ augmented with respect to ϵ and C' is given by:

$$C_{\leq\theta+\epsilon,C'} = C_{\leq\theta} \cup \{\!\!\{\, c \in C \mid c \notin C_{\leq\theta} \wedge \exists c' \in C'_{\leq\theta}.\, \forall i = 1, \ldots, |c|. \\ time(c')[i] \leq time(c)[i] \leq time(c')[i] + \epsilon \,\}\!\!\}. \quad \square$$

Based on this definition, $\mathcal{SC}^{|\theta|}_{\leq\theta+\epsilon,\mathcal{SC}^{|\theta|}(P_1,T)}(P_2, T)$ contains not only the successful T-driven computations of P_2 that meet the θ-constraint, but also the computations that are stepwise slower, up to ϵ, than those of $\mathcal{SC}^{|\theta|}_{\leq\theta}(P_1, T)$. Then, the relaxed variant of $\sim_{\mathrm{MT}}$ states that P_1 must be functionally and probabilistically equated by a version P_2 with less restrictive temporal constraints.

Definition 3.2. Let $P_1, P_2 \in \mathcal{P}$ and $\epsilon \in \mathbb{R}_{\geq 0}$. We say that P_2 is a slow Markovian testing ϵ-approximation of P_1, written $P_1 \sim^{\epsilon}_{\mathrm{MT,slow}} P_2$, iff for all reactive tests $T \in \mathbb{T}_{\mathrm{R}}$ and sequences $\theta \in (\mathbb{R}_{>0})^*$ of average amounts of time:

$$prob(\mathcal{SC}^{|\theta|}_{\leq\theta}(P_1, T)) = prob(\mathcal{SC}^{|\theta|}_{\leq\theta+\epsilon,\mathcal{SC}^{|\theta|}(P_1,T)}(P_2, T)). \quad \square$$

Example 3.3. Consider a version of the running example, call it CS_λ, where for simplicity all the rates are equal to λ. Now, take a variant of CS_λ, call it $CS_{\lambda,10\%}$, in which the system clock cycle is reduced by 10%, i.e. the rate λ becomes $0.9 \cdot \lambda$. Then, it turns out that $CS_\lambda \sim^{0.\bar{1}}_{\mathrm{MT,slow}} CS_{\lambda,10\%}$. Here the intuition is that, from the behavioral and probabilistic standpoints, every computation of CS_λ is matched by a corresponding computation of $CS_{\lambda,10\%}$ which, however, is slower by a 10% factor.

The following two lemmata formalize the intuition underlying $\sim^{\epsilon}_{\mathrm{MT,slow}}$ and establish conditions that are useful to state that the good properties mentioned before are satisfied.

Lemma 3.4. *Let $P_1, P_2 \in \mathcal{P}$. Whenever $P_1 \sim^{\epsilon}_{\mathrm{MT,slow}} P_2$ then there exists $c_2 \in \mathcal{C}_{\mathrm{f}}(P_2)$ iff there exists $c_1 \in \mathcal{C}_{\mathrm{f}}(P_1)$ such that $trace(c_1) = trace(c_2)$ and $time(c_1) \leq time(c_2) \leq time(c_1) + \epsilon$.*

Proof. We proceed by induction on $|c_2|$, with $c_2 \in \mathcal{C}_{\mathrm{f}}(P_2)$. Then, we can argue in the same way by exchanging the roles of c_2 and c_1.

On the one hand, let $|c_2| = 0$. Then, there trivially exists $c_1 \in \mathcal{C}_{\mathrm{f}}(P_1)$ such that $trace(c_1) = \delta = trace(c_2)$ and $time(c_1) = \delta = time(c_2)$.

On the other hand, let $|c_2| = n > 0$ and suppose $c_2 \equiv c'_2 \xrightarrow{a,\lambda} P^n_2$. By the induction hypothesis, there exists $c'_1 \in \mathcal{C}_{\mathrm{f}}(P_1)$ such that $trace(c'_1) = trace(c'_2)$ and $time(c'_1) \leq time(c'_2) \leq time(c'_1) + \epsilon$. As a consequence, $trace(c_2) = trace(c'_2) \circ a = trace(c'_1) \circ a = trace(c_1)$ and $time(c_1) \leq time(c_2) \leq time(c_1) + \epsilon$, where $c_1 \equiv c'_1 \xrightarrow{a,\mu} P^n_1$ belongs to $\mathcal{C}_{\mathrm{f}}(P_1)$, otherwise a test whose only trace coincides with $trace(c_2)$ would be enough to distinguish P_2 and P_1 when considering successful test-driven computations of length n. $\square$

In the following, given $t \in Name^*$ and $\theta \in (\mathbb{R}_{>0})^*$, such that $|t| = |\theta|$, let $\mathcal{C}_{t,\theta}(P) = \{\!\!\{\, c \in \mathcal{C}_{\mathrm{f}}(P) \mid trace(c) = t \wedge time(c) = \theta \,\}\!\!\}$ be the multiset of finite-length computations of P with concrete trace t and stepwise average duration θ. Then, let $\mathcal{C}_{t,\theta_\epsilon}(P) = \{\!\!\{\, c \in \mathcal{C}_{\mathrm{f}}(P) \mid trace(c) = t \wedge \forall i = 1, \ldots, |c|.\, \theta[i] \leq time(c)[i] \leq \theta[i] + \epsilon \,\}\!\!\}$ be the multiset of finite-length computations of P with concrete trace t and stepwise average duration confined between θ and $\theta + \epsilon$.

Lemma 3.5. *Let $P_1, P_2 \in \mathcal{P}$. Whenever $P_1 \sim^{\epsilon}_{\mathrm{MT,slow}} P_2$ then for every $\mathcal{C}_{t,\theta}(P_1)$, where $t \in Name^*$ and $\theta \in (\mathbb{R}_{>0})^*$, it holds that:*

$$prob(\mathcal{C}_{t,\theta}(P_1)) = prob(C_{t,\theta_\epsilon}(P_2)).$$

Proof. We proceed by induction on the number n of disjoint multisets of finite-length computations with a given trace and stepwise average duration occurring in $\mathcal{C}_{\mathrm{f}}(P_1)$. The base case $n = 0$ is trivial.

Let us assume $n > 0$ and consider any nonempty multiset $C_{t',\theta'}(P_1)$ such that there exist $T \in \mathbb{T}_{\mathrm{R}}, \theta \in (\mathbb{R}_{>0})^*$, and $1 \leq i \leq |\theta|$ satisfying the conditions $C_{t',\theta'}(P_1) \subseteq \mathcal{SC}^{|\theta|}_{\leq\theta}(P_1, T)$ and $\theta'[i] = \max\{time(c)[i] \mid c \in \mathcal{SC}^{|\theta|}_{\leq\theta}(P_1, T)\}$, for which we define a new sequence $\bar{\theta} = \theta$ except $\bar{\theta}[i] = \theta[i] - \bar{\epsilon}$, with $\bar{\epsilon} \in \mathbb{R}_{>0}$ such that $\mathcal{SC}^{|\bar{\theta}|}_{\leq\bar{\theta}}(P_1, T)$ contains all the multisets of computations occurring in $\mathcal{SC}^{|\theta|}_{\leq\theta}(P_1, T)$ except $C_{t',\theta'}$. By virtue of Lemma 3.4 it must be that $\mathcal{C}_{t',\theta'_\epsilon}(P_2) \subseteq \mathcal{SC}^{|\theta|}_{\leq\theta+\epsilon,\mathcal{SC}^{|\theta|}(P_1,T)}(P_2, T)$ but $\mathcal{C}_{t',\theta'_\epsilon}(P_2) \not\subseteq \mathcal{SC}^{|\bar{\theta}|}_{\leq\bar{\theta}+\epsilon,\mathcal{SC}^{|\bar{\theta}|}(P_1,T)}(P_2, T)$. Hence, since $P_1 \sim^{\epsilon}_{\mathrm{MT,slow}} P_2$, it holds that $prob(\mathcal{C}_{t',\theta'}(P_1)) = prob(\mathcal{C}_{t',\theta'_\epsilon}(P_2))$. By the induction hypothesis, the analogous result holds for all the remaining $\mathcal{C}_{t'',\theta''}(P_1)$. □

Proposition 3.6. *Let $P_1, P_2 \in \mathcal{P}$. Then, $P_1 \sim_{\mathrm{MT}} P_2$ iff $P_1 \sim^{0}_{\mathrm{MT,slow}} P_2$.*

Proof. The result is a consequence of the equality $C_{\leq\theta+0,C'} = C$. □

Proposition 3.7. *Let $P_1, P_2, P_3 \in \mathcal{P}$ and $\epsilon, \gamma \in \mathbb{R}_{\geq 0}$. If $P_1 \sim^{\epsilon}_{\mathrm{MT,slow}} P_2$ and $P_2 \sim^{\gamma}_{\mathrm{MT,slow}} P_3$, then $P_1 \sim^{\delta}_{\mathrm{MT,slow}} P_3$ for some $\delta \leq \epsilon + \gamma$.*

Proof. Let $T \in \mathbb{T}_{\mathrm{R}}$ and $\theta \in (\mathbb{R}_{>0})^*$. By hypothesis, we have that:

$$prob(\mathcal{SC}^{|\theta|}_{\leq\theta}(P_1, T)) = prob(\mathcal{SC}^{|\theta|}_{\leq\theta+\epsilon,\mathcal{SC}^{|\theta|}(P_1,T)}(P_2, T))$$

and:

$$prob(\mathcal{SC}^{|\theta|}_{\leq\theta}(P_2, T)) = prob(\mathcal{SC}^{|\theta|}_{\leq\theta+\gamma,\mathcal{SC}^{|\theta|}(P_2,T)}(P_3, T)).$$

By virtue of Lemma 3.4, it holds that $\exists c_1 \in \mathcal{C}_{\mathrm{f}}(P_1)$ iff $\exists c_2 \in \mathcal{C}_{\mathrm{f}}(P_2)$ such that $trace(c_1) = trace(c_2)$ and $time(c_1) \leq time(c_2) \leq time(c_1) + \epsilon$ iff $\exists c_3 \in \mathcal{C}_{\mathrm{f}}(P_3)$ such that $trace(c_2) = trace(c_3)$ and $time(c_2) \leq time(c_3) \leq time(c_2) + \gamma$, from which we derive the following relation: there exists $\delta \leq \epsilon + \gamma$ such that $\exists c_1 \in \mathcal{C}_{\mathrm{f}}(P_1)$ iff $\exists c_3 \in \mathcal{C}_{\mathrm{f}}(P_3)$ such that $trace(c_1) = trace(c_3)$ and $time(c_1) \leq time(c_3) \leq time(c_1) + \delta$. We now prove the equality:

$$prob(\mathcal{SC}^{|\theta|}_{\leq\theta+\epsilon,\mathcal{SC}^{|\theta|}(P_1,T)}(P_2, T)) = prob(\mathcal{SC}^{|\theta|}_{\leq\theta+\delta,\mathcal{SC}^{|\theta|}(P_1,T)}(P_3, T)).$$

By virtue of the relation surveyed above, for all $t' \in Name^*$ and $\theta' \in (\mathbb{R}_{>0})^*$ such that $|t'| = |\theta'|$, it holds that $\mathcal{C}_{t',\theta'}(P_2) \subseteq \mathcal{SC}^{|\theta|}_{\leq\theta+\epsilon,\mathcal{SC}^{|\theta|}(P_1,T)}(P_2, T)$ iff $\mathcal{C}_{t',\theta'_\gamma}(P_3) \subseteq \mathcal{SC}^{|\theta|}_{\leq\theta+\delta,\mathcal{SC}^{|\theta|}(P_1,T)}(P_3, T)$. Hence, by virtue of Lemma 3.5, we derive that $prob(\mathcal{C}_{t',\theta'}(P_2)) = prob(\mathcal{C}_{t',\theta'_\gamma}(P_3))$ from which we obtain the equality above and, as a consequence, the expected result:

$$prob(\mathcal{SC}^{|\theta|}_{\leq\theta}(P_1, T)) = prob(\mathcal{SC}^{|\theta|}_{\leq\theta+\delta,\mathcal{SC}^{|\theta|}(P_1,T)}(P_3, T)).$$ □

Proposition 3.8. *Let $P, Q, P', Q' \in \mathcal{P}$, and $\epsilon \in \mathbb{R}_{\geq 0}$. If* (i) $P \sim^{\epsilon}_{\mathrm{MT,slow}} Q$, (ii) $P \sim_{\mathrm{MT}} P'$, *and* (iii) $Q \sim_{\mathrm{MT}} Q'$, *then $P' \sim^{\epsilon}_{\mathrm{MT,slow}} Q'$.*

Proof. Let $T \in \mathbb{T}_{\mathrm{R}}$ and $\theta \in (\mathbb{R}_{>0})^*$. By virtue of (ii) and (i):

$$prob(\mathcal{SC}^{|\theta|}_{\leq\theta}(P', T)) = prob(\mathcal{SC}^{|\theta|}_{\leq\theta}(P, T)) = prob(\mathcal{SC}^{|\theta|}_{\leq\theta+\epsilon,\mathcal{SC}^{|\theta|}(P,T)}(Q, T)).$$

Now, by virtue of Lemma 3.5 and (iii), it holds that:

$$prob(\mathcal{SC}^{|\theta|}_{\leq\theta+\epsilon,\mathcal{SC}^{|\theta|}(P,T)}(Q, T)) = prob(\mathcal{SC}^{|\theta|}_{\leq\theta+\epsilon,\mathcal{SC}^{|\theta|}(P,T)}(Q', T))$$

and, again by virtue of Lemma 3.5 and (ii), it holds that:

$$prob(\mathcal{SC}^{|\theta|}_{\leq\theta+\epsilon,\mathcal{SC}^{|\theta|}(P,T)}(Q', T)) = prob(\mathcal{SC}^{|\theta|}_{\leq\theta+\epsilon,\mathcal{SC}^{|\theta|}(P',T)}(Q', T))$$

from which the result follows. □

We conclude the discussion concerning the properties of $\sim^{\epsilon}_{\mathrm{MT,slow}}$ by observing that the alternative characterization expressed in Corollary 2.8 is preserved in this relaxed framework.

Proposition 3.9. *Let $P_1, P_2 \in \mathcal{P}$ and $\epsilon \in \mathbb{R}_{\geq 0}$. Then, $P_1 \sim^{\epsilon}_{\mathrm{MT,slow}} P_2$ iff for all $T \in \mathbb{T}_{\mathrm{R,c}}$ and $\theta \in (\mathbb{R}_{>0})^*$ of average amounts of time:*

$$prob(\mathcal{SC}^{|\theta|}_{\leq\theta}(P_1, T)) = prob(\mathcal{SC}^{|\theta|}_{\leq\theta+\epsilon,\mathcal{SC}^{|\theta|}(P_1,T)}(P_2, T)).$$

Proof Sketch. We first observe that Corollary 2.8 is a consequence of an alternative characterization of $\sim_{\mathrm{MT}}$ that fully abstracts from tests by considering, instead, computations that are extended at each step with the set of visible action names enabled by the environment at that step [5]. The same characterization applies also to $\sim^{\epsilon}_{\mathrm{MT,slow}}$. In particular, in the case of $\mathcal{SC}^{|\theta|}_{\leq\theta+\epsilon,\mathcal{SC}^{|\theta|}(P_1,T)}(P_2, T)$ it is sufficient to replace in the related proof each occurrence of the condition $\leq \theta$ with $\leq \theta + \epsilon$, $\mathcal{SC}^{|\theta|}(P_1, T)$ and to extend analogously the verification of the θ-constraint according with Definition 3.1. □

After the introduction of $\sim^{\epsilon}_{\mathrm{MT,slow}}$, it is natural to investigate its counterpart $\sim^{\epsilon}_{\mathrm{MT,fast}}$ that allows a process term P_1 to be approximated by a process term P_2 that is "slightly faster" than P_1. As opposite to the argument applied in the case of $\sim^{\epsilon}_{\mathrm{MT,slow}}$, when considering the successful T-driven computations of P_2 with respect to $\theta \in (\mathbb{R}_{>0})^*$, we should not count every computation that satisfies the θ-constraint with stepwise average duration that is lower, up to a threshold $\epsilon \in \mathbb{R}_{\geq 0}$, than that of a corresponding successful T-driven computation of P_1 which, instead, does not satisfy the θ-constraint. This intuition results into the following symmetric reformulations of Definitions 3.1 and 3.2.

Definition 3.10. Let C, C' be multisets of finite-length computations, $\epsilon \in \mathbb{R}_{\geq 0}$, and $\theta \in (\mathbb{R}_{>0})^*$. The multiset of computations of $C_{\leq\theta}$ reduced with respect to ϵ and C' is given by:

$$C_{\leq\theta-\epsilon,C'} = C_{\leq\theta} - \{\!|\, c \in C \mid c \in C_{\leq\theta} \wedge \exists c' \in C'.\, c' \notin C'_{\leq\theta} \wedge \forall i = 1, \ldots, |c|. \\ time(c')[i] - \epsilon \leq time(c)[i] \leq time(c')[i] \,|\!\}. \quad \square$$

Definition 3.11. Let $P_1, P_2 \in \mathcal{P}$ and $\epsilon \in \mathbb{R}_{\geq 0}$. We say that P_2 is a fast Markovian testing ϵ-approximation of P_1, written $P_1 \sim^{\epsilon}_{\mathrm{MT,fast}} P_2$, iff for all reactive tests $T \in \mathbb{T}_{\mathrm{R}}$ and sequences $\theta \in (\mathbb{R}_{>0})^*$ of average amounts of time:

$$prob(\mathcal{AC}^{|\theta|}_{\leq\theta}(P_1, T)) = prob(\mathcal{AC}^{|\theta|}_{\leq\theta-\epsilon,\mathcal{AC}^{|\theta|}(P_1,T)}(P_2, T)). \quad \square$$

By following symmetric arguments, through a trivial recasting it is possible to prove the counterparts of Lemmata 3.4 and 3.5, from which we immediately derive the counterparts of Propositions 3.6–3.8, and, similarly, Proposition 3.9.

3.1.1. Verification algorithm

The temporal approximations of Markovian testing equivalence can be decided in polynomial time by exploiting a smooth reworking of the algorithm for $\sim_{\mathrm{MT}}$, because the main objective – i.e. equating the execution probability of certain computations – does not change.

The original algorithm works as follows. Firstly, we observe that $\sim_{\mathrm{MT}}$ coincides with Markovian ready equivalence, which is verified by reducing it to probabilistic ready equivalence under a suitable transformation of the semantics of the process terms to compare, say P_1 and P_2. The labeled continuous-time Markov chains underlying $[\![P_1]\!]$ and $[\![P_2]\!]$ are transformed into corresponding embedded labeled discrete-time Markov chains, say $[\![P_1]\!]_{\mathrm{d}}$ and $[\![P_2]\!]_{\mathrm{d}}$, in the following way:

- Turn the rate of each transition into the corresponding execution probability. This is obtained by dividing the transition rate by the total exit rate of its source state.
- Augment the name of each transition with the total exit rate of its source state.

While the first step is needed to turn rates into probabilities, the second step is motivated by the fact that the state condition concerning the average sojourn time imposed by $\sim_{\mathrm{MT}}$ should not be lost when passing from the Markovian setting to the probabilistic setting.

Secondly, probabilistic ready equivalence is known to be decidable [15] by applying the algorithm for probabilistic language equivalence [17] as follows:

a. Compute the equivalence relation $\mathcal{R}$ equating any two states of $[\![P_1]\!]_{\mathrm{d}}$ and $[\![P_2]\!]_{\mathrm{d}}$ whenever the two sets of augmented action names labeling the transitions departing from the two states are equal.
b. For each equivalence class R induced by $\mathcal{R}$, consider the two probabilistic automata $\mathcal{U}_1$ and $\mathcal{U}_2$ that are obtained by extending $[\![P_1]\!]_{\mathrm{d}}$ and $[\![P_2]\!]_{\mathrm{d}}$, respectively, with the accepting set R. Then, apply the algorithm of [17] to decide probabilistic language equivalence for these automata.

Note that, by construction, $[\![P_i]\!]$, $[\![P_i]\!]_{\mathrm{d}}$, and $\mathcal{U}_i$ have the same number of states n_i, with $i \in \{1, 2\}$.

Now, we elucidate item b. in detail. Let us denote with *AugName* the set of augmented action names labeling the transitions of $\mathcal{U}_1$ and $\mathcal{U}_2$. The algorithm of [17] performs a breadth-first visit of the tree containing a node for each element of $AugName^*$ and establishes the linear independence of state probability vectors associated with a finite subset of the tree nodes.

To borrow terminology used in [17], $\mathcal{U}_1$ and $\mathcal{U}_2$ are equivalent iff for each $\sigma \in AugName^*$ the accepting probabilities of σ for these automata are equal. Formally, $\rho_1 M_1(\sigma)\eta_1 = \rho_2 M_2(\sigma)\eta_2$, where for $i \in \{1, 2\}$ we have that ρ_i is an n_i-dimensional row vector representing the initial state distribution, η_i is an n_i-dimensional column vector such that the i-th entry is 1 if it corresponds to an accepting state and 0 otherwise, and $M_i(\sigma)$ is a matrix such that $M_i(\sigma)[j, k]$ is the probability of reading σ along paths from the j-th state to the k-th state. The equation above can be reformulated as follows:

$$\forall \sigma \in AugName^* : \quad [\rho_1\ \rho_2] \begin{bmatrix} M_1(\sigma) & 0_{n_1\times n_2} \\ 0_{n_2\times n_1} & M_2(\sigma) \end{bmatrix} \begin{bmatrix} \eta_1 \\ -\eta_2 \end{bmatrix} = 0$$

where $0_{n\times m}$ is the $(n\times m)$-dimensional zero matrix. The vector–matrix multiplication in the formulation above, call it $P_{1,2}(\sigma)$, represents the state probability vector induced by σ. The idea is to find a basis V of vectors generating all the possible state probability vectors $P_{1,2}(\sigma)$, with $\sigma \in AugName^*$, and to check whether for all $v \in V :\ v\ [\eta_1\ -\eta_2]^T = 0$.

Initially, V is an empty set and an empty queue is initialized by adding the empty string to it. While the queue is not empty, its first element σ is removed and if the related state probability vector v, which can be computed in time $O(n^2)$, does not belong to the vector space generated by V – the cost of this check is $O(n^3)$ – then v is added to V and $\sigma \circ a$ to the queue for each $a \in AugName$. Whenever a basis V of (at most) n elements is generated, the algorithm returns 'yes' if $v\ [\eta_1\ -\eta_2]^T = 0$ for all $v \in V$ and 'no' if at least one state probability vector fails to meet the condition.

The time complexity of the algorithm of [17] is $O(n^4)$ (where $n = n_1 + n_2$) and since it is executed once for each class R induced by $\mathcal{R}$, the time complexity of the overall algorithm is $O(n^5)$. A reworking of this algorithm applies as well in the case of $\sim^{\epsilon}_{\mathrm{MT,slow}}$ and $\sim^{\epsilon}_{\mathrm{MT,fast}}$. The two modifications to consider are as follows. Firstly, replace Markovian ready equivalence with the corresponding relaxed version that equates to the slow (resp. fast) approximation of Markovian testing equivalence. Secondly, relax item a. by imposing that a state of $[\![P_1]\!]_{\mathrm{d}}$ is related to a state of $[\![P_2]\!]_{\mathrm{d}}$ (i.e. they are put into the same accepting set) iff the two sets of action names labeling the transitions departing from the two states coincide and the average sojourn time associated with the state of $[\![P_2]\!]_{\mathrm{d}}$ is $\geq$ (resp. $\leq$), up to ϵ, with respect to the average sojourn time associated with the state of $[\![P_1]\!]_{\mathrm{d}}$. These modifications do not alter the complexity of the overall algorithm, which is still $O(n^5)$.

3.2. Approximating probability

By following the same intuitions surveyed in the previous section, introducing a relaxation at the level of the probabilistic behavior of process terms would result into the following definition.

Definition 3.12. Let $P_1, P_2 \in \mathcal{P}$ and $\epsilon \in \mathbb{R}_{\geq 0}$. We say that P_2 is a probabilistic Markovian testing ϵ-approximation of P_1, written $P_1 \sim^{\epsilon}_{\mathrm{MT,prob}} P_2$ iff for all reactive tests $T \in \mathbb{T}_{\mathrm{R}}$ and sequences $\theta \in (\mathbb{R}_{>0})^*$ of average amounts of time:

$$|prob(\mathcal{BC}^{|\theta|}_{\leq\theta}(P_1, T)) - prob(\mathcal{BC}^{|\theta|}_{\leq\theta}(P_2, T))| \leq \epsilon. \quad \square$$

The good properties stating that $\sim^{\epsilon}_{\mathrm{MT,prob}}$ is a conservative extension of $\sim_{\mathrm{MT}}$, which are represented by an adequate recast of Propositions 3.6–3.8, are easily guaranteed.

Unfortunately, we cannot argue similarly for the verification algorithm. We recall that in the exact setting the problem of verifying $\sim_{\mathrm{MT}}$ corresponds to checking probabilistic language equivalence for a given set of pairs of probabilistic automata (see Section 3.1.1). By virtue of the transformations from labeled continuous-time Markov chains to probabilistic automata described in Section 3.1.1, we derive that the verification of $\sim^{\epsilon}_{\mathrm{MT,prob}}$ is an instance of the probabilistic language similarity problem. However, the problem of deciding whether two probabilistic automata accept the same input strings with close probabilities is undecidable [10], meaning that $\sim^{\epsilon}_{\mathrm{MT,prob}}$ is undecidable.

As discussed in the related work section, in the literature several different approaches – mainly based on bisimulation based semantics – have been proposed to overcome the limitations deriving from the introduction of a tolerance to fluctuations in the probabilistic behaviors. As an alternative, naive approach, in the next section we will show that decidability can be obtained by relaxing the condition over tests.

3.3. Approximating tests

While so far we have restricted the comparison between process terms to their quantitative properties, in this section we consider a notion of approximation that is based on the exemplary behavior of the tests guiding the comparison. The proposed approach is inspired by [9], where processes are compared with respect to an event log describing typical behaviors. Processes are defined in terms of Petri nets, while the event log is a multiset of firing sequences. In this setting, the degree with which an alternative model P_2 approximates the original model P_1 is estimated by measuring the mutual overlap in (partially) fitting these sequences, by comparing all enabled transitions at any point in each sequence. This idea results into two measures expressing precision and recall metrics. Precision establishes how much of the behavior of the alternative model P_2 exists in the behavior of the original model P_1 (soundness of the approximation). Recall expresses the fraction of behaviors of P_1 that is covered by P_2 (completeness of the approximation).

Firstly, we recast the definition of event log in the setting of Markovian testing equivalence, by observing that the notion of typical behavior with respect to which the comparison is conducted is explicitly represented by the set of canonical reactive tests. Therefore, while in [9] it is suggested to define the event log through simulation or by describing by hand some behavior of interest, here we formally define an event log as a finite subset of $T \in \mathbb{T}_{\mathrm{R,c}}$, call it $\mathbb{T}_{\mathrm{R,c},\phi}$, whose elements are tests satisfying properties described in terms of a logical formula ϕ of any variant of the classical Hennessy–Milner logic. In general, tests belonging to $\mathbb{T}_{\mathrm{R,c},\phi}$ represent the set of typical behaviors parameterized by ϕ which guide the estimation of the degree of similarity between process terms.

Secondly, we use a test-based formulation of the fitness notion to estimate the similarity between tests. The motivation is that we intend to relax $\sim_{\mathrm{MT}}$ – which imposes to compare the process terms P_1 and P_2 under the same test T – by permitting the comparison between P_1 under a test T_1 and P_2 under a test T_2 such that T_2 is an approximation of T_1. In other words, if P_1 satisfies a test with a certain probability and within a given amount of time, then we assume that P_2 can approximate the behavior of P_1 by satisfying with the same probability and by the same amount of time another test that fits the first test according to a quantitative notion of test similarity.

We now recast from [9] the two formulations of behavioral precision and recall for canonical test similarity. Let $trace(T, s)$ be the concrete trace associated with the unique successful computation of the canonical reactive test T, $|T|$ be the length of this trace, and T_i be the i-th process term of it, such that $T_1 ::= T$ and $T_{|T|}$ is the state that reaches success in one step. Then, $\forall i = 1, \ldots, |T|$, $enabled(T, i, s) = \{trace(T, s)[i]\}$ contains the action name associated with the transition belonging to the successful computation of T that is enabled at the i-th step. Intuitively, the similarity between two canonical tests is evaluated in terms of their capability of fitting the same successful computation.

Table 1
Transitivity relations for *prec* and *rec*: $z, w, x, y \in [0, 1[$.

$prec(T_1, T_2)$	$rec(T_1, T_2)$	$prec(T_2, T_3)$	$rec(T_2, T_3)$	$prec(T_1, T_3)$	$rec(T_1, T_3)$
z	w	x	y	≤ 1	≤ 1
z	w	x	1	< 1	$\geq w$
z	w	1	y	≤ 1	$\leq w$
z	w	1	1	z	w
z	1	x	y	$\leq x$	≤ 1
z	1	x	1	$< x$	1
z	1	1	y	≤ 1	≤ 1
z	1	1	1	z	1
1	w	x	y	$\geq x$	≤ 1
1	w	x	1	$\geq x$	$\geq w$
1	w	1	y	1	$< w$
1	w	1	1	1	w
1	1	x	y	x	y
1	1	x	1	x	1
1	1	1	y	1	y
1	1	1	1	1	1

Definition 3.13. The precision function $prec : \mathbb{T}_{\mathrm{R,c}} \times \mathbb{T}_{\mathrm{R,c}} \to [0, 1]$ is:

$$prec(T, T') = \frac{1}{|T'|} \sum_{i=1}^{|T'|} |(enabled(T, i, s) \cap enabled(T', i, s))|$$

while the recall function $rec : \mathbb{T}_{\mathrm{R,c}} \times \mathbb{T}_{\mathrm{R,c}} \to [0, 1]$ is:

$$rec(T, T') = \frac{1}{|T|} \sum_{i=1}^{|T|} |(enabled(T, i, s) \cap enabled(T', i, s))|. \quad \square$$

At each step the two transitions belonging to the successful computations of T and T' are compared, while the transitions leading to failure in one step are not considered, because their impact upon the estimation of precision and recall would contrast with the notion of Markovian testing equivalence, which is based on properties related to the behavior of the successful computations.

Note that T and T' are not imposed to have the same length. For instance, if $|T| = 2 \cdot |T'| = 2 \cdot n$ and the behaviors of T and T' coincide in the first n steps, then $prec(T, T') = 1$ because each behavior of T' is possible according to the behavior of T, while $rec(T, T') = \frac{1}{2}$ because only half of the behavior of T is covered by the behavior of T'.

Example 3.14. Consider the following tests for the running example:

$$T_1 = <arrive, *_1>.(<serve, *_1>.\mathrm{s} + <fail, *_1>.\mathrm{f})$$
$$T_2 = <arrive, *_1>.(<serve, *_1>.\mathrm{f} + <fail, *_1>.\mathrm{s})$$

modeling opposite reactions of the first core to the arrival of a single request. Then, $prec(T_1, T_2) = rec(T_1, T_2) = 0.5$. Indeed, they coincide in the first step, while in the second step they behave in opposite ways because we distinguish transitions leading to success from those leading to failure. Without this distinction, it would result $prec(T_1, T_2) = rec(T_1, T_2) = 1$.

Precision and recall are mutually symmetric, $prec(T, T') = rec(T', T)$, and satisfy the transitivity relations reported in Table 1. The transitivity proofs are a straightforward recast of those in [9].

By using a notion of test similarity based on the precision and recall functions, we now introduce the following relaxation of $\sim_{\mathrm{MT}}$.

Definition 3.15. Let $P_1, P_2 \in \mathcal{P}$, $\mathbb{T}_{\mathrm{R,c},\phi}$ be a finite subset of $\mathbb{T}_{\mathrm{R,c}}$ parameterized by the logical formula ϕ, and $p, r \in [0, 1]$. We say that P_2 is a behavioral Markovian testing (p, r)-approximation of P_1 with respect to ϕ, written $P_1 \sim^{p,r}_{\mathrm{MT},\phi} P_2$, iff for each $T \in \mathbb{T}_{\mathrm{R,c},\phi}$ there exists $T' \in \mathbb{T}_{\mathrm{R,c},\phi}$ such that for each sequence $\theta \in (\mathbb{R}_{>0})^*$ of average amounts of time:

1. $prec(T, T') \geq p$ and $rec(T, T') \geq r$.
2. $prob(\mathcal{SC}^{|\theta|}_{\leq\theta}(P_1, T)) = prob(\mathcal{SC}^{|\theta|}_{\leq\theta}(P_2, T'))$. $\square$

Note that $\sim^{p,r}_{\mathrm{MT},\phi}$ relies on the comparison between the observed behaviors expressed in terms of test-driven computations, where instead of a single test we consider a pair of tests that fit almost the same.

Example 3.16. Consider an alternative server for the running example that offers a functionally different service, modeled through the action name *serve*′ replacing the action name *serve*:

$$Core \triangleq <arrive, *_1>.(<serve', \mu>.Core + <fail, \varphi>.Core).$$

We call CS' the obtained system. Now, for instance, consider the family of tests satisfying the logical formula:

$$\phi = \langle arrive \rangle (\langle serve \rangle \text{true} \vee \langle serve' \rangle \text{true})$$

which is finite if we restrict the alphabet of action names occurring in these tests to the set {*serve*, *serve*′, *arrive*, *fail*}. Hence, for each test $T \in \mathbb{T}_{\mathrm{R,c},\phi}$ enabling the trace of action names $arrive \circ serve$, there exists a test T' replacing the occurrence of *serve* in the trace with the action name *serve*′ such that $prec(T, T') \geq 0.5$ ($rec(T, T') \geq 0.5$) and, for each $\theta \in (\mathbb{R}_{>0})^*$:

$$prob(\mathcal{BC}^{|\theta|}_{\leq\theta}(CS, T)) = prob(\mathcal{BC}^{|\theta|}_{\leq\theta}(CS', T'))$$

from which we derive that $CS \sim^{0.5,0.5}_{\mathrm{MT},\phi} CS'$.

Proposition 3.17. *Let $P_1, P_2, P_3, P_4 \in \mathcal{P}$, $\mathbb{T}_{\mathrm{R,c},\phi}$ be a finite subset of $\mathbb{T}_{\mathrm{R,c}}$ parameterized by the logical formula ϕ, and $p, r, p', r' \in [0, 1]$. Then:*

1. *If $P_1 \sim_{\mathrm{MT}} P_2$ then $P_1 \sim^{1,1}_{\mathrm{MT},\phi} P_2$.*
2. *If $P_1 \sim^{p,r}_{\mathrm{MT},\phi} P_2$ and $P_2 \sim^{p',r'}_{\mathrm{MT},\phi} P_3$ then $P_1 \sim^{u,w}_{\mathrm{MT},\phi} P_3$ where u and w derive from p, r, p', and r' by following the conditions of Table 1.*
3. *If (i) $P_1 \sim^{p,r}_{\mathrm{MT},\phi} P_2$, (ii) $P_1 \sim_{\mathrm{MT}} P_3$, (iii) $P_2 \sim_{\mathrm{MT}} P_4$, then $P_3 \sim^{p,r}_{\mathrm{MT},\phi} P_4$.*

Proof. Let $T \in \mathbb{T}_{\mathrm{R,c},\phi}$ and $\theta \in (\mathbb{R}_{>0})^*$.

Case 1. is an immediate consequence of $prec(T, T') = rec(T, T') = 1$ whenever T and T' coincide.

Case 2.: by hypothesis, there exists $T' \in \mathbb{T}_{\mathrm{R,c},\phi}$ such that:

$$prob(\mathcal{BC}^{|\theta|}_{\leq\theta}(P_1, T)) = prob(\mathcal{BC}^{|\theta|}_{\leq\theta}(P_2, T')).$$

Similarly, we also have $T'' \in \mathbb{T}_{\mathrm{R,c},\phi}$ such that:

$$prob(\mathcal{BC}^{|\theta|}_{\leq\theta}(P_2, T')) = prob(\mathcal{BC}^{|\theta|}_{\leq\theta}(P_3, T'')).$$

Hence, we obtain:

$$prob(\mathcal{BC}^{|\theta|}_{\leq\theta}(P_1, T)) = prob(\mathcal{BC}^{|\theta|}_{\leq\theta}(P_3, T''))$$

from which the result follows.

To show Case 3., we have to prove that there exists $T' \in \mathbb{T}_{\mathrm{R,c},\phi}$ such that:

$$prob(\mathcal{BC}^{|\theta|}_{\leq\theta}(P_3, T)) = prob(\mathcal{BC}^{|\theta|}_{\leq\theta}(P_4, T'))$$

with $prec(T, T') = p$ and $rec(T, T') = r$. By virtue of (i), there exists $T' \in \mathbb{T}_{\mathrm{R,c},\phi}$ such that:

$$prob(\mathcal{BC}^{|\theta|}_{\leq\theta}(P_1, T)) = prob(\mathcal{BC}^{|\theta|}_{\leq\theta}(P_2, T'))$$

with $prec(T, T') = p$ and $rec(T, T') = r$. By (ii), it must be:

$$prob(\mathcal{BC}^{|\theta|}_{\leq\theta}(P_1, T)) = prob(\mathcal{BC}^{|\theta|}_{\leq\theta}(P_3, T))$$

and, similarly, by (iii) we also have:

$$prob(\mathcal{BC}^{|\theta|}_{\leq\theta}(P_2, T')) = prob(\mathcal{BC}^{|\theta|}_{\leq\theta}(P_4, T'))$$

from which we obtain the expected result. □

3.3.1. Verification algorithm

The algorithm for $\sim_{\mathrm{MT}}$ illustrated in Section 3.1.1 can be applied to check $\sim^{p,r}_{\mathrm{MT},\phi}$ with the following modifications concerning the verification of probabilistic language equivalence. The comparison between the probabilistic automata $\mathcal{U}_1$ and $\mathcal{U}_2$ does not require the computation of a basis for the set of all the possible state probability vectors $P_{1,2}(\sigma)$, with $\sigma \in AugName^*$, as the definition of $\sim^{p,r}_{\mathrm{MT},\phi}$ is restricted to the finite set of tests parameterized by ϕ. In particular, the parameter of interest is the number m of different concrete traces associated with the successful computations of tests in $\mathbb{T}_{\mathrm{R,c},\phi}$. Since we have one such traces for each canonical test, m corresponds (at most) to the number of tests satisfying ϕ. Hence, the procedure detailing item *b*. in Section 3.1.1 is changed as follows.

Firstly, a node of the tree is marked whenever the projection of its string to $Name^*$ is equal to the concrete trace of the successful computation of a test satisfying ϕ. Since tests are nonrecursive, the longest branch (from the root to a leaf) is of length $\leq n$ and the string associated with each tree node is of length $\leq n$, so that the cost of the additional check is $\mathcal{O}(n)$. Then, the state probability vector is calculated – at cost $\mathcal{O}(n^2)$ – only for marked nodes, while linear independence is not to be checked anymore. Secondly, the procedure above terminates whenever all the concrete traces associated with successful computations have been generated. Hence, the cost is $O(m \cdot n^2)$.

Finally, for each state probability vector v, we check whether there exists a possibly different state probability vector v' such that:

$$v\ [\eta_1\ 0_{n_2}]^T + v'\ [0_{n_1}\ -\eta_2]^T = 0$$

where v, v' are associated with two tree nodes corresponding to the successful computations of two tests $T, T' \in \mathbb{T}_{\mathrm{R,c},\phi}$ satisfying $prec(T, T') \geq p$ and $rec(T, T') \geq r$. The cost of this check is $O(n \cdot m^2)$. Summarizing, the overall cost of the verification algorithm is $O((m \cdot n^3) + (n^2 \cdot m^2))$.

3.4. Joining the three approximations

Every approximation surveyed above deals with an aspect of the similarity problem – time, probability, behavior – in isolation. In real-world examples, this means that these orthogonal approaches can be used to evaluate limiting scenarios in which strong requirements constraint two of the three aspects above, while one of the three is perceived as malleable. For instance, whenever security is an issue, it is commonly accepted that the temporal behavior of programs can be artificially made worse in order to thwart any possible timing covert channel. Ideally, the secure version of these programs should not alter the functional and probabilistic properties of the original programs, otherwise new covert channels could arise. On the other hand, whenever the main issue is safety, it is acceptable to change the behavior of programs in order to make them more robust against failures of several different combinations of components. While these changes may alter some of the functional requirements of these programs – in a way that is controlled through the application of a logical formula ϕ parameterizing the behavior approximation of interest – they are not expected to jeopardize the quality of service.

In several real cases, the objectives above can be met only at some cost in terms of trade-offs among the orthogonal aspects we have considered in this paper. In order to investigate the dependences among these aspects and to obtain, if possible, a balanced trade-off, it is necessary to provide a unified definition joining the conditions expressed by each approximation. Thanks to the uniformity of the proposed approaches, the following unified definition is a natural and obvious consequence of its constituting elements and represents the final (and most important) contribution of this paper.

For notational convenience, we assume that a positive threshold denotes slow Markovian testing approximation and a negative threshold is used in the case of fast Markovian testing approximation.

Definition 3.18. Let $P_1, P_2 \in \mathcal{P}$ and $\mathbb{T}_{\mathrm{R,c},\phi}$ a finite set of tests. We say that P_2 is a Markovian testing approximation of P_1 with respect to ϕ, precision $p \in [0, 1]$, recall $r \in [0, 1]$, temporal threshold $\epsilon \in \mathbb{R}$, and probability threshold $\nu \in \mathbb{R}_{\geq 0}$, written $P_1 \sim^{p,r,\epsilon,\nu}_{\mathrm{MT},\phi} P_2$, iff for each $T \in \mathbb{T}_{\mathrm{R,c},\phi}$ there exists $T' \in \mathbb{T}_{\mathrm{R,c},\phi}$ such that for all $\theta \in (\mathbb{R}_{>0})^*$:

1. $prec(T, T') \geq p$ and $rec(T, T') \geq r$.
2. $|prob(\mathcal{SC}^{|\theta|}_{\leq\theta}(P_1, T)) - prob(\mathcal{SC}^{|\theta|}_{\leq\theta+\epsilon, \mathcal{SC}^{|\theta|}(P_1,T)}(P_2, T))| \leq \nu$. □

Example 3.19. Assume that the two cores of the running example are replaced by an alternative model ensuring more efficiency from a performance standpoint but offering a slightly different service from the functional standpoint. In real systems, this trade-off can be a consequence of the application of code optimization techniques. Such a model can be described as follows:

$$Core' \triangleq <arrive, *_1>.(<serve', \mu + \epsilon>.Core' + <fail, \varphi - \epsilon>.Core').$$

Let $CS' \triangleq Arrivals \parallel_{\{arrive\}} (Core' \parallel_{\emptyset} Core')$. Then, under the family of tests satisfying the logical formula:

$$\phi = \langle arrive\rangle(\langle fail\rangle \mathrm{true} \vee \langle serve\rangle \mathrm{true} \vee \langle serve'\rangle \mathrm{true} \vee \langle arrive\rangle(\langle fail\rangle \mathrm{true} \vee \langle serve\rangle \mathrm{true} \vee \langle serve'\rangle \mathrm{true}))$$

it holds that the new system CS' is a Markovian testing approximation of the original system CS with respect to ϕ, precision 0.5, recall 0.5, temporal threshold 0, and probability threshold ϵ.

All the conservativeness properties of this unifying definition are inherited from the results shown in the previous sections.

Proposition 3.20. *Let $P_1, P_2, P_3, P_4 \in \mathcal{P}$, $\mathbb{T}_{\mathrm{R,c},\phi}$ be a finite subset of $\mathbb{T}_{\mathrm{R,c}}$ parameterized by the logical formula ϕ. Moreover, let $\epsilon, \epsilon' \in \mathbb{R}$, $\nu, \nu' \in \mathbb{R}_{\geq 0}$, and $p, r, p', r' \in [0, 1]$. Then:*

1. *If $P_1 \sim_{\mathrm{MT}} P_2$ then $P_1 \sim^{1,1,0,0}_{\mathrm{MT},\phi} P_2$.*
2. *If $P_1 \sim^{p,r,\epsilon,\nu}_{\mathrm{MT},\phi} P_2$ and $P_2 \sim^{p',r',\epsilon',\nu'}_{\mathrm{MT},\phi} P_3$ then $P_1 \sim^{u,w,\delta,\gamma}_{\mathrm{MT},\phi} P_3$, where $\delta \leq \epsilon + \epsilon'$, $\gamma \leq \nu + \nu'$, while u and w derive from p, r, p', and r' by following the conditions of Table 1.*
3. *If (i) $P_1 \sim^{p,r,\epsilon,\nu}_{\mathrm{MT},\phi} P_2$, (ii) $P_1 \sim_{\mathrm{MT}} P_3$, (iii) $P_2 \sim_{\mathrm{MT}} P_4$, then $P_3 \sim^{p,r,\epsilon,\nu}_{\mathrm{MT},\phi} P_4$.*

Proof. Each case is a consequence of the corresponding result shown in the previous sections. □

As far as the verification algorithm is concerned, it is sufficient to add the two modifications illustrated in Section 3.1.1 to the reworking of the algorithm for $\sim_{\mathrm{MT}}$ described in the previous section. The unique additional relaxation concerns the final condition, which becomes $|v\ [\eta_1\ 0_{n_2}]^T + v'\ [0_{n_1}\ -\eta_2]^T| \leq \nu$. Hence, the overall complexity is the same as that determined in Section 3.3.1.

4. Conclusions and related work

In the setting of approximate notions of behavioral equivalences, two alternative research lines emerged in the formal methods community, which we divide into pseudometric-based approaches and (intransitive) relation-based approaches. In both cases, the typical notion of equivalence that is relaxed is bisimulation. The main difficulties behind the definition of the approximation concern the tradeoff between efficiency of the verification algorithm and interpretation of the obtained distance in terms of, e.g., influence of the degree of similarity on the observable differences between the quantitative profiles of the models.

In the pseudometric approach, a function d, inspired by Hutchinson-like metrics on probability measures, is defined that yields a (real number) distance for the models, say P and Q, such that some good properties are preserved like, e.g., $d(P, Q) = 0$ if and only if P and Q are equivalent and d satisfies the triangular inequality, i.e. $d(P, Q) \leq (P, R) + (R, Q)$. Then, the classical logical characterization of bisimulation can be turned into an alternative characterization using a specific set of functions into the reals instead of the logic. Two models are bisimilar if and only if they satisfy the same logical formulas, if and only if they have the same values for each functional expression of the set. In the case they are not bisimilar, the set of functional expressions induces a distance function d with the good properties mentioned above. This idea is formalized, e.g., in [11,18].

While these approaches provide interesting results in terms of, e.g., non-expansiveness with respect to process combinators like parallel composition (non-expansiveness is an analogue of the congruence property of bisimulation), they suffer from some practical limitations. For instance, the pseudometrics provide a distance between process states, but do not suggest which pairs of states would be worth comparing. This is because the process states are not compared through any relation relaxing bisimulation. Moreover, it is not easy to establish a clear relation between the measure estimating similarity and its interpretation in a practical, activity oriented setting.

Other approaches rely on relations approximating bisimulation equivalence (see, e.g., [4,12,13]). These relations cannot be transitive and, for this reason, their investigation did not receive attention for many years. However, they can offer an interesting framework for real application domains.

For instance, [4] proposes an intuitive relaxation of weak probabilistic bisimulation, which is in direct relation with approximate lumping for Markov chains. The characterization of lumpability is useful, because the knowledge of a lumpable partition of the states of a Markov chain allows the generation of a (smaller) aggregated Markov chain that leads to several results for the original one without an error. In this setting, approaches that rely on perturbation theory establish bounds on the error made when approximating lumpability. These bounds are related to the numerical analysis of Markov chains and, therefore, provide a clear interpretation of their impact upon the performance behavior of a system. On the other hand, meta-heuristics search techniques are needed to make the verification algorithm of approximate weak probabilistic bisimulation tractable in practice.

Similarly, [12] defines approximate bisimulation for probabilistic processes with logic-based and game-theoretic characterizations, a poly-time verification algorithm, but strong usability limitations with respect to its aggregation power. As another example, [13] introduces a relation approximating bisimulation in a framework in which the distance between processes is measured in terms of the norm of a linear operator applied to a matrix representation of the processes with respect to a classification operator based on the approximating relation. The computation of this relation is efficient, but the measure strictly depends on the chosen norms and classification linear operators, with an impact on the interpretation of the measure that is not completely intuitive.

In this paper, we have shown that in the framework of testing equivalence for Markovian processes it is possible to define several notions of approximation that meet good properties and can be verified efficiently. In order to compare these results with previous approaches, it is necessary to consider a notion of bisimulation for Markovian processes, which should be relaxed according to the orthogonal strategies proposed in this paper. In this case, it would be interesting to confirm that, as expected, the notions of approximate Markovian bisimulation are strictly finer than the approximate variants of Markovian testing equivalence. Finally, it is left as a future work the study of possible logic-based characterizations and, at least, sound axiomatizations of approximate Markovian testing equivalence.

Appendix. Approximate Markovian ready equivalence

The results concerning the verification algorithms are related to Markovian ready equivalence. In this section, we show how to approximate it, which represents a necessary condition to reduce the problem of deciding the approximate versions of Markovian testing equivalence to a probabilistic language equivalence problem (as shown in Section 3.1.1).

Definition A.1. Let $P \in \mathcal{P}$, $c \in \mathcal{C}_{\mathrm{f}}(P)$, and $\alpha \in (Name_{\mathrm{v}})^*$. We say that c is compatible with α iff:

$$trace(c) = \alpha.$$

We denote with $\mathcal{CC}(P, \alpha)$ the multiset of computations in $\mathcal{C}_{\mathrm{f}}(P)$ that are compatible with α. □

Definition A.2. Let $P \in \mathcal{P}$, $c \in \mathcal{C}_{\mathrm{f}}(P)$, and $\rho \equiv (\alpha, R) \in (Name_{\mathrm{v}})^* \times 2^{Name_{\mathrm{v}}}$. We say that computation c is compatible with the ready pair ρ iff $c \in \mathcal{CC}(P, \alpha)$ and the set of names of visible actions that can be performed by the last state reached by c coincides with the ready set R. We denote with $\mathcal{RCC}(P, \rho)$ the multiset of computations in $\mathcal{C}_{\mathrm{f}}(P)$ that are compatible with ρ. □

Definition A.3. Let $P_1, P_2 \in \mathcal{P}$. We say that P_1 is Markovian ready equivalent to P_2, written $P_1 \sim_{\mathrm{MR}} P_2$, iff for all ready pairs $\rho \in (Name_{\mathrm{v}})^* \times 2^{Name_{\mathrm{v}}}$ and sequences $\theta \in (\mathbb{R}_{>0})^*$ of average amounts of time:

$$prob(\mathcal{RCC}^{|\theta|}_{\leq\theta}(P_1, \rho)) = prob(\mathcal{RCC}^{|\theta|}_{\leq\theta}(P_2, \rho)). \quad \square$$

In the following, we show how to relax $\sim_{\mathrm{MR}}$ with respect to $\sim^{\epsilon}_{\mathrm{MT,slow}}$. The case of $\sim^{\epsilon}_{\mathrm{MT,fast}}$ follows symmetrically as expected.

Definition A.4. Let $P_1, P_2 \in \mathcal{P}$. We say that P_2 is a slow Markovian ready approximation of P_2, written $P_1 \sim^{\epsilon}_{\mathrm{MR,slow}} P_2$, iff for all ready pairs $\rho \in (Name_{\mathrm{v}})^* \times 2^{Name_{\mathrm{v}}}$ and sequences $\theta \in (\mathbb{R}_{>0})^*$ of average amounts of time:

$$prob(\mathcal{RCC}^{|\theta|}_{\leq\theta}(P_1, \rho)) = prob(\mathcal{RCC}^{|\theta|}_{\leq\theta+\epsilon, \mathcal{RCC}^{|\theta|}(P_1,\rho)}(P_2, \rho)). \quad \square$$

Proposition A.5. *Let $P_1, P_2 \in \mathcal{P}$. Then, $P_1 \sim^{\epsilon}_{\mathrm{MR,slow}} P_2 \Leftrightarrow P_1 \sim^{\epsilon}_{\mathrm{MT,slow}} P_2$.*

Proof. $\Rightarrow$) The result derives by virtue of the alternative characterization mentioned in the proof of Proposition 3.9 and of the corresponding result of [5]. $\Leftarrow$) The result derives by virtue of Lemma 3.4 and of the corresponding result of [5]. $\square$

In the case of $\sim^{p,r}_{\mathrm{MT},\phi}$, we can follow the same intuition if we assume a notion of precision and recall for ready pairs that derives directly from their test-based counterparts.

Given a ready pair $\rho \equiv (\alpha, R)$, we assume that $|\rho|$ denotes the length of the string α and $\rho(i)$ is the i-th action name occurring in α. Then, the precision and recall functions for ready pairs are defined as:

$$prec(\rho, \rho') = \frac{1}{|\rho'|} \sum_{i=1}^{|\rho'|} |(\rho(i) \cap \rho'(i))|$$

$$rec(\rho, \rho') = \frac{1}{|\rho|} \sum_{i=1}^{|\rho|} |(\rho(i) \cap \rho'(i))|.$$

Proposition A.6. *Let $\rho \equiv (\alpha, R)$ and $\rho' \equiv (\alpha', R')$ be two ready pairs and T, T' be two canonical reactive tests. If $trace(T, s) = \alpha$ and $trace(T', s) = \alpha'$ then $prec(\rho, \rho') = prec(T, T')$ and $rec(\rho, \rho') = rec(T, T')$.*

Proof. It is an immediate consequence of the definitions of the precision and recall functions. $\square$

References

[1] A. Aldini, Approximate testing equivalence based on time, probability, and observed behavior, in: Int. Workshop on Quantitative Aspects of Programming Languages, QAPL'10, in: EPTCS, vol. 28, 2010, pp. 1–15.
[2] A. Aldini, M. Bernardo, A formal approach to the integrated analysis of security and QoS, Journal of Reliability Engineering & System Safety 92 (11) (2007) 1503–1520.
[3] A. Aldini, M. Bernardo, F. Corradini, A Process Algebraic Approach to Software Architecture Design, Springer, 2010.
[4] A. Aldini, A. Di Pierro, Estimating the maximum information leakage, Journal of Information Security 7 (2008) 219–242.
[5] M. Bernardo, Markovian testing equivalence and exponentially timed internal actions, in: Int. Workshop on Quantitative Formal Methods, QFM 2009, in: EPTCS, vol. 13, 2009, pp. 13–25.
[6] C. Canal, E. Pimentel, J.M. Troya, Compatibility and inheritance in software architectures, Science of Computer Programming 41 (2001) 105–138.
[7] R. Cleaveland, O. Sokolsky, Equivalence and preorder checking for finite-state systems, in: Handbook of Process Algebra, Elsevier, 2001, pp. 391–424.
[8] R. De Nicola, D. Latella, M. Loreti, M. Massink, Rate-based transition systems for stochastic process calculi, in: Int. Colloquium on Automata, Languages and Programming, ICALP'09, in: LNCS, vol. 5556, 2009, pp. 435–446.
[9] A.K.A. de Medeiros, W.M.P. van der Aalst, A.J.M.M. Weijters, Quantifying process equivalence based on observed behavior, Data & Knowledge Engineering 64 (2008) 55–74.
[10] M. de Rougemont, M. Tracol, Static analysis for probabilistic processes, in: Int. Symp. on Logic in Computer Science, LICS'09, IEEE-CS, 2009, pp. 299–308.
[11] J. Desharnais, V. Gupta, R. Jagadeesan, P. Panangaden, Metrics for labelled markov processes, Theoretical Computer Science 318 (2004) 323–354.
[12] J. Desharnais, F. Laviolette, M. Tracol, Approximate analysis of probabilistic processes: logic, simulation and games, in: Int. Conf. on Quantitative Evaluation of Systems, QEST'08, IEEE-CS, 2008, pp. 264–273.
[13] A. Di Pierro, C. Hankin, H. Wiklicky, Quantifying timing leaks and cost optimisation, in: Conf. on Information and Comm. Security, ICICS'08, in: LNCS, vol. 5308, Springer, 2008, pp. 81–96.
[14] C.A.R. Hoare, Communicating Sequential Processes, Prentice Hall, 1985.
[15] D.T. Huynh, L. Tian, On some equivalence relations for probabilistic processes, Fundamenta Informaticae 17 (1992) 211–234.
[16] B. Klin, V. Sassone, Structural operational semantics for stochastic process calculi, in: Int. Conf. on Foundations of Software Science and Computational Structures, FOSSACS'08, in: LNCS, vol. 4962, Springer, 2008, pp. 428–442.
[17] W.G. Tzeng, A polynomial-time algorithm for the equivalence of probabilistic automata, SIAM Journal on Computing 21 (1994) 216–227.
[18] F. van Breugel, J. Worrell, A behavioural pseudometric for probabilistic transition systems, Theoretical Computer Science 331 (2005) 115–142.