Available online at www.sciencedirect.com
ScienceDirect
ELSEVIER
Electronic Notes in Theoretical Computer Science 253 (2009) 3–23
Electronic Notes in Theoretical Computer Science
www.elsevier.com/locate/entcs

Uniform Logical Characterizations of Testing Equivalences for Nondeterministic, Probabilistic and Markovian Processes

Marco Bernardo

Università di Urbino "Carlo Bo" – Italy
Istituto di Scienze e Tecnologie dell'Informazione

Abstract

Logical characterizations of nondeterministic, probabilistic, and Markovian variants of bisimulation equivalence rely on similar modal languages, each including true, negation, conjunction, and diamond. Likewise, logical characterizations of the corresponding variants of trace equivalence rely on similar modal languages, each including only true and diamond. Unfortunately, this is not the case with the existing logical characterizations of the corresponding variants of testing equivalence, as they are based on different modal languages. In this paper we show that the logical characterizations of testing equivalences for fully nondeterministic processes, fully probabilistic processes, and fully Markovian processes without silent moves can be harmonized by means of a modal language comprising true, disjunction, and diamond.

Keywords: testing equivalence, modal logic, nondeterministic processes, probabilistic processes, Markovian processes.

1 Introduction

Behavioral equivalences [14] establish whether computing systems possess the same behavioral properties. The specific set of properties that are preserved by a specific behavioral equivalence clearly depends on how the system behavior is observed and can usually be characterized by means of a modal logic.

In [3] we considered three different approaches to the definition of behavioral equivalences – bisimulation [23], testing [12], and trace [6] – applied to three different classes of processes – fully nondeterministic, fully probabilistic, and fully Markovian. Then we surveyed the nine resulting modal logic characterizations, each of which relies on a subset of (possibly modified) operators of the Hennessy-Milner logic (HML) [17]: true, negation, conjunction, and diamond.

The survey was the basis for a comparative study of the involved modal languages. The comparison is shown in Fig. 1, where on the horizontal axis we have the three approaches to the definition of behavioral equivalences, while on the ver-

doi:10.1016/j.entcs.2009.10.002

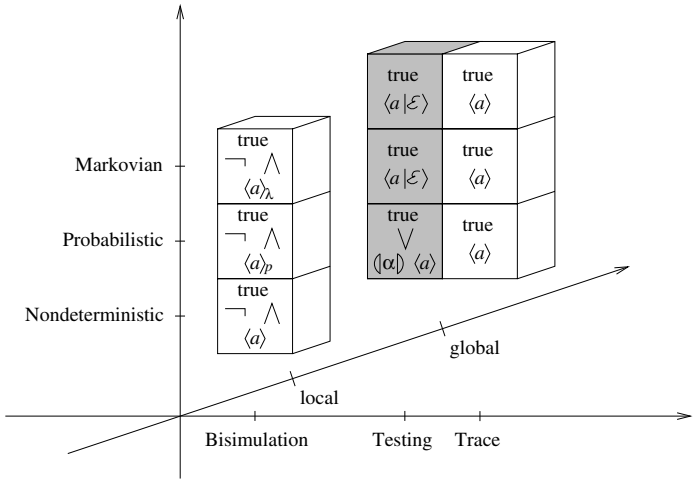

Fig. 1. Comparison of logical characterizations for behavioral equivalences

tical axis we have the three classes of processes. The figure emphasizes both differences across the three approaches to the definition of behavioral equivalences (inter-column comparison) and regularities within some of the three approaches (intra-column comparisons).

The inter-column comparison reveals that, when moving from bisimulation equivalence to trace equivalence, the number of logical operators that are needed tends to diminish, in accordance with the decreasing distinguishing power of the three approaches. More precisely, in the case of bisimulation equivalence we have all the logical operators of HML, then negation is dropped in the case of testing equivalence, and finally the binary operator is left out as well in the case of trace equivalence.

Another inter-column difference is related to probabilistic and Markovian processes and has to do with the treatment of probabilistic and temporal information, which can be local or global as shown by the third axis in Fig. 1.

In a local characterization, probabilistic and temporal aspects are considered at the level of each individual action occurring in the computations. Thus probabilistic and temporal constraints have to be attached to individual logical operators and the interpretation of the resulting formulas is qualitative as usual, i.e. it returns a truth value. This is the case with probabilistic and Markovian bisimulation equivalences where the diamond operator is decorated with a probability/rate lower bound [21,8], consistently with the fact that the notion of bisimulation captures step-by-step behavior mimicking.

In a global characterization, instead, probabilistic and temporal aspects are considered at the level of the overall computations. Thus probabilities and durations of entire computations become of interest and the interpretation of the usual nondecorated formulas is quantitative, in the sense that it returns a number that measures how much a formula is satisfied. This is the case with probabilistic and Markovian testing and trace equivalences [20,3].

On the other hand, intra-column comparisons reveal regularities with respect to the involved logical operators among the three variants of bisimulation equivalence

(up to the diamond decorations discussed before) and among the three variants of trace equivalence.

In particular, the three logical characterizations of bisimulation equivalence rely on true, negation, conjunction, and diamond. More precisely, negation occurring in the logical characterization of [22] for the probabilistic case is not necessary as long as the considered probabilistic processes are reactive [13], but it has to be reintroduced as soon as the considered probabilistic processes admit nondeterminism [24]. [1] The logical characterization of [8] for the Markovian case includes an additional operator for asserting that actions with a certain name cannot be executed, which is clearly unnecessary in the presence of negation (and hence is not shown in Fig. 1).

Unfortunately, this regularity disappears in the case of testing equivalence (see the grey column). For nondeterministic processes, the logical characterization provided by [16] captures a broom semantics, as the related modal language permits to ask simple questions after a certain trace has been executed. Therefore, the syntax of the modal language has a two-layer definition. In the top layer we have a modal operator on traces. In the bottom layer we have true, disjunction, and diamond with no continuation. Then two satisfaction relations are defined, which express that a process may or must satisfy a certain bottom-layer formula after executing a certain trace. By contrast, for probabilistic and Markovian processes, the logical characterizations of testing equivalence provided by [3] – on the basis of the original equivalence definition of [7] and of the fully abstract characterizations of [9,2] – rely on true and a conditional variant of diamond.

This paper addresses the problem of reconciling the logical characterizations for the three testing equivalences by means of a single modal language.

A way of tackling this problem is to exploit the relation between testing equivalence and failure equivalence [11] – which extends to ready equivalence in the probabilistic and Markovian cases [19,2] – and hence using the logical characterization of the latter. This comprises true, diamond, and a failure predicate, i.e. a predicate for expressing the fact that actions whose name belongs to a given set cannot be executed [14].

However, we deem this solution not to be satisfactory because it disrupts the direct connection with HML operators. In other words, what we are searching for is a uniform modal language for the three variants of testing equivalence built from a suitable subset of (direct and dual) operators of HML, so as to preserve the inter-column decreasing trend shown in Fig. 1 for the operator set.

Since testing equivalence is finer than trace equivalence and the latter is characterized by true and diamond, those two operators will have to be part of the uniform modal language for testing equivalence. While trace equivalence is a linear-time relation, testing equivalence is sensitive to branching points, hence a binary operator will be necessary too. The logical characterization of [16] seems to suggest the inclusion of disjunction rather than conjunction. Including conjuction would in fact

[1] Should probability and nondeterminism coexist in the same states, a further operator is even necessary.

provide a different distinguishing power, as the modal language formed by true, diamond, and conjunction characterizes simulation equivalence [17], which is independent of testing equivalence [14].

In this paper we demonstrate that a uniform modal language for the three variants of testing equivalence that retains a direct connection with HML exists and is precisely formed by true, disjunction, and diamond. With respect to [16], the language is no more concerned with traces but still keeps a may-satisfaction relation and a must-satisfaction relation. The latter has to be carefully defined for disjunction and diamond, in order to be consistent with the notion of having to pass a test. In the probabilistic and Markovian cases, a syntax restriction has to be imposed on formulas of the form $\phi_1 \vee \phi_2$, which requires the set of initial actions of ϕ_1 to be disjoint from the set of initial actions of ϕ_2. This constraint is necessary for a correct and easier computation of the probability of satisfying such formulas. For the sake of simplicity and uniformity, we restrict ourselves to finite-state processes without silent moves and we consider only their finite-length test-driven computations.

This paper is organized as follows. In Sect. 2 we recall the definitions of testing equivalence for fully nondeterministic, fully probabilistic, and fully Markovian finite-state processes without silent moves. In Sect. 3, 4, and 5 we provide three alternative logical characterizations for the three testing equivalences, each based on the uniform modal language mentioned above. Finally, in Sect. 6 we provide some concluding remarks as well as some comments on the presence of silent moves and the coexistence of nondeterminism and probabilistic/temporal aspects.

2 Testing Equivalences

In this section we present the definitions of the three variants of testing equivalence in a process algebraic setting. Firstly we introduce three calculi based on the same set *Name* of observable action names and the same set of behavioral operators – the inactive term, the action prefix operator, the alternative composition operator, and recursion – which generate all the fully nondeterministic, fully probabilistic, and fully Markovian finite-state processes. For the sake of simplicity, silent moves are not admitted within such processes, as it is not possible to abstract from them in the Markovian case.

Secondly, we recall the notions of execution probability and stepwise average duration for finite-length computations. Since in the Markovian testing case the presence of average time upper bounds makes infinite computations unimportant, for the sake of uniformity we focus only on finite-length test-driven computations for all three kinds of processes. As a consequence, since the behavior of a process under test is observed for a finite amount of time, we restrict ourselves to nonrecursive, finite-state tests for all three kinds of processes.

Thirdly, we describe the structure of tests and interaction systems. In order not to interfere with the quantitative aspects of the behavior of probabilistic and Markovian processes under test, we avoid the introduction of a success action ω. For the sake of uniformity, for all three kinds of processes the successful completion

of a test is formalized in the text syntax by replacing the inactive term with a zeroary operator denoting a success state. A failure state is not necessary as it can be encoded through an action not occurring in the processes under test.

2.1 Nondeterministic, Probabilistic, and Markovian Processes

In the nondeterministic process calculus (NPC for short), the choice among all the actions that are simultaneously enabled is nondeterministic. We denote by $Act_{\rm N} = Name$ the set of actions of NPC, which are all observable.

Definition 2.1 The set of process terms of NPC is generated by the following syntax:

$$\left| P ::= \underline{0} \mid a.P \mid P + P \mid A \right|$$

where $\underline{0}$ is the inactive term, $a \in Act_{\rm N}$, and A is a process constant defined through the (possibly recursive) equation $A \stackrel{\Delta}{=} P$.

The semantics for NPC can be provided in the usual operational style by associating a labeled transition system with each process term. The transition relation $\longrightarrow_{\rm N}$ for the set $\mathbb{P}_{\rm N}$ of closed and guarded terms of NPC is defined as the least subset of $\mathbb{P}_{\rm N} \times Act_{\rm N} \times \mathbb{P}_{\rm N}$ satisfying the rules shown in the first column of Table 1.

In the probabilistic process calculus (PPC for short), every action is represented as a pair composed of the name of the action and the probability of executing the action. We denote by $Act_{\rm P} = Name \times \mathbb{R}_{]0,1]}$ the set of actions of PPC. In order to ensure that the sum of the probabilities of the actions that are simultaneously enabled is either 0 or 1, we replace the action prefix operator and the binary alternative composition operator with a set of n-ary guarded alternative composition operators, with n ranging over the whole $\mathbb{N}_{>0}$.

Definition 2.2 The set of process terms of PPC is generated by the following syntax:

$$\left| P ::= \underline{0} \mid \sum_{i \in I} <a_i, p_i>.P_i \mid A \right|$$

where I is a nonempty finite index set, $<a_i, p_i> \in Act_{\rm P}$ for all $i \in I$, and it holds $\sum_{i \in I} p_i = 1$.

The semantics for PPC can be defined in the usual operational style, provided that a labeled multitransition system is associated with each process term. In other words, the multiplicity of the transitions has to be taken into account, where by multiplicity we mean the number of different ways in which a transition can be derived by applying the operational semantic rules. The reason is that idempotency (i.e. $P + P = P$) no longer holds when moving from nondeterministic processes to probabilistic ones. As an example, a term like $<a, 0.5>.P + <a, 0.5>.P$ cannot be equated to $<a, 0.5>.P$, which can be achieved by observing that there are two different ways of deriving – through the operational semantic rules – a 0.5-probability transition labeled with a and reaching P (one for the left-hand side $<a, 0.5>.P$

summand plus another one for the right-hand side $<a, 0.5>.P$ summand). The multitransition relation $\longrightarrow_{\rm P}$ for the set $\mathbb{P}_{\rm P}$ of closed and guarded terms of PPC is defined as the least multiset of elements of $\mathbb{P}_{\rm P} \times Act_{\rm P} \times \mathbb{P}_{\rm P}$ satisfying the rules shown in the second column of Table 1. The stochastic process underlying a labeled multitransition system produced by the application of these rules turns out to be a discrete-time Markov chain.

In the Markovian process calculus (MPC for short), every action is represented as a pair composed of the name of the action and the rate of the exponential distribution quantifying the duration of the action. We denote by $Act_{\rm M} = Name \times \mathbb{R}_{>0}$ the set of actions of MPC. The choice among all the actions that are simultaneously enabled is governed by the race policy. As a consequence, the execution probability of each action is proportional to its rate and the average sojourn time in the state associated with a process term is quantified by an exponentially distributed random variable whose rate is the sum of the rates of the enabled actions.

Definition 2.3 The set of process terms of MPC is generated by the following syntax:

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

where $<a, \lambda> \in Act_{\rm M}$.

Similarly to PPC, idempotency no longer holds and hence the multiplicity of the transitions has to be taken into account by associating a labeled multitransition system with each process term. As an example, a term like $<a, 4.6>.P + <a, 4.6>.P$ is not equivalent to $<a, 4.6>.P$ but to $<a, 9.2>.P$, because rates sum up due to the race policy. The multitransition relation $\longrightarrow_{\rm M}$ for the set $\mathbb{P}_{\rm M}$ of closed and guarded terms of MPC is defined as the least multiset of elements of $\mathbb{P}_{\rm M} \times Act_{\rm M} \times \mathbb{P}_{\rm M}$ satisfying the rules shown in the third column of Table 1. The stochastic process underlying a labeled multitransition system produced by the application of these rules turns out to be a continuous-time Markov chain.

In the following, we denote by $init(P)$ the set of names of actions initially enabled by $P \in \mathbb{P}_{\rm N} \cup \mathbb{P}_{\rm P} \cup \mathbb{P}_{\rm M}$.

2.2 Computations

A computation of a process term $P \in \mathbb{P}_{\rm N} \cup \mathbb{P}_{\rm P} \cup \mathbb{P}_{\rm M}$ is a sequence of transitions that can be executed starting from P. The length of a computation is given by the number of transitions occurring in it. We say that two distinct computations are independent of each other if neither is a proper prefix of the other one. We denote by $\mathcal{C}_{\rm f}(P)$ the multiset of finite-length computations of P and by $\mathcal{I}_{\rm f}(P)$ the multiset of maximal finite-length computations of P. Moreover, we denote by:

$$rate_{\rm t}(P) = \sum\{\!|\, \lambda \in \mathbb{R}_{>0} \mid \exists a, P'.\, P \xrightarrow{a,\lambda}_{\rm M} P' \,|\!\}$$

$$prob_{\rm c}(P|N) = \sum_{a \in N} \{\!|\, p \in \mathbb{R}_{]0,1]} \mid \exists P'.\, P \xrightarrow{a,p}_{\rm P} P' \,|\!\}$$

$$rate_{\rm c}(P|N) = \sum_{a \in N} \{\!|\, \lambda \in \mathbb{R}_{>0} \mid \exists P'.\, P \xrightarrow{a,\lambda}_{\rm M} P' \,|\!\}$$

$a.P \xrightarrow{a}_{\mathrm{N}} P$	$\sum_{i \in I} <a_i, p_i>.P_i \xrightarrow{a_j, p_j}_{\mathrm{P}} P_j, \quad j \in I$	$<a, \lambda>.P \xrightarrow{a,\lambda}_{\mathrm{M}} P$
$P_1 \xrightarrow{a}_{\mathrm{N}} P'$ $P_1 + P_2 \xrightarrow{a}_{\mathrm{N}} P'$		$P_1 \xrightarrow{a,\lambda}_{\mathrm{M}} P'$ $P_1 + P_2 \xrightarrow{a,\lambda}_{\mathrm{M}} P'$
$P_2 \xrightarrow{a}_{\mathrm{N}} P'$ $P_1 + P_2 \xrightarrow{a}_{\mathrm{N}} P'$		$P_2 \xrightarrow{a,\lambda}_{\mathrm{M}} P'$ $P_1 + P_2 \xrightarrow{a,\lambda}_{\mathrm{M}} P'$
$P \xrightarrow{a}_{\mathrm{N}} P' \quad A \triangleq P$ $A \xrightarrow{a}_{\mathrm{N}} P'$	$P \xrightarrow{a,p}_{\mathrm{P}} P' \quad A \triangleq P$ $A \xrightarrow{a,p}_{\mathrm{P}} P'$	$P \xrightarrow{a,\lambda}_{\mathrm{M}} P' \quad A \triangleq P$ $A \xrightarrow{a,\lambda}_{\mathrm{M}} P'$

Table 1
Semantic rules for $\mathbb{P}_{\mathrm{N}}$, $\mathbb{P}_{\mathrm{P}}$, and $\mathbb{P}_{\mathrm{M}}$

the total rate of $P \in \mathbb{P}_{\mathrm{M}}$ (which corresponds to the reciprocal of the average sojourn time in the state associated with P), the conditional probability for $P \in \mathbb{P}_{\mathrm{P}}$ of executing actions whose name belongs to $N \subseteq Name$, and the conditional rate for $P \in \mathbb{P}_{\mathrm{M}}$ of executing actions whose name belongs to $N \subseteq Name$, respectively.

Definition 2.4 Let $P \in \mathbb{P}_{\mathrm{P}} \cup \mathbb{P}_{\mathrm{M}}$ 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 following $\mathbb{R}_{]0,1]}$-valued function:

$$prob(c) \;=\; \begin{cases} 1 & \text{if } length(c) = 0 \\ p \cdot prob(c') & \text{if } c \equiv P \xrightarrow{a,p}_{\mathrm{P}} c' \text{ with } P \in \mathbb{P}_{\mathrm{P}} \\ \underset{rate_{\mathrm{t}}(P)}{\lambda} \cdot prob(c') & \text{if } c \equiv P \xrightarrow{a,\lambda}_{\mathrm{M}} c' \text{ with } P \in \mathbb{P}_{\mathrm{M}} \end{cases}$$

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

$$\left| prob(C) \;=\; \sum_{c \in C} prob(c) \right|$$

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

Definition 2.5 Let $P \in \mathbb{P}_{\mathrm{M}}$ and $c \in \mathcal{C}_{\mathrm{f}}(P)$. The stepwise average duration of c is the sequence of the average sojourn times in the states traversed by c, which is defined by induction on the length of c through the following $(\mathbb{R}_{>0})^*$-valued function:

$$time(c) \;=\; \begin{cases} \varepsilon & \text{if } length(c) = 0 \\ \underset{rate_{\mathrm{t}}(P)}{1} \circ time(c') & \text{if } c \equiv P \xrightarrow{a,\lambda}_{\mathrm{M}} c' \end{cases}$$

where ε is the empty sequence and $\circ$ is the concatenation operator. We also define the multiset of computations in $C \subseteq \mathcal{C}_{\mathrm{f}}(P)$ whose stepwise average duration is not

greater than $\theta \in (\mathbb{R}_{>0})^*$ as:

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

2.3 Tests and Interaction Systems

The most convenient way to represent a test is through another process term, which interacts with the term under test by means of a parallel composition operator that enforces synchronization on all action names. We denote by s the success state of a test. Ambiguous tests including several summands among which at least one equal to s are avoided through a two-level syntax.

In the nondeterministic case, tests are made out of the same kind of actions that can occur in nondeterministic process terms.

Definition 2.6 The set $\mathbb{T}_{\mathrm{N}}$ of nondeterministic tests is generated by the following syntax:

$$\left| \begin{array}{rcl} T & ::= & \mathrm{s} \mid T' \\ T' & ::= & a.T \mid T' + T' \end{array} \right|$$

where $a \in Name$.

The following operational rule defines the interaction of $P \in \mathbb{P}_{\mathrm{N}}$ and $T \in \mathbb{T}_{\mathrm{N}}$:

$$\left| \begin{array}{c} P \stackrel{a}{\longrightarrow}_{\mathrm{N}} P' \quad T \stackrel{a}{\longrightarrow}_{\mathrm{N}} T' \\ P \parallel T \stackrel{a}{\longrightarrow}_{\mathrm{N}} P' \parallel T' \end{array} \right|$$

In the probabilistic and Markovian cases, instead, tests are made out of passive actions, each equipped with a weight $w \in \mathbb{R}_{>0}$. The idea is that, in any of its states, a process term under test generates the proposal of an action to be executed by probabilistically selecting one of the actions enabled in that state. Then the test reacts by probabilistically selecting a passive action (if any) with the same name as the proposed action.

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

$$\left| \begin{array}{rcl} T & ::= & \mathrm{s} \mid T' \\ T' & ::= & <a, *_w>.T \mid T' + T' \end{array} \right|$$

where $a \in Name$ and $w \in \mathbb{R}_{>0}$.

Let us denote by $\longrightarrow_{\mathrm{R}}$ the multitransition relation for reactive tests. According to the terminology of [15], the following operational rule defines the probabilistic generative-reactive interaction [5] of $P \in \mathbb{P}_{\mathrm{P}}$ and $T \in \mathbb{T}_{\mathrm{R}}$:

$$\frac{P \stackrel{a,p}{\longrightarrow}_{\mathrm{P}} P' \quad T \stackrel{a,*_w}{\longrightarrow}_{\mathrm{R}} T'}{P \parallel T \stackrel{a,\, prob_{\mathrm{c}}(P \mid init(T)) \cdot weight(T,a)}{\longrightarrow}_{\mathrm{P}} P' \parallel T'}$$

where $init(T) = \{a \in Name \mid \exists w, T'.\, T \xrightarrow{a,*_w}_{\mathrm{R}} T'\}$ is the set of names of actions enabled by T and $weight(T,a) = \sum\{\!|\, w \in \mathbb{R}_{>0} \mid \exists T'.\, T \xrightarrow{a,*_w}_{\mathrm{R}} T' \,|\!\}$ is the weight of T with respect to a. Note that, due to the normalization taking place in the conclusion of the rule above, if $P \parallel T$ has outgoing transitions, then the probabilities of those transitions sum up to 1.

Likewise, the following operational rule defines the Markovian generative-reactive interaction [4] of $P \in \mathbb{P}_{\mathrm{M}}$ and $T \in \mathbb{T}_{\mathrm{R}}$, which preserves the conditional rate of P with respect to $init(T)$ due to the normalization taking place in the conclusion of the rule:

$$\frac{P \xrightarrow{a,\lambda}_{\mathrm{M}} P' \qquad T \xrightarrow{a,*_w}_{\mathrm{R}} T'}{P \parallel T \xrightarrow{a,\lambda \cdot \frac{w}{weight(T,a)}}_{\mathrm{M}} P' \parallel T'}$$

Definition 2.8 Let $P \in \mathbb{P}_{\mathrm{N}}$ and $T \in \mathbb{T}_{\mathrm{N}}$, or $P \in \mathbb{P}_{\mathrm{P}} \cup \mathbb{P}_{\mathrm{M}}$ and $T \in \mathbb{T}_{\mathrm{R}}$. The interaction system of P and T is process term $P \parallel T$, where we say that:

- A configuration is a state of the labeled (multi)transition system underlying $P \parallel T$.
- A configuration is successful iff its test component is s.
- A computation is successful iff so is its last configuration.

We denote by $\mathcal{SC}(P,T)$ the multiset of successful computations in $\mathcal{C}_{\mathrm{f}}(P \parallel T)$.

All the computations in $\mathcal{SC}(P,T)$ are independent of each other because of the maximality of successful test-driven computations. Moreover, $\mathcal{SC}(P,T)$ is finite because of the finitely-branching structure of the considered terms.

2.4 Nondeterministic Testing Equivalence

Two nondeterministic processes are testing equivalent if they are indistinguishable with respect to the possibility and the necessity of passing an arbitrary test [12].

Definition 2.9 Let $P \in \mathbb{P}_{\mathrm{N}}$ and $T \in \mathbb{T}_{\mathrm{N}}$. We say that:

- P may pass T iff at least one computation of $P \parallel T$ is successful:
$$\mathcal{SC}(P,T) \neq \emptyset$$
- P must pass T iff all maximal computations of $P \parallel T$ are successful:
$$\mathcal{SC}(P,T) = \mathcal{I}_{\mathrm{f}}(P \parallel T)$$

Definition 2.10 Let $P_1, P_2 \in \mathbb{P}_{\mathrm{N}}$. We say that P_1 is nondeterministic testing equivalent to P_2, written $P_1 \sim_{\mathrm{NT}} P_2$, iff for all nondeterministic tests $T \in \mathbb{T}_{\mathrm{N}}$:

$$P_1 \text{ may pass } T \iff P_2 \text{ may pass } T$$

$$P_1 \text{ must pass } T \iff P_2 \text{ must pass } T$$

2.5 Probabilistic Testing Equivalence

In the probabilistic case, the possibility and the necessity of passing an arbitrary test are subsumed by the probability of passing the test.

Definition 2.11 Let $P_1, P_2 \in \mathbb{P}_{\rm P}$. We say that P_1 is probabilistic testing equivalent to P_2, written $P_1 \sim_{\rm PT} P_2$, iff for all reactive tests $T \in \mathbb{T}_{\rm R}$:

$$prob(\mathcal{SC}(P_1, T)) \;=\; prob(\mathcal{SC}(P_2, T))$$

2.6 Markovian Testing Equivalence

In the Markovian case, we have to consider the probability of passing an arbitrary test within an arbitrary sequence of average amounts of time.

Definition 2.12 Let $P_1, P_2 \in \mathbb{P}_{\rm M}$. We say that P_1 is Markovian testing equivalent to P_2, written $P_1 \sim_{\rm MT} P_2$, iff for all reactive tests $T \in \mathbb{T}_{\rm R}$ and sequences $\theta \in (\mathbb{R}_{>0})^*$ of average amounts of time:

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

3 Alternative Logical Characterization of $\sim_{\rm NT}$

The modal language characterizing $\sim_{\rm NT}$ provided by [16] has a two-layer syntax. In the top layer there is a modal operator on traces, whereas in the bottom layer there are true, disjunction, and a modal operator on actions with no continuation. This language is interpreted by means of two satisfaction relations, which express that a process may or must perform certain actions after executing a certain trace.

In the following we show an alternative characterization based on a modal language with a syntax related to actions only that includes true, disjunction, and diamond. With respect to HML, this language is less powerful as it is devoid of negation. Moreover, it replaces conjunction with disjunction. This is unavoidable, as $\sim_{\rm NT}$ is the intersection of may-testing equivalence and must-testing equivalence, with the former coinciding with trace equivalence and hence not admitting conjunction in its characterization. We also observe that conjunction cannot simply be discarded but must be replaced, as having only true and diamond would not be enough to characterize must-testing equivalence.

Definition 3.1 The set of formulas of $\mathcal{ML}_{\rm T}$ is generated by the following syntax:

$$\left|\begin{array}{lcl} \phi & ::= & {\rm true} \mid \phi' \\ \phi' & ::= & \langle a\rangle\phi \mid \phi' \vee \phi' \end{array}\right|$$

where $a \in Name$.

Definition 3.2 The set $init(\phi)$ of names of actions initially occurring in a formula $\phi \in \mathcal{ML}_{\rm T}$ is defined by structural induction as follows:

$$\boxed{\begin{array}{rcl} init({\rm true}) & = & \emptyset \\ init(\phi_1 \vee \phi_2) & = & init(\phi_1) \cup init(\phi_2) \\ init(\langle a\rangle\phi) & = & \{a\} \end{array}}$$

The new modal language retains the two satisfaction relations, but the one for the must case has to be carefully defined for disjunction and diamond in order for it to be consistent with the notion of having to pass a test.

Definition 3.3 The satisfaction relation $\models_{\rm may}$ of $\mathcal{ML}_{\rm T}$ over $\mathbb{P}_{\rm N}$ is defined by structural induction as follows:

$$\begin{array}{lll}
P \models_{\rm may} \text{true} & & \\
P \models_{\rm may} \phi_1 \vee \phi_2 & \text{if } P \models_{\rm may} \phi_1 \text{ or } P \models_{\rm may} \phi_2 \\
P \models_{\rm may} \langle a \rangle \phi & \text{if there exists } P' \text{ such that } P \stackrel{a}{\longrightarrow}_{\rm N} P' \text{ and } P' \models_{\rm may} \phi
\end{array}$$

Definition 3.4 The satisfaction relation $\models_{\rm must}$ of $\mathcal{ML}_{\rm T}$ over $\mathbb{P}_{\rm N}$ is defined by structural induction as follows:

$$\begin{array}{lll}
P \models_{\rm must} \text{true} & & \\
P \models_{\rm must} \phi_1 \vee \phi_2 & \text{if } \mathit{init}(P) \cap (\mathit{init}(\phi_1) \cup \mathit{init}(\phi_2)) \neq \emptyset \\
& \quad \text{and } \mathit{init}(P) \cap \mathit{init}(\phi_1) \neq \emptyset \text{ implies } P \models_{\rm must} \phi_1 \\
& \quad \text{and } \mathit{init}(P) \cap \mathit{init}(\phi_2) \neq \emptyset \text{ implies } P \models_{\rm must} \phi_2 \\
P \models_{\rm must} \langle a \rangle \phi & \text{if there exists } P' \text{ such that } P \stackrel{a}{\longrightarrow}_{\rm N} P' \\
& \quad \text{and each such } P' \models_{\rm must} \phi
\end{array}$$

The intuition behind the definition of $\models_{\rm must}$ is that, given $P \in \mathbb{P}_{\rm N}$ and $\phi \in \mathcal{ML}_{\rm T} - \{\text{true}\}$, $\mathit{init}(P)$ must intersect $\mathit{init}(\phi)$ and P must satisfy ϕ along each of its computations starting with a transition labeled with an action name belonging to $\mathit{init}(P) \cap \mathit{init}(\phi)$.

A must-interpretation of disjunction like the following:

$$P \models_{\rm must} \phi_1 \vee \phi_2 \quad \text{if } P \models_{\rm must} \phi_1 \text{ or } P \models_{\rm must} \phi_2$$

would be wrong. As an example, it is not the case that $a.\underline{0} + b.\underline{0}$ must pass $a.{\rm s} + b.c.{\rm s}$ because of the unsuccessful maximal test-driven computation composed of transition b, but it would be $a.\underline{0} + b.\underline{0} \models_{\rm must} \langle a \rangle \text{true} \vee \langle b \rangle \langle c \rangle \text{true}$ because $a.\underline{0} + b.\underline{0} \models_{\rm must} \langle a \rangle \text{true}$.

Likewise, a must-interpretation of diamond like the following:

$$P \models_{\rm must} \langle a \rangle \phi \quad \text{if for all } P' \text{ whenever } P \stackrel{a}{\longrightarrow}_{\rm N} P' \text{ then } P' \models_{\rm must} \phi$$

would be wrong. As an example, it is not the case that $\underline{0}$ must pass $a.{\rm s}$ because $\mathcal{SC}(\underline{0}, a.{\rm s}) = \emptyset \neq \{\varepsilon\} = \mathcal{I}_{\rm f}(\underline{0} \parallel a.{\rm s})$, but it would trivially be $\underline{0} \models_{\rm must} \langle a \rangle \text{true}$ because there is no P' reachable from $\underline{0}$ via a.

We observe that our must-interpretation of diamond differs from the must-interpretation of the modal operator on traces defined in [16]. In fact, we additionally require the existence of at least one a-transition leaving P as a necessary condition for the must-satisfaction of $\langle a \rangle \phi$. By contrast, due to the two-layer syntax of the modal language, and consistently with an alternative characterization of [12], for the modal operator on traces it is simply required that P does not diverge when accepting the specified trace.

Lemma 3.5 *For all $T \in \mathbb{T}_{\rm N}$ there exists $\phi_T \in \mathcal{ML}_{\rm T}$ such that $init(\phi_T) = init(T)$ and for all $P \in \mathbb{P}_{\rm N}$:*

$$P \textit{ may pass } T \implies P \models_{\rm may} \phi_T$$

$$P \textit{ must pass } T \implies P \models_{\rm must} \phi_T$$

Lemma 3.6 *For all $\phi \in \mathcal{ML}_{\rm T}$ there exists $T_\phi \in \mathbb{T}_{\rm N}$ such that $init(T_\phi) = init(\phi)$ and for all $P \in \mathbb{P}_{\rm N}$:*

$$P \models_{\rm may} \phi \implies P \textit{ may pass } T_\phi$$

$$P \models_{\rm must} \phi \implies P \textit{ must pass } T_\phi$$

Theorem 3.7 *Let $P_1, P_2 \in \mathbb{P}_{\rm N}$. Then $P_1 \sim_{\rm NT} P_2$ iff for all $\phi \in \mathcal{ML}_{\rm T}$:*

$$P_1 \models_{\rm may} \phi \iff P_2 \models_{\rm may} \phi$$

$$P_1 \models_{\rm must} \phi \iff P_2 \models_{\rm must} \phi$$

4 Alternative Logical Characterization of $\sim_{\rm PT}$

A slight variant of $\mathcal{ML}_{\rm T}$ denoted by $\mathcal{ML}_{\rm T,ind}$ can be employed to characterize $\sim_{\rm PT}$. In order for probabilities to be computed correctly, in $\mathcal{ML}_{\rm T,ind}$ each formula of the form $\phi_1 \vee \phi_2$ must satisfy the constraint $init(\phi_1) \cap init(\phi_2) = \emptyset$. In this way, ϕ_1 and ϕ_2 are guaranteed to exercise independent computations of an arbitrary process term, hence the probability that the term satisfies $\phi_1 \vee \phi_2$ can essentially be computed as the sum of the probabilities of satisfying ϕ_1 and ϕ_2.

On the test side, this constraint amounts to work with the set $\mathbb{T}_{\rm R,det}$ of name-deterministic reactive tests, in which each test of the form $T_1 + T_2$ must satisfy $init(T_1) \cap init(T_2) = \emptyset$. From [7,9] it is known that the canonical reactive tests for $\sim_{\rm PT}$ are generated by the following syntax:

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

where $\mathcal{E}$ is a finite subset of *Name* including a, the summation disappears whenever $\mathcal{E} - \{a\} = \emptyset$, and z is a fresh action name that cannot occur in any process term. Since canonical reactive tests are name-deterministic, we can restrict ourselves to $\mathbb{T}_{\rm R,det}$ without loss of distinguishing power.

While $\mathcal{ML}_{\rm T}$ has been interpreted over $\mathbb{P}_{\rm N}$ through two qualitative satisfaction relations, $\mathcal{ML}_{\rm T,ind}$ will be equipped with a quantitative interpretation function in the spirit of [20]. This function measures the probability of satisfying a formula under the condition that – consistently with the testing approach – the formula itself establishes the names of the only actions that can be enabled.

Definition 4.1 The interpretation function $[\![.]\!]_{\rm PT}$ of $\mathcal{ML}_{\rm T,ind}$ over $\mathbb{P}_{\rm P}$ is defined by letting $[\![\phi]\!]_{\rm PT}(P) = 0$ for $\phi \in \mathcal{ML}_{\rm T,ind} - \{\mathrm{true}\}$ whenever $init(P) \cap init(\phi) = \emptyset$,

otherwise by structural induction as follows:

$$\begin{array}{rcl} [\![\mathrm{true}]\!]_{\mathrm{PT}}(P) & = & 1 \\ [\![\phi_1 \vee \phi_2]\!]_{\mathrm{PT}}(P) & = & p_1 \cdot [\![\phi_1]\!]_{\mathrm{PT}}(P) + p_2 \cdot [\![\phi_2]\!]_{\mathrm{PT}}(P) \\ [\![\langle a \rangle \phi]\!]_{\mathrm{PT}}(P) & = & \displaystyle\sum_{P \xrightarrow{a,p}_{\mathrm{P}} P'} prob_{\mathrm{c}}^{p}(P|\{a\}) \cdot [\![\phi]\!]_{\mathrm{PT}}(P') \end{array}$$

where $p_j = \frac{prob_{\mathrm{c}}(P|init(\phi_j))}{prob_{\mathrm{c}}(P|init(\phi_1 \vee \phi_2))}$ for $j \in \{1, 2\}$.

In the definition above, p_j represents the probability with which P performs actions whose name is in $init(\phi_j)$ rather than actions whose name is in $init(\phi_k)$, $k = 3 - j$, given that P can perform actions whose name is in $init(\phi_1 \vee \phi_2)$. These probabilities are used as weights for the correct account of the probabilities with which P satisfies ϕ_1 alone and ϕ_2 alone in the context of the satisfaction of $\phi_1 \vee \phi_2$. If such weights were omitted, then the fact that $\phi_1 \vee \phi_2$ offers a set of initial actions at least as large as the ones offered by ϕ_1 alone and ϕ_2 alone would be ignored, thus leading to a potential overestimate of the probability of satisfying $\phi_1 \vee \phi_2$.

Lemma 4.2 *For all $T \in \mathbb{T}_{\mathrm{R,det}}$ there exists $\phi_T \in \mathcal{ML}_{\mathrm{T,ind}}$ such that $init(\phi_T) = init(T)$ and for all $P \in \mathbb{P}_{\mathrm{P}}$:*

$$[\![\phi_T]\!]_{\mathrm{PT}}(P) \;=\; prob(\mathcal{SC}(P, T))$$

Lemma 4.3 *For all $\phi \in \mathcal{ML}_{\mathrm{T,ind}}$ there exists $T_\phi \in \mathbb{T}_{\mathrm{R,det}}$ such that $init(T_\phi) = init(\phi)$ and for all $P \in \mathbb{P}_{\mathrm{P}}$:*

$$prob(\mathcal{SC}(P, T_\phi)) \;=\; [\![\phi]\!]_{\mathrm{PT}}(P)$$

Theorem 4.4 *Let $P_1, P_2 \in \mathbb{P}_{\mathrm{P}}$. Then $P_1 \sim_{\mathrm{PT}} P_2$ iff for all $\phi \in \mathcal{ML}_{\mathrm{T,ind}}$:*

$$[\![\phi]\!]_{\mathrm{PT}}(P_1) \;=\; [\![\phi]\!]_{\mathrm{PT}}(P_2)$$

5 Alternative Logical Characterization of $\sim_{\mathrm{MT}}$

The same constrained version $\mathcal{ML}_{\mathrm{T,ind}}$ of $\mathcal{ML}_{\mathrm{T}}$ can be used for $\sim_{\mathrm{MT}}$ so as to compute probabilities correctly. Since the canonical reactive tests for $\sim_{\mathrm{MT}}$ are the same as those for $\sim_{\mathrm{PT}}$ [2], also in this case we can restrict ourselves to $\mathbb{T}_{\mathrm{R,det}}$ without loss of distinguishing power. The difference with respect to the probabilistic case is that the quantitative interpretation function will have to measure the probability of satisfying a formula within a sequence of average amounts of time.

Definition 5.1 The interpretation function $[\![.]\!]_{\mathrm{MT}}$ of $\mathcal{ML}_{\mathrm{T,ind}}$ over $\mathbb{P}_{\mathrm{M}} \times (\mathbb{R}_{>0})^*$ is defined by letting $[\![\phi]\!]_{\mathrm{MT}}(P, \theta) = 0$ for $\phi \in \mathcal{ML}_{\mathrm{T,ind}} - \{\mathrm{true}\}$ whenever $init(P) \cap$

$init(\phi) = \emptyset$ or $\theta = \varepsilon$, otherwise by structural induction as follows:

$$\begin{array}{rcl}
[\![\mathrm{true}]\!]_{\mathrm{MT}}(P,\theta) & = & 1 \\
[\![\phi_1 \vee \phi_2]\!]_{\mathrm{MT}}(P,t \circ \theta) & = & p_1 \cdot [\![\phi_1]\!]_{\mathrm{MT}}(P,t_1 \circ \theta) + p_2 \cdot [\![\phi_2]\!]_{\mathrm{MT}}(P,t_2 \circ \theta) \\
[\![\langle a \rangle \phi]\!]_{\mathrm{MT}}(P,t \circ \theta) & = & \begin{cases} \displaystyle\sum_{P \xrightarrow{a,\lambda}_{\mathrm{M}} P'} \overset{\lambda}{rate_{\mathrm{c}}(P|\{a\})} \cdot [\![\phi]\!]_{\mathrm{MT}}(P',\theta) & \text{if } \overset{1}{rate_{\mathrm{c}}(P|\{a\})} \leq t \\ 0 & \text{if } \overset{1}{rate_{\mathrm{c}}(P|\{a\})} > t \end{cases}
\end{array}$$

where $p_j = \frac{rate_{\mathrm{c}}(P|init(\phi_j))}{rate_{\mathrm{c}}(P|init(\phi_1 \vee \phi_2))}$ and $t_j = t + (\overset{1}{rate_{\mathrm{c}}(P|init(\phi_j))} - \overset{1}{rate_{\mathrm{c}}(P|init(\phi_1 \vee \phi_2))})$ for $j \in \{1,2\}$.

In the definition above, t_j represents the extra average time granted to P for satisfying ϕ_j. This extra average time is equal to the difference between the average sojourn time in P when only actions whose name is in $init(\phi_j)$ are enabled and the average sojourn time in P when also actions whose name is in $init(\phi_k)$, $k = 3 - j$, are enabled. Since the latter cannot be greater than the former due to the race policy (more enabled actions means less time spent on average in a state), considering t instead of t_j in the satisfaction of ϕ_j would lead to a potential underestimate of the probability of satisfying $\phi_1 \vee \phi_2$ within the given time bound, as P may satisfy $\phi_1 \vee \phi_2$ within $t \circ \theta$ even if P satisfies neither ϕ_1 alone nor ϕ_2 alone within $t \circ \theta$.

Lemma 5.2 *For all $T \in \mathbb{T}_{\mathrm{R,det}}$ there exists $\phi_T \in \mathcal{ML}_{\mathrm{T,ind}}$ such that $init(\phi_T) = init(T)$ and for all $P \in \mathbb{P}_{\mathrm{M}}$ and $\theta \in (\mathbb{R}_{>0})^*$:*

$$[\![\phi_T]\!]_{\mathrm{MT}}(P,\theta) = prob(\mathcal{SC}_{\leq\theta}(P,T))$$

Lemma 5.3 *For all $\phi \in \mathcal{ML}_{\mathrm{T,ind}}$ there exists $T_\phi \in \mathbb{T}_{\mathrm{R,det}}$ such that $init(T_\phi) = init(\phi)$ and for all $P \in \mathbb{P}_{\mathrm{M}}$ and $\theta \in (\mathbb{R}_{>0})^*$:*

$$prob(\mathcal{SC}_{\leq\theta}(P,T_\phi)) = [\![\phi]\!]_{\mathrm{MT}}(P,\theta)$$

Theorem 5.4 *Let $P_1, P_2 \in \mathbb{P}_{\mathrm{M}}$. Then $P_1 \sim_{\mathrm{MT}} P_2$ iff for all $\phi \in \mathcal{ML}_{\mathrm{T,ind}}$ and $\theta \in (\mathbb{R}_{>0})^*$:*

$$[\![\phi]\!]_{\mathrm{MT}}(P_1,\theta) = [\![\phi]\!]_{\mathrm{MT}}(P_2,\theta)$$

6 Conclusion

Starting from the comparison of [3], in this paper we have tackled the problem of finding a uniform modal language for characterizing three variants of testing equivalence. We have demonstrated that such a language exists and comprises true, disjunction, and diamond. In the nondeterministic case, similarly to [16] we have defined a may-satisfaction relation and a must-satisfaction relation, with the latter being carefully designed for disjunction and diamond in order to characterize correctly must-testing equivalence. In the probabilistic and Markovian cases, we have provided a quantitative interpretation inspired by [20], after imposing an independence constraint on the occurrences of disjunction. The revised comparison resulting

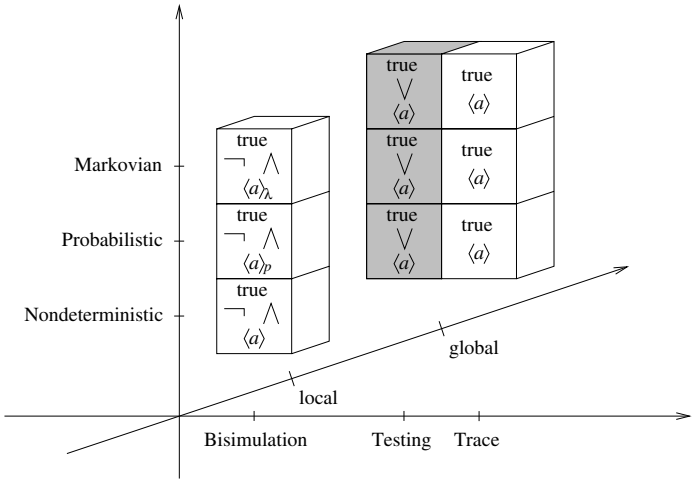

Fig. 2. Revised comparison of logical characterizations for behavioral equivalences

from the alternative characterizations of the three variants of testing equivalence is shown in Fig. 2.

The figure hides the fact that two different satisfaction relations have been provided in the nondeterministic case, and that must-satisfaction results in a nonstandard interpretation of disjunction and diamond. However, this seems to be unavoidable due to the way testing equivalence is defined. By contrast, a single interpretation function is enough in the probabilistic and Markovian cases, as the fact that a process may or must pass a test is encoded within a probability interval. It is also worth pointing out that, similarly to [20], in those two cases no quantititative information is necessary within the uniform modal language.

In [3] and in the present work we have compared (on a modal logic basis) bisimulation, testing, and trace equivalences for fully nondeterministic, fully probabilistic, and fully Markovian finite-state processes without silent moves. In the nondeterministic and probabilistic cases, τ actions can be handled by means of a weak interpretation of diamond, as shown e.g. in [23,24,10]. However, the situation is more complicated for Markovian processes, as it is rarely the case that we can abstract from the durations of τ actions while remaining in the field of exponential distributions [18].

As far as the coexistence of nondeterminism and probabilistic/temporal aspects is concerned, for bisimulation equivalence we have already mentioned in the introduction that the modal language shown in the first column of Fig. 2 is valid as long as nondeterminism and probability are not simultaneously present in the same states, otherwise a further operator is necessary [24]. For the testing case, in [10] it is shown that a modal language different from the one in the second column of Fig. 2 is necessary, which is composed of true, conjunction, diamond, failure predicate, and convex combination. However, the testing equivalence considered in [10] for mixed nondeterministic/probabilistic processes is not a conservative extension of the testing equivalence of [12] for fully nondeterministic processes. The reason is that [10] allows for probabilistic choices within tests, which is enough for distinguishing non-

deterministic processes that are equivalent according to [12]. In fact, probabilistic choices within tests provide the capability of making copies of the states of the processes being tested and experimenting on each of them independently, which increases the distinguishing power [1].

Future work will be devoted to investigating whether and to which extent our uniform modal language for testing equivalence is suitable for the framework of [10] under the constraint that tests are fully nondeterministic. Moreover, it would be interesting to find a way of encoding the uniform modal language of this paper into modal languages for which model-checking algorithms already exist.

Acknowledgement

We thank Michele Loreti for some initial discussions on modal logics for testing equivalences. This work has been funded by MIUR-PRIN project *PaCo – Performability-Aware Computing: Logics, Models, and Languages.*

References

[1] S. Abramsky, *"Observational Equivalence as a Testing Equivalence"*, in Theoretical Computer Science 53:225-241, 1987.

[2] M. Bernardo, *"Non-Bisimulation-Based Markovian Behavioral Equivalences"*, in Journal of Logic and Algebraic Programming 72:3-49, 2007.

[3] M. Bernardo and S. Botta, *"A Survey of Modal Logics Characterizing Behavioral Equivalences for Nondeterministic and Stochastic Systems"*, in Mathematical Structures in Computer Science 18:29-55, 2008.

[4] M. Bernardo and M. Bravetti, *"Performance Measure Sensitive Congruences for Markovian Process Algebras"*, in Theoretical Computer Science 290:117-160, 2003.

[5] M. Bravetti and A. Aldini, *"Discrete Time Generative-Reactive Probabilistic Processes with Different Advancing Speeds"*, in Theoretical Computer Science 290:355-406, 2003.

[6] S.D. Brookes, C.A.R. Hoare, and A.W. Roscoe, *"A Theory of Communicating Sequential Processes"*, in Journal of the ACM 31:560-599, 1984.

[7] I. Christoff, *"Testing Equivalences and Fully Abstract Models for Probabilistic Processes"*, in Proc. of the *1st Int. Conf. on Concurrency Theory (CONCUR 1990)*, LNCS 458:126-140, 1990.

[8] G. Clark, S. Gilmore, and J. Hillston, *"Specifying Performance Measures for PEPA"*, in Proc. of the *5th AMAST Int. Workshop on Formal Methods for Real Time and Probabilistic Systems (ARTS 1999)*, LNCS 1601:211-227, 1999.

[9] R. Cleaveland, Z. Dayar, S.A. Smolka, and S. Yuen, *"Testing Preorders for Probabilistic Processes"*, in Information and Computation 154:93-148, 1999.

[10] Y. Deng, R.J. van Glabbeek, M. Hennessy, C. Morgan, and C. Zhang, *"Characterising Testing Preorders for Finite Probabilistic Processes"*, in Proc. of the *22nd IEEE Symp. on Logic in Computer Science (LICS 2007)*, IEEE-CS Press, pp. 313-325, 2007.

[11] R. De Nicola, *"Extensional Equivalences for Transition Systems"*, in Acta Informatica 24:211-237, 1987.

[12] R. De Nicola and M. Hennessy, *"Testing Equivalences for Processes"*, in Theoretical Computer Science 34:83-133, 1984.

[13] J. Desharnais, A. Edalat, and P. Panangaden, *"Bisimulation for Labelled Markov Processes"*, in Information and Computation 179:163-193, 2002.

[14] R.J. van Glabbeek, *"The Linear Time - Branching Time Spectrum I"*, in *"Handbook of Process Algebra"*, pp. 3-99, Elsevier, 2001.

[15] R.J. van Glabbeek, S.A. Smolka, and B. Steffen, *"Reactive, Generative and Stratified Models of Probabilistic Processes"*, in Information and Computation 121:59-80, 1995.

[16] M. Hennessy, *"Acceptance Trees"*, in Journal of the ACM 32:896-928, 1985.

[17] M. Hennessy and R. Milner, *"Algebraic Laws for Nondeterminism and Concurrency"*, in Journal of the ACM 32:137-162, 1985.

[18] J. Hillston, *"A Compositional Approach to Performance Modelling"*, Cambridge University Press, 1996.

[19] C.-C. Jou and S.A. Smolka, *"Equivalences, Congruences, and Complete Axiomatizations for Probabilistic Processes"*, in Proc. of the *1st Int. Conf. on Concurrency Theory (CONCUR 1990)*, LNCS 458:367-383, 1990.

[20] M.Z. Kwiatkowska and G.J. Norman, *"A Testing Equivalence for Reactive Probabilistic Processes"*, in Proc. of the *2nd Int. Workshop on Expressiveness in Concurrency (EXPRESS 1998)*, ENTCS 16(2):114-132, 1998.

[21] K.G. Larsen and A. Skou, *"Bisimulation through Probabilistic Testing"*, in Information and Computation 94:1-28, 1991.

[22] K.G. Larsen and A. Skou, *"Compositional Verification of Probabilistic Processes"*, in Proc. of the *3rd Int. Conf. on Concurrency Theory (CONCUR 1992)*, LNCS 630:456-471, 1992.

[23] R. Milner, *"Communication and Concurrency"*, Prentice Hall, 1989.

[24] A. Parma and R. Segala, *"Logical Characterizations of Bisimulations for Discrete Probabilistic Systems"*, in Proc. of the *10th Int. Conf. on Foundations of Software Science and Computational Structures (FOSSACS 2007)*, LNCS 4423:287-301, 2007.

Appendix: Proofs

Proof of Lemma 3.5: We proceed by induction on the syntactical structure of $T \in \mathbb{T}_{\mathrm{N}}$:

- Let $T \equiv \mathrm{s}$ and take $\phi_T \equiv \mathrm{true}$. Since for all $P \in \mathbb{P}_{\mathrm{N}}$ we have that P may pass s, P must pass s, $P \models_{\mathrm{may}} \mathrm{true}$, and $P \models_{\mathrm{must}} \mathrm{true}$, the result follows.
- Let $T \equiv T_1 + T_2$ and $P \in \mathbb{P}_{\mathrm{N}}$.
  If P may pass T, then $P \,\|\, T$ has at least one successful computation. Since $T \equiv T_1 + T_2$, this successful computation must belong to $P \,\|\, T_1$ or $P \,\|\, T_2$, hence P may pass T_1 or P may pass T_2. From the induction hypothesis it follows that there exist $\phi_{T_1}, \phi_{T_2} \in \mathcal{ML}_{\mathrm{T}}$ with $\mathit{init}(\phi_{T_1}) = \mathit{init}(T_1)$ and $\mathit{init}(\phi_{T_2}) = \mathit{init}(T_2)$ such that $P \models_{\mathrm{may}} \phi_{T_1}$ or $P \models_{\mathrm{may}} \phi_{T_2}$. Therefore $P \models_{\mathrm{may}} \phi_{T_1} \vee \phi_{T_2}$.
  If P must pass T, then all the maximal computations of $P \,\|\, T$ are successful. Since $T \equiv T_1 + T_2$, it must be the case that $\mathit{init}(P) \cap (\mathit{init}(T_1) \cup \mathit{init}(T_2)) \neq \emptyset$. Moreover, $\mathit{init}(P) \cap \mathit{init}(T_1) \neq \emptyset$ necessarily implies that P must pass T_1. Likewise, $\mathit{init}(P) \cap \mathit{init}(T_2) \neq \emptyset$ necessarily implies that P must pass T_2. From the induction hypothesis it follows that there exist $\phi_{T_1}, \phi_{T_2} \in \mathcal{ML}_{\mathrm{T}}$ with $\mathit{init}(\phi_{T_1}) = \mathit{init}(T_1)$ and $\mathit{init}(\phi_{T_2}) = \mathit{init}(T_2)$ such that $\mathit{init}(P) \cap \mathit{init}(\phi_{T_1}) \neq \emptyset$ implies $P \models_{\mathrm{must}} \phi_{T_1}$ and $\mathit{init}(P) \cap \mathit{init}(\phi_{T_2}) \neq \emptyset$ implies $P \models_{\mathrm{must}} \phi_{T_2}$. From $\mathit{init}(P) \cap (\mathit{init}(\phi_{T_1}) \cup \mathit{init}(\phi_{T_2})) \neq \emptyset$ we derive that $P \models_{\mathrm{must}} \phi_{T_1} \vee \phi_{T_2}$.
  The result then follows by taking $\phi_T \equiv \phi_{T_1} \vee \phi_{T_2}$.
- Let $T \equiv a.T'$ and $P \in \mathbb{P}_{\mathrm{N}}$.
  If P may pass T, then $P \,\|\, T$ has at least one successful computation. Since $T \equiv a.T'$, there must exist $P' \in \mathbb{P}_{\mathrm{N}}$ such that $P \xrightarrow{a}_{\mathrm{N}} P'$ and P' may pass T'.

From the induction hypothesis it follows that there exists $\phi_{T'} \in \mathcal{ML}_{\rm T}$ such that $P' \models_{\rm may} \phi_{T'}$. Therefore $P \models_{\rm may} \langle a \rangle \phi_{T'}$.
If P must pass T, then all the maximal computations of $P \parallel T$ are successful. Since $T \equiv a.T'$, there must exist $P' \in \mathbb{P}_{\rm N}$ such that $P \stackrel{a}{\longrightarrow}_{\rm N} P'$ and each such P' must pass T'. From the induction hypothesis it follows that there exists $\phi_{T'} \in \mathcal{ML}_{\rm T}$ such that each such $P' \models_{\rm must} \phi_{T'}$. Therefore $P \models_{\rm must} \langle a \rangle \phi_{T'}$.
The result then follows by taking $\phi_T \equiv \langle a \rangle \phi_{T'}$. □

Proof of Lemma 3.6: We proceed by induction on the syntactical structure of $\phi \in \mathcal{ML}_{\rm T}$:

- Let $\phi \equiv$ true and take $T_\phi \equiv$ s. Since for all $P \in \mathbb{P}_{\rm N}$ we have that $P \models_{\rm may}$ true, $P \models_{\rm must}$ true, P may pass s, and P must pass s, the result follows.
- Let $\phi \equiv \phi_1 \vee \phi_2$ and $P \in \mathbb{P}_{\rm N}$.
  If $P \models_{\rm may} \phi$, then $P \models_{\rm may} \phi_1$ or $P \models_{\rm may} \phi_2$. From the induction hypothesis it follows that there exist $T_{\phi_1}, T_{\phi_2} \in \mathbb{T}_{\rm N}$ with $init(T_{\phi_1}) = init(\phi_1)$ and $init(T_{\phi_2}) = init(\phi_2)$ such that P may pass T_{ϕ_1} or P may pass T_{ϕ_2}. Therefore P may pass $T_{\phi_1} + T_{\phi_2}$.
  If $P \models_{\rm must} \phi$, then it must be the case that $init(P) \cap (init(\phi_1) \cup init(\phi_2)) \neq \emptyset$. Moreover, $init(P) \cap init(\phi_1) \neq \emptyset$ necessarily implies that $P \models_{\rm must} \phi_1$. Likewise, $init(P) \cap init(\phi_2) \neq \emptyset$ necessarily implies that $P \models_{\rm must} \phi_2$. From the induction hypothesis it follows that there exist $T_{\phi_1}, T_{\phi_2} \in \mathbb{T}_{\rm N}$ with $init(T_{\phi_1}) = init(\phi_1)$ and $init(T_{\phi_2}) = init(\phi_2)$ such that $init(P) \cap init(T_{\phi_1}) \neq \emptyset$ implies that P must pass T_{ϕ_1} and $init(P) \cap init(T_{\phi_2}) \neq \emptyset$ implies that P must pass T_{ϕ_2}. From $init(P) \cap (init(T_{\phi_1}) \cup init(T_{\phi_2})) \neq \emptyset$ we derive that P must pass $T_{\phi_1} + T_{\phi_2}$.
  The result then follows by taking $T_\phi \equiv T_{\phi_1} + T_{\phi_2}$.
- Let $\phi \equiv \langle a \rangle \phi'$ and $P \in \mathbb{P}_{\rm N}$.
  If $P \models_{\rm may} \phi$, then there exists $P' \in \mathbb{P}_{\rm N}$ such that $P \stackrel{a}{\longrightarrow}_{\rm N} P'$ and $P' \models_{\rm may} \phi'$. From the induction hypothesis it follows that there exists $T_{\phi'} \in \mathbb{T}_{\rm N}$ such that P' may pass $T_{\phi'}$. Therefore P may pass $a.T_{\phi'}$.
  If $P \models_{\rm must} \phi$, then there exists $P' \in \mathbb{P}_{\rm N}$ such that $P \stackrel{a}{\longrightarrow}_{\rm N} P'$ and each such $P' \models_{\rm must} \phi'$. From the induction hypothesis it follows that there exists $T_{\phi'} \in \mathbb{T}_{\rm N}$ such that each such P' must pass $T_{\phi'}$. Therefore P must pass $a.T_{\phi'}$.
  The result then follows by taking $T_\phi \equiv a.T_{\phi'}$. □

Proof of Thm. 3.7: A straightforward consequence of the one-to-one correspondence between tests in $\mathbb{T}_{\rm N}$ and formulas in $\mathcal{ML}_{\rm T}$ established by Lemma 3.5 and Lemma 3.6. □

Proof of Lemma 4.2: We proceed by induction on the syntactical structure of $T \in \mathbb{T}_{\rm R,det}$:

- Let $T \equiv$ s and take $\phi_T \equiv$ true. Since for all $P \in \mathbb{P}_{\rm P}$ we have $[\![{\rm true}]\!]_{\rm PT}(P) = 1 = prob(\mathcal{SC}(P, {\rm s}))$, the result follows.
- Let $T \equiv T_1 + T_2$ and $P \in \mathbb{P}_{\rm P}$. In order to avoid trivial cases, assume $init(P) \cap$

$init(T) \neq \emptyset$. So $prob(\mathcal{SC}(P,T)) = \frac{prob_{\rm c}(P|init(T_1))}{prob_{\rm c}(P|init(T))} \cdot prob(\mathcal{SC}(P,T_1)) + \frac{prob_{\rm c}(P|init(T_2))}{prob_{\rm c}(P|init(T))} \cdot prob(\mathcal{SC}(P,T_2))$. From the induction hypothesis it follows that there exist $\phi_{T_1}, \phi_{T_2} \in \mathcal{ML}_{\rm T,ind}$ with $init(\phi_{T_1}) = init(T_1)$ and $init(\phi_{T_2}) = init(T_2)$ s.t. $[\![\phi_{T_1}]\!]_{\rm PT}(P) = prob(\mathcal{SC}(P,T_1))$ and $[\![\phi_{T_2}]\!]_{\rm PT}(P) = prob(\mathcal{SC}(P,T_2))$. Thus $prob(\mathcal{SC}(P,T)) = \frac{prob_{\rm c}(P|init(\phi_{T_1}))}{prob_{\rm c}(P|init(\phi_{T_1} \vee \phi_{T_2}))} \cdot [\![\phi_{T_1}]\!]_{\rm PT}(P) + \frac{prob_{\rm c}(P|init(\phi_{T_2}))}{prob_{\rm c}(P|init(\phi_{T_1} \vee \phi_{T_2}))} \cdot [\![\phi_{T_2}]\!]_{\rm PT}(P)$. From $T \in \mathbb{T}_{\rm R,det}$ we derive that $\phi_{T_1} \vee \phi_{T_2} \in \mathcal{ML}_{\rm T,ind}$, hence the result follows by taking $\phi_T \equiv \phi_{T_1} \vee \phi_{T_2}$.

- Let $T \equiv <a, *_w>.T'$ and $P \in \mathbb{P}_{\rm P}$. In order to avoid trivial cases, assume $init(P) \cap init(T) \neq \emptyset$. Then $prob(\mathcal{SC}(P,T)) = \sum_{P \stackrel{a,p}{\longrightarrow}_{\rm P} P'} prob_{\rm c}^{\,p}(P|\{a\}) \cdot prob(\mathcal{SC}(P',T'))$. From the induction hypothesis it follows that there exists $\phi_{T'} \in \mathcal{ML}_{\rm T,ind}$ such that $[\![\phi_{T'}]\!]_{\rm PT}(P') = prob(\mathcal{SC}(P',T'))$ for each P' reachable from P via a. Thus $prob(\mathcal{SC}(P,T)) = \sum_{P \stackrel{a,p}{\longrightarrow}_{\rm P} P'} prob_{\rm c}^{\,p}(P|\{a\}) \cdot [\![\phi_{T'}]\!]_{\rm PT}(P')$. The result follows by taking $\phi_T \equiv \langle a \rangle \phi_{T'}$. □

Proof of Lemma 4.3: We proceed by induction on the syntactical structure of $\phi \in \mathcal{ML}_{\rm T,ind}$:

- Let $\phi \equiv$ true and take $T_\phi \equiv$ s. Since for all $P \in \mathbb{P}_{\rm P}$ we have $prob(\mathcal{SC}(P,{\rm s})) = 1 = [\![{\rm true}]\!]_{\rm PT}(P)$, the result follows.
- Let $\phi \equiv \phi_1 \vee \phi_2$ and $P \in \mathbb{P}_{\rm P}$. In order to avoid trivial cases, assume $init(P) \cap init(\phi) \neq \emptyset$. Then $[\![\phi]\!]_{\rm PT}(P) = \frac{prob_{\rm c}(P|init(\phi_1))}{prob_{\rm c}(P|init(\phi))} \cdot [\![\phi_1]\!]_{\rm PT}(P) + \frac{prob_{\rm c}(P|init(\phi_2))}{prob_{\rm c}(P|init(\phi))} \cdot [\![\phi_2]\!]_{\rm PT}(P)$. From the induction hypothesis it follows that there exist $T_{\phi_1}, T_{\phi_2} \in \mathbb{T}_{\rm R,det}$ with $init(T_{\phi_1}) = init(\phi_1)$ and $init(T_{\phi_2}) = init(\phi_2)$ s.t. $prob(\mathcal{SC}(P,T_{\phi_1})) = [\![\phi_1]\!]_{\rm PT}(P)$ and $prob(\mathcal{SC}(P,T_{\phi_2})) = [\![\phi_2]\!]_{\rm PT}(P)$, so $[\![\phi]\!]_{\rm PT}(P) = \frac{prob_{\rm c}(P|init(T_{\phi_1}))}{prob_{\rm c}(P|init(T_{\phi_1}+T_{\phi_2}))} \cdot prob(\mathcal{SC}(P,T_{\phi_1})) + \frac{prob_{\rm c}(P|init(T_{\phi_2}))}{prob_{\rm c}(P|init(T_{\phi_1}+T_{\phi_2}))} \cdot prob(\mathcal{SC}(P,T_{\phi_2}))$. From $\phi \in \mathcal{ML}_{\rm T,ind}$ we derive that $T_{\phi_1} + T_{\phi_2} \in \mathbb{T}_{\rm R,det}$, hence the result follows by taking $T_\phi \equiv T_{\phi_1} + T_{\phi_2}$.
- Let $\phi \equiv \langle a \rangle \phi'$ and $P \in \mathbb{P}_{\rm P}$. In order to avoid trivial cases, assume $init(P) \cap init(\phi) \neq \emptyset$. Then $[\![\phi]\!]_{\rm PT}(P) = \sum_{P \stackrel{a,p}{\longrightarrow}_{\rm P} P'} prob_{\rm c}^{\,p}(P|\{a\}) \cdot [\![\phi']\!]_{\rm PT}(P')$. From the induction hypothesis it follows that there exists $T_{\phi'} \in \mathbb{T}_{\rm R,det}$ such that $prob(\mathcal{SC}(P',T_{\phi'})) = [\![\phi']\!]_{\rm PT}(P')$ for each P' reachable from P via a. Thus $[\![\phi]\!]_{\rm PT}(P) = \sum_{P \stackrel{a,p}{\longrightarrow}_{\rm P} P'} prob_{\rm c}^{\,p}(P|\{a\}) \cdot prob(\mathcal{SC}(P',T_{\phi'}))$. The result follows by taking $T_\phi \equiv <a, *_w>.T_{\phi'}$. □

Proof of Thm. 4.4: A straightforward consequence of the one-to-one correspondence between classes of tests in $\mathbb{T}_{\rm R,det}$ differring only for the action weights and formulas in $\mathcal{ML}_{\rm T,ind}$ established by Lemma 4.2 and Lemma 4.3. □

Proof of Lemma 5.2: We proceed by induction on the syntactical structure of $T \in \mathbb{T}_{\rm R,det}$:

- Let $T \equiv$ s and take $\phi_T \equiv$ true. Since for all $P \in \mathbb{P}_{\rm M}$ and $\theta \in (\mathbb{R}_{>0})^*$ we have $[\![{\rm true}]\!]_{\rm MT}(P,\theta) = 1 = prob(\mathcal{SC}_{\leq\theta}(P,{\rm s}))$, the result follows.

- Let $T \equiv T_1 + T_2$, $P \in \mathbb{P}_{\rm M}$, and $\theta \in (\mathbb{R}_{>0})^*$. In order to avoid trivial cases, assume $init(P) \cap init(T) \neq \emptyset$ and $\theta \equiv t \circ \theta'$. Then $prob(\mathcal{SC}_{\leq\theta}(P,T)) = \frac{rate_{\rm c}(P|init(T_1))}{rate_{\rm c}(P|init(T))} \cdot prob(\mathcal{SC}_{\leq t_1\circ\theta'}(P,T_1)) + \frac{rate_{\rm c}(P|init(T_2))}{rate_{\rm c}(P|init(T))} \cdot prob(\mathcal{SC}_{\leq t_2\circ\theta'}(P,T_2))$ where $t_j = t + (\frac{1}{rate_{\rm c}(P|init(T_j))} - \frac{1}{rate_{\rm c}(P|init(T))})$ for $j \in \{1,2\}$. From the induction hypothesis it follows that there exist $\phi_{T_1}, \phi_{T_2} \in \mathcal{ML}_{\rm T,ind}$ with $init(\phi_{T_1}) = init(T_1)$ and $init(\phi_{T_2}) = init(T_2)$ such that $[\![\phi_{T_1}]\!]_{\rm MT}(P, t_1 \circ \theta') = prob(\mathcal{SC}_{\leq t_1\circ\theta'}(P,T_1))$ and $[\![\phi_{T_2}]\!]_{\rm MT}(P, t_2 \circ \theta') = prob(\mathcal{SC}_{\leq t_2\circ\theta'}(P,T_2))$. Thus $prob(\mathcal{SC}_{\leq\theta}(P,T)) = \frac{rate_{\rm c}(P|init(\phi_{T_1}))}{rate_{\rm c}(P|init(\phi_{T_1}\vee\phi_{T_2}))} \cdot [\![\phi_{T_1}]\!]_{\rm MT}(P, t_1\circ\theta') + \frac{rate_{\rm c}(P|init(\phi_{T_2}))}{rate_{\rm c}(P|init(\phi_{T_1}\vee\phi_{T_2}))} \cdot [\![\phi_{T_2}]\!]_{\rm MT}(P, t_2\circ\theta')$. From $T \in \mathbb{T}_{\rm R,det}$ we derive that $\phi_{T_1} \vee \phi_{T_2} \in \mathcal{ML}_{\rm T,ind}$, hence the result follows by taking $\phi_T \equiv \phi_{T_1} \vee \phi_{T_2}$.
- Let $T \equiv <a, *_w>.T'$, $P \in \mathbb{P}_{\rm M}$, and $\theta \in (\mathbb{R}_{>0})^*$. In order to avoid trivial cases, assume $init(P) \cap init(T) \neq \emptyset$, $\theta \equiv t \circ \theta'$, and $\frac{1}{rate_{\rm c}(P|\{a\})} \leq t$. Then $prob(\mathcal{SC}_{\leq\theta}(P,T)) = \sum_{P \xrightarrow{a,\lambda}_{\rm M} P'} \frac{\lambda}{rate_{\rm c}(P|\{a\})} \cdot prob(\mathcal{SC}_{\leq\theta'}(P',T'))$. From the induction hypothesis it follows that there exists $\phi_{T'} \in \mathcal{ML}_{\rm T,ind}$ such that $[\![\phi_{T'}]\!]_{\rm MT}(P',\theta') = prob(\mathcal{SC}_{\leq\theta'}(P',T'))$ for each P' reachable from P via a. Thus $prob(\mathcal{SC}_{\leq\theta}(P,T)) = \sum_{P \xrightarrow{a,\lambda}_{\rm M} P'} \frac{\lambda}{rate_{\rm c}(P|\{a\})} \cdot [\![\phi_{T'}]\!]_{\rm MT}(P',\theta')$. The result follows by taking $\phi_T \equiv \langle a\rangle\phi_{T'}$. □

Proof of Lemma 5.3: We proceed by induction on the syntactical structure of $\phi \in \mathcal{ML}_{\rm T,ind}$:

- Let $\phi \equiv$ true and take $T_\phi \equiv$ s. Since for all $P \in \mathbb{P}_{\rm M}$ and $\theta \in (\mathbb{R}_{>0})^*$ we have $prob(\mathcal{SC}_{\leq\theta}(P,{\rm s})) = 1 = [\![{\rm true}]\!]_{\rm MT}(P,\theta)$, the result follows.
- Let $\phi \equiv \phi_1 \vee \phi_2$, $P \in \mathbb{P}_{\rm M}$, and $\theta \in (\mathbb{R}_{>0})^*$. In order to avoid trivial cases, assume $init(P) \cap init(\phi) \neq \emptyset$ and $\theta \equiv t \circ \theta'$. Then $[\![\phi]\!]_{\rm MT}(P,\theta) = \frac{rate_{\rm c}(P|init(\phi_1))}{rate_{\rm c}(P|init(\phi))} \cdot [\![\phi_1]\!]_{\rm MT}(P, t_1\circ\theta) + \frac{rate_{\rm c}(P|init(\phi_2))}{rate_{\rm c}(P|init(\phi))} \cdot [\![\phi_2]\!]_{\rm MT}(P, t_2\circ\theta)$ where $t_j = t + (\frac{1}{rate_{\rm c}(P|init(\phi_j))} - \frac{1}{rate_{\rm c}(P|init(\phi))})$ for $j \in \{1,2\}$. From the induction hypothesis it follows that there exist $T_{\phi_1}, T_{\phi_2} \in \mathbb{T}_{\rm R,det}$ with $init(T_{\phi_1}) = init(\phi_1)$ and $init(T_{\phi_2}) = init(\phi_2)$ such that $prob(\mathcal{SC}_{\leq t_1\circ\theta}(P,T_{\phi_1})) = [\![\phi_1]\!]_{\rm MT}(P, t_1\circ\theta)$ and $prob(\mathcal{SC}_{\leq t_2\circ\theta}(P,T_{\phi_2})) = [\![\phi_2]\!]_{\rm MT}(P, t_2\circ\theta)$. Thus $[\![\phi]\!]_{\rm MT}(P,\theta) = \frac{rate_{\rm c}(P|init(T_{\phi_1}))}{rate_{\rm c}(P|init(T_{\phi_1}+T_{\phi_2}))} \cdot prob(\mathcal{SC}_{\leq t_1\circ\theta}(P,T_{\phi_1})) + \frac{rate_{\rm c}(P|init(T_{\phi_2}))}{rate_{\rm c}(P|init(T_{\phi_1}+T_{\phi_2}))} \cdot prob(\mathcal{SC}_{\leq t_2\circ\theta}(P,T_{\phi_2}))$. From $\phi \in \mathcal{ML}_{\rm T,ind}$ we derive that $T_{\phi_1} + T_{\phi_2} \in \mathbb{T}_{\rm R,det}$, hence the result follows by taking $T_\phi \equiv T_{\phi_1} + T_{\phi_2}$.
- Let $\phi \equiv \langle a\rangle\phi'$, $P \in \mathbb{P}_{\rm M}$, and $\theta \in (\mathbb{R}_{>0})^*$. In order to avoid trivial cases, assume $init(P) \cap init(\phi) \neq \emptyset$, $\theta \equiv t \circ \theta'$, and $\frac{1}{rate_{\rm c}(P|\{a\})} \leq t$. Then $[\![\phi]\!]_{\rm MT}(P,\theta) = \sum_{P \xrightarrow{a,\lambda}_{\rm M} P'} \frac{\lambda}{rate_{\rm c}(P|\{a\})} \cdot [\![\phi']\!]_{\rm MT}(P',\theta')$. From the induction hypothesis it follows that there exists $T_{\phi'} \in \mathbb{T}_{\rm R,det}$ such that $prob(\mathcal{SC}_{\leq\theta'}(P',T_{\phi'})) = [\![\phi']\!]_{\rm PT}(P',\theta')$ for each P' reachable from P via a, so $[\![\phi]\!]_{\rm MT}(P,\theta) = \sum_{P \xrightarrow{a,\lambda}_{\rm M} P'} \frac{\lambda}{rate_{\rm c}(P|\{a\})} \cdot prob(\mathcal{SC}_{\leq\theta'}(P',T_{\phi'}))$. The result follows by taking $T_\phi \equiv <a, *_w>.T_{\phi'}$. □

Proof of Thm. 5.4: A straightforward consequence of the one-to-one correspondence between classes of tests in $\mathbb{T}_{\mathrm{R,det}}$ differring only for the action weights and formulas in $\mathcal{ML}_{\mathrm{T,ind}}$ established by Lemma 5.2 and Lemma 5.3. □