# Real-Reward Testing for Probabilistic Processes
## (Extended Abstract)

Yuxin Deng[1]* Rob van Glabbeek[2] Matthew Hennessy[3]† Carroll Morgan[4]‡

[1] Shanghai Jiao Tong University and Chinese Academy of Sciences, China

[2] National ICT Australia, Australia

[3] Trinity College Dublin, Ireland

[2,4] University of New South Wales, Australia

We introduce a notion of real-valued reward testing for probabilistic processes by extending the traditional nonnegative-reward testing with negative rewards. In this richer testing framework, the may and must preorders turn out to be inverses. We show that for convergent processes with finitely many states and transitions, but not in the presence of divergence, the real-reward must-testing preorder coincides with the nonnegative-reward must-testing preorder. To prove this coincidence we characterise the usual resolution-based testing in terms of the weak transitions of processes, without having to involve policies, adversaries, schedulers, resolutions, or similar structures that are external to the process under investigation. This requires establishing the continuity of our function for calculating testing outcomes.

## 1 Introduction

Extending classical testing semantics [1, 7] to a setting in which probability and nondeterminism co-exist was initiated in [13]. The application of a test to a process yields a set of probabilities for reaching a success state. *Reward testing* was introduced in [8]; here the success states are labelled by nonnegative real numbers—*rewards*—to indicate degrees of success, and reaching a success state accumulates the associated reward. In [12] an infinite set of success actions is used to report success, and the testing outcomes are vectors of probabilities of performing these success actions. Compared to [8] this amounts to distinguishing different qualities of success, rather than different quantities.

In [13] and [12], both tests and testees are nondeterministic probabilistic processes, whereas [8] allows nonprobabilistic tests only, thereby obtaining a less discriminating form of testing. In [6] we strengthened reward testing by also allowing probabilistic tests. Taking rewards testing in this form we showed that for finitary processes, i.e. finite-state and finitely branching processes, all three modes of testing lead to the same testing preorders. Thus, vector-based testing is no more powerful than *scalar* testing that employs only one success action, and likewise reward testing is no more powerful than the special case of reward testing in which all rewards are 1. [1]

*Deng was supported by the National Natural Science Foundation of China (61033002).

†Supported by SFI project SFI 06 IN.1 1898.

‡Morgan acknowledges the support of ARC Discovery Grant DP0879529.

[1] In spite of this there *is* a difference in power between the notions of testing from [13] and [12], but this is an issue that is entirely orthogonal to the distinction between scalar testing, reward testing and vector-based testing. In [12] it is the execution of a success *action* that constitutes success, whereas in [1, 7, 13, 8] it is reaching a success *state* (even though typically success actions are used to identify those states). In [2, Ex 5.3] we showed that state-based testing is (slightly) more powerful than action-based testing. The results presented in [6] about the coincidence of scalar, reward, and vector-based testing preorders pertain to action-based version of each, but in the conclusion it is observed that the same coincidence could be obtained for their state-based versions. In the current paper we stick to state-based testing.

M. Massink and G. Norman (Eds.): 9th Workshop on
Quantitative Aspects of Programming Languages (QAPL 2011)
EPTCS 57, 2011, pp. 61–73, doi:10.4204/EPTCS.57.5

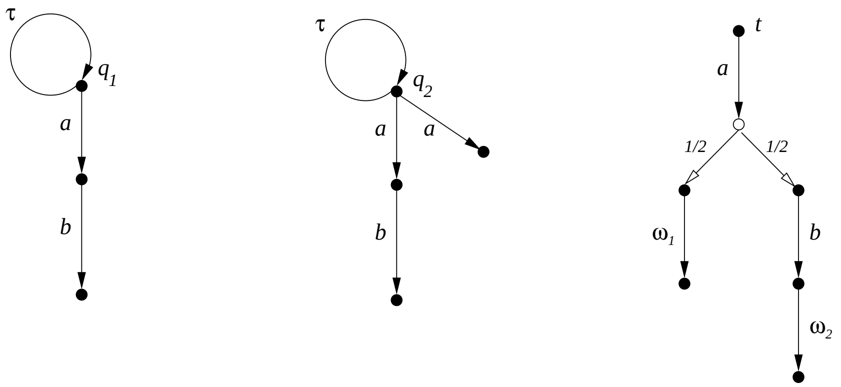

Figure 1: Two processes with divergence and a test

In certain situations it is natural to introduce negative rewards. This is the case, for instance, in the theory of Markov Decision Processes [9]. Intuitively, we could understand negative rewards as costs, while positive rewards are often viewed as benefits or profits. This leads to the question: *if negative rewards are also allowed, how would the original reward-testing semantics change?* We refer to the more relaxed form of testing, using positive and negative rewards, as *real-reward testing* and the original one (from [8], but with probabilistic tests as in [6]) as *nonnegative-reward testing*.

The power of real-reward testing is illustrated in Figure 1. The two (nonprobabilistic) processes in the left- and central diagrams are equivalent under (probabilistic) may- as well as must testing; the $\tau$-loops in the initial states cause both processes to fail any nontrivial must test. Yet, if a reward of $-1$ is associated with performing the action $a$, and a reward of 2 with the subsequent performance of $b$ (implemented by the test in the right diagram; see Example 3.8 for more details), in the first process the net reward is either 0 (if the process remains stuck in its initial state) or positive, whereas running the second process may yield a loss. This example shows that for processes that may exhibit divergence, real-reward testing is more discriminating than nonnegative-reward testing, or other forms of probabilistic testing. It also illustrates that the extra power may be relevant in applications.

As remarked, in [6] we established that for finitary processes the nonnegative-reward must-testing preorder ($\sqsubseteq_{\mathbf{nr}\mathrm{must}}$) coincides with the probabilistic must-testing preorder ($\sqsubseteq_{\mathrm{pmust}}$), and likewise for the may preorders. Here we show that, in contrast to the situation for nonnegative-reward (or scalar) testing, for real-reward testing the may- and must preorders are the inverse of each other, i.e. for any processes $\Delta$ and $\Gamma$,

$$\Delta \sqsubseteq_{\mathbf{rr}\mathrm{may}} \Gamma \quad \text{iff} \quad \Gamma \sqsubseteq_{\mathbf{rr}\mathrm{must}} \Delta. \tag{1}$$

Our main result is that restricted to finitary convergent processes, the real-reward must preorder coincides with the nonnegative-reward must preorder, i.e. for any finitary convergent processes $\Delta, \Gamma$,

$$\Delta \sqsubseteq_{\mathbf{rr}\mathrm{must}} \Gamma \quad \text{iff} \quad \Delta \sqsubseteq_{\mathbf{nr}\mathrm{must}} \Gamma. \tag{2}$$

Here by convergence we mean that there is no infinite sequence of internal transitions of the form $\Lambda_0 \xrightarrow{\tau} \Lambda_1 \xrightarrow{\tau} \cdots$ with distribution $\Lambda_0$ (and thus its successors) reachable from either $\Delta$ or $\Gamma$. This rules out the processes of Figure 1. Although it is easy to see that in (2) the former implies the latter, to prove the opposite is far from trivial. We employ a novel characterisation of the usual resolution-based testing approach, without introducing concepts like *policy* [9], *adversary* [10], *scheduler* [11] or *resolution* [6] that are external to the process under investigation; instead we describe the mechanism for gathering test

$$(\sqsubseteq_{\mathbf{rr}\mathrm{may}})^{-1} \overset{\text{Thm. }3.7}{=} \sqsubseteq_{\mathbf{rr}\mathrm{must}} \overset{\text{Thm. }5.6}{=} \sqsubseteq_{\mathbf{nr}\mathrm{must}} \overset{[6]}{=} \sqsubseteq_{\mathrm{pmust}} \overset{[3]}{=} \sqsubseteq_{FS}$$

*The symbol $=$ between two relations means that they coincide for finitary convergent processes.*

Figure 2: The relationship of different testing preorders.

results in terms of the *weak $\tau$-moves* or *derivations* [3] the investigated process can make, and hence speak of *derivation-based testing*.

This allows us to exploit the failure simulation preorder $\sqsubseteq_{FS}$ that in [3] was proven to coincide with the probabilistic must testing preorder $\sqsubseteq_{\mathrm{pmust}}$ based on resolutions, at least for finitary processes. Using the derivational characterisation we can show that, for finitary convergent processes, $\sqsubseteq_{FS}$ is contained in $\sqsubseteq_{\mathbf{rr}\mathrm{must}}$. Convergence is essential here, even though it is not needed to establish that $\sqsubseteq_{FS}$ is contained in $\sqsubseteq_{\mathbf{nr}\mathrm{must}}$. Combining this with the results from [6] and [3] mentioned above leads to our required result that $\sqsubseteq_{\mathbf{nr}\mathrm{must}}$ is included in $\sqsubseteq_{\mathbf{rr}\mathrm{must}}$, as far as finitary convergent processes are concerned. Consequently, in this case, all the relations of Figure 2 collapse into one.

The rest of this paper is organised as follows. We start by recalling notation for probabilistic labelled transition systems. In Section 3 we review the resolution-based testing approach and show that the real-reward may preorder is simply the inverse of the real-reward must preorder. Moreover, using the example of Figure 1, we show that in the presence of divergence the inclusion of $\sqsubseteq_{\mathbf{nr}\mathrm{must}}$ in $\sqsubseteq_{\mathbf{rr}\mathrm{must}}$ is proper. In Section 4 we present the derivation-based testing approach and also show that the two approaches agree. Then in Section 5 we show for finitary convergent processes that real-reward must testing coincides with nonnegative-reward must testing. We conclude in Section 6.

Due to lack of space, we omit all proofs: they are reported in [4]. Besides the related work already mentioned above, many other studies on probabilistic testing and simulation semantics have appeared in the literature. They are reviewed in [5, 2].

## 2 Probabilistic Processes

A (discrete) probability *subdistribution* over a set $S$ is a function $\Delta : S \to [0,1]$ with $\sum_{s \in S} \Delta(s) \leq 1$; the *support* of such a $\Delta$ is $\lceil \Delta \rceil := \{\, s \in S \mid \Delta(s) > 0 \,\}$, and its *mass* $|\Delta|$ is $\sum_{s \in \lceil \Delta \rceil} \Delta(s)$. A subdistribution is a (total, or full) *distribution* if $|\Delta| = 1$. The point distribution $\overline{s}$ assigns probability 1 to $s$ and 0 to all other elements of $S$, so that $\lceil \overline{s} \rceil = \{s\}$. With $\mathscr{D}_{sub}(S)$ we denote the set of subdistributions over $S$, and with $\mathscr{D}(S)$ its subset of full distributions.

Let $\{\Delta_k \mid k \in K\}$ be a set of subdistributions, possibly infinite. Then $\sum_{k \in K} \Delta_k$ is the real-valued function in $S \to \mathbb{R}$ defined by $(\sum_{k \in K} \Delta_k)(s) := \sum_{k \in K} \Delta_k(s)$. This is a partial operation on subdistributions because for some state $s$ the sum of $\Delta_k(s)$ might exceed 1. If the index set is finite, say $\{1..n\}$, we often write $\Delta_1 + \ldots + \Delta_n$. For $p$ a real number from $[0,1]$ we use $p \cdot \Delta$ to denote the subdistribution given by $(p \cdot \Delta)(s) := p \cdot \Delta(s)$. Finally we use $\varepsilon$ to denote the everywhere-zero subdistribution that thus has empty support. These operations on subdistributions do not readily adapt themselves to distributions; yet if $\sum_{k \in K} p_k = 1$ for some $p_k \geq 0$, and the $\Delta_k$ are distributions, then so is $\sum_{k \in K} p_k \cdot \Delta_k$.

The expected value $\sum_{s \in S} \Delta(s) \cdot f(s)$ over a subdistribution $\Delta$ of a bounded nonnegative function $f$ to the reals or tuples of them is written $\mathrm{Exp}_\Delta(f)$, and the image of a subdistribution $\Delta$ through a function $f : S \to T$, for some set $T$, is written $\mathrm{Img}_f(\Delta)$ — the latter is the subdistribution over $T$ given by $\mathrm{Img}_f(\Delta)(t) := \sum_{f(s)=t} \Delta(s)$ for each $t \in T$.

**Definition 2.1** A *probabilistic labelled transition system* (pLTS) is a triple $\langle S, \mathsf{Act}, \rightarrow\rangle$, where

(i) $S$ is a set of states,

(ii) $\mathsf{Act}$ is a set of visible actions,

(iii) relation $\rightarrow$ is a subset of $S \times \mathsf{Act}_\tau \times \mathscr{D}(S)$.

Here $\mathsf{Act}_\tau$ denotes $\mathsf{Act} \cup \{\tau\}$, where $\tau \notin \mathsf{Act}$ is the invisible- or internal action.

A (nonprobabilistic) labelled transition system (LTS) may be viewed as a degenerate pLTS — one in which only point distributions are used. In this paper a *(probabilistic) process* will simply be a distribution over the state set of a pLTS. As with LTSs, we write $s \xrightarrow{\alpha} \Delta$ for $(s,\alpha,\Delta) \in \rightarrow$, as well as $s \xrightarrow{\alpha}$ for $\exists \Delta : s \xrightarrow{\alpha} \Delta$ and $s \rightarrow$ for $\exists \alpha : s \xrightarrow{\alpha}$, with $s \stackrel{\alpha}{\not\longrightarrow}$ and $s \not\rightarrow$ representing their negations. A pLTS is *deterministic* if for any state $s$ and label $\alpha$ there is at most one distribution $\Delta$ with $s \xrightarrow{\alpha} \Delta$. It is *finitely branching* if the set $\{\Delta \mid s \xrightarrow{\alpha} \Delta,\ \alpha \in L\}$ is finite for all states $s$; if moreover $S$ is finite, then the pLTS is *finitary*. A subdistribution $\Delta$ over the state set $S$ of an arbitrary pLTS is *finitary* if restricting $S$ to the states reachable from $\Delta$ yields a finitary sub-pLTS.

## 3 Testing probabilistic processes

A *test* is a distribution over the state set of a pLTS having $\mathsf{Act}_\tau \cup \Omega$ as its set of transition labels, where $\Omega$ is a set of fresh *success* actions, not already in $\mathsf{Act}_\tau$, introduced specifically to report testing outcomes.[2] For simplicity we may assume a fixed pLTS of processes—our results apply to any choice of such a pLTS—and a fixed pLTS of tests. Since the power of testing depends on the expressivity of the pLTS of tests—in particular certain types of tests are necessary for our results—let us just postulate that this pLTS is sufficiently expressive for our purposes — for example that it can be used to interpret all processes from the language pCSP, as in our previous papers [5, 2, 3].

Although we use success *actions*, they are used merely to mark certain states as success states, namely the sources of transitions labelled by success actions. For this reason we systematically ignore the distributions that can be reached after a success action. We impose two requirements on all states in a pLTS of tests, namely

(A) if $t \xrightarrow{\omega_1}$ and $t \xrightarrow{\omega_2}$ with $\omega_1, \omega_2 \in \Omega$ then $\omega_1 = \omega_2$.

(B) if $t \xrightarrow{\omega}$ with $\omega \in \Omega$ and $t \xrightarrow{\alpha} \Delta$ with $\alpha \in \mathsf{Act}_\tau$ then $u \xrightarrow{\omega}$ for all $u \in \lceil \Delta \rceil$.

The first condition says that a success state can have one success identity only, whereas the second condition is slight weakening of the requirement from [8] that success states must be end states; it allows further progress from an $\omega$-success state, for some $\omega \in \Omega$, but $\omega$ must remain enabled. [3]

To apply test $\Theta$ to process $\Delta$ we form a parallel composition $\Theta \| \Delta$ in which *all* visible actions of $\Delta$ must synchronise with $\Theta$. The synchronisations are immediately renamed into $\tau$. The resulting composition is a process whose only possible actions are the elements of $\Omega_\tau := \Omega \cup \{\tau\}$. Formally, if $\langle \mathbf{P}, \mathsf{Act}, \rightarrow_{\mathbf{P}}\rangle$ and $\langle \mathbf{T}, \mathsf{Act} \cup \Omega, \rightarrow_{\mathbf{T}}\rangle$ are the pLTSs of processes and tests, then the pLTS of applications of tests to processes is $\langle \mathbf{C}, \Omega, \rightarrow\rangle$, with $\mathbf{C} = \{t \| p \mid t \in \mathbf{T} \wedge p \in \mathbf{P}\}$ and $\rightarrow$ the transition relation generated by the rules in Fig. 3. Here if $\Theta \in \mathscr{D}(\mathbf{T})$ and $\Delta \in \mathscr{D}(\mathbf{P})$, then $\Theta \| \Delta$ is the distribution given by $(\Theta \| \Delta)(t \| p) := \Theta(t) \cdot \Delta(p)$. The resulting pLTS also satisfies (A), (B) above; this would not be the case if we had strengthened (B) to require that success states must be end states.

We will define the result $\mathscr{A}(\Theta, \Delta)$ of applying the test $\Theta$ to the process $\Delta$ to be a set of testing outcomes, exactly one of which results from each resolution of the choices in $\Theta \| \Delta$. Each *testing outcome*

[2] For *vector-based* testing we normally take $\Omega$ to be countably infinite [12]. This way we have an unbounded supply of success actions for building tests, of course without obligation to use them all. *Scalar* testing is obtained by taking $|\Omega| = 1$.

[3] Justification for imposing such restrictions can be found in Appendix A of [6].

$$\frac{t \xrightarrow{\alpha}_{\mathbf{T}} \Theta \qquad \alpha \notin \mathsf{Act}}{t \| p \xrightarrow{\alpha} \Theta \| \overline{p}} \qquad \frac{p \xrightarrow{\alpha}_{\mathbf{P}} \Delta \qquad \alpha \notin \mathsf{Act}}{t \| p \xrightarrow{\alpha} \overline{t} \| \Delta} \qquad \frac{t \xrightarrow{a}_{\mathbf{T}} \Theta \quad p \xrightarrow{a}_{\mathbf{P}} \Delta \qquad a \in \mathsf{Act}}{t \| p \xrightarrow{\tau} \Theta \| \Delta}$$

Figure 3: Synchronous parallel composition between tests and processes

is an $\Omega$-tuple of real numbers in the interval [0,1], i.e. a function $o : \Omega \to [0,1]$, and its $\omega$-component $o(\omega)$, for $\omega \in \Omega$, gives the probability that the resolution in question will reach an *$\omega$-success state*, one in which the success action $\omega$ is possible.

Due to the presence of nondeterminism in pLTSs, we need a mechanism to reduce a nondeterministic structure into a set of deterministic structures, each of which determines a single possible outcome. Here we adapt the notion of *resolution*, defined in [6] for probabilistic automata, to pLTSs.

**Definition 3.1 [Resolution]** A *resolution* of a subdistribution $\Delta \in \mathscr{D}_{sub}(S)$ in a pLTS $\langle S, \Omega, \to \rangle$ is a triple $\langle R, \Lambda, \to_R \rangle$ where $\langle R, \Omega, \to_R \rangle$ is a deterministic pLTS and $\Lambda \in \mathscr{D}_{sub}(R)$, such that there exists a *resolving function* $f : R \to S$ satisfying

(i) $\mathrm{Img}_f(\Lambda) = \Delta$

(ii) if $r \xrightarrow{\alpha}_R \Lambda'$ for $\alpha \in \Omega_\tau$ then $f(r) \xrightarrow{\alpha} \mathrm{Img}_f(\Lambda')$

(iii) if $f(r) \xrightarrow{\alpha}$ for $\alpha \in \Omega_\tau$ then $r \xrightarrow{\alpha}_R$ .

The reader is referred to Section 2 of [6] for a detailed discussion of the concept of resolution, and the manner in which a resolution represents a run of a process; in particular in a resolution states in $S$ are allowed to be resolved into distributions, and computation steps can be *probabilistically interpolated*. Our resolutions match the results of applying a scheduler as defined in [11].

We now explain how to associate an outcome with a particular resolution, which in turn will associate a set of outcomes with a subdistribution in a pLTS. Given a deterministic pLTS $\langle R, \Omega, \to_R \rangle$ consider the functional $\mathscr{R} : (R \to [0,1]^\Omega) \to (R \to [0,1]^\Omega)$ defined by

$$\mathscr{R}(f)(r)(\omega) := \begin{cases} 1 & \text{if } r \xrightarrow{\omega} \\ 0 & \text{if } r \not\xrightarrow{\omega} \text{ and } r \not\xrightarrow{\tau} \\ \mathrm{Exp}_\Delta(f)(\omega) & \text{if } r \not\xrightarrow{\omega} \text{ and } r \xrightarrow{\tau} \Delta. \end{cases} \tag{3}$$

We view the unit interval $[0,1]$ ordered in the standard manner as a complete lattice; this induces the structure of a complete lattice on the product $[0,1]^\Omega$ and in turn on the set of functions $R \to [0,1]^\Omega$. The functional $\mathscr{R}$ is easily seen to be monotonic and therefore has a least fixed point, which we denote by $\mathbb{V}_{\langle R, \Omega, \to_R \rangle}$; this is abbreviated to $\mathbb{V}$ when the deterministic pLTS in question is understood.

Now we define $\mathscr{A}(\Theta, \Delta)$ to be the set of vectors

$$\mathscr{A}(\Theta, \Delta) \;:=\; \{\mathrm{Exp}_\Lambda(\mathbb{V}_{\langle R, \Omega, \to_R \rangle}) \mid \langle R, \Lambda, \to_R \rangle \text{ is a resolution of } \Theta \| \Delta\}\,. \tag{4}$$

**Example 3.2** Consider the process $\overline{q_1}$ depicted in Figure 4(a). Here states are represented by filled nodes • and distributions by open nodes ◦. We leave out point-distributions — diverting an incoming edge to the unique state in its support. When we apply the test $\bar{t}$ depicted in Figure 4(b) to it we get the process $\overline{t \| q_1}$ depicted in Figure 4(c). This process is already deterministic, hence has essentially only one resolution: itself. Moreover the outcome $\mathrm{Exp}_{\overline{t \| q_1}}(\mathbb{V}) = \mathbb{V}(t \| q_1)$ associated with it is the least solution of the equation $\mathbb{V}(t \| q_1) = \frac{1}{2} \cdot \mathbb{V}(t \| q_1) + \frac{1}{2}\vec{\omega}$ where $\vec{\omega} : \Omega \to [0,1]$ is the $\Omega$-tuple with $\vec{\omega}(\omega) = 1$ and $\vec{\omega}(\omega') = 0$ for all $\omega' \neq \omega$. In fact this equation has a unique solution in $[0,1]^\Omega$, namely $\vec{\omega}$. Thus $\mathscr{A}(\bar{t}, \overline{q_1}) = \{\vec{\omega}\}$. □

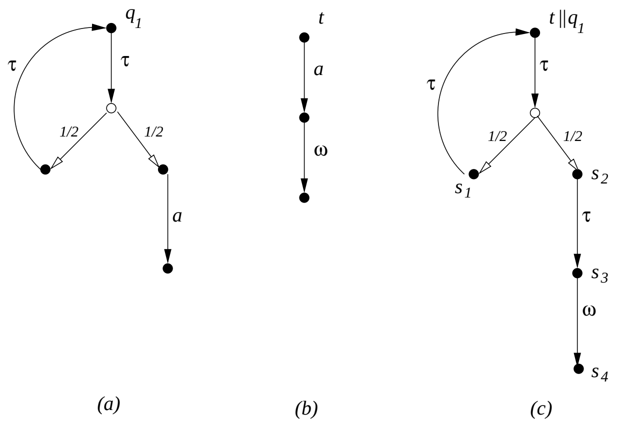

Figure 4: Testing the process $\overline{q_1}$

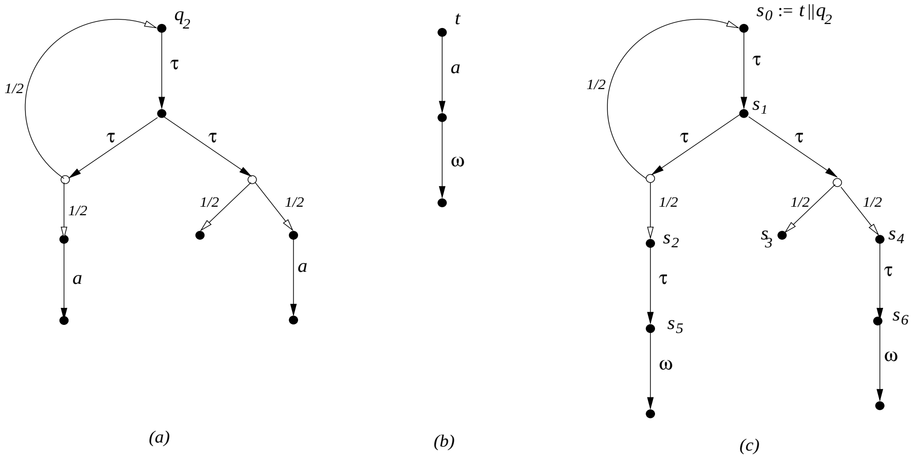

Figure 5: Testing the process $\overline{q_2}$

**Example 3.3** Consider the process $\overline{q_2}$ and the application of the test $\bar{t}$ to it, as outlined in Figure 5. For each $k \geq 1$ the process $\overline{t \| q_2}$ has a resolution $\langle R_k, \Lambda, \rightarrow_{R_k} \rangle$ such that $\mathrm{Exp}_\Lambda(\mathbb{V}) = (1 - \frac{1}{2^k})\overrightarrow{\omega}$; intuitively it goes around the loop $(k-1)$ times before at last taking the right hand $\tau$ action. Thus $\mathscr{A}(\bar{t}, \overline{q_2})$ contains $(1 - \frac{1}{2^k})\overrightarrow{\omega}$ for every $k \geq 1$. But it also contains $\overrightarrow{\omega}$, because of the resolution which takes the left hand $\tau$-move every time. Thus $\mathscr{A}(\bar{t}, \overline{q_2})$ includes the set

$$\{(1 - \tfrac{1}{2})\overrightarrow{\omega},\ (1 - \tfrac{1}{2^2})\overrightarrow{\omega}, \ldots, (1 - \tfrac{1}{2^k})\overrightarrow{\omega}, \ldots, \overrightarrow{\omega}\}$$

As resolutions allow any interpolation between the two $\tau$-transitions from state $s_1$, $\mathscr{A}(\bar{t}, \overline{q_2})$ is actually the convex closure of the above set. □

There are two standard methods for comparing two sets of ordered outcomes:

$$\begin{array}{ll} O_1 \leq_{\mathrm{Ho}} O_2 & \text{if for every } o_1 \in O_1 \text{ there exists some } o_2 \in O_2 \text{ such that } o_1 \leq o_2 \\ O_1 \leq_{\mathrm{Sm}} O_2 & \text{if for every } o_2 \in O_2 \text{ there exists some } o_1 \in O_1 \text{ such that } o_1 \leq o_2 \end{array}$$

This gives us our definition of the probabilistic may- and must-testing preorders; they are decorated with $\cdot^\Omega$ for the repertoire $\Omega$ of testing actions they employ.

**Definition 3.4 [Probabilistic testing preorders]**

(i) $\Delta \sqsubseteq^{\Omega}_{\text{pmay}} \Gamma$ if for every $\Omega$-test $\Theta$, $\mathscr{A}(\Theta,\Delta) \leq_{\text{Ho}} \mathscr{A}(\Theta,\Gamma)$.

(ii) $\Delta \sqsubseteq^{\Omega}_{\text{pmust}} \Gamma$ if for every $\Omega$-test $\Theta$, $\mathscr{A}(\Theta,\Delta) \leq_{\text{Sm}} \mathscr{A}(\Theta,\Gamma)$.

These preorders are abbreviated to $\Delta \sqsubseteq_{\text{pmay}} \Gamma$ and $\Delta \sqsubseteq_{\text{pmust}} \Gamma$ when $|\Omega| = 1$.

In [6] we established that for finitary processes $\sqsubseteq^{\Omega}_{\text{pmay}}$ coincides with $\sqsubseteq_{\text{pmay}}$ and $\sqsubseteq^{\Omega}_{\text{pmust}}$ with $\sqsubseteq_{\text{pmust}}$ for any choice of $\Omega$. We also defined the reward testing preorders in terms of the mechanism set up so far. The idea is to associate with each success action $\omega \in \Omega$ a reward, which is a nonnegative number in the unit interval $[0,1]$; and then a run of a probabilistic process in parallel with a test yields an expected reward accumulated by those states which can enable success actions. A reward tuple $h \in [0,1]^{\Omega}$ is used to assign reward $h(\omega)$ to success action $\omega$, for each $\omega \in \Omega$. Due to the presence of nondeterminism, the application of a test $\Theta$ to a process $\Delta$ produces a set of expected rewards. Two sets of rewards can be compared by examining their suprema/infima; this gives us two methods of testing called reward may/must testing. In [6] all rewards are required to be nonnegative, so we refer to that approach of testing as *nonnegative-reward testing*. If we also allow negative rewards, which intuitively can be understood as costs, then we obtain an approach of testing called *real-reward testing*. Technically, we simply let reward tuples $h$ range over the set $[-1,1]^{\Omega}$. If $o \in [0,1]^{\Omega}$, we use the dot-product $h \cdot o = \sum_{\omega\in\Omega} h(\omega)\cdot o(\omega)$. It can apply to a set $O \subseteq [0,1]^{\Omega}$ so that $h \cdot O = \{h \cdot o \mid o \in O\}$. Let $A \subseteq [-1,1]$. We use the notation $\bigsqcup A$ for the supremum of set $A$, and $\bigsqcap A$ for the infimum.

**Definition 3.5 [Reward testing preorders]**

(i) $\Delta \sqsubseteq^{\Omega}_{\mathbf{nr}\text{may}} \Gamma$ if for every $\Omega$-test $\Theta$ and nonnegative-reward tuple $h \in [0,1]^{\Omega}$,
$\bigsqcup h \cdot \mathscr{A}(\Theta,\Delta) \leq \bigsqcup h \cdot \mathscr{A}(\Theta,\Gamma)$.

(ii) $\Delta \sqsubseteq^{\Omega}_{\mathbf{nr}\text{must}} \Gamma$ if for every $\Omega$-test $\Theta$ and nonnegative-reward tuple $h \in [0,1]^{\Omega}$,
$\bigsqcap h \cdot \mathscr{A}(\Theta,\Delta) \leq \bigsqcap h \cdot \mathscr{A}(\Theta,\Gamma)$.

(iii) $\Delta \sqsubseteq^{\Omega}_{\mathbf{rr}\text{may}} \Gamma$ if for every $\Omega$-test $\Theta$ and real-reward tuple $h \in [-1,1]^{\Omega}$,
$\bigsqcup h \cdot \mathscr{A}(\Theta,\Delta) \leq \bigsqcup h \cdot \mathscr{A}(\Theta,\Gamma)$.

(iv) $\Delta \sqsubseteq^{\Omega}_{\mathbf{rr}\text{must}} \Gamma$ if for every $\Omega$-test $\Theta$ and real-reward tuple $h \in [-1,1]^{\Omega}$,
$\bigsqcap h \cdot \mathscr{A}(\Theta,\Delta) \leq \bigsqcap h \cdot \mathscr{A}(\Theta,\Gamma)$.

This time we drop the superscript $\Omega$ iff $\Omega$ is countably infinite.

It is shown in Corollary 1 of [6] that nonnegative-reward testing is equally powerful as probabilistic testing.

**Theorem 3.6 [6]** For any finitary processes $\Delta$ and $\Gamma$,

(i) $\Delta \sqsubseteq_{\mathbf{nr}\text{may}} \Gamma$ if and only if $\Delta \sqsubseteq_{\text{pmay}} \Gamma$.

(ii) $\Delta \sqsubseteq_{\mathbf{nr}\text{must}} \Gamma$ if and only if $\Delta \sqsubseteq_{\text{pmust}} \Gamma$.

In this paper we focus on the real-reward testing preorders $\sqsubseteq_{\mathbf{rr}\text{may}}$ and $\sqsubseteq_{\mathbf{rr}\text{must}}$, by comparing them with the nonnegative reward testing preorders $\sqsubseteq_{\mathbf{nr}\text{may}}$ and $\sqsubseteq_{\mathbf{nr}\text{must}}$. Although these two nonnegative-reward testing preorders are in general incomparable we have:

**Theorem 3.7** For any processes $\Delta$ and $\Gamma$, it holds that $\Delta \sqsubseteq_{\mathbf{rr}\text{may}} \Gamma$ if and only if $\Gamma \sqsubseteq_{\mathbf{rr}\text{must}} \Delta$.

Our next task is to compare $\sqsubseteq_{\mathbf{rr}\text{must}}$ with $\sqsubseteq_{\mathbf{nr}\text{must}}$. The former is included in the latter, which directly follows from Definition 3.5. Surprisingly, it turns out that for finitary convergent processes the latter is also included in the former, thus establishing that the two preorders are in fact the same. The rest of the paper is devoted to proving this result. However, we first show that this result does not extend to divergent processes.

**Example 3.8** Consider the processes $\overline{q_1}$ and $\overline{q_2}$ depicted in Figure 1. Using the characterisations of $\sqsubseteq_{\text{pmay}}$ and $\sqsubseteq_{\text{pmust}}$ in [3], it is easy to see that these processes cannot be distinguished by probabilistic may- and must testing, and hence not by nonnegative-reward testing either. However, let $\bar{t}$ be the test in the right diagram of Figure 1 that first synchronises on the action $a$, and then with probability $\frac{1}{2}$ reaches a state in which a reward of $-2$ is allocated, and with the remaining probability $\frac{1}{2}$ synchronises with the action $b$ and reaches a state that yields a reward of 4. Thus the test employs two success actions $\omega_1$ and $\omega_2$, and we use the reward tuple $h$ with $h(\omega_1) = -2$ and $h(\omega_2) = 4$. Then the resolution of $\overline{q_1}$ that does not involve the $\tau$-loop contributes the value $-2 \cdot \frac{1}{2} + 4 \cdot \frac{1}{2} = -1 + 2 = 1$ to the set $h \cdot \mathscr{A}(\bar{t}, \overline{q_1})$, whereas the resolution that only involves the $\tau$-loop contributes the value 0. Due to interpolation, $h \cdot \mathscr{A}(\bar{t}, \overline{q_1})$ is in fact the entire interval $[0,1]$. On the other hand, the resolution corresponding to the $a$-branch of $q_2$ contributes the value $-1$ and $h \cdot \mathscr{A}(\bar{t}, \overline{q_2}) = [-1,1]$. Thus $\bigsqcap h \cdot \mathscr{A}(\bar{t}, \overline{q_1}) = 0 > -1 = \bigsqcap h \cdot \mathscr{A}(\bar{t}, \overline{q_2})$, and hence $\overline{q_1} \not\sqsubseteq_{\mathbf{rr}\text{must}} \overline{q_2}$. □

## 4 Derivation-based testing

In this section we give an alternative definition of $\mathscr{A}(\Theta, \Delta)$. Our definition has four ingredients. First of all, for technical reasons we normalise our pLTS of applications of tests to processes by pruning away all outgoing $\tau$-transitions from success states. This way an $\omega$-success state will only have outgoing transitions labelled $\omega$.

**Definition 4.1 [$\omega$-respecting]** A pLTS $\langle S, \Omega, \rightarrow \rangle$ is said to be *$\omega$-respecting* whenever $s \xrightarrow{\omega}$, for any $\omega \in \Omega$, implies $s \stackrel{\tau}{\not\longrightarrow}$.

It is straightforward to modify the pLTS of applications of tests to processes into one that it is $\omega$-respecting, namely by removing all transitions $s \xrightarrow{\tau} \Delta$ for states $s$ with $s \xrightarrow{\omega}$. With $[\Theta \| \Delta]$ we denote the distribution $\Theta \| \Delta$ in this pruned pLTS.

Secondly, we recall the definition of weak derivations from [3]. In a pLTS actions are only performed by states, in that actions are given by relations from states to distributions. But processes in general correspond to distributions over states, so in order to define what it means for a process to perform an action, we need to *lift* these relations so that they also apply to distributions. In fact we will find it convenient to lift them to subdistributions.

**Definition 4.2** Let $(S, L, \rightarrow)$ be a pLTS and $\mathscr{R} \subseteq S \times \mathscr{D}_{sub}(S)$ be a relation from states to subdistributions. Then $\overline{\mathscr{R}} \subseteq \mathscr{D}_{sub}(S) \times \mathscr{D}_{sub}(S)$ is the smallest relation that satisfies:

(i) $s \,\mathscr{R}\, \Delta$ implies $\overline{s} \,\overline{\mathscr{R}}\, \Delta$, and

(ii) (Linearity) $\Gamma_i \,\overline{\mathscr{R}}\, \Delta_i$ for $i \in I$ implies $(\sum_{i \in I} p_i \cdot \Gamma_i) \;\overline{\mathscr{R}}\; (\sum_{i \in I} p_i \cdot \Delta_i)$ for any $p_i \in [0,1]$ $(i \in I)$ with $\sum_{i \in I} p_i \leq 1$.

An application of this notion is when the relation is $\xrightarrow{\alpha}$ for $\alpha \in \mathsf{Act}_\tau$; in that case we also write $\xrightarrow{\alpha}$ for $\overline{\xrightarrow{\alpha}}$. Thus, as source of a relation $\xrightarrow{\alpha}$ we now also allow distributions, and even subdistributions. A subtlety of this approach is that for any action $\alpha$, we have $\varepsilon \xrightarrow{\alpha} \varepsilon$ simply by taking $I = \emptyset$ or $\sum_{i \in I} p_i = 0$ in Definition 4.2. That turns out to make $\varepsilon$ especially useful for modelling the "chaotic" aspects of divergence in [3], in particular that in the must-case a divergent process can simulate any other.

**Definition 4.3 [Weak derivation]** Suppose we have subdistributions $\Delta, \Delta_k^{\rightarrow}, \Delta_k^{\times}$, for $k \geq 0$, with the following properties:

$$\begin{array}{rcl} \Delta & = & \Delta_0^{\rightarrow} + \Delta_0^{\times} \\ \Delta_0^{\rightarrow} & \xrightarrow{\tau} & \Delta_1^{\rightarrow} + \Delta_1^{\times} \\ & \vdots & \\ \Delta_k^{\rightarrow} & \xrightarrow{\tau} & \Delta_{k+1}^{\rightarrow} + \Delta_{k+1}^{\times} \; . \\ & \vdots & \end{array}$$

Then we call $\Delta' := \sum_{k=0}^{\infty} \Delta_k^{\times}$ a *weak derivative* of $\Delta$, and write $\Delta \Longrightarrow \Delta'$ to mean that $\Delta$ can make a *weak derivation* to its derivative $\Delta'$.

There is always at least one weak derivative of any subdistribution (the subdistribution itself) and there can be many.

Thirdly, we identify a class of special weak derivatives called extreme derivatives.

**Definition 4.4 [Extreme derivatives]** A state $s$ in a pLTS is called *stable* if $s \stackrel{\tau}{\not\longrightarrow}$, and a subdistribution $\Delta$ is called *stable* if every state in its support is stable. We write $\Delta \Longrightarrow\!\!\!\!\succ \Delta'$ whenever $\Delta \Longrightarrow \Delta'$ and $\Delta'$ is stable, and call $\Delta'$ an *extreme* derivative of $\Delta$.

Referring to Definition 4.3, we see this means that in the extreme derivation of $\Delta'$ from $\Delta$ at every stage a state must move on if it can, so that every stopping component can contain only states which *must* stop: for $s \in \lceil \Delta_k^{\rightarrow} + \Delta_k^{\times} \rceil$ we have $s \in \lceil \Delta_k^{\times} \rceil$ if *and now also* only if $s \stackrel{\tau}{\not\longrightarrow}$. Moreover if the pLTS is $\omega$-respecting then whenever $s \in \lceil \Delta_k^{\rightarrow} \rceil$, it is not successful, i.e. $s \stackrel{\omega}{\not\longrightarrow}$ for every $\omega \in \Omega$.

**Lemma 4.5 [Existence of extreme derivatives]**

(i) For every subdistribution $\Delta$ there exists some (stable) $\Delta'$ such that $\Delta \Longrightarrow\!\!\!\!\succ \Delta'$.

(ii) In a deterministic pLTS if $\Delta \Longrightarrow\!\!\!\!\succ \Delta'$ and $\Delta \Longrightarrow\!\!\!\!\succ \Delta''$ then $\Delta' = \Delta''$.

*Sub*distributions are essential here. Consider a state $t$ that has only one transition, a self $\tau$-loop $t \xrightarrow{\tau} \bar{t}$. Then it diverges and it has a unique extreme derivative $\varepsilon$, the empty subdistribution. More generally, suppose a subdistribution $\Delta$ diverges, that is there is an infinite sequence of internal transitions $\Delta \xrightarrow{\tau} \Delta_1 \xrightarrow{\tau} \ldots \Delta_k \xrightarrow{\tau} \ldots$. Then one extreme derivative of $\Delta$ is $\varepsilon$, but it may have others.

The final ingredient in the definition of a set of outcomes of an application of a test to a process is the outcome of a particular extreme derivative. Note that all states $s \in \lceil \Delta \rceil$ in the support of an extreme derivative either satisfy $s \xrightarrow{\omega}$ for a unique $\omega \in \Omega$, or have $s \not\longrightarrow$.

**Definition 4.6 [Outcomes]** The outcome $\$\Delta \in [0,1]^{\Omega}$ of a stable subdistribution $\Delta$ is given by $\$\Delta(\omega) = \sum\{\Delta(s) \mid s \in \lceil \Delta \rceil,\ s \xrightarrow{\omega}\}$.

Putting all four ingredients together, we arrive at a definition of $\mathscr{A}^{\mathrm{d}}(\Theta, \Delta)$:

**Definition 4.7** Let $\Delta$ be a process and $\Theta$ an $\Omega$-test. Then $\mathscr{A}^{\mathrm{d}}(\Theta, \Delta) = \{\$\Lambda \mid [\Theta \| \Delta] \Longrightarrow\!\!\!\!\succ \Lambda\}$.

The role of pruning in the above definition can be seen via the following example.

**Example 4.8** Let $\overline{p}$ be a process that first does an $a$-action, to the point distribution $\overline{q}$, and then diverges, via the $\tau$-loop $q \xrightarrow{\tau} \overline{q}$. Let $\bar{t}$ be the test used in Examples 3.2 and 3.3. Then $\bar{t} \| \overline{p}$ has a unique extreme derivative $\varepsilon$, whereas $[\bar{t} \| \overline{p}]$ has a unique extreme derivative $[\omega \| q]$. Here we give the name $\omega$ to the state reachable from $\bar{t}$ with the outgoing $\omega$-transition. The outcome in $\mathscr{A}^{\mathrm{d}}(\bar{t}, \overline{p})$ shows that process $\overline{p}$ passes test $\bar{t}$ with probability 1, which is what we expect for state-based testing. Without pruning we would get an outcome saying that $\overline{p}$ passes $\bar{t}$ with probability 0. □

As this example is nonprobabilistic, it also illustrates how pruning enables the standard notion of nonprobabilistic testing to be captured by derivation-based testing.

**Example 4.9** (Revisiting Example 3.2.) The pLTS in Figure 4(c) is deterministic and unaffected by pruning; from part (ii) of Lemma 4.5 it follows that $\overline{t\|q_1}$ has a unique extreme derivative $\Lambda$. Moreover $\Lambda$ can be calculated to be $\sum_{k\geq 1}\frac{1}{2^k}\cdot\overline{s_3}$, which simplifies to the distribution $\overline{s_3}$. Therefore, $\mathscr{A}^{\mathrm{d}}(\overline{t},\overline{q_1}) = \{\$\overline{s_3}\} = \{\overrightarrow{\omega}\}$. □

**Example 4.10** (Revisiting Example 3.3.) The application of the test $\overline{t}$ to processes $\overline{q_2}$ is outlined in Figure 5(c). Consider any extreme derivative $\Delta'$ from $s_0 = [\overline{t}\|\overline{q_2}]$; note that here again pruning actually has no effect. Using the notation of Definition 4.3, it is clear that $\Delta_0^{\times}$ and $\Delta_0^{\rightarrow}$ must be $\varepsilon$ and $\overline{s_0}$ respectively. Similarly, $\Delta_1^{\times}$ and $\Delta_1^{\rightarrow}$ must be $\varepsilon$ and $\overline{s_1}$ respectively. But $s_1$ is a nondeterministic state, having two possible transitions:

(i) $s_1 \xrightarrow{\tau} \Lambda_0$ where $\Lambda_0$ has support $\{s_0, s_2\}$ and assigns each of them the weight $\frac{1}{2}$

(ii) $s_1 \xrightarrow{\tau} \Lambda_1$ where $\Lambda_1$ has the support $\{s_3, s_4\}$, again dividing the mass equally among them.

So there are many possibilities for $\Delta_2$; from Definition 4.3 one sees that in fact $\Delta_2$ can be of the form

$$p\cdot\Lambda_0 + (1-p)\cdot\Lambda_1 \tag{5}$$

for any choice of $p\in[0,1]$.

Let us consider one possibility, an extreme one where $p$ is chosen to be 0; only the transition (ii) above is used. Here $\Delta_2^{\rightarrow}$ is the subdistribution $\frac{1}{2}\overline{s_4}$, and $\Delta_k^{\rightarrow} = \varepsilon$ whenever $k>2$. A simple calculation shows that in this case the extreme derivative generated is $\Lambda_1^e = \frac{1}{2}\overline{s_3} + \frac{1}{2}\overline{s_6}$ which implies that $\frac{1}{2}\overrightarrow{\omega} \in \mathscr{A}^{\mathrm{d}}(\overline{t},\overline{q_2})$.

Another possibility for $\Delta_2$ is $\Lambda_0$, corresponding to $p=1$ in (5) above. Continuing this derivation leads to $\Delta_3$ being $\frac{1}{2}\cdot\overline{s_1} + \frac{1}{2}\cdot\overline{s_5}$; thus $\Delta_3^{\times} = \frac{1}{2}\cdot\overline{s_5}$ and $\Delta_3^{\rightarrow} = \frac{1}{2}\cdot\overline{s_1}$. Now in the generation of $\Delta_4$ from $\Delta_3^{\rightarrow}$ again we resolve a transition from the nondeterministic state $s_1$, by choosing some arbitrary $p\in[0,1]$ in (5). Suppose we choose $p=1$ every time, completely ignoring transition (ii) above. Then the extreme derivative generated is

$$\Lambda_0^e = \sum_{k\geq 1}\frac{1}{2^k}\cdot\overline{s_5}$$

which simplifies to the distribution $\overline{s_5}$. This in turn means that $\overrightarrow{\omega} \in \mathscr{A}^{\mathrm{d}}(\overline{t},\overline{q_2})$.

We have seen two possible derivations of extreme derivatives from $\overline{s_0}$. But there are many others. In general whenever $\Delta_k^{\rightarrow}$ is of the form $q\cdot\overline{s_1}$ we have to resolve the nondeterminism by choosing a $p\in[0,1]$ in (5) above; moreover each such choice is independent. It turns out that every extreme derivative $\Delta'$ of $\overline{s_0}$ is of the form $q\cdot\Lambda_0^e + (1-q)\cdot\Lambda_1^e$ for some choice of $q\in[0,1]$, which implies that $\mathscr{A}^{\mathrm{d}}(\overline{t},\overline{q_2})$ is the convex closure of the set $\{\frac{1}{2}\overrightarrow{\omega}, \overrightarrow{\omega}\}$. □

We have now seen two ways of associating sets of outcomes with the application of a test to a process. The first, in Section 3, associates with a test and a process a set of deterministic structures called resolutions, while the second, in this section, uses extreme derivations in which nondeterministic choices are resolved dynamically as the derivation proceeds. We proceed to show that these two approaches give rise to the same outcomes. The key result to this end is

**Proposition 4.11** Let $\Lambda$ be a subdistribution in an $\omega$-respecting deterministic pLTS $\langle R,\Omega,\rightarrow_R\rangle$. If $\Lambda \Longrightarrow\!\!\!\succ \Lambda'$ then $\mathrm{Exp}_{\Lambda}(\mathbb{V}_{\langle R,\Omega,\rightarrow_R\rangle}) = \mathrm{Exp}_{\Lambda'}(\mathbb{V}_{\langle R,\Omega,\rightarrow_R\rangle})$.

To obtain it, we need the crucial property that the evaluation function $\mathbb{V}$ applied to $\omega$-respecting deterministic pLTSs is continuous (with respect to the standard Euclidean metric).

The next proposition maintains that for each extreme derivative there is a corresponding resolution, and vice versa.

**Proposition 4.12** Let $\Delta$ be a subdistribution over the state set of a pLTS $\langle S,\Omega,\rightarrow\rangle$.

(i) Suppose $\Delta \Longrightarrow\!\!\!\succ \Delta'$. Then there is a resolution $\langle R,\Lambda,\rightarrow_R\rangle$ of $\Delta$, with resolving function $f$, such that $\Lambda \Longrightarrow\!\!\!\succ_R \Lambda'$ for some $\Lambda'$ for which $\Delta' = \mathrm{Img}_f(\Lambda')$.

(ii) Suppose $\langle R,\Lambda,\rightarrow_R\rangle$ is a resolution of a $\Delta$ with resolving function $f$. Then $\Lambda \Longrightarrow\!\!\!\succ_R \Lambda'$ implies $\Delta \Longrightarrow\!\!\!\succ \mathrm{Img}_f(\Lambda')$.

The definitions of outcomes, resolutions and the functional $\mathscr{R}$ directly imply that if $\langle R,\Lambda,\rightarrow_R\rangle$ is a resolution of a subdistribution $\Delta\in\mathscr{D}_{sub}(S)$ in a pLTS $\langle S,\Omega,\rightarrow\rangle$, with resolving function $f$, and $\Lambda'\in\mathscr{D}_{sub}(R)$ is stable, then $\mathrm{Img}_f(\Lambda')$ is stable and

$$\mathrm{Exp}_{\Lambda'}(\mathbb{V}_{\langle R,\Omega,\rightarrow_R\rangle}) = \$\Lambda' = \$(\mathrm{Img}_f(\Lambda')).$$

In combination with Propositions 4.11 and 4.12, this yields:

**Corollary 4.13** In an $\omega$-respecting pLTS $\langle S,\Omega,\rightarrow\rangle$, the following statements hold.

(i) If $\Delta \Longrightarrow\!\!\!\succ \Delta'$ then there is a resolution $\langle R,\Lambda,\rightarrow_R\rangle$ of $\Delta$ such that $\mathrm{Exp}_\Lambda(\mathbb{V}_{\langle R,\Omega,\rightarrow_R\rangle}) = \$(\Delta')$.

(ii) For any resolution $\langle R,\Lambda,\rightarrow_R\rangle$ of $\Delta$, there exists an extreme derivative $\Delta'$ such that $\Delta \Longrightarrow\!\!\!\succ \Delta'$ and $\mathrm{Exp}_\Lambda(\mathbb{V}_{\langle R,\Omega,\rightarrow_R\rangle}) = \$(\Delta')$.

Together with an argument that pruning does not affect $\mathscr{A}(\Theta,\Delta)$, this proves:

**Theorem 4.14** For any test $\Theta$ and process $\Delta$ we have that $\mathscr{A}^{\mathrm{d}}(\Theta,\Delta) = \mathscr{A}(\Theta,\Delta)$.

## 5 Agreement of nonnegative- and real-reward must testing

In this section we prove the agreement of $\sqsubseteq_{\mathbf{nr}\mathrm{must}}$ with $\sqsubseteq_{\mathbf{rr}\mathrm{must}}$ for finitary convergent processes, by using failure simulation [3] as a stepping stone. We start with defining the weak action relations $\stackrel{\alpha}{\Longrightarrow}$ for $\alpha\in\mathsf{Act}_\tau$ and the refusal relations $\stackrel{A}{\not\longrightarrow}$ for $A\subseteq\mathsf{Act}$ that are the key ingredients in the definition of the failure-simulation preorder.

**Definition 5.1** Let $\Delta$ and its variants be subdistributions in a pLTS $\langle S,\mathsf{Act},\rightarrow\rangle$.

- For $a\in\mathsf{Act}$ write $\Delta \stackrel{a}{\Longrightarrow} \Delta'$ whenever $\Delta \Longrightarrow \Delta^{\mathrm{pre}} \stackrel{a}{\longrightarrow} \Delta^{\mathrm{post}} \Longrightarrow \Delta'$, for some $\Delta^{\mathrm{pre}}$ and $\Delta^{\mathrm{post}}$. Extend this to $\mathsf{Act}_\tau$ by allowing as a special case that $\stackrel{\tau}{\Longrightarrow}$ is simply $\Longrightarrow$, i.e. including identity (rather than requiring at least one $\stackrel{\tau}{\longrightarrow}$).
- For $A\subseteq\mathsf{Act}$ and $s\in S$ write $s \stackrel{A}{\not\longrightarrow}$ if $s \stackrel{\alpha}{\not\longrightarrow}$ for every $\alpha\in A\cup\{\tau\}$; write $\Delta \stackrel{A}{\not\longrightarrow}$ if $s \stackrel{A}{\not\longrightarrow}$ for every $s\in\lceil\Delta\rceil$.
- More generally write $\Delta \Longrightarrow\stackrel{A}{\not\longrightarrow}$ if $\Delta \Longrightarrow \Delta^{\mathrm{pre}}$ for some $\Delta^{\mathrm{pre}}$ such that $\Delta^{\mathrm{pre}} \stackrel{A}{\not\longrightarrow}$.

**Definition 5.2 [Failure simulation preorder]** Define $\lhd_{FS}$ to be the largest relation in $S\times\mathscr{D}_{sub}(S)$ such that if $s \lhd_{FS} \Delta$ then

(i) whenever $\overline{s} \stackrel{\alpha}{\Longrightarrow} \Gamma'$, for $\alpha\in\mathsf{Act}_\tau$, then there is a $\Delta'\in\mathscr{D}_{sub}(S)$ with $\Delta \stackrel{\alpha}{\Longrightarrow} \Delta'$ and $\Gamma' \mathrel{\overline{\lhd_{FS}}} \Delta'$,

(ii) and whenever $\overline{s} \Longrightarrow\stackrel{A}{\not\longrightarrow}$ then $\Delta \Longrightarrow\stackrel{A}{\not\longrightarrow}$.

Any relation $\mathscr{R}\subseteq S\times\mathscr{D}_{sub}(S)$ that satisfies the two clauses above is called a *failure simulation*. The failure simulation preorder $\sqsubseteq_{FS}\subseteq\mathscr{D}_{sub}(S)\times\mathscr{D}_{sub}(S)$ is defined by letting $\Delta\sqsubseteq_{FS}\Gamma$ whenever there is a $\Delta^\natural$ with $\Delta \Longrightarrow \Delta^\natural$ and $\Gamma \mathrel{\overline{\lhd_{FS}}} \Delta^\natural$.

Note that the simulating process, $\Delta$, occurs at the right of $\lhd_{FS}$, but at the left of $\sqsubseteq_{FS}$.

The failure simulation preorder is preserved under parallel composition with a test, followed by pruning, and it is sound and complete for probabilistic must testing of finitary processes.

**Theorem 5.3 [3]** For finitary processes $\Delta$ and $\Gamma$,

(i) If $\Delta \sqsubseteq_{FS} \Gamma$ then for any $\Omega$-test $\Theta$ it holds that $[\Delta \| \Theta] \sqsubseteq_{FS} [\Gamma \| \Theta]$.

(ii) $\Delta \sqsubseteq_{FS} \Gamma$ if and only if $\Delta \sqsubseteq_{\mathrm{pmust}} \Gamma$.

Because we prune our pLTSs before extracting values from them, we will be concerned mainly with $\omega$-respecting structures. Moreover, we require the pLTSs to be *convergent* in the sense that there is no wholly divergent state $s$, i.e. with $s \Longrightarrow \varepsilon$.

**Lemma 5.4** Let $\Delta$ and $\Gamma$ be two subdistributions in an $\omega$-respecting convergent pLTS $\langle S, \Omega, \rightarrow \rangle$. If $\Delta \sqsubseteq_{FS} \Gamma$, then it holds that $\mathscr{V}(\Delta) \supseteq \mathscr{V}(\Gamma)$. Here $\mathscr{V}(\Delta)$ denotes $\{\$\Delta' \mid \Delta \Longrightarrow\!\!\succ \Delta'\}$.

This lemma shows that the failure-simulation preorder is a very strong relation in the sense that if $\Delta$ is related to $\Gamma$ by the failure-simulation preorder then the set of outcomes generated by $\Delta$ includes the set of outcomes given by $\Gamma$. It is mainly due to this strong requirement that we can show that the failure-simulation preorder is sound for the real-reward must-testing preorder. Convergence is a crucial condition in this lemma.

**Theorem 5.5** For any finitary convergent processes $\Delta$ and $\Gamma$, if $\Delta \sqsubseteq_{FS} \Gamma$ then we have that $\Delta \sqsubseteq_{\mathbf{rr}\mathrm{must}} \Gamma$.

The proof of the above theorem is subtle. The failure-simulation preorder is defined via weak derivations (cf. Definition 5.2), while the reward must-testing preorder is defined in terms of resolutions (cf. Definition 3.5). Fortunately, we have shown in Corollary 4.14 that we can just as well characterise the reward must-testing preorder in terms of weak derivations. Based on this observation, the proof can be carried out by exploiting Theorem 5.3(i) and Lemma 5.4.

This result does not extend to divergent processes. One witness example is given in Figure 1. A simpler example is as follows. Let $\Delta$ be a process that diverges, by performing a $\tau$-loop only, and let $\Gamma$ be a process that merely performs a single action $a$. It holds that $\Delta \sqsubseteq_{FS} \Gamma$ because $\Delta \Longrightarrow \varepsilon$ and the empty subdistribution can failure-simulate any processes. However, if we apply the test $\bar{t}$ from Example 3.2 again, and the reward tuple $h$ with $h(\omega) = -1$, then

$$\begin{array}{rclclcr}
\bigsqcap h \cdot \mathscr{A}^{\mathrm{d}}(\bar{t}, \Delta) & = & \bigsqcap h \cdot \{\$\varepsilon\} & = & \bigsqcap \{0\} & = & 0 \\
\bigsqcap h \cdot \mathscr{A}^{\mathrm{d}}(\bar{t}, \Gamma) & = & \bigsqcap h \cdot \{\overrightarrow{\omega}\} & = & \bigsqcap \{-1\} & = & -1
\end{array}$$

As $\bigsqcap h \cdot \mathscr{A}^{\mathrm{d}}(\bar{t}, \Delta) \not\leq \bigsqcap h \cdot \mathscr{A}^{\mathrm{d}}(\bar{t}, \Gamma)$, we see that $\Delta \not\sqsubseteq_{\mathbf{rr}\mathrm{must}} \Gamma$. Since $\mathscr{V}([\bar{t} \| \Gamma]) = \{\overrightarrow{\omega}\}$ but $\overrightarrow{\omega} \notin \mathscr{V}([\bar{t} \| \Delta])$, this also is a counterexample against an extension of Lemma 5.4 with divergence.

Finally, by combining Theorems 3.6(ii) and 5.3(ii), together with Theorem 5.5, we obtain the main result of the paper which states that, in the absence of divergence, nonnegative-reward must testing is as discriminating as real-reward must testing.

**Theorem 5.6** For any finitary convergent processes $\Delta$ and $\Gamma$, it holds that $\Delta \sqsubseteq_{\mathbf{rr}\mathrm{must}} \Gamma$ if and only if $\Delta \sqsubseteq_{\mathbf{nr}\mathrm{must}} \Gamma$.

## 6 Conclusion

We have studied a notion of real-reward testing which extends the traditional nonnegative-reward testing with negative rewards. It turned out that real-reward may preorder is the inverse of real-reward must preorder, and vice versa. More interestingly, for finitary convergent processes, the real-reward must testing preorder coincides with the nonnegative-reward testing preorder. In order to prove this result, we have presented two testing approaches and shown their coincidence, which involved proving some analytic properties such as the continuity of a function for calculating testing outcomes.

Although for finitary convergent processes real-reward must testing is no more powerful than nonnegative-reward must testing, the same does not hold for may testing. This is immediate from our result that (the inverse of) real-reward may testing is as powerful as real-reward must testing, that is known not to hold for nonnegative-reward may- and must testing. Thus, real-reward may testing is strictly more discriminating than nonnegative-reward may testing, even without divergence.

## References

[1] R. De Nicola & M. Hennessy (1984): *Testing equivalences for processes*. *Theoretical Computer Science* 34, pp. 83–133, doi:10.1016/0304-3975(84)90113-0.

[2] Y. Deng, R.J. van Glabbeek, M. Hennessy & C.C. Morgan (2008): *Characterising testing preorders for finite probabilistic processes*. *Logical Methods in Computer Science* 4(4):4, doi:10.2168/LMCS-4(4:4)2008.

[3] Y. Deng, R.J. van Glabbeek, M. Hennessy & C.C. Morgan (2009): *Testing finitary probabilistic processes*. In: Proc. *CONCUR'09*, LNCS 5710, Springer, pp. 274–288, doi:10.1007/978-3-642-04081-8_19.

[4] Y. Deng, R.J. van Glabbeek, M. Hennessy & C.C. Morgan (2010): *Real Reward Testing for Probabilistic Processes*. Full version of the current paper. Available at http://basics.sjtu.edu.cn/~yuxin/temp/reward.pdf.

[5] Y. Deng, R.J. van Glabbeek, M. Hennessy, C.C. Morgan & C. Zhang (2007): *Remarks on Testing Probabilistic Processes*. *ENTCS* 172, pp. 359–397, doi:10.1016/j.entcs.2007.02.013.

[6] Y. Deng, R.J. van Glabbeek, C.C. Morgan & C. Zhang (2007): *Scalar Outcomes Suffice for Finitary Probabilistic Testing*. In: Proceedings *ESOP'07*, LNCS 4421, Springer, pp. 363–368, doi:10.1007/978-3-540-71316-6_25.

[7] M. Hennessy (1988): *An Algebraic Theory of Processes*. MIT Press.

[8] B. Jonsson, C. Ho-Stuart & Wang Yi (1994): *Testing and Refinement for Nondeterministic and Probabilistic Processes*. In: Proceedings *FTRTFT'94*, LNCS 863, Springer, pp. 418–430, doi:10.1007/3-540-58468-4_176.

[9] M.L. Puterman (1994): *Markov Decision Processes*. Wiley, doi:10.1002/9780470316887.

[10] J.J.M.M. Rutten, M.Kwiatkowska, G. Norman & D. Parker (2004): *Mathematical Techniques for Analyzing Concurrent and Probabilistic Systems,* P. Panangaden and F. van Breugel (eds.). *CRM Monograph Series* 23, American Mathematical Society.

[11] R. Segala (1995): *Modeling and Verification of Randomized Distributed Real-Time Systems*. Ph.D. thesis, MIT.

[12] R. Segala (1996): *Testing Probabilistic Automata*. In: Proceedings *CONCUR'96*, LNCS 1119, Springer, pp. 299–314, doi:10.1007/3-540-61604-7_62.

[13] Wang Yi & K.G. Larsen (1992): *Testing Probabilistic and Nondeterministic Processes*. In: Proc. *PSTV'92, IFIP Transactions* C-8, North-Holland, pp. 47–61.