Characterising Testing Preorders for Finite Probabilistic Processes

Yuxin Deng[*] — Jiao Tong University, Shanghai, China
Rob van Glabbeek — National ICT Australia, Sydney, Australia
(Yuxin Deng, Rob van Glabbeek) School of CSE, University of New South Wales, Sydney, Australia

Matthew Hennessy — Department of Informatics, University of Sussex, Falmer, Brighton, UK

Carroll Morgan[*]
Chenyi Zhang — National ICT Australia
(Carroll Morgan, Chenyi Zhang) School of Computer Science and Engineering, The University of New South Wales, Sydney, Australia

Abstract

In 1992 Wang & Larsen extended the may- and must preorders of De Nicola and Hennessy to processes featuring probabilistic as well as nondeterministic choice. They concluded with two problems that have remained open throughout the years, namely to find complete axiomatisations and alternative characterisations for these preorders. This paper solves both problems for finite processes with silent moves. It characterises the may preorder in terms of simulation, and the must preorder in terms of failure simulation. It also gives a characterisation of both preorders using a modal logic. Finally it axiomatises both preorders over a probabilistic version of CSP.

1. Introduction

A satisfactory semantic theory for processes which encompass both nondeterministic and probabilistic behaviour has been a long-standing research problem [12, 35, 23, 17, 32, 33, 30, 18, 27, 31, 13, 21, 26, 1, 19, 24, 3, 34, 7]. In 1992 Wang & Larsen posed the problems of finding complete axiomatisations and alternative characterisations for a natural generalisation of the standard testing preorders [6] to such processes [35]. Here we solve both problems, at least for finite processes, by providing a detailed account of both may- and must testing preorders for a finite version of the process calculus CSP extended with probabilistic choice. For each preorder we provide three independent characterisations, using (i) co-inductive simulation relations, (ii) a modal logic and (iii) sets of inequations.

Testing processes: Our starting point is the finite process calculus pCSP [8] obtained by adding a probabilistic choice operator to finite CSP; like others who have done the same, we now have *three* choice operators, external $P \Box Q$, internal $P \sqcap Q$ and the newly added probabilistic choice $P\,{}_p\!\oplus Q$. So a semantic theory for pCSP will have to provide a coherent account of the precise relationships between these operators.

As a first step, in Sec. 2 we provide an interpretation of pCSP as a *probabilistic labelled transition system*, in which, following [32], transitions like $s \xrightarrow{\alpha} s'$ from standard labelled transition systems are generalised to the form $s \xrightarrow{\alpha} \Delta$, where Δ is a *distribution*, a mapping assigning probabilities to states. With this interpretation we obtain in Sec. 3 a version of the testing preorders of [6] for pCSP processes, $\sqsubseteq_{\text{pmay}}$ and $\sqsubseteq_{\text{pmust}}$. These are based on the ability of processes to pass *tests*; the tests we use are simply pCSP processes in which certain *states* are marked as *success states*. See [8] for a detailed discussion of the power of such tests.

The object of this paper is to give useful characterisations of these testing preorders. This problem was addressed previously by Segala in [31], but using testing preorders $\hat{\sqsubseteq}^{\Omega}_{\text{pmay}}$ and $\hat{\sqsubseteq}^{\Omega}_{\text{pmust}}$ that differ in two ways from the ones in [6, 14, 35, 8] and the present paper. First of all, in [31] the success of a test is achieved by the *actual execution* of a predefined *success action*, rather than the reaching of a success state. We call this an *action*-based, as opposed to a *state*-based, approach. Secondly, [31] employs a countable number of success actions instead of a single one; we call this *vector-based*, as opposed to *scalar*, testing. Segala's results in [31] depend crucially on this form of testing. To achieve our current results, we need Segala's preorders as a stepping stone. We relate them to ours by considering intermediate preorders $\hat{\sqsubseteq}_{\text{pmay}}$ and $\hat{\sqsubseteq}_{\text{pmust}}$ that arise from action-based but scalar testing, and use a recent result [9] saying that for finite processes the preorders $\hat{\sqsubseteq}^{\Omega}_{\text{pmay}}$ and $\hat{\sqsubseteq}^{\Omega}_{\text{pmust}}$ coincide with $\hat{\sqsubseteq}_{\text{pmay}}$ and $\hat{\sqsubseteq}_{\text{pmust}}$. Here we show that on pCSP $\hat{\sqsubseteq}_{\text{pmay}}$ and $\hat{\sqsubseteq}_{\text{pmust}}$ also coincide with $\sqsubseteq_{\text{pmay}}$ and $\sqsubseteq_{\text{pmust}}$.[1]

Simulation preorders: In Sec. 4 we use the transitions $s \xrightarrow{\alpha} \Delta$ to define two co-inductive preorders, the *simula-*

[*]We acknowledge the support of the Australian Research Council (ARC) Grant DP034557.

[1]However in the presence of divergence they are slightly different.

tion preorder $\sqsubseteq_S$ [30, 24, 8], and the novel *failure simulation* preorder $\sqsubseteq_{FS}$ over pCSP processes. The latter extends the failure simulation preorder of [10] to probabilistic processes. Their definition uses a natural generalisation of the transitions, first (Kleisli-style) to take the form $\Delta \xrightarrow{\alpha} \Delta'$, and then to *weak* versions $\Delta \stackrel{\alpha}{\Longrightarrow} \Delta'$. The latter preorder differs from the former in the use of a *failure* predicate $s \stackrel{X}{\not\longrightarrow}$, indicating that in the state s none of the actions in X can be performed.

Both preorders are preserved by all the operators in pCSP, and are *sound* with respect to the testing preorders; that is $P \sqsubseteq_S Q$ implies $P \sqsubseteq_{\text{pmay}} Q$ and $P \sqsubseteq_{FS} Q$ implies $P \sqsubseteq_{\text{pmust}} Q$. For $\sqsubseteq_S$ this was established in [8], and the proofs for $\sqsubseteq_{FS}$ are similar. But *completeness*, that the testing preorders imply the respective simulation preorders, requires some ingenuity. We prove it indirectly, involving a characterisation of the testing and simulation preorders in terms of a modal logic.

Modal logic: Our modal logic, defined in Sec. 7, uses finite conjunction $\bigwedge_{i\in I}\varphi_i$, the modality $\langle a\rangle\varphi$ from the Hennessy-Milner Logic, and a novel probabilistic construct $\bigoplus_{i\in I} p_i\cdot\varphi_i$. A satisfaction relation between processes and formulae then gives, in a natural manner, a *logical preorder* between processes: $P \sqsubseteq^{\mathcal{L}} Q$ means that every $\mathcal{L}$-formula satisfied by P is also satisfied by Q. We establish that $\sqsubseteq^{\mathcal{L}}$ coincides with $\sqsubseteq_S$ and $\sqsubseteq_{\text{pmay}}$.

To capture failures, we add, for every set of actions X, a formula $\mathbf{ref}(X)$ to our logic, satisfied by any process which, after it can do no further internal actions, can perform none of the actions in X either. The constructs $\bigwedge$, $\langle a\rangle$ and $\mathbf{ref}()$ stem from the modal characterisation of the non-probabilistic failure simulation preorder, given in [10]. We show that $\sqsubseteq_{\text{pmust}}$, as well as $\sqsubseteq_{FS}$, can be characterised in a similar manner with this extended modal logic.

Proof strategy: We prove these characterisation results through two cycles of inclusions:

$$\begin{array}{ccccccccccc}
\sqsubseteq^{\mathcal{L}} & \subseteq & \sqsubseteq_S & \overset{[8]}{\subseteq} & \sqsubseteq_{\text{pmay}} & \subseteq & \widehat{\sqsubseteq}_{\text{pmay}} & \overset{[9]}{=} & \widehat{\sqsubseteq}^{\Omega}_{\text{pmay}} & \subseteq & \sqsubseteq^{\mathcal{L}} \\
\sqsubseteq^{\mathcal{F}} & \subseteq & \sqsubseteq_{FS} & \subseteq & \sqsubseteq_{\text{pmust}} & \subseteq & \widehat{\sqsubseteq}_{\text{pmust}} & \overset{[9]}{=} & \widehat{\sqsubseteq}^{\Omega}_{\text{pmust}} & \subseteq & \sqsubseteq^{\mathcal{F}} \\
\underbrace{\qquad}_{\textit{Sec. 7}} & & \underbrace{\qquad}_{\textit{Sec. 4}} & & \underbrace{\qquad}_{\textit{Sec. 3}} & & \underbrace{\qquad}_{\textit{Sec. 5}} & & \underbrace{\qquad}_{\textit{Sec. 6}} & & \underbrace{\qquad}_{\textit{Sec. 8}}
\end{array}$$

In Sec. 7 we show that $P \sqsubseteq^{\mathcal{L}} Q$ implies $P \sqsubseteq_S Q$ (and hence $P \sqsubseteq_{\text{pmay}} Q$), and likewise for $\sqsubseteq^{\mathcal{F}}$ and $\sqsubseteq_{FS}$; the proof involves constructing, for each pCSP process P, a *characteristic formula* φ_P. To obtain the other direction, in Sec. 8 we show how every modal formula φ can be captured, in some sense, by a test T_φ; essentially the ability of a pCSP process to satisfy φ is determined by its ability to pass the test T_φ. We capture the conjunction of two formulae by a probabilistic choice between the corresponding tests; in order to prevent the results from these tests getting mixed up, we employ the vector-based tests of [31], so that we can use different success actions in the separate probabilistic branches. Therefore, we complete our proof by demonstrating that the state-based testing preorders imply the action-based ones (Sec. 5) and recalling the result from [9] that the action-based scalar testing preorders imply the vector-based ones (Sec. 6).

Equations: It is well-known that may- and must testing for standard CSP can be captured equationally [6, 2, 14]. In [8] we showed that most of the standard equations are no longer valid in the probabilistic setting of pCSP; we also provided a set of axioms which are complete with respect to (probabilistic) may-testing for the sub-language of pCSP without probabilistic choice. Here we extend this result, by showing, in Sec. 10, that both $P \sqsubseteq_{\text{pmay}} Q$ and $P \sqsubseteq_{\text{pmust}} Q$ can still be captured equationally over full pCSP. In the may case the essential (in)equation required is

$$a.(P \,{}_p{\oplus}\, Q) \quad \sqsubseteq \quad a.P \,{}_p{\oplus}\, a.Q$$

The must case is more involved: in the absence of the distributivity of the external and internal choices over each other, to obtain completeness we require a complicated inequational schema.

2. Finite probabilistic CSP

Let Act be a finite set of actions, ranged over by $a, b, \cdots$, which processes can perform. Then the finite probabilistic CSP processes are given by the following two-sorted syntax:

$$\begin{array}{lcl}
P & ::= & S \mid P \,{}_p{\oplus}\, P \\
S & ::= & \mathbf{0} \mid a.P \mid P \sqcap P \mid S \,\Box\, S \mid S \,|_A\, S
\end{array}$$

Here $P \,{}_p{\oplus}\, Q$, for $0 < p < 1$, represents a *probabilistic choice* between P and Q: with probability p it will act like P and with probability $1{-}p$ it will act like Q. Any process is a probabilistic combination of state-based processes (the sub-sort S above) built by repeated application of the operator ${}_p{\oplus}$. The state-based processes have a CSP-like syntax, involving the stopped process $\mathbf{0}$, action prefixing $a.\_$, *internal-* and *external choices* $\sqcap$ and $\Box$, and a *parallel composition* $|_A$ for $A \subseteq \text{Act}$.

The process $P \sqcap Q$ will first do a so-called *internal action* $\tau \notin \text{Act}$, choosing *nondeterministically* between P and Q. Therefore $\sqcap$, like $a.\_$, acts as a *guard*, in the sense that it converts any process arguments into a state-based process.

The process $P \,\Box\, Q$ on the other hand does not perform actions itself, but merely allows its arguments to proceed, disabling one argument as soon as the other has done a visible action. In order for this process to start from a state rather than a probability distribution of states, we require its arguments to be state-based as well; the same applies to $|_A$. Expressions $P \;\Box\; Q$ and $P \;|_A\; Q$ for processes P and Q that are *not* state-based are therefore syntactic sugar for an expression in the above syntax obtained by distributing $\Box$ and $|_A$ over ${}_p\oplus$.

Finally, the expression $P \;|_A\; Q$, where $A \subseteq \mathsf{Act}$, represents processes P and Q running in parallel. They may synchronise by performing the same action from A simultaneously; such a synchronisation results in τ. In addition P and Q may independently do any action from $(\mathsf{Act}\backslash A) \cup \{\tau\}$.

We write pCSP for the set of process terms defined by this grammar, and sCSP for the subset comprising only the state-based process terms. The full language of CSP [2, 15, 29] has many more operators; we have simply chosen a representative selection, and have added probabilistic choice. Our parallel operator is not a CSP primitive, but it can easily be expressed in terms of them—in particular $P \;|_A\; Q = (P \|_A Q)\backslash A$, where $\|_A$ and $\backslash A$ are the parallel composition and hiding operators of [29]. It can also be expressed in terms of the parallel composition, renaming and restriction operators of CCS. We have chosen this (non-associative) operator for convenience in defining the application of tests to processes.

As usual we may elide $\mathbf{0}$; the prefixing operator $a.\_$ binds stronger than any binary operator; and precedence between binary operators is indicated via brackets or spacing. We will also sometimes use indexed binary operators, such as $\bigoplus_{i\in I} p_i{\cdot}P_i$ with $\sum_{i\in I} p_i = 1$ and all $p_i > 0$, and $\Box_{i\in I} P_i$.

The above intuitions are formalised by an *operational semantics* associating with each process term a graph-like structure representing its possible reactions to users' requests: we use a generalisation of labelled transition systems [25] that includes probabilities.

A (discrete) probability distribution over a set S is a function $\Delta : S \to [0,1]$ with $\sum_{s\in S}\Delta(s) = 1$; the *support* of Δ is given by $\lceil\Delta\rceil = \{\, s \in S \mid \Delta(s) > 0 \,\}$. We write $\mathcal{D}(S)$, ranged over by Δ, Θ, Φ, for the set of all distributions over S with finite support; these finite distributions are sufficient for the results of this paper. We also write $\overline{s}$ to denote the point distribution assigning probability 1 to s and 0 to all others, so that $\lceil\overline{s}\rceil = \{s\}$.

For Δ a distribution over S and function $f : S \to X$ into a vector space X (typically the reals[2]) we write $\bigoplus_{s\in S}\Delta_s{\cdot}f_s$ or $\mathrm{Exp}_\Delta(f)$ for $\sum_{s\in S}\Delta(s){\cdot}f(s)$, the *weighted average* of the f_s, or *expected value* of f. When $p \in [0,1]$, we also write $f_1 \;{}_p\oplus\; f_2$ for $p\cdot f_1 + (1{-}p)\cdot f_2$. More generally, for function $F : S \to \mathcal{P}^+(X)$ with $\mathcal{P}^+(X)$ being the collection of non-empty subsets of X, we define $\mathrm{Exp}_\Delta F := \{\, \mathrm{Exp}_\Delta(f) \mid f \,\overline{\in}\, F \,\}$, where $f \,\overline{\in}\, F$ means that f is a *choice function* with $f(s) \in F(s)$ for all $s \in S$.

[2]Other possibilities are tuples of reals, or distributions over some set.

$$a.P \xrightarrow{a} [\![P]\!]$$

$$P \sqcap Q \xrightarrow{\tau} [\![P]\!] \qquad\qquad P \sqcap Q \xrightarrow{\tau} [\![Q]\!]$$

$$\frac{s_1 \xrightarrow{a} \Delta}{s_1 \;\Box\; s_2 \xrightarrow{a} \Delta} \qquad\qquad \frac{s_2 \xrightarrow{a} \Delta}{s_1 \;\Box\; s_2 \xrightarrow{a} \Delta}$$

$$\frac{s_1 \xrightarrow{\tau} \Delta}{s_1 \;\Box\; s_2 \xrightarrow{\tau} \Delta \;\Box\; s_2} \qquad\qquad \frac{s_2 \xrightarrow{\tau} \Delta}{s_1 \;\Box\; s_2 \xrightarrow{\tau} s_1 \;\Box\; \Delta}$$

$$\frac{s_1 \xrightarrow{\alpha} \Delta \qquad \alpha \notin A}{s_1 \;|_A\; s_2 \xrightarrow{\alpha} \Delta \;|_A\; s_2} \qquad\qquad \frac{s_2 \xrightarrow{\alpha} \Delta \qquad \alpha \notin A}{s_1 \;|_A\; s_2 \xrightarrow{\alpha} s_1 \;|_A\; \Delta}$$

$$\frac{s_1 \xrightarrow{a} \Delta_1,\; s_2 \xrightarrow{a} \Delta_2 \qquad a \in A}{s_1 \;|_A\; s_2 \xrightarrow{\tau} \Delta_1 \;|_A\; \Delta_2}$$

Figure 1. Operational semantics of pCSP.

We now give the probabilistic generalisation (pLTSs) of labelled transition systems (LTSs):

Definition 1 A *probabilistic labelled transition system* is a triple $\langle S, \mathsf{Act}_\tau, \to\rangle$, where

(i) S is a set of states

(ii) Act_τ is a set of actions Act, augmented by $\tau \notin \mathsf{Act}$; we let a range over Act and α over Act_τ.

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

As with LTSs, we usually write $s \xrightarrow{\alpha} \Delta$ for $(s, \alpha, \Delta) \in \to$, $s \xrightarrow{\alpha}$ for $\exists\Delta : s \xrightarrow{\alpha} \Delta$ and $s \to$ for $\exists\alpha : s \xrightarrow{\alpha}$. An LTS may be viewed as a degenerate pLTS, one in which only point distributions are used.

We now define the operational semantics of pCSP by means of a particular pLTS $\langle \mathsf{sCSP}, \mathsf{Act}_\tau, \to\rangle$, constructed by taking sCSP to be the set of states and interpreting pCSP processes P as distributions $[\![P]\!] \in \mathcal{D}(\mathsf{sCSP})$ as follows:

$$\begin{array}{rcll} [\![s]\!] & := & \overline{s} & \text{for } s \in \mathsf{sCSP} \\ [\![P \;{}_p\oplus\; Q]\!] & := & [\![P]\!] \;{}_p\oplus\; [\![Q]\!] \;. & \end{array}$$

Note that for each $P \in \mathsf{pCSP}$ the distribution $[\![P]\!]$ is finite, i.e. it has finite support. The definition of the relations $\xrightarrow{\alpha}$ is given in Fig. 1. These rules are very similar to the standard ones used to interpret CSP as an LTS [29], but modified so that the result of an action is a distribution. We sometimes write $\tau.P$ for $P \sqcap P$, thus giving $\tau.P \xrightarrow{\tau} [\![P]\!]$.

3. Testing pCSP processes

A *test* is a pCSP process except that it may have subterms $\omega.P$ for fresh $\omega \notin \mathsf{Act}_\tau$, a special action reporting success; and the operational semantics above is extended by treating ω like any other action from Act. To apply test T to process P we form the process $T \mid_{\mathsf{Act}} P$ in which *all* visible actions of P must synchronise with T, and define a set of testing outcomes $\mathcal{A}(T, P)$ where each outcome, in $[0, 1]$, arises from a resolution of the nondeterministic choices in $T \mid_{\mathsf{Act}} P$ and gives the probability that this resolution will reach a *success state*, one in which ω is possible.

To this end, we inductively define a *results-gathering* function $\mathbb{V} : S \to \mathcal{P}^{+}([0, 1])$; it extends to type $\mathcal{D}(S) \to \mathcal{P}^{+}([0, 1])$ via the convention $\mathbb{V}(\Delta) := \mathrm{Exp}_{\Delta}\mathbb{V}$.

$$\mathbb{V}(s) \;:=\; \begin{cases} \{1\} & \text{if } s \xrightarrow{\omega}, \\ \bigcup\{\, \mathbb{V}(\Delta) \mid s \xrightarrow{\tau} \Delta \,\} & \text{if } s \not\xrightarrow{\omega},\ s \xrightarrow{\tau}, \\ \{0\} & \text{if } s \not\rightarrow \end{cases}$$

Note that these choices are exhaustive because $T \mid_{\mathsf{Act}} P$ has only τ, ω actions, and that $\mathbb{V}$ is well defined when applied to finite, loop-free pLTSs, such as the one of pCSP.

Definition 2 For any pCSP process P and test T, define

$$\mathcal{A}(T, P) \;:=\; \mathbb{V}[\![T \mid_{\mathsf{Act}} P]\!] \;.$$

With this definition, the general testing framework of [6] yields two testing preorders for pCSP, one based on *may* testing, written $P \sqsubseteq_{\text{pmay}} Q$, and the other on *must* testing, written $P \sqsubseteq_{\text{pmust}} Q$.

Definition 3 The *may*- and *must* preorders are given by

$$\begin{array}{lll} P \sqsubseteq_{\text{pmay}} Q & \text{iff} & \forall \text{ tests } T\colon \mathcal{A}(T, P) \leq_{\text{Ho}} \mathcal{A}(T, Q) \\ P \sqsubseteq_{\text{pmust}} Q & \text{iff} & \forall \text{ tests } T\colon \mathcal{A}(T, P) \leq_{\text{Sm}} \mathcal{A}(T, Q) \end{array}$$

with $\leq_{\text{Ho}}, \leq_{\text{Sm}}$ the Hoare, Smyth preorders on $\mathcal{P}^{+}[0, 1]$.[3]

4. Simulation and failure simulation

Let $\mathcal{R} \subseteq S \times \mathcal{D}(S)$ be a relation from states to distributions. We lift it to a relation $\overline{\mathcal{R}} \subseteq \mathcal{D}(S) \times \mathcal{D}(S)$ by letting $\Delta \,\overline{\mathcal{R}}\, \Theta$ whenever there is an index set I and $p \in \mathcal{D}(I)$ such that

(i) $\Delta = \bigoplus_{i \in I} p_i \cdot \overline{s_i}$,

(ii) For each $i \in I$ there is a distribution Φ_i s.t. $s_i \;\mathcal{R}\; \Phi_i$,

(iii) $\Theta = \bigoplus_{i \in I} p_i \cdot \Phi_i$.

For notational convenience, the lifted versions of the transition relations $\xrightarrow{\alpha}$ for $\alpha \in \mathsf{Act}_{\tau}$ are again denoted $\xrightarrow{\alpha}$.

We write $s \xrightarrow{\hat{\tau}} \Delta$ if either $s \xrightarrow{\tau} \Delta$ or $\Delta = \overline{s}$; again $\Delta_1 \xrightarrow{\hat{\tau}} \Delta_2$ denotes the lifted relation. Thus e.g. we have

$$[\![(a \sqcap b) \,{}_{\frac{1}{2}}\!\oplus (a \sqcap c)]\!] \;\xrightarrow{\hat{\tau}}\; [\![a \,{}_{\frac{1}{2}}\!\oplus ((a \sqcap b) \,{}_{\frac{1}{2}}\!\oplus c)]\!] \qquad [8].$$

We now define the weak transition relation $\stackrel{\hat{\tau}}{\Longrightarrow}$ as the transitive and reflexive closure $\xrightarrow{\hat{\tau}}{}^{*}$ of $\xrightarrow{\hat{\tau}}$, while for $a \neq \tau$ $\Delta_1 \stackrel{\hat{a}}{\Longrightarrow} \Delta_2$ denotes $\Delta_1 \stackrel{\hat{\tau}}{\Longrightarrow}\xrightarrow{a}\stackrel{\hat{\tau}}{\Longrightarrow} \Delta_2$. We write $s \stackrel{X}{\not\rightarrow}$ with $X \subseteq \mathsf{Act}$ when $\forall \alpha \in X \cup \{\tau\} : s \stackrel{\alpha}{\not\rightarrow}$, and $\Delta \stackrel{X}{\not\rightarrow}$ when $\forall s \in \lceil \Delta \rceil : s \stackrel{X}{\not\rightarrow}$.

[3]The Hoare order is defined by $X \leq_{\text{Ho}} Y$ iff $\forall x \in X\colon \exists y \in Y\colon x \leq y$, similarly the Smyth order by $X \leq_{\text{Sm}} Y$ iff $\forall y \in Y\colon \exists x \in X\colon x \leq y$.

Definition 4 A relation $\mathcal{R} \subseteq S \times \mathcal{D}(S)$ is said to be a *failure simulation* if for all $s, \Theta, \alpha, \Delta$ we have that

- $s \;\mathcal{R}\; \Theta \wedge s \xrightarrow{\alpha} \Delta$ implies $\exists \Theta'\colon \Theta \stackrel{\hat{\alpha}}{\Longrightarrow} \Theta' \wedge \Delta \;\overline{\mathcal{R}}\; \Theta'$
- $s \;\mathcal{R}\; \Theta \wedge s \stackrel{X}{\not\rightarrow} \Delta$ implies $\exists \Theta'\colon \Theta \stackrel{\hat{\tau}}{\Longrightarrow} \Theta' \wedge \Theta' \stackrel{X}{\not\rightarrow}$.

We write $s \triangleright_{FS} \Theta$ to mean that there is some failure simulation $\mathcal{R}$ such that $s \;\mathcal{R}\; \Theta$. Similarly, we define *simulation* and $s \triangleright_{S} \Theta$ by dropping the second clause in Def. 4.

Definition 5 The *simulation preorder* $\sqsubseteq_S$ and *failure simulation preorder* $\sqsubseteq_{FS}$ on pCSP are defined as follows:

$$\begin{array}{lll} P \sqsubseteq_S Q & \text{iff} & [\![Q]\!] \stackrel{\hat{\tau}}{\Longrightarrow} \Theta \text{ for some } \Theta \text{ with } [\![P]\!] \;\overline{\triangleright_{S}}\; \Theta \\ P \sqsubseteq_{FS} Q & \text{iff} & [\![P]\!] \stackrel{\hat{\tau}}{\Longrightarrow} \Theta \text{ for some } \Theta \text{ with } [\![Q]\!] \;\overline{\triangleright_{FS}}\; \Theta \,. \end{array}$$

(Note the opposing directions.) The kernels of $\sqsubseteq_S$ and $\sqsubseteq_{FS}$ are called *(failure) simulation equivalence*, denoted $\simeq_S$ and $\simeq_{FS}$, respectively.

We have already shown in [8] that $\sqsubseteq_S$ is a precongruence and that it implies $\sqsubseteq_{\text{pmay}}$. Similar results can be established for $\sqsubseteq_{FS}$ as well. We summarise these facts as follows:

Proposition 1 Suppose $\sqsubseteq \,\in \{\sqsubseteq_S, \sqsubseteq_{FS}\}$. Then $\sqsubseteq$ is a preorder, and if $P_i \sqsubseteq Q_i$ for $i = 1, 2$ then $a.P_1 \sqsubseteq a.Q_1$ for $a \in \mathsf{Act}$ and $P_1 \odot P_2 \;\sqsubseteq\; Q_1 \odot Q_2$ for $\odot \in \{\sqcap, \Box, {}_p\oplus, |_A\}$.

Proof: The case $\sqsubseteq_S$ was proved in [8, Cor. 6.10 and Thm. 6.13]; the case $\sqsubseteq_{FS}$ is analogous. □

Theorem 1

1. If $P \sqsubseteq_S Q$ then $P \sqsubseteq_{\text{pmay}} Q$.
2. If $P \sqsubseteq_{FS} Q$ then $P \sqsubseteq_{\text{pmust}} Q$.

Proof: The first clause was proved in [8, Thm. 6.17]; the second can be shown similarly. □

The next four sections are devoted to obtaining the converse.

5. State- versus action-based testing

Much work on testing [6, 35, 8] uses success *states* marked by outgoing ω-actions; in other work [31, 9], however, it is the *actual execution* of ω that constitutes success. The former, *state-based* testing, leads to the preorders we defined in Sec. 3; the latter, *action-based* testing, leads to slightly different preorders $\widehat{\sqsubseteq}_{\text{pmay}}$ and $\widehat{\sqsubseteq}_{\text{pmust}}$. Without probability there is no difference between $\widehat{\sqsubseteq}_{\text{may}}$ and $\sqsubseteq_{\text{may}}$; but possible divergence makes $\sqsubseteq_{\text{must}}$ strictly more discriminating than $\widehat{\sqsubseteq}_{\text{must}}$, and in fact $\sqsubseteq_{\text{must}}$ coincides with CSP refinement based on failures and divergences [2, 15, 29]. The action-based approach is formalised as in the state-based approach, via a suitable $\widehat{\mathbb{V}}$:

$$\widehat{\mathbb{V}}(s) := \begin{cases} \bigcup\{\widehat{\mathbb{V}}(\Delta) \mid s \xrightarrow{\tau} \Delta\} \cup \{1 \mid s \xrightarrow{\omega}\} & \text{if } s \rightarrow \\ \{0\} & \text{otherwise} \end{cases}$$

Proposition 2

1. If $P \sqsubseteq_{\text{pmay}} Q$ then $P \mathrel{\widehat{\sqsubseteq}}_{\text{pmay}} Q$.
2. If $P \sqsubseteq_{\text{pmust}} Q$ then $P \mathrel{\widehat{\sqsubseteq}}_{\text{pmust}} Q$.

Proof: For any test $\widehat{T}$ construct T by replacing each subterm $\omega.Q$ by $\tau.\omega$; then $\mathbb{V}[\![T \mid_{\mathsf{Act}} P]\!] = \widehat{\mathbb{V}}[\![\widehat{T} \mid_{\mathsf{Act}} P]\!]$ for all pCSP processes P. □

In fact we use the action-based preorders in Thm. 2 below, a (quasi-, thus) converse of Thm. 1; but with Prop. 2 above the two kinds of preorders become identified so that Thms. 1 and 2 are converse to each other.

Theorem 2

1. If $P \mathrel{\widehat{\sqsubseteq}}_{\text{pmay}} Q$ then $P \sqsubseteq_S Q$.
2. If $P \mathrel{\widehat{\sqsubseteq}}_{\text{pmust}} Q$ then $P \sqsubseteq_{FS} Q$.

We set this theorem as our goal in the next three sections.

6. Vector-based testing

This section describes another variation on testing, a richer testing framework due to Segala [31], in which countably many success actions exist: the application of a test to a process yields a set of *vectors* over the real numbers, rather than a set of scalars. The resulting action-based testing preorders will serve as a stepping stone in proving Thm. 2.

Let Ω be a *set* of fresh success actions with $\Omega \cap \mathsf{Act}_\tau = \emptyset$. An Ω-test is again a pCSP process, but this time allowing subterms $\omega.P$ for any $\omega \in \Omega$. Applying such a test to a process yields a non-empty set of test outcome-*tuples* $\widehat{\mathcal{A}}^\Omega(T,P) \subseteq [0,1]^\Omega$. Each *tuple* arises from a resolution within $T \mid_{\mathsf{Act}} P$ of nondeterministic choices into probabilistic choices, and its ω-component gives the probability that this resolution will perform the success action ω.

For vectors we again inductively define a results-gathering function $\widehat{\mathbb{V}}^\Omega_\updownarrow : S \to \mathcal{P}^+[0,1]^\Omega$; it extends to type $\mathcal{D}(S) \to \mathcal{P}^+[0,1]^\Omega$ via $\widehat{\mathbb{V}}^\Omega_\updownarrow(\Delta) := \mathrm{Exp}_\Delta \widehat{\mathbb{V}}^\Omega_\updownarrow$ just as $\mathbb{V}$ and $\widehat{\mathbb{V}}$ did. First, for any α define $\alpha! : [0,1]^\Omega \to [0,1]^\Omega$ so that $\alpha!o(\omega)$ becomes 1 if $\omega{=}\alpha$ but remains $o(\omega)$ otherwise; this function lifts to sets $O \subseteq [0,1]^\Omega$ as usual, via $\alpha!O := \{\alpha!o \mid o \in O\}$. Now we define

$$\widehat{\mathbb{V}}^\Omega_\updownarrow(s) \;:=\; \begin{cases} \updownarrow \bigcup\{\, \alpha!(\widehat{\mathbb{V}}^\Omega_\updownarrow(\Delta)) \mid s \xrightarrow{\alpha} \Delta \,\} & \text{if } s \to \\ \{\vec{0}\} & \text{otherwise} \end{cases}$$

where $\vec{0} \in [0,1]^\Omega$ is given by $\vec{0}(\omega) = 0$ for all $\omega \in \Omega$, and the *convex closure* $\updownarrow X$ of a set X is defined

$$\updownarrow X := \{\, \textstyle\bigoplus_{i \in I} p_i \cdot o_i \mid p \in \mathcal{D}(I) \text{ and } o : I \to X \,\}\ .$$

We extend our earlier results-gathering definitions so that $\mathbb{V}_\updownarrow(s) := \updownarrow\mathbb{V}(s)$ and $\widehat{\mathbb{V}}_\updownarrow(s) := \updownarrow\widehat{\mathbb{V}}(s)$. Note that in case $\Omega := \{\omega\}$ we have $\widehat{\mathbb{V}}^\Omega_\updownarrow = \widehat{\mathbb{V}}_\updownarrow$, and the convex-closing preorders based on $\mathbb{V}_\updownarrow, \widehat{\mathbb{V}}_\updownarrow$ coincide with the simpler ones based on $\mathbb{V}, \widehat{\mathbb{V}}$. Thus convex closure matters only for proper vectors, as explained in the remark following Def. 6.

In [9] the results-gathering function $\widehat{\mathbb{V}}^\Omega_\updownarrow$ with $\Omega = \{\omega_1, \omega_2, \cdots\}$ was called simply $\mathbb{W}$ (because action-based/convex/vector-based testing was assumed there throughout, making the $\widehat{\cdot}^\Omega_\updownarrow$-indicators superfluous); and it was defined in terms of a formalisation of the notion of a resolution. The inductive definition above yields the same results.

Definition 6 For any pCSP process P and Ω-test T, let

$$\widehat{\mathcal{A}}^\Omega_\updownarrow(T,P) \;:=\; \widehat{\mathbb{V}}^\Omega_\updownarrow[\![T \mid_{\mathsf{Act}} P]\!]\ .$$

The *vector-based may-* and *must* preorders are given by

$$\begin{array}{lll} P \mathrel{\widehat{\sqsubseteq}}^\Omega_{\text{pmay}} Q & \text{iff} & \forall\,\Omega\text{-tests } T\text{: } \widehat{\mathcal{A}}^\Omega_\updownarrow(T,P) \leq_{\text{Ho}} \widehat{\mathcal{A}}^\Omega_\updownarrow(T,Q) \\ P \mathrel{\widehat{\sqsubseteq}}^\Omega_{\text{pmust}} Q & \text{iff} & \forall\,\Omega\text{-tests } T\text{: } \widehat{\mathcal{A}}^\Omega_\updownarrow(T,P) \leq_{\text{Sm}} \widehat{\mathcal{A}}^\Omega_\updownarrow(T,Q) \end{array}$$

where $\leq_{\text{Ho}}$ and $\leq_{\text{Sm}}$ are the Hoare- and Smyth preorders on $\mathcal{P}^+[0,1]^\Omega$ generated from $\leq$ index-wise on $[0,1]^\Omega$ itself.

Remark: For proper vector-based testing, convex closure matters, as it allows internal choice to simulate probabilistic choice [13]. Consider the following two processes

$$P := a \sqcap b \sqcap (a \,{}_{\frac{1}{2}}\!\oplus b) \quad \text{and} \quad Q := a \sqcap b.$$

It is obvious that $P \sqsubseteq_S Q$, and from Thm. 1 it therefore follows that $P \sqsubseteq_{\text{pmay}} Q$. However if $\Omega = \{\omega_1, \omega_2\}$ and we remove the convex closure in the definition of $\widehat{\mathbb{V}}^\Omega_\updownarrow$, then with the test $T := a.w_1 \,\Box\, b.w_2$ we would have

$$\begin{aligned} \widehat{\mathcal{A}}^\Omega(T,P) &= \{(1,0),(0,1),(0.5,0.5)\} \\ \widehat{\mathcal{A}}^\Omega(T,Q) &= \{(1,0),(0,1)\} \end{aligned}$$

and so $\widehat{\mathcal{A}}^\Omega(T,P) \not\leq_{\text{Ho}} \widehat{\mathcal{A}}^\Omega(T,Q)$. However, their convex closures $\widehat{\mathcal{A}}^\Omega_\updownarrow(T,P)$ and $\widehat{\mathcal{A}}^\Omega_\updownarrow(T,Q)$ *are* related under the Hoare preorder. □

The testing preorders of [31] are obtained by taking Ω to be a countably infinite set, whereas the preorders $\widehat{\sqsubseteq}_{\text{pmay}}$ and $\widehat{\sqsubseteq}_{\text{pmust}}$ of Sec. 5 were obtained by taking Ω to be the singleton set $\{\omega\}$. In [9] we established that for our finite pCSP processes the two coincide:

Theorem 3 [9].

1. $P \mathrel{\widehat{\sqsubseteq}}^\Omega_{\text{pmay}} Q$ iff $P \mathrel{\widehat{\sqsubseteq}}_{\text{pmay}} Q$
2. $P \mathrel{\widehat{\sqsubseteq}}^\Omega_{\text{pmust}} Q$ iff $P \mathrel{\widehat{\sqsubseteq}}_{\text{pmust}} Q$. □

Thus, with the *if*-direction of Thm. 3, for Thm. 2 it will suffice to show that $P \mathrel{\widehat{\sqsubseteq}}^\Omega_{\text{pmay}} Q$ implies $P \sqsubseteq_S Q$ and $P \mathrel{\widehat{\sqsubseteq}}^\Omega_{\text{pmust}} Q$ implies $P \sqsubseteq_{FS} Q$. The crucial characteristics of $\widehat{\mathcal{A}}^\Omega_\updownarrow$ needed for that implication are summarised in this lemma (proof omitted):

Lemma 1 Let P be a pCSP process, and T, T_i be tests.

1. $o \in \widehat{\mathcal{A}}^{\Omega}_{\updownarrow}(\omega, P)$ iff $o = \vec{\omega}$.
2. Suppose the action ω does not occur in the test T. Then $o \in \widehat{\mathcal{A}}^{\Omega}_{\updownarrow}(\omega \,\Box\, a.T, P)$ with $o(\omega) = 0$ iff there is a $\Delta \in \mathcal{D}(\mathsf{sCSP})$ with $[\![P]\!] \stackrel{\hat{a}}{\Longrightarrow} \Delta$ and $o \in \widehat{\mathcal{A}}^{\Omega}_{\updownarrow}(T, \Delta)$.
3. $\vec{0} \in \widehat{\mathcal{A}}^{\Omega}_{\updownarrow}(\Box_{a \in X}\, a.\omega, P)$ iff $\exists \Delta : [\![P]\!] \stackrel{\hat{\tau}}{\Longrightarrow} \Delta \stackrel{X}{\not\longrightarrow}$.
4. $o \in \widehat{\mathcal{A}}^{\Omega}_{\updownarrow}(\bigoplus_{i \in I} p_i{\cdot}T_i, P)$ iff $o = \sum_{i \in I} p_i o_i$ for certain $o_i \in \widehat{\mathcal{A}}^{\Omega}_{\updownarrow}(T_i, P)$.
5. $o \in \widehat{\mathcal{A}}^{\Omega}_{\updownarrow}(\prod_{i \in I} T_i, P)$ iff for all $i \in I$ there are $p_i \in [0,1]$ and $\Delta_i \in \mathcal{D}(\mathsf{sCSP})$ such that $[\![P]\!] \stackrel{\hat{\tau}}{\Longrightarrow} \bigoplus_{i \in I} p_i{\cdot}\Delta_i$ and $o = \sum_{i \in I} p_i o_i$ for certain $o_i \in \widehat{\mathcal{A}}^{\Omega}_{\updownarrow}(T_i, \Delta_i)$.

Here $\vec{\omega} \in [0,1]^{\Omega}$ is given by $\vec{\omega}(\omega) = 1$ and $\vec{\omega}(\omega') = 0$ for $\omega' \neq \omega$. □

In writing $\widehat{\mathcal{A}}^{\Omega}_{\updownarrow}(T, \Delta)$ above we treat a distribution Δ as the pCSP expression $\bigoplus_{s \in \lceil \Delta \rceil} \Delta_s{\cdot}s$; and as usual we define $\widehat{\mathcal{A}}^{\Omega}_{\updownarrow}(T, \Delta) := \mathrm{Exp}_{\Delta} \widehat{\mathcal{A}}^{\Omega}_{\updownarrow}(T, \_)$.

7. Modal logic

Our next step towards Thm. 2 is to define a set $\mathcal{F}$ of modal formulae, inductively, as follows:

- $\langle a \rangle \varphi \in \mathcal{F}$ when $\varphi \in \mathcal{F}$ and $a \in \mathsf{Act}$,
- $\mathbf{ref}(X) \in \mathcal{F}$ when $X \subseteq \mathsf{Act}$,
- $\bigwedge_{i \in I} \varphi_i \in \mathcal{F}$ when $\varphi_i \in \mathcal{F}$ for all $i \in I$, with I finite
- and $\bigoplus_{i \in I} p_i{\cdot}\varphi_i \in \mathcal{F}$ when $p_i \in [0,1]$ and $\varphi_i \in \mathcal{F}$ for all $i \in I$, with I a finite index set, and $\sum_{i \in I} p_i = 1$.

We often write $\varphi_1 \;{}_p\!\oplus\; \varphi_2$ for $\bigoplus_{i \in \{1,2\}} p_i{\cdot}\varphi_i$ with $p = p_1$, and $\varphi_1 \wedge \varphi_2$ for $\bigwedge_{i \in \{1,2\}} \varphi_i$ and finally $\top$ for $\bigwedge_{i \in \emptyset} \varphi_i$.

The *satisfaction relation* $\models \subseteq \mathcal{D}(\mathsf{sCSP}) \times \mathcal{F}$ is given by:

- $\Delta \models \langle a \rangle \varphi$ iff there is a Δ' with $\Delta \stackrel{\hat{a}}{\Longrightarrow} \Delta'$ and $\Delta' \models \varphi$,
- $\Delta \models \mathbf{ref}(X)$ iff there is a Δ' with $\Delta \stackrel{\hat{\tau}}{\Longrightarrow} \Delta'$ and $\Delta' \stackrel{X}{\not\longrightarrow}$,
- $\Delta \models \bigwedge_{i \in I} \varphi_i$ iff $\Delta \models \varphi_i$ for all $i \in I$
- and $\Delta \models \bigoplus_{i \in I} p_i{\cdot}\varphi_i$ iff there are $\Delta_i \in \mathcal{D}(\mathsf{sCSP})$, for all $i \in I$, with $\Delta_i \models \varphi_i$, such that $\Delta \stackrel{\hat{\tau}}{\Longrightarrow} \bigoplus_{i \in I} p_i{\cdot}\Delta_i$.

Let $\mathcal{L}$ be the subclass of $\mathcal{F}$ obtained by skipping the $\mathbf{ref}(X)$ clause. We write $P \sqsubseteq^{\mathcal{L}} Q$ just when $[\![P]\!] \models \varphi$ implies $[\![Q]\!] \models \varphi$ for all $\varphi \in \mathcal{L}$, and $P \sqsubseteq^{\mathcal{F}} Q$ just when $[\![P]\!] \models \varphi$ is implied by $[\![Q]\!] \models \varphi$ for all $\varphi \in \mathcal{F}$. (Note the opposing directions.)

Definition 7 The *characteristic formula* φ_s or φ_Δ of a process $s \in \mathsf{sCSP}$ or $\Delta \in \mathcal{D}(\mathsf{sCSP})$ is defined inductively:

- $\varphi_s := \bigwedge_{s \xrightarrow{a} \Delta} \langle a \rangle \varphi_\Delta \wedge \mathbf{ref}(\{a \mid s \stackrel{a}{\not\longrightarrow}\})$ if $s \stackrel{\tau}{\not\longrightarrow}$,
- $\varphi_s := \bigwedge_{s \xrightarrow{a} \Delta} \langle a \rangle \varphi_\Delta \wedge \bigwedge_{s \xrightarrow{\tau} \Delta} \varphi_\Delta$ otherwise,
- $\varphi_\Delta := \bigoplus_{s \in \lceil \Delta \rceil} \Delta(s){\cdot}\varphi_s$.

Here the conjunctions $\bigwedge_{s \xrightarrow{a} \Delta}$ range over suitable pairs a, Δ, and $\bigwedge_{s \xrightarrow{\tau} \Delta}$ ranges over suitable Δ.

Write $\varphi \Rrightarrow \psi$ with $\varphi, \psi \in \mathcal{F}$ if for each distribution Δ one has $\Delta \models \varphi$ implies $\Delta \models \psi$. Then it is easy to see that $\varphi_{\overline{s}} \Lleftarrow\!\Rrightarrow \varphi_s$ and $\bigwedge_{i \in I} \varphi_i \Rrightarrow \varphi_i$ for any $i \in I$.

The following property can be established by an easy inductive proof.

Lemma 2 For any $\Delta \in \mathcal{D}(\mathsf{sCSP})$ we have $\Delta \models \varphi_\Delta$. □

It and the following lemma help to prove Thm. 4.

Lemma 3 For any processes $P, Q \in \mathsf{pCSP}$ we have that $[\![P]\!] \models \varphi_{[\![Q]\!]}$ implies $P \sqsubseteq_{FS} Q$.

Proof: Define $\mathcal{R}$ by $s \,\mathcal{R}\, \Theta$ iff $\Theta \models \varphi_s$. We first show that

$$\Theta \models \varphi_\Delta \;\text{ implies }\; \exists \Theta' : \Theta \stackrel{\hat{\tau}}{\Longrightarrow} \Theta' \wedge \Delta \,\overline{\mathcal{R}}\, \Theta'. \qquad (1)$$

Suppose $\Theta \models \varphi_\Delta$ with $\varphi_\Delta = \bigoplus_{i \in I} p_i{\cdot}\varphi_{s_i}$, so that we have $\Delta = \bigoplus_{i \in I} p_i{\cdot}\overline{s_i}$ and for all $i \in I$ there are $\Theta_i \in \mathcal{D}(\mathsf{sCSP})$ with $\Theta_i \models \varphi_{s_i}$ such that $\Theta \stackrel{\hat{\tau}}{\Longrightarrow} \Theta'$ with $\Theta' := \bigoplus_{i \in I} p_i{\cdot}\Theta_i$. Since $s_i \,\mathcal{R}\, \Theta_i$ for all $i \in I$ we have $\Delta \,\overline{\mathcal{R}}\, \Theta'$.

Now we show that $\mathcal{R}$ is a failure simulation.

- Suppose $s \,\mathcal{R}\, \Theta$ and $s \xrightarrow{\tau} \Delta$. Then $\varphi_s \Rrightarrow \varphi_\Delta$, so $\Theta \models \varphi_\Delta$. Now apply (1).
- Suppose $s \,\mathcal{R}\, \Theta$ and $s \xrightarrow{a} \Delta$ with $a \in A$. Then $\varphi_s \Rrightarrow \langle a \rangle \varphi_\Delta$, so $\Theta \models \langle a \rangle \varphi_\Delta$. Hence $\exists \Theta'$ with $\Theta \stackrel{\hat{a}}{\Longrightarrow} \Theta'$ and $\Theta' \models \varphi_\Delta$. Now apply (1).
- Suppose $s \,\mathcal{R}\, \Theta$ and $s \stackrel{X}{\not\longrightarrow}$ with $X \subseteq A$. Then $\varphi_s \Rrightarrow \mathbf{ref}(X)$, so $\Theta \models \mathbf{ref}(X)$. Hence $\exists \Theta'$ with $\Theta \stackrel{\hat{\tau}}{\Longrightarrow} \Theta'$ and $\Theta' \stackrel{X}{\not\longrightarrow}$.

Thus we have $\Theta \models \varphi_s$ implies $s \,\triangleright_{FS}\, \Theta$. Using (1) with $[\![P]\!] \models \varphi_{[\![Q]\!]}$ gives $P \sqsubseteq_{FS} Q$ via Def. 5. □

Theorem 4

1. If $P \sqsubseteq^{\mathcal{L}} Q$ then $P \sqsubseteq_S Q$.
2. If $P \sqsubseteq^{\mathcal{F}} Q$ then $P \sqsubseteq_{FS} Q$.

Proof: Suppose $P \sqsubseteq^{\mathcal{F}} Q$. By Lem. 2 we have $[\![Q]\!] \models \varphi_{[\![Q]\!]}$ and hence $[\![P]\!] \models \varphi_{[\![Q]\!]}$. Lem. 3 gives $P \sqsubseteq_{FS} Q$.

For the $\sqsubseteq^{\mathcal{L}}$ case, omit $\mathbf{ref}(X)$ from the definition of a characteristic formula and begin with $[\![P]\!] \models \varphi_{[\![P]\!]}$. The counterpart of Lem. 3 now says that $[\![Q]\!] \models \varphi_{[\![P]\!]}$ implies $P \sqsubseteq_S Q$. □

8. Characteristic tests

Our final step towards Thm. 2 is taken in this section, where we show that every modal formula φ can be characterised by a vector-based test T_φ such that any pCSP process satisfies φ just when it passes the test T_φ.

Lemma 4 For every $\varphi \in \mathcal{F}$ there exists a pair (T_φ, v_φ) with T_φ an Ω-test and $v_\varphi \in [0,1]^\Omega$, such that

$$\Delta \models \varphi \quad \Leftrightarrow \quad \exists o \in \widehat{\mathcal{A}}^\Omega_\updownarrow(T_\varphi, \Delta) : \ o \le v_\varphi \tag{2}$$

for all $\Delta \in \mathcal{D}(\mathsf{sCSP})$, and in case $\varphi \in \mathcal{L}$ we also have

$$\Delta \models \varphi \quad \Leftrightarrow \quad \exists o \in \widehat{\mathcal{A}}^\Omega_\updownarrow(T_\varphi, \Delta) : \ o \ge v_\varphi \,. \tag{3}$$

T_φ is called a *characteristic test* of φ and v_φ its *target value*.

Proof: First of all note that if a pair (T_φ, v_φ) satisfies the requirements above, then any pair obtained from (T_φ, v_φ) by bijectively renaming the elements of Ω also satisfies these requirements. Hence a characteristic test can always be chosen in such a way that there is a success action $\omega \in \Omega$ that does not occur in (the finite) T_φ. Moreover, any countable collection of characteristic tests can be assumed to be Ω-*disjoint*, meaning that no $\omega \in \Omega$ occurs in two different elements of the collection.

The required characteristic tests and target values are obtained as follows.

- Let $\varphi = \top$. Take $T_\varphi := \omega$ for some $\omega \in \Omega$, and $v_\varphi := \vec{\omega}$.
- Let $\varphi = \langle a \rangle \psi$. By induction, ψ has a characteristic test T_ψ with target value v_ψ. Take $T_\varphi := \omega \,\Box\, a.T_\psi$ where $\omega \in \Omega$ does not occur in T_ψ, and $v_\varphi := v_\psi$.
- Let $\varphi = \mathbf{ref}(X)$ with $X \subseteq \mathsf{Act}$. Take $T_\varphi := \Box_{a \in X}\, a.\omega$ for some $\omega \in \Omega$, and $v_\varphi = \vec{0}$.
- Let $\varphi = \bigwedge_{i \in I} \varphi_i$ with I a finite and non-empty index set. Choose a Ω-disjoint family $(T_i, v_i)_{i \in I}$ of characteristic tests T_i with target values v_i for each φ_i. Furthermore, let $p_i \in (0,1]$ for $i \in I$ be chosen arbitrarily such that $\sum_{i \in I} p_i = 1$. Take $T_\varphi := \bigoplus_{i \in I} p_i{\cdot}T_i$ and $v_\varphi := \sum_{i \in I} p_i v_i$.
- Let $\varphi = \bigoplus_{i \in I} p_i{\cdot}\varphi_i$. Choose a Ω-disjoint family $(T_i, v_i)_{i \in I}$ of characteristic tests T_i with target values v_i for each φ_i, such that there are distinct success actions ω_i for $i \in I$ that do not occur in any of those tests. Let $T'_i := T_i \, {}_{\frac{1}{2}}\!\oplus\, \omega_i$ and $v'_i := \frac{1}{2} v_i + \frac{1}{2}\vec{\omega}_i$. Note that for all $i \in I$ also T'_i is a characteristic test of φ_i with target value v'_i. Take $T_\varphi := \bigsqcap_{i \in I} T'_i$ and $v_\varphi := \sum_{i \in I} p_i v'_i$.

Note that $v_\varphi(\omega) = 0$ whenever $\omega \in \Omega$ does not occur in T_φ. By induction on φ we now check (2) above.

- Let $\varphi = \top$. For all $\Delta \in \mathcal{D}(\mathsf{sCSP})$ we have $\Delta \models \varphi$ as well as $\exists o \in \widehat{\mathcal{A}}^\Omega_\updownarrow(T_\varphi, \Delta) : \ o \le v_\varphi$, using Lem. 1(1).
- Let $\varphi = \langle a \rangle \psi$ with $a \in \mathsf{Act}$. Suppose $\Delta \models \varphi$. Then there is a Δ' with $\Delta \stackrel{\hat{a}}{\Longrightarrow} \Delta'$ and $\Delta' \models \psi$. By induction, $\exists o \in \widehat{\mathcal{A}}^\Omega_\updownarrow(T_\psi, \Delta') : \ o \le v_\psi$. By Lem. 1(2), $o \in \widehat{\mathcal{A}}^\Omega_\updownarrow(T_\varphi, \Delta)$.

  Now suppose $\exists o \in \widehat{\mathcal{A}}^\Omega_\updownarrow(T_\varphi, \Delta) : \ o \le v_\varphi$. This implies $o(\omega) = 0$, so by Lem. 1(2) there is a Δ' with $\Delta \stackrel{\hat{a}}{\Longrightarrow} \Delta'$ and $o \in \widehat{\mathcal{A}}^\Omega_\updownarrow(T_\psi, \Delta')$. By induction, $\Delta' \models \psi$, so $\Delta \models \varphi$.
- Let $\varphi = \mathbf{ref}(X)$ with $X \subseteq \mathsf{Act}$. Suppose $\Delta \models \varphi$. Then there is a Δ' with $\Delta \stackrel{\hat{\tau}}{\Longrightarrow} \Delta'$ and $\Delta' \stackrel{X}{\not\longrightarrow}$. By Lem. 1(3), $\vec{0} \in \widehat{\mathcal{A}}^\Omega_\updownarrow(T_\varphi, \Delta)$.

  Now suppose $\exists o \in \widehat{\mathcal{A}}^\Omega_\updownarrow(T_\varphi, \Delta) : \ o \le v_\varphi$. This implies $o = \vec{0}$, so by Lem. 1(3) there is a Δ' with $\Delta \stackrel{\hat{\tau}}{\Longrightarrow} \Delta'$ and $\Delta' \stackrel{X}{\not\longrightarrow}$. Hence $\Delta \models \varphi$.
- Let $\varphi = \bigwedge_{i \in I} \varphi_i$ with I a finite and non-empty index set. Suppose $\Delta \models \varphi$. Then $\Delta \models \varphi_i$ for all $i \in I$, and hence, by induction, $\exists o_i \in \widehat{\mathcal{A}}^\Omega_\updownarrow(T_i, \Delta) : \ o_i \le v_i$. Thus $o := \sum_{i \in I} p_i o_i \in \widehat{\mathcal{A}}^\Omega_\updownarrow(T_\varphi, \Delta)$ by Lem. 1(4), and $o \le v_\varphi$.

  Now suppose $\exists o \in \widehat{\mathcal{A}}^\Omega_\updownarrow(T_\varphi, \Delta) : \ o \le v_\varphi$. Then, using Lem. 1(4), $o = \sum_{i \in I} p_i o_i$ for certain $o_i \in \widehat{\mathcal{A}}^\Omega_\updownarrow(T_i, \Delta)$. One has $o_i \le v_i$ for all $i \in I$, for if $o_i(\omega) > v_i(\omega)$ for some $i \in I$ and $\omega \in \Omega$, then ω must occur in T_i and hence cannot occur in T_j for $j \ne i$. This implies $v_j(\omega) = 0$ for all $j \ne i$ and thus $o(\omega) > v_\varphi(\omega)$, in contradiction with the assumption. By induction, $\Delta \models \varphi_i$ for all $i \in I$, and hence $\Delta \models \varphi$.
- Let $\varphi = \bigoplus_{i \in I} p_i{\cdot}\varphi_i$. Suppose $\Delta \models \varphi$. Then for all $i \in I$ there are $\Delta_i \in \mathcal{D}(\mathsf{sCSP})$ with $\Delta_i \models \varphi_i$ such that $\Delta \stackrel{\hat{\tau}}{\Longrightarrow} \bigoplus_{i \in I} p_i{\cdot}\Delta_i$. By induction, there are $o_i \in \widehat{\mathcal{A}}^\Omega_\updownarrow(\Delta_i, T'_i)$ with $o_i \le v'_i$. By Lem. 1(5), $o := \sum_{i \in I} p_i o_i \in \widehat{\mathcal{A}}^\Omega_\updownarrow(T_\varphi, \Delta)$, and $o \le v_\varphi$.

  Now suppose $\exists o \in \widehat{\mathcal{A}}^\Omega_\updownarrow(T_\varphi, \Delta) : \ o \le v_\varphi$. Then, by Lem. 1(5), there are $q \in \mathcal{D}(I)$ and Δ_i, for $i \in I$, such that $\Delta \stackrel{\hat{\tau}}{\Longrightarrow} \bigoplus_{i \in I} q_i{\cdot}\Delta_i$ and $o = \sum_{i \in I} q_i o_i$ for some $o_i \in \widehat{\mathcal{A}}^\Omega_\updownarrow(\Delta_i, T'_i)$. Now $\forall i : o_i(\omega_i) = v'_i(\omega_i) = \frac{1}{2}$, so $\frac{1}{2} q_i = q_i o_i(\omega_i) = o(\omega_i) \le v_\varphi(\omega_i) = p_i v'_i(\omega_i) = \frac{1}{2} p_i$. As $\sum_{i \in I} q_i = \sum_{i \in I} p_i = 1$, it must be that $q_i = p_i$ for all $i \in I$. Exactly as in the previous case one obtains $o_i \le v'_i$ all $i \in I$. By induction, $\Delta_i \models \varphi_i$ for all $i \in I$, and hence $\Delta \models \varphi$.

In case $\varphi \in \mathcal{L}$, a straightforward induction yields that $|v_\varphi| = 1$ and for all $\Delta \in \mathcal{D}(\mathsf{pCSP})$ and $o \in \widehat{\mathcal{A}}^\Omega_\updownarrow(T_\varphi, \Delta)$ we have $|o| = 1$. Here $|o|$ denotes $\sum_{\omega \in \Omega} o(\omega)$. Therefore, $o \le v$ iff $o \ge v$ iff $o = v$, yielding (3). □

Theorem 5

1. If $P \mathrel{\hat{\sqsubseteq}^\Omega_{\text{pmay}}} Q$ then $[\![P]\!] \sqsubseteq^{\mathcal{L}} [\![Q]\!]$.
2. If $P \mathrel{\hat{\sqsubseteq}^\Omega_{\text{pmust}}} Q$ then $[\![P]\!] \sqsubseteq^{\mathcal{F}} [\![Q]\!]$.

Proof: Suppose $P \mathrel{\hat{\sqsubseteq}^\Omega_{\text{pmust}}} Q$ and $[\![Q]\!] \models \varphi$ for some $\varphi \in \mathcal{F}$. Let T_φ be a characteristic test of φ with target value v_φ.

$$
\begin{array}{lrcl}
(\mathbf{P1}) & P\,{}_p\!\oplus P & = & P \\
(\mathbf{P2}) & P\,{}_p\!\oplus Q & = & Q\,{}_{1-p}\!\oplus P \\
(\mathbf{P3}) & (P\,{}_p\!\oplus Q)\,{}_q\!\oplus R & = & P\,{}_{p\cdot q}\!\oplus (Q\,{}_{\frac{(1-p)\cdot q}{1-p\cdot q}}\!\oplus R) \\
(\mathbf{I1}) & P \sqcap P & = & P \\
(\mathbf{I2}) & P \sqcap Q & = & Q \sqcap P \\
(\mathbf{I3}) & (P \sqcap Q) \sqcap R & = & P \sqcap (Q \sqcap R) \\
(\mathbf{E1}) & P \Box \mathbf{0} & = & P \\
(\mathbf{E2}) & P \Box Q & = & Q \Box P \\
(\mathbf{E3}) & (P \Box Q) \Box R & = & P \Box (Q \Box R) \\
(\mathbf{EI}) & a.P \Box a.Q & = & a.P \sqcap a.Q \\
(\mathbf{D1}) & P \Box (Q\,{}_p\!\oplus R) & = & (P \Box Q)\,{}_p\!\oplus (P \Box R) \\
(\mathbf{D2}) & a.P \Box (Q \sqcap R) & = & (a.P \Box Q) \sqcap (a.P \Box R) \\
(\mathbf{D3}) & P \Box Q & = & (P_1 \Box Q) \sqcap (P_2 \Box Q) \\
 & & & \sqcap (P \Box Q_1) \sqcap (P \Box Q_2), \\
 & & {\scriptstyle\text{provided}} & P = P_1 \sqcap P_2,\ Q = Q_1 \sqcap Q_2
\end{array}
$$

Figure 2. Common equations

Then Lem. 4 yields $\exists o \in \widehat{\mathcal{A}}^{\Omega}_{\updownarrow}(T_\varphi, [\![Q]\!]).\ o \leq v_\varphi$, and hence, given that $P \mathrel{\widehat{\sqsubseteq}^{\Omega}_{\text{pmust}}} Q$ and $\widehat{\mathcal{A}}^{\Omega}_{\updownarrow}(T_\varphi, [\![R]\!]) = \widehat{\mathcal{A}}^{\Omega}_{\updownarrow}(T_\varphi, R)$ for any $R \in$ pCSP, we have $\exists o' \in \widehat{\mathcal{A}}^{\Omega}_{\updownarrow}(T_\varphi, [\![P]\!]) :\ o' \leq v_\varphi$. Thus $[\![P]\!] \models \varphi$.

The may-case goes likewise. □

Combining Thms. 3-5 we obtain Thm. 2, the goal we set ourselves in Sec. 5. Thus, with Thm. 1 and Prop. 2, we have shown that the may preorder coincides with simulation and that the must preorder coincides with failure simulation. □

9. Equational theories

In order to focus on the essentials we now consider just those processes that do not use the parallel operator $|_A$; we call the resulting sub-language nCSP. For a discussion of the axiomatisation for terms involving $|_A$ and the other parallel operators commonly used in CSP see Sec. 11.

Let us write $P =_E Q$ to denote that $P = Q$ can be derived using the equations given in Fig. 2. Given the way we defined the syntax of pCSP, axiom $(\mathbf{D1})$ is merely a case of abbreviation-expansion. Many of the standard equations for CSP [15] are missing; they are not sound for $\simeq_{FS}$, as shown in Sec. 4 of [8]. Typical examples include:

$$a.(P \sqcap Q) = a.P \sqcap a.Q$$
$$P = P \Box P$$
$$P \Box (Q \sqcap R) = (P \Box Q) \sqcap (P \Box R)$$
$$P \sqcap (Q \Box R) = (P \sqcap Q) \Box (P \sqcap R)$$

Proposition 3 Suppose $P =_E Q$. Then $P \simeq_{FS} Q$.

Proof: Because of Prop. 1 it is sufficient to exhibit witness failure-simulations for axioms in Fig. 2. □

May:

$$
\begin{array}{lrl}
(\mathbf{May0}) & a.P \Box b.Q & = a.P \sqcap b.Q \\
(\mathbf{May1}) & P & \sqsubseteq P \sqcap Q \\
(\mathbf{May2}) & \mathbf{0} & \sqsubseteq P \\
(\mathbf{May3}) & a.(P\,{}_p\!\oplus Q) & \sqsubseteq a.P\,{}_p\!\oplus a.Q
\end{array}
$$

Must:

$$
\begin{array}{lrl}
(\mathbf{Must1}) & P \sqcap Q & \sqsubseteq Q \\
(\mathbf{Must2}) & R \sqcap \bigsqcap_{i\in I} P_i & \sqsubseteq \Box_{i\in I}\, a_i.Q_i, \\
 & {\scriptstyle\text{provided}} & P_i = \bigoplus_{j\in J_i} p_j(a_i.Q_{ij} \Box P_{ij}) \\
 & & Q_i = \bigoplus_{j\in J_i} p_j Q_{ij} \\
 & & \mathit{inits}(R) \subseteq \{a_i\}_{i\in I}
\end{array}
$$

Figure 3. Inequations

Note that this result also means $P =_E Q$ implies $P \simeq_S Q$.

Despite the weakness of this equational theory, it does allow us to reduce terms to a form in which the external choice operator is applied to prefix terms only.

Definition 8 [Normal forms] The set of *normal forms* N is given by the following grammar:

$$N ::= N_1\,{}_p\!\oplus N_2 \ \mid\ N_1 \sqcap N_2 \ \mid\ \Box_{i\in I} a_i.N_i$$

Proposition 4 For every $P \in$ nCSP there is a normal form N such that $P =_E N$.

Proof: A fairly straightforward induction, heavily relying on $(\mathbf{D1})$–$(\mathbf{D3})$. □

10. Inequational theories

In order to characterise the simulation preorders, and the associated testing preorders, we introduce *inequations*. We write $P \sqsubseteq_{E_{\text{may}}} Q$ when $P \sqsubseteq Q$ is derivable from the inequational theory obtained by adding the four *may* inequations in Fig. 3 to the equations in Fig. 2. The first three additions, $(\mathbf{May0})$–$(\mathbf{May2})$, are used in the standard testing theory of CSP [15, 6, 14]. For the *must* case, in addition to the standard inequation $(\mathbf{Must1})$, we require an inequational schema, $(\mathbf{Must2})$; this uses the notation *inits*(P) to denote the (finite) set of initial visible actions of P. Formally,

$$
\begin{array}{rcl}
\mathit{inits}(0) & = & \emptyset \\
\mathit{inits}(a.P) & = & \{a\} \\
\mathit{inits}(P\,{}_p\!\oplus Q) & = & \mathit{inits}(P) \cup \mathit{inits}(Q) \\
\mathit{inits}(P \Box Q) & = & \mathit{inits}(P) \cup \mathit{inits}(Q) \\
\mathit{inits}(P \sqcap Q) & = & \mathit{inits}(P) \cup \mathit{inits}(Q)
\end{array}
$$

The side conditions of $(\mathbf{Must2})$ entail that $[\![P_i]\!] \xrightarrow{a_i} [\![Q_i]\!]$ and that there exists some Δ such that $R \stackrel{\hat{\tau}}{\Longrightarrow} \Delta \stackrel{X}{\not\longrightarrow}$ with $X = \mathsf{Act}\backslash\{a_i\}_{i\in I}$. Note that $(\mathbf{Must2})$ can be used, together with $(\mathbf{I1})$, to derive the dual of $(\mathbf{May3})$, that is $a.P\ {}_p\!\oplus\ a.Q \sqsubseteq a.(P\ {}_p\!\oplus\ Q)$. We write $P \sqsubseteq_{E_{\text{must}}} Q$ when $P \sqsubseteq Q$ is derivable from the resulting inequational theory.

An important inequation that follows from $(\mathbf{May1})$ and $(\mathbf{P1})$ is

$$P\ {}_p\!\oplus\ Q\ \sqsubseteq_{E_{\text{may}}}\ P \sqcap Q$$

saying that any probabilistic choice can be simulated by an internal choice. Likewise, we have

$$P \sqcap Q\ \sqsubseteq_{E_{\text{must}}}\ P\ {}_p\!\oplus\ Q\ .$$

Theorem 6 For $P,\ Q$ in nCSP, it holds that

(i) $P \sqsubseteq_S Q$ if and only if $P \sqsubseteq_{E_{\text{may}}} Q$

(ii) $P \sqsubseteq_{FS} Q$ if and only if $P \sqsubseteq_{E_{\text{must}}} Q$

Proof: Omitted due to lack of space. □

11. Conclusions and related work

In this paper we continued our previous work [8, 9] in our quest for a testing theory for processes which exhibit both nondeterministic and probabilistic behaviour. We have studied three different aspects of may- and must testing preorders for finite processes: (i) we have shown that the may preorder can be characterised as a co-inductive simulation relation, and the must preorder as a failure simulation relation; (ii) we have given a characterisation of both preorders in a finitary modal logic; and (iii) we have also provided complete axiomatisations for both preorders over a probabilistic version of recursion-free CSP. Although we omitted our parallel operator $|_A$ from the axiomatisations, it and similar CSP and CCS-like parallel operators can be handled using standard techniques, in the must case at the expense of introducing auxiliary operators. In future work we hope to extend these results to recursive processes.

We believe these results, in each of the three areas, to be novel, although a number of partial results along similar lines exist in the literature. These are detailed below.

Related work: Early additions of probability to CSP include work by Lowe [23], Seidel [33] and Morgan et al. [27]; but all of them were forced to make compromises of some kind in order to address the potentially complicated interactions between the three forms of choice. The last [27] for example applied the Jones/Plotkin probabilistic powerdomain [16] directly to the failures model of CSP [2], the resulting compromise being that probability distributed outwards through all other operators; one controversial result of that was that internal choice was no longer idempotent, and that it was "clairvoyant" in the sense that it could adapt to probabilistic-choice outcomes that had not yet occurred. Mislove addressed this problem in [26] by presenting a denotational model in which internal choice distributed outwards through probabilistic choice. However, the distributivities of both [27] and [26] constitute identifications that cannot be justified by our testing approach; see [8].

In Jou and Smolka [20], as in [23, 33], probabilistic equivalences based on traces, failures and readies are defined. These equivalences are coarser than $\simeq_{\text{pmay}}$. For let

$$P := a.((b.d \,\Box\, c.e)\ {}_{\frac{1}{2}}\!\oplus\ (b.f \,\Box\, c.g))$$
$$Q := a.((b.d \,\Box\, c.g)\ {}_{\frac{1}{2}}\!\oplus\ (b.f \,\Box\, c.e)).$$

These two processes cannot be distinguished by the equivalences of [20, 23, 33]. However, we can tell them apart by the test

$$T := a.((b.d.\omega\ {}_{\frac{1}{2}}\!\oplus\ c.e.\omega) \sqcap (b.f.\omega\ {}_{\frac{1}{2}}\!\oplus\ c.g.\omega))$$

since $\mathcal{A}(T, P) = \{0, \frac{1}{2}, 1\}$ and $\mathcal{A}(T, Q) = \{\frac{1}{2}\}$, that is, $P \not\sqsubseteq_{\text{pmay}} Q$.

Probabilistic extensions of testing equivalences [6] have been widely studied. There are two different proposals on how to include probabilistic choice: (i) a test should be non-probabilistic, i.e., there is no occurrence of probabilistic choice in a test [22, 4, 17, 21, 11]; or (ii) a test can be probabilistic, i.e., probabilistic choice may occur in tests as well as processes [5, 35, 28, 18, 31, 19, 3]. This paper adopts the second approach.

Some work [22, 4, 5, 28] does not consider nondeterminism but deals exclusively with *fully probabilistic* processes. In this setting a process passes a test with a unique probability instead of a set of probabilities, and testing preorders in the style of [6] have been characterised in terms of *probabilistic traces* [5] and *probabilistic acceptance trees* [28]. Cazorla et al. [3] extended the results of [28] with nondeterminism, but suffered from the same problems as [27].

The work most closely related to ours is [18, 19]. In [18] Jonsson and Wang characterised may- and must-testing preorders in terms of "chains" of traces and failures, respectively, and in [19] they presented a "substantially improved" characterisation of their may-testing preorder using a notion of simulation which is weaker than $\sqsubseteq_S$ (cf. Def. 5). They only considered processes without τ-moves. In [8] we have shown that tests with internal moves can distinguish more processes than tests without internal moves, even when applied to processes that have no internal moves themselves.

Segala [31] defined two preorders called trace distribution precongruence ($\sqsubseteq_{TD}$) and failure distribution precongruence ($\sqsubseteq_{FD}$). He proved that the former coincides with an infinitary version of $\hat{\sqsubseteq}^{\Omega}_{\text{pmay}}$ (cf. Def. 6) and that the latter coincides with an infinitary version of $\hat{\sqsubseteq}^{\Omega}_{\text{pmust}}$. In [24] it has been shown that $\sqsubseteq_{TD}$ coincides with a notion of simulation akin to $\sqsubseteq_S$. Other probabilistic extensions of simulation occurring in the literature are reviewed in [8].

References

[1] E. Bandini & R. Segala (2001): *Axiomatizations for probabilistic bisimulation.* In Proc. *ICALP'01*, LNCS 2076, Springer, pp. 370–381.

[2] S.D. Brookes, C.A.R. Hoare & A.W. Roscoe (1984): *A theory of communicating sequential processes.* Journal of the *ACM* 31(3), pp. 560–599.

[3] D. Cazorla, F. Cuartero, V.V. Ruiz, F.L. Pelayo & J.J. Pardo (2003): *Algebraic theory of probabilistic and nondeterministic processes.* Journal of Logic and Algebraic Programming 55, pp. 57–103.

[4] I. Christoff (1990): *Testing equivalences and fully abstract models for probabilistic processes.* In Proc. *CONCUR'90*, LNCS 458, Springer, pp. 126–140.

[5] R. Cleaveland, Z. Dayar, S.A. Smolka & S. Yuen (1999): *Testing preorders for probabilistic processes.* Information and Computation 154(2), pp. 93–148.

[6] R. De Nicola & M. Hennessy (1984): *Testing equivalences for processes.* Theoretical Computer Science 34, pp. 83–133.

[7] Y. Deng & C. Palamidessi (2007): *Axiomatizations for probabilistic finite-state behaviors.* Theoretical Computer Science 373(1-2), pp. 92–114.

[8] Y. Deng, R.J. van Glabbeek, M. Hennessy, C.C. Morgan & C. Zhang (2007): *Remarks on testing probabilistic processes.* *ENTCS* 172, pp. 359–397.

[9] Y. Deng, R.J. van Glabbeek, C.C. Morgan & C. Zhang (2007): *Scalar outcomes suffice for finitary probabilistic testing.* In Proc. *ESOP'07*, LNCS 4421, Springer, pp. 363–368.

[10] R.J. van Glabbeek (1993): *The linear time – branching time spectrum II; the semantics of sequential systems with silent moves.* In Proc. *CONCUR'93*, LNCS 715, Springer, pp. 66–81.

[11] C. Gregorio-Rodríguez & M. Núñez (1999): *Denotational semantics for probabilistic refusal testing.* *ENTCS* 22, pp. 111–137.

[12] H. Hansson & B. Jonsson (1990): *A calculus for communicating systems with time and probabilities.* In Proc. *RTSS'90*, IEEE Computer Society Press, pp. 278–287.

[13] He Jifeng, K. Seidel & A.K. McIver (1997): *Probabilistic models for the guarded command language.* Science of Computer Programming 28, pp. 171–192.

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

[15] C.A.R. Hoare (1985): *Communicating Sequential Processes*. Prentice-Hall.

[16] C. Jones & G.D. Plotkin (1989): *A probabilistic powerdomain of evaluations.* In Proc. *LICS'89*, Computer Society Press, pp. 186–195.

[17] B. Jonsson, C. Ho-Stuart & Wang Yi (1994): *Testing and refinement for nondeterministic and probabilistic processes.* In Proc. *FTRTFT'94*, LNCS 863, Springer, pp. 418–430.

[18] B. Jonsson & Wang Yi (1995): *Compositional testing preorders for probabilistic processes.* In Proc. *LICS'95*, IEEE Computer Society Press, pp. 431–441.

[19] B. Jonsson & Wang Yi (2002): *Testing preorders for probabilistic processes can be characterized by simulations.* Theoretical Computer Science 282(1), pp. 33–51.

[20] C.-C. Jou & S.A. Smolka (1990): *Equivalences, congruences, and complete axiomatizations for probabilistic processes.* In Proc. *CONCUR '90*, LNCS 458, Springer, pp. 367–383.

[21] M.Z. Kwiatkowska & G. Norman (1998): *A testing equivalence for reactive probabilistic processes.* *ENTCS* 16(2), pp. 114–132.

[22] K.G. Larsen & A. Skou (1991): *Bisimulation through probabilistic testing.* Information and Computation 94(1), pp. 1–28.

[23] G. Lowe (1993): *Representing nondeterminism and probabilistic behaviour in reactive processes.* Technical Report TR-11-93, Computing laboratory, Oxford University.

[24] N. Lynch, R. Segala & F.W. Vaandrager (2003): *Compositionality for probabilistic automata.* In Proc. *CONCUR'03*, LNCS 2761, Springer, pp. 204–222.

[25] R. Milner (1989): *Communication and Concurrency*. Prentice-Hall.

[26] M.W. Mislove (2000): *Nondeterminism and probabilistic choice: Obeying the laws.* In Proc. *CONCUR'00*, LNCS 1877, Springer, pp. 350–364.

[27] C.C. Morgan, A.K. McIver, K. Seidel & J.W. Sanders (1996): *Refinement oriented probability for CSP.* Formal Aspects of Computing 8, pp. 617–647.

[28] M. Núñez (2003): *Algebraic theory of probabilistic processes.* Journal of Logic and Algebraic Programming 56, pp. 117–177.

[29] E.-R. Olderog & C.A.R. Hoare (1986): *Specification-oriented semantics for communicating processes.* Acta Informatica 23, pp. 9–66.

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

[31] R. Segala (1996): *Testing probabilistic automata.* In Proc. *CONCUR'96*, LNCS 1119, Springer, pp. 299–314.

[32] R. Segala & N.A. Lynch (1994): *Probabilistic simulations for probabilistic processes.* In Proc. *CONCUR'94*, LNCS 836, Springer, pp. 481–496.

[33] K. Seidel (1995): *Probabilistic communicating processes.* Theoretical Computer Science 152(2), pp. 219–249.

[34] R. Tix, K. Keimel & G.D. Plotkin (2005): *Semantic domains for combining probability and non-determinism.* *ENTCS* 129, pp. 1–104.

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