CHARACTERISING TESTING PREORDERS FOR FINITE PROBABILISTIC PROCESSES

YUXIN DENG[1], ROB VAN GLABBEEK[2], MATTHEW HENNESSY[3] & CARROLL MORGAN[4]

[1] Shanghai Jiao Tong University, China

[2] National ICT Australia, Australia

[3] University of Sussex, UK

[1,2,4] University of New South Wales, 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 finite CSP.

1. Introduction

A satisfactory semantic theory for processes which encompass both nondeterministic and probabilistic behaviour has been a long-standing research problem [13, 41, 28, 20, 38, 39, 36, 22, 32, 37, 14, 26, 31, 1, 23, 29, 3, 40, 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 [41]. 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.

1998 ACM Subject Classification: F.3.2, D.3.1.

Key words and phrases: Probabilistic processes, testing semantics, simulation, axiomatisation.

An extended abstract of this paper has appeared as [9].

Deng was supported by the National Natural Science Foundation of China (60703033). Morgan and Deng would like to acknowledge the support of the Australian Research Council (ARC) Grant DP034557. Hennessy would like to acknowledge the support of The Royal Society, UK.

LOGICAL METHODS IN COMPUTER SCIENCE

DOI:10.2168/LMCS-???

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 Section 2 we provide an interpretation of pCSP as a *probabilistic labelled transition system*, in which, following [38, 20], state-to-state 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 Section 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 alternative characterisations of these testing preorders. This problem was addressed previously by Segala in [37], but using testing preorders ($\widehat{\sqsubseteq}^{\Omega}_{\text{pmay}}$ and $\widehat{\sqsubseteq}^{\Omega}_{\text{pmust}}$) that differ in two ways from the ones in [6, 15, 41, 8] and the present paper. First of all, in [37] 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 approach, as opposed to the *state*-based approach used in this paper. Secondly, [37] 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 [37] 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 $\widehat{\sqsubseteq}_{\text{pmay}}$ and $\widehat{\sqsubseteq}_{\text{pmust}}$ that arise from action-based but scalar testing, and use a recent result [10] saying that for finite processes the preorders $\widehat{\sqsubseteq}^{\Omega}_{\text{pmay}}$ and $\widehat{\sqsubseteq}^{\Omega}_{\text{pmust}}$ coincide with $\widehat{\sqsubseteq}_{\text{pmay}}$ and $\widehat{\sqsubseteq}_{\text{pmust}}$. Here we show that on pCSP the preorders $\widehat{\sqsubseteq}_{\text{pmay}}$ and $\widehat{\sqsubseteq}_{\text{pmust}}$ also coincide with $\sqsubseteq_{\text{pmay}}$ and $\sqsubseteq_{\text{pmust}}$.[1]

Simulation preorders: In Section 4 we use the transitions $s \xrightarrow{\alpha} \Delta$ to define two co-inductive preorders, the *simulation* preorder $\sqsubseteq_S$ [36, 29, 8], and the novel *failure simulation* preorder $\sqsubseteq_{FS}$ over pCSP processes. The latter extends the failure simulation preorder of [11] 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 second preorder differs from the first one 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 here we use similar techniques in the proofs for $\sqsubseteq_{FS}$. 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 Section 7, uses finite conjunction $\bigwedge_{i\in I} \varphi_i$, the modality $\langle a\rangle\varphi$ from the Hennessy-Milner Logic [16], 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

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

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 [11]. 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:

$$\begin{array}{cccccccccccc} \underbrace{\sqsubseteq^{\mathcal{L}} \ \subseteq}_{\textit{Sec. 7}} & \underbrace{\sqsubseteq_S \ \overset{[8]}{\subseteq}}_{\textit{Sec. 4}} & \underbrace{\sqsubseteq_{\text{pmay}} \ \subseteq}_{\textit{Sec. 3}} & \underbrace{\hat{\sqsubseteq}_{\text{pmay}} \ \overset{[10]}{=}}_{\textit{Sec. 5}} & \underbrace{\hat{\sqsubseteq}^{\Omega}_{\text{pmay}} \ \subseteq}_{\textit{Sec. 6}} & \underbrace{\sqsubseteq^{\mathcal{L}}}_{\textit{Sec. 8}} \end{array}$$

$$\begin{array}{lllllllllll} \sqsubseteq^{\mathcal{L}} & \subseteq & \sqsubseteq_S & \overset{[8]}{\subseteq} & \sqsubseteq_{\text{pmay}} & \subseteq & \hat{\sqsubseteq}_{\text{pmay}} & \overset{[10]}{=} & \hat{\sqsubseteq}^{\Omega}_{\text{pmay}} & \subseteq & \sqsubseteq^{\mathcal{L}} \\ \sqsubseteq^{\mathcal{F}} & \subseteq & \sqsubseteq_{FS} & \subseteq & \sqsubseteq_{\text{pmust}} & \subseteq & \hat{\sqsubseteq}_{\text{pmust}} & \overset{[10]}{=} & \hat{\sqsubseteq}^{\Omega}_{\text{pmust}} & \subseteq & \sqsubseteq^{\mathcal{F}} \\ \multicolumn{2}{c}{\underbrace{\qquad\quad}_{\textit{Sec. 7}}} & \multicolumn{2}{c}{\underbrace{\qquad\quad}_{\textit{Sec. 4}}} & \multicolumn{2}{c}{\underbrace{\qquad\quad}_{\textit{Sec. 3}}} & \multicolumn{2}{c}{\underbrace{\qquad\quad}_{\textit{Sec. 5}}} & \multicolumn{2}{c}{\underbrace{\qquad\quad}_{\textit{Sec. 6}}} & \underbrace{\quad}_{\textit{Sec. 8}} \end{array}$$

In Section 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 Section 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 [37], 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 (Section 5) and recalling the result from [10] that the action-based scalar testing preorders imply the vector-based ones (Section 6).

(In)equations: It is well-known that may- and must testing for standard CSP can be captured equationally [6, 2, 15]. 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 Section 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

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

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 *visible* (or *external*) 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:

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

We write pCSP, ranged over by P, Q, for the set of process terms defined by this grammar, and sCSP, ranged over by s, t, for the subset comprising only the state-based process terms (the sub-sort S above).

The process $P\,{}_p\!\oplus\, Q$, 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 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 $a.\_$ for $a \in \mathsf{Act}$, *internal-* and *external choices* $\sqcap$ and $\Box$, and a *parallel composition* $|_A$ for $A \subseteq \mathsf{Act}$.

The process $P \sqcap Q$ will first do a so-called *internal action* $\tau \notin \mathsf{Act}$, 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 $s \Box t$ 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 $|_A$.

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

Although formally the operators $\Box$ and $|_A$ can only be applied to state-based processes, informally we use expressions of the form $P \Box Q$ and $P \,|_A\, Q$, where P and Q are *not* state-based, as syntactic sugar for expressions in the above syntax obtained by distributing $\Box$ and $|_A$ over ${}_p\!\oplus$. Thus for example $s \Box (t_1\, {}_p\!\oplus\, t_2)$ abbreviates the term $(s \Box t_1)\, {}_p\!\oplus\, (s \Box t_2)$.

The full language of CSP [2, 17, 34] 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 [34]. 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 $\Box_{i\in I} P_i$, for some finite index set I.

The above intuitions are formalised by an *operational semantics*[2] associating with each process term a graph-like structure representing its possible reactions to users' requests: we use a generalisation of labelled transition systems [30] 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 $\lceil\overline{s}\rceil = \{s\}$. If $p_i \geq 0$ and Δ_i is a distribution for each i in some finite index set I, and $\sum_{i\in I} p_i = 1$, then the

[2]Although the syntax of pCSP is similar to other probabilistic extensions of CSP [28, 32, 31], our semantics differs. For more detailed comparisons, see Section 12.

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

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

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

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

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

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

Figure 1: Operational semantics of pCSP

probability distribution $\sum_{i\in I} p_i \cdot \Delta_i \in \mathcal{D}(S)$ is given by

$$(\sum_{i\in I} p_i \cdot \Delta_i)(s) \;=\; \sum_{i\in I} p_i \cdot \Delta_i(s) \;;$$

we will sometimes write it as $p_1 \cdot \Delta_1 + \ldots + p_n \cdot \Delta_n$ when the index set I is $\{1, \ldots, n\}$.

For Δ a distribution over S and function $f: S \to X$ into a vector space X we sometimes write $\mathrm{Exp}_\Delta(f)$ for $\sum_{s\in S} \Delta(s)\cdot f(s)$, the *expected value* of f. Our primary use of this notation is with X being the vector space of reals or tuples of reals. 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 $\mathrm{Exp}_\Delta F := \{\, \mathrm{Exp}_\Delta(f) \mid f \mathrel{\overline{\in}} F \,\}$; here $f \mathrel{\overline{\in}} F$ means that $f: S \to X$ is a *choice function* for F, that is it satisfies the constraint that $f(s) \in F(s)$ for all $s \in S$.

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

Definition 2.1. A *probabilistic labelled transition system* (pLTS)[3] is a triple $\langle S, L, \to \rangle$, where

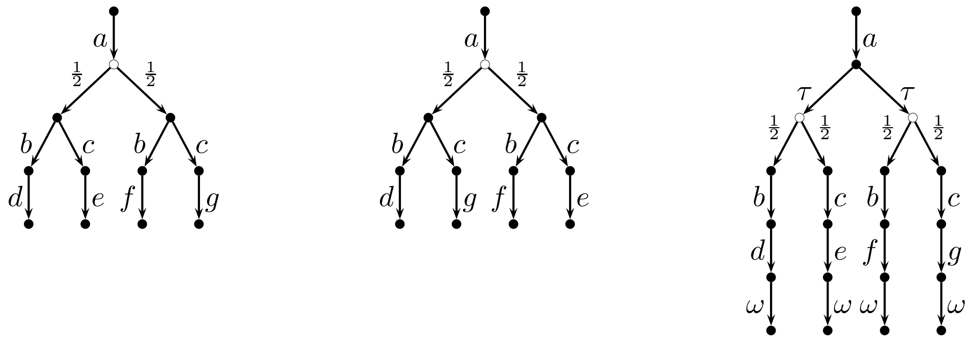
(i) S is a set of states,
(ii) L is a set of transition labels,
(iii) relation $\to$ is a subset of $S \times L \times \mathcal{D}(S)$.

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.

The operational semantics of pCSP is defined by a particular pLTS $\langle \mathsf{sCSP}, \mathsf{Act}_\tau, \to \rangle$, constructed by taking sCSP to be the set of states and $\mathsf{Act}_\tau := \mathsf{Act} \cup \{\tau\}$ the set of transition labels; we let a range over Act and α over Act_τ. We interpret pCSP processes P as distributions $[\![P]\!] \in \mathcal{D}(\mathsf{sCSP})$ via the function $[\![\_]\!] : \mathsf{pCSP} \to \mathcal{D}(\mathsf{sCSP})$ defined below:

$$\begin{aligned} [\![s]\!] &:= \overline{s} \text{ for } s \in \mathsf{sCSP} \\ [\![P\,{}_p\!\oplus Q]\!] &:= p \cdot [\![P]\!] + (1-p) \cdot [\![Q]\!] \;. \end{aligned}$$

[3] Essentially the same model has appeared in the literature under different names such as *NP-systems* [20], *probabilistic processes* [22], *simple probabilistic automata* [36], *probabilistic transition systems* [23] etc. Furthermore, there are strong structural similarities with *Markov Decision Processes* [35, 10].



(i) P (ii) Q (iii) T

Figure 2: Example processes P, Q and test T

Note that for each $P \in \mathsf{pCSP}$ the distribution $[\![P]\!]$ is finite, that is it has finite support. The definition of the relations $\xrightarrow{\alpha}$ is given in Figure 1. These rules are very similar to the standard ones used to interpret CSP as an LTS [34], but modified so that the result of an action is a distribution. The rules for external choice and parallel composition use an obvious notation for distributing an operator over a distribution; for example $\Delta \;\Box\; s$ represents the distribution given by

$$(\Delta \;\Box\; s)(t) = \begin{cases} \Delta(s') & \text{if } t = s' \;\Box\; s \\ 0 & \text{otherwise.} \end{cases}$$

We sometimes write $\tau.P$ for $P \sqcap P$, thus giving $\tau.P \xrightarrow{\tau} [\![P]\!]$.

We graphically depict the operational semantics of a pCSP expression P by drawing the part of the pLTS defined above that is reachable from $[\![P]\!]$ as a finite acyclic directed graph, often unwound into a tree. States are represented by nodes of the form $\bullet$ and distributions by nodes of the form $\circ$. For any state s and distribution Δ with $s \xrightarrow{\alpha} \Delta$ we draw an edge from s to Δ, labelled with α. For any distribution Δ and state s in $\lceil \Delta \rceil$, the support of Δ, we draw an edge from Δ to s, labelled with $\Delta(s)$.

Example 2.2. Consider the two processes

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

Their tree representations are depicted in Figure 2 (i) and (ii). To make these trees more compact we omit nodes $\circ$ when they represent trivial point distributions.

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; we write pCSP^ω for the set of all tests, and sCSP^ω for the subset of state-based process terms that may involve the action ω, 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 $T \,|_{\mathsf{Act}}\, P$ 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 $\mathbb{V} : \mathsf{sCSP}^\omega \to \mathcal{P}^+([0,1])$; it extends to type $\mathcal{D}(\mathsf{sCSP}^\omega) \to \mathcal{P}^+([0,1])$ via the convention $\mathbb{V}(\Delta) := \mathrm{Exp}_\Delta \mathbb{V}$.

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

In the first case above $s \xrightarrow{\omega}$ signifies that s is a success state. In the second case we mean that ω is not possible from s—hence s is not a success state—but that at least one "non-success" action $\alpha \in \mathsf{Act}_\tau$ is—and possibly several—and then the union is over all such α. This is done so that $\mathbb{V}$ accounts for success actions in processes generally; when applied to test outcomes, however, the only non-success action is τ. Note that $\mathbb{V}$ is well defined when applied to finite, loop-free processes, such as the ones of pCSP.

Definition 3.1. 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 $P \sqsubseteq_{\mathrm{pmay}} Q$, and the other on *must* testing, written $P \sqsubseteq_{\mathrm{pmust}} Q$.

Definition 3.2. The *may*- and *must* preorders are given by

$$\begin{array}{rll} P \sqsubseteq_{\mathrm{pmay}} Q & \text{iff} & \text{for all tests } T\text{: } \mathcal{A}(T,P) \leq_{\mathrm{Ho}} \mathcal{A}(T,Q) \\ P \sqsubseteq_{\mathrm{pmust}} Q & \text{iff} & \text{for all tests } T\text{: } \mathcal{A}(T,P) \leq_{\mathrm{Sm}} \mathcal{A}(T,Q) \end{array}$$

with $\leq_{\mathrm{Ho}}, \leq_{\mathrm{Sm}}$ the Hoare, Smyth preorders on $\mathcal{P}^+[0,1]$. These are defined as follows:

$$\begin{array}{rll} X \leq_{\mathrm{Ho}} Y & \text{iff} & \forall x \in X\colon \exists y \in Y\colon x \leq y \\ X \leq_{\mathrm{Sm}} Y & \text{iff} & \forall y \in Y\colon \exists x \in X\colon x \leq y \end{array}$$

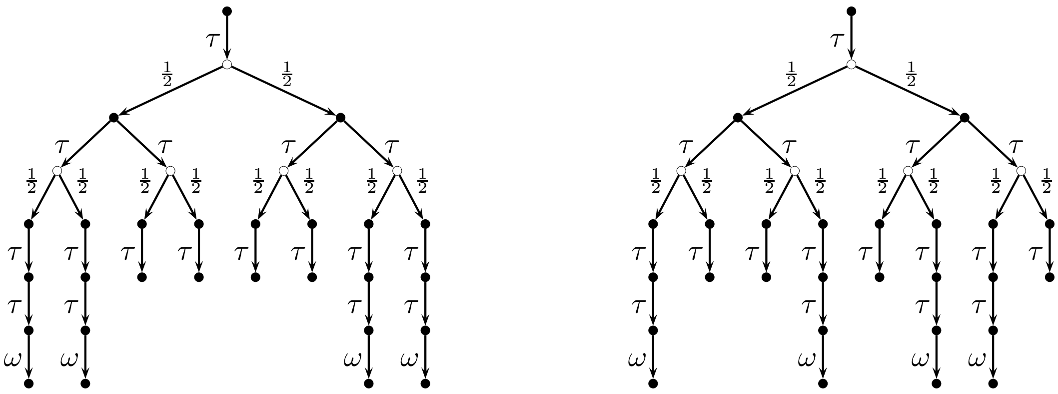
In other words, Q is a correct refinement of P in the probabilistic may-testing preorder if each outcome (in [0,1]) of applying a test to process P can be matched or increased by applying the same test to process Q. Likewise, Q is a correct refinement of P in the probabilistic must-testing preorder if each outcome of applying a test to Q matches or increases an outcome obtainable by applying the same test to P.

Example 3.3. Consider the test

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

which is graphically depicted in Figure 2 (iii). If we apply T to processes P and Q given in Example 2.2, we form the two processes described in Figure 3. It is then easy to calculate the testing outcomes:

$$\begin{array}{rcl} \mathcal{A}(T,P) & = & \frac{1}{2}\cdot\{1,0\} + \frac{1}{2}\cdot\{1,0\} \\ & = & \{0,\ \frac{1}{2},\ 1\} \\[1em] \mathcal{A}(T,Q) & = & \frac{1}{2}\cdot\{\frac{1}{2}\} + \frac{1}{2}\cdot\{\frac{1}{2}\} \\ & = & \{\frac{1}{2}\}. \end{array}$$



$(i)\ T\,|_{\mathsf{Act}}\,P$ $(ii)\ T\,|_{\mathsf{Act}}\,Q$

Figure 3: Testing P and Q with T.

We can see that P and Q can be distinguished by the test T since $\mathcal{A}(T,P) \not\leq_{\mathrm{Ho}} \mathcal{A}(T,Q)$ and $\mathcal{A}(T,Q) \not\leq_{\mathrm{Sm}} \mathcal{A}(T,P)$. In other words, we have $P \not\sqsubseteq_{\mathrm{pmay}} Q$ and $Q \not\sqsubseteq_{\mathrm{pmust}} P$ because of the witness test T.

In [8] we applied the testing framework described above to show that many standard laws of CSP are no longer valid in the probabilistic setting of pCSP, and to provide counterexamples for a few distributive laws involving probabilistic choice that may appear plausible at first sight. We also showed that $P \sqsubseteq_{\mathrm{pmust}} Q$ implies $Q \sqsubseteq_{\mathrm{pmay}} P$ for all pCSP processes P and Q, i.e. that must testing is more discriminating than may testing and that the preorders $\sqsubseteq_{\mathrm{pmay}}$ and $\sqsubseteq_{\mathrm{pmust}}$ are oriented in opposite directions.

4. Simulation and failure simulation

Let $\mathcal{R} \subseteq S \times \mathcal{D}(S)$ be a relation from states to distributions. As in [8], 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 a finite index set I and $p \in \mathcal{D}(I)$ such that

(i) $\Delta = \sum_{i\in I} p_i \cdot \overline{s_i}$,
(ii) For each $i \in I$ there is a distribution Φ_i s.t. $s_i \;\mathcal{R}\; \Phi_i$,
(iii) $\Theta = \sum_{i\in I} p_i \cdot \Phi_i$.

For functions, the lifting operation can be understood as a Kleisli construction on a probabilistic power domain [18], and was implicit in the work of Kozen [25]; in our more general setting of relations, it can equivalently be defined in terms of a distribution on $\mathcal{R}$, sometimes called *weight function* (see e.g. [21, 36]). An important point here is that in the decomposition (i) of Δ_1 into $\sum_{i\in I} p_i \cdot \overline{s_i}$, the states s_i are *not necessarily distinct*: that is, the decomposition is not in general unique. 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 for example we have $[\![(a \sqcap b)\,{}_{\frac{1}{2}}\!\oplus\,(a \sqcap c)]\!] \xrightarrow{\hat{\tau}} [\![a\,{}_{\frac{1}{2}}\!\oplus\,((a \sqcap b)\,{}_{\frac{1}{2}}\!\oplus\,c)]\!]$ because

(i) $[\![(a \sqcap b)\,{}_{\frac{1}{2}}\!\oplus\,(a \sqcap c)]\!] = \frac{1}{4}\cdot[\![(a \sqcap b)]\!] + \frac{1}{4}\cdot[\![(a \sqcap b)]\!] + \frac{1}{4}\cdot[\![(a \sqcap c)]\!] + \frac{1}{4}\cdot[\![(a \sqcap c)]\!]$,

(ii)
$$\begin{array}{lcl} [\![(a \sqcap b)]\!] & \xrightarrow{\tau} & [\![a]\!] \\ [\![(a \sqcap b)]\!] & \xrightarrow{\hat{\tau}} & [\![a \sqcap b]\!] \\ [\![(a \sqcap c)]\!] & \xrightarrow{\tau} & [\![a]\!] \\ [\![(a \sqcap c)]\!] & \xrightarrow{\tau} & [\![c]\!] \end{array}$$

(iii) and $[\![a \,_{\frac{1}{2}}\!\oplus ((a \sqcap b) \,_{\frac{1}{2}}\!\oplus c)]\!] = \frac{1}{4} \cdot [\![a]\!] + \frac{1}{4} \cdot [\![(a \sqcap b)]\!] + \frac{1}{4} \cdot [\![a]\!] + \frac{1}{4} \cdot [\![c]\!]$.

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 $a \neq \tau$ we let $\Delta_1 \stackrel{\hat{a}}{\Longrightarrow} \Delta_2$ denote $\Delta_1 \stackrel{\hat{\tau}}{\Longrightarrow}\xrightarrow{a}\stackrel{\hat{\tau}}{\Longrightarrow} \Delta_2$. Finally, we write $s \stackrel{X}{\not\longrightarrow}$ with $X \subseteq \mathsf{Act}$ when $\forall \alpha \in X \cup \{\tau\} : s \stackrel{\alpha}{\not\longrightarrow}$, and $\Delta \stackrel{X}{\not\longrightarrow}$ when $\forall s \in \lceil \Delta \rceil : s \stackrel{X}{\not\longrightarrow}$. The main properties of the lifted weak transition relations which are used throughout the paper are given in the following lemma.

Lemma 4.1. *Suppose $\sum_{i\in I} p_i = 1$ and $\Delta_i \stackrel{\hat{\alpha}}{\Longrightarrow} \Phi_i$ for each $i \in I$, with I a finite index set. Then*
$$\sum_{i\in I} p_i \cdot \Delta_i \stackrel{\hat{\alpha}}{\Longrightarrow} \sum_{i\in I} p_i \cdot \Phi_i \,.$$
Conversely, if $\sum_{i\in I} p_i \cdot \Delta_i \stackrel{\hat{\alpha}}{\Longrightarrow} \Phi$ then $\Phi = \sum_{i\in I} p_i \cdot \Phi_i$ for some Φ_i such that $\Delta_i \stackrel{\hat{\alpha}}{\Longrightarrow} \Phi_i$ for each $i \in I$.

Proof. The first claim occurs as Lemma 6.6 of [8]. The second follows by repeated application of Proposition 6.1(ii) of [8], taking $\mathcal{R}$ to be $\xrightarrow{\hat{\tau}}$ and $\xrightarrow{a}$ for $a \in \mathsf{Act}$. ☐

Definition 4.2. A relation $\mathcal{R} \subseteq \mathsf{sCSP} \times \mathcal{D}(\mathsf{sCSP})$ is said to be a *failure simulation* if for all $s, \Theta, \alpha, \Delta, X$ we have that

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

We write $s \lhd_{FS} \Theta$ to mean that there is some failure simulation $\mathcal{R}$ such that $s \,\mathcal{R}\, \Theta$. Similarly, we define *simulation*[4] and $s \lhd_S \Theta$ by dropping the second clause in Definition 4.2.[5]

Definition 4.3. The *simulation preorder* $\sqsubseteq_S$ and *failure simulation preorder* $\sqsubseteq_{FS}$ on pCSP are defined as follows:
$$\begin{array}{lll} P \sqsubseteq_S Q & \text{iff} & [\![Q]\!] \stackrel{\hat{\tau}}{\Longrightarrow} \Theta \text{ for some } \Theta \text{ with } [\![P]\!] \,\overline{\lhd_S}\, \Theta \\ P \sqsubseteq_{FS} Q & \text{iff} & [\![P]\!] \stackrel{\hat{\tau}}{\Longrightarrow} \Theta \text{ for some } \Theta \text{ with } [\![Q]\!] \,\overline{\lhd_{FS}}\, \Theta \,. \end{array}$$

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

Example 4.4. Compare the processes $P = a \,_{\frac{1}{2}}\!\oplus b$ and $P \sqcap P$. Note that $[\![P]\!]$ is the distribution $\frac{1}{2}\cdot \overline{a} + \frac{1}{2}\cdot \overline{b}$ whereas $[\![P \sqcap P]\!]$ is the point distribution $\overline{P \sqcap P}$. The relation $\mathcal{R}$ given by
$$(P \sqcap P) \;\mathcal{R}\; (\tfrac{1}{2}\cdot \overline{a} + \tfrac{1}{2}\cdot \overline{b}) \qquad a \;\mathcal{R}\; \overline{a} \qquad b \;\mathcal{R}\; \overline{b} \qquad \mathbf{0} \;\mathcal{R}\; \overline{\mathbf{0}}$$
is a simulation, because the τ-step $P \sqcap P \xrightarrow{\tau} (\frac{1}{2}\cdot \overline{a} + \frac{1}{2}\cdot \overline{b})$ can be matched by the idle transition $(\frac{1}{2}\cdot \overline{a} + \frac{1}{2}\cdot \overline{b}) \stackrel{\hat{\tau}}{\Longrightarrow} (\frac{1}{2}\cdot \overline{a} + \frac{1}{2}\cdot \overline{b})$, and we have $(\frac{1}{2}\cdot \overline{a} + \frac{1}{2}\cdot \overline{b}) \;\overline{\mathcal{R}}\; (\frac{1}{2}\cdot \overline{a} + \frac{1}{2}\cdot \overline{b})$. Thus $(P \sqcap P) \lhd_S (\frac{1}{2}\cdot \overline{a} + \frac{1}{2}\cdot \overline{b}) = [\![P]\!]$, hence $[\![P \sqcap P]\!] \,\overline{\lhd_S}\, [\![P]\!]$, and therefore $P \sqcap P \sqsubseteq_S P$.

This type of reasoning does not apply to the other direction. Any simulation $\mathcal{R}$ with $(\frac{1}{2}\cdot \overline{a} + \frac{1}{2}\cdot \overline{b}) \;\overline{\mathcal{R}}\; \overline{P \sqcap P}$ would have to satisfy $a \;\mathcal{R}\; \overline{P \sqcap P}$ and $b \;\mathcal{R}\; \overline{P \sqcap P}$. However, the

[4]It is called *forward simulation* in [36].

[5]We have reversed the orientation of the symbols $\rhd_S$ and $\rhd_{FS}$ w.r.t. [8] and [9]; the pointy side now points to a single state, and the flat side to a distribution.

move $a \xrightarrow{a} \mathbf{0}$ cannot be matched by the process $\overline{P \sqcap P}$, as the only transition the latter process can do is $\overline{P \sqcap P} \xrightarrow{\tau} (\frac{1}{2}\cdot \overline{a} + \frac{1}{2}\cdot \overline{b})$, and only half of that distribution can match the a-move. Thus, no such simulation exists, and we find $[\![P]\!] \;\overline{\not\lhd_S}\; [\![P \sqcap P]\!]$. Nevertheless, we still have $P \sqsubseteq_S P \sqcap P$. Here, the transition $\stackrel{\hat{\tau}}{\Longrightarrow}$ from Definition 4.3 comes to the rescue. As $[\![P \sqcap P]\!] \stackrel{\hat{\tau}}{\Longrightarrow} [\![P]\!]$ and $[\![P]\!] \;\overline{\lhd_S}\; [\![P]\!]$, we obtain $P \sqsubseteq_S P \sqcap P$.

Example 4.5. Let $P = a \;{}_{\frac{1}{2}}\!\oplus\; b$ and $Q = P \;\Box\; P$. We have $P \sqsubseteq_S Q$ because $[\![P]\!] \;\overline{\lhd_S}\; [\![Q]\!]$ which comes from the following observations:

(1) $[\![P]\!] = \frac{1}{2}\cdot \overline{a} + \frac{1}{2}\cdot \overline{b}$
(2) $[\![Q]\!] = \frac{1}{2}\cdot(\frac{1}{2}\cdot \overline{a \;\Box\; a} + \frac{1}{2}\cdot \overline{a \;\Box\; b}) + \frac{1}{2}\cdot(\frac{1}{2}\cdot \overline{b \;\Box\; a} + \frac{1}{2}\cdot \overline{b \;\Box\; b})$
(3) $a \lhd_S (\frac{1}{2}\cdot \overline{a \;\Box\; a} + \frac{1}{2}\cdot \overline{a \;\Box\; b})$
(4) $b \lhd_S (\frac{1}{2}\cdot \overline{b \;\Box\; a} + \frac{1}{2}\cdot \overline{b \;\Box\; b})$

This kind of reasoning does not apply to $\lhd_{FS}$. For example, we have $a \not\lhd_{FS} (\frac{1}{2}\cdot \overline{a \;\Box\; a} + \frac{1}{2}\cdot \overline{a \;\Box\; b})$ because the state on the left hand side can refuse to do action b while the distribution on the right hand side cannot. Indeed, it holds that $Q \not\sqsubseteq_{FS} P$.

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

Proposition 4.6. *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, Corollary 6.10 and Theorem 6.13]; the case $\sqsubseteq_{FS}$ is analogous. As an example, we show that $\sqsubseteq_{FS}$ is preserved under parallel composition. The key step is to show that the binary relation $\mathcal{R} \subseteq \mathsf{sCSP} \times \mathcal{D}(\mathsf{sCSP})$ defined by

$$\mathcal{R} := \{(s_1|_A s_2, \Delta_1|_A\Delta_2) \mid s_1 \lhd_{FS} \Delta_1 \wedge s_2 \lhd_{FS} \Delta_2\}.$$

is a failure simulation.

Suppose $s_i \lhd_{FS} \Delta_i$ for $i = 1, 2$ and $s_1 \,|_A\, s_2 \stackrel{X}{\not\longrightarrow}$ for some $X \subseteq \mathsf{Act}$. For each $a \in X$ there are two possibilities:

- If $a \notin A$ then $s_1 \stackrel{a}{\not\longrightarrow}$ and $s_2 \stackrel{a}{\not\longrightarrow}$, since otherwise we would have $s_1 \,|_A\, s_2 \xrightarrow{a}$.
- If $a \in A$ then either $s_1 \stackrel{a}{\not\longrightarrow}$ or $s_2 \stackrel{a}{\not\longrightarrow}$, since otherwise we would have $s_1 \,|_A\, s_2 \xrightarrow{\tau}$.

Hence we can partition the set X into three subsets: X_0, X_1 and X_2 such that $X_0 = X\backslash A$ and $X_1 \cup X_2 \subseteq A$ with $s_1 \stackrel{X_1}{\not\longrightarrow}$ and $s_2 \stackrel{X_2}{\not\longrightarrow}$, but allowing $s_1 \stackrel{a}{\not\longrightarrow}$ for some $a \in X_2$ and $s_2 \stackrel{a}{\not\longrightarrow}$ for some $a \in X_1$. We then have that $s_i \stackrel{X_0 \cup X_i}{\not\longrightarrow}$ for $i = 1, 2$. By the assumption that $s_i \lhd_{FS} \Delta_i$ for $i = 1, 2$, there is a Δ_i' with $\Delta_i \stackrel{\hat{\tau}}{\Longrightarrow} \Delta_i' \stackrel{X_0 \cup X_i}{\not\longrightarrow}$. Therefore $\Delta_1'|_A\Delta_2' \stackrel{X}{\not\longrightarrow}$ as well. It is stated in [8, Lemma 6.12(i)] that if $\Phi \stackrel{\hat{\tau}}{\Longrightarrow} \Phi'$ then $\Phi \,|_A\, \Delta \stackrel{\hat{\tau}}{\Longrightarrow} \Phi' \,|_A\, \Delta$ and $\Delta \,|_A\, \Phi \stackrel{\hat{\tau}}{\Longrightarrow} \Delta \,|_A\, \Phi'$. So we have $\Delta_1 \,|_A\, \Delta_2 \stackrel{\hat{\tau}}{\Longrightarrow} \Delta_1' \,|_A\, \Delta_2'$. Hence $\Delta_1 \,|_A\, \Delta_2$ can match up the failures of $s_1 \,|_A\, s_2$.

The matching up of transitions and the using of $\mathcal{R}$ to prove the preservation property of $\sqsubseteq_{FS}$ under parallel composition are similar to those in the corresponding proof for simulations [8, Theorem 6.13(v)], so we omit them. ☐

We recall the following result from [8, Theorem 6.17].

Theorem 4.7. *If* $P \sqsubseteq_S Q$ *then* $P \sqsubseteq_{\mathrm{pmay}} Q$.

Proof. For any test $T \in \mathsf{pCSP}^\omega$ and process $P \in \mathsf{pCSP}$ the set $\mathbb{V}(T \,|_{\mathsf{Act}}\, P)$ is finite, so

$$P \sqsubseteq_{\mathrm{pmay}} Q \text{ iff } \mathit{max}(\mathbb{V}([\![T \,|_{\mathsf{Act}}\, P]\!])) \leq \mathit{max}(\mathbb{V}([\![T \,|_{\mathsf{Act}}\, Q]\!])) \text{ for every test } T. \tag{4.1}$$

The following properties for $\Delta_1, \Delta_2 \in \mathsf{pCSP}^\omega$ and $\alpha \in \mathsf{Act}_\tau$ are not hard to establish:

$$\Delta_1 \stackrel{\hat{\alpha}}{\Longrightarrow} \Delta_2 \text{ implies } \mathit{max}(\mathbb{V}(\Delta_1)) \geq \mathit{max}(\mathbb{V}(\Delta_2)). \tag{4.2}$$

$$\Delta_1 \overline{\lhd_S}\, \Delta_2 \text{ implies } \mathit{max}(\mathbb{V}(\Delta_1)) \leq \mathit{max}(\mathbb{V}(\Delta_2)). \tag{4.3}$$

In [8, Lemma 6.15 and Proposition 6.16] similar properties are proven using a function *maxlive* instead of $\mathit{max}\circ\mathbb{V}$. The same arguments apply here.

Now suppose $P \sqsubseteq_S Q$. Since $\sqsubseteq_S$ is preserved by the parallel operator we have that $T\,|_{\mathsf{Act}}\, P \sqsubseteq_S T\,|_{\mathsf{Act}}\, Q$ for an arbitrary test T. By definition, this means that there is a distribution Δ such that $[\![T\,|_{\mathsf{Act}}\, Q]\!] \stackrel{\hat{\tau}}{\Longrightarrow} \Delta$ and $[\![T\,|_{\mathsf{Act}}\, P]\!]\, \overline{\lhd_S}\, \Delta$. By (4.2) and (4.3) we infer that $\mathit{max}(\mathbb{V}([\![T\,|_{\mathsf{Act}}\, P]\!])) \leq \mathit{max}(\mathbb{V}([\![T\,|_{\mathsf{Act}}\, Q]\!]))$. The result now follows from (4.1). $\square$

It is tempting to use the same idea to prove that $\sqsubseteq_{FS}$ implies $\sqsubseteq_{\text{pmust}}$, but now using the function $\mathit{min}\circ\mathbb{V}$. However, the *min*-analogue of Property (4.2) is in general invalid. For example, let R be the process $a\,|_{\mathsf{Act}}\,(a \,\Box\, \omega)$. We have $\mathit{min}(\mathbb{V}(R)) = 1$, yet $R \xrightarrow{\tau} \overline{\mathbf{0}\,|_{\mathsf{Act}}\,\mathbf{0}}$ and $\mathit{min}(\mathbb{V}(\overline{\mathbf{0}\,|_{\mathsf{Act}}\,\mathbf{0}})) = 0$. Therefore, it is not the case that $\Delta_1 \stackrel{\hat{\tau}}{\Longrightarrow} \Delta_2$ implies $\mathit{min}(\mathbb{V}(\Delta_1)) \leq \mathit{min}(\mathbb{V}(\Delta_2))$.

Our strategy is therefore as follows. Write $s \xrightarrow{\alpha}_\omega \Delta$ if both $s \stackrel{\omega}{\not\longrightarrow}$ and $s \xrightarrow{\alpha} \Delta$ hold. We define $\xrightarrow{\hat{\tau}}_\omega$ as $\xrightarrow{\hat{\tau}}$ using $\xrightarrow{\tau}_\omega$ in place of $\xrightarrow{\tau}$. Similarly we define $\Longrightarrow_\omega$ and $\stackrel{\hat{\alpha}}{\Longrightarrow}_\omega$. Thus the subscript ω on a transition of any kind indicates that no state is passed through in which ω is enabled. A version of failure simulation adapted to these transition relations is then defined as follows.

Definition 4.8. Let $\lhd^e_{FS} \subseteq \mathsf{sCSP}^\omega \times \mathcal{D}(\mathsf{sCSP}^\omega)$ be the largest relation such that $s \lhd^e_{FS} \Theta$ implies

- if $s \xrightarrow{\alpha}_\omega \Delta$ then there is some Θ' with $\Theta \stackrel{\hat{\alpha}}{\Longrightarrow}_\omega \Theta'$ and $\Delta\, \overline{\lhd^e_{FS}}\, \Theta'$
- if $s \stackrel{X}{\not\longrightarrow}$ with $\omega \in X$ then there is some Θ' with $\Theta \stackrel{\hat{\tau}}{\Longrightarrow}_\omega \Theta'$ and $\Theta' \stackrel{X}{\not\longrightarrow}$.

Let $P \sqsubseteq^e_{FS} Q$ iff $[\![P]\!] \stackrel{\hat{\tau}}{\Longrightarrow}_\omega \Theta$ for some Θ with $[\![Q]\!]\, \overline{\lhd^e_{FS}}\, \Theta$.

Note that for processes P, Q in pCSP (as opposed to pCSP^ω), we have $P \sqsubseteq_{FS} Q$ iff $P \sqsubseteq^e_{FS} Q$.

Proposition 4.9. *If P, Q are processes in pCSP with $P \sqsubseteq_{FS} Q$ and T is a process in pCSP^ω then $T\,|_{\mathsf{Act}}\, P \sqsubseteq^e_{FS} T\,|_{\mathsf{Act}}\, Q$.*

Proof. Similar to the proof of Proposition 4.6. $\square$

Proposition 4.10. *The following properties hold for $\mathit{min}\circ\mathbb{V}$, with $\Delta_1, \Delta_2 \in \mathcal{D}(\mathsf{sCSP}^\omega)$:*

$$P \sqsubseteq_{\text{pmust}} Q \text{ iff } \mathit{min}(\mathbb{V}([\![T\,|_{\mathsf{Act}}\, P]\!])) \leq \mathit{min}(\mathbb{V}([\![T\,|_{\mathsf{Act}}\, Q]\!])) \text{ for every test } T. \tag{4.4}$$

$$\Delta_1 \stackrel{\hat{\alpha}}{\Longrightarrow}_\omega \Delta_2 \text{ for } \alpha \in \mathsf{Act}_\tau \text{ implies } \mathit{min}(\mathbb{V}(\Delta_1)) \leq \mathit{min}(\mathbb{V}(\Delta_2)). \tag{4.5}$$

$$\Delta_1 \overline{\lhd^e_{FS}}\, \Delta_2 \text{ implies } \mathit{min}(\mathbb{V}(\Delta_1)) \geq \mathit{min}(\mathbb{V}(\Delta_2)). \tag{4.6}$$

Proof. Property (4.4) is again straightforward, and Property (4.5) can be established just as in Lemma 6.15 in [8], but with all $\leq$-signs reversed. Property (4.6) follows by structural induction, simultaneously with the property, for $s \in \mathsf{sCSP}^\omega$ and $\Delta \in \mathcal{D}(\mathsf{sCSP}^\omega)$, that

$$s \lhd^e_{FS} \Delta \text{ implies } \mathit{min}(\mathbb{V}(s)) \geq \mathit{min}(\mathbb{V}(\Delta))\ . \tag{4.7}$$

The reduction of Property (4.6) to (4.7) proceeds exactly as in [8, Lemma 6.16(ii)]. For (4.7) itself we distinguish three cases:

- If $s \xrightarrow{\omega}$, then $\mathit{min}(\mathbb{V}(s)) = 1 \geq \mathit{min}(\mathbb{V}(\Delta))$ trivially.

- If $s \xrightarrow{\omega}\!\!\!\!\not\;\;$ but $s \rightarrow$, then we can closely follow the proof of [8, Lemma 6.16(i)]: Whenever $s \xrightarrow{\alpha}_{\omega} \Theta$, for $\alpha \in \mathsf{Act}_\tau$ and $\Theta \in \mathcal{D}(\mathsf{sCSP}^\omega)$, then $s \lhd^e_{FS} \Delta$ implies the existence of some Δ_Θ such that $\Delta \xrightarrow{\hat{\alpha}}{}^{*}_{\omega} \Delta_\Theta$ and $\Theta \overline{\lhd^e_{FS}} \Delta_\Theta$. By induction, using (4.6), it follows that $min(\mathbb{V}(\Theta)) \geq min(\mathbb{V}(\Delta_\Theta))$. Consequently, we have that

$$\begin{array}{rcll} min(\mathbb{V}(s)) & = & min(\{min(\mathbb{V}(\Theta)) \mid s \xrightarrow{\alpha} \Theta\}) & \\ & \geq & min(\{min(\mathbb{V}(\Delta_\Theta)) \mid s \xrightarrow{\alpha} \Theta\}) & \\ & \geq & min(\{min(\mathbb{V}(\Delta)) \mid s \xrightarrow{\alpha} \Theta\}) & \text{(by (4.5))} \\ & = & min(\mathbb{V}(\Delta)) . & \end{array}$$

- If $s \not\rightarrow$, that is $s \xrightarrow{\mathsf{Act}^\omega}\!\!\!\!\!\not\;\;\;$, then there is some Δ' such that $\Delta \stackrel{\hat{\tau}}{\Longrightarrow}_\omega \Delta'$ and $\Delta' \xrightarrow{\mathsf{Act}^\omega}\!\!\!\!\!\not\;\;\;$. By the definition of $\mathbb{V}$, $min(\mathbb{V}(\Delta'))=0$. Using (4.5), we have $min(\mathbb{V}(\Delta)) \leq min(\mathbb{V}(\Delta'))$, so $min(\mathbb{V}(\Delta)) = 0$ as well. Thus, also in this case $min(\mathbb{V}(s)) \geq min(\mathbb{V}(\Delta))$. ☐

Theorem 4.11. *If $P \sqsubseteq_{FS} Q$ then $P \sqsubseteq_{\mathrm{pmust}} Q$.*

Proof. Similar to the proof of Theorem 4.7, using (4.4)–(4.6). ☐

The next four sections are devoted to proving the converse of Theorems 4.7 and 4.11.

5. State- versus action-based testing

Much work on testing [6, 41, 8] uses success *states* marked by outgoing ω-actions; this is referred to as *state-based* testing, which we have used in Section 3 to define the preorders $\sqsubseteq_{\mathrm{may}}$ and $\sqsubseteq_{\mathrm{must}}$. In other work [37, 10], however, it is the *actual execution* of ω that constitutes success. This *action-based* approach is formalised as in the state-based approach, via a modified results-gathering function:

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

As in the original $\mathbb{V}$, the α's are non-success actions, including τ; and again, this is done for generality, since in testing outcomes the only non-success action is τ.

If we use this results-gathering function rather than $\mathbb{V}$ in Definitions 3.1 and 3.2 we obtain the two slightly different testing preorders, $\widehat{\sqsubseteq}_{\mathrm{pmay}}$ and $\widehat{\sqsubseteq}_{\mathrm{pmust}}$. The following proposition shows that state-based testing is at least as discriminating as action-based testing:

Proposition 5.1.

(1) If $P \sqsubseteq_{\mathrm{pmay}} Q$ then $P \widehat{\sqsubseteq}_{\mathrm{pmay}} Q$.
(2) If $P \sqsubseteq_{\mathrm{pmust}} Q$ then $P \widehat{\sqsubseteq}_{\mathrm{pmust}} Q$.

Proof. For any action-based test $\widehat{T}$ we construct a state-based test T by replacing each subterm $\omega.Q$ by $\tau.\omega$; then we have $\mathbb{V}[\![T \mid_{\mathsf{Act}} P]\!] = \widehat{\mathbb{V}}[\![\widehat{T} \mid_{\mathsf{Act}} P]\!]$ for all pCSP processes P. ☐

Proposition 5.1 enables us to reduce our main goal, the converse of Theorems 4.7 and 4.11, to the following property.

Theorem 5.2.

(1) If $P \widehat{\sqsubseteq}_{\mathrm{pmay}} Q$ then $P \sqsubseteq_S Q$.
(2) If $P \widehat{\sqsubseteq}_{\mathrm{pmust}} Q$ then $P \sqsubseteq_{FS} Q$.

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

Once we have obtained this theorem, it follows that in our framework of finite probabilistic processes the state-based and action-based testing preorders coincide. This result no longer holds in the presence of divergence, at least for must-testing.

Example 5.3. Suppose we extend our syntax with a state-based process Ω, to model divergence, and the operational semantics of Figure 1 with the rule

$$\Omega \xrightarrow{\tau} \overline{\Omega}.$$

It is possible to extend the results-gathering functions $\mathbb{V}$ and $\widehat{\mathbb{V}}$ to these infinite processes, although the definitions are no longer inductive (cf. Definition 5 of [10] or Definition A.3 of the appendix). In this extended setting we will have $a.\Omega \not\sqsubseteq_{\text{pmust}} a.\Omega \sqcap 0$ because of the test $a.\omega$:

$$\mathbb{V}([\![a.\omega\mid_{\mathsf{Act}} a.\Omega]\!]) = \{1\} \text{ while } \mathbb{V}([\![a.\omega\mid_{\mathsf{Act}} a.\Omega \sqcap 0]\!]) = \{0,1\}.$$

This intuitively is due to the fact that the Ω-encoded divergence of the left-hand process occurs only *after* the first action a; and since the left-hand process cannot deadlock before that action, relation $\sqsubseteq_{\text{must}}$ would prevent the right-hand process from doing so.

However, a peculiarity of action-based testing is that success actions can be indefinitely inhibited by infinite τ-branches. We have

$$\widehat{\mathbb{V}}([\![a.\omega\mid_{\mathsf{Act}} a.\Omega]\!]) = \widehat{\mathbb{V}}([\![a.\omega\mid_{\mathsf{Act}} a.\Omega \sqcap 0]\!]) = \{0,1\}.$$

Indeed no test can be found to distinguish them, and so one can show $a.\Omega \mathrel{\widehat{\sqsubseteq}_{\text{pmust}}} a.\Omega \sqcap 0$.

Note that probabilistic behaviour plays no role in this counter-example. In CSP (without probabilities) there is no difference between $\widehat{\sqsubseteq}_{\text{may}}$ and $\sqsubseteq_{\text{may}}$, whereas $\widehat{\sqsubseteq}_{\text{must}}$ is strictly less discriminating than $\sqsubseteq_{\text{must}}$. For finitely branching processes, the CSP refinement preorder based on failures and divergences [2, 17, 34] coincides with the state-based relation $\sqsubseteq_{\text{must}}$.

6. Vector-based testing

This section describes another variation on testing, a richer testing framework due to Segala [37], 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 Theorem 5.2.

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* $\widehat{\mathcal{A}}^\Omega(T,P) \subseteq [0,1]^\Omega$. As with standard scalar testing, each outcome arises from a resolution of the nondeterministic choices in $T\mid_{\mathsf{Act}} P$. However, here an outcome is a *tuple* and its ω-component gives the probability that this resolution will perform the success action ω.

For vector-based testing we again inductively define a results-gathering function, but first we require some auxiliary notation. For any action α define $\alpha! : [0,1]^\Omega \to [0,1]^\Omega$ by

$$\alpha!o(\omega) = \begin{cases} 1 & \text{if } \omega{=}\alpha \\ o(\omega) & \text{otherwise} \end{cases}$$

so that if α is a success action, in Ω, then $\alpha!$ updates the tuple to 1 at that point, leaving it unchanged otherwise, and when $\alpha \notin \Omega$ the function $\alpha!$ is the identity. These functions lift to sets $O \subseteq [0,1]^\Omega$ as usual, via $\alpha!O := \{\alpha!o \mid o \in O\}$.

Next, for any set X define its *convex closure* $\updownarrow X$ by

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

Here, as usual, I is assumed to be a finite index set. Finally, $\vec{0} \in [0,1]^\Omega$ is given by $\vec{0}(\omega) = 0$ for all $\omega \in \Omega$. Let pCSP^Ω be the set of Ω-tests, and sCSP^Ω the set of state-based Ω-tests.

Definition 6.1. The *action-based, vector-based, convex-closed results-gathering function* $\widehat{\mathbb{V}}^\Omega_\updownarrow : \mathsf{sCSP}^\Omega \to \mathcal{P}^+([0,1]^\Omega)$ is given by

$$\widehat{\mathbb{V}}^\Omega_\updownarrow(s) := \begin{cases} \updownarrow \bigcup \{ \alpha!(\widehat{\mathbb{V}}^\Omega_\updownarrow(\Delta)) \mid s \xrightarrow{\alpha} \Delta,\ \alpha \in \Omega \cup \mathsf{Act}_\tau \} & \text{if } s \to \\ \{\vec{0}\} & \text{otherwise} \end{cases} \tag{6.1}$$

As with our previous results-gathering functions $\mathbb{V}$ and $\widehat{\mathbb{V}}$, this function extends to the type $\mathcal{D}(\mathsf{sCSP}^\Omega) \to \mathcal{P}^+([0,1]^\Omega)$ via the convention $\widehat{\mathbb{V}}^\Omega_\updownarrow(\Delta) := \mathrm{Exp}_\Delta \widehat{\mathbb{V}}^\Omega_\updownarrow$.

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

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

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

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

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

We will explain the rôle of convex-closure $\updownarrow$ in this definition. Let $\widehat{\mathbb{V}}^\Omega$ be defined as $\widehat{\mathbb{V}}^\Omega_\updownarrow$ above, but omitting the use of $\updownarrow$. It is easy to see that $\widehat{\mathbb{V}}^\Omega_\updownarrow(s) = \updownarrow \widehat{\mathbb{V}}^\Omega(s)$ for all $s \in \mathsf{sCSP}^\Omega$.

Applying convex closure to subsets of the one-dimensional interval $[0,1]$ (such as arise from applying scalar tests to processes) has no effect on the Hoare and Smyth orders between these subsets:

Lemma 6.2. *Suppose* $X,\ Y \subseteq [0,1]$. *Then*

(1) $X \leq_{\mathrm{Ho}} Y$ *if and only if* $\updownarrow X \leq_{\mathrm{Ho}} \updownarrow Y$.
(2) $X \leq_{\mathrm{Sm}} Y$ *if and only if* $\updownarrow X \leq_{\mathrm{Sm}} \updownarrow Y$.

Proof. We restrict attention to (1); the proof of (2) goes likewise. It suffices to show that (i) $X \leq_{\mathrm{Ho}} \updownarrow X$ and (ii) $\updownarrow X \leq_{\mathrm{Ho}} X$. We only prove (ii) since (i) is obvious. Suppose $x \in \updownarrow X$, then $x = \sum_{i\in I} p_i x_i$ for a finite set I with $\sum_{i\in I} p_i = 1$ and $x_i \in X$. Let $x^* = max\{x_i \mid i \in I\}$. Then

$$x \ = \ \sum_{i\in I} p_i x_i \ \leq \ \sum_{i\in I} p_i x^* \ = \ x^* \ \in \ X. \qquad \square$$

It follows that for scalar testing it makes no difference whether convex closure is employed or not. Vector-based testing, as proposed in Definition 6.1, is a conservative extension of action-based testing, as described in Section 5:

Corollary 6.3. *Suppose* Ω *is the singleton set* $\{\omega\}$. *Then*

(1) $P \mathrel{\widehat{\sqsubseteq}^\Omega_{\mathrm{pmay}}} Q$ *if and only if* $P \mathrel{\widehat{\sqsubseteq}_{\mathrm{pmay}}} Q$.
(2) $P \mathrel{\widehat{\sqsubseteq}^\Omega_{\mathrm{pmust}}} Q$ *if and only if* $P \mathrel{\widehat{\sqsubseteq}_{\mathrm{pmust}}} Q$.

Proof. $\widehat{\mathbb{V}}^\Omega_\updownarrow = \updownarrow \widehat{\mathbb{V}}^\Omega = \updownarrow \widehat{\mathbb{V}}$ when Ω is $\{\omega\}$, so the result follows from Lemma 6.2. $\square$

Lemma 6.2 does not generalise to $[0,1]^k$, when $k > 1$, as the following example demonstrates:

Example 6.4. Let X, Y denote $\{(0.5, 0.5)\}$, $\{(1,0),(0,1)\}$ respectively. Then it is easy to show that $\updownarrow X \leq_{\rm Ho} \updownarrow Y$ although obviously $X \not\leq_{\rm Ho} Y$.

This example can be exploited to show that for vector-based testing it *does* make a difference whether convex closure is employed.

Example 6.5. Consider the two processes

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

Take $\Omega = \{\omega_1, \omega_2\}$. Employing the results-gathering function $\widehat{\mathbb{V}}^\Omega$, without convex closure, with the test $T := a.\omega_1 \,\Box\, b.\omega_2$ we obtain

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

As pointed out in Example 6.4, this entails $\widehat{\mathcal{A}}^\Omega(T,P) \not\leq_{\rm Ho} \widehat{\mathcal{A}}^\Omega(T,Q)$, although their convex closures $\widehat{\mathcal{A}}^\Omega_{\updownarrow}(T,P)$ and $\widehat{\mathcal{A}}^\Omega_{\updownarrow}(T,Q)$ *are* related under the Hoare preorder.

Convex closure is a uniform way of ensuring that internal choice can simulate an arbitrary probabilistic choice [14]. For the processes P and Q of Example 6.5 it is obvious that $P \sqsubseteq_S Q$, and from Theorem 4.7 it therefore follows that $P \sqsubseteq_{\rm pmay} Q$. This fits with the intuition that a probabilistic choice is an acceptable implementation of a nondeterministic choice occurring in a specification. Considering that we use $\widehat{\sqsubseteq}^\Omega_{\rm pmay}$ as a stepping stone in showing the coincidence of $\sqsubseteq_S$ and $\sqsubseteq_{\rm pmay}$, we must have $P \mathrel{\widehat{\sqsubseteq}^\Omega_{\rm pmay}} Q$. For this reason we use convex closure in Definition 6.1.

In [10] the results-gathering function $\widehat{\mathbb{V}}^\Omega_{\updownarrow}$ with $\Omega = \{\omega_1, \omega_2, \cdots\}$ was called simply $\mathbb{W}$ (because action-based/vector-based/convex-closed 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. As we show in Proposition A.6 of the appendix, the inductive Definition 6.1 above yields the same results. In the present paper our interest in vector-based testing stems from the following result.

Theorem 6.6.

(1) $P \mathrel{\widehat{\sqsubseteq}^\Omega_{\rm pmay}} Q$ *iff* $P \mathrel{\widehat{\sqsubseteq}_{\rm pmay}} Q$

(2) $P \mathrel{\widehat{\sqsubseteq}^\Omega_{\rm pmust}} Q$ *iff* $P \mathrel{\widehat{\sqsubseteq}_{\rm pmust}} Q$. □

Proof. In [10, Theorem 3] this theorem has been established for versions of $\widehat{\sqsubseteq}^\Omega_{\rm pmay}$ and $\widehat{\sqsubseteq}^\Omega_{\rm pmust}$ where tests are finite probabilistic automata, as defined in our Appendix A. The key argument is that when $P \mathrel{\widehat{\sqsubseteq}^\Omega_{\rm pmay}} Q$ can be refuted by means of a vector-based test T, then $P \mathrel{\widehat{\sqsubseteq}_{\rm pmay}} Q$ can be refuted by means of a scalar test $T \| U$, where U is administrative code which collates the vector of results produced by T and effectively renders them as a unique scalar result, and similarly for $\widehat{\sqsubseteq}^\Omega_{\rm pmust}$. This theorem applies to our setting as well, due to the observation that if a test T can be represented as a pCSP^Ω-expression, then so can the test $T \| U$. □

Because of Theorem 6.6, in order to establish Theorem 5.2 it will suffice to show that

- $P \mathrel{\widehat{\sqsubseteq}^{\Omega}_{\rm pmay}} Q$ implies $P \sqsubseteq_S Q$ and
- $P \mathrel{\widehat{\sqsubseteq}^{\Omega}_{\rm pmust}} Q$ implies $P \sqsubseteq_{FS} Q$.

This shift from scalar testing to vector-based testing is motivated by the fact that the latter enables us to use more informative tests, allowing us to discover more intensional properties of the processes being tested.

The crucial characteristics of $\widehat{\mathcal{A}}^{\Omega}_{\updownarrow}$ needed for the above implications are summarised in Lemmas 6.7 and 6.8. For convenience of presentation, we write $\vec{\omega}$ for the vector in $[0,1]^{\Omega}$ defined by $\vec{\omega}(\omega) = 1$ and $\vec{\omega}(\omega') = 0$ for $\omega' \neq \omega$. Sometimes we treat a distribution Δ of finite support as the pCSP expression $\bigoplus_{s\in\lceil\Delta\rceil} \Delta(s){\cdot}s$, so that $\widehat{\mathcal{A}}^{\Omega}_{\updownarrow}(T,\Delta) := \mathrm{Exp}_{\Delta}\widehat{\mathcal{A}}^{\Omega}_{\updownarrow}(T,\_)$.

Lemma 6.7. *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) $\vec{0} \in \widehat{\mathcal{A}}^{\Omega}_{\updownarrow}(\Box_{a\in X}\, a.\omega, P)$ *iff* $\exists\Delta : [\![P]\!] \stackrel{\hat{\tau}}{\Longrightarrow} \Delta \stackrel{X}{\not\longrightarrow}$.

(3) 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* $o \in \widehat{\mathcal{A}}^{\Omega}_{\updownarrow}(T,\Delta)$.

(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 some* $o_i \in \widehat{\mathcal{A}}^{\Omega}_{\updownarrow}(T_i, P)$.

(5) $o \in \widehat{\mathcal{A}}^{\Omega}_{\updownarrow}(\bigsqcap_{i\in I} T_i, P)$ *if for all* $i \in I$ *there are* $q_i \in [0,1]$ *and* $\Delta_i \in \mathcal{D}(\mathsf{sCSP})$ *such that* $\sum_{i\in I} q_i = 1$, $[\![P]\!] \stackrel{\hat{\tau}}{\Longrightarrow} \sum_{i\in I} q_i \cdot \Delta_i$ *and* $o = \sum_{i\in I} q_i o_i$ *for some* $o_i \in \widehat{\mathcal{A}}^{\Omega}_{\updownarrow}(T_i, \Delta_i)$.

Proof. Straightforward, by induction on the structure of P. $\square$

The converse of Lemma 6.7 (5) also holds, as the following lemma says. However, the proof is less straightforward.

Lemma 6.8. *Let P be a* pCSP *process, and T_i be tests. If* $o \in \widehat{\mathcal{A}}^{\Omega}_{\updownarrow}(\bigsqcap_{i\in I} T_i, P)$ *then for all* $i \in I$ *there are* $q_i \in [0,1]$ *and* $\Delta_i \in \mathcal{D}(\mathsf{sCSP})$ *with* $\sum_{i\in I} q_i = 1$ *such that* $[\![P]\!] \stackrel{\hat{\tau}}{\Longrightarrow} \sum_{i\in I} q_i \cdot \Delta_i$ *and* $o = \sum_{i\in I} q_i o_i$ *for some* $o_i \in \widehat{\mathcal{A}}^{\Omega}_{\updownarrow}(T_i, \Delta_i)$.

Proof. Given that the states of our pLTS are sCSP expressions, there exists a well-founded order on the combination of states in sCSP and distributions in $\mathcal{D}(\mathsf{sCSP})$, such that $s \stackrel{\alpha}{\longrightarrow} \Delta$ implies that s is larger than Δ, and any distribution is larger than the states in its support. Intuitively, this order corresponds to the usual order on natural numbers if we graphically depict a pLTS as a finite tree (cf. Section 2) and assign to each node a number to indicate its level in the tree. Let $T = \bigsqcap_{i\in I} T_i$. We prove the following two claims

(a) If s is a state-based process and $o \in \widehat{\mathcal{A}}^{\Omega}_{\updownarrow}(T, s)$ then there are some $\{q_i\}_{i\in I}$ with $\sum_{i\in I} q_i = 1$ such that $s \stackrel{\hat{\tau}}{\Longrightarrow} \sum_{i\in I} q_i \cdot \Delta_i$, $o = \sum_{i\in I} q_i o_i$, and $o_i \in \widehat{\mathcal{A}}^{\Omega}_{\updownarrow}(T_i, \Delta_i)$.

(b) If $\Delta \in \mathcal{D}(\mathsf{sCSP})$ and $o \in \widehat{\mathcal{A}}^{\Omega}_{\updownarrow}(T, \Delta)$ then there are some $\{q_i\}_{i\in I}$ with $\sum_{i\in I} q_i = 1$ such that $\Delta \stackrel{\hat{\tau}}{\Longrightarrow} \sum_{i\in I} q_i \cdot \Delta_i$, $o = \sum_{i\in I} q_i o_i$, and $o_i \in \widehat{\mathcal{A}}^{\Omega}_{\updownarrow}(T_i, \Delta_i)$.

by simultaneous induction on the order mentioned above, applied to s and Δ.

(a) We have two sub-cases depending on whether s can make an initial τ-move or not.

- If s cannot make a τ-move, that is $s \stackrel{\tau}{\not\longrightarrow}$, then the only possible moves from $T \mid_{\mathsf{Act}} s$ are τ-moves originating in T; T has no non-τ moves, and any non-τ moves that might be possible for s on its own are inhibited by the alphabet Act of the composition. Suppose $o \in \widehat{\mathcal{A}}^{\Omega}_{\updownarrow}(T, s)$. Then by definition (6.1) there are some $\{q_i\}_{i\in I}$ with $\sum_{i\in I} q_i = 1$ such that $o = \sum_{i\in I} q_i o_i$ and $o_i \in \widehat{\mathcal{A}}^{\Omega}_{\updownarrow}(T_i, s) = \widehat{\mathcal{A}}^{\Omega}_{\updownarrow}(T_i, \overline{s})$. Obviously we also have $[\![s]\!] \stackrel{\hat{\tau}}{\Longrightarrow} \sum_{i\in I} q_i \cdot \overline{s}$.

- If s can make one or more τ-moves, then we have $s \xrightarrow{\tau} \Delta'_j$ for $j \in J$, where without loss of generality J can be assumed to be a non-empty finite set disjoint from I, the index set for T. The possible first moves for $T \mid_{\mathsf{Act}} s$ are τ-moves either of T or of s, because T cannot make initial non-τ moves and that prevents a proper synchronisation from occurring on the first step. Suppose that $o \in \widehat{\mathcal{A}}^{\Omega}_{\updownarrow}(T, s)$. Then by definition (6.1) there are some $\{p_k\}_{k\in I\cup J}$ with $\sum_{k\in I\cup J} p_k = 1$ and
$$o = \sum_{k\in I\cup J} p_k o'_k \tag{6.2}$$
$$o'_i \in \widehat{\mathcal{A}}^{\Omega}_{\updownarrow}(T_i, s) \qquad \text{for all } i \in I \tag{6.3}$$
$$o'_j \in \widehat{\mathcal{A}}^{\Omega}_{\updownarrow}(T, \Delta_j) \qquad \text{for all } j \in J. \tag{6.4}$$
For each $j \in J$, we know by the induction hypothesis that
$$\Delta'_j \stackrel{\hat{\tau}}{\Longrightarrow} \sum_{i\in I} p_{ji} \cdot \Delta'_{ji} \tag{6.5}$$
$$o'_j = \sum_{i\in I} p_{ji} o'_{ji} \tag{6.6}$$
$$o'_{ji} \in \widehat{\mathcal{A}}^{\Omega}_{\updownarrow}(T_i, \Delta'_{ji}) \tag{6.7}$$
for some $\{p_{ji}\}_{i\in I}$ with $\sum_{i\in I} p_{ji} = 1$. Let
$$q_i = p_i + \sum_{j\in J} p_j p_{ji}$$
$$\Delta_i = \frac{1}{q_i}(p_i \cdot \overline{s} + \sum_{j\in J} p_j p_{ji} \cdot \Delta'_{ji})$$
$$o_i = \frac{1}{q_i}(p_i o'_i + \sum_{j\in J} p_j p_{ji} o'_{ji})$$
for each $i \in I$, except that Δ_i and o_i are chosen arbitrarily in case $q_i = 0$. It can be checked by arithmetic that q_i, Δ_i, o_i have the required properties, viz. that $\sum_{i\in I} q_i = 1$, that $o = \sum_{i\in I} q_i o_i$ and that
$$\begin{aligned} s &\stackrel{\hat{\tau}}{\Longrightarrow} \sum_{i\in I} p_i \cdot \overline{s} + \sum_{j\in J} p_j \cdot \Delta'_j \\ &\stackrel{\hat{\tau}}{\Longrightarrow} \sum_{i\in I} p_i \cdot \overline{s} + \sum_{j\in J} p_j \cdot \sum_{i\in I} p_{ji} \cdot \Delta'_{ji} \qquad \text{by (6.5) and Lemma 4.1} \\ &= \sum_{i\in I} q_i \cdot \Delta_i \ . \end{aligned}$$

  Finally, it follows from (6.3) and (6.7) that $o_i \in \widehat{\mathcal{A}}^{\Omega}_{\updownarrow}(T_i, \Delta_i)$ for each $i \in I$.

(b) Let $\lceil\Delta\rceil = \{s_j\}_{j\in J}$ and $r_j = \Delta(s_j)$. W.l.o.g. we may assume that J is a non-empty finite set disjoint from I. Using that $\widehat{\mathcal{A}}^{\Omega}_{\updownarrow}(T, \Delta) := \mathrm{Exp}_{\Delta}\widehat{\mathcal{A}}^{\Omega}_{\updownarrow}(T, \_)$, if $o \in \widehat{\mathcal{A}}^{\Omega}_{\updownarrow}(T, \Delta)$ then

$$o = \sum_{j\in J} r_j o'_j \tag{6.8}$$

$$o'_j \in \widehat{\mathcal{A}}^{\Omega}_{\updownarrow}(T, s_j) \tag{6.9}$$

For each $j \in J$, we know by the induction hypothesis that

$$s_j \stackrel{\hat{\tau}}{\Longrightarrow} \sum_{i\in I} q_{ji} \cdot \Delta'_{ji} \tag{6.10}$$

$$o'_j = \sum_{i\in I} q_{ji} o'_{ji} \tag{6.11}$$

$$o'_{ji} \in \widehat{\mathcal{A}}^{\Omega}_{\updownarrow}(T_i, \Delta'_{ji}) \tag{6.12}$$

for some $\{q_{ji}\}_{i\in I}$ with $\sum_{i\in I} q_{ji} = 1$. Thus let

$$q_i = \sum_{j\in J} r_j q_{ji}$$

$$\Delta_i = \frac{1}{q_i} \sum_{j\in J} r_j q_{ji} \cdot \Delta'_{ji}$$

$$o_i = \frac{1}{q_i} \sum_{j\in J} r_j q_{ji} o'_{ji}$$

again choosing Δ_i and o_i arbitrarily in case $q_i = 0$. As in the first case, it can be shown by arithmetic that the collection r_i, Δ_i, o_i has the required properties. $\square$

7. Modal logic

In this section we present logical characterisations $\sqsubseteq^{\mathcal{L}}$ and $\sqsubseteq^{\mathcal{F}}$ of our testing preorders. Besides their intrinsic interest, these logical preorders also serves as a stepping stone in proving Theorem 5.2. In this section we show that the logical preorders are sound w.r.t. the simulation and failure simulation preorders, and hence w.r.t. the testing preorders; in the next section we establish completeness. To start, we define a set $\mathcal{F}$ of modal formulae, inductively, as follows:

- $\mathbf{ref}(X) \in \mathcal{F}$ when $X \subseteq \mathsf{Act}$,
- $\langle a\rangle\varphi \in \mathcal{F}$ when $\varphi \in \mathcal{F}$ and $a \in \mathsf{Act}$,
- $\bigwedge_{i\in I} \varphi_i \in \mathcal{F}$ 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 $\varphi_1 \wedge \varphi_2$ for $\bigwedge_{i\in\{1,2\}} \varphi_i$ and $\top$ for $\bigwedge_{i\in\emptyset} \varphi_i$.

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

- $\Delta \models \mathbf{ref}(X)$ iff there is a Δ' with $\Delta \stackrel{\hat{\tau}}{\Longrightarrow} \Delta'$ and $\Delta' \stackrel{X}{\not\longrightarrow}$,
- $\Delta \models \langle a\rangle\varphi$ iff there is a Δ' with $\Delta \stackrel{\hat{a}}{\Longrightarrow} \Delta'$ and $\Delta' \models \varphi$,
- $\Delta \models \bigwedge_{i\in I} \varphi_i$ 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 $\Delta \stackrel{\hat{\tau}}{\Longrightarrow} \sum_{i\in I} p_i \cdot \Delta_i$.

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.)

In order to obtain the main result of this section, Theorem 7.4, we introduce the following tool.

Definition 7.1. The *$\mathcal{F}$-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. The *$\mathcal{L}$-characteristic formulae* ψ_s and ψ_Δ are defined likewise, but omitting the conjuncts $\mathbf{ref}(\{a \mid s \stackrel{a}{\not\longrightarrow}\})$.

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 $i \in I$; furthermore, the following property can be established by an easy inductive proof.

Lemma 7.2. *For any $\Delta \in \mathcal{D}(\mathsf{sCSP})$ we have $\Delta \models \varphi_\Delta$, as well as $\Delta \models \psi_\Delta$.* □

It and the following lemma help to establish Theorem 7.4.

Lemma 7.3. *For any processes $P, Q \in \mathsf{pCSP}$ we have that $[\![P]\!] \models \varphi_{[\![Q]\!]}$ implies $P \sqsubseteq_{FS} Q$, and likewise that $[\![Q]\!] \models \psi_{[\![P]\!]}$ implies $P \sqsubseteq_S Q$.*

Proof. To establish the first statement, we define the relation $\mathcal{R}$ by $s \,\mathcal{R}\, \Theta$ iff $\Theta \models \varphi_s$; to show that it is a failure simulation we first prove the following technical result:

$$\Theta \models \varphi_\Delta \text{ implies } \exists \Theta' : \Theta \stackrel{\hat{\tau}}{\Longrightarrow} \Theta' \wedge \Delta \,\overline{\mathcal{R}}\, \Theta'. \tag{7.1}$$

Suppose $\Theta \models \varphi_\Delta$ with $\varphi_\Delta = \bigoplus_{i \in I} p_i \cdot \varphi_{s_i}$, so that we have $\Delta = \sum_{i \in I} p_i \cdot \overline{s_i}$ 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 $\Theta' := \sum_{i \in I} p_i \cdot \Theta_i$. 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 from Definition 7.1 we have $\varphi_s \Rrightarrow \varphi_\Delta$, so that $\Theta \models \varphi_\Delta$. Applying (7.1) gives us $\Theta \stackrel{\hat{\tau}}{\Longrightarrow} \Theta'$ with $\Delta \,\overline{\mathcal{R}}\, \Theta'$ for some Θ'.
- Suppose $s \,\mathcal{R}\, \Theta$ and $s \xrightarrow{a} \Delta$ with $a \in \mathsf{Act}$. 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 $\Theta' \models \varphi_\Delta$. Again apply (7.1).
- 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 $\mathcal{R}$ is indeed a failure simulation. By our assumption $[\![P]\!] \models \varphi_{[\![Q]\!]}$, using (7.1), there exists a Θ' such that $[\![P]\!] \stackrel{\hat{\tau}}{\Longrightarrow} \Theta'$ and $[\![Q]\!] \,\overline{\mathcal{R}}\, \Theta'$, which gives $P \sqsubseteq_{FS} Q$ via Definition 4.3.

To establish the second statement, define the relation $\mathcal{S}$ by $s \,\mathcal{S}\, \Theta$ iff $\Theta \models \psi_s$; exactly as above one obtains

$$\Theta \models \psi_\Delta \text{ implies } \exists \Theta' : \Theta \stackrel{\hat{\tau}}{\Longrightarrow} \Theta' \wedge \Delta \,\overline{\mathcal{S}}\, \Theta'. \tag{7.2}$$

Just as above it follows that $\mathcal{S}$ is a simulation. By the assumption $[\![Q]\!] \models \varphi_{[\![P]\!]}$, using (7.2), there exists a Θ' such that $[\![Q]\!] \stackrel{\hat{\tau}}{\Longrightarrow} \Theta'$ and $[\![P]\!] \,\overline{\mathcal{S}}\, \Theta'$. Hence $P \sqsubseteq_S Q$ via Definition 4.3. □

Theorem 7.4.

(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 Lemma 7.2 we have $[\![Q]\!] \models \varphi_{[Q]}$ and hence $[\![P]\!] \models \varphi_{[Q]}$. Lemma 7.3 gives $P \sqsubseteq_{FS} Q$.

For (1), assuming $P \sqsubseteq^{\mathcal{L}} Q$, we have $[\![P]\!] \models \psi_{[P]}$, hence $[\![Q]\!] \models \psi_{[P]}$, and thus $P \sqsubseteq_S Q$. □

8. Characteristic tests

Our final step towards Theorem 5.2 is taken in this section, where we show that every modal formula φ can be characterised by a vector-based test T_φ with the property that any pCSP process satisfies φ just when it passes the test T_φ.

Lemma 8.1. *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*

$$\Delta \models \varphi \quad \textit{iff} \quad \exists o \in \widehat{\mathcal{A}}^{\Omega}_{\updownarrow}(T_\varphi, \Delta) :\ o \leq v_\varphi \tag{8.1}$$

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

$$\Delta \models \varphi \quad \textit{iff} \quad \exists o \in \widehat{\mathcal{A}}^{\Omega}_{\updownarrow}(T_\varphi, \Delta) :\ o \geq v_\varphi\,. \tag{8.2}$$

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 $\varphi = \mathbf{ref}(X)$ 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 $\varphi = \langle a\rangle\psi$. 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 $\varphi = \bigwedge_{i\in I} \varphi_i$ 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 (8.1) above.

- Let $\varphi = \top$. For all $\Delta \in \mathcal{D}(\mathsf{sCSP})$ we have $\Delta \models \varphi$ as well as $\exists o \in \widehat{\mathcal{A}}^{\Omega}_{\updownarrow}(T_\varphi, \Delta) :\ o \leq v_\varphi$, using Lemma 6.7(1).

- 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 Lemma 6.7(2), $\vec{0} \in \widehat{\mathcal{A}}^{\Omega}_{\updownarrow}(T_\varphi, \Delta)$.

  Now suppose $\exists o \in \widehat{\mathcal{A}}^{\Omega}_{\updownarrow}(T_\varphi, \Delta) : \ o \leq v_\varphi$. This implies $o = \vec{0}$, so by Lemma 6.7(2) there is a Δ' with $\Delta \stackrel{\hat{\tau}}{\Longrightarrow} \Delta'$ and $\Delta' \stackrel{X}{\not\longrightarrow}$. Hence $\Delta \models \varphi$.
- 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, $\exists o \in \widehat{\mathcal{A}}^{\Omega}_{\updownarrow}(T_\psi, \Delta') : \ o \leq v_\psi$. By Lemma 6.7(3), $o \in \widehat{\mathcal{A}}^{\Omega}_{\updownarrow}(T_\varphi, \Delta)$.

  Now suppose $\exists o \in \widehat{\mathcal{A}}^{\Omega}_{\updownarrow}(T_\varphi, \Delta) : \ o \leq v_\varphi$. This implies $o(\omega) = 0$, so by Lemma 6.7(3) 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 $\varphi = \bigwedge_{i \in I} \varphi_i$ 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, $\exists o_i \in \widehat{\mathcal{A}}^{\Omega}_{\updownarrow}(T_i, \Delta) : \ o_i \leq v_i$. Thus $o := \sum_{i \in I} p_i o_i \in \widehat{\mathcal{A}}^{\Omega}_{\updownarrow}(T_\varphi, \Delta)$ by Lemma 6.7(4), and $o \leq v_\varphi$.

  Now suppose $\exists o \in \widehat{\mathcal{A}}^{\Omega}_{\updownarrow}(T_\varphi, \Delta) : \ o \leq v_\varphi$. Then, using Lemma 6.7(4), $o = \sum_{i \in I} p_i o_i$ for certain $o_i \in \widehat{\mathcal{A}}^{\Omega}_{\updownarrow}(T_i, \Delta)$. Note that $(T_i)_{i \in I}$ is an Ω-disjoint family of tests. One has $o_i \leq v_i$ 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 $j \neq i$. This implies $v_j(\omega) = 0$ for all $j \neq i$ 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 $\Delta \stackrel{\hat{\tau}}{\Longrightarrow} \sum_{i \in I} p_i \cdot \Delta_i$. By induction, there are $o_i \in \widehat{\mathcal{A}}^{\Omega}_{\updownarrow}(T_i, \Delta_i)$ with $o_i \leq v_i$. Hence, there are $o'_i \in \widehat{\mathcal{A}}^{\Omega}_{\updownarrow}(T'_i, \Delta_i)$ with $o'_i \leq v'_i$. Thus $o := \sum_{i \in I} p_i o'_i \in \widehat{\mathcal{A}}^{\Omega}_{\updownarrow}(T_\varphi, \Delta)$ by Lemma 6.7(5), and $o \leq v_\varphi$.

  Now suppose $\exists o \in \widehat{\mathcal{A}}^{\Omega}_{\updownarrow}(T_\varphi, \Delta) : \ o \leq v_\varphi$. Then, by Lemma 6.8, there are $q \in \mathcal{D}(I)$ and Δ_i, for $i \in I$, such that $\Delta \stackrel{\hat{\tau}}{\Longrightarrow} \sum_{i \in I} q_i \cdot \Delta_i$ and $o = \sum_{i \in I} q_i o'_i$ for some $o'_i \in \widehat{\mathcal{A}}^{\Omega}_{\updownarrow}(T'_i, \Delta_i)$. Now $\forall i : o'_i(\omega_i) = v'_i(\omega_i) = \frac{1}{2}$, so, using that $(T_i)_{i \in I}$ is an Ω-disjoint family of tests, $\frac{1}{2} q_i = q_i o'_i(\omega_i) = o(\omega_i) \leq v_\varphi(\omega_i) = p_i v'_i(\omega_i) = \frac{1}{2} p_i$. 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 $o'_i \leq v'_i$ for all $i \in I$. Given that $T'_i = T_i \,{}_{\frac{1}{2}}\!\oplus \omega_i$, using Lemma 6.7(4), it must be that $o' = \frac{1}{2} o_i + \frac{1}{2} \vec{\omega_i}$ for some $o_i \in \widehat{\mathcal{A}}^{\Omega}_{\updownarrow}(T_i, \Delta_i)$ with $o_i \leq v_i$. By induction, $\Delta_i \models \varphi_i$ for all $i \in I$, and hence $\Delta \models \varphi$.

In case $\varphi \in \mathcal{L}$, the formula cannot be of the form $\mathbf{ref}(X)$. Then a straightforward induction yields that $\sum_{\omega \in \Omega} v_\varphi(\omega) = 1$ and for all $\Delta \in \mathcal{D}(\mathsf{pCSP})$ and $o \in \widehat{\mathcal{A}}^{\Omega}_{\updownarrow}(T_\varphi, \Delta)$ we have $\sum_{\omega \in \Omega} o(\omega) = 1$. Therefore, $o \leq v_\varphi$ iff $o \geq v_\varphi$ iff $o = v_\varphi$, yielding (8.2). □

Theorem 8.2.

(1) If $P \mathrel{\widehat{\sqsubseteq}^{\Omega}_{\mathrm{pmay}}} Q$ then $P \sqsubseteq^{\mathcal{L}} Q$.

(2) If $P \mathrel{\widehat{\sqsubseteq}^{\Omega}_{\mathrm{pmust}}} Q$ then $P \sqsubseteq^{\mathcal{F}} Q$.

Proof. Suppose $P \mathrel{\widehat{\sqsubseteq}^{\Omega}_{\mathrm{pmust}}} Q$ and $[\![Q]\!] \models \varphi$ for some $\varphi \in \mathcal{F}$. Let T_φ be a characteristic test of φ with target value v_φ. Then Lemma 8.1 yields $\exists o \in \widehat{\mathcal{A}}^{\Omega}_{\updownarrow}(T_\varphi, [\![Q]\!]) : o \leq v_\varphi$, and hence, given that $P \mathrel{\widehat{\sqsubseteq}^{\Omega}_{\mathrm{pmust}}} Q$ and $\widehat{\mathcal{A}}^{\Omega}_{\updownarrow}(T_\varphi, [\![R]\!]) = \widehat{\mathcal{A}}^{\Omega}_{\updownarrow}(T_\varphi, R)$ for any $R \in \mathsf{pCSP}$, by the Smyth preorder 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, via the Hoare preorder. □

Combining Theorems 6.6, 8.2 and 7.4, we obtain Theorem 5.2, the goal we set ourselves in Section 5. Thus, with Theorems 4.7 and 4.11 and Proposition 5.1, we have shown that the

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

Figure 4: Common equations

may preorder coincides with simulation and that the must preorder coincides with failure simulation. These results also imply the converse of both statements in Theorem 8.2, and thus that the logics $\mathcal{L}$ and $\mathcal{F}$ give logical characterisations of the simulation and failure simulation preorders $\sqsubseteq_S$ and $\sqsubseteq_{FS}$. □

9. Equational theories

Having settled the problem of characterising the may preorder in terms of simulation, and the must preorder in terms of failure simulation, we now turn to complete axiomatisations of the preorders.

In order to focus on the essentials we consider just those pCSP processes that do not use the parallel operator $|_A$; we call the resulting sub-language nCSP. For a brief discussion of the axiomatisation for terms involving $|_A$ and the other parallel operators commonly used in CSP see Section 12.

Let us write $P =_E Q$ for equivalences that can be derived using the equations given in Figure 4. Given the way we defined the syntax of pCSP, axiom (**D1**) is merely a case of abbreviation-expansion; thanks to (**D1**) there is no need for (meta-)variables ranging over the sub-sort of state-based processes anywhere in the axioms. Many of the standard equations for CSP [17] are missing; they are not sound for $\simeq_{FS}$. Typical examples include:

$$
\begin{aligned}
a.(P \sqcap Q) &= a.P \sqcap a.Q \\
P &= P \mathrel{\Box} P \\
P \mathrel{\Box} (Q \sqcap R) &= (P \mathrel{\Box} Q) \sqcap (P \mathrel{\Box} R) \\
P \sqcap (Q \mathrel{\Box} R) &= (P \sqcap Q) \mathrel{\Box} (P \sqcap R)
\end{aligned}
$$

For a detailed discussion of the standard equations for CSP in the presence of probabilistic processes see Section 4 of [8].

Proposition 9.1. *Suppose $P =_E Q$. Then $P \simeq_{FS} Q$.*

May:

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

Must:

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

Figure 5: Inequations

Proof. Because of Proposition 4.6, that $\sqsubseteq_{FS}$ is a precongruence, it is sufficient to exhibit witness failure simulations for the axioms in Figure 4. These are exactly the same as the witness simulations for the same axioms, given in [8]. The only axiom for which it is nontrivial to check that these simulations are in fact failure simulations is (**EI**). That axiom, as stated in [8], is unsound here; it will return in the next section as (**May0**). But the special case of $a = b$ yields the axiom (**EI**) above, and then the witness simulation from [8] is a failure simulation indeed. □

As $\simeq_S$ is a less discriminating equivalence than $\simeq_{FS}$ it follows that $P =_E Q$ implies $P \simeq_S Q$.

This equational theory allows us to reduce terms to a form in which the external choice operator is applied to prefix terms only.

Definition 9.2 (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 9.3. *For every $P \in \mathsf{nCSP}$ there is a normal form N such that $P =_E N$.*

Proof. A fairly straightforward induction, heavily relying on (**D1**)–(**D3**). □

We can also show that the axioms (**P1**)–(**P3**) and (**D1**) are in some sense all that are required to reason about probabilistic choice. Let $P =_{\text{prob}} Q$ denote that equivalence of P and Q can be derived using those axioms alone. Then we have the following property.

Lemma 9.4. *Let $P, Q \in \mathsf{nCSP}$. Then $[\![P]\!] = [\![Q]\!]$ implies $P =_{\text{prob}} Q$.*

Here $[\![P]\!] = [\![Q]\!]$ says that $[\![P]\!]$ and $[\![Q]\!]$ are the very same distributions of state-based processes in sCSP; this is a much stronger prerequisite than P and Q being testing equivalent.

Proof. The axioms (**P1**)–(**P3**) and (**D1**) essentially allow any processes to be written in the unique form $\bigoplus_{i \in I} p_i s_i$, where the $s_i \in \mathsf{sCSP}$ are all different. □

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 Figure 5 to the equations in Figure 4. The first three additions, (**May0**)–(**May2**), are used in the standard testing theory of CSP [17, 6, 15]. For the *must* case, in addition to the standard inequation (**Must1**), we require an inequational schema, (**Must2**); this uses the notation $\mathit{inits}(P)$ to denote the (finite) set of initial actions of P. Formally,

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

The axiom (**Must2**) can equivalently be formulated as follows:

$$\bigoplus_{k\in K} \mathop{\Box}_{\ell\in L_k} a_{k\ell}.R_{k\ell} \sqcap \bigsqcap_{i\in I} \bigoplus_{j\in J_i} p_j\cdot(a_i.Q_{ij} \,\Box\, P_{ij}) \;\sqsubseteq\; \mathop{\Box}_{i\in I} a_i.\bigoplus_{j\in J_i} p_j\cdot Q_{ij},$$

$$\text{provided} \qquad \{a_{k\ell} \mid k\in K, \ell\in K_k\} \subseteq \{a_i \mid i\in I\}\,.$$

This is the case because a term R satisfies $\mathit{inits}(R) \subseteq \{a_i\}_{i\in I}$ iff it can be converted into the form $\bigoplus_{k\in K} \mathop{\Box}_{\ell\in L_k} a_{k\ell}.R_{k\ell}$ by means of axioms (**D1**), (**P1**)–(**P3**) and (**E1**)–(**E3**) of Figure 5. This axiom can also be reformulated in an equivalent but more semantic style:

$$(\textbf{Must2}') \qquad R \sqcap \textstyle\bigsqcap_{i\in I} P_i \;\sqsubseteq\; \Box_{i\in I}\, a_i.Q_i,$$

$$\text{provided } [\![P_i]\!] \xrightarrow{a_i} [\![Q_i]\!] \text{ and } [\![R]\!] \stackrel{X}{\not\longrightarrow} \text{ with } X = \mathsf{Act}\backslash\{a_i\}_{i\in I}.$$

This is the case because $[\![P]\!] \xrightarrow{a} [\![Q]\!]$ iff, up to the axioms in Figure 4, P has the form $\bigoplus_{j\in J} p_j\cdot(a.Q_j \,\Box\, P_j)$ and Q has the form $a.\bigoplus_{j\in J} p_j\cdot Q_j$ for certain P_j, Q_j and p_j, for $j\in J$.

Note that (**Must2**) can be used, together with (**I1**), to derive the dual of (**May3**) via the following inference:

$$\begin{aligned}
a.P \,{}_p\!\oplus a.Q \;&=_E\; (a.P \,{}_p\!\oplus a.Q) \sqcap (a.P \,{}_p\!\oplus a.Q) \\
&\sqsubseteq_{E_{\text{must}}}\; a.(P \,{}_p\!\oplus Q)
\end{aligned}$$

where 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 (**May1**) and (**P1**) is

$$(\textbf{May4}) \quad P \,{}_p\!\oplus Q \;\sqsubseteq_{E_{\text{may}}}\; P \sqcap Q$$

saying that any probabilistic choice can be simulated by an internal choice. It is derived as follows:

$$\begin{aligned}
P \,{}_p\!\oplus Q \;&\sqsubseteq_{E_{\text{may}}}\; (P \sqcap Q) \,{}_p\!\oplus (P \sqcap Q) \\
&=_E\; (P \sqcap Q)
\end{aligned}$$

Likewise, we have

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

Theorem 10.1. *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. For one direction it is sufficient to check that the inequations, and the inequational schema in Figure 5 are sound. For $\sqsubseteq_S$ this has been done in [8], and the soundness of (**Must1**) and (**Must2′**) for $\sqsubseteq_{FS}$ is trivial. The converse, completeness, is established in the next section. □

11. Completeness

The completeness proof of Theorem 10.1 depends on the following variation on the *Derivative lemma* of [30]:

Lemma 11.1 (Derivative lemma)**.** *Let $P, Q \in \mathsf{nCSP}$.*

(i) If $[\![P]\!] \stackrel{\hat{\tau}}{\Longrightarrow} [\![Q]\!]$ then $P \sqsubseteq_{E_{\text{must}}} Q$ and $Q \sqsubseteq_{E_{\text{may}}} P$.
(ii) If $[\![P]\!] \stackrel{a}{\Longrightarrow} [\![Q]\!]$ then $a.Q \sqsubseteq_{E_{\text{may}}} P$.

Proof. The proof of (i) proceeds in four stages. We only deal with $\sqsubseteq_{E_{\text{may}}}$, as the proof for $\sqsubseteq_{E_{\text{must}}}$ is entirely analogous.

First we show by structural induction on $s \in \mathsf{sCSP} \cap \mathsf{nCSP}$ that $s \xrightarrow{\tau} [\![Q]\!]$ implies $Q \sqsubseteq_{E_{\text{may}}} s$. So suppose $s \xrightarrow{\tau} [\![Q]\!]$. In case s has the form $P_1 \sqcap P_2$ it follows by the operational semantics of pCSP that $Q = P_1$ or $Q = P_2$. Hence $Q \sqsubseteq_{E_{\text{may}}} s$ by (**May1**). The only other possibility is that s has the form $s_1 \Box s_2$. In that case there must be a distribution Δ such that either $s_1 \xrightarrow{\tau} \Delta$ and $[\![Q]\!] = \Delta \Box s_2$, or $s_2 \xrightarrow{\tau} \Delta$ and $[\![Q]\!] = s_1 \Box \Delta$. Using symmetry, we may restrict attention to the first case. Let R be a term such that $[\![R]\!] = \Delta$. Then $[\![R \Box s_2]\!] = \Delta \Box s_2 = [\![Q]\!]$, so Lemma 9.4 yields $Q =_{\text{prob}} R \Box s_2$. By induction we have $R \sqsubseteq_{E_{\text{may}}} s_1$, hence $R \Box s_2 \sqsubseteq_{E_{\text{may}}} s_1 \Box s_2$, and thus $Q \sqsubseteq_{E_{\text{may}}} s$.

Now we show that $s \xrightarrow{\hat{\tau}} [\![Q]\!]$ implies $Q \sqsubseteq_{E_{\text{may}}} s$. This follows because $s \xrightarrow{\hat{\tau}} [\![Q]\!]$ means that either $s \xrightarrow{\tau} [\![Q]\!]$ or $[\![Q]\!] = \overline{s}$, and in the latter case Lemma 9.4 yields $Q =_{\text{prob}} s$.

Next we show that $[\![P]\!] \xrightarrow{\hat{\tau}} [\![Q]\!]$ implies $Q \sqsubseteq_{E_{\text{may}}} P$. So suppose $[\![P]\!] \xrightarrow{\hat{\tau}} [\![Q]\!]$, that is

$$[\![P]\!] = \sum_{i \in I} p_i \cdot \overline{s_i} \qquad s_i \xrightarrow{\hat{\tau}} [\![Q_i]\!] \qquad [\![Q]\!] = \sum_{i \in I} p_i \cdot [\![Q_i]\!]$$

for some I, $p_i \in (0,1]$, $s_i \in \mathsf{sCSP} \cap \mathsf{nCSP}$ and $Q_i \in \mathsf{nCSP}$. Now

(1) $[\![P]\!] = [\![\bigoplus_{i \in I} p_i {\cdot} s_i]\!]$. By Lemma 9.4 we have $P =_{\text{prob}} \bigoplus_{i \in I} p_i {\cdot} s_i$.
(2) $[\![Q]\!] = [\![\bigoplus_{i \in I} p_i {\cdot} Q_i]\!]$. Again Lemma 9.4 yields $Q =_{\text{prob}} \bigoplus_{i \in I} p_i {\cdot} Q_i$.
(3) $s_i \xrightarrow{\hat{\tau}} [\![Q_i]\!]$ implies $Q_i \sqsubseteq_{E_{\text{may}}} s_i$. Therefore, $\bigoplus_{i \in I} p_i {\cdot} Q_i \sqsubseteq_{E_{\text{may}}} \bigoplus_{i \in I} p_i {\cdot} s_i$.

Combining (1), (2) and (3) we obtain $Q \sqsubseteq_{E_{\text{may}}} P$.

Finally, the general case, when $[\![P]\!] \xrightarrow{\hat{\tau}}{}^{*} \Delta$, is now a simple inductive argument on the length of the derivation.

The proof of (ii) is similar: first we treat the case when $s \xrightarrow{a} [\![Q]\!]$ by structural induction, using (**May2**); then the case $[\![P]\!] \xrightarrow{a} [\![Q]\!]$, exactly as above; and finally use part (i) to derive the general case. □

The completeness result now follows from the following two propositions.

Proposition 11.2. *Let P and Q be in nCSP. Then $P \sqsubseteq_S Q$ implies $P \sqsubseteq_{E_{\text{may}}} Q$.*

Proof. The proof is by structural induction on P and Q, and we may assume that both P and Q are in normal form because of Proposition 9.3. So take $P, Q \in \mathsf{pCSP}$ and suppose the claim has been established for all subterms P' of P and Q' of Q, of which at least one of the two is a strict subterm. We start by proving that if $P \in \mathsf{sCSP}$ then we have

$$P \lhd_S [\![Q]\!] \quad \text{implies} \quad P \sqsubseteq_{E_{\mathrm{may}}} Q. \tag{11.1}$$

There are two cases to consider.

(1) P has the form $P_1 \sqcap P_2$. Since $P_i \sqsubseteq_{E_{\mathrm{may}}} P$ we know $P_i \sqsubseteq_S P \sqsubseteq_S Q$. We use induction to obtain $P_i \sqsubseteq_{E_{\mathrm{may}}} Q$, from which the result follows using (**I1**).

(2) P has the form $\Box_{i\in I}\, a_i.P_i$. If I contains two or more elements then P may also be written as $\bigsqcap_{i\in I} a_i.P_i$, using (**May0**) and (**D2**), and we may proceed as in case (1) above. If I is empty, that is P is $\mathbf{0}$, then we can use (**May2**). So we are left with the possibility that P is $a.P'$. Thus suppose that $a.P' \lhd_S [\![Q]\!]$. We proceed by a case analysis on the structure of Q.

- Q is $a.Q'$. We know from $a.P' \lhd_S [\![a.Q']\!]$ that $[\![P']\!] \overline{\lhd_S}\, \Theta$ for some Θ with $[\![Q']\!] \stackrel{\hat{\tau}}{\Longrightarrow} \Theta$, thus $P' \sqsubseteq_S Q'$. Therefore, we have $P' \sqsubseteq_{E_{\mathrm{may}}} Q'$ by induction. It follows that $a.P' \sqsubseteq_{E_{\mathrm{may}}} a.Q'$.
- Q is $\Box_{j\in I}\, a_j.Q_j$ with at least two elements in J. We use (**May0**) and then proceed as in the next case.
- Q is $Q_1 \sqcap Q_2$. We know from $a.P' \lhd_S [\![Q_1 \sqcap Q_2]\!]$ that $[\![P']\!] \overline{\lhd_S}\, \Theta$ for some Θ such that one of the following two conditions holds
  (a) $[\![Q_i]\!] \stackrel{a}{\Longrightarrow} \Theta$ for $i = 1$ or 2. In this case, $a.P' \lhd_S [\![Q_i]\!]$, hence $a.P' \sqsubseteq_S Q_i$. By induction we have $a.P' \sqsubseteq_{E_{\mathrm{may}}} Q_i$; then we apply (**May1**).
  (b) $[\![Q_1]\!] \stackrel{a}{\Longrightarrow} \Theta_1$ and $[\![Q_2]\!] \stackrel{a}{\Longrightarrow} \Theta_2$ such that $\Theta = p\cdot\Theta_1 + (1-p)\cdot\Theta_2$ for some $p \in (0,1)$. Let $\Theta_i = [\![Q_i']\!]$ for $i = 1, 2$. By the Derivative Lemma, we have $a.Q_1' \sqsubseteq_{E_{\mathrm{may}}} Q_1$ and $a.Q_2' \sqsubseteq_{E_{\mathrm{may}}} Q_2$. Clearly, $[\![Q_1' \,{}_p\!\oplus Q_2']\!] = \Theta$, thus $P' \sqsubseteq_S Q_1' \,{}_p\!\oplus Q_2'$. By induction, we infer that $P' \sqsubseteq_{E_{\mathrm{may}}} Q_1' \,{}_p\!\oplus Q_2'$. So

$$\begin{array}{lll} a.P' & \sqsubseteq_{E_{\mathrm{may}}} a.(Q_1' \,{}_p\!\oplus Q_2') & \\ & \sqsubseteq_{E_{\mathrm{may}}} a.Q_1' \,{}_p\!\oplus a.Q_2' & (\mathbf{May3}) \\ & \sqsubseteq_{E_{\mathrm{may}}} Q_1 \,{}_p\!\oplus Q_2 & \\ & \sqsubseteq_{E_{\mathrm{may}}} Q_1 \sqcap Q_2 & (\mathbf{May4}) \end{array}$$

- Q is $Q_1 \,{}_p\!\oplus Q_2$. We know from $a.P' \lhd_S [\![Q_1 \,{}_p\!\oplus Q_2]\!]$ that $[\![P']\!] \overline{\lhd_S}\, \Theta$ for some Θ such that $[\![Q_1 \,{}_p\!\oplus Q_2]\!] \stackrel{a}{\Longrightarrow} \Theta$. From Lemma 4.1 we know that Θ must take the form $p\cdot[\![Q_1']\!] + (1-p)\cdot[\![Q_2']\!]$, where $[\![Q_i]\!] \stackrel{a}{\Longrightarrow} [\![Q_i']\!]$ for $i = 1, 2$. Hence $P' \sqsubseteq_S Q_1' \,{}_p\!\oplus Q_2'$, and by induction we get $P' \sqsubseteq_{E_{\mathrm{may}}} Q_1' \,{}_p\!\oplus Q_2'$. Then we can derive $a.P' \sqsubseteq_{E_{\mathrm{may}}} Q_1 \,{}_p\!\oplus Q_2$ as in the previous case.

Now we use (11.1) to show that $P \sqsubseteq_S Q$ implies $P \sqsubseteq_{E_{\mathrm{may}}} Q$. Suppose $P \sqsubseteq_S Q$. Applying Definition 4.3 with the understanding that any distribution $\Theta \in \mathcal{D}(\mathsf{sCSP})$ can be written as $[\![Q']\!]$ for some $Q' \in \mathsf{pCSP}$, this means that $[\![P]\!] \overline{\lhd_S} [\![Q']\!]$ for some $[\![Q]\!] \stackrel{\hat{\tau}}{\Longrightarrow} [\![Q']\!]$. The Derivative Lemma yields $Q' \sqsubseteq_{E_{\mathrm{may}}} Q$. So it suffices to show $P \sqsubseteq_{E_{\mathrm{may}}} Q'$. We know that $[\![P]\!] \overline{\lhd_S} [\![Q']\!]$ means that

$$[\![P]\!] = \sum_{k\in K} r_k \cdot \overline{t_k} \qquad t_k \lhd_S [\![Q_k']\!] \qquad [\![Q']\!] = \sum_{k\in K} r_k \cdot [\![Q_k']\!]$$

for some K, $r_k \in (0,1]$, $t_k \in \mathsf{sCSP}$ and $Q_k' \in \mathsf{pCSP}$. Now

(1) $[\![P]\!] = [\![\bigoplus_{k\in K} r_k{\cdot}t_k]\!]$. By Lemma 9.4 we have $P =_{\mathrm{prob}} \bigoplus_{k\in K} r_k{\cdot}t_k$.
(2) $[\![Q']\!] = [\![\bigoplus_{k\in K} r_k{\cdot}Q_k']\!]$. Again Lemma 9.4 yields $Q' =_{\mathrm{prob}} \bigoplus_{k\in K} r_k{\cdot}Q_k'$.
(3) $t_k \lhd_S [\![Q_k']\!]$ implies $t_k \sqsubseteq_{E_{\mathrm{may}}} Q_k'$ by (11.1). Therefore, $\bigoplus_{k\in K} r_k{\cdot}t_k \sqsubseteq_{E_{\mathrm{may}}} \bigoplus_{k\in K} r_k{\cdot}Q_k'$.

Combining (1), (2) and (3) we obtain $P \sqsubseteq_{E_{\mathrm{may}}} Q'$, hence $P \sqsubseteq_{E_{\mathrm{may}}} Q$. $\square$

Proposition 11.3. *Let P and Q be in* nCSP*. Then $P \sqsubseteq_{FS} Q$ implies $P \sqsubseteq_{E_{\mathrm{must}}} Q$.*

Proof. Similar to the proof of Proposition 11.2, but using a reversed orientation of the preorders. The only real difference is the case (2), which we consider now. So assume $Q \lhd_{FS} [\![P]\!]$, where Q has the form $\Box_{i\in I}\, a_i.Q_i$. Let X be any set of actions such that $X \cap \{a_i\}_{i\in I} = \emptyset$; then $\Box_{i\in I}\, a_i.Q_i \xrightarrow{X}\!\!\!\!\!\!/\;\,$. Therefore, there exists a P' such that $[\![P]\!] \stackrel{\hat{\tau}}{\Longrightarrow} [\![P']\!] \xrightarrow{X}\!\!\!\!\!\!/\;\,$. By the Derivative lemma,

$$P \sqsubseteq_{E_{\mathrm{must}}} P' \tag{11.2}$$

Since $\Box_{i\in I}\, a_i.Q_i \xrightarrow{a_i} [\![Q_i]\!]$, there exist P_i, P_i', P_i'' such that $[\![P]\!] \stackrel{\hat{\tau}}{\Longrightarrow} [\![Pi]\!] \xrightarrow{a_i} [\![P_i']\!] \stackrel{\hat{\tau}}{\Longrightarrow} [\![P_i'']\!]$ and $[\![Q_i]\!] \;\overline{\lhd_{FS}}\; [\![P_i'']\!]$. Now

$$P \sqsubseteq_{E_{\mathrm{must}}} P_i \tag{11.3}$$

using the Derivative lemma, and $P_i' \sqsubseteq_{FS} Q_i$, by Definition 4.3. By induction, we have $P_i' \sqsubseteq_{E_{\mathrm{must}}} Q_i$, hence

$$\Box_{i\in I} a_i.P_i' \sqsubseteq_{E_{\mathrm{must}}} \Box_{i\in I} a_i.Q_i \tag{11.4}$$

The desired result is now obtained as follows:

$$\begin{array}{rlll} P & \sqsubseteq_{E_{\mathrm{must}}} & P' \sqcap \bigsqcap_{i\in I} P_i & \text{by } (\mathbf{I1}),\ (11.2) \text{ and } (11.3) \\ & \sqsubseteq_{E_{\mathrm{must}}} & \Box_{i\in I} a_i.P_i' & \text{by } (\mathbf{Must2'}) \\ & \sqsubseteq_{E_{\mathrm{must}}} & \Box_{i\in I} a_i.Q_i & \text{by } (11.4) \end{array}$$

$\square$

Propositions 11.2 and 11.3 give us the completeness result stated in Theorem 10.1.

12. Conclusions and related work

In this paper we continued our previous work [8, 10] 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 [28], Seidel [39] and Morgan et al. [32]; 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 [32] for example applied the Jones/Plotkin probabilistic powerdomain [19] 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 [31] by presenting a denotational model in which internal choice distributed outwards through probabilistic choice. However, the distributivities of both [32] and [31] constitute identifications that cannot be justified by our testing approach; see [8].

In Jou and Smolka [24], as in [28, 39], probabilistic equivalences based on traces, failures and readies are defined. These equivalences are coarser than $\simeq_{\rm pmay}$. For example, the two processes in Example 2.2 cannot be distinguished by the equivalences of [24, 28, 39]. However, we can tell them apart by the test given in Example 3.3.

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, that is there is no occurrence of probabilistic choice in a test [27, 4, 20, 26, 12]; or (ii) a test can be probabilistic, that is probabilistic choice may occur in tests as well as processes [5, 41, 33, 22, 37, 23, 3]. This paper adopts the second approach.

Some work [27, 4, 5, 33] 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* [33]. Cazorla et al. [3] extended the results of [33] with nondeterminism, but suffered from the same problems as [32].

The work most closely related to ours is [22, 23]. In [22] Jonsson and Wang characterised may- and must-testing preorders in terms of "chains" of traces and failures, respectively, and in [23] they presented a "substantially improved" characterisation of their may-testing preorder using a notion of simulation which is weaker than $\sqsubseteq_S$ (cf. Definition 4.3). 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 [37] 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 $\hat{\sqsubseteq}^{\Omega}_{\rm pmay}$ (cf. Definition 6.1) and that the latter coincides with an infinitary version of $\hat{\sqsubseteq}^{\Omega}_{\rm pmust}$. In [29] 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].

Appendix A. Resolution-based testing

A *probabilistic automaton* consists of a pLTS $\langle S, L, \rightarrow\rangle$ and a distribution Δ° over S. Since we only consider probabilistic automata with $L = \mathsf{Act}_\tau \cup \Omega$, we omit it and write a probabilistic automaton simply as a triple $\langle S, \Delta^\circ, \rightarrow\rangle$ and call Δ° the *initial distribution* of the automaton. The operational semantics of a pCSP^Ω process P can thus be viewed as a probabilistic automaton with initial distribution $\Delta^\circ := [\![P]\!]$. States in a probabilistic automata that are not reachable from the initial distribution are generally considered irrelevant and can be omitted.

A probabilistic automaton is called *finite* if there exists a function $depth : S \cup \mathcal{D}(S) \to \mathbb{N}$ such that $s \in \lceil \Delta \rceil$ implies $depth(s) < depth(\Delta)$ and $s \xrightarrow{\alpha} \Delta$ implies $depth(s) > depth(\Delta)$. Finite probabilistic automata can be drawn as explained at the end of Section 2.

A *fully probabilistic automaton* is one in which each state enables at most one action, and (general) probabilistic automata can be "resolved" into fully probabilistic automata by pruning away multiple action-choices until only single choices are left, possibly introducing some linear combinations in the process. We define this formally for probabilistic automata representing pCSP^Ω expressions.

Definition A.1. [10] A *resolution* of a distribution $\Delta^\circ \in \mathcal{D}(\mathsf{sCSP}^\Omega)$ is a fully probabilistic automaton $\langle R, \Theta^\circ, \to \rangle$ such that there is a resolving function $f : R \to \mathsf{sCSP}^\Omega$ which satisfies:

(i) $f(\Theta^\circ) = \Delta^\circ$
(ii) if $r \xrightarrow{\alpha} \Theta$ then $f(r) \xrightarrow{\alpha} f(\Theta)$
(iii) if $r \not\to$ then $f(r) \not\to$

where $f(\Theta)$ is the distribution defined by $f(\Theta)(s) := \sum_{f(r)=s} \Theta(r)$.

Note that resolutions of distributions $\Delta^\circ \in \mathcal{D}(\mathsf{sCSP}^\Omega)$ are always finite. We define a function which yields the probability that a given fully probabilistic automaton will start with a particular sequence of actions.

Definition A.2. [10] Given a fully probabilistic automaton $\mathrm{R} = \langle R, \Delta^\circ, \to \rangle$, the probability that R follows the sequence of actions $\sigma \in \Sigma^*$ from its initial distribution is given by $\mathrm{Pr}_{\mathrm{R}}(\sigma, \Delta^\circ)$, where $\mathrm{Pr}_{\mathrm{R}} : \Sigma^* \times R \to [0,1]$ is defined inductively by

$$\mathrm{Pr}_{\mathrm{R}}(\varepsilon, r) := 1 \quad \text{and} \quad \mathrm{Pr}_{\mathrm{R}}(\alpha\sigma, r) := \begin{cases} \mathrm{Pr}_{\mathrm{R}}(\sigma, \Delta) & \text{if } r \xrightarrow{\alpha} \Delta \\ 0 & \text{otherwise} \end{cases}$$

and $\mathrm{Pr}_{\mathrm{R}}(\sigma, \Delta) := \mathrm{Exp}_\Delta(\mathrm{Pr}_{\mathrm{R}}(\sigma, \_)) = \sum_{r \in \lceil \Delta \rceil} \Delta(r) \cdot \mathrm{Pr}_{\mathrm{R}}(\sigma, r)$. Here ε denotes the empty sequence of actions and $\alpha\sigma$ the sequence starting with $\alpha \in \Sigma$ and continuing with $\sigma \in \Sigma^*$. The value $\mathrm{Pr}_{\mathrm{R}}(\sigma, r)$ is the probability that R proceeds with sequence σ from state r.

Now let $\Sigma^{*\alpha}$ be the set of finite sequences in Σ^* that contain α exactly once, and that at the end. Then the probability that the fully probabilistic automaton R ever performs an action α is given by $\sum_{\sigma \in \Sigma^{*\alpha}} \mathrm{Pr}_{\mathrm{R}}(\sigma, \Delta^\circ)$.

We recall the results-gathering function $\mathbb{W}$ given in Definition 5 of [10].

Definition A.3. For a fully probabilistic automaton R, let its *success tuple* $\mathbb{W}(\mathrm{R}) \in [0,1]^\Omega$ be such that $\mathbb{W}(\mathrm{R})(\omega)$ is the probability that R ever performs the action ω.

Then for a distribution $\Delta^\circ \in \mathcal{D}(\mathsf{sCSP}^\Omega)$ we define the *set* of its success tuples to be those resulting as above from all its resolutions separately:

$$\mathbb{W}(\Delta^\circ) \ := \ \{\mathbb{W}(\mathrm{R}) \mid \mathrm{R} \text{ is a resolution of } \Delta^\circ\}.$$

We relate these sets of tuples to Definition 6.1, in which similar sets are produced "all at once," that is without introducing resolutions first. In fact we will find that they are the same. Note that Definition 6.1 of $\widehat{\mathbb{V}}^\Omega_\updownarrow$ extends smoothly to states and distributions in probabilistic automata. When applied to fully probabilistic automata, $\widehat{\mathbb{V}}^\Omega_\updownarrow$ always yields singleton sets, which we will loosely identify with their unique members; thus when we write $\widehat{\mathbb{V}}^\Omega_\updownarrow(\Delta)(\omega)$ with Δ a distribution in a fully probabilistic automaton, we actually mean the ω-component of the unique element of $\widehat{\mathbb{V}}^\Omega_\updownarrow(\Delta)$.

Lemma A.4. *If* $\mathrm{R} = \langle R, \Delta^\circ, \rightarrow\rangle$ *is a finite fully probabilistic automaton, then*
(1) $\widehat{\mathbb{V}}^\Omega(\Delta) = \widehat{\mathbb{V}}^\Omega_\updownarrow(\Delta)$ *for all* $\Delta \in \mathcal{D}(R)$*, and*
(2) $\mathbb{W}(\mathrm{R}) = \widehat{\mathbb{V}}^\Omega(\Delta^\circ)$.

Proof. (1) is immediate: since the automaton is fully probabilistic, convex closure has no effect. For (2) we need to show that for all $\omega \in \Omega$ we have $\mathbb{W}(\mathrm{R})(\omega) = \widehat{\mathbb{V}}^\Omega(\Delta^\circ)(\omega)$, i.e. that $\sum_{\sigma\in\Sigma^*\omega} \mathrm{Pr}_\mathrm{R}(\sigma, \Delta^\circ) = (\widehat{\mathbb{V}}^\Omega(\Delta^\circ))(\omega)$. So let $\omega \in \Omega$. We show

$$\sum_{\sigma\in\Sigma^*\omega} \mathrm{Pr}_\mathrm{R}(\sigma, \Delta) = \widehat{\mathbb{V}}^\Omega(\Delta)(\omega) \qquad \text{and} \qquad \sum_{\sigma\in\Sigma^*\omega} \mathrm{Pr}_\mathrm{R}(\sigma, r) = \widehat{\mathbb{V}}^\Omega(r)(\omega) \tag{A.1}$$

for all $\Delta \in \mathcal{D}(R)$ and $r \in R$, by simultaneous induction on the depths of Δ and r.

- In the base case r has no enabled actions. Then $\forall i : \sum_{\sigma\in\Sigma^*\omega} \mathrm{Pr}_\mathrm{R}(\sigma, r) = 0$ and $\widehat{\mathbb{V}}^\Omega(r) = \vec{0}$, so $\widehat{\mathbb{V}}^\Omega(r)(\omega) = 0$.
- Now suppose there is a transition $r \xrightarrow{\alpha} \Delta$ for some action α and distribution Δ. There are two possibilities:
  - $\alpha = \omega$. We then have $\widehat{\mathbb{V}}^\Omega(s)(\omega) = 1$. Now for any finite non-empty sequence σ without any occurrence of ω we have $\mathrm{Pr}_\mathrm{R}(\sigma\omega, r) = 0$. Thus $\sum_{\sigma\in\Sigma^*\omega} \mathrm{Pr}_\mathrm{R}(\sigma, r) = \mathrm{Pr}_\mathrm{R}(\omega, r) = 1$ as required.
  - $\alpha \neq \omega$. Since $\widehat{\mathbb{V}}^\Omega(r) = \alpha!\widehat{\mathbb{V}}^\Omega(\Delta)$, we have $\widehat{\mathbb{V}}^\Omega(r)(\omega) = \widehat{\mathbb{V}}^\Omega(\Delta)(\omega)$. On the other hand, $\mathrm{Pr}_\mathrm{R}(\beta\sigma, r) = 0$ for $\beta \neq \alpha$. Therefore

$$\begin{array}{rcll}
\sum_{\sigma\in\Sigma^*\omega} \mathrm{Pr}_\mathrm{R}(\sigma, r) & = & \sum_{\alpha\sigma\in\Sigma^*\omega} \mathrm{Pr}_\mathrm{R}(\alpha\sigma, r) & \\
& = & \sum_{\sigma\in\Sigma^*\omega} \mathrm{Pr}_\mathrm{R}(\alpha\sigma, r) & \\
& = & \sum_{\sigma\in\Sigma^*\omega} \mathrm{Pr}_\mathrm{R}(\sigma, \Delta) & \\
& = & \widehat{\mathbb{V}}^\Omega(\Delta)(\omega) & \text{by induction} \\
& = & \widehat{\mathbb{V}}^\Omega(r)(\omega) \ . &
\end{array}$$

- Finally, $\sum_{\sigma\in\Sigma^*\omega} \mathrm{Pr}_\mathrm{R}(\sigma, \Delta) = \sum_{\sigma\in\Sigma^*\omega} \mathrm{Exp}_\Delta(\mathrm{Pr}_\mathrm{R}(\sigma, \_)) = \mathrm{Exp}_\Delta(\sum_{\sigma\in\Sigma^*\omega} \mathrm{Pr}_\mathrm{R}(\sigma, \_))$ $= \mathrm{Exp}_\Delta(\widehat{\mathbb{V}}^\Omega(\_)(\omega)) = \mathrm{Exp}_\Delta(\widehat{\mathbb{V}}^\Omega(\_))(\omega) = \widehat{\mathbb{V}}^\Omega(\Delta)(\omega)$. □

Now we look more closely at the interaction of $\widehat{\mathbb{V}}^\Omega_\updownarrow$ and resolutions.

Lemma A.5. *Let* $\Delta^\circ \in \mathcal{D}(\mathsf{sCSP}^\Omega)$.
(1) If $\langle R, \Theta^\circ, \rightarrow\rangle$ *is a resolution of* Δ°*, then* $\widehat{\mathbb{V}}^\Omega_\updownarrow(\Theta^\circ) \in \widehat{\mathbb{V}}^\Omega_\updownarrow(\Delta^\circ)$.
(2) If $o \in \widehat{\mathbb{V}}^\Omega_\updownarrow(\Delta^\circ)$ *then there is a resolution* $\langle R, \Theta^\circ, \rightarrow\rangle$ *of* Δ° *such that* $\widehat{\mathbb{V}}^\Omega_\updownarrow(\Theta^\circ) = o$.

Proof.

(1) Let $\langle R, \Theta^\circ, \rightarrow\rangle$ be a resolution of Δ° with resolving function f. We observe that for any $\Theta \in \mathcal{D}(R)$ we have

$$\forall r \in \lceil\Theta\rceil : \widehat{\mathbb{V}}^\Omega_\updownarrow(r) \in \widehat{\mathbb{V}}^\Omega_\updownarrow(f(r)) \text{ implies } \widehat{\mathbb{V}}^\Omega_\updownarrow(\Theta) \in \widehat{\mathbb{V}}^\Omega_\updownarrow(f(\Theta)) \tag{A.2}$$

because

$$\begin{array}{rcl}
\widehat{\mathbb{V}}^\Omega_\updownarrow(\Theta) & = & \sum_{r\in\lceil\Theta\rceil} \Theta(r) \cdot \widehat{\mathbb{V}}^\Omega_\updownarrow(r) \\
& \in & \sum_{r\in\lceil\Theta\rceil} \Theta(r) \cdot \widehat{\mathbb{V}}^\Omega_\updownarrow(f(r)) \\
& = & \sum_{s\in\lceil f(\Theta)\rceil} f(\Theta)(s) \cdot \widehat{\mathbb{V}}^\Omega_\updownarrow(s) \\
& = & \widehat{\mathbb{V}}^\Omega_\updownarrow(f(\Theta)) \ .
\end{array}$$

We now prove by induction on *depth*(r) that $\forall r \in T : \widehat{\mathbb{V}}^\Omega_\updownarrow(r) \in \widehat{\mathbb{V}}^\Omega_\updownarrow(f(r))$, from which the required result follows in view of (A.2) and the fact that $f(\Theta^\circ) = \Delta^\circ$.

- In the base case we have $r \not\rightarrow$, which implies $f(r) \not\rightarrow$. Therefore, we have $\widehat{\mathbb{V}}^{\Omega}_{\updownarrow}(r) = \vec{0} \in \widehat{\mathbb{V}}^{\Omega}_{\updownarrow}(f(r))$.
- Otherwise r has a transition $r \xrightarrow{\alpha} \Theta$ for some α and Θ. By induction we have $\widehat{\mathbb{V}}^{\Omega}_{\updownarrow}(r') \in \widehat{\mathbb{V}}^{\Omega}_{\updownarrow}(f(r'))$ for all $r' \in \lceil\Theta\rceil$. Using (A.2) we get $\widehat{\mathbb{V}}^{\Omega}_{\updownarrow}(\Theta) \in \widehat{\mathbb{V}}^{\Omega}_{\updownarrow}(f(\Theta))$. Now
$$\widehat{\mathbb{V}}^{\Omega}_{\updownarrow}(r) \;=\; \alpha!\widehat{\mathbb{V}}^{\Omega}_{\updownarrow}(\Theta) \;\in\; \alpha!\widehat{\mathbb{V}}^{\Omega}_{\updownarrow}(f(\Theta)) \;\subseteq\; \widehat{\mathbb{V}}^{\Omega}_{\updownarrow}(f(r))$$
where the last step follows from the fact that $f(r) \xrightarrow{\alpha} f(\Theta)$ is one of the transitions of $f(r)$.

(2) This clause is proved by induction on $depth(\Delta^\circ)$. First consider the special case that Δ° is a point distribution on some state s.
- In the base case we have $s \not\rightarrow$. The probabilistic automaton $\langle\{s\}, \overline{s}, \emptyset\rangle$ is a resolution of $\Delta^\circ = \overline{s}$ with the resolving function being the identity. Clearly, this resolution satisfies our requirement.
- Otherwise there is a finite, non-empty index set I such that $s \xrightarrow{\alpha_i} \Delta_i$ for some actions α_i and distributions Δ_i. If $o \in \widehat{\mathbb{V}}^{\Omega}_{\updownarrow}(\Delta^\circ) = \widehat{\mathbb{V}}^{\Omega}_{\updownarrow}(s)$, then by the definition of $\widehat{\mathbb{V}}^{\Omega}_{\updownarrow}$ we have $o = \sum_{i\in I} p_i{\cdot}\alpha_i!o_i$ with $o_i \in \widehat{\mathbb{V}}^{\Omega}_{\updownarrow}(\Delta_i)$ and $\sum_{i\in I} p_i = 1$ for some $p_i \in [0,1]$. By induction, for each $i \in I$ there is a resolution $\langle R_i, \Theta_i^\circ, \rightarrow_i\rangle$ of Δ_i with resolving function f_i such that $\widehat{\mathbb{V}}^{\Omega}_{\updownarrow}(\Theta_i^\circ) = o_i$. Without loss of generality, we assume that R_i is disjoint from R_j for $i \neq j$, as well as from $\{r_i \mid i \in I\}$. We now construct a fully probabilistic automaton $\langle R, \Theta^\circ, \rightarrow'\rangle$ as follows:
  - $R := \{r_i \mid i \in I\} \cup \bigcup_{i\in I} R_i$
  - $\Theta^\circ := \sum_{i\in I} p_i \cdot \overline{r_i}$
  - $\rightarrow' := \{r_i \xrightarrow{\alpha_i} \Theta_i^\circ \mid i \in I\} \cup \bigcup_{i\in I} \rightarrow_i$.

  This automaton is a resolution of $\Delta^\circ = \overline{s}$ with resolving function f defined by
$$f(r) = \begin{cases} s & \text{if } r = r_i \text{ for } i \in I \\ f_i(r) & \text{if } r \in R_i \text{ for } i \in I. \end{cases}$$
  The resolution thus constructed satisfies our requirement because
$$\begin{array}{rcl} \widehat{\mathbb{V}}^{\Omega}_{\updownarrow}(\Theta^\circ) & = & \widehat{\mathbb{V}}^{\Omega}_{\updownarrow}(\sum_{i\in I} p_i \cdot \overline{r_i}) \\ & = & \sum_{i\in I} p_i \cdot \widehat{\mathbb{V}}^{\Omega}_{\updownarrow}(r_i) \\ & = & \sum_{i\in I} p_i \cdot \alpha_i!\widehat{\mathbb{V}}^{\Omega}_{\updownarrow}(\Theta_i^\circ) \\ & = & \sum_{i\in I} p_i \cdot \alpha_i!o_i \\ & = & o\ . \end{array}$$

We now consider the general case that Δ° is a proper distribution with $\lceil\Delta^\circ\rceil = \{s_j \mid j \in J\}$ for some finite index set J. Using the reasoning in the above special case, we have a resolution $\langle R_j, \Theta_j^\circ, \rightarrow_j\rangle$ of each distribution $\overline{s_j}$. Without loss of generality, we assume that R_j is disjoint from R_k for $j \neq k$. Consider the probabilistic automaton $\langle\bigcup_{j\in J} R_j, \sum_{j\in J} \Delta^\circ(s_j) \cdot \Theta_j^\circ, \bigcup_{j\in J} \rightarrow_j\rangle$. It is a resolution of Δ° satisfying our requirement. If $o \in \widehat{\mathbb{V}}^{\Omega}_{\updownarrow}(\Delta^\circ)$ then $o = \sum_{j\in J} \Delta^\circ(s_j){\cdot}o_j$ with $o_j \in \widehat{\mathbb{V}}^{\Omega}_{\updownarrow}(s_j)$. Since $o_j = \widehat{\mathbb{V}}^{\Omega}_{\updownarrow}(\Theta_j^\circ)$, we have $o = \widehat{\mathbb{V}}^{\Omega}_{\updownarrow}(\sum_{j\in J} \Delta^\circ(s_j) \cdot \Theta_j^\circ)$. □

We can now give the result relied on in Section 6.

Proposition A.6. *Let $\Delta^\circ \in \mathcal{D}(\mathsf{sCSP}^\Omega)$. Then we have that $\mathbb{W}(\Delta^\circ) = \widehat{\mathbb{V}}^{\Omega}_{\updownarrow}(\Delta^\circ)$.*

Proof. Combine Lemmas A.4 and A.5. □

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, M. Hennessy, C.C. Morgan & C. Zhang (2007): *Characterising testing preorders for finite probabilistic processes.* In Proc. *LICS'07*, IEEE Computer Society Press, pp. 313–322.

[10] 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.

[11] 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.

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

[13] 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.

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

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

[16] M. Hennessy & R. Milner (1985): *Algebraic Laws for Nondeterminism and Concurrency.* *Journal of the ACM* 32(1), pp. 137–161.

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

[18] C. Jones 1990: *Probabilistic Non-Determinism.* PhD thesis, Department of Computer Science, University of Edinburgh.

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

[20] 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.

[21] B. Jonsson & K.G. Larsen (1991): *Specification and refinement of probabilistic processes.* In Proceedings of the 6th Annual IEEE Symposium on *Logic in Computer Science*, Computer Society Press, pp. 266–277.

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

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

[24] 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.

[25] D. Kozen (1981): *Semantics of Probabilistic Programs.* Journal of Computer and System Sciences 22, pp. 328–350.

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

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

[28] G. Lowe (1993): *Representing nondeterminism and probabilistic behaviour in reactive processes.* Technical Report TR-11-93, Computing laboratory, Oxford University.
[29] N. Lynch, R. Segala & F.W. Vaandrager (2003): *Compositionality for probabilistic automata.* In Proc. *CONCUR'03*, LNCS 2761, Springer, pp. 204–222.
[30] R. Milner (1989): *Communication and Concurrency.* Prentice-Hall.
[31] M.W. Mislove (2000): *Nondeterminism and probabilistic choice: Obeying the laws.* In Proc. *CONCUR'00*, LNCS 1877, Springer, pp. 350–364.
[32] 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.
[33] M. Núñez (2003): *Algebraic theory of probabilistic processes. Journal of Logic and Algebraic Programming* 56, pp. 117–177.
[34] E.-R. Olderog & C.A.R. Hoare (1986): *Specification-oriented semantics for communicating processes. Acta Informatica* 23, pp. 9–66.
[35] M.L. Puterman (1994): *Markov Decision Processes.* Wiley.
[36] R. Segala (1995): *Modeling and Verification of Randomized Distributed Real-Time Systems.* PhD thesis, MIT.
[37] R. Segala (1996): *Testing probabilistic automata.* In Proc. *CONCUR'96*, LNCS 1119, Springer, pp. 299–314.
[38] R. Segala & N.A. Lynch (1994): *Probabilistic simulations for probabilistic processes.* In Proc. *CONCUR'94*, LNCS 836, Springer, pp. 481–496.
[39] K. Seidel (1995): *Probabilistic communicating processes. Theoretical Computer Science* 152(2), pp. 219–249.
[40] R. Tix, K. Keimel & G.D. Plotkin (2005): *Semantic domains for combining probability and nondeterminism. ENTCS* 129, pp. 1–104.
[41] Wang Yi & K.G. Larsen (1992): *Testing probabilistic and nondeterministic processes.* In Proc. *PSTV'92*, *IFIP Transactions* C-8, North-Holland, pp. 47–61.