CHARACTERISING TESTING PREORDERS
FOR FINITE PROBABILISTIC PROCESSES

YUXIN DENG\textsuperscript{a}, ROB VAN GLABBEEK\textsuperscript{b}, MATTHEW HENNESSY\textsuperscript{c}, AND CARROLL MORGAN\textsuperscript{d}

\textsuperscript{a} Shanghai Jiao Tong University, China, and University of New South Wales, Australia
\textit{e-mail address}: yuxindeng@sjtu.edu.cn

\textsuperscript{b} National ICT Australia, and University of New South Wales, Australia
\textit{e-mail address}: rvg@cs.stanford.edu

\textsuperscript{c} Trinity College Dublin, Ireland
\textit{e-mail address}: matthew.hennessy@cs.tcd.ie

\textsuperscript{d} University of New South Wales, Australia
\textit{e-mail address}: carrollm@cse.unsw.edu.au

\textbf{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 \cite{13,41,28,20,38,39,36,22,32,87,14,26,31,11,23,29,8,40,7}. In 1992 Wang & Larsen posed the problems of finding complete axiomatisations and alternative characterisations for a natural generalisation of the standard testing preorders \cite{0} to such processes \cite{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.

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 $P \square Q$, internal $P \triangle Q$ and the newly added probabilistic choice $P \oplus Q$. 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 $s \xrightarrow{\Delta}$, where $\Delta$ 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, $\preceq_{pmay}$ and $\preceq_{pmust}$. 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 ($\preceq_{pmay}$ and $\preceq_{pmust}$) 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 $\preceq_{pmay}$ and $\preceq_{pmust}$ that arise from action-based but scalar testing, and use a recent result [10] saying that for finite processes the preorders $\preceq_{pmay}$ and $\preceq_{pmust}$ coincide with $\preceq_{pmay}$ and $\preceq_{pmust}$. Here we show that on $pCSP$ the preorders $\preceq_{pmay}$ and $\preceq_{pmust}$ also coincide with $\preceq_{pmay}$ and $\preceq_{pmust}$.

Simulation preorders: In Section 4 we use the transitions $s \xrightarrow{\alpha} \Delta$ to define two co-inductive preorders, the simulation preorder $\preceq_{FS}$ [36, 29, 8], and the novel failure simulation preorder $\preceq_{FS}$ 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 $\Delta \xrightarrow{\alpha} \Delta'$ of the second preorder differs from the first one in the use of a failure predicate $s \xrightarrow{\alpha} X$, 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 $P \preceq S Q$ implies $P \preceq_{pmay} Q$ and $P \preceq_{FS} Q$ implies $P \preceq_{pmust} Q$. For $\preceq_{FS}$ this was established in [8], and here we use similar techniques in the proofs for $\preceq_{FS}$. 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.

\footnote{However in the presence of divergence they are slightly different.}
**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 manner, a logical preorder between processes: $P \sqsubseteq^L Q$ means that every $L$-formula satisfied by $P$ is also satisfied by $Q$. We establish that $\sqsubseteq^L$ coincides with $\sqsubseteq_S$ and $\sqsubseteq_{p \text{may}}$.

To capture failures, we add, for every set of actions $X$, a formula $\text{ref}(X)$ 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 $\text{ref}()$ stem from the modal characterisation of the non-probabilistic failure simulation preorder, given in [11]. We show that $\sqsubseteq_{p \text{must}}$, 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:

$$
\begin{align*}
\sqsubseteq^L & \subseteq \sqsubseteq^F & \text{(Sec. 7)} & \subseteq \sqsubseteq_{S} & \subseteq \sqsubseteq_{p \text{may}} & \subseteq \widehat{\sqsubseteq}_{p \text{may}} & \subseteq \sqsubseteq^L \\
\sqsubseteq^F & \subseteq \sqsubseteq_{FS} & \subseteq \sqsubseteq_{p \text{must}} & \subseteq \widehat{\sqsubseteq}_{p \text{must}} & \subseteq \sqsubseteq^F
\end{align*}
$$

In Section 7 we show that $P \sqsubseteq^L Q$ implies $P \sqsubseteq_S Q$ (and hence $P \sqsubseteq_{p \text{may}} Q$), and likewise for $\sqsubseteq^F$ and $\sqsubseteq_{FS}$; the proof involves constructing, for each $p\text{CSP}$ process $P$, a characteristic formula $\varphi_P$. To obtain the other direction, in Section 8 we show how every modal formula $\varphi$ can be captured, in some sense, by a test $T_\varphi$; essentially the ability of a $p\text{CSP}$ process to satisfy $\varphi$ is determined by its ability to pass the test $T_\varphi$. 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 [10, 2, 15]. In [8] we showed that most of the standard equations are no longer valid in the probabilistic setting of $p\text{CSP}$; we also provided a set of axioms which are complete with respect to (probabilistic) may-testing for the sub-language of $p\text{CSP}$ without probabilistic choice. Here we extend this result, by showing, in Section 10 that both $P \sqsubseteq_{p \text{may}} Q$ and $P \sqsubseteq_{p \text{must}} Q$ can still be captured equationally over full $p\text{CSP}$. 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 \( \text{Act} \) 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:

\[
\begin{align*}
P & ::= S \mid P \oplus P \\
S & ::= 0 \mid a.P \mid P \sqcap P \mid S \Box S \mid S \mid_A S
\end{align*}
\]

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

The process \( P \oplus Q \), 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 \( \oplus \). The state-based processes have a CSP-like syntax, involving the stopped process \( 0 \), action prefixing \( a._- \) for \( a \in \text{Act} \), internal- and external choices \( \sqcap \) and \( \Box \), and a parallel composition \( |_A \) for \( A \subseteq \text{Act} \).

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

The process \( s \Box t \) 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 \( s \mid_A t \), where \( A \subseteq \text{Act} \), represents processes \( s \) and \( t \) running in parallel. They may synchronise by performing the same action from \( A \) simultaneously; such a synchronisation results in \( \tau \). In addition \( s \) and \( t \) may independently do any action from \( (\text{Act} \setminus A) \cup \{ \tau \} \).

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 \( \oplus \). Thus for example \( s \Box (t_1 \oplus t_2) \) abbreviates the term \( (s \Box t_1) \oplus (s \Box t_2) \).

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 \( P \mid_A Q = (P\mid_AQ) \setminus A \), where \( \mid_A \) and \( \setminus_A \) 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 \( 0 \); 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 \( \bigcap_{i \in I} P_i \), 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 \(^3\) [30] that includes probabilities.

A (discrete) probability distribution over a set \(S\) is a function \(\Delta : S \rightarrow [0, 1]\) with \(\sum_{s \in S} \Delta(s) = 1\); the support of \(\Delta\) is given by \(\{s \in S \mid \Delta(s) > 0\}\). We write \(\mathcal{D}(S)\), ranged over by \(\Delta, \Theta, \Phi\), 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 \(\mathcal{P}\) to denote the point distribution assigning probability 1 to \(s\) and 0 to all others, so that \([\mathcal{P}] = \{s\}\). If \(p_i \geq 0\) and \(\Delta_i\) is a distribution for each \(i\) in some finite index set \(I\), and \(\sum_{i \in I} p_i = 1\), then the 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 \(\Delta\) a distribution over \(S\) and function \(f : S \rightarrow X\) into a vector space \(X\) we sometimes write \(\text{Exp}_\Delta(f)\) 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 \(F : S \rightarrow \mathcal{P}^+(X)\) with \(\mathcal{P}^+(X)\) being the collection of non-empty subsets of \(X\), we define \(\text{Exp}_\Delta F := \{ \text{Exp}_\Delta(f) \mid f \in \mathcal{P}^+(X) \}\); here \(f \in \mathcal{P}^+(X)\) means that \(f : S \rightarrow X\) 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)\(^4\) is a triple \(\langle S, L, \rightarrow \rangle\), where

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

As with LTSs, we usually write \(s \xrightarrow{\alpha} \Delta\) for \((s, \alpha, \Delta) \in \rightarrow\), \(s \xrightarrow{\alpha, \Delta}\) for \(\exists \Delta : s \xrightarrow{\alpha} \Delta\) and \(s \rightarrow\) for \(\forall \alpha : s \xrightarrow{\alpha}\). 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 \(\langle \text{sCSP}, \text{Act}_r, \rightarrow \rangle\), constructed by taking sCSP to be the set of states and \(\text{Act}_r := \text{Act} \cup \{\tau\}\) the set of transition labels; we let \(a\) range over Act and \(\alpha\) over \(\text{Act}_r\). We interpret pCSP processes \(P\) as distributions \([P] \in \mathcal{D}(\text{sCSP})\) via the function \([\cdot] : \text{sCSP} \rightarrow \mathcal{D}(\text{sCSP})\) defined below:

\[
[s] := \mathcal{P} \text{ for } s \in \text{sCSP}
\]

\[
[P_1 \oplus Q] := p \cdot [P] + (1 - p) \cdot [Q].
\]

Note that for each \(P \in \text{pCSP}\) 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 \([31]\), but modified so that the result of an action is a distribution. The rules for external choice and parallel composition use

---

\(^2\) Although the syntax of pCSP is similar to other probabilistic extensions of CSP \([28, 22, 31]\), our semantics differs. For more detailed comparisons, see Section 12.

\(^3\) Essentially the same model has appeared in the literature under different names such as NP-systems \([20]\), probabilistic processes \([22]\), simple probabilistic automata \([32]\), probabilistic transition systems \([23]\) etc. Furthermore, there are strong structural similarities with Markov Decision Processes \([35, 10]\).
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 $\Delta$ with $s \xrightarrow{\alpha} \Delta$ we draw an edge from $s$ to $\Delta$, labelled with $\alpha$. For any distribution $\Delta$ and state $s$ in $\lceil \Delta \rceil$, the support of $\Delta$, we draw an edge from $\Delta$ to $s$, labelled with $\Delta(s)$.

**Example 2.2.** Consider the two processes

$P := a.(b.d \Box c.e) \uplus (b.f \Box c.g)$

$Q := a.(b.d \Box c.g) \uplus (b.f \Box c.e)$.

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 $\omega \notin \text{Act}_\tau$, a special action reporting success; we write $\text{pCSP}^\omega$ for the set of all tests, and $\text{sCSP}^\omega$ for the subset of state-based process terms that may involve the action $\omega$, and the operational semantics above is extended by treating $\omega$ like any other action from $\text{Act}$. To apply test $T$ to process $P$ we form the process $T \mid_{\text{Act}} P$ 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 $T \mid_{\text{Act}} P$ and gives the probability that this resolution will reach a *success state*, one in which $\omega$ is possible.
To this end, we inductively define a results-gathering function $\mathbb{V} : sCSP^\omega \to \mathcal{P}^+([0,1])$;

it extends to type $\mathcal{D}(sCSP^\omega) \to \mathcal{P}^+([0,1])$ via the convention $\mathbb{V}(\Delta) := \text{Exp}_\Delta \mathbb{V}$.

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

In the first case above $s \xrightarrow{\omega}$ signifies that $s$ is a success state. In the second case we mean
that $\omega$ is not possible from $s$—hence $s$ is not a success state—but that at least one “non-success” action $\alpha \in \text{Act}_r$ is—and possibly several—and then the union is over all such $\alpha$. 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 $\tau$. 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

$$A(T, P) := \mathbb{V}[\mathcal{T} \mid \text{Act}_r P].$$

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

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

$$P \sqsubseteq_{\text{pmay}} Q \iff \text{ for all tests } T : A(T, P) \leq_{\text{Ho}} A(T, Q)$$

$$P \sqsubseteq_{\text{pmust}} Q \iff \text{ for all tests } T : A(T, P) \leq_{\text{Sm}} A(T, Q)$$

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

$$X \leq_{\text{Ho}} Y \iff \forall x \in X : \exists y \in Y : x \leq y$$

$$X \leq_{\text{Sm}} Y \iff \forall y \in Y : \exists x \in X : x \leq y$$

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 $P$. 

![Figure 2: Example processes P, Q and test T](image-url)
Example 3.3. Consider the test
\[ T := a((b.d.\omega_1 \oplus c.e.\omega) \sqcap (b.f.\omega_1 \oplus c.g.\omega)) \]
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:

\[ A(T, P) = \frac{1}{2} \cdot \{1, 0\} + \frac{1}{2} \cdot \{1, 0\} = \{0, \frac{1}{2}, 1\} \]
\[ A(T, Q) = \frac{1}{2} \cdot \{\frac{1}{2}\} + \frac{1}{2} \cdot \{\frac{1}{2}\} = \{\frac{1}{2}\}. \]

We can see that \( P \) and \( Q \) can be distinguished by the test \( T \) since \( A(T, P) \not\sim_{H0} A(T, Q) \) and \( A(T, Q) \not\sim_{Sm} A(T, P) \). In other words, we have \( P \not\preceq_{p\text{may}} Q \) and \( Q \not\preceq_{p\text{must}} P \) 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 \( P \preceq_{p\text{must}} Q \) implies \( Q \preceq_{p\text{may}} P \) for all pCSP processes \( P \) and \( Q \), i.e. that must testing is more discriminating than may testing and that the preorderings \( \preceq_{p\text{may}} \) and \( \preceq_{p\text{must}} \) are oriented in opposite directions.

4. Simulation and failure simulation

Let \( R \subseteq S \times \mathcal{D}(S) \) be a relation from states to distributions. As in [8], we lift it to a relation \( \overrightarrow{R} \subseteq \mathcal{D}(S) \times \mathcal{D}(S) \) by letting \( \Delta \overrightarrow{R} \Theta \) whenever there is a finite index set \( I \) and \( p \in \mathcal{D}(I) \) such that

(i) \( \Delta = \sum_{i \in I} p_i \cdot \overrightarrow{s_i} \),
(ii) For each \( i \in I \) there is a distribution \( \Phi_i \) 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 \( R \), sometimes called weight function (see e.g. [21, 36]). An important point here is that in the decomposition (i) of \( \Delta_1 \) into \( \sum_{i \in I} p_i \cdot s_i \), 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 \( \alpha \) are again denoted \( \alpha^L \).

We write \( s \xrightarrow{\tau} \Delta \) if either \( s \xrightarrow{\tau} \Delta \) or \( \Delta = \overline{s} \); again \( \Delta_1 \xrightarrow{\tau} \Delta_2 \) denotes the lifted relation. Thus for example we have \([[(a \cap b) \frac{1}{2} \oplus (a \cap c)] \xrightarrow{\tau} [a \frac{1}{2} \oplus ((a \cap b) \frac{1}{2} \oplus c)] \] because

(i) \([[(a \cap b) \frac{1}{2} \oplus (a \cap c)] = \frac{1}{4} \cdot [(a \cap b)] + \frac{1}{4} \cdot [(a \cap c)] + \frac{1}{4} \cdot [(a \cap c)]]

(ii) \([[(a \cap b)] \xrightarrow{\tau} [a] \]

\([[(a \cap b)] \xrightarrow{\tau} [a] \]

\([[(a \cap c)] \xrightarrow{\tau} [a] \]

\([[(a \cap c)] \xrightarrow{\tau} [c] \]

(iii) and \([a \frac{1}{2} \oplus ((a \cap b) \frac{1}{2} \oplus c)] = \frac{1}{4} \cdot [a] + \frac{1}{4} \cdot [(a \cap b)] + \frac{1}{4} \cdot [a] + \frac{1}{4} \cdot [c] \).

We now define the weak transition relation \( \xrightarrow{\ast} \) as the transitive and reflexive closure \( \xrightarrow{\tau} \ast \), while for \( a \neq \tau \) we let \( \Delta_1 \xrightarrow{a} \Delta_2 \) denote \( \Delta_1 \xrightarrow{a} \ast \Delta_2 \). Finally, we write \( s \xrightarrow{X} \Phi \) with \( X \subseteq \text{Act} \) when \( \forall \alpha \in X \cup \{\tau\} : s \xrightarrow{\alpha} \Phi \), and \( \Delta \xrightarrow{X} \Phi \) when \( \forall s \in [\Delta] : s \xrightarrow{X} \Phi \). 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 \( \Delta_i \xrightarrow{\alpha} \Phi_i \) for each \( i \in I \), with \( I \) a finite index set. Then

\[ \sum_{i \in I} p_i \cdot \Delta_i \xrightarrow{\alpha} \sum_{i \in I} p_i \cdot \Phi_i \]

Conversely, if \( \sum_{i \in I} p_i \cdot \Delta_i \xrightarrow{\alpha} \Phi \) then \( \Phi = \sum_{i \in I} p_i \cdot \Phi_i \) for some \( \Phi_i \) such that \( \Delta_i \xrightarrow{\alpha} \Phi_i \) 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 \( R \) to be \( \xrightarrow{\tau} \) and \( \xrightarrow{\alpha} \) for \( a \in \text{Act} \). \( \square \)

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

- \( s \xrightarrow{\mathcal{R} \Theta \wedge s} \xrightarrow{\alpha} \Delta \) implies \( \exists \Theta' : \Theta \xrightarrow{\alpha} \Theta' \wedge \Delta \xrightarrow{\mathcal{R}} \Theta' \)
- \( s \xrightarrow{\mathcal{R} \Theta \wedge s} \xrightarrow{\alpha} \Theta' \) implies \( \exists \Theta' : \Theta \xrightarrow{\alpha} \Theta' \wedge \Theta' \xrightarrow{\mathcal{R}} \).

We write \( s \xrightarrow{\mathcal{R}} \Theta \) to mean that there is some failure simulation \( R \) such that \( s \xrightarrow{\mathcal{R}} \Theta \). Similarly, we define simulation\(^4\) and \( s \xrightarrow{\mathcal{S}} \Theta \) by dropping the second clause in Definition [4.2].

**Definition 4.3.** The simulation preorder \( \sqsubseteq \) and failure simulation preorder \( \sqsubseteq_{FS} \) on \( \text{pCSP} \) are defined as follows:

\[ P \sqsubseteq Q \iff [Q] \xrightarrow{\mathcal{S}} \Theta \text{ for some } \Theta \text{ with } [P] \xrightarrow{\mathcal{S}} \Theta \]

\[ P \sqsubseteq_{FS} Q \iff [Q] \xrightarrow{\mathcal{S}} \Theta \text{ for some } \Theta \text{ with } [P] \xrightarrow{\mathcal{S}} \Theta \]

(Note the opposing directions.) The equivalences generated by \( \sqsubseteq \) and \( \sqsubseteq_{FS} \) are called (failure) simulation equivalence, denoted \( \approx \) and \( \approx_{FS} \), respectively.

\(^4\)It is called forward simulation in [30].

\(^5\)We have reversed the orientation of the symbols \( \succ \) and \( \succ_{FS} \) w.r.t. [8] and [9]; the pointy side now points to a single state, and the flat side to a distribution.
Example 4.4. Compare the processes $P = a \frac{1}{2} \oplus b$ and $P \cap P$. Note that $[P]$ is the distribution $\frac{1}{2} \cdot \pi + \frac{1}{2} \cdot \bar{b}$ whereas $[P \cap P]$ is the point distribution $\overline{P \cap P}$. The relation $R$ given by

$$(P \cap P) \mathcal{R} (\frac{1}{2} \cdot \pi + \frac{1}{2} \cdot \bar{b}) \quad a \mathcal{R} \pi \quad b \mathcal{R} \bar{b} \quad 0 \mathcal{R} \emptyset$$

is a simulation, because the $\tau$-step $P \cap P \xrightarrow{\tau} (\frac{1}{2} \cdot \pi + \frac{1}{2} \cdot \bar{b})$ can be matched by the idle transition $(\frac{1}{2} \cdot \pi + \frac{1}{2} \cdot \bar{b}) \mathcal{R} (\frac{1}{2} \cdot \pi + \frac{1}{2} \cdot \bar{b})$, and we have $(\frac{1}{2} \cdot \pi + \frac{1}{2} \cdot \bar{b}) \mathcal{R} (\frac{1}{2} \cdot \pi + \frac{1}{2} \cdot \bar{b})$. Thus $(P \cap P) \triangleleft_S (\frac{1}{2} \cdot \pi + \frac{1}{2} \cdot \bar{b}) = [P]$, hence $[P \cap P] \triangleleft_S [P]$, and therefore $P \cap P \subseteq_S P$.

This type of reasoning does not apply to the other direction. Any simulation $R$ with $(\frac{1}{2} \cdot \pi + \frac{1}{2} \cdot \bar{b}) \mathcal{R} P \cap P$ would have to satisfy $a \mathcal{R} P \cap P$ and $b \mathcal{R} P \cap P$. However, the move $a \xrightarrow{\text{move}} 0$ cannot be matched by the process $P \cap P$, as the only transition the latter process can do is $P \cap P \xrightarrow{\tau} (\frac{1}{2} \cdot \pi + \frac{1}{2} \cdot \bar{b})$, and only half of that distribution can match the $a$-move. Thus, no such simulation exists, and we find $[P] \ntriangleleft_S [P \cap P]$. Nevertheless, we still have $P \subseteq_S P \cap P$. Here, the transition $\overline{\tau}$ comes from Definition 4.3 comes to the rescue. As $\overline{P \cap P} \xrightarrow{\tau} [P]$ and $[P] \triangleleft_S [P]$, we obtain $P \subseteq_S P \cap P$.

Example 4.5. Let $P = a \frac{1}{2} \oplus b$ and $Q = P \square P$. We have $P \subseteq_S Q$ because $[P] \triangleleft_S [Q]$ which comes from the following observations:

1. $[P] = \frac{1}{2} \cdot \pi + \frac{1}{2} \cdot \bar{b}$
2. $[Q] = \frac{1}{2} \cdot \pi + \frac{1}{2} \cdot \bar{b}$
3. $a \prec_S (\frac{1}{2} \cdot \pi + \frac{1}{2} \cdot \bar{b})$
4. $b \prec_S (\frac{1}{2} \cdot \pi + \frac{1}{2} \cdot \bar{b})$

This kind of reasoning does not apply to $\prec_{FS}$. For example, we have $a \not\prec_{FS} (\frac{1}{2} \cdot \pi + \frac{1}{2} \cdot \bar{b})$ 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 $Q \nsubseteq_{FS} P$.

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

Proposition 4.6. Suppose $\subseteq \in \{\subseteq_S, \subseteq_{FS}\}$. Then $\subseteq$ is a preorder, and if $P_i \subseteq Q_i$ for $i = 1, 2$ then $a.P_i \subseteq a.Q_i$ for $a \in \text{Act}$ and $P_1 \cap P_2 \subseteq Q_1 \cap Q_2$ for $\cap \in \{\sqcap, \sqcup, \oplus, |A|\}$.

Proof. The case $\subseteq_S$ was proved in [8] Corollary 6.10 and Theorem 6.13; the case $\subseteq_{FS}$ is analogous. As an example, we show that $\subseteq_{FS}$ is preserved under parallel composition. The key step is to show that the binary relation $\mathcal{R} \subseteq \text{scSP} \times \mathcal{D}(\text{scSP})$ defined by

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

is a failure simulation.

Suppose $s_i \triangleleft_{FS} \Delta_i$ for $i = 1, 2$ and $s_1 |_{A}s_2 \xrightarrow{X}\text{}$ for some $X \subseteq \text{Act}$. For each $a \in X$ there are two possibilities:

- If $a \not\in \Delta$ then $s_1 \xrightarrow{a} \Delta$ and $s_2 \xrightarrow{a} \Delta$, since otherwise we would have $s_1 |_{A}s_2 \xrightarrow{a} \Delta$.
- If $a \in \Delta$ then either $s_1 \xrightarrow{a} \Delta$ or $s_2 \xrightarrow{a} \Delta$, since otherwise we would have $s_1 |_{A}s_2 \xrightarrow{a} \Delta$.

Hence we can partition the set $X$ into three subsets: $X_0$, $X_1$ and $X_2$ such that $X_0 = X \setminus A$ and $X_1 \cup X_2 \subseteq A$ with $s_1 \xrightarrow{X_{0}}\Delta$ and $s_2 \xrightarrow{X_{0}}\Delta$, but allowing $s_1 \xrightarrow{X_{1}}\Delta$ and $s_2 \xrightarrow{X_{1}}\Delta$ for some $a \in X_2$ and $s_2 \xrightarrow{X_{2}}\Delta$ for some $a \in X_1$. We then have that $s_i \xrightarrow{X_{0}\cup X_{1}}\Delta$ for $i = 1, 2$. By the assumption that $s_i \triangleleft_{FS} \Delta_i$ for $i = 1, 2$, there is a $\Delta_i$ with $\Delta_i \xrightarrow{X_{0}\cup X_{1}}\Delta_i$. Therefore $\Delta_1 |_{A}\Delta_2 \xrightarrow{X_{0}\cup X_{1}}\Delta$ as well. It is stated in [8] Lemma 6.12(i) that if $\Phi \equiv \Phi'$ then $\Phi |_{A}\Delta \equiv \Phi' |_{A}\Delta$ and
implies Definition 4.8. So we have \( \Delta_1 |_A \Delta_2 \Rightarrow_\phi \Delta'_1 |_A \Delta'_2 \). 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 \( \sqsubseteq_{\text{FS}} \) 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 \( P \sqsubseteq_{\text{pmay}} Q \).

**Proof.** For any test \( T \in \text{pCSP}^\omega \) and process \( P \in \text{pCSP} \) the set \( \forall(\delta |_{\text{Act}} P) \) is finite, so

\[
P \sqsubseteq_{\text{pmay}} Q \text{ iff } \max(\forall([\delta |_{\text{Act}} P])) \leq \max(\forall([\delta |_{\text{Act}} Q])) \text{ for every test } T. \tag{4.1}
\]

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

\[
\Delta_1 \Rightarrow_\phi \Delta_2 \text{ implies } \max(\forall(\Delta_1)) \geq \max(\forall(\Delta_2)). \tag{4.2}
\]

\[
\Delta_1 \triangleleft_S \Delta_2 \text{ implies } \max(\forall(\Delta_1)) \leq \max(\forall(\Delta_2)). \tag{4.3}
\]

In \([8] \) Lemma 6.15 and Proposition 6.16] similar properties are proven using a function \( \text{maxlive} \) instead of \( \text{max}\forall \). The same arguments apply here.

Now suppose \( P \sqsubseteq_S Q \). Since \( \sqsubseteq_S \) is preserved by the parallel operator we have that \( T |_{\text{Act}} P \sqsubseteq_S T |_{\text{Act}} Q \) for an arbitrary test \( T \). By definition, this means that there is a distribution \( \Delta \) such that \( [T |_{\text{Act}} Q] \Rightarrow_\phi \Delta \) and \( [T |_{\text{Act}} P] \triangleleft_S \Delta \). By (4.2) and (4.3) we infer that \( \max(\forall([T |_{\text{Act}} P])) \leq \max(\forall([T |_{\text{Act}} Q])) \). The result now follows from (4.1).

It is tempting to use the same idea to prove that \( \sqsubseteq_{\text{FS}} \) implies \( \sqsubseteq_{\text{pmust}} \), but now using the function \( \text{min}\forall \). However, the \( \min \)-analogue of Property (4.2) is in general invalid. For example, let \( R \) be the process \( a |_{\text{Act}} (a \square \omega) \). We have \( \min(\forall(R)) = 1 \), yet \( R \Rightarrow_\phi 0 |_{\text{Act}} 0 \) and \( \min(\forall(0 |_{\text{Act}} 0)) = 0 \). Therefore, it is not the case that \( \Delta_1 \Rightarrow_\phi \Delta_2 \) implies \( \min(\forall(\Delta_1)) \leq \min(\forall(\Delta_2)) \).

Our strategy is therefore as follows. Write \( s \Rightarrow_\phi \omega \) if both \( s \Rightarrow_\phi \) and \( s \Rightarrow_\phi \Delta \) hold. We define \( \Rightarrow_\phi \omega \) as \( \Rightarrow_\phi \omega \) in place of \( \Rightarrow_\phi \). Similarly we define \( \Rightarrow_\phi \omega \) and \( \Rightarrow_\phi \omega \). Thus the subscript \( \omega \) on a transition of any kind indicates that no state is passed through in which \( \omega \) is enabled. A version of failure simulation adapted to these transition relations is then defined as follows.

**Definition 4.8.** Let \( \triangleleft_{\text{FS}} \subseteq \text{sCSP}^\omega \times \mathcal{D}(\text{sCSP}^\omega) \) be the largest relation such that \( s \triangleleft_{\text{FS}} \Theta \) implies

- if \( s \Rightarrow_\phi \omega \) then there is some \( \Theta' \) with \( \Theta \Rightarrow_\phi \omega \Theta' \) and \( \Delta \triangleleft_{\text{FS}} \Theta' \)
- if \( s \not\Rightarrow_\phi \omega \) with \( \omega \in X \) then there is some \( \Theta' \) with \( \Theta \Rightarrow_\phi \omega \Theta' \) and \( \Theta' \not\Rightarrow_\phi \).

Let \( P \sqsubseteq_{\text{FS}} Q \) iff \( [P] \Rightarrow_{\text{FS}} \Theta \) for some \( \Theta \) with \( [Q] \triangleleft_{\text{FS}} \Theta \).

Note that for processes \( P, Q \) in \( \text{pCSP} \) (as opposed to \( \text{pCSP}^\omega \)), we have \( P \sqsubseteq_{\text{FS}} Q \) if \( P \sqsubseteq_{\text{FS}} Q \).

**Proposition 4.9.** If \( P, Q \) are processes in \( \text{pCSP} \) with \( P \sqsubseteq_{\text{FS}} Q \) and \( T \) is a process in \( \text{pCSP}^\omega \) then \( T |_{\text{Act}} P \sqsubseteq_{\text{FS}} T |_{\text{Act}} Q \).

**Proof.** Similar to the proof of Proposition 4.6.
Proposition 4.10. The following properties hold for \( \min \circ V \), with \( \Delta_1, \Delta_2 \in \mathcal{D}(sCSP^\omega) \):
\[
P \preceq_{\text{pmust}} Q \iff \min(V([T \mid \text{Act } P])) \leq \min(V([T \mid \text{Act } Q])) \text{ for every test } T.
\]
\[
\Delta_1 \xrightarrow{\alpha} \omega \Delta_2 \text{ for } \alpha \in \text{Act}_T \text{ implies } \min(V(\Delta_1)) \leq \min(V(\Delta_2)).
\]
\[
\Delta_1 \not\preceq_{\text{FS}} \Delta_2 \text{ implies } \min(V(\Delta_1)) \geq \min(V(\Delta_2)).
\]

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 \(s \in sCSP^\omega\) and \(\Delta \in \mathcal{D}(sCSP^\omega)\), that
\[
s \prec_{\text{FS}} \Delta \text{ implies } \min(V(s)) \geq \min(V(\Delta)).
\]

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 \(\min(V(s)) = 1 \geq \min(V(\Delta))\) trivially.
- If \(s \xrightarrow{\omega} \) but \(s \xrightarrow{}\), then we can closely follow the proof of \([8\), Lemma 6.16(i)\]:

Whenever \(s \xrightarrow{\alpha} \omega \Theta\), for \(\alpha \in \text{Act}_T\) and \(\Theta \in \mathcal{D}(sCSP^\omega)\), then \(s \prec_{\text{FS}} \Delta\) implies the existence of some \(\Delta_\Theta\) such that \(\Delta \xrightarrow{\Delta_\Theta} \omega\Delta_\Theta\) and \(\Theta \prec_{\text{FS}} \Delta_\Theta\). By induction, using (4.6), it follows that \(\min(V(\Theta)) \geq \min(V(\Delta_\Theta))\). Consequently, we have that
\[
\min(V(s)) = \min(\{\min(V(\Theta)) \mid s \xrightarrow{\alpha} \Theta\})
\]
\[
\geq \min(\{\min(V(\Delta_\Theta)) \mid s \xrightarrow{\alpha} \Theta\})
\]
\[
\geq \min(\{\min(V(\Delta)) \mid s \xrightarrow{\alpha} \Theta\}) \quad \text{(by (4.5))}
\]
\[
= \min(V(\Delta)).
\]

- If \(s \nRightarrow\), that is \(s \xrightarrow{\text{Act}^c} \), then there is some \(\Delta'\) such that \(\Delta \xrightarrow{\Delta'} \omega\Delta'\) and \(\Delta' \xrightarrow{\text{Act}^c}\). By the definition of \(V\), \(\min(V(\Delta')) = 0\). Using (4.5), we have \(\min(V(\Delta)) \leq \min(V(\Delta'))\), so \(\min(V(\Delta)) = 0\) as well. Thus, also in this case \(\min(V(s)) \geq \min(V(\Delta))\).

Theorem 4.11. If \(P \subseteq_{\text{FS}} Q\) then \(P \preceq_{\text{pmust}} Q\).

Proof. Similar to the proof of Theorem 4.7 using (4.4)–(4.6). \(\square\)

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 \([6, 11, 8]\) uses success states marked by outgoing \(\omega\)-actions; this is referred to as state-based testing, which we have used in Section 3 to define the preorders \(\preceq_{\text{may}}\) and \(\preceq_{\text{must}}\). In other work \([37, 10]\), however, it is the actual execution of \(\omega\) that constitutes success. This action-based approach is formalised as in the state-based approach, via a modified results-gathering function:
\[
\hat{V}(s) := \begin{cases} 
\bigcup \{V(\Delta) \mid s \xrightarrow{\alpha} \Delta \land \alpha \neq \omega\} \cup \{1 \mid s \xrightarrow{\omega} \} & \text{if } s \xrightarrow{} \\
\{0\} & \text{otherwise}
\end{cases}
\]

As in the original \(V\), the \(\alpha\)'s are non-success actions, including \(\tau\); and again, this is done for generality, since in testing outcomes the only non-success action is \(\tau\).

If we use this results-gathering function rather than \(V\) in Definitions 3.1 and 3.2 we obtain the two slightly different testing preorders, \(\hat{\preceq}_{\text{pmust}}\) and \(\hat{\preceq}_{\text{pmust}}\). The following proposition shows that state-based testing is at least as discriminating as action-based testing:
Proposition 5.1.

1. If \( P \sqsubseteq_{\text{pmay}} Q \) then \( P \hat{\sqsubseteq}_{\text{pmay}} Q \).
2. If \( P \sqsubseteq_{\text{pmust}} Q \) then \( P \hat{\sqsubseteq}_{\text{pmust}} Q \).

Proof. For any action-based test \( \hat{T} \) we construct a state-based test \( T \) by replacing each subterm \( \omega.Q \) by \( \tau.\omega \); then we have \( V[T|\text{Act } P] = \hat{V}[\hat{T}|\text{Act } P] \) 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 \( P \hat{\sqsubseteq}_{\text{pmay}} Q \) then \( P \sqsubseteq_{\text{S}} Q \).
2. If \( P \hat{\sqsubseteq}_{\text{pmust}} Q \) then \( P \sqsubseteq_{\text{FS}} Q \).

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 \( \Omega \), to model divergence, and the operational semantics of Figure 1 with the rule

\[
\Omega \xrightarrow{\tau} \Omega.
\]

It is possible to extend the results-gathering functions \( V \) and \( \hat{V} \) 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 \( a.\Omega \notin_{\text{pmust}} a.\Omega \cap 0 \) because of the test \( a.\omega \):

\[
V([a.\omega|\text{Act } a.\Omega]) = \{1\} \text{ while } V([a.\omega|\text{Act } a.\Omega \cap 0]) = \{0,1\}.
\]

This intuitively is due to the fact that the \( \Omega \)-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 \( \tau \)-branches. We have

\[
\hat{V}([a.\omega|\text{Act } a.\Omega]) = \hat{V}([a.\omega|\text{Act } a.\Omega \cap 0]) = \{0,1\}.
\]

Indeed no test can be found to distinguish them, and so one can show \( a.\Omega \hat{\sqsubseteq}_{\text{pmust}} a.\Omega \cap 0 \).

Note that probabilistic behaviour plays no role in this counter-example. In CSP (without probabilities) there is no difference between \( \hat{\sqsubseteq}_{\text{may}} \) and \( \sqsubseteq_{\text{may}} \), whereas \( \hat{\sqsubseteq}_{\text{must}} \) 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 \( \Omega \) be a set of fresh success actions with \( \Omega \cap \text{Act } \tau = \emptyset \). An \( \Omega \)-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-\textit{tuples} $\hat{\mathcal{A}}^\Omega(T, P) \subseteq [0, 1]^\Omega$. As with standard scalar testing, each outcome arises from a resolution of the nondeterministic choices in $T \upharpoonright \text{Act} \ P$. However, here an outcome is a \textit{tuple} and its $\omega$-component gives the probability that this resolution will perform the success action $\omega$.

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

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

so that if $\alpha$ is a success action, in $\Omega$, 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 \textit{convex closure} $\downarrow X$ by

$$
\downarrow X := \{ \sum_{i \in I} p_i o_i \mid p \in D(I) \text{ and } o : I \rightarrow X \}.
$$

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

**Definition 6.1.** The \textit{action-based, vector-based, convex-closed results-gathering function} $\hat{\mathcal{V}}^\Omega_1 : p\text{CSP}^\Omega \rightarrow \mathcal{P}^+(0, 1]^\Omega)$ is given by

$$
\hat{\mathcal{V}}^\Omega_1(s) := \left\{ \downarrow \cup \{ \alpha!(\hat{\mathcal{V}}^\Omega_1(\Delta)) \mid s \xrightarrow{\alpha} \Delta, \alpha \in \Omega \cup \text{Act} \} \right\} \text{ if } s \rightarrow \\
\{ \bar{0} \} \quad \text{otherwise}
$$

As with our previous results-gathering functions $\mathcal{V}$ and $\hat{\mathcal{V}}$, this function extends to the type $\mathcal{D}(s\text{CSP}^\Omega) \rightarrow \mathcal{P}^+(0, 1]^\Omega)$ via the convention $\hat{\mathcal{V}}^\Omega_1(\Delta) := \text{Exp}_\Delta \hat{\mathcal{V}}^\Omega_1$.

For any $p\text{CSP}$ process $P$ and $\Omega$-test $T$, let

$$
\hat{\mathcal{A}}^\Omega_1(T, P) := \hat{\mathcal{V}}^\Omega_1[T \upharpoonright \text{Act} \ P] .
$$

The \textit{vector-based may-} and \textit{must preorders} are given by

$$
P \overset{\text{may}}{\succeq} Q \text{ if and only if } \text{ for all } \Omega\text{-tests } T: \hat{\mathcal{A}}^\Omega_1(T, P) \leq_{\text{Ho}} \hat{\mathcal{A}}^\Omega_1(T, Q) \quad P \overset{\text{must}}{\succeq} Q \text{ if and only if } \text{ for all } \Omega\text{-tests } T: \hat{\mathcal{A}}^\Omega_1(T, P) \leq_{\text{Sm}} \hat{\mathcal{A}}^\Omega_1(T, Q)
$$

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

We will explain the rôle of convex-closure $\downarrow$ in this definition. Let $\hat{\mathcal{V}}^\Omega$ be defined as $\hat{\mathcal{V}}^\Omega_1$ above, but omitting the use of $\downarrow$. It is easy to see that $\hat{\mathcal{V}}^\Omega_1(s) = \downarrow \hat{\mathcal{V}}^\Omega(s)$ for all $s \in s\text{CSP}^\Omega$.

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) $X \leq_{\text{Ho}} Y$ if and only if $\downarrow X \leq_{\text{Ho}} \downarrow Y$.

(2) $X \leq_{\text{Sm}} Y$ if and only if $\downarrow X \leq_{\text{Sm}} \downarrow Y$.

**Proof.** We restrict attention to (1); the proof of (2) goes likewise. It suffices to show that (i) $X \leq_{\text{Ho}} \downarrow X$ and (ii) $\downarrow X \leq_{\text{Ho}} X$. We only prove (ii) since (i) is obvious. Suppose $x \in \downarrow X$, 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
\[ x = \sum_{i \in I} p_i x_i \leq \sum_{i \in I} p_i x^* = x^* \in X. \]

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 \( \Omega \) is the singleton set \( \{ \omega \} \). Then

1. \( P \overset{\Omega}{\preceq}_{\text{pmay}} Q \) if and only if \( P \overset{\Omega}{\preceq}_{\text{pmay}} Q \).
2. \( P \overset{\Omega}{\preceq}_{\text{pmust}} Q \) if and only if \( P \overset{\Omega}{\preceq}_{\text{pmust}} Q \).

**Proof.** \( \widehat{\mathcal{V}}_{\Omega} = \mathcal{V}_{\Omega} = \mathcal{V} \) when \( \Omega = \{ \omega \} \), so the result follows from Lemma 6.2.

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 \( \mathcal{V} \leq_{\text{Ho}} \mathcal{V} \) although obviously \( X \not\leq_{\text{Ho}} Y \).

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 \( P_a \oplus b \) and \( Q = a \cap b \).

Take \( \Omega = \{ \omega_1, \omega_2 \} \). Employing the results-gathering function \( \widehat{\mathcal{V}}_{\Omega} \), without convex closure, with the test \( T := a.\omega_1 \boxdot b.\omega_2 \) we obtain
\[
\widehat{\mathcal{A}}_{\Omega}(T, P) = \{(0.5, 0.5)\} \quad \widehat{\mathcal{A}}_{\Omega}(T, Q) = \{(1, 0), (0, 1)\}.
\]

As pointed out in Example 6.4, this entails \( \widehat{\mathcal{A}}_{\Omega}(T, P) \not\leq_{\text{Ho}} \widehat{\mathcal{A}}_{\Omega}(T, Q) \), although their convex closures \( \widehat{\mathcal{A}}_{\Omega}(T, P) \) and \( \widehat{\mathcal{A}}_{\Omega}(T, Q) \) 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 \( P \preceq_S Q \), and from Theorem 4.7 it therefore follows that \( P \preceq_{\text{pmay}} Q \). 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 \( \overset{\Omega}{\preceq}_{\text{pmay}} \) as a stepping stone in showing the coincidence of \( \preceq_S \) and \( \preceq_{\text{pmay}} \), we must have \( P \overset{\Omega}{\preceq}_{\text{pmay}} Q \). For this reason we use convex closure in Definition 6.1.

In [10] the results-gathering function \( \widehat{\mathcal{V}}_{\Omega} \) with \( \Omega = \{ \omega_1, \omega_2, \cdots \} \) was called simply \( \mathcal{V} \) (because action-based/vector-based/convex-closed testing was assumed there throughout, making the \( \overset{\Omega}{\preceq}_{\text{pmay}} \)-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. \( P \overset{\Omega}{\preceq}_{\text{pmay}} Q \) iff \( P \overset{\Omega}{\preceq}_{\text{pmay}} Q \).
2. \( P \overset{\Omega}{\preceq}_{\text{pmust}} Q \) iff \( P \overset{\Omega}{\preceq}_{\text{pmust}} Q \).
Proof. In [10, Theorem 3] this theorem has been established for versions of \( \exists \pi \text{may} \) and \( \exists \pi \text{must} \) where tests are finite probabilistic automata, as defined in our Appendix A. The key argument is that when \( P \exists \pi \text{may} Q \) can be refuted by means of a vector-based test \( T \), then \( P \exists \pi \text{may} Q \) 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 \( \exists \pi \text{must} \). This theorem applies to our setting as well, due to the observation that if a test \( T \) can be represented as a pCSP-\( \Omega \)-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

1. \( P \exists \pi \text{may} Q \) implies \( P \subseteq_S Q \) and
2. \( P \exists \pi \text{must} Q \) implies \( P \subseteq_{FS} Q \).

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 \( \hat{A}^\Omega_i \) needed for the above implications are summarised in Lemmas 6.7 and 6.8. For convenience of presentation, we write \( \bar{\omega} \) for the vector in \([0,1]^{\Omega}\) defined by \( \bar{\omega}(\omega) = 1 \) and \( \bar{\omega}(\omega') = 0 \) for \( \omega' \neq \omega \). Sometimes we treat a distribution \( \Delta \) of finite support as the pCSP expression \( \bigoplus_{s\in[\Delta]} \Delta(s) \cdot s \), so that \( \hat{A}^\Omega_i (T, \Delta) := \text{Exp}_{\Delta} \hat{A}^\Omega_i (T, \bar{\omega}) \).

Lemma 6.7. Let \( P \) be a pCSP process, and \( T, T_i \) be tests.

1. \( o \in \hat{A}^\Omega_i (\omega, P) \) iff \( o = \bar{\omega} \).
2. \( \bar{\omega} \in \hat{A}^\Omega_i (\bigwedge_{X \in \Delta} \omega, P) \) iff \( \exists \Delta : [P] \xrightarrow{\Delta} X \).
3. Suppose the action \( \omega \) does not occur in the test \( T \). Then \( o \in \hat{A}^\Omega_i (\omega \cdot a.T, P) \) with \( o(\omega) = 0 \) iff there is a \( \Delta \in \mathcal{D}(sCSP) \) with \( [P] \xrightarrow{\Delta} X \) and \( o \in \hat{A}^\Omega_i (T, \Delta) \).
4. \( o \in \hat{A}^\Omega_i (\bigoplus_{i\in I} p_i.T_i, P) \) iff \( o = \sum_{i\in I} p_i o_i \) for some \( o_i \in \hat{A}^\Omega_i (T_i, P) \).
5. \( o \in \hat{A}^\Omega_i (\bigcap_{i\in I} T_i, P) \) if for all \( i \in I \) there are \( q_i \in [0,1] \) and \( \Delta_i \in \mathcal{D}(sCSP) \) such that \( \sum_{i\in I} q_i = 1 \), \( [P] \xrightarrow{\Delta} \sum_{i\in I} q_i \cdot \Delta_i \) and \( o = \sum_{i\in I} o_i \) for some \( o_i \in \hat{A}^\Omega_i (T_i, \Delta_i) \).

Proof. Straightforward, by induction on the structure of \( P \).

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 \( o \in \hat{A}^\Omega_i (\bigcap_{i\in I} T_i, P) \) then for all \( i \in I \) there are \( q_i \in [0,1] \) and \( \Delta_i \in \mathcal{D}(sCSP) \) with \( \sum_{i\in I} q_i = 1 \) such that \( [P] \xrightarrow{\Delta} \sum_{i\in I} q_i \cdot \Delta_i \) and \( o = \sum_{i\in I} o_i \) for some \( o_i \in \hat{A}^\Omega_i (T_i, \Delta_i) \).

Proof. Given that the states of our pLTSs are sCSP expressions, there exists a well-founded order on the combination of states in sCSP and distributions in \( \mathcal{D}(sCSP) \), such that \( s \xrightarrow{\Delta} \Delta \) implies that \( s \) is larger than \( \Delta \), 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 \( T = \bigcap_{i\in I} T_i \). We prove the following two claims

(a) If \( s \) is a state-based process and \( o \in \hat{A}^\Omega_i (T, s) \) then there are some \( \{ q_i \}_{i\in I} \) with \( \sum_{i\in I} q_i = 1 \) such that \( s \xrightarrow{\Delta} \sum_{i\in I} q_i \cdot \Delta_i \) and \( o = \sum_{i\in I} q_i o_i \) for some \( o_i \in \hat{A}^\Omega_i (T_i, \Delta_i) \).
(b) If \( \Delta \in \mathcal{D}(sCSP) \) and \( o \in \hat{A}^\Omega_i (T, \Delta) \) then there are some \( \{ q_i \}_{i\in I} \) with \( \sum_{i\in I} q_i = 1 \) such that \( \Delta \xrightarrow{\Delta} \sum_{i\in I} q_i \cdot \Delta_i \) and \( o = \sum_{i\in I} q_i o_i \) for some \( o_i \in \hat{A}^\Omega_i (T_i, \Delta_i) \).
by simultaneous induction on the order mentioned above, applied to $s$ and $\Delta$.

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

- If $s$ cannot make a $\tau$-move, that is $s \not\xrightarrow{\tau}$, then the only possible moves from $T|_{\text{Act}}s$ are $\tau$-moves originating in $T$; $T$ has no non-$\tau$ moves, and any non-$\tau$ moves that might be possible for $s$ on its own are inhibited by the alphabet $\text{Act}$ of the composition. Suppose $o \in \hat{A}_1^\Omega(T, s)$. 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 $o_i \in A_1^\Omega(T_i, s) = A_1^\Omega(T, s)$. Obviously we also have $[s] \xrightarrow{\tau} \sum_{i \in I} q_i \cdot \overline{s}$.

- If $s$ can make one or more $\tau$-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 $T|_{\text{Act}}s$ are $\tau$-moves either of $T$ or of $s$, because $T$ cannot make initial non-$\tau$ moves and that prevents a proper synchronisation from occurring on the first step. Suppose that $o \in \hat{A}_1^\Omega(T, s)$. 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

$$o = \sum_{k \in I \cup J} p_k o_k' \quad (6.2)$$

$$o_i' \in \hat{A}_1^\Omega(T_i, s) \quad \text{for all } i \in I \quad (6.3)$$

$$o_j' \in \hat{A}_1^\Omega(T, \Delta_j) \quad \text{for all } j \in J. \quad (6.4)$$

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

$$\Delta_j' \Rightarrow \sum_{i \in I} p_{ji} \cdot \Delta_{ji}' \quad (6.5)$$

$$o_j' = \sum_{i \in I} p_{ji} o_{ji}' \quad (6.6)$$

$$o_{ji}' \in \hat{A}_1^\Omega(T_i, \Delta_{ji}') \quad (6.7)$$

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 $\Delta_i$ and $o_i$ are chosen arbitrarily in case $q_i = 0$. It can be checked by arithmetic that $q_i, \Delta_i, o_i$ have the required properties, viz. that
\[
\sum_{i \in I} q_i = 1, \text{ that } o = \sum_{i \in I} q_i o_i \text{ and that }
\]
\[
s \xrightarrow{\tau} \sum_{i \in I} p_i \cdot s + \sum_{j \in J} p_j \cdot \Delta'_j
\]
\[
\xrightarrow{\tau} \sum_{i \in I} p_i \cdot s + \sum_{j \in J} p_j \cdot \sum_{i \in I} p_{ji} \cdot \Delta'_{ji}
\]
\[
= \sum_{i \in I} q_i \cdot \Delta_i.
\]

Finally, it follows from (6.3) and (6.7) that \( o_i \in \hat{A}_i^\Omega(T, \Delta_i) \) for each \( i \in I \).

(b) Let \( |\Delta| = \{s_j\}_{j \in J} \) 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 \( \hat{A}_i^\Omega(T, \Delta) := \text{Exp}_\Delta \hat{A}_i^\Omega(T, \bot) \), if \( o \in \hat{A}_i^\Omega(T, \Delta) \) then
\[
o'_j \in \hat{A}_i^\Omega(T, s_j)
\]
For each \( j \in J \), we know by the induction hypothesis that
\[
s_j \xrightarrow{\tau} \sum_{i \in I} q_{ji} \cdot \Delta'_{ji}
\]
\[
o'_j = \sum_{i \in I} q_{ji} o'_{ji}
\]
\[
o'_{ji} \in \hat{A}_i^\Omega(T, \Delta'_{ji})
\]
for some \( \{q_{ji}\}_{i \in I} \) with \( \sum_{i \in I} q_{ji} = 1 \). Thus let
\[
q_i = \sum_{j \in J} r_{ji} q_{ji}
\]
\[
\Delta_i = \frac{1}{q_i} \sum_{j \in J} r_{ji} q_{ji} \cdot \Delta'_{ji}
\]
\[
o_i = \frac{1}{q_i} \sum_{j \in J} r_{ji} q_{ji} o'_{ji}
\]
again choosing \( \Delta_i \) and \( o_i \) arbitrarily in case \( q_i = 0 \). As in the first case, it can be shown by arithmetic that the collection \( r_i, \Delta_i, o_i \) has the required properties. \( \square \)

7. Modal logic

In this section we present logical characterisations \( \sqsubseteq^L \) and \( \sqsubseteq^F \) 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:

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

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

$\bullet \; \Delta \models \text{ref}(X)$ iff there is a $\Delta'$ with $\Delta \xrightarrow{\Delta} \Delta'$ and $\Delta' \not\models \psi$

$\bullet \; \Delta \models \langle a \rangle \varphi$ iff there is a $\Delta'$ with $\Delta \xrightarrow{\Delta} \Delta'$ and $\Delta' \models \varphi$

$\bullet \; \Delta \models \bigwedge_{i \in I} \varphi_i$ 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 $\Delta_i \in \mathcal{D} (\mathcal{sCSP})$, for all $i \in I$, with $\Delta_i \models \varphi_i$, such that $\Delta \models \sum_{i \in I} p_i \cdot \Delta_i$.

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

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 $\varphi_s$ or $\varphi_\Delta$ of a process $s \in \mathcal{sCSP}$ or $\Delta \in \mathcal{D} (\mathcal{sCSP})$ is defined inductively:

$\varphi_\Delta := \bigwedge_{s \xrightarrow{s \rightarrow \Delta} (a) \varphi_\Delta \land \text{ref} (\{a \mid s \not\models \varphi\})}$ if $s \not\models \varphi$,

$\varphi_s := \bigwedge_{s \xrightarrow{s \rightarrow \Delta} (a) \varphi_\Delta \land \bigwedge_{s \xrightarrow{s \rightarrow \Delta} \varphi_\Delta}$ otherwise,

$\varphi_\Delta := \bigoplus_{s \in [\Delta]} \varphi_s \cdot \varphi_s$.

Here the conjunctions $\bigwedge_{s \xrightarrow{s \rightarrow \Delta}$ range over suitable pairs $a, \Delta$, and $\bigwedge_{s \xrightarrow{s \rightarrow \Delta}$ ranges over suitable $\Delta$. The $\mathcal{L}$-characteristic formulae $\psi_s$ and $\psi_\Delta$ are defined likewise, but omitting the conjuncts ref (\{a \mid s \not\models \varphi\}).

Write $\varphi \Rightarrow \psi$ with $\varphi, \psi \in \mathcal{F}$ if for each distribution $\Delta$ one has $\Delta \models \varphi$ implies $\Delta \models \psi$. Then it is easy to see that $\varphi_\Delta \Leftrightarrow \varphi_s$ and $\bigwedge_{i \in I} \varphi_i \Rightarrow \varphi_i$ for any $i \in I$; furthermore, the following property can be established by an easy inductive proof.

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

It and the following lemma help to establish Theorem 7.3.

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

**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:

$\Theta \models \varphi_\Delta$ implies $\exists \Theta' : \Theta \xrightarrow{\delta} \Theta' \lor \Delta \not\models \mathcal{R} \Theta'$.

(7.1)

Suppose $\Theta \models \varphi_\Delta$ with $\varphi_\Delta = \bigoplus_{i \in I} p_i \cdot \varphi_s$, so that we have $\Delta = \sum_{i \in I} p_i \cdot \varphi_i$, and for all $i \in I$ there are $\Theta_i \in \mathcal{D} (\mathcal{sCSP})$ with $\Theta_i \models \varphi_s$ such that $\Theta \xrightarrow{\delta} \Theta'$ 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 $\Delta \not\models \mathcal{R} \Theta'$.

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

• Suppose $s \mathcal{R} \Theta$ and $s \xrightarrow{\delta} \Delta$. Then from Definition 7.1 we have $\varphi_s \models \varphi_\Delta$, so that $\Theta \models \varphi_\Delta$. Applying (7.1) gives us $\Theta \xrightarrow{\delta} \Theta'$ with $\Delta \not\models \mathcal{R} \Theta'$ for some $\Theta'$.

• Suppose $s \mathcal{R} \Theta$ and $s \xrightarrow{a \rightarrow \Delta}$ with $a \in \text{Act}$. Then $\varphi_s \models \langle a \rangle \varphi_\Delta$, so $\Theta \models \langle a \rangle \varphi_\Delta$. Hence $\exists \Theta'$ with $\Theta \not\models \Theta'$ and $\Theta' \models \varphi_\Delta$. Again apply (7.1).

• Suppose $s \mathcal{R} \Theta$ and $s \xrightarrow{X \not\models} \Delta$ with $X \subseteq A$. Then $\varphi_s \models \text{ref}(X)$, so $\Theta \models \text{ref}(X)$. Hence $\exists \Theta'$ with $\Theta \not\models \Theta'$ and $\Theta' \not\models \text{ref}(X)$.
Thus \( R \) is indeed a failure simulation. By our assumption \([P] \models \varphi,Q\) using (7.1), there exists a \( \Theta' \) such that \([P] \xrightarrow{\Delta} \Theta' \) and \([Q] \xrightarrow{R} \Theta'\), which gives \( P \subseteq_{FS} Q \) via Definition 4.3.

To establish the second statement, define the relation \( S \) by \( s \Theta \iff \Theta \models \psi_s \); exactly as above one obtains

\[
\Theta \models \psi_{\Delta} \iff \exists \Theta' : \Theta \xrightarrow{\Delta} \Theta' \wedge S(\Theta'). \tag{7.2}
\]

Just as above it follows that \( S \) is a simulation. By the assumption \([Q] \models \varphi_P\), using (7.2), there exists a \( \Theta' \) such that \([Q] \xrightarrow{\Delta} \Theta' \) and \([P] \xrightarrow{S} \Theta'\). Hence \( P \subseteq S Q \) via Definition 4.3. \( \square \)

**Theorem 7.4.**

(1) If \( P \subseteq \mathcal{C} Q \) then \( P \subseteq S Q \).

(2) If \( P \subseteq \mathcal{F} Q \) then \( P \subseteq_{FS} Q \).

**Proof.** Suppose \( P \subseteq \mathcal{C} Q \). By Lemma 7.2 we have \([Q] \models \varphi_Q\) and hence \([P] \models \varphi_Q\). Lemma 7.3 gives \( P \subseteq_{FS} Q \).

For (1), assuming \( P \subseteq \mathcal{C} Q \), we have \([P] \models \psi_P\), hence \([Q] \models \psi_P\), and thus \( P \subseteq S Q \). \( \square \)

8. Characteristic tests

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

**Lemma 8.1.** For every \( \varphi \in \mathcal{F} \) there exists a pair \((T_\varphi,v_\varphi)\) with \( T_\varphi \) an \( \Omega \)-test and \( v_\varphi \in [0,1]^\Omega \), such that

\[
\Delta \models \varphi \iff \exists \omega \in \hat{\mathcal{A}}_r^\Omega(T_\varphi,\Delta) : \omega \leq v_\varphi \tag{8.1}
\]

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

\[
\Delta \models \varphi \iff \exists \omega \in \hat{\mathcal{A}}^\Omega(T_\varphi,\Delta) : \omega \geq v_\varphi . \tag{8.2}
\]

\( T_\varphi \) is called a characteristic test of \( \varphi \) and \( v_\varphi \) its target value.

**Proof.** First of all note that if a pair \((T_\varphi,v_\varphi)\) satisfies the requirements above, then any pair obtained from \((T_\varphi,v_\varphi)\) by bijectively renaming the elements of \( \Omega \) 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 \( (\text{the finite}) \, T_\varphi \). Moreover, any countable collection of characteristic tests can be assumed to be \( \Omega \)-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 \( v_\varphi := \bar{\omega} \).
- Let \( \varphi = \text{ref}(X) \) with \( X \subseteq \text{Act} \). Take \( T_\varphi := \square_{a \in X} \omega \) for some \( \omega \in \Omega \), and \( v_\varphi := \bar{\omega} \).
- Let \( \varphi = \langle \alpha \rangle \psi \). By induction, \( \psi \) has a characteristic test \( T_\psi \) with target value \( v_\psi \). Take \( T_\varphi := \omega \square_a T_\psi \) where \( \omega \in \Omega \) does not occur in \( T_\psi \), and \( v_\varphi := v_\psi \).
- Let \( \varphi = \bigwedge_{i \in I} \varphi_i \) with \( I \) a finite and non-empty index set. Choose a \( \Omega \)-disjoint family \((T_i,v_i)_{i \in I}\) of characteristic tests \( T_i \) with target values \( v_i \) for each \( \varphi_i \). Furthermore, let \( p_i \in (0,1] \) for \( i \in I \) be chosen arbitrarily such that \( \sum_{i \in I} p_i = 1 \). Take \( T_\varphi := \bigoplus_{i \in I} p_i T_i \) and \( v_\varphi := \sum_{i \in I} p_i v_i \).
• Let $\varphi = \bigoplus_{i \in I} p_i \cdot \varphi_i$. Choose a $\Omega$-disjoint family $(T_i, v_i)_{i \in I}$ of characteristic tests $T_i$ with target values $v_i$ for each $\varphi_i$, such that there are distinct success actions $\omega_i$ for $i \in I$ that do not occur in any of those tests. Let $T'_i := T_i \cup \omega_i$ and $v'_i := \frac{1}{2} v_i + \frac{1}{2} \omega_i$. Note that for all $i \in I$ also $T'_i$ is a characteristic test of $\varphi_i$ 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 $T_{\varphi}$. By induction on $\varphi$ we now check (S.1) above.

• Let $\varphi = \top$. For all $\Delta \in \mathcal{D}(s\text{CSP})$ we have $\Delta \models \varphi$ as well as $\exists o \in \tilde{A}_i(\top, \Delta) : o \leq v_{\varphi}$, using Lemma 6.7(1).

• Let $\varphi = \text{ref}(X)$ with $X \subseteq \text{Act}$. Suppose $\Delta \models \varphi$. Then there is a $\Delta'$ with $\Delta \overset{\varphi}{\rightarrow} \Delta'$ and $\Delta' \overset{\varphi}{\rightarrow} \Delta$. By Lemma 6.7(2), $\exists o \in \tilde{A}_i(\top, \Delta)$.

Now suppose $\exists o \in \tilde{A}_i(\top, \Delta) : o \leq v_{\varphi}$. This implies $o = \hat{o}$, so by Lemma 6.7(2) there is a $\Delta'$ with $\Delta \overset{\varphi}{\rightarrow} \Delta'$ and $\Delta' \overset{\varphi}{\rightarrow} \Delta$. Hence $\Delta \models \varphi$.

• Let $\varphi = (a)\psi$ with $a \in \text{Act}$. Suppose $\Delta \models \varphi$. Then there is a $\Delta'$ with $\Delta \overset{\varphi}{\rightarrow} \Delta'$ and $\Delta' \models \psi$. By induction, $\exists o \in \tilde{A}_i(\top, \Delta') : o \leq v_{\varphi}$. By Lemma 6.7(3), $o \in \tilde{A}_i(\top, \Delta)$.

Now suppose $\exists o \in \tilde{A}_i(\top, \Delta) : o \leq v_{\varphi}$. This implies $o(\omega) = 0$, so by Lemma 6.7(3) there is a $\Delta'$ with $\Delta \overset{\varphi}{\rightarrow} \Delta'$ and $\exists o \in \tilde{A}_i(\top, \Delta')$. 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, $\exists o_i \in \tilde{A}_i(\top, \Delta) : o_i \leq v_{\varphi_i}$. Thus $o := \sum_{i \in I} p_i o_i \in \tilde{A}_i(\top, \Delta)$ by Lemma 6.7(4), and $o \leq v_{\varphi}$.

Now suppose $\exists o \in \tilde{A}_i(\top, \Delta) : o \leq v_{\varphi}$. Then, using Lemma 6.7(4), $o = \sum_{i \in I} p_i o_i$ for certain $o_i \in \tilde{A}_i(\top, \Delta)$. Note that $(T_i)_{i \in I}$ is an $\Omega$-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 $\omega$ 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 $\Delta_i \in \mathcal{D}(p\text{CSP})$ with $\Delta_i \models \varphi_i$, such that $\Delta \overset{\varphi}{\rightarrow} \sum_{i \in I} p_i \cdot \Delta_i$. By induction, there are $o_i \in \tilde{A}_i(\top, \Delta_i)$ with $o_i \leq v_i$. Hence, there are $o'_i \in \tilde{A}_i(\top, \Delta_i)$ with $o'_i \leq v'_i$. Thus $o := \sum_{i \in I} p_i o'_i \in \tilde{A}_i(\top, \Delta)$ by Lemma 6.7(5), and $o \leq v_{\varphi}$.

Now suppose $\exists o \in \tilde{A}_i(\top, \Delta) : o \leq v_{\varphi}$. Then, by Lemma 6.8 there are $q \in \mathcal{D}(I)$ and $\Delta_i$, for $i \in I$, such that $\Delta \overset{\varphi}{\rightarrow} \sum_{i \in I} q_i \cdot \Delta_i$ and $o = \sum_{i \in I} q_i o'_i$ for some $o'_i \in \tilde{A}_i(T'_i, \Delta_i)$. Now $\forall i : o'_i(\omega_i) = v'_i(\omega_i) = \frac{1}{2}$, so, using that $(T_i)_{i \in I}$ is an $\Omega$-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 $T'_i = T_i \cup \omega_i$, using Lemma 6.7(4), it must be that $o' = \frac{1}{2} o_i + \frac{1}{2} \omega_i$ for some $o_i \in \tilde{A}_i(\top, \Delta_i)$ 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 $\text{ref}(X)$. Then a straightforward induction yields that $\sum_{\omega \in \Omega} v_{\varphi}(\omega) = 1$ and for all $\Delta \in \mathcal{D}(p\text{CSP})$ and $o \in \tilde{A}_i(\top, \Delta)$ we have $\sum_{\omega \in \Omega} o(\omega) = 1$. Therefore, $o \leq v_{\varphi}$ iff $o \geq v_{\varphi}$, yielding (S.2). \qed

**Theorem 8.2.**

1. If $P \preceq_{\text{bmay}} Q$ then $P \preceq_{\mathcal{L}} Q$.
2. If $P \preceq_{\text{bmust}} Q$ then $P \preceq_{\mathcal{F}} Q$. 


(P1) \[ P_p \oplus P = P \]

(P2) \[ P_p \oplus Q = Q_1 \oplus P \]

(P3) \[ (P_p \oplus Q)_q \oplus R = P_p \oplus (Q \oplus R) \]

(I1) \[ P \cap P = P \]

(I2) \[ P \cap Q = Q \cap P \]

(I3) \[ (P \cap Q) \cap R = P \cap (Q \cap R) \]

(E1) \[ P \cap 0 = P \]

(E2) \[ P \cap Q = Q \cap P \]

(E3) \[ (P \cap Q) \cap R = P \cap (Q \cap R) \]

(E1) \[ a.P \cap a.Q = a.P \cap a.Q \]

(D1) \[ P \cap (Q_p \oplus R) = (P \cap Q)_p \oplus (P \cap R) \]

(D2) \[ a.P \cap (Q \cap R) = (a.P \cap Q) \cap (a.P \cap R) \]

(D3) \[ (P_1 \cap P_2) \cap (Q_1 \cap Q_2) = (P_1 \cap (Q_1 \cap Q_2)) \cap (P_2 \cap (Q_1 \cap Q_2)) \]

\[ \cap ((P_1 \cap P_2) \cap Q_1) \cap ((P_1 \cap P_2) \cap Q_2) \]

Figure 4: Common equations

Proof. Suppose \( P \preceq o \) \( p_{\text{must}} Q \) and \([Q] \models \varphi \) for some \( \varphi \in \mathcal{F} \). Let \( T_\varphi \) be a characteristic test of \( \varphi \) with target value \( v_\varphi \). Then Lemma 8.1 yields \( \exists o \in A_1^0(T_\varphi, [Q]) : o \subseteq v_\varphi \), and hence, given that \( P \preceq o \) \( p_{\text{must}} Q \) and \( A_1^0(T_\varphi, [R]) = A_1^0(T_\varphi, R) \) for any \( R \in \text{pCSP} \), by the Smyth preorder we have \( \exists o' \in A_1^0(T_\varphi, [P]) : o' \subseteq v_\varphi \). Thus \([P] \models \varphi \).

The may-case goes likewise, via the Hoare preorder. \( \square \)

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 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 \( \subseteq_S \) and \( \subseteq_{FS} \). \( \square \)

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 \( \text{pCSP} \) processes that do not use the parallel operator \( \parallel A \); we call the resulting sub-language \( \text{nCSP} \). For a brief discussion of the axiomatisation for terms involving \( \parallel A \) and the other parallel operators commonly used in \( \text{CSP} \) see Section 12.

Let us write \( \equiv F Q \) for equivalences that can be derived using the equations given in Figure 4. Given the way we defined the syntax of \( \text{pCSP} \), 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 are missing; they are not sound for \( \simeq_{FS} \). Typical examples include:

\[
\begin{align*}
\text{a.} & \quad (P \sqcap Q) = a.P \sqcap a.Q \\
\text{b.} & \quad P = P \Box P \\
\text{c.} & \quad P \Box (Q \sqcap R) = (P \Box Q) \sqcap (P \Box R) \\
\text{d.} & \quad P \sqcap (Q \sqcup R) = (P \sqcap Q) \sqcup (P \sqcap R)
\end{align*}
\]

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 \( P \simeq_{FS} Q \).

**Proof.** Because of Proposition 4.6, that \( \Box_{FS} \) 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:

\[
N ::= N_1 \bigoplus N_2 \mid N_1 \sqcap N_2 \mid \bigboxtimes_{i \in I} a_i.N_i
\]

**Proposition 9.3.** For every \( P \in nCSP \) 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 \( P =_{prob} Q \) denote that equivalence of \( P \) and \( Q \) can be derived using those axioms alone. Then we have the following property.

**Lemma 9.4.** Let \( P, Q \in nCSP \). Then \([P] = [Q]\) implies \( P =_{prob} Q \).

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 \( s_i \in sCSP \) are all different.
May:

\( \text{(May0)} \quad a.P \quad \square \quad b.Q = a.P \quad \square \quad b.Q \)

\( \text{(May1)} \quad P \quad \sqsubseteq \quad P \quad \square \quad Q \)

\( \text{(May2)} \quad 0 \quad \sqsubseteq \quad P \)

\( \text{(May3)} \quad a.(P \quad \oplus \quad Q) \quad \sqsubseteq \quad a.P \quad \oplus \quad a.Q \)

Must:

\( \text{(Must1)} \quad P \quad \cap \quad Q \quad \sqsubseteq \quad Q \)

\( \text{(Must2)} \quad R \quad \cap \quad \bigoplus_{i \in I} \bigoplus_{j \in J_i} p_j \cdot (a_i.Q_{ij} \quad \square \quad P_{ij}) \quad \sqsubseteq \quad \bigoplus_{i \in I} \bigoplus_{j \in J_i} a_i \cdot p_j \cdot Q_{ij}, \quad \text{provided} \quad \text{inits}(R) \subseteq \{a_i\}_{i \in I} \)

Figure 5: Inequations

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

\[
\text{inits}(0) = \emptyset \\
\text{inits}(a.P) = \{a\} \\
\text{inits}(P \quad \oplus \quad Q) = \text{inits}(P) \cup \text{inits}(Q) \\
\text{inits}(P \quad \square \quad Q) = \{\tau\} \\
\text{inits}(P \quad \cap \quad Q) = \}
\]

The axiom \textit{(Must2)} can equivalently be formulated as follows:

\[
\bigoplus_{k \in K} \bigoplus_{\ell \in L_k} a_{k\ell}.R_{k\ell} \quad \cap \quad \bigcap_{i \in I} \bigoplus_{j \in J_i} p_j \cdot (a_i.Q_{ij} \quad \square \quad P_{ij}) \quad \sqsubseteq \quad \bigcup_{i \in I} \bigoplus_{j \in J_i} a_i \cdot p_j \cdot Q_{ij}, \quad \text{provided} \quad \{a_{k\ell} \mid k \in K, \ell \in L_k\} \subseteq \{a_i \mid i \in I\}.
\]

This is the case because a term \( R \) satisfies \textit{inits}(\( R \)) \subseteq \{a_i\}_{i \in I} \) iff it can be converted into the form \( \bigoplus_{k \in K} \bigoplus_{\ell \in L_k} a_{k\ell}.R_{k\ell} \) by means of axioms \textit{(D1)}, \textit{(P1)}–\textit{(P3)} and \textit{(E1)}–\textit{(E3)} of Figure 5.

This axiom can also be reformulated in an equivalent but more semantic style:

\[
\text{\textit{(Must2')} \quad R \quad \cap \quad \bigcap_{i \in I} P_i \quad \sqsubseteq \quad \bigcup_{i \in I} a_i.Q_i, \quad \text{provided} \quad [P_i] \xrightarrow{a} [Q_i] \quad \text{and} \quad [R] \xrightarrow{X} \quad \text{with} \quad X = \text{Act}\setminus\{a_i\}_{i \in I}.}
\]

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

\[ a.P \oplus a.Q \models E (a.P \oplus a.Q) \sqsubseteq_E (a.P \oplus Q) \]

where we write \( P \sqsubseteq_E Q \) when \( P \sqsubseteq Q \) is derivable from the resulting inequational theory.

An important inequation that follows from (May1) and (P1) is

\[ (May4) \quad P \oplus Q \models_E P \sqcap Q \]

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

\[ P \oplus Q \models_E (P \sqcap Q) \oplus (P \sqcap Q) \]

Likewise, we have

\[ P \sqcap Q \models_E P \oplus Q \]

**Theorem 10.1.** For \( P, Q \in \text{nCSP} \), it holds that

(i) \( P \sqsubseteq_S Q \) if and only if \( P \sqsubseteq_E \text{must} \ Q \)

(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 3 are sound. For \( \sqsubseteq_S \) this has been done in [3], 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 \( P, Q \in \text{nCSP} \).

(i) If \( [P] \xrightarrow{\Delta} [Q] \) then \( P \sqsubseteq_E \text{must} \ Q \) and \( Q \sqsubseteq_E \text{may} \ P \).

(ii) If \( [P] \xrightarrow{a} [Q] \) 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 \( s \in \text{sCSP} \cap \text{nCSP} \) that \( s \xrightarrow{\Delta} [Q] \) implies \( Q \sqsubseteq_E \text{may} \ s \). So suppose \( s \xrightarrow{\Delta} [Q] \). In case \( s \) has the form \( P_1 \sqcap P_2 \) it follows by the operational semantics of \( \text{pCSP} \) 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 \( s_1 \boxdot s_2 \). In that case there must be a distribution \( \Delta \) such that either \( s_1 \xrightarrow{\Delta} \Delta \) and \( [Q] = \Delta \boxdot s_2 \), or \( s_2 \xrightarrow{\Delta} \Delta \) and \( [Q] = s_1 \boxdot \Delta \). Using symmetry, we may restrict attention to the first case. Let \( R \) be a term such that \( [R] = \Delta \). Then \( [R \boxdot s_2] = \Delta \boxdot s_2 = [Q] \), so Lemma 9.4 yields \( Q = \text{prob} R \boxdot s_2 \). By induction we have \( R \sqsubseteq_E \text{may} s_1 \), hence \( R \boxdot s_2 \sqsubseteq_E \text{may} s_1 \boxdot s_2 \), and thus \( Q \sqsubseteq_E \text{may} s \).

Now we show that \( s \xrightarrow{\Delta} [Q] \) implies \( Q \sqsubseteq_E \text{may} s \). This follows because \( s \xrightarrow{\Delta} [Q] \) means that either \( s \xrightarrow{\Delta} [Q] \) or \( [Q] = \emptyset \), and in the latter case Lemma 9.4 yields \( Q = \text{prob} s \).

Next we show that \( [P] \xrightarrow{a} [Q] \) implies \( Q \sqsubseteq_E \text{may} P \). So suppose \( [P] \xrightarrow{a} [Q] \), that is

\[ [P] = \sum_{i \in I} p_i \cdot [s_i] \quad s_i \xrightarrow{a} [Q_i] \quad [Q] = \sum_{i \in I} p_i \cdot [Q_i] \]
for some $I$, $p_i \in (0, 1]$, $s_i \in \text{sCSP} \cap \text{nCSP}$ and $Q_i \in \text{nCSP}$. Now

1. $[P] = \bigoplus_{i \in I} p_i \cdot s_i$. By Lemma [1.3] 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. $s_i \models [Q_i]$ implies $Q_i \subseteq E_{\text{may}} \cdot s_i$. Therefore, $\bigoplus_{i \in I} p_i \cdot Q_i \subseteq E_{\text{may}} \bigoplus_{i \in I} p_i \cdot s_i$.

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

Finally, the general case, when $[P] \xrightarrow{\tau} \Delta$, 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. $\square$

The completeness result now follows from the following two propositions.

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

**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 $P, Q \in \text{pCSP}$ 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 $P \in \text{sCSP}$ then we have

$$P \triangleleft_S [Q] \implies P \subseteq E_{\text{may}} \cdot Q.$$  \hspace{1cm} (11.1)

There are two cases to consider.

1. $P$ has the form $P_1 \cap P_2$. Since $P_1 \subseteq E_{\text{may}} \cdot P$ we know $P_1 \subseteq S \cdot P \subseteq S \cdot Q$. We use induction to obtain $P_2 \subseteq E_{\text{may}} \cdot Q$, from which the result follows using (I).

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

- $Q$ is $a \cdot Q'$. We know from $a \cdot P' \triangleleft_S [a \cdot Q']$ that $[P'] \triangleleft_S \Theta$ for some $\Theta$ with $[Q'] \xrightarrow{\tau} \Theta$, thus $P' \subseteq S \cdot Q'$. Therefore, we have $P' \subseteq E_{\text{may}} \cdot Q'$ by induction. It follows that $a \cdot P' \triangleleft_S E_{\text{may}} \cdot a \cdot Q'$.

- $Q$ is $\bigcap_{i \in J} a_i \cdot Q_i$ with at least two elements in $J$. We use (May0) and then proceed as in the next case.

- $Q$ is $Q_1 \cap Q_2$. We know from $a \cdot P' \triangleleft_S [Q_1 \cap Q_2]$ that $[P'] \triangleleft_S \Theta$ for some $\Theta$ such that one of the following two conditions holds
  - (a) $[Q_i] \xrightarrow{\tau} \Theta$ for $i = 1$ or 2. In this case, $a \cdot P' \triangleleft_S [Q_i]$, hence $a \cdot P' \subseteq S \cdot Q_i$. By induction we have $a \cdot P' \subseteq E_{\text{may}} \cdot Q_i$; then we apply (May1).
  - (b) $[Q_1] \xrightarrow{\tau} \Theta_1$ and $[Q_2] \xrightarrow{\tau} \Theta_2$ 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 $a \cdot Q'_1 \subseteq E_{\text{may}} \cdot Q_1$ and $a \cdot Q'_2 \subseteq E_{\text{may}} \cdot Q_2$. Clearly, $[Q_1'] \triangleleft S \cdot Q_1'$, thus $P' \subseteq S \cdot Q_1' \cdot a \cdot Q_2'$. By induction, we infer that $P' \subseteq E_{\text{may}} \cdot a \cdot (Q_1' \cdot a \cdot Q_2')$. So

\begin{align*}
a \cdot P' & \subseteq E_{\text{may}} \cdot a \cdot (Q_1' \cdot a \cdot Q_2') \quad \text{(May3)} \\
& \subseteq E_{\text{may}} \cdot a \cdot Q_1' \cdot a \cdot Q_2' \\
& \subseteq E_{\text{may}} \cdot Q_1' \cdot a \cdot Q_2' \\
& \subseteq E_{\text{may}} \cdot a \cdot Q_1' \cdot a \cdot Q_2' \\
& \subseteq E_{\text{may}} \cdot Q_1' \cdot a \cdot Q_2' \\
& \subseteq E_{\text{may}} \cdot Q_1' \cdot a \cdot Q_2' \\
& \subseteq E_{\text{may}} \cdot Q_1' \cdot a \cdot Q_2' \\
& \subseteq E_{\text{may}} \cdot Q_1' \cdot a \cdot Q_2' \quad \text{(May4)}
\end{align*}
\begin{itemize}
    \item $Q$ is $Q_1 \oplus Q_2$. We know from $a.P' \prec_S [Q_1 \oplus Q_2]$ that $[P'] \prec_S \Theta$ for some $\Theta$ such that $[Q_1 \oplus Q_2] \models \Theta$. From Lemma 4.4, we know that $\Theta$ must take the form $p \cdot [Q'_1] + (1 - p) \cdot [Q'_2]$, where $[Q'_i] \models [Q'_i]$ for $i = 1, 2$. Hence $P' \sqsubseteq Q'_1 \oplus Q'_2$, and by induction we get $P' \sqsubseteq Q_{\text{May}}$. Then we can derive $a.P' \sqsubseteq Q_{\text{May}}$ as in the previous case.

Now we use (11.1) to show that $P \sqsubseteq Q$ implies $P \sqsubseteq Q_{\text{May}}$. Suppose $P \sqsubseteq Q$. Applying Definition 4.3 with the understanding that any distribution $\Theta \in D(\text{sCSP})$ can be written as $[Q']$ for some $Q' \in \text{pCSP}$, this means that $[P] \prec_S [Q']$ for some $[Q] \models [Q']$. The Derivative Lemma yields $Q' \sqsubseteq Q_{\text{May}}$. So it suffices to show $P \sqsubseteq Q'$.

We know that $[P] \prec_S [Q']$ means that

$$[P] = \sum_{k \in K} r_k \cdot t_k \quad t_k \prec_S [Q'_k] \quad [Q'] = \sum_{k \in K} r_k \cdot [Q'_k]$$

for some $K$, $r_k \in (0, 1)$, $t_k \in \text{sCSP}$ and $Q'_k \in \text{pCSP}$. Now

1. $[P] = \bigoplus_{k \in K} r_k \cdot t_k$. By Lemma 9.3, we have $P = \bigoplus_{k \in K} r_k \cdot Q_k$.

2. $[Q'] = \bigoplus_{k \in K} r_k \cdot Q'_k$. Again Lemma 9.4 yields $Q' = \bigoplus_{k \in K} r_k \cdot Q'_k$.

3. $t_k \prec_S [Q'_k]$ implies $t_k \sqsubseteq Q_{\text{May}}$ by (11.1). Therefore, $\bigoplus_{k \in K} r_k \cdot Q'_k \sqsubseteq Q_{\text{May}}$. Combining (1), (2) and (3) we obtain $P \sqsubseteq Q_{\text{May}}$, hence $P \sqsubseteq Q_{\text{May}}$.

**Proposition 11.3.** Let $P$ and $Q$ be in $\text{nCSP}$. Then $P \sqsubseteq Q$ implies $P \sqsubseteq Q_{\text{Must}}$.

**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 $Q \prec_{\text{FS}} [P']$, where $Q$ has the form $\bigoplus_{i \in I} a_i.Q_i$. Let $X$ be any set of actions such that $X \not\sqsubseteq \bigoplus_{i \in I} a_i.Q_i$. Therefore, there exists a $P'$ such that $[P] \models [P']$. By the Derivative lemma,

$$P \sqsubseteq Q_{\text{Must}} P'$$

(11.2)

Since $\bigoplus_{i \in I} a_i.Q_i \models [P']$, there exist $P_i, P'_i, P''_i$ such that $[P] \models [P_i] \models [P_i'] \models [P''_i]$. Now

$$P \sqsubseteq a_i.P_i \sqsubseteq a_i.Q_i \quad \text{by Must2' \cite{43}}$$

and $[Q_i] \prec_{\text{FS}} [P''_i]$. By induction, we have $P_i \sqsubseteq Q_{\text{Must}} Q_i$.

The desired result is now obtained as follows:

$$P \sqsubseteq Q_{\text{Must}} P' \cap \bigoplus_{i \in I} P_i \quad \text{by (11), (11.2) and (11.3)}$$

$$P \sqsubseteq \bigoplus_{i \in I} a_i.P_i \quad \text{by (Must2')}$$

$$P \sqsubseteq \bigoplus_{i \in I} a_i.Q_i \quad \text{by (11.4)}$$

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 $\simeq_{\text{p\text{may}}}$. 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 $\subseteq_S$ (cf. Definition 4.3). They only considered processes without $\tau$-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 \(\sqsubseteq_{p\text{may}}\) (cf. Definition 6.1) and that the latter coincides with an infinitary
version of \(\sqsubseteq_{p\text{must}}\). 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 \(\Delta^o\)
over \(S\). Since we only consider probabilistic automata with \(L = \text{Act}_\tau \cup \Omega\), we omit it and write a prob-
abilistic automaton simply as a triple \(\langle S, \Delta^o, \rightarrow \rangle\) and call \(\Delta^o\) the
initial distribution of the automaton. The operational semantics of a pCSP\(^\Omega\) process \(P\) can thus be viewed as a prob-
abilistic automaton with initial distribution \(\Delta^o := \bigcup_{CJ} P\). 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 \(\text{depth} : S \cup D(S) \to \mathbb{N}\)
such that \(s \in [\Delta]\) implies \(\text{depth}(s) < \text{depth}(\Delta)\) and \(s \xrightarrow{\alpha} \Delta\) implies \(\text{depth}(s) > \text{depth}(\Delta)\).
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\(^\Omega\) expressions.

Definition A.1. [10] A resolution of a distribution \(\Delta^o \in D(\text{sCSP}\Omega)\) is a fully probabilistic
automaton \(\langle R, \Theta^o, \rightarrow \rangle\) such that there is a resolving function \(f : R \to \text{sCSP}\Omega\) which satisfies:
(i) \(f(\Theta^o) = \Delta^o\)
(ii) if \(r \xrightarrow{\alpha} \Theta\) then \(f(r) \xrightarrow{\alpha} f(\Theta)\)
(iii) if \(r \not\rightarrow\) then \(f(r) \not\rightarrow\)
where \(f(\Theta)\) is the distribution defined by \(f(\Theta)(s) := \sum_{r \in [\Delta]} \Theta(r)\).

Note that resolutions of distributions \(\Delta^o \in D(\text{sCSP}\Omega)\) 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 \(R = \langle R, \Delta^o, \rightarrow \rangle\), the probability
that \(R\) follows the sequence of actions \(\sigma \in \Sigma^*\) from its initial distribution is given by
\(\Pr_R(\sigma, \Delta^o)\), where \(\Pr_R : \Sigma^* \times R \to [0, 1]\) is defined inductively by
\[
\Pr_R(\varepsilon, r) := 1 \quad \text{and} \quad \Pr_R(\alpha \sigma, r) := \begin{cases} \Pr_R(\sigma, \Delta) & \text{if } r \xrightarrow{\alpha} \Delta \\ 0 & \text{otherwise} \end{cases}
\]
and \(\Pr_R(\sigma, \Delta) := \text{Exp}_\Delta(\Pr_R(\sigma, \_)) = \sum_{r \in [\Delta]} \Delta(r) \cdot \Pr_R(\sigma, r)\). Here \(\varepsilon\) denotes the empty
sequence of actions and \(\sigma\) the sequence starting with \(\alpha \in \Sigma\) and continuing with \(\sigma \in \Sigma^*\).
The value \(\Pr_R(\sigma, r)\) is the probability that \(R\) proceeds with sequence \(\sigma\) from state \(r\).
Now let $\Sigma^\alpha$ be the set of finite sequences in $\Sigma^*$ that contain $\alpha$ exactly once, and that at the end. Then the probability that the fully probabilistic automaton $R$ ever performs an action $\alpha$ is given by $\sum_{\sigma \in \Sigma^\alpha} \Pr_R(\sigma, \Delta^\alpha)$.

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 $\mathbb{W}(R) \in [0,1]^\Omega$ be such that $\mathbb{W}(R)(\omega)$ is the probability that $R$ ever performs the action $\omega$.

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

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

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 $\hat{\nu}^\Omega_1(\Delta)$ extends smoothly to states and distributions in probabilistic automata. When applied to fully probabilistic automata, $\hat{\nu}^\Omega_1$ always yields singleton sets, which we will loosely identify with their unique members; thus when we write $\hat{\nu}^\Omega_1(\Delta)(\omega)$ with $\Delta$ a distribution in a fully probabilistic automaton, we actually mean the $\omega$-component of the unique element of $\hat{\nu}^\Omega_1(\Delta)$.

**Lemma A.4.** If $R = \langle R, \Delta^\alpha, \rightarrow \rangle$ is a finite fully probabilistic automaton, then

1. $\hat{\nu}^\Omega_1(\Delta) = \hat{\nu}^\Omega_1(\Delta^\omega)$ for all $\Delta \in \mathcal{D}(R)$, and
2. $\mathbb{W}(R) = \hat{\nu}^\Omega_1(\Delta^\omega)$.

**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 $\mathbb{W}(R)(\omega) = \hat{\nu}^\Omega_1(\Delta^\omega)(\omega)$, i.e. that $\sum_{\sigma \in \Sigma^\omega} \Pr_R(\sigma, \Delta^\omega) = \hat{\nu}^\Omega_1(\Delta^\omega)(\omega)$. Let $\omega \in \Omega$. We show

$$\sum_{\sigma \in \Sigma^\omega} \Pr_R(\sigma, \Delta) = \hat{\nu}^\Omega_1(\Delta)(\omega) \quad \text{and} \quad \sum_{\sigma \in \Sigma^\omega} \Pr_R(\sigma, r) = \hat{\nu}^\Omega_1(r)(\omega) \quad \text{(A.1)}$$

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

- In the base case $r$ has no enabled actions. Then $\forall i : \sum_{\sigma \in \Sigma^\omega} \Pr_R(\sigma, r) = 0$ and $\hat{\nu}^\Omega_1(r) = 0$, so $\hat{\nu}^\Omega_1(r)(\omega) = 0$.
- Now suppose there is a transition $r \xrightarrow{\alpha} \Delta$ for some action $\alpha$ and distribution $\Delta$. There are two possibilities:
  - $\alpha = \omega$. We then have $\hat{\nu}^\Omega_1(s)(\omega) = 1$. Now for any finite non-empty sequence $\sigma$ without any occurrence of $\omega$ we have $\Pr_R(\sigma, r) = 0$. Thus $\sum_{\sigma \in \Sigma^\omega} \Pr_R(\sigma, r) = \Pr_R(\omega, r) = 1$ as required.
  - $\alpha \neq \omega$. Since $\hat{\nu}^\Omega_1(r) = \alpha!\hat{\nu}^\Omega_1(\Delta)$, we have $\hat{\nu}^\Omega_1(\Delta)(\omega) = \hat{\nu}^\Omega_1(\Delta)(\omega)$. On the other hand, $\Pr_R(\beta, r) = 0$ for $\beta \neq \alpha$. Therefore

$$\sum_{\sigma \in \Sigma^\omega} \Pr_R(\sigma, r) = \sum_{\alpha \sigma \in \Sigma^\omega} \Pr_R(\alpha \sigma, r) = \sum_{\sigma \in \Sigma^\omega} \Pr_R(\alpha, \sigma) \Pr_R(\sigma, r) = \sum_{\sigma \in \Sigma^\omega} \Pr_R(\sigma, \Delta) = \hat{\nu}^\Omega_1(\Delta)(\omega) \quad \text{by induction}$$

- Finally, $\sum_{\sigma \in \Sigma^\omega} \Pr_R(\sigma, \Delta) = \sum_{\sigma \in \Sigma^\omega} \text{Exp}_\Delta(\Pr_R(\sigma, \Delta)) = \text{Exp}_\Delta(\sum_{\sigma \in \Sigma^\omega} \Pr_R(\sigma, \Delta)) = \text{Exp}_\Delta(\hat{\nu}^\Omega_1(\Delta)(\omega)) = \text{Exp}_\Delta(\hat{\nu}^\Omega_1(\Delta)(\omega)) = \hat{\nu}^\Omega_1(\Delta)(\omega)$. 

\qed
Now we look more closely at the interaction of $\tilde{V}_1^\Theta$ and resolutions.

**Lemma A.5.** Let $\Delta^\circ \in D(s\text{CSP}_\Omega)$.

1. If $\langle R, \Theta^\circ, \to \rangle$ is a resolution of $\Delta^\circ$, then $\tilde{V}_1^\Theta(\Theta^\circ) \in \tilde{V}_1^\Theta(\Delta^\circ)$.
2. If $o \in \tilde{V}_1^\Theta(\Delta^\circ)$ then there is a resolution $\langle R, \Theta^\circ, \to \rangle$ of $\Delta^\circ$ such that $\tilde{V}_1^\Theta(\Theta^\circ) = o$.

**Proof.**

1. Let $\langle R, \Theta^\circ, \to \rangle$ be a resolution of $\Delta^\circ$ with resolving function $f$. We observe that for any $\Theta \in D(R)$ we have

   \[
   \forall r \in [\Theta] : \tilde{V}_1^\Theta(r) \in \tilde{V}_1^\Theta(f(r)) \implies \tilde{V}_1^\Theta(\Theta) \in \tilde{V}_1^\Theta(f(\Theta))
   \]  

   (A.2)

   because

   \[
   \tilde{V}_1^\Theta(\Theta) = \sum_{r \in [\Theta]} \Theta(r) \cdot \tilde{V}_1^\Theta(r)
   \]

   \[
   \in \sum_{r \in [\Theta]} \Theta(r) \cdot \tilde{V}_1^\Theta(f(r))
   \]

   \[
   = \sum_{s \in [f(\Theta)]} f(\Theta)(s) \cdot \tilde{V}_1^\Theta(s)
   \]

   \[
   = \tilde{V}_1^\Theta(f(\Theta)).
   \]

   We now prove by induction on depth($r$) that $\forall r \in T : \tilde{V}_1^\Theta(r) \in \tilde{V}_1^\Theta(f(r))$, 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 $\tilde{V}_1^\Theta(r) = 0 \in \tilde{V}_1^\Theta(f(r))$.

   - Otherwise $r$ has a transition $r \overset{\alpha}{\Rightarrow} \Theta$ for some $\alpha$ and $\Theta$. By induction we have $\tilde{V}_1^\Theta(r') \in \tilde{V}_1^\Theta(f(r'))$ for all $r' \in [\Theta]$. Using (A.2) we get $\tilde{V}_1^\Theta(\Theta) \in \tilde{V}_1^\Theta(f(\Theta))$. Now

   \[
   \tilde{V}_1^\Theta(r) = \alpha!\tilde{V}_1^\Theta(\Theta) \in \alpha!\tilde{V}_1^\Theta(f(\Theta)) \subseteq \tilde{V}_1^\Theta(f(r))
   \]

   where the last step follows from the fact that $f(r) \overset{\alpha}{\Rightarrow} f(\Theta)$ 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 $\Delta^\circ$ is a point distribution on some state $s$.

   - In the base case we have $s \not\Rightarrow$. The probabilistic automaton $\langle \{s\}, \Xi, 0 \rangle$ is a resolution of $\Delta^\circ = \Xi$ 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 $s \overset{\alpha_i}{\Rightarrow} \Delta_i$ for some actions $\alpha_i$ and distributions $\Delta_i$. If $o \in \tilde{V}_1^\Theta(\Delta^\circ) = \tilde{V}_1^\Theta(s)$, then by the definition of $\tilde{V}_1^\Theta$ we have $o = \sum_{i \in I} p_i \cdot \alpha_i \cdot o_i$ with $o_i \in \tilde{V}_1^\Theta(\Delta_i)$ 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 $\langle R_i, \Theta^\circ_i, \to_i \rangle$ of $\Delta_i$ with resolving function $f_i$ such that $\tilde{V}_1^\Theta(\Theta^\circ_i) = o_i$. 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 $\langle R, \Theta^\circ, \to' \rangle$ as follows:

     - $R := \{r_i \mid i \in I\} \cup \bigcup_{i \in I} R_i$

     - $\Theta^\circ := \sum_{i \in I} p_i \cdot r_i$

     - $\to':= \{r_i \overset{\alpha_i}{\Rightarrow} \Theta^\circ_i \mid i \in I\} \cup \bigcup_{i \in I} \to_i$

   This automaton is a resolution of $\Delta^\circ = \Xi$ 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
\[ \hat{\nu}^J_1(\Theta^o) = \hat{\nu}^J_1(\sum_{i \in I} p_i \cdot \cdot \cdot ) = \sum_{i \in I} p_i \cdot \hat{\nu}^J_1(r_i) = \sum_{i \in I} p_i \cdot \alpha_i \hat{\nu}^J_1(\Theta^o_i) = \sum_{i \in I} p_i \cdot \alpha_i o_i = o. \]

We now consider the general case that \( \Delta^o \) is a proper distribution with \( [\Delta^o] = \{ s_j \mid j \in J \} \) for some finite index set \( J \). Using the reasoning in the above special case, we have a resolution \( \langle R_j, \Theta^o_j, \cdot \rangle_j \) of each distribution \( \iota_j \). Without loss of generality, we assume that \( R_k \) is disjoint from \( R_k \) for \( j \neq k \). Consider the probabilistic automaton \( \langle \bigcup_{j \in J} R_j, \sum_{j \in J} \Delta^o(s_j \cdot \Theta^o_j), \bigcup_{j \in J} \cdot \rangle_j \). It is a resolution of \( \Delta^o \) satisfying our requirement. If \( o \in \hat{\nu}^J_1(\Delta^o) \) then \( o = \sum_{j \in J} \Delta^o(s_j) \cdot o_j \) with \( o_j \in \hat{\nu}^J_1(s_j) \). Since \( o_j = \hat{\nu}^J_1(\Theta^o_j) \), we have \( o = \sum_{j \in J} \Delta^o(s_j) \cdot \Theta^o_j \).

We can now give the result relied on in Section \[ \[ \]

**Proposition A.6.** Let \( \Delta^o \in \mathcal{D}(\text{sCS}^\Omega) \). Then we have that \( \mathcal{V}(\Delta^o) = \hat{\nu}^J_1(\Delta^o) \).

**Proof.** Combine Lemmas [A.4] and [A.5] \[ \square \]

**References**