

Probabilistic Operational Correspondence

Anna Schmitt  

TU Darmstadt, Germany

Kirstin Peters  

Augsburg University, Germany

Abstract

Encodings are the main way to compare process calculi. By applying quality criteria to encodings we analyse their quality and rule out trivial or meaningless encodings. Thereby, operational correspondence is one of the most common and most important quality criteria. It ensures that processes and their translations have the same abstract behaviour. We analyse probabilistic versions of operational correspondence to enable such a verification for probabilistic systems.

Concretely, we present three versions of probabilistic operational correspondence: weak, middle, and strong. We show the relevance of the weaker version using an encoding from a sublanguage of probabilistic CCS into the probabilistic π -calculus. Moreover, we map this version of probabilistic operational correspondence onto a probabilistic behavioural relation that directly relates source and target terms. Then we can analyse the quality of the criterion by analysing the relation it induces between a source term and its translation. For the second version of probabilistic operational correspondence we proceed in the opposite direction. We start with a standard simulation relation for probabilistic systems and map it onto a probabilistic operational correspondence criterion.

2012 ACM Subject Classification Theory of computation \rightarrow Process calculi

Keywords and phrases Probabilistic Process Calculi, Encodings, Operational Correspondence

Digital Object Identifier 10.4230/LIPIcs.CONCUR.2023.15

Related Version *Full Version*: <https://doi.org/10.48550/arXiv.2307.05218>

Acknowledgements We thank the anonymous reviewers of a preliminary version of this paper for their comments and hints.

1 Introduction

Encodings are used to compare process calculi and to reason about their expressive power. Encodability criteria are conditions that limit the existence of encodings. Their main purpose is to rule out trivial or meaningless encodings, but they can also be used to limit attention to encodings that are of special interest in a particular domain or for a particular purpose. These quality criteria are the main tool in separation results, saying that one calculus is not expressible in another one; here one has to show that no encoding meeting these criteria exists. To obtain stronger separation results, care has to be taken in selecting quality criteria that are not too restrictive. For encodability results, saying that one calculus is expressible in another one, all one needs is an encoding, together with criteria testifying for the quality of the encoding. Here it is important that the criteria are not too weak.

In the literature various different criteria and different variants of the same criteria are employed to achieve separation and encodability results (see e.g. [29] for an overview). Unfortunately it is not always obvious whether the criteria used to obtain a result in a particular setting do indeed fit to this setting. A way to formally analyse the quality of encodability criteria was presented in [30]. They propose to map the criteria on conditions on relations between source and target terms that in particular relate each source term with its literal translation. This allows us to formally reason about encodability criteria,



© Anna Schmitt and Kirstin Peters;

licensed under Creative Commons License CC-BY 4.0

34th International Conference on Concurrency Theory (CONCUR 2023).

Editors: Guillermo A. Pérez and Jean-François Raskin; Article No. 15; pp. 15:1–15:17

Leibniz International Proceedings in Informatics



LIPIC Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

to completely capture and describe their semantic effect, and to analyse side conditions of combinations of criteria. We want to use this technique to define and analyse versions of operational correspondence for probabilistic systems.

Intuitively, *operational correspondence* requires executions to be respected. It consists of a completeness and a soundness part. The completeness condition requires that for all source term executions there is one emulation in the target language such that the encoding of the result in the source and the result in the target are related by some relation \mathcal{R}_T on the target language. Intuitively, the completeness condition requires that any source term execution is emulated by the target term modulo \mathcal{R}_T . Soundness requires that for all executions of the target there exists some execution of the source such that again the results are related by \mathcal{R}_T . Intuitively, soundness requires that whatever the encoded term can do is a translation of some behaviour of the source term modulo \mathcal{R}_T .

To study and compare probabilistic languages we need probabilistic versions of operational correspondence. In § 4 we start in the traditional way and use a particular encoding, that we consider reasonable. We derive a version of probabilistic operational correspondence (PrOC) that captures the way in that the encoding relates the behaviour of the source and the target language. Concretely, we consider an encoding from a sublanguage of probabilistic CCS from [7] into the probabilistic π -calculus of [38]. As result we obtain a weak version of PrOC.

In § 5 we then map the conditions in weak PrOC onto requirements of a relation between source and target terms as proposed in [30]. Thereby we are able to show that weak PrOC ensures that source terms and their literal translations are related by a probabilistic version of coupled similarity ([27]).

For our second variant of PrOC we are interested in a stricter criterion that induces a stricter simulation relation such as bisimulation. In § 6 we therefore reverse the ideas of [30]. Instead of starting with a criterion that we map on conditions of a relation, we start with an interesting relation and derive which variant of PrOC induces this relation. Since bisimulation is often considered as *the* standard behavioural relation between processes, we start with probabilistic barbed bisimulation. From this relation we derive a version of PrOC that induces this relation between source terms and their translations. Finally, we also present a strong version of PrOC and its correspondence to strong probabilistic bisimulation.

We conclude in § 7. The proofs and some additional material can be found in [33].

2 Process Calculi

A process calculus is a language $\mathcal{L} = (\mathcal{P}, \mapsto)$ consisting of a set of terms \mathcal{P} – its *syntax* – and its *semantics* defining reduction steps. The syntax of a process calculus is usually defined by a context-free grammar defining operators. An operator of arity 0 is a constant. The arguments that are again process terms are called *subterms*. A *guard* is an operator that prevents the reductions of subterms until the guard is reduced first. In the languages we consider, guards are action prefixes.

We assume that the semantics is given as an *operational semantics* consisting of inference rules defined on the operators of the language [31]. For many process calculi, the semantics is provided in two forms, as *reduction semantics* and as *labelled transition semantics*. We assume that at least the reduction semantics \mapsto is given as part of the definition, because its treatment is easier in the context of encodings. We consider probabilistic calculi, where a process reduces in a step to a discrete probability distribution.

A (*discrete*) *probability distribution* over a set S is a mapping $\Delta : S \rightarrow [0, 1]$ with $\sum_{P \in S} \Delta(P) = 1$. Let $\mathcal{D}(S)$ ranged over by Δ, Θ, Φ denote the collection of all such distributions over S . The *support* of Δ is the set $[\Delta] = \{P \mid \Delta(P) > 0\}$ of elements with

a positive probability. We use \bar{P} to denote the point distribution assigning probability 1 to state P and 0 to all other states in S . If $\sum_{i \in I} p_i = 1$, $p_i \geq 0$, and Δ_i is a probability distribution for each i in some finite index set I , then $\sum_{i \in I} p_i \cdot \Delta_i$ is a probability distribution given by $(\sum_{i \in I} p_i \cdot \Delta_i)(P) = \sum_{i \in I} p_i \cdot \Delta_i(P)$. We sometimes use $\{p_1 P_1, \dots, p_n P_n\}$ to denote a distribution Δ with $[\Delta] = \{P_1, \dots, P_n\}$ and $\Delta(P_i) = p_i$ for all $1 \leq i \leq n$.

A (reduction) step $P \mapsto \Delta$ is a single application of the reduction relation $\mapsto \subseteq \mathcal{P} \times \mathcal{D}(\mathcal{P})$, where Δ is called *derivative*. We lift \mapsto to a relation between distributions (see e.g. [7] for a similar but stricter relation).

► **Definition 1** (Reductions of Distributions). *Let $\Delta \mapsto \Theta$ whenever*

- (a) $\Delta = \sum_{i \in I} p_i \bar{P}_i$, where I is a finite index set and $\sum_{i \in I} p_i = 1$,
- (b) for each $i \in I$ there is a distribution Θ_i such that $P_i \mapsto \Theta_i$ or $\Theta_i = \bar{P}_i$,
- (c) for some $i \in I$ we have $P_i \mapsto \Theta_i$, and
- (d) $\Theta = \sum_{i \in I} p_i \cdot \Theta_i$.

Let $P \mapsto$ and $\Delta \mapsto$ denote the existence of a single step from P or Δ . We write $P \mapsto^\omega$ if there is some Δ such that $P \mapsto \Delta$ and $\Delta \mapsto^\omega$, where $\Delta \mapsto^\omega$ if Δ has an *infinite sequence* of steps. Let \Longrightarrow be the reflexive and transitive closure of \mapsto on distributions and let $P \Longrightarrow \Delta$ if $\Delta = \bar{P}$ or $P \mapsto \Longrightarrow \Delta$.

To reason about environments of terms, we use functions on process terms called contexts. More precisely, a *context* $\mathcal{C}([\cdot]_1, \dots, [\cdot]_n) : \mathcal{P}^n \rightarrow \mathcal{P}$ with n holes is a function from n terms into one term, i.e., given $P_1, \dots, P_n \in \mathcal{P}$, the term $\mathcal{C}(P_1, \dots, P_n)$ is the result of inserting P_1, \dots, P_n in the corresponding order into the n holes of \mathcal{C} .

Assume a countably-infinite set \mathcal{N} of *names*. Let $\bar{\mathcal{N}} = \{\bar{n} \mid n \in \mathcal{N}\}$ and $\tau \notin \mathcal{N}$. Let $\text{fn}(P)$ denote the set of free names in P . A substitution σ is a finite mapping from names to names defined by a set $\{y_1/x_1, \dots, y_n/x_n\} = \{y_1, \dots, y_n/x_1, \dots, x_n\} = \{\tilde{y}/\tilde{x}\}$ of renamings, where the x_1, \dots, x_n are pairwise distinct. The application $P\{\tilde{y}/\tilde{x}\}$ of a substitution on a term is defined as the result of simultaneously replacing all free occurrences of x_i by y_i for $i \in \{1, \dots, n\}$, possibly applying α -conversion to avoid capture or name clashes. For all names in $\mathcal{N} \setminus \{x_1, \dots, x_n\}$ the substitution behaves as the identity mapping. We naturally extend substitution to distributions.

For the last criterion of [15] (see Section 3), we need a special constant \surd , called *success(ful termination)*. Therefore, we add \surd to the grammar of a language without explicitly mentioning it. Success is used as a barb to implement some form of (fair) testing, where $P \downarrow \surd$ if P has an unguarded occurrence of \surd , $\Delta \downarrow \surd$ if there is some P with $\Delta(P) > 0$ and $P \downarrow \surd$, and $P \Downarrow \surd = \exists \Delta. P \Longrightarrow \Delta \wedge \Delta \downarrow \surd$.

Languages can be augmented with (a set of) relations $\mathcal{R} \subseteq \mathcal{P}^2$ on their processes. If $\mathcal{R} \subseteq B^2$ is a relation and $B' \subseteq B$, then the relation $\mathcal{R}|_{B'} = \{(x, y) \mid x, y \in B' \wedge (x, y) \in \mathcal{R}\}$ denotes the restriction of \mathcal{R} to the domain B' .

Following [7], we lift relations $\mathcal{R} \subseteq \mathcal{P}^2$ to a relation $\bar{\mathcal{R}} \subseteq \mathcal{D}(\mathcal{P})^2$ on distributions.

► **Definition 2** (Relations on Distributions, [7]).

Let $\mathcal{R} \subseteq \mathcal{P}^2$ and let $\Delta, \Theta \in \mathcal{D}(\mathcal{P})$. Then $(\Delta, \Theta) \in \bar{\mathcal{R}}$ if

- (a) $\Delta = \sum_{i \in I} p_i \bar{P}_i$, where I is a finite index set and $\sum_{i \in I} p_i = 1$,
- (b) for each $i \in I$ there is a process Q_i such that $(P_i, Q_i) \in \mathcal{R}$, and
- (c) $\Theta = \sum_{i \in I} p_i \bar{Q}_i$.

An important property of this lifting operation is that it preserves reflexivity and transitivity, i.e., if \mathcal{R} is reflexive/transitive then so is $\bar{\mathcal{R}}$. The case of transitivity is shown in [7]. The case of reflexivity can be found in the technical report ([33]).

3 Encodings and Quality Criteria

Let $\mathcal{L}_S = \langle \mathcal{P}_S, \mapsto_S \rangle$ and $\mathcal{L}_T = \langle \mathcal{P}_T, \mapsto_T \rangle$ be two process calculi, denoted as *source* and *target* language. An *encoding* from \mathcal{L}_S into \mathcal{L}_T is a function $\llbracket \cdot \rrbracket : \mathcal{P}_S \rightarrow \mathcal{P}_T$. We often use S, S', \dots and T, T', \dots to range over \mathcal{P}_S and \mathcal{P}_T , respectively.

We naturally extend the encoding function to distributions, i.e., for all distributions Δ in the source language and all T in the target language: $\llbracket \Delta \rrbracket(T) = \sum_{S \in \{S \in \mathcal{P}_S \mid \llbracket S \rrbracket = T\}} \Delta(S)$.

Let $\varphi_{[\cdot]} : \mathcal{N} \rightarrow \mathcal{N}^k$ be a *renaming policy*, i.e., a mapping from a name to a vector of names that can be used by encodings to split names and to reserve special names, such that no two different names are translated into overlapping vectors of names. We use projection to obtain the respective elements of a translated name, i.e., if $\varphi_{[\cdot]}(a) = (a_1, a_2, a_3)$ then $\varphi_{[\cdot]}(a).2 = a_2$. Slightly abusing notation, we sometimes use the tuples that are generated by the renaming policy as sets. We require e.g. $\varphi_{[\cdot]}(a) \cap \varphi_{[\cdot]}(b) = \emptyset$ whenever $a \neq b$.

To analyse the quality of encodings and to rule out trivial or meaningless encodings, they are augmented with a set of quality criteria. One such set of criteria that is well suited for separation as well as encodability results between traditional processes calculi, i.e., calculi without probabilities, was proposed in [15]. It turns out that for probabilistic systems as defined above the only criterion that needs to be adapted is operational correspondence. Accordingly, we inherit the remaining criteria from [15]:

Compositionality: For every operator **op** with arity n of \mathcal{L}_S and for every subset of names N , there exists a context $\mathcal{C}_{\mathbf{op}}^N([\cdot]_1, \dots, [\cdot]_n)$ such that, for all S_1, \dots, S_n with $\text{fn}(S_1) \cup \dots \cup \text{fn}(S_n) = N$, it holds that $\llbracket \mathbf{op}(S_1, \dots, S_n) \rrbracket = \mathcal{C}_{\mathbf{op}}^N(\llbracket S_1 \rrbracket, \dots, \llbracket S_n \rrbracket)$.

Name Invariance w.r.t. a Relation $\mathcal{R}_T \subseteq \mathcal{P}_T^2$: For every $S \in \mathcal{P}_S$ and every substitution σ , it holds that $\llbracket S\sigma \rrbracket \equiv_\alpha \llbracket S \rrbracket \sigma'$ if σ is injective and $(\llbracket S\sigma \rrbracket, \llbracket S \rrbracket \sigma') \in \mathcal{R}_T$ otherwise, where σ' is such that $\varphi_{[\cdot]}(\sigma(a)) = \sigma'(\varphi_{[\cdot]}(a))$ for all $a \in \mathcal{N}$.

Divergence Reflection: For every S , $\llbracket S \rrbracket \mapsto^\omega$ implies $S \mapsto^\omega$.

Success Sensitiveness: For every S , $S \Downarrow_\checkmark$ iff $\llbracket S \rrbracket \Downarrow_\checkmark$.

Compositionality ensures that encodings are of practical relevance by enforcing that they can be implemented compositionally, i.e., by an algorithm that proceeds on the syntax and does not need to analyse the source term to compute its translation. However, the formulation of compositionality is rather strict, i.e., it rules out practically relevant translations. Note that the best known encoding from the asynchronous π -calculus into the Join Calculus in [11] is not compositional, but consists of an inner, compositional encoding surrounded by a fixed context – the implementation of so-called firewalls – that is parameterised on the free names of the source term. In order to capture this and similar encodings we relax the definition of compositionality.

Weak Compositionality: The encoding is either compositional or consists of an inner, compositional encoding surrounded by a fixed context that can be parameterised on the free names of the source term or information that are not part of the source term.

Precisely, we use this relaxation to capture process definitions that are a relevant part of the source language CCS_p but that are not contained in source terms.

A behavioural relation \mathcal{R}_T on the target is assumed for name invariance and operational correspondence. \mathcal{R}_T needs to be success sensitive, i.e., $(T_1, T_2) \in \mathcal{R}_T$ implies $T_1 \Downarrow_\checkmark$ iff $T_2 \Downarrow_\checkmark$.

Operational correspondence is arguably the most important of the five criteria in [15], since it compares the behaviour of source terms and their translations (though only the combination with success sensitiveness ensures that this requirement is not trivial). It consists of a soundness and a completeness condition. *Completeness* requires that every computation of a source term can be emulated by its translation. *Soundness* requires that

every computation of a target term corresponds to some computation of the corresponding source term. Different variants of operational correspondence are used in the literature (see e.g. [30, 29]). In particular, the following three variants are often used for encodings between process calculi without probabilities.

► **Definition 3** (Operational Correspondence, Non-Probabilistic).

An encoding $\llbracket \cdot \rrbracket$ is strongly operationally corresponding w.r.t. $\mathcal{R}_\top \subseteq \mathcal{P}_\top^2$ if it is:

Strongly Complete: $\forall S, S'. S \mapsto S'$ implies $(\exists T. \llbracket S \rrbracket \mapsto T \wedge (\llbracket S' \rrbracket, T) \in \mathcal{R}_\top)$

Strongly Sound: $\forall S, T. \llbracket S \rrbracket \mapsto T$ implies $(\exists S'. S \mapsto S' \wedge (\llbracket S' \rrbracket, T) \in \mathcal{R}_\top)$

$\llbracket \cdot \rrbracket$ is operationally corresponding w.r.t. $\mathcal{R}_\top \subseteq \mathcal{P}_\top^2$ if it is:

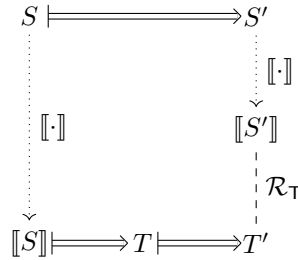
Complete: $\forall S, S'. S \Longrightarrow S'$ implies $(\exists T. \llbracket S \rrbracket \Longrightarrow T \wedge (\llbracket S' \rrbracket, T) \in \mathcal{R}_\top)$

Sound: $\forall S, T. \llbracket S \rrbracket \Longrightarrow T$ implies $(\exists S'. S \Longrightarrow S' \wedge (\llbracket S' \rrbracket, T) \in \mathcal{R}_\top)$

$\llbracket \cdot \rrbracket$ is weakly operationally corresponding w.r.t. $\mathcal{R}_\top \subseteq \mathcal{P}_\top^2$ if it is:

Complete: $\forall S, S'. S \Longrightarrow S'$ implies $(\exists T. \llbracket S \rrbracket \Longrightarrow T \wedge (\llbracket S' \rrbracket, T) \in \mathcal{R}_\top)$

Weakly Sound: $\forall S, T. \llbracket S \rrbracket \Longrightarrow T$ impl. $(\exists S', T'. S \Longrightarrow S' \wedge T \Longrightarrow T' \wedge (\llbracket S' \rrbracket, T') \in \mathcal{R}_\top)$



The first variant requires a strong correspondence between source and target term steps, i.e., a source term step is emulated by exactly one target term step and vice versa. The second variant allows for the emulation of a single source term step by a sequence of target term steps. With this variant encoding functions may use things like pre- and post-processing steps. The last variant additionally allows for *intermediate states*, i.e., target terms that are not directly related to the encoding of any source term but that are in between two such source term encodings [27, 28, 17]. Such an intermediate state is depicted by T on the right. Intermediate states often result from partial commitments: The decision on which step is performed for a single source term step is split into two or more partial decisions in a sequence of target term steps, where each decision already rules out some of the alternatives that existed in the source for this step but does not rule out all alternatives.

4 PrOC for a Reasonable Encoding

The quality criteria of encodings should rule out trivial and meaningless encodings but they should capture good encodings. Accordingly, we start with an intuitively reasonable encoding (from CCS_p into π_p) and derive a version of PrOC that captures how this encoding translates the behaviour of source terms. Then we analyse the quality of our new criterium by mapping it on a relation between source and target terms in Section 5.

Our source language, probabilistic CCS, is introduced in [7] as a probabilistic extension of CCS [21]. We omit the operator for non-deterministic choice from [7], because its summands are not necessarily guarded, whereas our target language has only guarded choice. Thus, ignoring this operator simplifies the task of finding an encoding. Since for the design of encodability criteria the consideration of non-determinism and unguarded choices is

15:6 Probabilistic Operational Correspondence

orthogonal to the consideration of probabilities, it is safe to neglect this operator here. The versions of PrOC we discover in the following can deal with combinations of non-determinism and probabilities. We denote the resulting calculus as CCS_p . Let u, v, \dots range over the set of *actions* $\text{Act} = \mathcal{N} \cup \bar{\mathcal{N}} \cup \{\tau\}$.

► **Definition 4** (Syntax of CCS_p). *The terms \mathcal{P}_C of CCS_p are given by:*

$$P ::= u. \bigoplus_{i \in I} p_i P_i \quad | \quad P_1 | P_2 \quad | \quad P \setminus A \quad | \quad P[f] \quad | \quad C\langle \tilde{x} \rangle$$

where $A \subseteq \mathcal{N}$ and $f : \mathcal{N} \rightarrow \mathcal{N}$ is a renaming function.

The probabilistic choice operator $u. \bigoplus_{i \in I} p_i P_i$ is guarded by an action u and offers branches with probabilities, where $p_i > 0$ is the probability of branch i with $\sum_{i \in I} p_i = 1$. We write $\bigoplus_{i \in 1..n} p_i P_i$ as $p_1 P_1 \oplus \dots \oplus p_n P_n$ for a finite index set I . The process $P_1 | P_2$ implements parallel composition, in $P \setminus A$ the names in A are restricted, and $P[f]$ behaves like P where each $a \in \mathcal{N}$ is replaced by $f(a)$. For simplicity, we assume that for all f the set $\{x \mid f(x) \neq x\}$ is finite. Each process constant C has a definition $C \stackrel{\text{def}}{=} (\tilde{x})P$, where $P \in \mathcal{P}_C$ and \tilde{x} collect all names in P that are not restricted. Then $C\langle \tilde{y} \rangle$ behaves as P with \tilde{y} replacing \tilde{x} .

As described in [33], we use a reduction semantics obtained from the labelled semantics in [7] by a rule that maps every τ -labelled step to a reduction step. Due to lack of space, we only highlight the rules for choice and communication here and refer to [33] for the rest.

$$\text{PROBCHOICE}_{\text{CCS}_p} \frac{\Delta(P) = \sum \{p_i \mid i \in I \wedge P_i = P\}}{u. \bigoplus_{i \in I} p_i P_i \xrightarrow{u} \Delta} \quad \text{COM}_{\text{CCS}_p} \frac{P_1 \xrightarrow{a} \Delta_1 \quad P_2 \xrightarrow{\bar{a}} \Delta_2}{P_1 | P_2 \xrightarrow{\tau} \Delta_1 | \Delta_2}$$

$$\text{where } (\Delta_1 | \Delta_2)(P) = \begin{cases} \Delta_1(P_1) \cdot \Delta_2(P_2) & \text{, if } P = P_1 | P_2 \\ 0 & \text{otherwise} \end{cases}$$

Our target language, the probabilistic π -calculus (π_p), is introduced in [38], as a probabilistic version of the π I-calculus [32], where output is endowed with probabilities.

► **Definition 5** (Syntax of π_p). *The terms \mathcal{P}_π of π_p are given by:*

$$P ::= \bar{x} \oplus_{i \in I} p_i \mathbf{in}_i(\tilde{y}_i).P_i \quad | \quad x \Phi_{i \in I} \mathbf{in}_i(\tilde{y}_i).P_i \quad | \quad P | P \quad | \quad (\nu x)P \quad | \quad \mathbf{0} \quad | \quad !x(\tilde{y}).P$$

The probabilistic π -calculus assigns probabilities to output. The process $\bar{x} \oplus_{i \in I} p_i \mathbf{in}_i(\tilde{y}_i).P_i$ is a probabilistic selecting output, where $p_i \in [0, 1]$ for all $i \in I$ and $\sum_{i \in I} p_i = 1$. The term $x \Phi_{i \in I} \mathbf{in}_i(\tilde{y}_i).P_i$ is a branching input, which does not attach probabilities to the single events. For branching/selection labels, the index i is the branch of the label. We write $\bar{x}(\tilde{y}).P$ and $x(\tilde{y}).P$ for single outputs or inputs and $\bar{x}(p_1 \mathbf{in}_1(\tilde{y}_1).P_1 \oplus \dots \oplus p_n \mathbf{in}_n(\tilde{y}_n).P_n)$ and $x(\mathbf{in}_1(\tilde{y}_1).P_1 \& \dots \& \mathbf{in}_n(\tilde{y}_n).P_n)$ for finite indexing sets $I = \{1, \dots, n\}$ in $\bigoplus_{i \in I}$ and $\Phi_{i \in I}$. The process $P | Q$ is a parallel composition, $(\nu x)P$ is a restriction, and $!x(\tilde{y}).P$ is a replicated input. We sometimes omit empty sequences of arguments $()$ as well as dangling $\mathbf{0}$, i.e., \bar{a} stands for $\bar{a}().\mathbf{0}$.

Structural congruence \equiv is defined, similarly to [22], as the smallest congruence containing α -equivalence \equiv_α that is closed under the following rules:

$$\begin{aligned} P | \mathbf{0} &\equiv P & P | Q &\equiv Q | P & P | (Q | R) &\equiv (P | Q) | R & (\nu x)\mathbf{0} &\equiv \mathbf{0} \\ (\nu xy)P &\equiv (\nu yx)P & (\nu x)(P | Q) &\equiv P | (\nu x)Q & \text{if } x \notin \text{fn}(P) & & & \end{aligned}$$

We lift structural congruence to distributions, i.e., $\Delta_1 \equiv \Delta_2$ if there is a finite index set I such that $\Delta_1 = \sum_{i \in I} p_i \bar{P}_i$, $\Delta_2 = \sum_{i \in I} p_i \bar{Q}_i$, and $P_i \equiv Q_i$ for all $i \in I$.

$$\begin{aligned}
\llbracket x. \bigoplus_{i \in I} p_i P_i \rrbracket_{\text{CCS}_p}^{\pi_p} &= x.(\nu z_i) \left(\bar{z}_i \bigoplus_{i \in I} p_i \text{in}_i. \llbracket P_i \rrbracket_{\text{CCS}_p}^{\pi_p} \mid z_i \right) \\
\llbracket \bar{x}. \bigoplus_{i \in I} p_i P_i \rrbracket_{\text{CCS}_p}^{\pi_p} &= \bar{x} \bigoplus_{i \in I} p_i \text{in}_i. \llbracket P_i \rrbracket_{\text{CCS}_p}^{\pi_p} \\
\llbracket \tau. \bigoplus_{i \in I} p_i P_i \rrbracket_{\text{CCS}_p}^{\pi_p} &= (\nu z_\tau) \left(\bar{z}_\tau \bigoplus_{i \in I} p_i \text{in}_i. \llbracket P_i \rrbracket_{\text{CCS}_p}^{\pi_p} \mid z_\tau \right) \\
\llbracket P \mid Q \rrbracket_{\text{CCS}_p}^{\pi_p} &= \llbracket P \rrbracket_{\text{CCS}_p}^{\pi_p} \mid \llbracket Q \rrbracket_{\text{CCS}_p}^{\pi_p} && \text{where for each } f \text{ the} \\
\llbracket P \setminus A \rrbracket_{\text{CCS}_p}^{\pi_p} &= (\nu A) \llbracket P \rrbracket_{\text{CCS}_p}^{\pi_p} \\
\llbracket P[f] \rrbracket_{\text{CCS}_p}^{\pi_p} &= \llbracket P \rrbracket_{\text{CCS}_p}^{\pi_p} \{ \text{ran}_f / \text{dom}_f \} \\
\llbracket C \langle \tilde{y} \rangle \rrbracket_{\text{CCS}_p}^{\pi_p} &= \bar{C}(\tilde{y}) \\
\llbracket \checkmark \rrbracket_{\text{CCS}_p}^{\pi_p} &= \checkmark \\
\text{ran}_f &= y_1, \dots, y_n \text{ and } \text{dom}_f = x_1, \dots, x_n \text{ are vectors of names such that} \\
&\{x_1, \dots, x_n\} = \{x \mid f(x) \neq x\} \text{ and } f(x_i) = y_i \text{ for all } 1 \leq i \leq n.
\end{aligned}$$

■ **Figure 1** Inner Encoding.

Again we refer to [33] or [38] for the semantics of π_p and highlight the rule for selection.

$$\text{SELECT}_{\pi_p} \bar{x} \bigoplus_{i \in I} p_i \text{in}_i(\tilde{y}_i). P_i \left\{ \frac{\bar{x} \text{in}_i \langle \tilde{y}_i \rangle}{p_i} P_i \right\}_{i \in I}$$

We observe that in [38] the semantics is given as a Segala automaton. To obtain a reduction semantics with steps into probability distributions, we add the following rule:

$$\text{RED}_{\pi_p} \frac{P \left\{ \frac{\tau}{p_i} Q_i \right\}_{i \in I} \quad \Delta(R) = \sum \{p_i \mid Q_i = R\}}{P \mapsto \Delta}$$

An encoding from CCS_p into π_p has to deal with the following challenge: Probabilistic choice in CCS_p can have output, input, and τ guards, but in π_p all summands of a probabilistic choice have to be output guarded. The input guarded choice operator in π_p has no probabilities and the τ is not part of the syntax. Therefore, the encoding of a probabilistic choice in CCS_p is split into three cases depending on its guard and the probabilistic selecting output of the target language is used in each of these cases to assign the probabilities.

We use the renaming policy $\varphi_{(\cdot)}^{\pi_p}_{\text{CCS}_p}$ to reserve the names z_i (for input guarded probabilistic choice) and z_τ (for τ -guarded probabilistic choice). Moreover, we translate process constants C into channel names C and use the renaming policy to keep these channel names C distinct from source term names. Precisely, we assume that $|\varphi_{(\cdot)}^{\pi_p}_{\text{CCS}_p}(n)| = 1$ for all $n \in \mathcal{N}$ and that $\varphi_{(\cdot)}^{\pi_p}_{\text{CCS}_p}(n) \cap \{z_i, z_\tau, C \mid C \text{ is a process constant}\} = \emptyset$ for all $n \in \mathcal{N}$. In the following, in order to increase readability, the indication of the renaming policy is omitted, i.e., we assume $z_i \neq n \neq z_\tau$ and $n \neq C$ for all source term names n and all process constants C and write n instead of $\varphi_{(\cdot)}^{\pi_p}_{\text{CCS}_p}(n)$.1 in the translation.

► **Definition 6** (Encoding $(\cdot)_{\text{CCS}_p}^{\pi_p} / \llbracket \cdot \rrbracket_{\text{CCS}_p}^{\pi_p}$ from CCS_p into π_p). *The encoding of $S \in \mathcal{P}_C$ with the process definitions $C_1 \stackrel{\text{def}}{=} (\tilde{x}_1).S_1, \dots, C_n \stackrel{\text{def}}{=} (\tilde{x}_n).S_n$ consists of the outer encoding $(\cdot)_{\text{CCS}_p}^{\pi_p}$, where $(S)_{\text{CCS}_p}^{\pi_p}$ is*

$$(\nu C_1, \dots, C_n) \left(\llbracket S \rrbracket_{\text{CCS}_p}^{\pi_p} \mid !C_1(\tilde{x}_1). \llbracket S_1 \rrbracket_{\text{CCS}_p}^{\pi_p} \mid \dots \mid !C_n(\tilde{x}_n). \llbracket S_n \rrbracket_{\text{CCS}_p}^{\pi_p} \right)$$

and the inner encoding $\llbracket \cdot \rrbracket_{\text{CCS}_p}^{\pi_p}$ is given in Figure 1.

15:8 Probabilistic Operational Correspondence

The encoding of a probabilistic choice is split into three cases. For input guards a single input on x is used, to enable the communication with a potential corresponding output. Then a probabilistic selecting output on the reserved name z_i composed in parallel with a matching input is used to encode the probabilities. The sequence of these two communication steps on x and z_i emulates the behaviour of a single communication step in the source.

The encoding of an output-guarded probabilistic choice is straightforward, as it is translated using the probabilistic selecting output.

For the guard τ , an output-guarded probabilistic choice in parallel to a single input on the reserved name z_τ is used.

The application of a renaming function is encoded by a substitution. A call $C(\tilde{y})$ is encoded by an output, where the corresponding process definitions are translated into replicated inputs and placed in parallel by the outer encoding. The remaining translations are homomorphic.

We are looking for a variant of operational correspondence with probabilities that captures the way in that our encoding emulates source term steps. By Definition 6, the emulation of a communication step consists of a sequence containing two target term steps. Accordingly, we are looking for a variant of operational correspondence that permits a sequence of target term steps to emulate a single source term step as in the second and third case of Definition 3.

► **Example 7.** Consider $S = \bar{x}. (\frac{3}{4}P \oplus \frac{1}{4}Q) \mid x. (\frac{1}{2}R \oplus \frac{1}{2}S)$ in CCS_p without process definitions. By the semantics of CCS_p , $S \mapsto \Delta_S = \{\frac{3}{8}(P \mid R), \frac{3}{8}(P \mid S), \frac{1}{8}(Q \mid R), \frac{1}{8}(Q \mid S)\}$. By Definition 6, $\langle S \rangle_{\text{CCS}_p}^{\pi_p} = \llbracket S \rrbracket_{\text{CCS}_p}^{\pi_p}$ and:

$$\begin{aligned} \llbracket S \rrbracket_{\text{CCS}_p}^{\pi_p} &= \bar{x} \left(\frac{3}{4} \text{in}_1. \llbracket P \rrbracket_{\text{CCS}_p}^{\pi_p} \oplus \frac{1}{4} \text{in}_2. \llbracket Q \rrbracket_{\text{CCS}_p}^{\pi_p} \right) \mid \\ &\quad x. (\nu z_i) \left(\bar{z}_i \left(\frac{1}{2} \text{in}_1. \llbracket R \rrbracket_{\text{CCS}_p}^{\pi_p} \oplus \frac{1}{2} \text{in}_2. \llbracket S \rrbracket_{\text{CCS}_p}^{\pi_p} \right) \mid z_i \right) \end{aligned}$$

By the semantics of π_p , $\langle S \rangle_{\text{CCS}_p}^{\pi_p}$ can perform exactly one maximal sequence of steps, namely $\langle S \rangle_{\text{CCS}_p}^{\pi_p} \mapsto \Delta_T \mapsto \Delta'_T$, where:

$$\begin{aligned} \Delta_T &= \left\{ \frac{3}{4} \left(\llbracket P \rrbracket_{\text{CCS}_p}^{\pi_p} \mid (\nu z_i) \left(\bar{z}_i \left(\frac{1}{2} \text{in}_1. \llbracket R \rrbracket_{\text{CCS}_p}^{\pi_p} \oplus \frac{1}{2} \text{in}_2. \llbracket S \rrbracket_{\text{CCS}_p}^{\pi_p} \right) \mid z_i \right) \right), \right. \\ &\quad \left. \frac{1}{4} \left(\llbracket Q \rrbracket_{\text{CCS}_p}^{\pi_p} \mid (\nu z_i) \left(\bar{z}_i \left(\frac{1}{2} \text{in}_1. \llbracket R \rrbracket_{\text{CCS}_p}^{\pi_p} \oplus \frac{1}{2} \text{in}_2. \llbracket S \rrbracket_{\text{CCS}_p}^{\pi_p} \right) \mid z_i \right) \right) \right\} \\ \Delta'_T &= \left\{ \frac{3}{8} \left(\llbracket P \rrbracket_{\text{CCS}_p}^{\pi_p} \mid \llbracket R \rrbracket_{\text{CCS}_p}^{\pi_p} \mid \mathbf{0} \right), \frac{3}{8} \left(\llbracket P \rrbracket_{\text{CCS}_p}^{\pi_p} \mid \llbracket S \rrbracket_{\text{CCS}_p}^{\pi_p} \mid \mathbf{0} \right), \right. \\ &\quad \left. \frac{1}{8} \left(\llbracket Q \rrbracket_{\text{CCS}_p}^{\pi_p} \mid \llbracket R \rrbracket_{\text{CCS}_p}^{\pi_p} \mid \mathbf{0} \right), \frac{1}{8} \left(\llbracket Q \rrbracket_{\text{CCS}_p}^{\pi_p} \mid \llbracket S \rrbracket_{\text{CCS}_p}^{\pi_p} \mid \mathbf{0} \right) \right\} \end{aligned}$$

We observe that $\langle \Delta_S \rangle_{\text{CCS}_p}^{\pi_p} \equiv \Delta'_T$ and in particular that the distributions Δ_S and Δ'_T have the same probabilities. However, the distribution Δ_T is not that obviously related to $\langle S \rangle_{\text{CCS}_p}^{\pi_p}$ or $\langle \Delta_S \rangle_{\text{CCS}_p}^{\pi_p}$. ◀

We already ruled out strong operational correspondence as defined in Definition 3. The other two versions differ in whether they allow for intermediate states. Another look at Example 7 tells us that intermediate states make sense. Δ_T is a finite probability distribution with two cases: the case containing $\llbracket P \rrbracket_{\text{CCS}_p}^{\pi_p}$ with probability $\frac{3}{4}$ and the case containing $\llbracket Q \rrbracket_{\text{CCS}_p}^{\pi_p}$ with probability $\frac{1}{4}$, but neither S nor Δ_S have cases with these probabilities. In the second variant of operational correspondence in Definition 3 without intermediate states, we would need to find a relation \mathcal{R}_\top that relates Δ_T either to $\langle S \rangle_{\text{CCS}_p}^{\pi_p}$ or Δ'_T . Such a relation \mathcal{R}_\top is difficult or at least not intuitive, since it has to relate states with different probabilities. It is easier to allow for intermediate states. So, we want to build a weak version of operational correspondence (third case of Definition 3) with probabilities.

As discussed above, we need to require that the resulting distribution in the source has the same probabilities as the resulting distribution in the target term sequence. Moreover, it makes sense to require that for all matching cases, i.e., all branches with the same probability, the encoding of the respective source term and the respective target term are related by \mathcal{R}_T . Definition 2 allows us to formalise this by comparing the distributions with $\overline{\mathcal{R}_T}$. This leads to the version of probabilistic operational correspondence below denoted as *weak probabilistic operational correspondence*.

► **Definition 8** (Weak Probabilistic Operational Correspondence). *An encoding $\llbracket \cdot \rrbracket : \mathcal{P}_S \rightarrow \mathcal{P}_T$ is weakly probabilistic operationally corresponding (weak PrOC) w.r.t. $\mathcal{R}_T \subseteq \mathcal{P}_T^2$ if it is:*

Probabilistic Complete:

$$\forall S, \Delta_S. S \Longrightarrow \Delta_S \text{ implies } (\exists \Delta_T. \llbracket S \rrbracket \Longrightarrow \Delta_T \wedge (\llbracket \Delta_S \rrbracket, \Delta_T) \in \overline{\mathcal{R}_T})$$

Weakly Probabilistic Sound: $\forall S, \Delta_T. \llbracket S \rrbracket \Longrightarrow \Delta_T \text{ implies}$

$$(\exists \Delta_S, \Delta'_T. S \Longrightarrow \Delta_S \wedge \Delta_T \Longrightarrow \Delta'_T \wedge (\llbracket \Delta_S \rrbracket, \Delta'_T) \in \overline{\mathcal{R}_T})$$

Every source term step is emulated by one or two target term steps modulo \equiv (the standard structural congruence on the target language). We prove in [33] that this holds for all source term steps and that the encoding satisfies weak PrOC. Weak compositionality holds by definition. Name invariance follows from the strict use of the renaming policy and because the encoding does not introduce free names. Divergence reflection results from weak probabilistic operational soundness. Finally, weak probabilistic operational correspondence and the homomorphic translation of success ensure that $(\cdot)_{\text{CCS}_p}^{\pi_p}$ is success sensitive.

► **Theorem 9.** *The encoding $(\cdot)_{\text{CCS}_p}^{\pi_p}$ satisfies weak compositionality, name invariance, weak probabilistic operational correspondence w.r.t. \equiv , divergence reflection, and success sensitivity.*

To ensure that weak PrOC is a meaningful criterion, we use the technique for analysing encodability criteria presented in [30]. Therefore, weak PrOC is mapped on requirements on a relation between source and target terms.

5 Analysing Weak Probabilistic Operational Correspondence

To analyse the quality of the criterion weak PrOC in Definition 8, we follow the technique presented in [30] and map this criterion on requirements of a behavioural relation between source and target. Thus, an encoding that satisfies this criterion relates source terms and their translations in the target modulo the respective behavioural relation. This transfers the task to analyse the quality of weak PrOC to the quality of the behavioural relation it is mapped on. As such relations are a well researched area, this allows us to formally reason about the underlying version of operational correspondence.

In [30] the relation between different versions of operational correspondence and behavioural relations is shown. Therefore, requirements are derived such that an encoding is operational corresponding w.r.t. the considered variant of operational correspondence iff a behavioural relation with these requirements exists that relates source terms with their translations. The derived relation describes how similar the behaviour of the source term is to its translation and, thus, tells us about the quality of the criterion.

Among the versions of operational correspondence considered in [30] weak PrOC is closest to weak operational correspondence in Definition 3. The corresponding result relates weak operational correspondence and correspondence similarity.

15:10 Probabilistic Operational Correspondence

► **Lemma 10** (Weak Operational Correspondence, [30]). $\llbracket \cdot \rrbracket$ is weakly operationally corresponding w.r.t. a preorder $\mathcal{R}_\top \subseteq \mathcal{P}_\top^2$ that is a correspondence simulation iff $\exists \mathcal{R}_{\llbracket \cdot \rrbracket}. (\forall S. (S, \llbracket S \rrbracket) \in \mathcal{R}_{\llbracket \cdot \rrbracket}) \wedge \mathcal{R}_\top = \mathcal{R}_{\llbracket \cdot \rrbracket} \upharpoonright_{\mathcal{P}_\top} \wedge (\forall S, T. (S, T) \in \mathcal{R}_{\llbracket \cdot \rrbracket} \rightarrow (\llbracket S \rrbracket, T) \in \mathcal{R}_\top) \wedge \mathcal{R}_{\llbracket \cdot \rrbracket}$ is a preorder and a correspondence simulation.

Remember that a preorder is a binary relation that is reflexive and transitive. The relation $\mathcal{R}_{\llbracket \cdot \rrbracket}$ (here and in all of the following results) is a set of pairs over the disjoint union of source and target terms. *Correspondence similarity* is a simulation relation in between bisimilarity and coupled similarity, that was derived in [30] to exactly capture the nature of weak operational correspondence.

► **Definition 11** (Correspondence Simulation, [30]). A relation \mathcal{R} is a (weak reduction) correspondence simulation if for each $(P, Q) \in \mathcal{R}$:

- $P \Longrightarrow P'$ implies $\exists Q'. Q \Longrightarrow Q' \wedge (P', Q') \in \mathcal{R}$
- $Q \Longrightarrow Q'$ implies $\exists P'', Q''. P \Longrightarrow P'' \wedge Q' \Longrightarrow Q'' \wedge (P'', Q'') \in \mathcal{R}$

Two terms are correspondence similar if a correspondence simulation relates them.

To obtain a result similar to Lemma 10 for weak PrOC, we use Definition 2 to lift the definition of correspondence similarity to probability distributions.

► **Definition 12** (Probabilistic Correspondence Simulation). A relation \mathcal{R} is a (weak) probabilistic (reduction) correspondence simulation if for each $(P, Q) \in \mathcal{R}$:

- $P \Longrightarrow \Delta$ implies $\exists \Theta. Q \Longrightarrow \Theta \wedge (\Delta, \Theta) \in \overline{\mathcal{R}}$
- $Q \Longrightarrow \Theta$ implies $\exists \Delta', \Theta'. P \Longrightarrow \Delta' \wedge \Theta \Longrightarrow \Theta' \wedge (\Delta', \Theta') \in \overline{\mathcal{R}}$

Two terms are probabilistic correspondence similar if a probabilistic correspondence simulation relates them.

To use probabilistic correspondence similarity, we have to show that the lifting operation in Definition 2 preserves the property of being a probabilistic correspondence simulation (at least for preorders). A relation $\overline{\mathcal{R}}$ on distributions is a probabilistic correspondence simulation if it satisfies Definition 11 with distributions instead of processes.

► **Lemma 13** (Preservation of the Correspondence Property).

If the preorder \mathcal{R} is a probabilistic correspondence simulation then so is $\overline{\mathcal{R}}$.

With the probabilistic version of correspondence similarity we can adapt Lemma 10 to weak PrOC.

► **Theorem 14** (Weak PrOC). $\llbracket \cdot \rrbracket$ is weakly probabilistically operationally corresponding w.r.t. a preorder $\mathcal{R}_\top \subseteq \mathcal{P}_\top^2$ that is a probabilistic correspondence simulation iff

$\exists \mathcal{R}_{\llbracket \cdot \rrbracket}. (\forall S. (S, \llbracket S \rrbracket) \in \mathcal{R}_{\llbracket \cdot \rrbracket}) \wedge \mathcal{R}_\top = \mathcal{R}_{\llbracket \cdot \rrbracket} \upharpoonright_{\mathcal{P}_\top} \wedge (\forall S, T. (S, T) \in \mathcal{R}_{\llbracket \cdot \rrbracket} \rightarrow (\llbracket S \rrbracket, T) \in \mathcal{R}_\top) \wedge \mathcal{R}_{\llbracket \cdot \rrbracket}$ is a preorder and a probabilistic correspondence simulation.

To prove this theorem, we have to construct (for the if-case) a relation $\mathcal{R}_{\llbracket \cdot \rrbracket}$ from \mathcal{R}_\top . Therefore, we use $\mathbf{t}(\mathbf{r}(\mathcal{R}_\top \cup \{(S, \llbracket S \rrbracket) \mid S \in \mathcal{P}_S\}))$, where \mathbf{t} and \mathbf{r} denote transitive and reflexive closure. Then we show that in both directions the respective properties imply each other. In particular, we establish the relation between weak PrOC and the definition of probabilistic correspondence simulation, i.e., completeness of weak PrOC in Definition 8

$$\forall S, \Delta_S. S \Longrightarrow \Delta_S \rightarrow (\exists \Delta_T. \llbracket S \rrbracket \Longrightarrow \Delta_T \wedge (\llbracket \Delta_S \rrbracket, \Delta_T) \in \overline{\mathcal{R}_\top})$$

is mapped on the first condition of Definition 12

$$P \Longrightarrow \Delta \rightarrow (\exists \Theta. Q \Longrightarrow \Theta \wedge (\Delta, \Theta) \in \overline{\mathcal{R}})$$

and weak soundness of weak PrOC

$$\forall S, \Delta_T. \llbracket S \rrbracket \Longrightarrow \Delta_T \longrightarrow (\exists \Delta_S, \Delta'_T. S \Longrightarrow \Delta_S \wedge \Delta_T \Longrightarrow \Delta'_T \wedge (\llbracket \Delta_S \rrbracket, \Delta'_T) \in \overline{\mathcal{R}_T})$$

is mapped on the seconded condition of Definition 12:

$$Q \Longrightarrow \Theta \longrightarrow (\exists \Delta', \Theta'. P \Longrightarrow \Delta' \wedge \Theta \Longrightarrow \Theta' \wedge (\Delta', \Theta') \in \overline{\mathcal{R}})$$

The condition $\forall S, T. (S, T) \in \mathcal{R}_{[\cdot, \cdot]} \longrightarrow (\llbracket S \rrbracket, T) \in \mathcal{R}_T$ is necessary to ensure (with the remaining properties) that the encoding satisfies weak PrOC in the “only if”-case of Theorem 14. Although the formulation of weak PrOC and probabilistic correspondence simulation are quite close, to prove that they are related is technically challenging. We have to show that the properties of the involved relations are preserved when we lift the relations on distributions with Definition 2 and that the probabilistic version of the correspondence similarity exactly captures the probabilistic nature of weak PrOC.

The property $\forall S. (S, \llbracket S \rrbracket) \in \mathcal{R}_{[\cdot, \cdot]}$ allows us to conclude from pairs of target terms in \mathcal{R}_T and $\mathcal{R}_{[\cdot, \cdot]}$ on pairs of a source and a target term. This is necessary to prove that the encoding satisfies weak PrOC in the “only if”-case, but also ensures that $\mathcal{R}_{[\cdot, \cdot]}$ is a relation that relates source terms with their literal translations. From Theorem 9 and Theorem 14 we can thus conclude that the encoding $(\cdot)_{\text{CCS}_p}^{\pi_p}$ from CCS_p into π_p relates a source term $S \in \mathcal{P}_C$ and its literal translation $(S)_{\text{CCS}_p}^{\pi_p}$ by a probabilistic correspondence simulation.

As derived in [30], weak operational correspondence does not directly map to a well-known, i.e., standard, kind of simulation relation, but is linked to the new relation correspondence similarity. Correspondence similarity is in between the standard simulation relations coupled similarity (see e.g. [27, 3]) and bisimilarity (see e.g. [23]).

Coupled similarity is strictly weaker than bisimilarity. As pointed out in [27], in contrast to bisimilarity it allows for intermediate states in simulations: states that cannot be identified with states of the simulated term. Each symmetric coupled simulation is a bisimulation.

► **Definition 15** (Coupled Simulation). *A relation \mathcal{R} is a (weak reduction) coupled simulation if both $(\exists Q'. Q \Longrightarrow Q' \wedge (P', Q') \in \mathcal{R})$ and $(\exists Q'. Q \Longrightarrow Q' \wedge (Q', P') \in \mathcal{R})$ whenever $(P, Q) \in \mathcal{R}$ and $P \Longrightarrow P'$. Two terms are coupled similar if they are related by a coupled simulation in both directions.*

Just as coupled similarity, correspondence similarity allows for intermediate states that result e.g. from partial commitments, but in contrast to coupled similarity these intermediate states are not necessarily covered in the relation. Correspondence similarity is obviously strictly weaker than bisimilarity, but as shown in [30] it implies coupled similarity. The same holds for the probabilistic variants of correspondence similarity and coupled similarity, where probabilistic coupled similarity is the adaptation of coupled similarity to distributions using Definition 2.

► **Definition 16** (Probabilistic Coupled Simulation). *A relation \mathcal{R} is a (weak) probabilistic (reduction) coupled simulation if we have both $(\exists \Theta. Q \Longrightarrow \Theta \wedge (\Delta, \Theta) \in \overline{\mathcal{R}})$ and $(\exists \Theta. Q \Longrightarrow \Theta \wedge (\Theta, \Delta) \in \overline{\mathcal{R}})$ whenever $(P, Q) \in \mathcal{R}$ and $P \Longrightarrow \Delta$. Two terms are probabilistic coupled similar if they are related by a probabilistic coupled simulation in both directions.*

For each probabilistic correspondence simulation \mathcal{R} there exists a probabilistic coupled simulation \mathcal{R}' such that $\forall (P, Q) \in \mathcal{R}. (P, Q), (Q, P) \in \mathcal{R}'$.

15:12 Probabilistic Operational Correspondence

Intuitively, coupled similarity (and also probabilistic coupled similarity) is the strictest standard simulation relation that allows for intermediate states. Accordingly, that a weak probabilistically operationally corresponding encoding ensures that a source term is probabilistic coupled similar to its literal translation is indeed an interesting property (see e.g. [27, 3] for the relevance of coupled similarity). This proves that our version of weak PrOC is meaningful. As described in [30], the combination with the criteria divergence reflection and success sensitiveness further strengthens the induced relation between source and target, i.e., we obtain a divergence reflecting, success sensitive, probabilistic correspondence (or coupled) simulation to relate source terms and their literal translations.

Weak operational correspondence is very flexible and allows us to encode source term concepts that have no direct counterpart in the target. As discussed in [27, 17, 29], relating source terms and their literal translations by a bisimulation does not allow for intermediate states, i.e., for bisimulation we need stricter encodability criteria as discussed next.

6 (Strong) Probabilistic Operational Correspondence

In this section we proceed in the opposite direction. We start with a standard simulation relation and derive a version of PrOC that induces such a relation between source terms and their translations. Often bisimilarity (see e.g. [23]) is considered as *the* standard relation between processes. Accordingly, we start from a probabilistic version of bisimilarity, namely probabilistic barbed bisimilarity as introduced and analysed in [7].

► **Definition 17** (Probabilistic Barbed Bisimulation, [7]). *An equivalence \mathcal{R} is a probabilistic barbed bisimulation if for each $(P, Q) \in \mathcal{R}$:*

- $P \mapsto \Delta$ implies $\exists \Theta. Q \Longrightarrow \Theta \wedge (\Delta, \Theta) \in \overline{\mathcal{R}}$
- for each atomic action a , if $P \downarrow_a$ then $Q \downarrow_a$

Two terms are probabilistic barbed bisimilar if a probabilistic bisimulation relates them.

Here $P \downarrow_a$ if $P \xrightarrow{a} \Delta$, and $\Delta \downarrow_a$ if $P \downarrow_a$ for all $P \in [\Delta]$ with $a \in \mathcal{N} \cup \overline{\mathcal{N}}$. Moreover, $P \downarrow_a = \exists \Delta. P \Longrightarrow \Delta \wedge \Delta \downarrow_a$ and $\Delta \downarrow_a = \exists \Delta'. \Delta \Longrightarrow \Delta' \wedge \Delta' \downarrow_a$. We use this notion of barbs on our source language CCS_p . Accordingly, barbs $\cdot \downarrow_a$ are defined via labelled steps. The versions of operational correspondence that we considered so far do not use labelled but only reduction steps. This is because the treatment of labels in encodings can be difficult. For instance the labels in Mobile Ambients ([5]) are technically and conceptionally very different from the labels in the Join-calculus ([11]) and both kinds of labels are technically and conceptionally very different from labels in CCS or the π -calculus. The consideration of labelled steps is only meaningful if the considered source and target language use similar kinds of labels. Because of that, operational correspondence usually considers reduction steps only. If the languages have similar notions of barbs, success sensitiveness can be replaced by barbed sensitiveness to establish this connection. We remove the condition on barbs.

► **Definition 18** (Probabilistic Bisimulation). *A relation \mathcal{R} is a probabilistic (reduction) bisimulation if for each $(P, Q) \in \mathcal{R}$:*

- $P \Longrightarrow \Delta$ implies $\exists \Theta. Q \Longrightarrow \Theta \wedge (\Delta, \Theta) \in \overline{\mathcal{R}}$
- $Q \Longrightarrow \Theta$ implies $\exists \Delta. P \Longrightarrow \Delta \wedge (\Delta, \Theta) \in \overline{\mathcal{R}}$

Two terms are probabilistic bisimilar if a probabilistic bisimulation relates them.

In comparison to Definition 17 we also replace $P \mapsto \Delta$ by $P \Longrightarrow \Delta$, remove the condition of \mathcal{R} being an equivalence, and add the symmetric case of the first condition. These adaptations have little consequence on the derived relation, but provide a structure more closely related to operational correspondence.

Note that the above notion of probabilistic bisimulation without barbs is trivial. However, also operational correspondence is a trivial encodability relation unless we combine it with success sensitiveness. We discuss the combination with success or barbs below.

We want to derive a version of PrOC for probabilistic bisimilarity. As we learnt in Section 5, the first condition

$$P \Longrightarrow \Delta \longrightarrow \exists \Theta. Q \Longrightarrow \Theta \wedge (\Delta, \Theta) \in \overline{\mathcal{R}}$$

for the left part of pairs has to be linked with completeness. Since it is identical to the first condition of probabilistic correspondence simulation in Definition 12 we can link it to the same version of completeness:

$$\forall S, \Delta_S. S \Longrightarrow \Delta_S \longrightarrow (\exists \Delta_T. \llbracket S \rrbracket \Longrightarrow \Delta_T \wedge (\llbracket \Delta_S \rrbracket, \Delta_T) \in \overline{\mathcal{R}_T})$$

The second condition

$$Q \Longrightarrow \Theta \longrightarrow \exists \Delta. P \Longrightarrow \Delta \wedge (\Delta, \Theta) \in \overline{\mathcal{R}}$$

of Definition 18 is simpler than the second condition of probabilistic correspondence simulation in Definition 12. The part that allows for intermediate states is missing. Thus, we link it to a similarly simplified version of soundness:

$$\forall S, \Delta_T. \llbracket S \rrbracket \Longrightarrow \Delta_T \longrightarrow (\exists \Delta_S. S \Longrightarrow \Delta_S \wedge (\llbracket \Delta_S \rrbracket, \Delta_T) \in \overline{\mathcal{R}_T})$$

We denote the result as probabilistic operational correspondence or shortly as PrOC.

► **Definition 19** (Probabilistic Operational Correspondence). *An encoding $\llbracket \cdot \rrbracket : \mathcal{P}_S \rightarrow \mathcal{P}_T$ is probabilistic operationally corresponding (PrOC) w.r.t. $\mathcal{R}_T \subseteq \mathcal{P}_T^2$ if it is:*

Probabilistic Complete:

$$\forall S, \Delta_S. S \Longrightarrow \Delta_S \text{ implies } (\exists \Delta_T. \llbracket S \rrbracket \Longrightarrow \Delta_T \wedge (\llbracket \Delta_S \rrbracket, \Delta_T) \in \overline{\mathcal{R}_T})$$

Probabilistic Sound:

$$\forall S, \Delta_T. \llbracket S \rrbracket \Longrightarrow \Delta_T \text{ implies } (\exists \Delta_S. S \Longrightarrow \Delta_S \wedge (\llbracket \Delta_S \rrbracket, \Delta_T) \in \overline{\mathcal{R}_T})$$

Of course we have to formally establish the described connection between probabilistic bisimilarity and PrOC.

► **Theorem 20** (PrOC). $\llbracket \cdot \rrbracket$ is probabilistically operationally corresponding w.r.t. a preorder $\mathcal{R}_T \subseteq \mathcal{P}_T^2$ that is a probabilistic bisimulation iff $\exists \mathcal{R}_{\llbracket \cdot \rrbracket}. (\forall S. (S, \llbracket S \rrbracket) \in \mathcal{R}_{\llbracket \cdot \rrbracket}) \wedge \mathcal{R}_T = \mathcal{R}_{\llbracket \cdot \rrbracket} \upharpoonright_{\mathcal{P}_T} \wedge (\forall S, T. (S, T) \in \mathcal{R}_{\llbracket \cdot \rrbracket} \longrightarrow (\llbracket S \rrbracket, T) \in \mathcal{R}_T) \wedge \mathcal{R}_{\llbracket \cdot \rrbracket}$ is a preorder and a probabilistic bisimulation.

The proof of Theorem 20 is as challenging as the proof of Theorem 14 but fortunately we can reuse the main proof strategy.

In the same way, we can also link strong probabilistic bisimilarity and a strong version of PrOC. We obtain strong probabilistic bisimilarity by considering only single steps.

► **Definition 21** (Strong Probabilistic Bisimulation). *A relation \mathcal{R} is a strong probabilistic (reduction) bisimulation if for each $(P, Q) \in \mathcal{R}$:*

- $P \mapsto \Delta$ implies $\exists \Theta. Q \mapsto \Theta \wedge (\Delta, \Theta) \in \overline{\mathcal{R}}$
- $Q \mapsto \Theta$ implies $\exists \Delta. P \mapsto \Delta \wedge (\Delta, \Theta) \in \overline{\mathcal{R}}$

Two terms are strong probabilistic bisimilar if a strong probabilistic bisimulation relates them.

Strong probabilistic operational correspondence is obtained in a similar way from PrOC by considering only single steps.

► **Definition 22** (Strong Probabilistic Operational Correspondence). *An encoding $\llbracket \cdot \rrbracket : \mathcal{P}_S \rightarrow \mathcal{P}_T$ is strongly probabilistic operationally corresponding (strong PrOC) w.r.t. $\mathcal{R}_T \subseteq \mathcal{P}_T^2$ if it is:*

Strongly Probabilistic Complete:

$$\forall S, \Delta_S. S \mapsto \Delta_S \text{ implies } (\exists \Delta_T. \llbracket S \rrbracket \mapsto \Delta_T \wedge (\llbracket \Delta_S \rrbracket, \Delta_T) \in \overline{\mathcal{R}_T})$$

Strongly Probabilistic Sound:

$$\forall S, \Delta_T. \llbracket S \rrbracket \mapsto \Delta_T \text{ implies } (\exists \Delta_S. S \mapsto \Delta_S \wedge (\llbracket \Delta_S \rrbracket, \Delta_T) \in \overline{\mathcal{R}_T})$$

Finally, we establish the connection between strong PrOC and strong probabilistic bisimilarity in Theorem 23. The proofs of Theorem 20 and Theorem 23 can be found in [33].

► **Theorem 23** (Strong PrOC). *$\llbracket \cdot \rrbracket$ is strongly probabilistically operationally corresponding w.r.t. a preorder $\mathcal{R}_T \subseteq \mathcal{P}_T^2$ that is a strong probabilistic bisimulation iff*

$\exists \mathcal{R}_{\llbracket \cdot \rrbracket}. (\forall S. (S, \llbracket S \rrbracket) \in \mathcal{R}_{\llbracket \cdot \rrbracket}) \wedge \mathcal{R}_T = \mathcal{R}_{\llbracket \cdot \rrbracket} \upharpoonright_{\mathcal{P}_T} \wedge (\forall S, T. (S, T) \in \mathcal{R}_{\llbracket \cdot \rrbracket} \longrightarrow (\llbracket S \rrbracket, T) \in \mathcal{R}_T) \wedge \mathcal{R}_{\llbracket \cdot \rrbracket}$ is a preorder and a strong probabilistic bisimulation.

If (strong) bisimilarity is *the* standard reference relation, i.e., if we usually do not record differences between terms that cannot be observed by (strong) probabilistic bisimilarity, then an encoding that ensures that source terms and their translations are (strongly) probabilistic bisimilar strongly validates the claim that the target language is at least as expressive as the source language. In this sense PrOC and strong PrOC are strict but also very meaningful criteria.

Again the combination with the criteria divergence reflection and success sensitiveness further strengthens the induced relation between source and target (see [30]), i.e., we obtain a divergence reflecting, success sensitive, (strong) probabilistic bisimulation to relate source terms and their literal translations.

As discussed above, the consideration of labels and barbs is difficult in the context of encodings unless the source and target language have very similar notions of labels and barbs. In our source language CCS_p , labels are of the form x or \bar{x} , whereas by [38] our target language π_p uses labels of the form $x \text{in}_i \langle \tilde{y}_i \rangle$, $\bar{x} \text{in}_i \langle \tilde{y}_i \rangle$, $x \langle \tilde{y} \rangle$, and $\bar{x} \langle \tilde{y} \rangle$. The main difference are the transmitted values and variables for reception. However, for all terms that are created by our encoding function $(\cdot)_{\text{CCS}_p}^{\pi_p}$ in Definition 6, i.e., all terms in Δ_T such that $(\llbracket S \rrbracket)_{\text{CCS}_p}^{\pi_p} \Longrightarrow \Delta_T$ for some source term S , all visible labels are of the form $x \text{in}_i \langle \cdot \rangle$ or $\bar{x} \text{in}_i \langle \cdot \rangle$. Labels of the form $x \langle \tilde{y} \rangle$ and $\bar{x} \langle \tilde{y} \rangle$ are used to emulate the unfolding of recursion but the respective channel names C_i are restricted, i.e., not visible. Moreover, CCS-like barbs, i.e., barbs without values or variables, are often also used for variants of the π -calculus.

This observation allows us to define a suitable notion of barbs for our target language π_p ¹. Let $P \downarrow_x$ if $P \left\{ \frac{x \text{in}_i \langle \cdot \rangle}{p_i} \rightarrow T_i \right\}_{i \in I}$, $P \downarrow_{\bar{x}}$ if $P \left\{ \frac{\bar{x} \text{in}_i \langle \cdot \rangle}{p_i} \rightarrow T_i \right\}_{i \in I}$, and $\Delta \downarrow_a$ if $P \downarrow_a$ for all $P \in [\Delta]$ with $a \in \mathcal{N} \cup \overline{\mathcal{N}}$. Moreover, $P \downarrow_a = \exists \Delta. P \Longrightarrow \Delta \wedge \Delta \downarrow_a$ and $\Delta \downarrow_a = \exists \Delta'. \Delta \Longrightarrow \Delta' \wedge \Delta' \downarrow_a$.

Note that our encoding $(\cdot)_{\text{CCS}_p}^{\pi_p}$ does not translate an input on x to a step with label $x \text{in}_i \langle \cdot \rangle$ but with label $\varphi_{(\cdot)_{\text{CCS}_p}^{\pi_p}}(x) \text{in}_i \langle \cdot \rangle$, i.e., we should not forget the renaming policy. Accordingly we compare the reachability of a barb n in the source, i.e., $S \downarrow_n$, with the reachability of the translated barb $\varphi_{(\cdot)_{\text{CCS}_p}^{\pi_p}}(n)$ in the translation, i.e., $(\llbracket S \rrbracket)_{\text{CCS}_p}^{\pi_p} \downarrow_{\varphi_{(\cdot)_{\text{CCS}_p}^{\pi_p}}(n)}$. With these notions of barbs on the source and target language, we can show that our encoding $(\cdot)_{\text{CCS}_p}^{\pi_p}$ is barb sensitive.

¹ Remember that the semantics of π_p is given as a Segala automaton, i.e., $P \left\{ \frac{\alpha_i}{p_i} \rightarrow T_i \right\}_{i \in I}$ means that for every $i \in I$ the term P can do a step to T_i with label α_i and probability p_i (see [38, 33]).

► **Lemma 24** (Barb Sensitiveness, $(\cdot)_{\text{CCS}_p}^{\pi_p} / \llbracket \cdot \rrbracket_{\text{CCS}_p}^{\pi_p}$).

For every S and all $n \in \mathcal{N} \cup \bar{\mathcal{N}}$, $S \Downarrow_n$ iff $(S)_{\text{CCS}_p}^{\pi_p} \Downarrow_{\varphi_{(\cdot)_{\text{CCS}_p}^{\pi_p}}(n)}$.

If we combine, as described in [30], barb sensitiveness with PrOC, we obtain that all source terms S are related to $\llbracket S \rrbracket$ by a probabilistic barbed bisimulation as described in Definition 17. However, since our encoding $(\cdot)_{\text{CCS}_p}^{\pi_p}$ satisfies only weak PrOC and not PrOC, S and $(S)_{\text{CCS}_p}^{\pi_p}$ are related by a probabilistic barbed correspondence (or coupled) simulation.

7 Conclusions

We provided three notions of probabilistic operational correspondence:

1. Weak Probabilistic Operational Correspondence (weak PrOC), where single source term steps can be translated by sequences of target term steps and intermediate states in the translation are possible.
2. Probabilistic Operational Correspondence (PrOC), where single source term steps can be translated by sequences of target term steps but intermediate states are forbidden.
3. Strong Probabilistic Operational Correspondence (strong PrOC), where a single source term step has to be translated to a single target term step.

We proved that strong PrOC induces a strong probabilistic bisimulation between source terms and their literal translations, i.e., strong PrOC is a very strict criterion that ensures a close connection between the source and target language. In contrast weak PrOC induces a probabilistic correspondence (or coupled) simulation between source terms and their literal translations. This allows for pre- and post-processing steps in the encoding and even for intermediate states, i.e., for more flexibility in the creation of encoding functions.

Related Work. There are several papers such as e.g. [4, 24, 14, 15, 13, 12, 36, 37, 16, 25, 26, 29, 30] that study quality criteria for encodings in the traditional setting. As far as we know, there are no studies of quality criteria for encodings between probabilistic systems.

Probabilistic versions of bisimulation for process calculi are studied e.g. in [18, 20, 34, 1, 2, 7]. Encodings between concrete probabilistic process calculi are studied e.g. in [35, 19]. They argue for the quality of the specific presented encodings but do not derive quality criteria for encodings in general. For instance [35] compares two versions of the probabilistic π -calculus (one with mixed choice and one with only separate choice). Since the two versions of the considered language are close, they prove the quality of their encoding by showing a direct correspondence between labelled steps of the respective source and target language. Essentially they prove a labelled variant of weak PrOC. However, as discussed above the consideration of labels in encodings is difficult, because different languages usually have very different notions of labels. Because of that, we use reduction steps and barbs or success for our general formulation of quality criteria. Moreover, since [35, 19] do not present general quality criteria for encodings, they also do not discuss the quality of such criteria.

We are focusing on encodings between process calculi. Instead [38] connects the probabilistic π -calculus and event structures. Therefore, they show an operational correspondence between the semantics of the π -calculus and event structures (see Theorem 6.3 in Section 6.2 of [38]). Their formulation of operational correspondence is basically a variant of strong PrOC in Definition 22.

Further Work. Our original motivation to study versions of PrOC steamed from quantum based systems. As probabilistic versions of simulation relations are essential for studying quantum based systems (see e.g. [10, 9, 8, 6]), this work supports the development of formal methods for quantum based systems.

References

- 1 C. Baier and H. Hermanns. Weak Bisimulation for Fully Probabilistic Processes. In *Computer Aided Verification*, pages 119–130. Springer, 1997. doi:10.1007/3-540-63166-6_14.
- 2 C. Baier, H. Hermanns, J.-P. Katoen, and V. Wolf. Bisimulation and Simulation Relations for Markov Chains. *Electronic Notes in Theoretical Computer Science*, 162:73–78, 2006. doi:10.1016/j.entcs.2005.12.078.
- 3 B. Bisping, U. Nestmann, and K. Peters. Coupled Similarity: the first 32 years. *Acta Informatica*, 57:439–463, 2019. doi:10.1007/s00236-019-00356-4.
- 4 G. Boudol. Asynchrony and the π -calculus (note). Note, INRIA, 1992.
- 5 L. Cardelli and A.D. Gordon. Mobile Ambients. *Theoretical Computer Science*, 240(1):177–213, 2000. doi:10.1016/S0304-3975(99)00231-5.
- 6 Y. Deng. Bisimulations for Probabilistic and Quantum Processes. In *Proc. of CONCUR*, volume 118, pages 2:1–2:14, 2018. doi:10.4230/LIPIcs.CONCUR.2018.2.
- 7 Y. Deng and W. Du. Probabilistic Barbed Congruence. *Electronic Notes in Theoretical Computer Science*, 190(3):185–203, 2007. doi:10.1016/j.entcs.2007.07.011.
- 8 Y. Deng and Y. Feng. Open Bisimulation for Quantum Processes. In *Proc. of TCS*, volume 7604, pages 119–133. Springer, 2012. doi:10.1007/978-3-642-33475-7_9.
- 9 Y. Feng, Y. Deng, and M. Ying. Symbolic Bisimulation for Quantum Processes. *CoRR*, 2012. doi:10.48550/arXiv.1202.3484.
- 10 Y. Feng, R. Duan, and M. Ying. Bisimulation for Quantum Processes. *ACM Trans. Program. Lang. Syst.*, 34(4):17:1–17:43, 2012. doi:10.1145/2400676.2400680.
- 11 C. Fournet and G. Gonthier. The Reflexive CHAM and the Join-Calculus. In *Proc. of POPL*, pages 372–385. ACM, 1996. doi:10.1145/237721.237805.
- 12 Y. Fu. Theory of Interaction. *Theoretical Computer Science*, 611:1–49, 2016. doi:10.1016/j.tcs.2015.07.043.
- 13 Y. Fu and H. Lu. On the expressiveness of interaction. *Theoretical Computer Science*, 411(11-13):1387–1451, 2010. doi:10.1016/j.tcs.2009.11.011.
- 14 D. Gorla. Towards a Unified Approach to Encodability and Separation Results for Process Calculi. In *Proc. of CONCUR*, volume 5201, pages 492–507. Springer, 2008. doi:10.1007/978-3-540-85361-9_38.
- 15 D. Gorla. Towards a Unified Approach to Encodability and Separation Results for Process Calculi. *Information and Computation*, 208(9):1031–1053, 2010. doi:10.1016/j.ic.2010.05.002.
- 16 D. Gorla and U. Nestmann. Full abstraction for expressiveness: history, myths and facts. *Mathematical Structures in Computer Science*, pages 1–16, 2014. doi:10.1017/S0960129514000279.
- 17 M. Hatzel, C. Wagner, K. Peters, and U. Nestmann. Encoding CSP into CCS. In *Proc. of EXPRESS/SOS*, volume 7, pages 61–75, 2015. doi:10.4204/EPTCS.190.5.
- 18 C. Jou and S.A. Smolka. Equivalences, Congruences, and Complete Axiomatizations for Probabilistic Processes. In *Proc. of CONCUR*, pages 367–383, 1990. doi:10.1007/BFb0039071.
- 19 M. Kwiatkowska, G. Norman, D. Parker, and M.G. Vigliotti. Probabilistic Mobile Ambients. *Theoretical Computer Science*, 410:1272–1303, 2009. doi:10.1016/j.tcs.2008.12.058.
- 20 K.G. Larsen and A. Skou. Bisimulation through Probabilistic Testing. *Information and Computation*, 94(1):1–28, 1991. doi:10.1016/0890-5401(91)90030-6.
- 21 R. Milner. *Communication and Concurrency*. Prentice-Hall, Inc., 1989.
- 22 R. Milner. *Communicating and mobile systems - the Pi-calculus*. Cambridge University Press, 1999.
- 23 R. Milner and D. Sangiorgi. Barbed Bisimulation. In *Automata, Languages and Programming*, pages 685–695, 1992.
- 24 C. Palamidessi. Comparing the Expressive Power of the Synchronous and the Asynchronous π -calculi. *Mathematical Structures in Computer Science*, 13(5):685–719, 2003. doi:10.1017/S0960129503004043.

- 25 J. Parrow. Expressiveness of Process Algebras. *Electronic Notes in Theoretical Computer Science*, 209:173–186, 2008. doi:10.1016/j.entcs.2008.04.011.
- 26 J. Parrow. General conditions for full abstraction. *Mathematical Structures in Computer Science*, 26(4):655–657, 2014. doi:10.1017/S0960129514000280.
- 27 J. Parrow and P. Sjödin. Multiway Synchronization Verified with Coupled Simulation. In *Proc. of CONCUR*, volume 630 of *LNCS*, pages 518–533, 1992. doi:10.1007/BFb0084813.
- 28 K. Peters. *Translational Expressiveness*. PhD thesis, TU Berlin, 2012. URL: <http://opus.kobv.de/tuberlin/volltexte/2012/3749/>.
- 29 K. Peters. Comparing Process Calculi Using Encodings. In *Proc. of EXPRESS/SOS*, EPTCS, pages 19–38, 2019. doi:10.48550/arXiv.1908.08633.
- 30 K. Peters and R. van Glabbeek. Analysing and Comparing Encodability Criteria. In *Proc. of EXPRESS/SOS*, volume 190 of *EPTCS*, pages 46–60, 2015. doi:10.4204/EPTCS.190.4.
- 31 G.D. Plotkin. A structural approach to operational semantics. *Log. Algebraic Methods Program.*, 60-61:17–139, 2004.
- 32 D. Sangiorgi. π -Calculus, internal mobility, and agent-passing calculi. *Theoretical Computer Science*, 167(1):235–274, 1996. doi:10.1016/0304-3975(96)00075-8.
- 33 Anna Schmitt and Kirstin Peters. Probabilistic Operational Correspondence (Technical Report). Technical report, TU Darmstadt and Augsburg University, 2023. doi:10.48550/arXiv.2307.05218.
- 34 R. Segala and N. Lynch. Probabilistic Simulations for Probabilistic Processes. In *Proc. of CONCUR*, pages 481–496, 1994. doi:10.1007/978-3-540-48654-1_35.
- 35 P. Sylvain and C. Palamidessi. Expressiveness of Probabilistic π -calculus. *Electronic Notes in Theoretical Computer Science*, 164:119–136, 2006. doi:10.1016/j.entcs.2006.07.015.
- 36 R. van Glabbeek. Musings on Encodings and Expressiveness. In *Proc. of EXPRESS/SOS*, volume 89, pages 81–98, 2012. doi:10.4204/EPTCS.89.7.
- 37 R. van Glabbeek. A Theory of Encodings and Expressiveness (Extended Abstract). In *Proc. of FOSSACS*, volume 10803, pages 183–202. Springer, 2018. doi:10.1007/978-3-319-89366-2_10.
- 38 D. Varacca and N. Yoshida. Probabilistic pi-Calculus and Event Structures. *Electronic Notes in Theoretical Computer Science*, 190(3):147–166, 2007. doi:10.1016/j.entcs.2007.07.009.