Probabilistic process algebra and strategic interleaving

Middelburg, C.A.

Publication date
2019

Document Version
Final published version

Link to publication

Citation for published version (APA):
Middelburg, C. A. (2019). *Probabilistic process algebra and strategic interleaving*. (1 ed.) Informatics Institute, University of Amsterdam. https://arxiv.org/abs/1912.10041v1

Probabilistic Process Algebra and Strategic Interleaving

C.A. Middelburg

Informatics Institute, Faculty of Science, University of Amsterdam,
Science Park 904, 1098 XH Amsterdam, the Netherlands
C.A.Middelburg@uva.nl

Abstract. We first present a probabilistic version of ACP that rests on the principle that probabilistic choices are resolved before choices involved in alternative composition and parallel composition are resolved and then extend this probabilistic version of ACP with a form of interleaving in which parallel processes are interleaved according to what is known as a process-scheduling policy in the field of operating systems. We use the term strategic interleaving for this more constrained form of interleaving and the term interleaving strategy for process-scheduling policy. The extension covers probabilistic interleaving strategies.

Keywords: process algebra, probabilistic choice, parallel composition, arbitrary interleaving, strategic interleaving, probabilistic interleaving strategy

1998 ACM Computing Classification: D.1.3, D.4.1, F.1.2

1 Introduction

First of all, we present a probabilistic version of ACP [12,9], called pACP (probabilistic ACP). pACP is a minor variant of the subtheory of pACP_τ [5] in which the operators for abstraction from certain actions are lacking. In pACP, we take functions whose range is the carrier of a signed cancellation meadow (see below) instead of a field as probability measures, add probabilistic choice operators for the probabilities 0 and 1, and have an additional axiom because of the inclusion of these operators (axiom pA6). The probabilistic choice operators for the probabilities 0 and 1 cause no problem because a meadow has a total multiplicative inverse operation where the multiplicative inverse of zero is zero. Because of this property, we could also improve the operational semantics. In particular, we could reduce the number of rules for the operational semantics and replace all negative premises by positive premises in the remaining rules.

We also extend pACP with a form of interleaving in which parallel processes are interleaved according to what is known as a process-scheduling policy in the field of operating systems. In [14], we have extended ACP with this more constrained form of interleaving. In that paper, we introduced the term strategic interleaving for this form of interleaving and the term interleaving strategy for process-scheduling policy. Unlike in the extension presented in [14], probabilistic interleaving strategies are covered in the extension presented in the current paper. More precisely, the latter extension assumes a generic interleaving strategy that can be instantiated with different specific interleaving strategies, including probabilistic ones.

Our interest in strategic interleaving originates from an important feature of many contemporary programming languages, namely multi-threading. In algebraic theories of processes, such as ACP [9], CCS [25], and CSP [24], processes are discrete behaviours that proceed by doing steps in a sequential fashion. In these theories, parallel composition of two processes is usually interpreted as arbitrary interleaving of the steps of the processes concerned. Arbitrary interleaving turns out to be appropriate for many applications and to facilitate formal algebraic reasoning. Multi-threading as found in programming languages such as Java [22] and C# [23], gives rise to parallel composition of processes. In the case of multi-threading, however, the steps of the processes concerned are interleaved according to what is known as a process-scheduling policy in the field of operating systems.

The fact that the multiplicative inverse of zero is undefined in a field quite often complicates matters. In [17], meadows have been proposed as alternatives for fields with a purely equational axiomatization. Meadows are commutative rings with a multiplicative identity element and a total multiplicative inverse operation where the multiplicative inverse of zero is zero. Cancellation meadows, with the exception of the trivial one, are fields whose multiplicative inverse operation is made total by imposing that the multiplicative inverse of zero is zero, and signed cancellation meadows are cancellation meadows expanded with a signum operation. In [16], Kolmogorov's probability axioms for finitely additive probability spaces are rephrased for the case where probability measures are functions whose range is the carrier of a signed cancellation meadow.

To our knowledge, the work presented in [14] and this paper is the only work on strategic interleaving in the setting of a general algebraic theory of processes like ACP, CCS and CSP. In this work, we consider strategic interleaving where process creation is taken into account. The approach to process creation followed originates from the one first followed in [10] to extend ACP with process creation and later followed in [6,15,7] to extend different timed versions of ACP with process creation. The only other approach that we know of is the approach, based on [2], that has for instance been followed in [8,19]. However, with that approach, it is most unlikely that data about the creation of processes can be made available for the decision making concerning the strategic interleaving of processes.

The rest of this paper is organized as follows. First, we briefly summarize the theory of signed cancellation meadows (Section 2). Next, we present pACP and its extension with guarded recursion (Section 3). After that, we present the extension of the resulting theory with strategic interleaving (Section 4). Finally, we make some concluding remarks (Section 5).

Table 1. Axioms of a meadow

$(x+y)+z = x+(y+z)$	$(x \cdot y) \cdot z = x \cdot (y \cdot z)$	$(x^{-1})^{-1} = x$
$x+y = y+x$	$x \cdot y = y \cdot x$	$x \cdot (x \cdot x^{-1}) = x$
$x+0 = x$	$x \cdot 1 = x$	
$x+(-x) = 0$	$x \cdot (y+z) = x \cdot y + x \cdot z$	

Table 2. Additional axioms for the signum operator

$\mathsf{s}(x/x) = x/x$	$\mathsf{s}(x^{-1}) = \mathsf{s}(x)$
$\mathsf{s}(1 - x/x) = 1 - x/x$	$\mathsf{s}(x \cdot y) = \mathsf{s}(x) \cdot \mathsf{s}(y)$
$\mathsf{s}(-1) = -1$	$(1 - \frac{\mathsf{s}(x)-\mathsf{s}(y)}{\mathsf{s}(x)-\mathsf{s}(y)}) \cdot (\mathsf{s}(x+y) - \mathsf{s}(x)) = 0$

2 Signed Cancellation Meadows

Later in this paper, we will take functions whose range is the carrier of a signed cancellation meadow as probability measures. Therefore, we briefly summarize the theory of signed cancellation meadows in this section.

In [17], meadows are proposed as alternatives for fields with a purely equational axiomatization. Meadows are commutative rings with a multiplicative identity element and a total multiplicative inverse operation where the multiplicative inverse of zero is zero. Fields whose multiplicative inverse operation is made total by imposing that the multiplicative inverse of zero is zero are called zero-totalized fields. All zero-totalized fields are meadows, but not conversely.

Cancellation meadows are meadows that satisfy the *cancellation axiom* $x \neq 0 \wedge x \cdot y = x \cdot z \Rightarrow y = z$. The cancellation meadows that satisfy in addition the *separation axiom* $0 \neq 1$ are exactly the zero-totalized fields.

Signed cancellation meadows are cancellation meadows expanded with a signum operation. The signum operation makes it possible that the predicates $<$ and $\leq$ are defined (see below).

The signature of signed cancellation meadows consists of the following constants and operators: the constants 0 and 1, the binary *addition* operator $+$, the binary *multiplication* operator $\cdot$, the unary *additive inverse* operator $-$, the unary *multiplicative inverse* operator $^{-1}$, and the unary *signum* operator s.

Terms are build as usual. We use prefix notation, infix notation, and postfix notation as usual. We also use the usual precedence convention. We introduce subtraction and division as abbreviations: $t - t'$ abbreviates $t + (-t')$ and t/t' abbreviates $t \cdot (t'^{-1})$.

Signed cancellation meadows are axiomatized by the equations in Tables 1 and 2 and the above-mentioned cancellation axiom.

The predicates $<$ and $\leq$ are defined in signed cancellation meadows as follows: $x < y \Leftrightarrow \mathsf{s}(y-x) = 1$ and $x \leq y \Leftrightarrow \mathsf{s}(\mathsf{s}(y-x)+1) = 1$. Because $\mathsf{s}(\mathsf{s}(y-x)+1) \neq -1$, we have $0 \leq x \leq 1 \Leftrightarrow \mathsf{s}(\mathsf{s}(x)+1) \cdot \mathsf{s}(\mathsf{s}(1-x)+1) = 1$. We will use this equivalence below to describe the set of probabilities.

3 pACP with Guarded Recursion

In this section, we introduce pACP (probabilistic Algebra of Communicating Processes), guarded recursion in the setting of pACP, and some relevant results about the extension of pACP with guarded recursion. The algebraic theory pACP is a minor variant of the subtheory of pACP_τ [5] in which the operators for abstraction from certain actions are lacking. In pACP, we take functions whose range is the carrier of a signed cancellation meadow instead of a field as probability measures, include probabilistic choice operators for the probabilities 0 and 1, and have an additional axiom concerning the inclusion of these operators (axiom pA6).

3.1 pACP

In pACP, it is assumed that a fixed but arbitrary set A of *actions*, with $\delta \notin \mathsf{A}$, has been given. We write A_δ for $\mathsf{A} \cup \{\delta\}$. Related to this, it is assumed that a fixed but arbitrary commutative and associative *communication* function $\gamma : \mathsf{A}_\delta \times \mathsf{A}_\delta \to \mathsf{A}_\delta$, with $\gamma(\delta, a) = \delta$ for all $a \in \mathsf{A}_\delta$, has been given. The function γ is regarded to give the result of synchronously performing any two actions for which this is possible, and to give δ otherwise.

It is also assumed that a fixed but arbitrary signed cancellation meadow $\mathfrak{M}$ has been given. We denote the interpretations of the constants and operators of signed cancellation meadows in $\mathfrak{M}$ by the constants and operators themselves. We write $\mathcal{P}$ for the set $\{\pi \in \mathfrak{M} \mid \mathsf{s}(\mathsf{s}(\pi)+1) \cdot \mathsf{s}(\mathsf{s}(1-\pi)+1) = 1\}$ of *probabilities*.

The signature of pACP consists of the following constants and operators:

- for each $a \in \mathsf{A}$, the *action* constant a ;
- the *inaction* constant δ ;
- the binary *alternative composition* operator $+$;
- the binary *sequential composition* operator $\cdot$;
- for each $\pi \in \mathcal{P}$, the binary *probabilistic choice* operator $\boxplus_\pi$;
- the binary *parallel composition* operator $\parallel$;
- the binary *left merge* operator $\mathbin{\|\!\!\lfloor}$;
- the binary *communication merge* operator $\mid$;
- for each $H \subseteq \mathsf{A}$, the unary *encapsulation* operator ∂_H .

We assume that there is a countably infinite set $\mathcal{X}$ of variables, which contains x, y and z, with and without subscripts. Terms are built as usual. We use infix notation for the binary operators. The precedence conventions used with respect to the operators of pACP are as follows: $+$ binds weaker than all others, $\cdot$ binds stronger than all others, and the remaining operators bind equally strong.

The constants and operators of pACP can be explained as follows:

- the constant a denotes the process that can only perform action a and after that terminate successfully;
- the constant δ denotes the process that cannot do anything;

– a closed term of the form $t + t'$ denotes the process that can behave as the process denoted by t or as the process denoted by t', where the choice between the two is resolved exactly when the first action of one of them is performed;
– a closed term of the form $t \cdot t'$ denotes the process that can first behave as the process denoted by t and can next behave as the process denoted by t';
– a closed term of the form $t \mathbin{+\!\!\!+}_\pi t'$ denotes the process that will behave as the process denoted by t with probability π and as the process denoted by t' with probability $1-\pi$, where the choice between the two processes is resolved before the first action of one of them is performed;
– a closed term of the form $t \parallel t'$ denotes the process that can behave as the process that proceeds with the processes denoted by t and t' in parallel;[1]
– a closed term of the form $t \mathbin{\|\!\_} t'$ denotes the process that can behave the same as the process denoted by $t \parallel t'$, except that it starts with performing an action of the process denoted by t;
– a closed term of the form $t \mid t'$ denotes the process that can behave the same as the process denoted by $t \parallel t'$, except that it starts with performing an action of the process denoted by t and an action of the process denoted by t' synchronously;
– a closed term of the form $\partial_H(t)$ denotes the process that can behave the same as the process denoted by t, except that actions from H are blocked.

The operators $\mathbin{\|\!\_}$ and $\mid$ are of an auxiliary nature. They are needed to axiomatize pACP.

The axioms of pACP are the equations given in Table 3. In these equations, a and b stand for arbitrary constants of pACP, H stands for an arbitrary subset of A, and π and ρ stand for arbitrary probabilities from $\mathcal{P}$. Moreover, $\gamma(a, b)$ stands for the action constant for the action $\gamma(a, b)$. In D1 and D2, side conditions restrict what a and H stand for.

The equations in Table 3 above the dotted lines, with A3$'$ replaced by the equation $x + x = x$ and CM1$'$ replaced by its consequent, constitute an axiomatization of ACP. In presentations of ACP, $\gamma(a, b)$ is regularly replaced by $a \mid b$ in CM5–CM7. By CM12, which is more often called CF, these replacements give rise to an equivalent axiomatization. Moreover, CM10 and CM11 are usually absent. These equations are not derivable from the other axioms, but all their closed substitution instances are derivable from the other axioms and they hold in all models that are known to have been devised for ACP.

With regard to axiom CM1$'$, we remark that, for each closed term t of pACP that is not derivably equal to a term of the form $t' \mathbin{+\!\!\!+}_\pi t''$ with $\pi \in \mathcal{P} \setminus \{0, 1\}$, $t = t + t$ is derivable. In other words, if the process denoted by t is not initially probabilistic in nature, then $t = t + t$ is derivable.

pACP has pA1–pA5, pCM1–pCM2, and pD in common with pACP$_\tau$ as presented in [5]. Axioms pCM3–pCM6 are among the axioms of a variant of this probabilistic version of ACP, in which action constants have been replaced by

[1] Parallel composition of two processes is basically interpreted as arbitrary interleaving. With it, probabilistic choices are resolved before interleaving steps are enacted.

Table 3. Axioms of pACP

$x + y = y + x$		A1	$x = x + x \land y = y + y \Rightarrow$	
$(x + y) + z = x + (y + z)$		A2	$\quad x \parallel y = x \mathbin{\|\!\!\lfloor} y + y \mathbin{\|\!\!\lfloor} x + x \mid y$	CM1′
$a + a = a$		A3′	$a \mathbin{\|\!\!\lfloor} x = a \cdot x$	CM2
$(x + y) \cdot z = x \cdot z + y \cdot z$		A4	$a \cdot x \mathbin{\|\!\!\lfloor} y = a \cdot (x \parallel y)$	CM3
$(x \cdot y) \cdot z = x \cdot (y \cdot z)$		A5	$(x + y) \mathbin{\|\!\!\lfloor} z = x \mathbin{\|\!\!\lfloor} z + y \mathbin{\|\!\!\lfloor} z$	CM4
$x + \delta = x$		A6	$a \cdot x \mid b = \gamma(a, b) \cdot x$	CM5
$\delta \cdot x = \delta$		A7	$a \mid b \cdot x = \gamma(a, b) \cdot x$	CM6
			$a \cdot x \mid b \cdot y = \gamma(a, b) \cdot (x \parallel y)$	CM7
			$(x + y) \mid z = x \mid z + y \mid z$	CM8
$\partial_H(a) = a$	if $a \notin H$	D1	$x \mid (y + z) = x \mid y + x \mid z$	CM9
$\partial_H(a) = \delta$	if $a \in H$	D2	$\delta \mid x = \delta$	CM10
$\partial_H(x + y) = \partial_H(x) + \partial_H(y)$		D3	$x \mid \delta = \delta$	CM11
$\partial_H(x \cdot y) = \partial_H(x) \cdot \partial_H(y)$		D4	$a \mid b = \gamma(a, b)$	CM12
....			....	
$x \sqcup\!\!\!\!+_{\pi} y = y \sqcup\!\!\!\!+_{1-\pi} x$		pA1	$(x \sqcup\!\!\!\!+_{\pi} y) \parallel z = (x \parallel z) \sqcup\!\!\!\!+_{\pi} (y \parallel z)$	pCM1
$x \sqcup\!\!\!\!+_{\pi} (y \sqcup\!\!\!\!+_{\rho} z) =$			$x \parallel (y \sqcup\!\!\!\!+_{\pi} z) = (x \parallel y) \sqcup\!\!\!\!+_{\pi} (x \parallel z)$	pCM2
$\quad (x \sqcup\!\!\!\!+_{\frac{\pi}{\pi+\rho-\pi\cdot\rho}} y) \sqcup\!\!\!\!+_{\pi+\rho-\pi\cdot\rho} z)$		pA2	$(x \sqcup\!\!\!\!+_{\pi} y) \mathbin{\|\!\!\lfloor} z = (x \mathbin{\|\!\!\lfloor} z) \sqcup\!\!\!\!+_{\pi} (y \mathbin{\|\!\!\lfloor} z)$	pCM3
$x \sqcup\!\!\!\!+_{\pi} x = x$		pA3	$x \mathbin{\|\!\!\lfloor} (y \sqcup\!\!\!\!+_{\pi} z) = (x \mathbin{\|\!\!\lfloor} y) \sqcup\!\!\!\!+_{\pi} (x \mathbin{\|\!\!\lfloor} z)$	pCM4
$(x \sqcup\!\!\!\!+_{\pi} y) \cdot z = x \cdot z \sqcup\!\!\!\!+_{\pi} y \cdot z$		pA4	$(x \sqcup\!\!\!\!+_{\pi} y) \mid z = (x \mid z) \sqcup\!\!\!\!+_{\pi} (y \mid z)$	pCM5
$(x \sqcup\!\!\!\!+_{\pi} y) + z = (x + z) \sqcup\!\!\!\!+_{\pi} (y + z)$		pA5	$x \mid (y \sqcup\!\!\!\!+_{\pi} z) = (x \mid y) \sqcup\!\!\!\!+_{\pi} (x \mid z)$	pCM6
$x \sqcup\!\!\!\!+_{1} y = x$		pA6	$\partial_H(x \sqcup\!\!\!\!+_{\pi} y) = \partial_H(x) \sqcup\!\!\!\!+_{\pi} \partial_H(y)$	pD

action prefixing operators and a constant for the process that is only capable of terminating successfully, presented in [20]. Therefore, we believe that axioms pCM3–pCM6 are absent in [5] by mistake.

Axiom pA6 is new. Notice that $x \sqcup\!\!\!\!+_{0} (y \sqcup\!\!\!\!+_{0} z) = z$ and $(x \sqcup\!\!\!\!+_{0} y) \sqcup\!\!\!\!+_{0} z = z$ are derivable from pA1 and pA6. This is consistent with the instance of pA2 where $\pi = \rho = 0$ because in meadows $0/0 = 0$.

The replacement of pA2 by $(x \sqcup\!\!\!\!+_{\pi} y) \sqcup\!\!\!\!+_{\rho} z = x \sqcup\!\!\!\!+_{\pi\cdot\rho} (y \sqcup\!\!\!\!+_{\frac{(1-\pi)\cdot\rho}{(1-\pi\cdot\rho)}} z)$ gives rise to an equivalent axiomatization.

In the sequel, we will use the notation $\sum_{i=1}^{n} t_i$, where $n \geq 1$, for right-nested alternative compositions. For each $n \in \mathbb{N}_1$, the term $\sum_{i=1}^{n} t_i$ is defined by induction on n as follows:[2]

$$\textstyle\sum_{i=1}^{1} t_i = t_1 \quad \text{and} \quad \sum_{i=1}^{n+1} t_i = t_1 + \sum_{i=1}^{n} t_{i+1}\ .$$

In addition, we will use the convention that $\sum_{i=1}^{0} t_i = \delta$.

[2] We write $\mathbb{N}_1$ for the set $\{n \in \mathbb{N} \mid n \geq 1\}$ of positive natural numbers.

In the sequel, we will also use the notation $\biguplus_{i=1}^{n} [\pi_i]\, t_i$ where $n \geq 1$ and $\sum_{i<n} \pi_i = 1$, for right-nested probabilistic choices. For each $n \in \mathbb{N}_1$, the term $\biguplus_{i=1}^{n} [\pi_i]\, t_i$ is defined by induction on n as follows:

$$\biguplus_{i=1}^{1} [\pi_i]\, t_i = t_1 \quad \text{and} \quad \biguplus_{i=1}^{n+1} [\pi_i]\, t_i = t_1 \uplus_{\pi_1} (\biguplus_{i=1}^{n} [\tfrac{\pi_{i+1}}{1-\pi_1}]\, t_{i+1})\ .$$

The process denoted by $\biguplus_{i=1}^{n+1} [\pi_i]\, t_i$ will behave like the process denoted by t_1 with probability $\pi_1, \ldots,$ like the process denoted by t_n with probability π_n, and like the process denoted by t_{n+1} with probability $1 - \sum_{i=1}^{n} \pi_i$.

3.2 Guarded Recursion

A closed pACP term denotes a process with a finite upper bound to the number of actions that it can perform. Guarded recursion allows the description of processes without a finite upper bound to the number of actions that it can perform.

Sections 3.2 and 3.3 apply to both pACP and its extension pACP+pSI introduced in Section 4. Therefore, in Sections 3.2 and 3.3, let *PPA* be pACP or pACP+pSI.

Let t be a *PPA* term containing a variable X. Then an occurrence of X in t is *guarded* if t has a subterm of the form $a \cdot t'$ where $a \in \mathsf{A}$ and t' is a *PPA* term containing this occurrence of X. A *PPA* term t is a *guarded PPA* term if all occurrences of variables in t are guarded.

A *guarded recursive specification* over *PPA* is a set $\{X_i = t_i \mid i \in I\}$, where I is finite or countably infinite set, each X_i is a variable, each t_i is either a guarded *PPA* term in which variables other than the variables from $\{X_i \mid i \in I\}$ do not occur or a *PPA* term rewritable to such a term using the axioms of *PPA* in either direction and/or the equations in $\{X_j = t_j \mid j \in I \wedge i \neq j\}$ from left to right, and $X_i \neq X_j$ for all $i, j \in I$ with $i \neq j$.

We write $\mathrm{V}(E)$, where E is a guarded recursive specification, for the set of all variables that occur in E. The equations occurring in a guarded recursive specification are called *recursion equations*.

A solution of a guarded recursive specification E in some model of *PPA* is a set $\{P_X \mid X \in \mathrm{V}(E)\}$ of elements of the carrier of that model such that the equations of E hold if, for all $X \in \mathrm{V}(E)$, X is assigned P_X. We are only interested in models of *PPA* in which guarded recursive specifications have unique solutions.

We extend *PPA* with guarded recursion by adding constants for solutions of guarded recursive specifications over *PPA* and axioms concerning these additional constants. For each guarded recursive specification E over *PPA* and each $X \in \mathrm{V}(E)$, we add a constant standing for the unique solution of E for X to the constants of *PPA*. The constant standing for the unique solution of E for X is denoted by $\langle X|E\rangle$. We use the following notation. Let t be a *PPA* term and E be a guarded recursive specification over *PPA*. Then we write $\langle t|E\rangle$ for t with, for all $X \in \mathrm{V}(E)$, all occurrences of X in t replaced by $\langle X|E\rangle$. We add the equation RDP and the conditional equation RSP given in Table 4 to the axioms of *PPA*. In RDP and RSP, X stands for an arbitrary variable from $\mathcal{X}$, t stands

Table 4. Axioms for guarded recursion

$\langle X\|E\rangle = \langle t\|E\rangle$	if $X = t \in E$	RDP
$E \Rightarrow X = \langle X\|E\rangle$	if $X \in \mathrm{V}(E)$	RSP

for an arbitrary *PPA* term, and E stands for an arbitrary guarded recursive specification over *PPA*. Side conditions restrict what X, t and E stand for. We write PPA_{rec} for the resulting theory.

The equations $\langle X|E\rangle = \langle t|E\rangle$ for a fixed E express that the constants $\langle X|E\rangle$ make up a solution of E. The conditional equations $E \Rightarrow X = \langle X|E\rangle$ express that this solution is the only one.

In extensions of pACP whose axioms include RSP, we have to deal with conditional equational formulas with a countably infinite number of premises. Therefore, infinitary conditional equational logic is used in deriving equations from the axioms of extensions of pACP whose axioms include RSP. A complete inference system for infinitary conditional equational logic can be found in, for example, [21]. It is noteworthy that derivations are allowed to be of countably infinite length in infinitary conditional equational logic.

3.3 Results about Guarded Terms and Guarded Recursion

This section is concerned with guarded terms of a special form and with legitimate ways of manipulating guarded recursive specifications.

The set *HNF* of *head normal forms of PPA* is inductively defined by the following rules:

- $\delta \in HNF$;
- if $a \in \mathsf{A}$, then $a \in HNF$;
- if $a \in \mathsf{A}$ and t is a *PPA* term, then $a \cdot t \in HNF$;
- if $t, t' \in HNF$, then $t + t' \in HNF$;
- if $t, t' \in HNF$ and $\pi \in \mathcal{P}$, then $t \sqcup\!\!\sqcup_\pi t' \in HNF$.

Each head normal form of *PPA* is derivably equal to a head normal form of the form $\bigsqcup_{i=1}^{n} [\pi_i]\, s_i$, where $n \in \mathbb{N}_1$ and, for each $i \in \mathbb{N}_1$ with $i \leq n$, s_i is of the form $\sum_{j=1}^{n_i} a_{ij} \cdot t_{ij} + \sum_{k=1}^{m_i} b_{ik}$, where $n_i, m_i \in \mathbb{N}_1$ and, for all $j \in \mathbb{N}_1$ with $j \leq n_i$, $a_{ij} \in \mathsf{A}$ and t_{ij} is a *PPA* term, and, for all $k \in \mathbb{N}_1$ with $k \leq m_i$, $b_{ik} \in \mathsf{A}$.

Each guarded pACP term is derivably equal to a head normal form of pACP.

Proposition 1. *For each guarded* pACP *term t, there exists a head normal form t' of* pACP *such that $t = t'$ is derivable from the axioms of* pACP*.*

Proof. The proof is straightforward by induction on the structure of t. The case where t is of the form δ and the case where t is of the form a ($a \in \mathsf{A}$) are trivial. The case where t is of the form $t_1 \cdot t_2$ follows immediately from the induction hypothesis and the claim that, for all head normal forms t_1 and t_2 of pACP, there exists a head normal form t' of pACP such that $t_1 \cdot t_2 = t'$ is derivable from the axioms of pACP. This claim is easily proved by induction on the structure of t_1. The cases where t is of the form $t_1 + t_2$ or $t_1 \sqcup\!\!\!+_\pi t_2$ follow immediately from the induction hypothesis. The cases where t is of one of the forms $t_1 \mathbin{|\!\!\lfloor} t_2$, $t_1 \mid t_2$ or $\partial_H(t_1)$ are proved along the same line as the case where t is of the form $t_1 \cdot t_2$. In the case that t is of the form $t_1 \mid t_2$, each of the cases to be considered in the inductive proof of the claim demands a proof by induction on the structure of t_2. The case that t is of the form $t_1 \parallel t_2$ follows immediately from the case that t is of the form $t_1 \mathbin{|\!\!\lfloor} t_2$ and the case that t is of the form $t_1 \mid t_2$. Because t is a guarded pACP term, the case where t is a variable cannot occur. □

Each guarded recursive specification over *PPA* can be manipulated in several ways that are justified by RDP and RSP.

Proposition 2. *For all guarded recursive specifications E over PPA for all $X \in \mathrm{V}(E)$:*

(1) if $Y = t_Y \in E$ and $t_Y = t'_Y$ is derivable from the axioms of PPA then $\langle X|E\rangle = \langle X|(E \setminus \{Y = t_Y\}) \cup \{Y = t'_Y\}\rangle$ is derivable from the axioms of PPA, RDP*, and* RSP*;*
(2) if $Y = t_Y \in E$, $Z = t_Z \in E$, and t'_Y is t_Y with some occurrence of Z in t_Y replaced by t_Z, then $\langle X|E\rangle = \langle X|(E \setminus \{Y = t_Y\}) \cup \{Y = t'_Y\}\rangle$ is derivable from the axioms of PPA, RDP*, and* RSP*;*
(3) if $Y \notin \mathrm{V}(E)$ and t_Y is a guarded PPA term in which variables other than the variables from $\mathrm{V}(E)$ do not occur, then $\langle X|E\rangle = \langle X|E \cup \{Y = t_Y\}\rangle$ is derivable from the axioms of PPA, RDP*, and* RSP*.*

Proof. In case (1), first we apply RDP for each recursion equation in E, next we apply $t_Y = t'_Y$ to $\langle Y|E\rangle = \langle t_Y|E\rangle$, and finally we apply RSP to the resulting set of equations. In case (2), first we apply RDP for each recursion equation in E, next we apply $\langle Z|E\rangle = \langle t_Z|E\rangle$ to $\langle Y|E\rangle = \langle t_Y|E\rangle$, and finally we apply RSP to the resulting set of equations. In case (3), we first apply RDP for each recursion equation in $E \cup \{Y = t_Y\}$ and then we apply RSP to the resulting set of equations.[3] □

Proposition 2 will be used in the proof of Theorem 2 in Section 4.2.

3.4 Semantics of pACP with Guarded Recursion

In this section, we present a structural operational semantics of $\mathrm{pACP_{rec}}$ and define a notion of bisimulation equivalence based on this semantics.

We start with the presentation of a structural operational semantics of $\mathrm{pACP_{rec}}$. The following relations on closed $\mathrm{pACP_{rec}}$ terms are used:

– for each $a \in \mathsf{A}$, a unary relation $\xrightarrow{a}\surd$;

[3] Further details on cases (1) and (2) can be found in the proof of Theorem 4.3.2 from [18].

– for each $a \in \mathsf{A}$, a binary relation $\xrightarrow{a}$;
– for each $\pi \in \mathcal{P}$, a binary relation $\xmapsto{\pi}$.

We write $t \xrightarrow{a} \surd$ instead of $\xrightarrow{a}\surd(t)$, $t \xrightarrow{a} t'$ instead of $\xrightarrow{a}(t,t')$, and $t \xmapsto{\pi} t'$ instead of $\xmapsto{\pi}(t,t')$. Moreover, $t \overset{(0,1]}{\not\mapsto} t'$ abbreviates $\{\neg\ t \xmapsto{\pi} t' \mid \pi \in \mathcal{P} \setminus \{0\}\}$. The relations $\xrightarrow{a}\surd$, $\xrightarrow{a}$, and $\xmapsto{\pi}$ can be explained as follows:

– $t \xrightarrow{a}\surd$: t can perform action a and then terminate successfully;
– $t \xrightarrow{a} t'$: t can perform action a and then behave as t';
– $t \xmapsto{\pi} t'$: t will behave as t' with probability π;

The structural operational semantics of $\mathrm{pACP_{rec}}$ is described by the rules given in Tables 5 and 6. In these tables, a and b stand for arbitrary actions from A, π, ρ, and ρ' stand for arbitrary probabilities from $\mathcal{P}$, X stands for an arbitrary variable from $\mathcal{X}$, t stands for an arbitrary pACP term, and E stands for an arbitrary guarded recursive specification over pACP.

Because of the negative premises in one of the rules for the operational semantics concerning the relations $\xmapsto{\pi}$, the following important result is not self-evident.

Proposition 3. *For each $\pi \in \mathcal{P}$, the relation $\xmapsto{\pi}$ is well-defined by the rules given in Table 6.*

Proof. This is easy to prove by induction on the structure of t. □

Notice that, if t is not derivably equal to a term whose outermost operator is a probabilistic choice operator, then t can only behave as itself and consequently we have that $t \xmapsto{1} t$ and $t \xmapsto{0} t'$ for each term t' other than t.

We define a probability distribution function P from the set of all pairs of closed $\mathrm{pACP_{rec}}$ terms to $\mathcal{P}$ as follows:

$$P(t,t') = \sum_{\pi \in \Pi(t,t')} \pi\ , \quad \text{where}\ \ \Pi(t,t') = \{\pi \mid t \xmapsto{\pi} t'\}\ .$$

This function can be explained as follows: $P(t,t')$ is the total probability that t will behave as t'.

The relations used in an operational semantics are often called transition relations. It is questionable whether the relations $\xmapsto{\pi}$ deserve this name. Recall that $t \xmapsto{\pi} t'$ means that t will behave as t' with probability π. It is rather far-fetched to suppose that a transition from t to t' has taken place at the time that t starts to behave as t'. The relations $\xmapsto{\pi}$ primarily constitute a representation of the probability distribution function P defined above. This representation turns out to be a convenient one in the setting of structural operational semantics.

We write $P(t,T)$, where t is a closed $\mathrm{pACP_{rec}}$ term and T is a set of closed $\mathrm{pACP_{rec}}$ terms, for $\sum_{t' \in T} P(t,t')$.

We write $[t]_R$, where t is a closed $\mathrm{pACP_{rec}}$ term and R is an equivalence relation on closed $\mathrm{pACP_{rec}}$ terms, for the equivalence class of t with respect to R.

Table 5. Rules for the operational semantics of $\mathrm{pACP_{rec}}$ (part 1)

$$\frac{}{a \xrightarrow{a} \surd}$$

$$\frac{x \xrightarrow{a} \surd,\ y \overset{1}{\longmapsto} y'}{x + y \xrightarrow{a} \surd} \qquad \frac{x \overset{1}{\longmapsto} x',\ y \xrightarrow{a} \surd}{x + y \xrightarrow{a} \surd} \qquad \frac{x \xrightarrow{a} x',\ y \overset{1}{\longmapsto} y'}{x + y \xrightarrow{a} x'} \qquad \frac{x \overset{1}{\longmapsto} x',\ y \xrightarrow{a} y'}{x + y \xrightarrow{a} y'}$$

$$\frac{x \xrightarrow{a} \surd}{x \cdot y \xrightarrow{a} y} \qquad \frac{x \xrightarrow{a} x'}{x \cdot y \xrightarrow{a} x' \cdot y}$$

$$\frac{x \xrightarrow{a} \surd,\ y \overset{1}{\longmapsto} y'}{x \parallel y \xrightarrow{a} y} \qquad \frac{x \overset{1}{\longmapsto} x',\ y \xrightarrow{a} \surd}{x \parallel y \xrightarrow{a} x} \qquad \frac{x \xrightarrow{a} x',\ y \overset{1}{\longmapsto} y'}{x \parallel y \xrightarrow{a} x' \parallel y} \qquad \frac{x \overset{1}{\longmapsto} x',\ y \xrightarrow{a} y'}{x \parallel y \xrightarrow{a} x \parallel y'}$$

$$\frac{x \xrightarrow{a} \surd,\ y \xrightarrow{b} \surd}{x \parallel y \xrightarrow{\gamma(a,b)} \surd}\ \gamma(a,b) \in \mathsf{A} \qquad \frac{x \xrightarrow{a} \surd,\ y \xrightarrow{b} y'}{x \parallel y \xrightarrow{\gamma(a,b)} y'}\ \gamma(a,b) \in \mathsf{A}$$

$$\frac{x \xrightarrow{a} x',\ y \xrightarrow{b} \surd}{x \parallel y \xrightarrow{\gamma(a,b)} x'}\ \gamma(a,b) \in \mathsf{A} \qquad \frac{x \xrightarrow{a} x',\ y \xrightarrow{b} y'}{x \parallel y \xrightarrow{\gamma(a,b)} x' \parallel y'}\ \gamma(a,b) \in \mathsf{A}$$

$$\frac{x \xrightarrow{a} \surd}{x \mathbin{\|\!\!\lfloor} y \xrightarrow{a} y} \qquad \frac{x \xrightarrow{a} x'}{x \mathbin{\|\!\!\lfloor} y \xrightarrow{a} x' \parallel y}$$

$$\frac{x \xrightarrow{a} \surd,\ y \xrightarrow{b} \surd}{x \mid y \xrightarrow{\gamma(a,b)} \surd}\ \gamma(a,b) \in \mathsf{A} \qquad \frac{x \xrightarrow{a} \surd,\ y \xrightarrow{b} y'}{x \mid y \xrightarrow{\gamma(a,b)} y'}\ \gamma(a,b) \in \mathsf{A}$$

$$\frac{x \xrightarrow{a} x',\ y \xrightarrow{b} \surd}{x \mid y \xrightarrow{\gamma(a,b)} x'}\ \gamma(a,b) \in \mathsf{A} \qquad \frac{x \xrightarrow{a} x',\ y \xrightarrow{b} y'}{x \mid y \xrightarrow{\gamma(a,b)} x' \parallel y'}\ \gamma(a,b) \in \mathsf{A}$$

$$\frac{x \xrightarrow{a} \surd}{\partial_H(x) \xrightarrow{a} \surd}\ a \notin H \qquad \frac{x \xrightarrow{a} x'}{\partial_H(x) \xrightarrow{a} \partial_H(x')}\ a \notin H$$

$$\frac{\langle t|E\rangle \xrightarrow{a} \surd}{\langle X|E\rangle \xrightarrow{a} \surd}\ X = t\ \in\ E \qquad \frac{\langle t|E\rangle \xrightarrow{a} x'}{\langle X|E\rangle \xrightarrow{a} x'}\ X = t\ \in\ E$$

A *probabilistic bisimulation* is an equivalence relation R on closed $\mathrm{pACP_{rec}}$ terms such that, for all closed $\mathrm{pACP_{rec}}$ terms t_1, t_2 with $R(t_1, t_2)$, the following conditions hold:

- if $t_1 \xrightarrow{a} t_1'$ for some closed $\mathrm{pACP_{rec}}$ term t_1' and $a \in \mathsf{A}$, then there exists a closed $\mathrm{pACP_{rec}}$ term t_2' such that $t_2 \xrightarrow{a} t_2'$ and $R(t_1', t_2')$;
- if $t_1 \xrightarrow{a} \surd$ for some $a \in \mathsf{A}$, then $t_2 \xrightarrow{a} \surd$;
- $P(t_1, [t]_R) = P(t_2, [t]_R)$ for all closed $\mathrm{pACP_{rec}}$ terms t.

Table 6. Rules for the operational semantics of $\mathrm{pACP_{rec}}$ (part 2)

$$\frac{}{a \overset{1}{\longmapsto} a} \qquad \frac{}{\delta \overset{1}{\longmapsto} \delta}$$

$$\frac{x \overset{\pi}{\longmapsto} x',\ y \overset{\rho}{\longmapsto} y'}{x + y \overset{\pi\cdot\rho}{\longmapsto} x' + y'} \qquad \frac{x \overset{\pi}{\longmapsto} x'}{x \cdot y \overset{\pi}{\longmapsto} x' \cdot y} \qquad \frac{x \overset{\rho}{\longmapsto} z,\ y \overset{\rho'}{\longmapsto} z}{x \mathbin{+\!\!\!+}_{\pi} y \overset{\pi\cdot\rho+(1-\pi)\cdot\rho'}{\longmapsto} z}$$

$$\frac{x \overset{\pi}{\longmapsto} x',\ y \overset{\rho}{\longmapsto} y'}{x \parallel y \overset{\pi\cdot\rho}{\longmapsto} x' \parallel y'} \qquad \frac{x \overset{\pi}{\longmapsto} x',\ y \overset{\rho}{\longmapsto} y'}{x \mathbin{\lfloor\!\lfloor} y \overset{\pi\cdot\rho}{\longmapsto} x' \mathbin{\lfloor\!\lfloor} y'} \qquad \frac{x \overset{\pi}{\longmapsto} x',\ y \overset{\rho}{\longmapsto} y'}{x \mid y \overset{\pi\cdot\rho}{\longmapsto} x' \mid y'}$$

$$\frac{x \overset{\pi}{\longmapsto} x'}{\partial_H(x) \overset{\pi}{\longmapsto} \partial_H(x')} \qquad \frac{\langle t|E\rangle \overset{\pi}{\longmapsto} z}{\langle X|E\rangle \overset{\pi}{\longmapsto} z}\ X = t \ \in\ E$$

$$\frac{x \overset{(0,1]}{\not\longmapsto} x'}{x \overset{0}{\longmapsto} x'}$$

Two closed $\mathrm{pACP_{rec}}$ terms t_1, t_2 are *probabilistic bisimulation equivalent*, written $t_1 \mathrel{\underline{\leftrightarrow}} t_2$, if there exists a probabilistic bisimulation R such that $R(t_1, t_2)$. Let R be a probabilistic bisimulation such that $R(t_1, t_2)$. Then we say that R is a probabilistic bisimulation *witnessing* $t_1 \mathrel{\underline{\leftrightarrow}} t_2$.

The next three propositions state some useful results about $\underline{\leftrightarrow}$.

Proposition 4. *For all closed* $\mathrm{pACP_{rec}}$ *terms* t, $t \mathrel{\underline{\leftrightarrow}} t + t$ *only if* $t \overset{1}{\longmapsto} t$.

Proof. This follows immediately from the rules for the operational semantics of $\mathrm{pACP_{rec}}$. □

Proposition 5. $\underline{\leftrightarrow}$ *is the maximal probabilistic bisimulation.*

Proof. It follows from the definition of $\underline{\leftrightarrow}$ that it is sufficient to prove that $\underline{\leftrightarrow}$ is a probabilistic bisimulation.

We start with proving that $\underline{\leftrightarrow}$ is an equivalence relation. The proofs of reflexivity and symmetry are trivial. Proving transitivity amounts to showing that the conditions from the definition of a probabilistic bisimulation hold for the composition of two probabilistic bisimulations. The proofs that the conditions concerning the relations $\xrightarrow{a}$ and $\xrightarrow{a}\surd$ hold are trivial. The proof that the condition concerning the function P holds is also easy using the following easy-to-check property of P: if I is a finite or countably infinite set and, for each $i \in I$, T_i is a set of closed $\mathrm{pACP_{rec}}$ terms such that, for all $i, j \in I$ with $i \neq j$, $T_i \cap T_j = \emptyset$, then $P(t, \bigcup_{i \in I} T_i) = \sum_{i \in I} P(t, T_i)$.

We also have to prove that the conditions from the definition of a probabilistic bisimulation hold for $\underline{\leftrightarrow}$. The proofs that the conditions concerning the relations $\xrightarrow{a}$ and $\xrightarrow{a}\surd$ hold are trivial. The proof that the condition concerning the function P holds is easy knowing the above-mentioned property of P. □

Henceforth, we write R^{e}, where R is a binary relation, for the equivalence closure of R.

Proposition 6. $\leftrightarroweq$ *is a congruence with respect to the operators of* pACP$_{\mathrm{rec}}$.

Proof. In this proof, we write $R_1 \diamond R_2$, where R_1 and R_2 are probabilistic bisimulations and $\diamond$ is a binary operator of pACP$_{\mathrm{rec}}$, for the equivalence relation $\{(t_1 \diamond t_2, t'_1 \diamond t'_2) \mid R_1(t_1, t'_1) \land R_2(t_2, t'_2)\}$.

Let t_1, t'_1, t_2, t'_2 be closed pACP$_{\mathrm{rec}}$ terms such that $t_1 \leftrightarroweq t'_1$ and $t_2 \leftrightarroweq t'_2$, and let R_1 and R_2 be probabilistic bisimulations witnessing $t_1 \leftrightarroweq t'_1$ and $t_2 \leftrightarroweq t'_2$, respectively.

For each binary operator $\diamond$ of pACP$_{\mathrm{rec}}$, we construct an equivalence relation $R_\diamond$ on closed pACP$_{\mathrm{rec}}$ terms as follows:

in the case that $\diamond$ is $\cdot$:	$R_\diamond = ((R_1 \diamond R_2) \cup R_2)^{\mathsf{e}}$;
in the case that $\diamond$ is $+, \uplus_\pi$ or $\parallel$:	$R_\diamond = ((R_1 \diamond R_2) \cup R_1 \cup R_2)^{\mathsf{e}}$;
in the case that $\diamond$ is $\mathbin{\lfloor\!\lfloor}$ or $\mid$:	$R_\diamond = ((R_1 \diamond R_2) \cup (R_1 \parallel R_2) \cup R_1 \cup R_2)^{\mathsf{e}}$

and for each encapsulation operator ∂_H, we construct an equivalence relation R_{∂_H} on closed pACP$_{\mathrm{rec}}$ terms as follows:

$$R_{\partial_H} = (\{(\partial_H(t_1), \partial_H(t'_1)) \mid R_1(t_1, t'_1)\} \cup R_1)^{\mathsf{e}} .$$

For each operator $\diamond$ of pACP$_{\mathrm{rec}}$, we have to show that the conditions from the definition of a probabilistic bisimulation hold for the constructed relation $R_\diamond$.

Notwithstanding the additional premise $x \overset{1}{\mapsto} x'$ or $y \overset{1}{\mapsto} y'$ in some of the rules for the operational semantics concerning the relations $\xrightarrow{a}$ and $\xrightarrow{a}\surd$, the proofs that the conditions concerning the relations $\xrightarrow{a}$ and $\xrightarrow{a}\surd$ hold are similar to the proofs in the case of ACP$_{\mathrm{rec}}$. The proof that the condition concerning the function P holds is straightforward using the property of P mentioned in the proof of Proposition 5 and the following easy-to-check properties of P:

$$\begin{aligned}
&P(t \cdot t', T \cdot T') &&= 0 && \text{if } t' \notin T' , \\
&P(t \cdot t', T \cdot T') &&= P(t, T) && \text{if } t' \in T' , \\
&P(t + t', T + T') &&= P(t, T) \cdot P(t', T') , \\
&P(t \uplus_\pi t', T) &&= \pi \cdot P(t, T) + (1 - \pi) \cdot P(t', T) , \\
&P(t \parallel t', T \parallel T') &&= P(t, T) \cdot P(t', T') , \\
&P(t \mathbin{\lfloor\!\lfloor} t', T \mathbin{\lfloor\!\lfloor} T') &&= P(t, T) \cdot P(t', T') , \\
&P(t \mid t', T \mid T') &&= P(t, T) \cdot P(t', T') , \\
&P(\partial_H(t), \partial_H(T)) &&= P(t, T) ,
\end{aligned}$$

where we write $T \diamond T'$, where T and T' are sets of closed pACP$_{\mathrm{rec}}$ terms and $\diamond$ is a binary operator of pACP$_{\mathrm{rec}}$, for the set $\{t \diamond t' \mid t \in T \land t' \in T'\}$ and we write $\partial_H(T)$, where T is a set of closed pACP$_{\mathrm{rec}}$ terms, for the set $\{\partial_H(t) \mid t \in T\}$. $\square$

pACP$^+$ is the variant of pACP with a different parallel composition operator that is presented in [3,4].[4] A detailed proof of Proposition 6 is to a large extent a simplified version of the detailed proof of the fact that $\leftrightarrow$ is a congruence with respect to the operators of pACP$^+$ that is given in [4]. This is because of the fact that, except for the parallel composition operator, the structural operational semantics of pACP presented in this paper can essentially be obtained from the structural operational semantics of pACP$^+$ that is presented in [4] by removing unnecessary complexity.

For a frequently occurring kind of bisimulation equivalence, constraints have been proposed on the form of operational semantics rules which ensure that bisimulation equivalence is a congruence (see e.g. [1]). Probabilistic bisimulation equivalence as defined in this paper is unfortunately not of the kind of bisimulation equivalence concerned.

pACP$_{\mathrm{rec}}$ is sound with respect to probabilistic bisimulation equivalence for equations between closed terms.

Theorem 1 (Soundness). *For all closed* pACP$_{\mathrm{rec}}$ *terms t and t', $t = t'$ is derivable from the axioms of* pACP$_{\mathrm{rec}}$ *only if $t \leftrightarrow t'$.*

Proof. Since $\leftrightarrow$ is a congruence for pACP$_{\mathrm{rec}}$, we only need to verify the soundness of each axiom of pACP$_{\mathrm{rec}}$.

For each equational axiom e of pACP$_{\mathrm{rec}}$ (all axioms of pACP$_{\mathrm{rec}}$ except CM1$'$ and RSP are equational), we construct an equivalence relation R_e on closed pACP$_{\mathrm{rec}}$ terms as follows:

$$R_e = \{(t, t') \mid t = t' \text{ is a closed substitution instance of } e\}^{\mathsf{e}} \;.$$

For axiom CM1$'$, we construct an equivalence relation R' on closed pACP$_{\mathrm{rec}}$ terms as follows:

$$R' = \{(t, t') \mid t = t' \text{ is a closed substitution instance of } e \land t \overset{1}{\mapsto} t \land t' \overset{1}{\mapsto} t'\}^{\mathsf{e}} \;,$$

where e is the consequent of CM1$'$.

For an arbitrary instance $\{X_i = t_i \mid i \in I\} \;\Rightarrow\; X = \langle X | \{X_i = t_i \mid i \in I\}\rangle$ of RSP, we construct an equivalence relation R'' on closed pACP$_{\mathrm{rec}}$ terms as follows:

$$R'' = \{(\theta(X), \langle X | \{X_i = t_i \mid i \in I\}\rangle) \mid \theta \in \Theta \land \textstyle\bigwedge_{i \in I} \theta(X_i) \leftrightarrow \theta(t_i)\}^{\mathsf{e}} \;,$$

where Θ is the set of all functions from $\mathcal{X}$ to the set of all closed pACP$_{\mathrm{rec}}$ terms and $\theta(t)$, where $\theta \in \Theta$ and t is a pACP$_{\mathrm{rec}}$ term, stands for t with, for all $X \in \mathcal{X}$, all occurrences of X replaced by $\theta(X)$.

For each equational axiom e of pACP$_{\mathrm{rec}}$, we have to show that the conditions from the definition of a probabilistic bisimulation hold for the constructed relation R_e. For axiom CM1$'$, we have to show that the conditions from the definition of a probabilistic bisimulation hold for the constructed relation R'. That

[4] pACP$^+$ is called ACP$_\pi^+$ in [3].

this is sufficient for the soundness of axiom CM1′ follows from Proposition 4. For the instances of axiom RSP, we have to show that the conditions from the definition of a probabilistic bisimulation hold for the constructed relation R''.

Notwithstanding the additional premise $x \overset{1}{\longmapsto} x'$ or $y \overset{1}{\longmapsto} y'$ in some of the rules for the operational semantics concerning the relations $\xrightarrow{a}$ and $\xrightarrow{a}\surd$, the proofs that the conditions concerning the relations $\xrightarrow{a}$ and $\xrightarrow{a}\surd$ hold are similar to the proofs in the case of $\mathrm{ACP_{rec}}$. The proof that the condition concerning the function P holds is straightforward using the following easy-to-check property of P: if β is a bijection on T and $P(t', t) = P(t'', \beta(t))$ for all $t \in T$, then $P(t', T) = P(t'', T)$. □

pACP is a minor variant of the subtheory of pACP_τ from [5] in which the operators for abstraction from certain actions are lacking. Soundness and completeness results with respect to branching bisimulation equivalence of an unspecified operational semantics of pACP_τ are claimed in [5]. However, the operational semantics concerned can be reconstructed from the operational semantics of pTCP_τ given in [20].[5] It turns out that a mistake has been made in the rules for the probabilistic choice operators that concern the relations $\overset{\pi}{\longmapsto}$. The mistake concerned manifests only in closed terms of the form $t \mathbin{+\!\!\!+}_{1/2} t$. For example, if t is not derivably equal to a term whose outermost operator is a probabilistic choice operator, then both the left-hand side and the right-hand side of $t \mathbin{+\!\!\!+}_{1/2} t$ give rise to $t \mathbin{+\!\!\!+}_{1/2} t \overset{1/2}{\longmapsto} t$. Consequently, the total probability that $t \mathbin{+\!\!\!+}_{1/2} t$ behaves as t is $1/2$ instead of 1. This is counterintuitive and inconsistent with axiom pA3.

A meadow has a total multiplicative inverse operation where the multiplicative inverse of zero is zero. This is why there is no reason to exclude the probabilistic choice operators $\mathbin{+\!\!\!+}_\pi$ for $\pi \in \{0, 1\}$ if a meadow is used instead of a field. Because we have included these operators, we also have included relations $\overset{\pi}{\longmapsto}$ for $\pi \in \{0, 1\}$. As a bonus of the inclusion of these relations, we could achieve that for all pairs (t, t') of closed $\mathrm{pACP_{rec}}$ terms, there exists a $\pi \in \mathcal{P}$ such that $t \overset{\pi}{\longmapsto} t'$. Due to this, we could at the same time reduce the number of rules for the operational semantics that concern the relations $\overset{\pi}{\longmapsto}$, replace all negative premises by positive premises in rules for the operational semantics that concern the relations $\xrightarrow{a}$ and $\xrightarrow{a}\surd$, and correct the above-mentioned mistake in the rules for the probabilistic choice operators that concern the relations $\overset{\pi}{\longmapsto}$.

Above, we already mentioned that a variant of pACP, called pACP^+, is presented in [3,4]. pACP, just like pACP_τ from [5], differs from pACP^+ with respect to the parallel composition operator. Moreover, in [3,4], the probability distribution function is defined directly instead of via the structural operational semantics. However, except for parallel composition and left merge, the probability distribution function corresponds to the probability distribution function P defined above. The direct definition of the probability distribution function removes the root of the above-mentioned mistake made in [20].

[5] In pTCP_τ, action constants have been replaced by action prefixing operators and a constant for the process that is only capable of terminating successfully.

4 Probabilistic Strategic Interleaving

In this section, we extend pACP with probabilistic strategic interleaving, i.e. interleaving according to some probabilistic interleaving strategy. Interleaving strategies are known as process-scheduling policies in the field of operating systems. A well-known probabilistic process-scheduling policy is lottery scheduling [27]. In the presented extension of pACP deterministic interleaving strategies are special cases of probabilistic interleaving strategies: they are the ones obtained by restriction to the trivial probabilities 0 and 1.

4.1 pACP with Probabilistic Strategic Interleaving

In the extension of pACP with probabilistic strategic interleaving presented below, it is expected that an interleaving strategy uses the interleaving history in one way or another to make process-scheduling decisions.

The sets $\mathcal{H}_n$ of *interleaving histories for n processes*, for $n \in \mathbb{N}_1$, are the subsets of $(\mathbb{N}_1 \times \mathbb{N}_1)^*$ that are inductively defined by the following rules:

- $\langle\,\rangle \in \mathcal{H}_n$;
- if $i \leq n$, then $(i, n) \in \mathcal{H}_n$;
- if $h \frown (i, n) \in \mathcal{H}_n$, $j \leq n$, and $n{-}1 \leq m \leq n{+}1$, then $h \frown (i, n) \frown (j, m) \in \mathcal{H}_m$.[6]

The intuition concerning interleaving histories is as follows: if the kth pair of an interleaving history is (i, n), then the ith process got a turn in the kth interleaving step and after its turn there were n processes to be interleaved. The number of processes to be interleaved may increase due to process creation (introduced below) and decrease due to successful termination of processes.

The presented extension of pACP is called pACP+pSI (pACP with probabilistic Strategic Interleaving). It covers a generic probabilistic interleaving strategy that can be instantiated with different specific probabilistic interleaving strategies that can be represented in the way that is explained below.

In pACP+pSI, it is assumed that the following has been given:

- a fixed but arbitrary set S;
- for each $n \in \mathbb{N}_1$, a fixed but arbitrary function $\sigma_n{:}\mathcal{H}_n \times S \to (\{1, \ldots, n\} \to \mathcal{P})$ satisfying, for each $h \in \mathcal{H}$ and $s \in S$, $\sum_{i=1}^{n} \sigma_n(h, s)(i) = 1$;
- for each $n \in \mathbb{N}_1$, a fixed but arbitrary function $\vartheta_n : \mathcal{H}_n \times S \times \{1, \ldots, n\} \times \mathsf{A} \to S$.

The elements of S are called *control states*, σ_n is called an *abstract scheduler* (*for n processes*), and ϑ_n is called a *control state transformer* (*for n processes*). The intuition concerning S, σ_n, and ϑ_n is as follows:

- the control states from S encode data that are relevant to the interleaving strategy, but not derivable from the interleaving history;

[6] We write $\langle\,\rangle$ for the empty sequence, d for the sequence having d as sole element, and $\alpha \frown \alpha'$ for the concatenation of sequences α and α'. We assume that the usual identities, such as $\langle\,\rangle \frown \alpha = \alpha$ and $(\alpha \frown \alpha') \frown \alpha'' = \alpha \frown (\alpha' \frown \alpha'')$, hold.

– $\sigma_n(h, s)$ is the probability distribution on n processes that assigns to each of the processes the probability that it gets the next turn after interleaving history h in control state s;
– $\vartheta_n(h, s, i, a)$ is the control state that arises from the ith process doing a after interleaving history h in control state s.

Thus, S, $\langle \sigma_n \rangle_{n \in \mathbb{N}_1}$, and $\langle \vartheta_n \rangle_{n \in \mathbb{N}_1}$ make up a way to represent an interleaving strategy. This way to represent an interleaving strategy is engrafted on [26].

Consider the case where S is a singleton set, for each $n \in \mathbb{N}_1$, σ_n is defined by

$$\begin{aligned}
&\sigma_n(\langle\,\rangle, s)(i) = 1 && \text{if } i = 1\ , \\
&\sigma_n(\langle\,\rangle, s)(i) = 0 && \text{if } i \neq 1\ , \\
&\sigma_n(h \frown (j, n), s)(i) = 1 && \text{if } i = (j + 1) \bmod n\ , \\
&\sigma_n(h \frown (j, n), s)(i) = 0 && \text{if } i \neq (j + 1) \bmod n
\end{aligned}$$

and, for each $n \in \mathbb{N}_1$, ϑ_n is defined by

$$\vartheta_n(h, s, i, a) = s\ .$$

In this case, the interleaving strategy corresponds to the round-robin scheduling algorithm. This deterministic interleaving strategy is called cyclic interleaving in our work on interleaving strategies in the setting of thread algebra (see e.g. [13]). In the current setting, an interleaving strategy is deterministic if, for all $n \in \mathbb{N}_1$, for all $h \in \mathcal{H}_n$, $s \in S$, and $i \in \{1, \dots, n\}$, $\sigma_n(h, s)(i) \in \{0, 1\}$. In the case that S and ϑ_n are as above, but σ_n is defined by

$$\sigma_n(h, s)(i) = 1/n\ ,$$

the interleaving strategy is a purely probabilistic one. The probability distribution used is a uniform distribution.

More advanced strategies can be obtained if the scheduling makes more advanced use of the interleaving history and the control state. The interleaving history may, for example, be used to factor the individual lifetimes of the processes to be interleaved and their creation hierarchy into the process-scheduling decision making. Individual properties of the processes to be interleaved that depend on the actions performed by them can be taken into account by making use of the control state. The control state may, for example, be used to factor the processes being interleaved that currently wait to acquire a lock from a process that manages a shared resource into the process-scheduling decision making.[7]

In pACP+pSI, it is also assumed that a fixed but arbitrary set D of *data* and a fixed but arbitrary function $\phi : D \to P$, where P is the set of all closed terms over the signature of pACP+pSI (given below), have been given and that,

[7] In [13], various examples of interleaving strategies are given in the setting of thread algebra. The representation of the more serious of these examples in the current setting demands nontrivial use of the control state.

for each $d \in D$ and $a, b \in \mathsf{A}$, $\mathsf{cr}(d), \overline{\mathsf{cr}}(d) \in \mathsf{A}$, $\gamma(\mathsf{cr}(d), a) = \delta$, and $\gamma(a, b) \neq \mathsf{cr}(d)$. The action $\mathsf{cr}(d)$ can be considered a process creation request and the action $\overline{\mathsf{cr}}(d)$ can be considered a process creation act. They represent the request to start the process denoted by $\phi(d)$ in parallel with the requesting process and the act of carrying out that request, respectively.

The signature of pACP+pSI consists of the constants and operators from the signature of pACP and in addition the following operators:

- for each $n \in \mathbb{N}_1$, $h \in \mathcal{H}$, and $s \in S$, the n-ary *strategic interleaving* operator $\|^n_{h,s}$;
- for each $n, i \in \mathbb{N}_1$ with $i \leq n$, $h \in \mathcal{H}$, and $s \in S$, the n-ary *positional strategic interleaving* operator $\lfloor\!\lfloor^{n,i}_{h,s}$.

The strategic interleaving operators can be explained as follows:

- a closed term of the form $\|^n_{h,s}(t_1, \ldots, t_n)$ denotes the process that results from interleaving of the n processes denoted by $t_1, \ldots, t_n$ after interleaving history h in control state s, according to the interleaving strategy represented by S, $\langle \sigma_n \rangle_{n \in \mathbb{N}_1}$, and $\langle \vartheta_n \rangle_{n \in \mathbb{N}_1}$.

The positional strategic interleaving operators are auxiliary operators used to axiomatize the strategic interleaving operators. The role of the positional strategic interleaving operators in the axiomatization is similar to the role of the left merge operator found in pACP.

The axioms of pACP+pSI are the axioms of pACP and in addition the equations given in Table 7. In the additional equations, n and i stand for arbitrary numbers from $\mathbb{N}_1$, h stands for an arbitrary interleaving history from $\mathcal{H}$, s stands for an arbitrary control state from S, a stands for an arbitrary action constant that is not of the form $\mathsf{cr}(d)$ or $\overline{\mathsf{cr}}(d)$, and d stands for an arbitrary datum d from D.

The equations in Table 7 above the dotted line, with SI1$'$ replaced by its consequent, are essentially the axioms for strategic interleaving presented in [14] for the deterministic case. The only difference is in the consequent of SI1$'$. This is inevitable because there are no probabilistic choice operators in the deterministic setting of [14].

Axiom SI2 expresses that, in the event of inactiveness of the process whose turn it is, the whole becomes inactive immediately. A plausible alternative is that, in the event of inactiveness of the process whose turn it is, the whole becomes inactive only after all other processes have terminated or become inactive. In that case, the functions $\vartheta_n : \mathcal{H} \times S \times \{1, \ldots, n\} \times \mathsf{A} \to S$ must be extended to functions $\vartheta_n : \mathcal{H} \times S \times \{1, \ldots, n\} \times (\mathsf{A} \cup \{\delta\}) \to S$ and axiom SI2 must be replaced by the axioms in Table 8.

In $(\text{pACP+pSI})_{\text{rec}}$, i.e. pACP+pSI extended with guarded recursion in the way described in Section 3.2, the processes that can be created are restricted to the ones denotable by a closed pACP+pSI term. This restriction stems from the requirement that ϕ is a function from D to the set of all closed pACP+pSI terms. The restriction can be removed by relaxing this requirement to the requirement that ϕ is a function from D to the set of all closed $(\text{pACP+pSI})_{\text{rec}}$

Table 7. Axioms for strategic interleaving

$x_1 = x_1 + x_1 \land \ldots \land x_1 = x_n + x_n \Rightarrow$	
$\quad \|^n_{h,s}(x_1, \ldots, x_n) = \biguplus^n_{i=1} [\sigma_n(h,s)(i)] \; \lfloor\!\lfloor^{n,i}_{h,s}(x_1, \ldots, x_n)$	SI1′
$\lfloor\!\lfloor^{n,i}_{h,s}(x_1, \ldots, x_{i-1}, \delta, x_{i+1}, \ldots, x_n) = \delta$	SI2
$\lfloor\!\lfloor^{1,i}_{h,s}(a) = a$	SI3
$\lfloor\!\lfloor^{n+1,i}_{h,s}(x_1, \ldots, x_{i-1}, a, x_{i+1}, \ldots, x_{n+1}) =$	
$\quad a \cdot \|^n_{h \frown (i,n), \vartheta_{n+1}(h,s,i,a)}(x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_{n+1})$	SI4
$\lfloor\!\lfloor^{n,i}_{h,s}(x_1, \ldots, x_{i-1}, a \cdot x'_i, x_{i+1}, \ldots, x_n) =$	
$\quad a \cdot \|^n_{h \frown (i,n), \vartheta_n(h,s,i,a)}(x_1, \ldots, x_{i-1}, x'_i, x_{i+1}, \ldots, x_n)$	SI5
$\lfloor\!\lfloor^{n,i}_{h,s}(x_1, \ldots, x_{i-1}, \mathsf{cr}(d), x_{i+1}, \ldots, x_n) =$	
$\quad \overline{\mathsf{cr}}(d) \cdot \|^n_{h \frown (i,n), \vartheta_n(h,s,i,\mathsf{cr}(d))}(x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n, \phi(d))$	SI6
$\lfloor\!\lfloor^{n,i}_{h,s}(x_1, \ldots, x_{i-1}, \mathsf{cr}(d) \cdot x'_i, x_{i+1}, \ldots, x_n) =$	
$\quad \overline{\mathsf{cr}}(d) \cdot \|^{n+1}_{h \frown (i,n+1), \vartheta_n(h,s,i,\mathsf{cr}(d))}(x_1, \ldots, x_{i-1}, x'_i, x_{i+1}, \ldots, x_n, \phi(d))$	SI7
$\lfloor\!\lfloor^{n,i}_{h,s}(x_1, \ldots, x_{i-1}, x'_i + x''_i, x_{i+1}, \ldots, x_n) =$	
$\quad \lfloor\!\lfloor^{n,i}_{h,s}(x_1, \ldots, x_{i-1}, x'_i, x_{i+1}, \ldots, x_n) + \lfloor\!\lfloor^{n,i}_{h,s}(x_1, \ldots, x_{i-1}, x''_i, x_{i+1}, \ldots, x_n)$	SI8
$\|^n_{h,s}(x_1, \ldots, x_{i-1}, x'_i \uplus_\pi x''_i, x_{i+1}, \ldots, x_n) =$	
$\quad \|^n_{h,s}(x_1, \ldots, x_{i-1}, x'_i, x_{i+1}, \ldots, x_n) \uplus_\pi \|^n_{h,s}(x_1, \ldots, x_{i-1}, x''_i, x_{i+1}, \ldots, x_n)$	pSI1
$\lfloor\!\lfloor^{n,i}_{h,s}(x_1, \ldots, x_{i-1}, x'_i \uplus_\pi x''_i, x_{i+1}, \ldots, x_n) =$	
$\quad \lfloor\!\lfloor^{n,i}_{h,s}(x_1, \ldots, x_{i-1}, x'_i, x_{i+1}, \ldots, x_n) \uplus_\pi \lfloor\!\lfloor^{n,i}_{h,s}(x_1, \ldots, x_{i-1}, x''_i, x_{i+1}, \ldots, x_n)$	pSI2

Table 8. Alternative axioms for SI2

$\lfloor\!\lfloor^{1,i}_{h,s}(\delta) = \delta$	SI2a
$\lfloor\!\lfloor^{n+1,i}_{h,s}(x_1, \ldots, x_{i-1}, \delta, x_{i+1}, \ldots, x_{n+1}) =$	
$\quad \|^n_{h \frown (i,n), \vartheta_{n+1}(h,s,i,\delta)}(x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_{n+1}) \cdot \delta$	SI2b

terms. We write $(\text{pACP+pSI})^+_{\text{rec}}$ for the theory resulting from this relaxation. In other words, $(\text{pACP+pSI})^+_{\text{rec}}$ differs from $(\text{pACP+pSI})_{\text{rec}}$ in that it is assumed that a fixed but arbitrary function $\phi : D \to P$, where P is the set of all closed terms over the signature of $(\text{pACP+pSI})_{\text{rec}}$, has been given.

4.2 Results about Guarded Terms and Guarded Recursion

This section is concerned with guarded terms in head normal form and with the reduction of guarded recursive specifications over pACP+pSI to guarded recursive specification over pACP.

Each guarded pACP+pSI term is derivably equal to a head normal form of pACP+pSI.

Proposition 7. *For each guarded* pACP+pSI *term t, there exists a head normal form t' of* pACP+pSI *such that $t = t'$ is derivable from the axioms of* pACP+pSI*.*

Proof. The proof is straightforward by induction on the structure of t. The cases other than $\|^{n}_{h,s}(t_1,\ldots,t_n)$ and $\rfloor\!\lfloor^{n,i}_{h,s}(t_1,\ldots,t_n)$ are proved along the same line as in the proof of Proposition 1. The case where t is of the form $\rfloor\!\lfloor^{n,i}_{h,s}(t_1,\ldots,t_n)$ is proved along the same line as the case where t is of the form $t_1 \cdot t_2$, but the claim is of course proved by induction on the structure of t_i instead of t_1. The case that t is of the form $\|^{n}_{h,s}(t_1,\ldots,t_n)$ follows immediately from the case that t is of the form $\rfloor\!\lfloor^{n,i}_{h,s}(t_1,\ldots,t_n)$. Because t is a guarded pACP+pSI term, the case where t is a variable cannot occur. □

The following theorem refers to three process algebras. It is implicit that the same set A of actions and the same communication function γ are assumed in the process algebras referred to.

Each guarded recursive specification over pACP+pSI can be reduced to a guarded recursive specification over pACP.

Theorem 2 (Reduction). *For each guarded recursive specification E over* pACP+pSI *and each $X \in \mathrm{V}(E)$, there exists a guarded recursive specification E' over* pACP *such that $\langle X|E\rangle = \langle X|E'\rangle$ is derivable from the axioms of* $(\text{pACP+pSI})_{\text{rec}}$*.*

Proof. Let E be a guarded recursive specification over pACP+pSI. Assume that, for each equation $X = t$ from E, t is a guarded pACP+pSI term. It follows from Proposition 2 that this assumption does not lead to loss of generality.

Let $X = t_X$ be an equation from E. Now, by Proposition 7, there exists an $n \in \mathbb{N}_1$ such that, for each $i \in \mathbb{N}_1$ with $i \leq n$, there exists a $\pi_i \in \mathcal{P}$ and $n_i, m_i \in \mathbb{N}_1$ such that, for each $j \in \mathbb{N}_1$ with $j \leq n_i$ and $k \in \mathbb{N}_1$ with $k \leq m_i$, there exist an $a_{ij} \in \mathsf{A}$, a pACP+pSI term t_{ij}, and a $b_{ik} \in \mathsf{A}$ such that $t_X = \bigsqcup^{n}_{i=1} [\pi_i]\,(\sum^{n_i}_{j=1} a_{ij} \cdot t_{ij} + \sum^{m_i}_{k=1} b_{ik})$ is derivable from the axioms of pACP+pSI. For each $i \in \mathbb{N}_1$ with $i \leq n$, for each $j \in \mathbb{N}_1$ with $j \leq n_i$, let t'_{ij} be t_{ij} with, for each equation $Y = t_Y$ from E, each unguarded occurrence of Y in t_{ij} replaced by the guarded pACP+pSI term t_Y. For each $i \in \mathbb{N}_1$ with $i \leq n$, for each $j \in \mathbb{N}_1$ with $j \leq n_i$, by its construction, the term t'_{ij} is a guarded pACP+pSI term in which variables other than the ones from $\mathrm{V}(E)$ do not occur. Now, by Proposition 2 (a) for each $i \in \mathbb{N}_1$ with $i \leq n$, for each $j \in \mathbb{N}_1$ with $j \leq n_i$, the equation $X_{ij} = t'_{ij}$, where X_{ij} is a fresh variable, can be added to E and (b) the equation $X = t_X$ can be replaced by the equation $X = \bigsqcup^{n}_{i=1} [\pi_i]\,(\sum^{n_i}_{j=1} a_{ij} \cdot X_{ij} + \sum^{m_i}_{k=1} b_{ik})$ in E. The other equations from E can be replaced by a set of equations in the same way as the equation $X = t_X$.

The set of equations so obtained can be manipulated following the same procedure as in the case of E, but the manipulation can be restricted to the added equations. Repeating this procedure, perhaps countably infinitely many times, we obtain a guarded recursive specification E' over pACP for which $\langle X|E\rangle = \langle X|E'\rangle$ is derivable from the axioms of $(\text{pACP+pSI})_{\text{rec}}$. □

Table 9. Additional rules for the operational semantics of (pACP+pSI)$^{+}_{\mathrm{rec}}$

$$\frac{x \xrightarrow{a} \surd}{\rfloor\!\rfloor^{1,1}_{h,s}(x) \xrightarrow{a} \surd}$$

$$\frac{x_1 \overset{1}{\longmapsto} x'_1,\ \ldots,\ x_{i-1} \overset{1}{\longmapsto} x'_{i-1},\ x_i \xrightarrow{a} \surd,\ x_{i+1} \overset{1}{\longmapsto} x'_{i+1},\ \ldots,\ x_{n+1} \overset{1}{\longmapsto} x'_{n+1}}{\rfloor\!\rfloor^{n+1,i}_{h,s}(x_1,\ldots,x_{n+1}) \xrightarrow{a} \|^{n}_{h \frown (i,n),\vartheta_{n+1}(h,s,i,a)}(x_1,\ldots,x_{i-1},x_{i+1},\ldots,x_{n+1})}$$

$$\frac{x_1 \overset{1}{\longmapsto} x'_1,\ \ldots,\ x_{i-1} \overset{1}{\longmapsto} x'_{i-1},\ x_i \xrightarrow{a} x'_i,\ x_{i+1} \overset{1}{\longmapsto} x'_{i+1},\ \ldots,\ x_n \overset{1}{\longmapsto} x'_n}{\rfloor\!\rfloor^{n,i}_{h,s}(x_1,\ldots,x_n) \xrightarrow{a} \|^{n}_{h \frown (i,n),\vartheta_n(h,s,i,a)}(x_1,\ldots,x_{i-1},x'_i,x_{i+1},\ldots,x_n)}$$

$$\frac{x_1 \overset{1}{\longmapsto} x'_1,\ \ldots,\ x_{i-1} \overset{1}{\longmapsto} x'_{i-1},\ x_i \xrightarrow{\mathsf{cr}(d)} \surd,\ x_{i+1} \overset{1}{\longmapsto} x'_{i+1},\ \ldots,\ x_n \overset{1}{\longmapsto} x'_n}{\rfloor\!\rfloor^{n,i}_{h,s}(x_1,\ldots,x_n) \xrightarrow{\overline{\mathsf{cr}}(d)} \|^{n}_{h \frown (i,n),\vartheta_n(h,s,i,\mathsf{cr}(d))}(x_1,\ldots,x_{i-1},x_{i+1},\ldots,x_n,\phi(d))}$$

$$\frac{x_1 \overset{1}{\longmapsto} x'_1,\ \ldots,\ x_{i-1} \overset{1}{\longmapsto} x'_{i-1},\ x_i \xrightarrow{\mathsf{cr}(d)} x'_i,\ x_{i+1} \overset{1}{\longmapsto} x'_{i+1},\ \ldots,\ x_n \overset{1}{\longmapsto} x'_n}{\rfloor\!\rfloor^{n,i}_{h,s}(x_1,\ldots,x_n) \xrightarrow{\overline{\mathsf{cr}}(d)} \|^{n+1}_{h \frown (i,n+1),\vartheta_n(h,s,i,\mathsf{cr}(d))}(x_1,\ldots,x_{i-1},x'_i,x_{i+1},\ldots,x_n,\phi(d))}$$

$$\frac{x_1 \overset{\pi_1}{\longmapsto} x'_1,\ \ldots,\ x_n \overset{\pi_n}{\longmapsto} x'_n}{\|^{n}_{h,s}(x_1,\ldots,x_n) \overset{\sigma_n(h,s)(i)\cdot\pi_1\cdot\ldots\cdot\pi_n}{\longmapsto} \rfloor\!\rfloor^{n,i}_{h,s}(x'_1,\ldots,x'_n)}$$

$$\frac{x_1 \overset{\pi_1}{\longmapsto} x'_1,\ \ldots,\ x_n \overset{\pi_n}{\longmapsto} x'_n}{\rfloor\!\rfloor^{n,i}_{h,s}(x_1,\ldots,x_n) \overset{\pi_1\cdot\ldots\cdot\pi_n}{\longmapsto} \rfloor\!\rfloor^{n,i}_{h,s}(x'_1,\ldots,x'_n)}$$

4.3 Semantics of pACP+pSI with Guarded Recursion

In this section, we present a structural operational semantics of (pACP+pSI)$^{+}_{\mathrm{rec}}$.

The structural operational semantics of (pACP+pSI)$^{+}_{\mathrm{rec}}$ is described by the rules for the operational semantics of pACP$_{\mathrm{rec}}$ (given in Tables 5 and 6) and in addition the rules given in Table 9. In the additional rules, n and i stand for arbitrary numbers from $\mathbb{N}_1$, h stands for an arbitrary interleaving history from $\mathcal{H}$, s stands for an arbitrary control state from S, a stands for an arbitrary action from A that is not of the form $\mathsf{cr}(d)$ or $\overline{\mathsf{cr}}(d)$, d stands for an arbitrary datum d from D, and $\pi_1, \ldots, \pi_n$ stand for arbitrary probabilities from $\mathcal{P}$.

Proposition 8. $\leftrightarrow$ *is a congruence w.r.t. the operators of* (pACP+pSI)$^{+}_{\mathrm{rec}}$.

Proof. The proof goes along the same line as the proof of Proposition 6 □

(pACP+pSI)$^{+}_{\mathrm{rec}}$ is sound with respect to probabilistic bisimulation equivalence for equations between closed terms.

Theorem 3 (Soundness). *For all closed* (pACP+pSI)$^{+}_{\mathrm{rec}}$ *terms* t *and* t', $t = t'$ *is derivable from the axioms of* (pACP+pSI)$^{+}_{\mathrm{rec}}$ *only if* $t \leftrightarrow t'$.

Proof. The proof goes along the same line as the proof of Theorem 1. □

5 Concluding Remarks

We have presented a probabilistic version of ACP [12,9] that rests on the principle that probabilistic choices are resolved before choices involved in alternative composition and parallel composition are resolved. By taking functions whose range is the carrier of a signed cancellation meadow [17,11] instead of a field as probability measures, we could include probabilistic choice operators for the probabilities 0 and 1 without any problem and give a simple operational semantics. We have also extended this probabilistic version of ACP with a form of interleaving in which parallel processes are interleaved according to what is known as a process-scheduling policy in the field of operating systems. This is the form of interleaving that underlies multi-threading as found in contemporary programming languages.

References

1. Aceto, L., Fokkink, W.J., Verhoef, C.: Structural operational semantics. In: Bergstra, J.A., Ponse, A., Smolka, S.A. (eds.) Handbook of Process Algebra, pp. 197–292. Elsevier, Amsterdam (2001)
2. America, P., de Bakker, J.W.: Designing equivalent semantic models for process creation. Theoretical Computer Science 60(2), 109–176 (1988)
3. Andova, S.: Process algebra with probabilistic choice. In: Katoen, J.P. (ed.) ARTS'99. Lecture Notes in Computer Science, vol. 1601, pp. 111–129. Springer-Verlag (1999)
4. Andova, S.: Probabilistic Process Algebra. Ph.D. thesis, Department of Mathematics and Computer Science, Eindhoven University of Technology, Eindhoven (2002)
5. Andova, S., Georgievska, S.: On compositionality, efficiency, and applicability of abstraction in probabilistic systems. In: Nielsen, M., et al. (eds.) SOFSEM 2009. Lecture Notes in Computer Science, vol. 5404, pp. 67–78. Springer-Verlag (2009)
6. Baeten, J.C.M., Bergstra, J.A.: Real space process algebra. Formal Aspects of Computing 5(6), 481–529 (1993)
7. Baeten, J.C.M., Middelburg, C.A.: Process Algebra with Timing. Monographs in Theoretical Computer Science, An EATCS Series, Springer-Verlag, Berlin (2002)
8. Baeten, J.C.M., Vaandrager, F.W.: An algebra of process creation. Acta Informatica 29(4), 303–334 (1992)
9. Baeten, J.C.M., Weijland, W.P.: Process Algebra, Cambridge Tracts in Theoretical Computer Science, vol. 18. Cambridge University Press, Cambridge (1990)
10. Bergstra, J.A.: A process creation mechanism in process algebra. In: Baeten, J.C.M. (ed.) Applications of Process Algebra, Cambridge Tracts in Theoretical Computer Science, vol. 17, pp. 81–88. Cambridge University Press, Cambridge (1990)
11. Bergstra, J.A., Bethke, I., Ponse, A.: Cancellation meadows: A generic basis theorem and some applications. Computer Journal 56(1), 3–14 (2013)
12. Bergstra, J.A., Klop, J.W.: The algebra of recursively defined processes and the algebra of regular processes. In: Paredaens, J. (ed.) Proceedings 11th ICALP. Lecture Notes in Computer Science, vol. 172, pp. 82–95. Springer-Verlag (1984)
13. Bergstra, J.A., Middelburg, C.A.: Thread algebra for strategic interleaving. Formal Aspects of Computing 19(4), 445–474 (2007)

14. Bergstra, J.A., Middelburg, C.A.: Process algebra with strategic interleaving. Theory of Computing Systems 63(3), 488–505 (2019)
15. Bergstra, J.A., Middelburg, C.A., Usenko, Y.S.: Discrete time process algebra and the semantics of SDL. In: Bergstra, J.A., Ponse, A., Smolka, S.A. (eds.) Handbook of Process Algebra, pp. 1209–1268. Elsevier, Amsterdam (2001)
16. Bergstra, J.A., Ponse, A.: Probability functions in the context of signed involutive meadows. In: James, P., Roggenbach, M. (eds.) WADT 2016. Lecture Notes in Computer Science, vol. 10644, pp. 73–87. Springer-Verlag (2017)
17. Bergstra, J.A., Tucker, J.V.: The rational numbers as an abstract data type. Journal of the ACM 54(2), Article 7 (2007)
18. Fokkink, W.J.: Introduction to Process Algebra. Texts in Theoretical Computer Science, An EATCS Series, Springer-Verlag, Berlin (2000)
19. Gehrke, T., Rensink, A.: Process creation and full sequential composition in a name-passing calculus. Electronic Notes in Theoretical Computer Science 7, 141–160 (1997)
20. Georgievska, S.: Probability and Hiding in Concurrent Processes. Ph.D. thesis, Department of Mathematics and Computer Science, Eindhoven University of Technology, Eindhoven (2011)
21. van Glabbeek, R.J., Vaandrager, F.W.: Modular specification of process algebras. Theoretical Computer Science 113(2), 293–348 (1993)
22. Gosling, J., Joy, B., Steele, G., Bracha, G.: The Java Language Specification. Addison-Wesley, Reading, MA, second edn. (2000)
23. Hejlsberg, A., Wiltamuth, S., Golde, P.: C# Language Specification. Addison-Wesley, Reading, MA (2003)
24. Hoare, C.A.R.: Communicating Sequential Processes. Prentice-Hall, Englewood Cliffs (1985)
25. Milner, R.: Communication and Concurrency. Prentice-Hall, Englewood Cliffs (1989)
26. Sabelfeld, A., Sands, D.: Probabilistic noninterference for multi-threaded programs. In: Computer Security Foundations Workshop 2000. pp. 200–214. IEEE Computer Society Press (2000)
27. Waldspurger, C.A., Weihl, W.E.: Lottery scheduling: Flexible proportional-share resource management. In: OSDI '94. p. Article 1. USENIX Association, Berkeley (1994)