Axiomatizing probabilistic processes : ACP with generative probabilities

Citation for published version (APA):

Document status and date:
Published: 01/01/1992

Document Version:
Publisher’s PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:
• A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
• The final author version and the galley proof are versions of the publication after peer review.
• The final published version features the final layout of the paper including the volume, issue and page numbers.

Link to publication

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

• Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
• You may not further distribute the material or use it for any profit-making activity or commercial gain
• You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the “Taverne” license above, please follow below link for the End User Agreement:
www.tue.nl/taverne

Take down policy
If you believe that this document breaches copyright please contact us at:
openaccess@tue.nl
providing details and we will investigate your claim.

Download date: 07. Apr. 2021
Axiomatizing Probabilistic Processes: ACP with Generative Probabilities

by

J.C.M. Baeten  J.A. Bergstra  S.A. Smolka

92/19
COMPUTING SCIENCE NOTES

This is a series of notes of the Computing Science Section of the Department of Mathematics and Computing Science Eindhoven University of Technology. Since many of these notes are preliminary versions or may be published elsewhere, they have a limited distribution only and are not for review. Copies of these notes are available from the author.

Copies can be ordered from:
Mrs. F. van Neerven
Eindhoven University of Technology
Department of Mathematics and Computing Science
P.O. Box 513
5600 MB EINDHOVEN
The Netherlands
ISSN 0926-4515

All rights reserved
editors: prof.dr.M.Rem
prof.dr.K.M.van Hee.
Axiomatizing Probabilistic Processes: ACP with Generative Probabilities*

J.C.M. Baeten
Dept. of Math and Computing Science, Eindhoven University of Technology
P.O. Box 513, 5600 MB Eindhoven, The Netherlands

J.A. Bergstra
Programming Research Group, University of Amsterdam
P.O. Box 41882, 1009 DB Amsterdam, The Netherlands

S.A. Smolka
Department of Computer Science, SUNY at Stony Brook
Stony Brook, NY 11794-4400, U.S.A.

Abstract

This paper is concerned with finding complete axiomatizations of probabilistic processes. We examine this problem within the context of the process algebra ACP and obtain as our end-result the axiom system prACP, a probabilistic version of ACP which can be used to reason algebraically about the reliability and performance of concurrent systems. Our goal was to introduce probability into ACP in as simple a fashion as possible. Optimally, ACP should be the homomorphic image of the probabilistic version in which the probabilities are forgotten.

We begin by weakening slightly ACP to obtain the axiom system ACP. The main difference between ACP and ACP is that the axiom $z + \delta = z$, which does not yield a plausible interpretation in the generative model of probabilistic computation, is rejected in ACP. We argue that this does not affect the usefulness of ACP in practice, and show how ACP can be reconstructed from ACP with a minimal amount of technical machinery.

prACP is obtained from ACP through the introduction of probabilistic alternative and parallel composition operators, and a process graph model for prACP based on probabilistic bisimulation is developed. We show that prACP is a sound and complete axiomatization of probabilistic bisimulation for finite processes, and that prACP can be homomorphically embedded in ACP as desired.

Our results for ACP and prACP are presented in a modular fashion by first considering several subsets of the signatures. We conclude with a discussion about the suitability of an internal probabilistic choice operator in the context of prACP.
1 Introduction

It is intriguing to consider the notion of probability (or probabilistic behavior) within the context of process algebra: a formal system of algebraic, equational, and operational techniques for the specification and verification of concurrent systems. Through the introduction of probabilistic measures, one can begin to analyze — in an algebraic fashion — "quantitative" aspects of concurrency such as reliability, performance, and fault tolerance.

In this paper, we address this problem in terms of complete axiomatizations of probabilistic processes within the context of the axiom system ACP [BK84]. ACP models an asynchronous merge, with synchronous communication, by means of arbitrary interleaving. It uses an additional constant $\delta$, which plays the role of $NIL$ from CCS [Mil80] (CCS is a predecessor of ACP). The key axioms for $\delta$ are:

\[
\begin{align*}
& x + \delta = x \quad \text{(A6)} \\
& \delta \cdot x = \delta \quad \text{(A7)}
\end{align*}
\]

The process $\delta$ represents an unfeasible option; i.e., a task that cannot be performed and therefore will be postponed indefinitely. The interaction with merge (parallel composition) is as follows:

\[
x || \delta = x \cdot \delta
\]

(This is not provable from ACP but for each closed process expression $p$ we find that $ACP \vdash p || \delta = p \cdot \delta$. Now $\delta$ represents deadlock according to the explanation of [BK84].

Our goal is to introduce probability into ACP in as simple a fashion as possible. Optimally we would like ACP to be the homomorphic image of the probabilistic version in which the probabilities are forgotten. To this end, we first develop a weaker version of ACP called $ACP_f$. This axiom system is just a minor alteration expressing almost the same process identities on finite processes. The virtues of this weaker axiom system are as follows:

(i) $ACP_f$ does not imply $x + \delta = x$. In fact, this axiom has often been criticized as being non-obvious for the interpretation $\delta=\text{deadlock}=\text{inaction}$.

(ii) $ACP_f + \{x + \delta = x\}$ implies the same identities on finite processes as ACP (but it is slightly weaker on identities between open processes).

(iii) $ACP_f$ has for all practical purposes the same expressiveness as ACP. I.e., if one can specify a protocol in ACP, this can be done just as well in $ACP_f$.

(iv) $ACP_f$ allows a probabilistic interpretation of $\div$, and for this reason we need it as a point of departure for the development of a probabilistic version of ACP.

We introduce probability into $ACP_f$ by replacing the operators for alternative and parallel composition with probabilistic counterparts to obtain the axiom system $prACP_f$. Probabilistic choice in $prACP_f$ is of the generative variety, as defined in [vGSST90], in that a single probability distribution is ascribed to all alternatives. Consequently, choices involving possibly different actions are resolved probabilistically. In contrast, in the reactive model of probabilistic computation [LS89, vGSST90], a separate distribution is associated with each action, and choices involving different actions are resolved nondeterministically.

A property of the generative model of probabilistic computation is that, unlike the reactive model, the probabilities of alternatives are conditional with respect to the set of actions offered by
the environment. A more detailed comparison of the reactive and generative models can be found in [vGSST90]. There the stratified model is also considered and it is shown that the generative model is an abstraction of the stratified model and the reactive model is an abstraction of the generative model.

Previous work on probabilistic process algebra [LS89, GJS90, vGSST90, Chr90, BM89, JL91, CSZ92] has been primarily of an operational/behavioral nature. Three exceptions, however, are [JS90, Tof90, LS92]. In [JS90], a complete axiomatization of generative probabilistic processes built from a limited set of operators (NIL, action prefix, probabilistic alternative composition, and tail recursion) are provided, while in [Tof90], axioms for synchronously composed "weighted processes" are given. A complete axiomatization of an SCCS-like calculus with reactive probabilities is presented in [LS92].

Summary of Technical Results

We have obtained the following results toward our goal of finding complete axiomatizations of probabilistic processes.

- We first present the axiom system $\text{ACPi}_r$, our point of departure from ACP. Its development is modular beginning with BPA (consisting of process constants, alternative composition, and sequential composition), to which we add a merge and left-merge operator to obtain PA. Finally, a communication merge operator, the constant $\delta$, and an auxiliary initials operator $I$ are added to PA to obtain $\text{ACPi}_r$. In each case, we present a process graph model based on bisimulation and prove that the system is a sound and complete axiomatization of bisimulation for finite processes.

- We show in a technical sense how ACP can be reconstructed from $\text{ACPi}_r$ through the reintroduction of the axiom A6.

- The axiom systems $\text{prBPA}$, $\text{prPA}$, and $\text{prACPi}_r$ for probabilistic processes are considered next. In each case, we present a process graph model based on probabilistic bisimulation, Larsen and Skou's [LS89] probabilistic extension of strong bisimulation, and prove that the system is a sound and complete axiomatization of probabilistic bisimulation for finite probabilistic processes.

- Connections between $\text{ACPi}_r$ and its probabilistic counterpart are then explored. We show that $\text{ACPi}_r$ is the homomorphic image of $\text{prACPi}_r$ in which the probabilities are forgotten. This result is obtained for both the graph model — the homomorphism preserves the structure of the bisimulation congruence classes, and the proof theory — the homomorphic image of a valid proof in $\text{prACPi}_r$ is a valid proof in $\text{ACPi}_r$.

- We show that certain technical problems arise when a probabilistic internal choice operator is added to $\text{prACPi}_r$, and argue that a state operator should be introduced to remedy the situation.

The structure of the rest of this paper is as follows. Section 2 presents the equational specifications BPA, PA, and $\text{ACPi}_r$, and their accompanying process graph models and completeness results. Section 3 treats the probabilistic versions of these axiom systems, namely, $\text{prBPA}$, $\text{prPA}$, and $\text{prACPi}_r$. The homomorphic derivability of $\text{ACPi}_r$ from $\text{prACPi}_r$ is the subject of Section 4. Section 5 discusses the suitability of an internal probabilistic choice operator in the context of $\text{prACPi}_r$, and, finally, Section 6 concludes. Note that we do not treat internal or $\tau$-moves in this paper, so we stay within the setting of concrete process algebra.
2 A Weaker Version of ACP

In this section we present the equational theory \( \text{ACP}^- \), which, as described in Section 1, will be our point of departure for a probabilistic version of ACP. The main difference between ACP and \( \text{ACP}^- \) is that the axiom \( z + \delta = z \), which does not yield a plausible interpretation in the generative model of probabilistic computation, is rejected in \( \text{ACP}^- \).

As is the practice in ACP, we begin with the theory BPA (Basic Process Algebra) which describes processes constructed from constants, plus, and sequential composition. We will then add to BPA a notion of parallel composition (merge and left-merge) to obtain PA (Process Algebra). Finally, the theory \( \text{ACP}^-_1(A) \) is derived by extending BPA with the constant \( \delta \) (for deadlock), a combined notion of parallel composition and communication, and a restriction operator.

2.1 BPA

2.1.1 Equational Specification

The signature \( \Sigma(\text{BPA}(A)) \) consists of one sort \( P \) (for processes) and three types of operators: constant processes \( a \), for each atomic action \( a \), the sequential composition (or sequencing) operator '·', and the alternative composition (or nondeterministic choice) operator '+'.

The set of all constants is denoted by \( A \), and is considered a parameter to the theory.

\[
\Sigma(\text{BPA}(A)) = \{a : \rightarrow P | a \in A\} \cup \{+ : P \times P \rightarrow P\} \cup \{\cdot : P \times P \rightarrow P\}
\]

The axiom system BPA\((A)\) is given by:

\[
\begin{align*}
z + y &= y + z & \text{A1} \\
(z + y) + z &= z + (y + z) & \text{A2} \\
z + z &= z & \text{A3} \\
(z + y) \cdot z &= z \cdot (y + z) & \text{A4} \\
(z \cdot y) \cdot z &= z \cdot (y \cdot z) & \text{A5}
\end{align*}
\]

Note the absence of the axiom \( z \cdot (y + z) = z \cdot y + z \cdot z \), which does not hold in our bisimulation model.

2.1.2 Graph Model

We define a process graph model for BPA\((A)\). The underlying notion of equivalence for process graphs is bisimulation, and we prove completeness of BPA\((A)\) in this model. We will later extend our graph model to PA\((A)\) and \( \text{ACP}^-_1(A) \). As before, let \( A \) be the set of atomic actions. We consider process graphs with labels from \( A \).

**Definition 2.1** A process graph \( g \) is a triple \( < V, r, \rightarrow > \) such that

- \( V \) is the set of nodes (vertices) of \( g \)
- \( r \in V \) is the root of \( g \)
• \( \rightarrow \subseteq V \times A \times V \) is the transition relation of \( g \)

The endpoints of \( g \) are those nodes devoid of outgoing transitions representing successful termination. The major role played by endpoints is in the definition, given below, of the sequential composition of two process graphs. We often write \( v \xrightarrow{a} v' \) to denote the fact that \( (v, a, v') \in \rightarrow \). We denote by \( \mathcal{G} \) the family of all process graphs. Bisimulation, due to Milner and Park [Mil80, Par81], is the primary equivalence relation we consider on process graphs.

**Definition 2.2** Let \( g_1 = < V_1, r_1, \rightarrow_1 > \), \( g_2 = < V_2, r_2, \rightarrow_2 > \) be two process graphs. A bisimulation between \( g_1 \) and \( g_2 \) is a relation \( \mathcal{R} \subseteq V_1 \times V_2 \) with the following properties:

1. \( \mathcal{R}(r_1, r_2) \)
2. \( \forall v \in V_1, w \in V_2 \text{ with } \mathcal{R}(v, w): \forall a \in A \text{ and } v' \in V_1, \text{ if } v \xrightarrow{a} v' \text{ then } \exists w' \in V_2 \text{ with } \mathcal{R}(v', w') \text{ and } w \xrightarrow{a} w' \)
3. and vice versa with the roles of \( v \) and \( w \) reversed.

Graphs \( g_1 \) and \( g_2 \) are said to be bisimilar, written \( g_1 \equiv g_2 \), if there exists a bisimulation between \( g_1 \) and \( g_2 \).

We now define the operators from \( \Sigma(BPA(A)) \) on the domain \( \mathcal{F} \) of finite process graphs, i.e., process graphs that are finitely branching and acyclic in their transition relations. Therefore, \( \mathcal{F} \subset \mathcal{G} \). For this purpose, it is convenient to assume that a process graph root-node is not an endpoint. For the remainder of Section 2, unless otherwise stated, let \( g_1 = < V_1, r_1, \rightarrow_1 > \), \( g_2 = < V_2, r_2, \rightarrow_2 > \) be finite process graphs satisfying the non-endpoint root assumption such that \( V_1 \cap V_2 = \emptyset \).

**Definition 2.3** The operators \( a \in A, +, \text{ and } \cdot \) are defined on \( \mathcal{F} \) as follows:

- \( a \in A \): The process graph for each of these constants consists of a single transition and is given by \( < \{ r_a, v \}, r_a, \{ < r_a, a, v > \} > \).

- \( g_1 + g_2 \): is given by \( < V_1 \cup V_2 \cup \{ r \}, r, \rightarrow > \) such that \( r \notin V_1 \cup V_2 \) and \( v \xrightarrow{a} v' \) if one or more of the following holds:
  - \( r_1 \xrightarrow{a} v' \) and \( v = r \)
  - \( r_2 \xrightarrow{a} v' \) and \( v = r \)
  - \( v \xrightarrow{a} v' \)

- \( g_1 \cdot g_2 \): is obtained by appending a copy of \( g_2 \) at each endpoint of \( g_1 \). In detail, \( g_1 \cdot g_2 \) is given by \( < V_1 \times V_2, (r_1, r_2), \rightarrow > \) where \( (g_1, g_2) \xrightarrow{a} (q_1, q_2) \) if one or more of the following holds:
  - \( g_1 \xrightarrow{a} q_1 \) and \( g_2 = q_2 = r_2 \)
  - \( g_2 \xrightarrow{a} q_2 \) and \( g_1 = q_1 \) is an endpoint
For $t$ a closed $\text{BPA}(A)$ term, we write $\text{graph}(t) = \langle V_t, r_t, \rightarrow_t \rangle$ to denote the process graph obtained inductively on $t$ using Definition 2.3. We take the liberty to write expressions like $p \equiv q$, instead of the more precise $\text{graph}(p) \equiv \text{graph}(q)$, when this is clear from the context. The definition of $\text{graph}(t)$ and the just-mentioned notational liberty extend in the obvious way to the axiom systems $\text{PA}(A)$ and $\text{ACP}_\ell(A)$, to be considered later in this section.

In the setting of $\text{BPA}$, $\equiv$ is a congruence (see, e.g., [BW90]).

**Proposition 2.1** If $g_1 \equiv g_2$, then $g + g_1 \equiv g + g_2$, $g \cdot g_1 \equiv g \cdot g_2$, and $g_1 \cdot g \equiv g_2 \cdot g$.

We have that $\mathcal{F}/\equiv$, the graph model, is indeed a model of the axiom system $\text{BPA}(A)$, and that $\text{BPA}(A)$ constitutes a complete axiomatization of process equivalence in $\mathcal{F}/\equiv$.

**Theorem 2.1** ([BW90])

1. $\mathcal{F}/\equiv \models \text{BPA}(A)$
2. For all closed expressions $p, q$ over $\Sigma(\text{BPA}(A))$:
   
   $\mathcal{F}/\equiv \models p = q \implies \text{BPA}(A) \vdash p = q$.

2.2 \text{PA}

2.2.1 Equational Specification

The signature $\Sigma(\text{PA}(A))$ is obtained from $\Sigma(\text{BPA}(A))$ by adding an interleaving merge operator $\|$ and a left-merge operator $\lbrack$:

$$
\Sigma(\text{PA}(A)) = \Sigma(\text{BPA}(A)) \cup \{\| : P \times P \rightarrow P\} \cup \{\lbrack : P \times P \rightarrow P\}
$$

Intuitively, the process $z \| y$ is obtained by interleaving (shuffling) the atomic actions of $z$ and $y$ together. Left-merge is an auxiliary operator in that it permits $\|$ to be specified in finitely many equations. The process $z \lbrack y$ has the same meaning as $z \| y$, but with the restriction that the first step must come from $z$.

The axiom system $\text{PA}(A)$ is given by:

$\text{BPA}(A) +$

- $z \| y = z \lbrack y + y \rbrack z$ \quad M1
- $a \lbrack z = a \cdot z$ \quad M2
- $(a \cdot z) \lbrack y = a \cdot (z \| y)$ \quad M3
- $(x + y) \lbrack z = z \lbrack z + y \rbrack z$ \quad M4

6
2.2.2 Graph Model

The two new operators of \( PA(A) \) are now defined on finite process graphs (as before, with non-endpoint roots).

Definition 2.4 The operators \( \parallel \) and \( \triangleright \) are defined on \( F \) as follows:

\[ g_1 \parallel g_2: \text{is given by } <V_1 \times V_2,(r_1,r_2),\longrightarrow \text{ where } (v_1,v_2) \xrightarrow{a} (v'_1,v'_2) \text{ if either of the following holds:} \]

- \( v_1 \xrightarrow{a_1} v'_1 \) and \( v_2 = v'_2 \)
- \( v_2 \xrightarrow{a_2} v'_2 \) and \( v_1 = v'_1 \)

\[ g_1 \triangleright g_2: \text{As } g_1 \parallel g_2 \text{ but without transitions of the form } (r_1,r_2) \xrightarrow{a} (r_1,v). \]

Again one may notice that \( \equiv \) is a congruence, \( F/\equiv = PA(A) \) and that \( PA(A) \) constitutes a complete axiomatization of process equivalence in \( F/\equiv \) [BW90].

2.3 ACP without A6

2.3.1 Equational Specification

The equational system \( ACP^*_f(A) \) treats the operators of \( BPA(A) \) as well as the new constant \( \delta \) representing deadlock; a communication merge operator \( \parallel \) describing the result of a communication between any two atomic actions; a merge operator \( \parallel \) and left-merge operator \( \triangleright \) like those of \( PA(A) \) but which additionally admit the possibility of communication; and a family of restriction operators \( \partial_H, H \subseteq A \). We will also need an auxiliary operator \( I \) that defines the initial actions that a process can perform.

Letting \( A_\delta = A \cup \{ \delta \} \), the signature of \( ACP^*_f(A) \) extends that of \( PA(A) \) as follows:

\[ \Sigma(ACP^*_f(A)) = \Sigma(PA(A)) \cup \{ \delta \rightarrow P \} \cup \{ | : P \times P \rightarrow P \} \cup \{ \partial_H : P \rightarrow P \mid H \subseteq A \} \cup \{ I : P \rightarrow 2^{A_\delta} \} \]

It is convenient to define communication merge as a binary commutative and associative function on atomic actions (i.e., \( | : A_\delta \times A_\delta \rightarrow A_\delta \) with \( \delta \) acting as a multiplicative zero. This is accomplished with axioms C1-3 below. We further require \( | \) to be total and to capture this axiomatically we need a way to effectively enumerate all the constant processes. For this purpose, we define the characteristic predicate \( A_\delta \) of \( A_\delta \) in the usual way:

\[ \overline{A_\delta}(x) = \bigvee_{a \in A_\delta} (x = a) \]

The totality of \( | \) is now given by the following axiom: axiom:

\[ \forall a, b \in P \ A_\delta(a) \land A_\delta(b) \implies \exists c \in P \ A_\delta(c) \land a \upharpoonright b = c \quad C0 \]

\[ ^1 \text{Axiom C0 is often replaced by choosing a total function } \gamma : A_\delta \times A_\delta \rightarrow A_\delta \text{ and having all identities of the graph of } \gamma \text{ as axioms: } a \upharpoonright b = \gamma(a,b). \text{ In this way, } \gamma \text{ becomes another parameter to the theory (see, e.g., [BW90]).} \]
The axioms of $\text{ACP}_I(A)$ are now given. In this system, $a, b, c$ range over $A_\delta$, $H_\delta = H \cup \{\delta\}$, and $\cap, \cup$ are used on $2^A$, without further specification.

**BPA(A)** +

| A7 | $\delta \cdot z = \delta$ |

**C0** +

| C1 | $a \| b = b \| a$ |
| C2 | $(a \| b) \| c = a \| (b \| c)$ |
| C3 | $\delta \| a = \delta$ |

| CM1 | $x \| y = z \| y + y \| x + z \| y$ |
| CM2 | $a \| z = a \cdot z$ |
| CM3 | $(a \cdot z) \| y = a(z \| y)$ |
| CM4 | $(x + y) \| z = (z \| z) + (y \| z)$ |
| CM5 | $a \| (b \cdot z) = (a \| b) \cdot z$ |
| CM6 | $(a \cdot z) \| b = (a \| b) \cdot z$ |
| CM7 | $(a \cdot z) \| (b \cdot y) = (a \| b) \cdot (z \| y)$ |
| CM8 | $(x + y) \| z = z \| z + y \| z$ |
| CM9 | $x \| (y + z) = z \| y + z \| z$ |

| I1 | $I(a) = \{a\}$ |
| I2 | $I(z \cdot y) = I(z)$ |
| I3 | $I(z + y) = I(z) \cup I(y)$ |

| D1 | $a \in H \Rightarrow \partial_H(a) = \delta$ |
| D2 | $a \notin H \Rightarrow \partial_H(a) = a$ |
| D3.1 | $I(z) \subseteq H_\delta \Rightarrow \partial_H(z + y) = \partial_H(y)$ |
| D3.2 | $I(z + y) \cap H_\delta = \emptyset \Rightarrow \partial_H(z + y) = \partial_H(z) + \partial_H(y)$ |
| D4 | $\partial_H(z \cdot y) = \partial_H(z) \cdot \partial_H(y)$ |
Comments: ACP\(\overline{7}\)\((A)\) differs from ACP by the absence of A6 and the presence of D3.1-2 instead of axiom D3: \(\partial_H(x + y) = \partial_H(x) + \partial_H(y)\). The following examples illustrate the new axiom system.

\[
\begin{align*}
\partial_{\{c\}}(a + (b + c)) &= \partial_{\{c\}}(c + (a + b)) \quad \text{(by A1 and A2)} \\
&= \partial_{\{c\}}(a + b) \quad \text{(by D3.1)} \\
&= \partial_{\{c\}}(a) + \partial_{\{c\}}(b) \quad \text{(by D3.2)} \\
&= a + b \quad \text{(by D2 twice)}
\end{align*}
\]

\[
\begin{align*}
\partial_{\{a\}}(a + \delta) &= \partial_{\{a\}}(\delta + a) \quad \text{(by A1)} \\
&= \partial_{\{a\}}(a) \quad \text{(by D3.1)} \\
&= \delta \quad \text{(by D1)}
\end{align*}
\]

\[
\begin{align*}
\partial_{\{a\}}(a + \delta) &= \partial_{\{a\}}(\delta) \quad \text{(by D3.1)} \\
&= \delta \quad \text{(by D2)}
\end{align*}
\]

2.3.2 Graph Model

Let \(\text{initials}(v) \subseteq A_6\) be the set of actions \(\{a \in A_6 \mid \exists v' \; v \xrightarrow{a} v'\}\) for \(v\) a process graph node. The operators of ACP\(\overline{7}\)\((A)\), beyond those of BPA(A), are now defined on finite process graphs (with non-endpoint roots).

**Definition 2.5** The ACP\(\overline{7}\)\((A)\) operators \(\delta, \|, [\cdot], \partial_H\) (for \(H \subseteq A\)), and \(I\) are defined on \(\mathcal{F}\) as follows:

\(\delta\): is given by \(<\{r_5, v_6\}, r_6, \{<r_6, \delta, v_6>\}>\).

\(g_1 \| g_2\): is given by \(<V_1 \times V_2, (r_1, r_2), \rightarrow>\) where \((v_1, v_2) \xrightarrow{a} (v_1', v_2')\) if one or more of the following holds:

- \(v_1 \xrightarrow{a} v_1'\) and \(v_2 = v_2'\)
- \(v_2 \xrightarrow{a} v_2'\) and \(v_1 = v_1'\)
- \(v_1 \xrightarrow{b} v_1'\), \(v_2 \xrightarrow{c} v_2'\), and \(a = b|c\) (for some \(b\) and \(c\))

\(g_1 \| g_2\): As \(g_1 \| g_2\) but without transitions of the form \((r_1, r_2) \xrightarrow{a} (r_1, v)\).

\(g_1 | g_2\): As \(g_1 \| g_2\) but without transitions of the form \((r_1, r_2) \xrightarrow{a} (r, r_2)\) or \((r_1, r_2) \xrightarrow{a} (r_1, v)\).

\(\partial_H(g_1)\): is given by \(<V_1, r_1, \rightarrow>\) where

\[
\rightarrow = \{(v, a, v') \in \rightarrow_1 \mid a \notin H\} \cup \{(v, \delta, v') \mid (v, a, v') \in \rightarrow_1 \text{ and } \text{initials}(v) \subseteq H\}
\]

\(I(g_1)\): gives the set of actions \(\text{initials}(r_1)\).

Our algebra of process graphs is standard (see, e.g., [BW90]) with the exception of restriction. This operator removes all edges labeled with actions from the set of restricted actions \(H\). It also removes \(\delta\)-edges, which it must do to ensure the soundness of D3.1. In case a node in \(g_1\) qualifies to have all its edges removed, then these edges are not removed but rather renamed into \(\delta\)-transitions.
The presence of δ-transitions, which intuitively represent the capability for a process to deadlock, requires a new definition of bisimulation in which a weaker condition is imposed on δ-transitions.

**Definition 2.6** Let $g_1 = <V_1, r_1, \rightarrow_1>$, $g_2 = <V_2, r_2, \rightarrow_2>$ be two process graphs. A δ-bisimulation between $g_1$ and $g_2$ is a relation $\mathcal{R} \subseteq V_1 \times V_2$ with the following properties:

- $\mathcal{R}(r_1, r_2)$
- $\forall v \in V_1, w \in V_2 \text{ with } \mathcal{R}(v, w)$:
  - $\forall a \in A$ and $v' \in V_1$, if $v \xrightarrow{a} v'$ then $\exists w' \in V_2$ with $\mathcal{R}(v', w')$ and $w \xrightarrow{a} w'$
  - if $v \xrightarrow{\delta} v'$, for some $v'$, then $w \xrightarrow{\delta} w'$, for some $w'$
- and vice versa with the roles of $v$ and $w$ reversed.

Graphs $g_1$ and $g_2$ are δ-bisimilar, written $g_1 \equiv_\delta g_2$, if there exists a δ-bisimulation between $g_1$ and $g_2$.

This definition is the same as Definition 2.2 with the additional stipulation that for two nodes $v, w$ related by a δ-bisimulation, $v$ possesses a δ-edge iff $w$ does. We once again have that $\equiv_\delta$ is a congruence.

**Proposition 2.2** If $g_1 \equiv_\delta g_2$, then $g \parallel g_1 \equiv_\delta g \parallel g_2$, $g \parallel g_1 \equiv_\delta g_2$, $g \parallel g_1 \equiv_\delta g \parallel g_2$ and $\partial_H(g_1) \equiv_\delta \partial_H(g_2)$, for all $H \subseteq A$.

**Proof:** The proof for all operators, except $\partial_H$, follows the standard arguments of ACP (see, e.g., [BW90]). For $\partial_H$, $H \subseteq A$, the proof proceeds as follows. Suppose $g_1 \equiv_\delta g_2$ and let $\mathcal{R} \subseteq V_1 \times V_2$ be a δ-bisimulation between $g_1$ and $g_2$. We show that $\mathcal{R}$ is also a δ-bisimulation between $\partial_H(g_1)$ and $\partial_H(g_2)$, $H \subseteq A$.

Let $(v_1, v_2) \in \mathcal{R}$. There are three cases to consider:

- **initials**($v_1$) $\subseteq H_\delta$: then in $\partial_H(g_1)$ the transitions of $v_1$ are of the form $v_1 \xrightarrow{a} v'_1$ with $a \in H_\delta$. Since $g_1 \equiv_\delta g_2$, in $\partial_H(g_2)$ there exists a $v'_2$ with $\mathcal{R}(v'_1, v'_2)$ and $v_2 \xrightarrow{a} v'_2$.

- **initials**($v_1$) $\not\subseteq H_\delta$: then in $\partial_H(g_1)$ all transitions of $v_1$ are of the form $v_1 \xrightarrow{\delta} v'_1$. Since $g_1 \equiv_\delta g_2$, in $\partial_H(g_2)$ all transitions of $v_2$ are likewise of the form $v_2 \xrightarrow{\delta} v'_2$. By the weaker condition on δ-transitions in a δ-bisimulation, this is enough.

- **initials**($v_1$) = ∅: then **initials**($v_1$) = ∅ in $\partial_H(g_1)$ and, since $g_1 \equiv_\delta g_2$, **initials**($v_2$) = ∅ in $\partial_H(g_2)$.

By considering the same three cases with the roles of $v_1$ and $v_2$ reversed, we are done.

To prove the completeness of $\text{ACP}_T(A)$ for finite processes, we first introduce the notion of a "basic term" for closed $\text{ACP}_T(A)$ terms. We will subsequently prove an "elimination theorem" stating that any closed $\text{ACP}_T(A)$ term can be reduced to a basic term using the axioms of $\text{ACP}_T(A)$. Combined with the completeness of BPA(A), this will be enough to prove the completeness of $\text{ACP}_T(A)$. 

10
Definition 2.7 A basic term is defined inductively as follows:

- \( a \in A_\delta \) is a basic term.
- Let \( t_1, t_2 \) be basic and \( a \in A \). Then \( t_1 + t_2 \) and \( a \cdot t_1 \) are basic.

Note that a basic term uses a restricted form of sequential composition known as action prefixing, and that a basic term is a \( \text{BPA}(A_\delta) \) term; i.e., a \( \text{BPA}(A) \) term treating \( \delta \) as an additional atomic action.

To prove the elimination theorem we introduce a term rewriting system based on \( \text{ACP}_\delta^+(A) \) for which we prove a strong normalization result. The rewrite system \( \text{RACP}_\delta^+(A) \) consists of axioms A1-5, A7, C3, CM1-9, I1-3, and D1-2, treated as rewrite rules with left-to-right orientation, plus the rules

\[
\begin{align*}
    x + (y + z) & \rightarrow (x + y) + z & \text{A2'}
    a|b = c & \Rightarrow a|b \rightarrow c & \text{C0'}
    a|\delta & \rightarrow \delta & \text{C3'}
    c \in H_\delta & \Rightarrow \partial_H(c + z) \rightarrow \partial_H(z) & \text{D3.1'}
    c \in H_\delta & \Rightarrow \partial_H(c \cdot z + y) \rightarrow \partial_H(y) & \text{D3.1''}
    I(x + y) \cap H_\delta = \emptyset & \Rightarrow \partial_H(x + y) \rightarrow \partial_H(x) + \partial_H(y) & \text{D3.2'}
    \partial_H(a \cdot z) & \rightarrow \partial_H(a) \cdot \partial_H(z) & \text{D4'}
\end{align*}
\]

Notice that all these rules follow easily from \( \text{ACP}_\delta(A) \). The normal forms of the rewrite system \( \text{RACP}_\delta^+(A) \) are defined as follows.

Definition 2.8 A closed \( \text{ACP}_\delta^+(A) \) term \( t \) is in normal form if for all \( \text{RACP}_\delta^+(A) \) reduction paths of the form

\[
t = t_0 \rightarrow t_1 \rightarrow t_2 \rightarrow \cdots
\]

\( t_{i+1} \) follows from \( t_i \) through the application of either rule A1, A2, or A2' (and no other), for all \( i \geq 0 \).

Proposition 2.3 A normal form is a basic term.

Proof: Let \( t \) be a normal form and suppose \( t \) is not basic. Let \( t' \) be a minimal subterm of \( t \) that is not basic. Then \( t' \) has one of the following forms:

1. \( p \parallel q \)
2. \( p \parallel q \)
3. \( p|q \)
4. \( \partial_H(p) \)
5. \( p \cdot q \) (with \( p \) not an atom or \( p = \delta \))

and both \( p \) and \( q \) basic terms due to minimality. We show that in each case a rule of \( \text{RACP}_\delta^+(A) - \{\text{A1, A2, A2'}\} \) can still be applied, thereby proving the result by contradiction. Take, for example, the second case. Since \( p \) is a basic term, there are three subcases to consider:

\[
\begin{align*}
    x + (y + z) & \rightarrow (x + y) + z & \text{A2'}
    a|b = c & \Rightarrow a|b \rightarrow c & \text{C0'}
    a|\delta & \rightarrow \delta & \text{C3'}
    c \in H_\delta & \Rightarrow \partial_H(c + z) \rightarrow \partial_H(z) & \text{D3.1'}
    c \in H_\delta & \Rightarrow \partial_H(c \cdot z + y) \rightarrow \partial_H(y) & \text{D3.1''}
    I(x + y) \cap H_\delta = \emptyset & \Rightarrow \partial_H(x + y) \rightarrow \partial_H(x) + \partial_H(y) & \text{D3.2'}
    \partial_H(a \cdot z) & \rightarrow \partial_H(a) \cdot \partial_H(z) & \text{D4'}
\end{align*}
\]
(a) \( p \) is of the form \( p_1 + p_2 \). Apply CM4.
(b) \( p \) is an atomic action \( a \in A_5 \). Apply CM2.
(c) \( p \) is of the form \( a \cdot p_1, a \in A_5 \). Apply CM3.

The other four cases are proved similarly.  

Note that the converse of this result does not hold, e.g., \( a + a \) is basic but not in normal form.

**Lemma 2.1** The rewrite system \( \text{RACP}_T(A) \) is strongly normalizing modulo \( A_1, A_2, A_2' \), i.e., every infinite reduction path contains \( A_1, A_2, A_2' \) steps only from some point onwards.

**Proof:** Let \( \Pi = (\gamma_0, t_0) \rightarrow (\gamma_1, t_1) \rightarrow (\gamma_2, t_2) \rightarrow \cdots \) be an infinite reduction path in \( \text{RACP}_T(A) \) where \( \gamma_i \) is the (possibly empty) condition associated with rewriting \( t_i \) into \( t_{i+1} \). We omit from \( \Pi \) any steps having to do with normalizing the expression \( I(x + y) \) in the condition to D3.2'-steps. We prove that only finitely many of the steps in \( \Pi \) can differ from \( A_1, A_2, A_2' \).

We transform the reduction sequence \( \Pi \) into a reduction sequence \( \Pi' \) of \( \text{RACP}(A) \) [BK84] as follows:

- Expand each D3.1' step of the form \( (\gamma_i, t_i) \rightarrow (\gamma_{i+1}, t_{i+1}) \) into a finite valid rewriting of \( \text{RACP}(A) \) depending on the condition \( \gamma_i \) as follows:
  
  \[
  c = \delta: \theta_H(\delta + z) \xrightarrow{A_1} \theta_H(\delta + \delta) \xrightarrow{A_6} (\gamma_{i+1}, \theta_H(z))
  \]

  \[
  c \in H: (c \in H, \theta_H(c + x)) \xrightarrow{A_3} (c \in H, \theta_H(c) + \theta_H(x)) \xrightarrow{D_2} \delta + \theta_H(\delta + \delta) \xrightarrow{A_1} \theta_H(z) + \delta \xrightarrow{A_6} (\gamma_{i+1}, \theta_H(x))
  \]

- Expand each D3.1'' step of the form \( (\gamma_i, t_i) \rightarrow (\gamma_{i+1}, t_{i+1}) \) into a finite valid rewriting of \( \text{RACP}(A) \) depending on the condition \( \gamma_i \) as follows:
  
  \[
  c = \delta: \theta_H(\delta \cdot x + y) \xrightarrow{A_7} \theta_H(\delta + y) \xrightarrow{A_3} \theta_H(y + \delta) \xrightarrow{A_6} (\gamma_{i+1}, \theta_H(y))
  \]

  \[
  c \in H: (c \in H, \theta_H(c \cdot x + y)) \xrightarrow{D_3} \theta_H(c \cdot x) + \theta_H(y) \xrightarrow{D_4} (c \in H, \theta_H(c) \cdot \theta_H(x) + \theta_H(y)) \xrightarrow{D_2} \delta \cdot \theta_H(x) + \theta_H(y) \xrightarrow{A_7} \delta + \theta_H(y) \xrightarrow{A_3} \theta_H(x) + \delta \xrightarrow{A_6} (\gamma_{i+1}, \theta_H(y))
  \]

- Transform each D3.2' step of the form \( (\gamma_i, t_i) \rightarrow (\gamma_{i+1}, t_{i+1}) \) into the conditionless step \( t_i \rightarrow (\gamma_{i+1}, t_{i+1}) \), as D3.2' is valid in \( \text{RACP}(A) \) in all cases (i.e., restriction distributes over plus).\(^2\)

Now we obtain an infinite reduction path in \( \text{RACP}(A) \) and from [BW90] it follows that this reduction path contains finitely many non-\( A_1, A_2, A_2' \) steps. But the same must hold for the original reduction sequence.  

Note that in the transformation of a \( \text{RACP}_T(A) \) reduction sequence to a \( \text{RACP}(A) \) reduction sequence, each non-\( A_1, A_2, A_2' \) step is replaced by at most six non-\( A_1, A_2, A_2' \) steps.

We now present the "elimination theorem" for \( \text{ACP}_T(A) \).

**Lemma 2.2** Let \( p \) be a closed \( \text{ACP}_T(A) \) term. Then using \( \text{RACP}_T(A) \), \( p \) can be reduced in finitely many steps to a basic term.

\(^2\)One could, of course, leave condition \( \gamma_i \) intact and still have a valid reduction step in \( \text{RACP}(A) \).
Proof: If \( p \) is a basic term we are done. Otherwise, by Proposition 2.3, \( p \) is not in normal form. By Definition 2.8, there exists a reduction sequence

\[
p = p_0 \rightarrow t_0^0 \rightarrow t_1^1 \rightarrow \cdots \rightarrow t_n^0 = p_1
\]
such that \( t_{n-1}^0 \rightarrow t_n^0 \) is not an A1, A2, A2' reduction. If \( p_1 \) is basic we are done. Otherwise there exists another reduction sequence

\[
p_1 = t_0^1 \rightarrow t_1^1 \rightarrow \cdots \rightarrow t_n^1 = p_2
\]
such that \( t_{n-1}^1 \rightarrow t_n^1 \) is not an A1, A2, A2' reduction. This line of reasoning cannot proceed indefinitely: due to strong normalization (Lemma 2.1) \( p_i \), for some \( i \geq 0 \), is a basic term. Otherwise, an infinite reduction with infinitely many non-A1, A2, A2' steps would have been constructed which is impossible.

\[\square\]

**Theorem 2.2**

1. \( \mathcal{F}/\equiv_{\delta} \models ACP_\delta^+ (A) \)

2. For all closed expressions \( p, q \) over \( \Sigma(ACP_\delta^+ (A)) \):

\[\mathcal{F}/\equiv_{\delta} \models p = q \implies ACP_\delta^+ (A) \vdash p = q.\]

**Proof:** For part 1, we consider axioms A7 and D1-D4. The fact that \( \mathcal{F}/\equiv_{\delta} \) is a model of the rest of the axioms of \( ACP_\delta^+ (A) \) follows standard arguments as presented, e.g., in [BW90]. For A7, both \( \delta \cdot z \) and \( \delta \) initially can perform but a single \( \delta \)-transition. Since \( \equiv_{\delta} \) matches one \( \delta \)-transition with any other \( \delta \)-transition (i.e., without regard to the destination states), we are done. The soundness of D1 and D2 is trivial since in both cases the left- and right-hand side terms represent isomorphic processes.

For D3.1, the initial transitions of \( z \) will be deleted from the root of \( z + y \) by the \( \partial_H \) operation, thereby again resulting in isomorphic processes. D3.2 could fail only if \( z, y \not\equiv_{\delta} \) and either \( \partial_H(z) = \delta \) or \( \partial_H(y) = \delta \). The condition to the axiom ensures against this. Note that D3.2 is still sound under the weaker condition

\[I(z) - H_{\delta} \neq \emptyset \quad \text{and} \quad I(y) - H_{\delta} \neq \emptyset\]

but the natural probabilistic extension of the resulting axiom is not sound (see Section 3.4), and is thus rejected. Finally, D4 also represents isomorphic processes.

For part 2, suppose \( p \equiv_{\delta} q \). Reduce \( p, q \) to normal forms \( p', q' \) using RACP_{\delta}^+ (A); by Lemma 2.2, \( p', q' \) are basic terms. E.g. part 1, \( p' \equiv_{\delta} p \equiv_{\delta} q \equiv_{\delta} q' \), and thus \( p' \equiv_{\delta} q' \). In reducing \( p, q \) to their normal forms, we have been rewriting by A7 whenever possible. We may therefore conclude that \( p' \equiv_{\delta} q' \) (treating \( \xi \) as just another atomic action), and by Theorem 2.1, BPA(A_{\delta}) \vdash p' = q'. Then \( ACP_{\delta}^+ (A) \vdash p = p' = q' = q \). \[\square\]

### 2.3.3 Connections Between ACP and ACP^+_\delta

Let \( A \) be the usual bisimulation model for \( ACP(A) \), and let \( A^- = \mathcal{F}/\equiv_{\delta} \) be the bisimulation model for \( ACP_{\delta}^+ (A) \). Then for \( p, q \) closed expressions over \( \Sigma(ACP(A)) \) we have the following results, which we state without proof.

1. *Completeness of ACP_{\delta}^+ (A):* \( A^- \models p = q \implies ACP_{\delta}^+ (A) \vdash p = q \)
   (This is just part 2 of Theorem 2.2.)

13
2. Completeness of ACP (A)[BW90]: \( A \vdash p = q \implies ACP(A) \vdash p = q \)

3. \( A^- \vdash p = q \implies A \vdash p = q \). This implies that A can be homomorphically embedded in A using the identity mapping.

4. \( A \vdash p = q \implies A^- \vdash \delta_0(p) = \delta_0(q) \). This implies that A can be homomorphically embedded in A using the homomorphism \( \varphi : A \rightarrow A^- \), such that \( \varphi(x) = \delta_0(x) \).

5. \( ACP(A) \vdash \delta_0(p) = p \)

6. \( ACP(A) \vdash p = q \implies ACP_I(A) + \{ x + \delta = z \} \vdash p = q \)

7. \( ACP_I(A) \vdash \delta_0(x + \delta) = \delta_0(x) \)

3 A Probabilistic Version of ACP

Our discussion of probabilistic ACP will proceed in a manner similar to before. For each of the axiom systems \( AX \in \{ BPA(A), PA(A), ACP_I(A) \} \), a probabilistic version \( prAX \) will be introduced, along with a probabilistic version of its process graph model. Completeness in these models will also be demonstrated.

3.1 Probabilistic BPA

3.1.1 Equational Specification

Notation: As usual, \((0,1)\) denotes the open interval of the real line \( \{ r \in \mathbb{R} | 0 < r < 1 \} \), and \([0,1]\) denotes the closed interval of the real line \( \{ r \in \mathbb{R} | 0 \leq r \leq 1 \} \). We let \( \pi, \rho, \sigma, \text{ and } \theta \), possibly subscripted, range over these intervals.

The signature \( \Sigma(prBPA(A)) \) over the sort \( prP \) (for probabilistic processes) is given by:

\[
\Sigma(prBPA(A)) = \{ a : \rightarrow prP | a \in A \} \cup \{ + : prP \times prP \rightarrow prP | \pi \in (0,1) \} \cup \\
\{ : prP \times prP \rightarrow prP \}
\]

The operator + has been replaced by the family of operators \(+,\) for each probability \( \pi \) in the interval \((0,1)\), and is now called probabilistic alternative composition. Intuitively, the expression \( x + \pi y \) behaves like \( x \) with probability \( \pi \) and like \( y \) with probability \( 1 - \pi \). Probabilistic alternative composition is generative [vGSST90] in that a single distribution (viz. the discrete probability distribution \( \{ p, 1 - p \} \)) is associated with the two alternatives \( x \) and \( y \). As mentioned in Section 1, these probabilities are conditional with respect to the set of actions permitted by the environment. This will become clear in Section 3.4 with the introduction of the restriction operator \( \delta_H \) in the setting of probabilistic ACP.

We have the following axioms for \( prBPA(A) \):

\[
\begin{align*}
& z +_\pi y = y +_1 -\pi z & \quad prA1 \\
& z +_{\pi} (y +_\rho z) = (z +_{\pi/(\pi +_\rho -_{\pi\rho})} y) +_{\pi+\rho -_{\pi\rho}} z & \quad prA2 \\
& z +_\pi z = z & \quad prA3 \\
& (z +_\pi y) \cdot z = z \cdot z +_\pi y \cdot z & \quad prA4 \\
& (z \cdot y) \cdot z = z \cdot (y \cdot z) & \quad prA5
\end{align*}
\]

14
Axiom \( prA2 \) has a left-to-right orientation in that the probability indices on the right-hand side are derived from probability indices \( \pi, \rho \) on the left-hand side. A right-to-left version of \( prA2 \), which will prove useful later, is given by:

\[
(x +_\pi y) +_\rho z = x + \frac{\pi}{1-\pi} (y + \frac{\rho}{1-\rho} z)
\]

### 3.2 Probabilistic Graph Model

As in Section 2.1.2, we consider process graphs, with labels from \( A \), as a model for \( prBPA(A) \). Additionally, a probability distribution will be ascribed to each node's outgoing transitions.

**Definition 3.1** A probabilistic process graph \( g \) is a triple \( <V, \tau, \mu> \) such that \( V \) and \( \tau \) are as in Definition 2.1 and \( \mu : (V \times A \times V) \rightarrow [0,1] \), the transition distribution function of \( g \), is a total function satisfying the following stochasticity condition:

\[
\forall v \in V \sum_{a \in A, v' \in V} \mu(v, a, v') \in \{0, 1\}
\]

Intuitively, \( \mu(v, a, v') = \pi \) means that, with probability \( \pi \), node \( v \) can perform an \( a \)-transition to node \( v' \). A node in a stochastic probabilistic process graph performs some transition with probability 1, unless it is an endpoint. Predicate \( endpoint(v) \) is true iff \( v \) is an endpoint. We denote by \( prG \) the family of all probabilistic process graphs.

The notion of strong bisimulation for nondeterministic processes has been extended by Larsen and Skou [LS89] to reactive probabilistic processes in the form of probabilistic bisimulation. Here we define probabilistic bisimulation on generative probabilistic processes and to do so we first need to lift the definition of the transition distribution function as follows:

\[
\mu : (V \times A \times 2^V) \rightarrow [0,1] \text{ such that } \mu(v, a, S) = \sum_{v' \in S} \mu(v, a, v')
\]

Intuitively, \( \mu(v, a, S) = \rho \) means that node \( v \), with total probability \( \rho \), can perform an \( a \)-transition to some node in \( S \).

**Definition 3.2** ([LS89]) Let \( g_1 = <V_1, r_1, \mu_1> \), \( g_2 = <V_2, r_2, \mu_2> \) be probabilistic process graphs. A probabilistic bisimulation between \( g_1 \) and \( g_2 \) is an equivalence relation \( \mathcal{R} \subseteq (V_1 \cup V_2) \times (V_1 \cup V_2) \) with the following properties:

- \( \mathcal{R}(r_1, r_2) \)
- \( \forall v \in V_1, w \in V_2 \text{ such that } \mathcal{R}(v, w) \); \( \forall a \in A, S \subseteq (V_1 \cup V_2) / \mathcal{R}, \mu_1(v, a, S \cap V_1) = \mu_2(w, a, S \cap V_2) \)

Graphs \( g_1 \) and \( g_2 \) are probabilistically bisimilar, written \( g_1 \Leftrightarrow^{pr} g_2 \), if there exists a probabilistic bisimulation between \( g_1 \) and \( g_2 \).

Intuitively, two nodes are probabilistically bisimilar if, for all actions in \( A \), they transit to probabilistic bisimulation classes with equal probability. Note the somewhat subtle use of recursion in the definition.
We now define the operators of \( \text{prBPA}(A) \) on the domain \( \text{pr}\mathcal{F} \) of finite probabilistic process graphs, i.e., probabilistic process graphs that are finitely branching and acyclic in terms of their transitions of non-zero probability. Therefore, \( \text{pr}\mathcal{F} \subset \text{pr}\mathcal{G} \). For this purpose, it is convenient to assume, as in the non-probabilistic case, that the root nodes of probabilistic process graphs are not endpoints. For the remainder of Section 3, unless otherwise stated, let \( g_1 = \langle V_1, r_1, \mu_1 \rangle, g_2 = \langle V_2, r_2, \mu_2 \rangle \) be finite probabilistic process graphs satisfying the non-endpoint root assumption such that \( V_1 \cap V_2 = \emptyset \).

**Definition 3.3** The operators \( a \in A, +_\pi, \text{ and } \cdot \) are defined on \( \text{pr}\mathcal{F} \) as follows:

- \( a \in A \): The process graph for each of these constants is given by \( \langle \{ r_a, v \}, r_a, \mu_a \rangle \), where \( \mu_a(r_a, a, v) = 1 \) is the only transition with non-zero probability.
- \( g_1 +_\pi g_2 \): is given by \( \langle V_1 \cup V_2 \cup \{ r \} - \{ r_1, r_2 \}, r, \mu \rangle \) where \( r \notin V_1 \cup V_2 \) and
  \[
  \mu(r, a, v') = \pi \cdot \mu_1(r_1, a, v') \quad \text{if } v' \in V_1 \\
  \mu(r, a, v') = (1 - \pi) \cdot \mu_2(r_2, a, v') \quad \text{if } v' \in V_2 \\
  \mu(v, a, v') = \mu_1(v, a, v') \quad \text{if } v, v' \in V_1 \\
  \mu(v, a, v') = \mu_2(v, a, v') \quad \text{if } v, v' \in V_2 \\
  \mu(v, a, v') = 0 \quad \text{otherwise}
  \]
- \( g_1 \cdot g_2 \) is obtained by appending a copy of \( g_2 \) at each endpoint of \( g_1 \), and is analogous to sequential composition in the non-probabilistic setting (Definition 2.3). In detail, \( g_1 \cdot g_2 \) is given by \( \langle V_1 \cup V_2 - \{ r_2 \}, r_1, \mu \rangle \) where
  \[
  \mu(v, a, v') = \begin{cases} 
  \mu_1(v, a, v') & \text{if } v, v' \in V_1 \\
  \mu_2(r_2, a, v') & \text{if } v \in V_1, \text{ endpoint}(v), v' \in V_2 \\
  \mu_2(v, a, v') & \text{if } v, v' \in V_2 \\
  0 & \text{otherwise}
  \end{cases}
  \]

So, in the definition of \( g_1 +_\pi g_2 \), the transitions from \( r_1, r_2 \) are now assumed by the new root \( r \), with their probability of occurrence weighted appropriately. Similarly, the transitions of \( r_2 \) in \( g_1 \cdot g_2 \) are assumed by each endpoint of \( g_1 \), with their original probabilities intact.

As in the non-probabilistic case, for \( t \) a closed \( \text{prBPA}(A) \) term, we write \( \text{graph}(t) = \langle V_t, r_t, \mu_t \rangle \) to denote the probabilistic process graph obtained inductively on \( t \) using Definition 3.3. We also write \( p \trianglelefteq^\pi q \) as shorthand for \( \text{graph}(p) \trianglelefteq^\pi \text{graph}(q) \). The definition of \( \text{graph}(t) \) and the just-mentioned notational shorthand extend in the obvious way to the axiom systems \( \text{prBPA} A \) and \( \text{prACP}_? (A) \) considered later in this section.

We will subsequently prove that the axioms of \( \text{prBPA}(A) \) are complete in this model. To admit sound equational reasoning, in particular, the substitution of equals for equals, we first show that \( \equiv^\pi \) is a congruence in \( \text{prBPA}(A) \). Let \( V \) be an arbitrary set with \( v \in V \). For any equivalence relation \( \mathcal{R} \) over \( V \) we use \( [v]_{\mathcal{R}} \) to denote the set \( \{ w \in V \mid (v, w) \in \mathcal{R} \} \); i.e., \( [v]_{\mathcal{R}} \) is the equivalence class of \( v \) induced by \( \mathcal{R} \). Also, \( \text{Id}_V = \{ (v, v) \mid v \in V \} \) denotes the identity relation on \( V \).

**Proposition 3.1** If \( g_1 \equiv^\pi g_2 \), then \( g +_\pi g_1 \equiv^\pi g +_\pi g_2 \), \( g \cdot g_1 \equiv^\pi g \cdot g_2 \), and \( g_1 \cdot g \equiv^\pi g_2 \cdot g \).

**Proof:** Let \( g = \langle V, r, \mu \rangle \) such that \( V \cap (V_1 \cup V_2) = \emptyset \), assume \( g_1 \equiv^\pi g_2 \), and let \( \mathcal{R} \) be a probabilistic bisimulation between \( g_1 \) and \( g_2 \). We now consider each of the operators in succession.
For $+_\pi$, let $r_t^+$ be the root and $\mu_t^+$ be the tdf of $g +_\pi g_i$, $i = 1, 2$. We show that

$$R' = \{(r_1^+, r_2^+), (r_2^+, r_1^+)\} \cup R \cup Id_{V \cup \{r^+_1, r^+_2\}}$$

is a probabilistic bisimulation between $g +_\pi g_1$ and $g +_\pi g_2$. First note that because $R$ is an equivalence relation, so is $R'$. By the nature of $R'$, we are left to show that the "carrier condition" (the second condition of Definition 3.3) holds for $(r_1^+, r_2^+)$. For $a \in A$, the only $a$-transitions of $r_t^+$ of non-zero probability are of the form:

1. $\mu_t^+(r_t^+, a, [v_t]^R) = \mu(r_t, a, v') \cdot \pi$, where $v' \in V$; or
2. $\mu_t^+(r_t^+, a, [v_t']R) = \mu_1(r_1, a, [v_t']R) \cdot (1 - \pi)$, where $v_t' \in V_1$.

Well, we also have $\mu_t^+(r_t^+, a, [v_t']R) = \mu(r_t, a, v') \cdot \pi$ and, because $g_1 \equiv_{pr} g_2$, $\mu_t^+(r_t^+, a, [v_t']R) = \mu_1(r_1, a, [v_t']R) \cdot (1 - \pi)$. This completes the case for $+_\pi$.

For both cases of sequential composition, a straightforward argument demonstrates that $R \cup Id_V$ is an appropriate probabilistic bisimulation.

The graph model for $prBPA(A)$ is now given by $prF / \equiv_{pr}$. To prove completeness of $prBPA(A)$, we introduce the notation

$$\sum_{i=1}^{n} \pi_i x_i$$

with $\sum \pi_i = 1$ and $\pi_i > 0$ for all $i$. So, in particular, when $n = 1$, $\pi_1 = 1$. This notation abbreviates right-nested probabilistic alternative composition expressions as follows:

$$\sum_{i=1}^{1} \pi_i x_i = x_1 \quad \text{and} \quad \sum_{i=1}^{n+1} \pi_i x_i = x_1 + \pi_1 \left( \sum_{i=1}^{n} \pi_i x_i \right)$$

Note that in this notation $\sum_{i=1}^{n} \pi_i$ is a derived $n$-ary operator with operands $x_i$. To illustrate, the left-hand side of equation $prA2$ may be written:

$$\sum_{i=1}^{3} \pi_i x_i$$

where $\pi_1 = \pi$, $\pi_2 = (1 - \pi)\rho$, $\pi_3 = (1 - \pi)(1 - \rho)$, and $x_1 = x$, $x_2 = y$, $x_3 = z$.

This summation form notation is useful as it directly reflects the transition structure of the probabilistic process graph underlying the nested probabilistic alternative composition. That is, consider the summation form $\sum \pi_i a \cdot z_i$ of action-prefixed processes. The graph of this summation will have, for each $i$, a probability-$\pi_i$ $a_i$-transition from its root to the node representing the root of $graph(z_i)$.

The following two lemmas for manipulating summation forms, the proofs of which appear in Appendix A, will prove useful in the completeness proof for $prBPA(A)$. The first allows summands to be reordered arbitrarily, retaining their original probabilities, while the second allows two syntactically identical summands to be merged into one summand, summing the probabilities in the process.

**Lemma 3.1** For any permutation $\xi$ of $\{1, \cdots, n\}$, $n \geq 2$,

$$prBPA(A) \vdash \sum_{i=1}^{n} \pi_i x_i = \sum_{i=1}^{n} \pi_{\xi(i)} x_{\xi(i)}$$
Lemma 3.2 In the summation form \( \sum_{i=1}^{n+1} [\pi_i] z_i \), let \( z_1 \) and \( z_2 \) be syntactically identical. Then

\[
prBPA(A) \vdash \sum_{i=1}^{n+1} [\pi_i] z_i = \sum_{i=1}^{n} [\rho_i] y_i
\]

where \( \rho_1 = \pi_1 + \pi_2, y_1 = z_1 \), and \( \rho_i = \pi_{i+1}, y_i = z_{i+1}, 2 \leq i \leq n \).

We now use summation-form notation to define a kind of normal form for closed \( prBPA(A) \) terms.

Definition 3.4 A probabilistic basic term is a summation form \( \sum_{i=1}^{n} [\pi_i] t_i \) where \( t_i \) is either some \( a \in A \) or of the form \( b \cdot t'_i \), where \( b \in A \) and \( t'_i \) is a probabilistic basic term. A probabilistic normal form is a probabilistic basic term \( \sum_{i=1}^{n} [\pi_i] t_i \) such that \( t_i \neq pr t_j, 1 \leq i \neq j \leq n \).

Note that a probabilistic basic term, like a basic \( ACpi(A) \) term of Section 2.3, uses action prefixing, while a probabilistic normal form bears the additional constraint that its summands are pairwise inequivalent.

The depth of a probabilistic basic term \( t \), denoted \( d(t) \), is essentially the maximum number of nested prefixes in \( t \). The inductive definition of \( d \) is as follows:

- \( d(a) = 1 \)
- \( d(a \cdot t) = 1 + d(t) \)
- \( d(\sum_{i=1}^{n} [\pi_i] t_i) = \max(d(t_i)) \)

Lemma 3.3 For every closed \( prBPA(A) \) term \( t \), there is a probabilistic normal form \( s \) such that \( prBPA(A) \vdash t = s \).

Proof: The proof has two parts. In the first part, we prove that a closed term \( t \) can be proven equal to a probabilistic basic term. The second part handles the constraint that the summands are pairwise inequivalent. The first part is simpler and follows the line of reasoning in [BW90]. That is, we use a term rewriting system to convert \( t \) into a term whose only instances of sequential composition are of the form \( a \cdot t' \), i.e., action prefixing. The rewrite system is based on \( prBPA(A) \) axioms \( prA4 \) and \( prA5 \) and is given by:

\[
\begin{align*}
(x \rightarrow y) \cdot z & \rightarrow z \cdot x \rightarrow y \cdot z \\
(x \cdot y) \cdot z & \rightarrow z \cdot (y \cdot z)
\end{align*}
\]

It is not hard to see that this term rewrite system is confluent and strongly normalizing, and that a normal form of a closed term uses only action prefixing. Therefore, given a closed \( prBPA(A) \) term \( t \), we can convert it into a probabilistic basic term by:

1. Reduce \( t \) until a normal form is reached.
2. Use \( prA3 \), with right-to-left orientation, to rewrite all instances of left-nested summations into right-nested summations. The resulting term can then be expressed as a summation form.
By the first part of the proof, assume $t$ is a probabilistic basic term of the form $\sum_{i=1}^{n} [\pi_i] t_i$ and consider the partition $\{ B_1, \ldots, B_k \}$ of $\{1, \ldots, n\}$ such that $(i, i') \in B_j$ if $t_i \Rightarrow in t'$. We prove by induction on the depth of $t$ that:

$$\text{prBPA}(A) \vdash t = \sum_{j=1}^{m} [\rho_j] t'_j$$

where $\rho_j = \sum \{ \pi_i \mid i \in B_j \}$, $t'_j = t_i$ for an arbitrarily chosen $i \in B_j$, and $m \leq n$. Note that the term on the right-hand-side of this equation is indeed a probabilistic normal form. If the depth of $t$ is 1 then each $t_i$ is a constant and the indices in a block $B_j$ correspond to (all of the) multiple occurrences of a constant $a$. If $|B_j| = 1$ then we are done. Otherwise, apply the following procedure $|B_j| - 1$ times: move two instances of $a$ to the two left-most positions within the summation form using Lemma 3.1. Merge the two instances into one, occupying the left-most position in the resulting summation form, using Lemma 3.2. The associated probability of this single instance of $a$ will be the sum of the probabilities of the original two instances, as desired.

Next, assume the result for probabilistic basic terms of depth $k$ and let $d(t) = k + 1$. There are two cases.

1. The indices in a block $B_j$ correspond to the multiple occurrences of a constant $a$. The base case reasoning suffices here.
2. The indices in a block $B_j$ correspond to equivalent terms of the form $a \cdot t'$, $b \cdot t''$, where $t'$, $t''$ are basic. If $|B_j| = 1$ then we are done. Otherwise, apply the following procedure $|B_j| - 1$ times. Choose two instances $a \cdot t'$, $b \cdot t''$ of equivalent terms from $B_j$. Since $a \cdot t' \Rightarrow in b \cdot t''$, then $a = b$ and $t' \Rightarrow in t''$, and, by the induction hypothesis, $\text{prBPA}(A) \vdash t' = t''$. By substitution of equals for equals, we have $\text{prBPA}(A) \vdash a \cdot t' = b \cdot t''$, and, as in the first case, we can use Lemmas 3.1 and 3.2 to merge these two summands into a single summand, either $a \cdot t'$ or $b \cdot t''$, the choice being arbitrary. The associated probability of the merged term will be the sum of the associated probabilities of $a \cdot t'$ and $b \cdot t''$, as desired.

The relationship observed above between a probabilistic summation form and its underlying probabilistic process graph can be strengthened in the case of probabilistic normal forms.

**Proposition 3.2** For $t$ a probabilistic normal form, $t$ has a summand $a \in A$, with associated probability $\pi$, iff $\mu_{t}([v] \rightarrow \pi \in \mathcal{G}) = \pi$ where $v$ is an endpoint. Also, $t$ has a summand $a \cdot t'$, with associated probability $\pi$, iff $\mu_{t'}([v] \rightarrow \pi \in \mathcal{G}) = \pi$ where $v' \in V'$ is the node in $\text{graph}(t')$ representing the root of $\text{graph}(t')$.

We now prove that our algebra $\text{prF/} \Leftrightarrow \Rightarrow$ is a model of $\text{prBPA}(A)$ and that $\text{prBPA}(A)$ constitutes a complete axiomatization of process equivalence in $\text{prF/} \Leftrightarrow$.

**Theorem 3.1**

1. $\text{prF/} \Leftrightarrow \Rightarrow \models \text{prBPA}(A)$
2. For all closed expressions $s, t$ over $\Sigma(\text{prBPA}(A))$:
   $$\text{prF/} \Leftrightarrow \Rightarrow \models s = t \implies \text{prBPA}(A) \vdash s = t.$$
Proof: For part 1, consider first prA1 and prA2. In both cases the left- and right-hand side terms represent isomorphic probabilistic process graphs, with the transitions from the root of $x$ weighted by $\pi$ and the transitions from the root of $y$ weighted by $1 - \pi$, in the case of prA1; and the root transitions of $x$ weighted by $\pi$, the root transitions of $y$ rooted by $(1 - \pi)\rho$, and the root transitions of $z$ weighted by $(1 - \pi)(1 - \rho)$, in the case of prA2.

Graph isomorphism arguments also suffice for prA4 and prA5, while the soundness of prA3 is established by the probabilistic bisimulation $\{(r_{x+z}, r_{x}), (r_{x}, r_{x+z})\} \cup I_d v \cdot u \cdot \{,.+••\}$.

For part 2, assume $s \equiv_{pr} t$ and also (relying on Lemma 3.3) that $s$ and $t$ are probabilistic normal forms, $s = \sum_i [\pi_i]s_i$ and $t = \sum_j [\rho_j]t_j$. We prove the result by induction on the maximum depth of $s$ and $t$. If the maximum depth is 1 then each summand of $s$ is a constant from $A$. Let $s_i = a$. Since $s \equiv_{pr} t$ and $t$ is a probabilistic normal form, by Proposition 3.2, $t$ also has a summand $t_j = a$ with $\rho_j = \pi_i$. A symmetric argument matches each constant summand of $t$ with a summand of $s$. Thus, $\mathsf{prBPA}(A) \vdash s = t$ by using Lemma 3.1 to reorder summands as necessary.

Next, assume the result for maximum depth $k$ and let the maximum depth of $s, t$ be $k + 1$. There are two cases.

1. The term $s$ has a constant summand. Here the base case reasoning suffices.

2. The term $s$ has a summand $s_i$ of the form $a \cdot s'$ and, by Proposition 3.2, $\mu_s(r,s,a,[r_i^{sr}]_{\equiv_{pr}}) = \pi_i$.
   Since $s \equiv_{pr} t$, $\mu_s(r,s,a,[r_i^{sr}]_{\equiv_{pr}}) = \pi_i$. But $t$ is a probabilistic normal form so, by Proposition 3.2 again, $t$ has a summand $t_j = a \cdot t'$ such that $t' \equiv_{pr} s'$ and $\rho_j = \pi_i$. By induction, $\mathsf{prBPA}(A) \vdash s' = t'$ and therefore (using substitution of equals for equals), $\mathsf{prBPA}(A) \vdash s_i = t_j$. A symmetric argument matches each action-prefixed summand of $t$ with a summand of $s$.

From the two cases, it follows that every summand of $s$ can be proved equal to a summand of $t$ and vice versa. Thus, $\mathsf{prBPA}(A) \vdash s = t$, by using Lemma 3.1 to reorder summands as necessary.

We also prove the following proposition:

**Proposition 3.3** The various forms of $+_{\pi}$ distribute over one another:

$$(x +_{\pi} y) +_{\rho} z = (x +_{\rho} z) +_{\pi} (y +_{\rho} z)$$

**Proof:**

$$(x +_{\rho} z) +_{\pi} (y +_{\rho} z) = x +_{\rho \pi} (z +_{\rho \pi} (y +_{\rho} z)) \quad (prA2)$$

$$= x +_{\rho \pi} (z +_{\rho \pi} (y +_{1-\rho} z)) \quad (prA1)$$

$$= x +_{\rho \pi} ((z +_{\rho} z) +_{\rho \pi} y) \quad (prA2)$$

$$= x +_{\rho \pi} (z +_{\rho} (y +_{\rho \pi} z)) \quad (prA3)$$

$$= x +_{\rho \pi} (y +_{\rho \pi} z) \quad (prA1)$$

$$= (x +_{\rho \pi} y) +_{\rho} z \quad (prA2)$$

Note that the last step makes direct use of the right-to-left oriented version of prA2.
3.3 Probabilistic PA

3.3.1 Equational Specification

The signature $\Sigma(\text{prPA}(A))$ extends that of $\text{prBPA}(A)$.

\[
\Sigma(\text{prPA}(A)) = \Sigma(\text{prBPA}(A)) \cup \{\lll_\sigma : \text{prP} \times \text{prP} \rightarrow \text{prP} \mid \sigma \in (0, 1)\} \cup \\
\{\lll_\sigma : \text{prP} \times \text{prP} \rightarrow \text{prP} \mid \sigma \in (0, 1)\}
\]

Intuitively, $\lll_\sigma$ is a probabilistic merge operator, with the left operand receiving relative probability $\sigma$ and the right operand relative probability $1 - \sigma$. As in $\text{PA}(A)$, $\lll_\sigma$ is a restricted version of $\lll_\sigma$ in which the first step must come from the left operand.

The axiom system $\text{prPA}(A)$ is obtained by adding to $\text{prBPA}(A)$ the following axioms for probabilistic merge and left-merge:

\[
\begin{align*}
& z \lll_\sigma y = z \lll_\sigma y + \sigma \lll_\sigma (1 - \sigma) z & \text{prM1} \\
& a \lll_\sigma y = a \cdot y & \text{prM2} \\
& (a \cdot z) \lll_\sigma y = a \cdot (z \lll_\sigma y) & \text{prM3} \\
& (z + \sigma y) \lll_\sigma z = (z \lll_\sigma z) + \sigma (y \lll_\sigma z) & \text{prM4}
\end{align*}
\]

3.3.2 Graph Model

As for $\text{prBPA}(A)$, we provide a bisimulation model for $\text{prPA}(A)$, and prove the completeness of the axioms on finite probabilistic processes.

Definition 3.5 The operators $\lll_\sigma$ and $\lll_\sigma$ are defined on $\text{prF}$ as follows:

$g_1 \lll_\sigma g_2$: is given by $<V_1 \times V_2, (r_1, r_2), \mu>$ where for all $a \in A$, $v_1, v'_1 \in V_1$, $v_2, v'_2 \in V_2$

\[
\mu((v_1, v_2), a, (v'_1, v'_2)) = \begin{cases} 
\sigma \cdot \mu_1(v_1, a, v'_1) & \text{if } -\text{endpoint}(v_2) \\
\mu_1(v_1, a, v'_1) & \text{otherwise}
\end{cases}
\]

$g_1 \lll_\sigma g_2$: is given by $<V_1 \times V_2, (r_1, r_2), \mu>$ where for all $a \in A$, $v_1, v'_1 \in V_1$, $v_2, v'_2 \in V_2$

\[
\begin{align*}
& \mu((v_1, v_2), a, (v'_1, v'_2)) = \begin{cases} 
\sigma \cdot \mu_1(v_1, a, v'_1) & \text{if } -\text{endpoint}(v_2) \\
\mu_1(v_1, a, v'_1) & \text{otherwise}
\end{cases} \\
& \mu((v_1, v_2), a, (v'_1, v'_2)) = \begin{cases} 
(1 - \sigma) \cdot \mu_2(v_2, a, v'_2) & \text{if } -\text{endpoint}(v_1) \\
\mu_2(v_2, a, v'_2) & \text{otherwise}
\end{cases}
\]

- $\mu((r_1, r_2), a, (v'_1, v'_2)) = \mu_1(r_1, a, v'_1)$
- if $v_1 \neq r_1$ or $v_2 \neq r_2$

\[
\begin{align*}
\mu((v_1, v_2), a, (v'_1, v'_2)) &= \begin{cases} 
\sigma \cdot \mu_1(v_1, a, v'_1) & \text{if } -\text{endpoint}(v_2) \\
\mu_1(v_1, a, v'_1) & \text{otherwise}
\end{cases} \\
\mu((v_1, v_2), a, (v'_1, v'_2)) &= \begin{cases} 
(1 - \sigma) \cdot \mu_2(v_2, a, v'_2) & \text{if } -\text{endpoint}(v_1) \\
\mu_2(v_2, a, v'_2) & \text{otherwise}
\end{cases}
\]

- if $v'_2 \neq r_2$

Note the careful treatment of endpoints in the above definition: in a merge, if one process terminates, the other continues with its original, unweighted probability. Also, in a left-merge,
special attention is paid to transitions from the root \((r_1, r_2)\) of \(g_1 \parallel g_2\): the first and third clauses collectively define the transition distribution function \(\mu\) on all transitions from \((r_1, r_2)\), with the third clause giving probability 0 to transitions starting with \(r_2\). We have that probabilistic bisimulation is a congruence in \(\text{prPA}(A)\).

**Proposition 3.4** If \(g_1 \equiv_{\text{pr}} g_2\), then \(g \parallel_{\sigma} g_1 \equiv_{\text{pr}} g \parallel_{\sigma} g_2\), \(g \parallel_{\sigma} g_1 \equiv_{\text{pr}} g \parallel_{\sigma} g_2\), and \(g_1 \parallel_{\sigma} g \equiv_{\text{pr}} g_2 \parallel_{\sigma} g\).

**Proof:** Let \(g = <V, r, \mu>\) such that \(V \cap (V_1 \cup V_2) = \emptyset\), assume \(g_1 \equiv_{\text{pr}} g_2\), and let \(\mathcal{R}\) be a probabilistic bisimulation between \(g_1\) and \(g_2\). We first show that

\[\mathcal{R}' = \{(v_1, v_2) \mid (v_1, v_2) \in \mathcal{R}\}\]

is a probabilistic bisimulation between \(g \parallel_{\sigma} g_1\) and \(g \parallel_{\sigma} g_2\). First note that because \(\mathcal{R}\) is an equivalence relation, so is \(\mathcal{R}'\). Also, for \(v \in V\), \(w \in (V_1 \cup V_2)\), \((v, w)_{\mathcal{R}'} = \{v\} \times \{w\}_{\mathcal{R}}\). Now consider the pair \((v_1, v_2)\) \(\in \mathcal{R}'\) and let \(\mu_{i}^{\parallel}\) be the tdf of \(g \parallel_{\sigma} g_i\), \(i = 1, 2\). For \(a \in A\), the only \(a\)-transitions of \((v_1, v_2)\) of non-zero probability are of the form:

1. \(\mu_1^{\parallel}((v, v_1), a, [(v', v_1)]_{\mathcal{R}'})) = \sigma \cdot \mu((v, a, v'))\)
2. \(\mu_2^{\parallel}((v, v_1), a, [(v, v')])_{\mathcal{R}'})) = (1 - \sigma) \cdot \mu_2((v_1, a, [v']_{\mathcal{R}}))\)

Well, we also have that \(\mu_2^{\parallel}((v, v_2), a, [(v', v_2)]_{\mathcal{R}'})) = \sigma \cdot \mu((v, a, v'))\) and, because \(g_1 \equiv_{\text{pr}} g_2\), \(\mu_2^{\parallel}((v, v_2), a, [(v, v')])_{\mathcal{R}}) = (1 - \sigma) \cdot \mu_2((v_1, a, [v']_{\mathcal{R}})).\) The argument is similar in case (1) if \(v_1\) is an endpoint (the value of \(\mu_1^{\parallel}\) would not be weighted by \(\sigma\)), and in case (2) if \(v\) is an endpoint (the value of \(\mu_2^{\parallel}\) would not be weighted by \(1 - \sigma)\).

An argument similar to the above can be used to show that \(\mathcal{R}'\) is also a probabilistic bisimulation between \(g \parallel_{\sigma} g_1\) and \(g \parallel_{\sigma} g_2\). In particular, there are fewer transitions of non-zero probability from \((r_1, r_1)\)and \((r_2, r_2)\) since such transitions can come from \(g\) only. Like in the endpoint cases considered just above, the probabilities of these transitions are not weighted by \(\sigma\).

A nearly symmetric argument establishes that

\[\mathcal{R}'' = \{(v_1, v_2) \mid (v_1, v_2) \in \mathcal{R}, v \in V\}\]

is a probabilistic bisimulation between \(g_1 \parallel_{\sigma} g\) and \(g_2 \parallel_{\sigma} g\).

**Theorem 3.2**

1. \(\text{prF}/ \equiv_{\text{pr}} \models \text{prPA}(A)\)

2. For all closed expressions \(s, t\) over \(\Sigma(\text{prPA}(A))\):

\[\text{prF}/ \equiv_{\text{pr}} \models s = t \iff \text{prPA}(A) \models s = t.\]

**Proof:** For part 1, the soundness of axioms \(\text{prM1} - \text{prM4}\) is immediate by probabilistic process graph isomorphism arguments. The following comments, however, are in order. Axiom \(\text{prM1}\) is a kind of expansion law for probabilistic merge. In \(\text{prM2}\), \(a \parallel_{\sigma} y\) behaves like \(y\) after performing \(a\) as it will have reached a state where \(y\) is in a probabilistic merge with an endpoint. In \(\text{prM3}\), \((a \cdot x) \parallel_{\sigma} y\) behaves like \(x \parallel_{\sigma} y\) after performing \(a\) since left-merge behaves like merge after its root
transitions. The left-hand and right-hand side processes of \(prM4\) both represent a probabilistic merge with \(z\), the first step of which must come from \(x\) (with probability \(\pi\)) or \(y\) (with probability \(1 - \pi\)).

For part 2, the proof is similar to the one given in [BW90] for the completeness of \(PA(A)\). We use the following term rewrite system, with rules corresponding to \(prBPA(A)\) axioms \(prA3-5\) and \(prPA(A)\) axioms \(prM1 - prM4\), to eliminate all occurrences of \(\|_\sigma\) and \(\|_\sigma\) in a closed \(prPA(A)\) term:

\[
\begin{align*}
    x +_\pi z & \rightarrow x \\
    (z +_\pi y) \cdot z & \rightarrow z \cdot z +_\pi y \cdot z \\
    (z \cdot y) \cdot z & \rightarrow z \cdot (y \cdot z) \\
    z \|_\sigma y & \rightarrow z \|_\sigma y +_\pi y \|_{(1-\pi)} z \\
    a \|_\sigma y & \rightarrow a \cdot y \\
    a \cdot z \|_\sigma y & \rightarrow a \cdot (z \|_\sigma y) \\
    (x +_\pi y) \|_\sigma z & \rightarrow (x \|_\sigma z) +_\pi (y \|_\sigma z)
\end{align*}
\]

It can be proved that this term rewriting system is strongly normalizing and that a normal form of a closed term must be a probabilistic basic term. By part 1 of the theorem (the soundness of \(prPA(A)\)) and Theorem 3.1 (the soundness and completeness of \(prBPA(A)\)), the result is proven.

\[ \square \]

### 3.4 Probabilistic ACP

#### 3.4.1 Equational Specification

The signature of \(prACP^+_7(A)\) also extends that of \(prBPA(A)\). Recalling that \(A_\delta = A \cup \delta\), we have:

\[
\Sigma(prACP^+_7(A)) = \Sigma(prBPA(A)) \cup \{ \delta : prP \} \cup \{ I : prP \rightarrow 2^A_\delta \} \cup \\
\{ \|_{\sigma,\theta} : prP \times prP \rightarrow prP \mid \sigma, \theta \in (0,1) \} \cup \{ a, b : prP \rightarrow prP \mid a, b \in (0,1) \} \cup \\
\{ \|_{\sigma,\theta} : prP \times prP \rightarrow prP \mid \sigma, \theta \in (0,1) \} \cup \{ \delta_H : prP \rightarrow prP \mid H \subseteq A \}
\]

Thus, for each of the operators \(\|\), \(\|\), and \(\|\) we have a family of operators, each indexed by two probabilities from the interval \((0,1)\). These operators work intuitively as follows. Consider first the merge operator. In the expression \(z \|_{\sigma,\theta} y\), a communication between \(z\) and \(y\) occurs with probability \(1 - \theta\), and an autonomous move by either \(z\) or \(y\) occurs with probability \(\theta\). Given that an autonomous move occurs, it comes from \(z\) with probability \(\sigma\) and from \(y\) with probability \(1 - \sigma\).

The situation is similar for \(z \|_{\sigma,\theta} y\) except the first step must (with probability 1) come from \(z\). Likewise, the first step of \(z \|_{\sigma,\theta} y\) must result from a communication between \(z\) and \(y\).

The treatment of the communication merge is exactly analogous to the situation in the non-probabilistic case (Section 2.3). The "totality" axiom C0 now becomes:

\[
\forall a, b \in prP \ A_\delta(a) \wedge A_\delta(b) \implies \exists c \in prP \forall \sigma, \theta \in (0,1) A_\delta(c) \wedge a \|_{\sigma,\theta} b = c \quad prC0
\]

The axioms of \(prACP^+_7(A)\) are as follows. In this system, \(a, b, c\) range over \(A_\delta\), \(H_\delta = H \cup \{\delta\}\), and \(I\) has functionality \(I : prP \rightarrow 2^{A_\delta}\). Also, \(\cap, \cup\) are used on \(2^{A_\delta}\) without further specification.

\[
prBPA(A) \quad +
\]

23
\[ \delta \cdot x = \delta \]  

\[ a \mid_{\sigma, \theta} b = b \mid_{(1-\sigma), \theta} a \]  

\[ (a \mid_{\sigma, \theta} b) \mid_{\sigma', \theta'} c = a \mid_{\sigma, \theta} (b \mid_{\sigma', \theta'} c) \]  

\[ \delta \mid_{\sigma, \theta} a = \delta \]  

\[ x \mid_{\sigma, \theta} y = ((x \mid_{\sigma, \theta} y) +_{\sigma} (y \parallel_{(1-\sigma), \theta} x)) +_{\theta} (x \mid_{\sigma, \theta} y) \]  

\[ a \parallel_{\sigma, \theta} y = a \cdot y \]  

\[ (a \cdot x) \parallel_{\sigma, \theta} y = a \cdot (x \parallel_{\sigma, \theta} y) \]  

\[ (x +_{\pi} y) \parallel_{\sigma, \theta} z = (x \parallel_{\sigma, \theta} z) +_{\pi} (y \parallel_{\sigma, \theta} z) \]  

\[ a \mid_{\sigma, \theta} (b \cdot z) = (a \mid_{\sigma, \theta} b) \cdot z \]  

\[ (a \cdot x) \mid_{\sigma, \theta} (b \cdot y) = (a \mid_{\sigma, \theta} b) \cdot (x \parallel_{\sigma, \theta} y) \]  

\[ (x +_{\pi} y) \mid_{\sigma, \theta} z = x \mid_{\sigma, \theta} z +_{\pi} y \mid_{\sigma, \theta} z \]  

\[ x \mid_{\sigma, \theta} (y +_{\pi} z) = x \mid_{\sigma, \theta} y +_{\pi} x \mid_{\sigma, \theta} z \]  

\[ I(a) = \{ a \} \]  

\[ I(x \cdot y) = I(x) \]  

\[ I(x +_{\pi} y) = I(x) \cup I(y) \]  

\[ c \in H \implies \delta_H(a) = \varepsilon \]  

\[ c \in H \implies \delta_H(a) = a \]  

\[ I(x) \subseteq H_\delta \implies \delta_H(x +_{\pi} y) = \delta_H(y) \]  

\[ I(x +_{\pi} y) \cap H_\delta = \emptyset \implies \delta_H(x +_{\pi} y) = \delta_H(x) +_{\pi} \delta_H(y) \]  

\[ \delta_H(x \cdot y) = \delta_H(x) \cdot \delta_H(y) \]  

### 3.4.2 Graph Model

As for \( prBPA(A) \) and \( prPA(A) \), we provide a bisimulation model for \( prACP_\pi(A) \) and prove completeness for finite processes. We begin with the definition of the \( prACP_\pi(A) \) operators on probabilistic process graphs, and for this purpose we need to introduce a "normalization factor" to be used in computing conditional probabilities in a restricted process.
Definition 3.6 Let $g = \langle V, r, \mu \rangle$ be a probabilistic process graph. Then, for $v \in V$, the normalization factor of $v$ with respect to the set of actions $H \subseteq A$ is given by

$$\nu_H(v) = 1 - \sum \{ \mu(v, a, v') \mid a \in H, v' \in V \}$$

Intuitively, $\nu_H(v)$ is the sum of the probabilities of those transitions from $v$ that remain after restricting by the set of actions $H$. In the following, let $\text{initials}(v) = \{ a \in A \mid \exists v' \mu(v, a, v') > 0 \}$ for $v$ a probabilistic process graph node, and let the empty summation of probabilities be 0.

Definition 3.7 The operators $\delta, \|_{\sigma, \delta}, \|_{\sigma, \delta}, \sigma_H, H, H \cap A, I$ are defined on $prF$ as follows:

$\delta$: is given by $\langle \{ r_0 \}, r_1 \rangle$, $r_0, r_1 \in V$, $r_1 \neq r_0$

$$\mu((v_1, v_2), a, (v'_1, v'_2)) = \begin{cases} \sigma \cdot \mu_1(v_1, a, v'_1) & \text{if } \text{endpoint}(v_2) \\ \mu_1(v_1, a, v'_1) & \text{otherwise} \end{cases}$$

$$\mu((v_1, v_2), a, (v'_1, v'_2)) = \begin{cases} (1 - \sigma) \cdot \mu_2(v_2, a, v'_2) & \text{if } \text{endpoint}(v_1) \\ \mu_2(v_2, a, v'_2) & \text{otherwise} \end{cases}$$

$$\mu((v_1, v_2), a, (v'_1, v'_2)) = (1 - \theta) \cdot \sum_{b,c \in \mathcal{A}} \mu_1(v_1, b, v'_1) \cdot \mu_2(v_2, c, v'_2)$$

$\|_{\sigma, \delta} g_2$: is given by $\langle V_1 \times V_2, (r_1, r_2), \mu \rangle$ where for all $a \in A$, $v_1, v'_1, v_2, v'_2 \in V$

- $\mu((r_1, r_2), a, (v'_1, r_2)) = \mu_1(r_1, a, v'_1)$
- if $v_1 \neq r_1$ or $v'_2 \neq r_2$

$$\mu((v_1, v_2), a, (v'_1, v'_2)) = \begin{cases} \sigma \cdot \mu_1(v_1, a, v'_1) & \text{if } \text{endpoint}(v_2) \\ \mu_1(v_1, a, v'_1) & \text{otherwise} \end{cases}$$

$$\mu((v_1, v_2), a, (v'_1, v'_2)) = \begin{cases} (1 - \sigma) \cdot \mu_2(v_2, a, v'_2) & \text{if } \text{endpoint}(v_1) \\ \mu_2(v_2, a, v'_2) & \text{otherwise} \end{cases}$$

$$\mu((v_1, v_2), a, (v'_1, v'_2)) = (1 - \theta) \cdot \sum_{b,c \in \mathcal{A}} \mu_1(v_1, b, v'_1) \cdot \mu_2(v_2, c, v'_2)$$

- if $v'_2 \neq r_2$ $\mu((v_1, r_2), a, (v'_1, v'_2)) = 0$

$g_1 \|_{\sigma, \delta} g_2$: is given by $\langle V_1 \times V_2, (r_1, r_2), \mu \rangle$ where for all $a \in A$,

- $\mu((r_1, r_2), a, (v'_1, r_2)) = \mu_1(r_1, a, v'_1)$
- if $v_1 \neq r_1$ or $v_2 \neq r_2$

$$\mu((v_1, v_2), a, (v'_1, v'_2)) = \begin{cases} \sigma \cdot \mu_1(v_1, a, v'_1) & \text{if } \text{endpoint}(v_2) \\ \mu_1(v_1, a, v'_1) & \text{otherwise} \end{cases}$$

$$\mu((v_1, v_2), a, (v'_1, v'_2)) = \begin{cases} (1 - \sigma) \cdot \mu_2(v_2, a, v'_2) & \text{if } \text{endpoint}(v_1) \\ \mu_2(v_2, a, v'_2) & \text{otherwise} \end{cases}$$

$$\mu((v_1, v_2), a, (v'_1, v'_2)) = (1 - \theta) \cdot \sum_{b,c \in \mathcal{A}} \mu_1(v_1, b, v'_1) \cdot \mu_2(v_2, c, v'_2)$$

- if $(v'_1 \neq r_1$ and $v'_2 = r_2)$ or $(v'_1 = r_1$ and $v'_2 \neq r_2)$ $\mu((r_1, r_2), a, (v'_1, v'_2)) = 0$
\( \delta_H(g_1) \): is given by \( <V_1, r_1, \mu > \) where, for all \( a \in A, v, v' \in V_1 \),

- if \( \text{initials}(v) \subseteq H_\delta \)
  \[ \mu(v, a, v') = 0 \]
  \[ \mu(v, \delta, v') = \sum_{a \in A_\delta} \mu_1(v, a, v') \]
- if \( \text{initials}(v) \not\subseteq H_\delta \)
  \[ \mu(v, a, v') = \begin{cases} 
    0 & \text{if } a \in H_\delta \\
    \mu_1(v, a, v')/\nu_H(v) & \text{otherwise}
  \end{cases} \]

\( I(g_1) \): gives the set of actions \( \text{initials}(r_1) \).

Similar to the case of \( prPA(A) \), the first and third clauses of the definitions of \( g_1 \) \( \rhd_\delta g_2 \) and \( g_1 \rhd_\delta g_2 \) collectively define the transition distribution function \( \mu \) on all transitions from the root \( (r_1, r_2) \). Also note that in the definition of \( \delta H(g_1) \), division by the normalization factor \( \nu_H(v) \) occurs only when \( \text{initials}(v) \not\subseteq H_\delta \), which ensures that \( \nu_H(v) > 0 \).

Processes are still stochastic in the graph model of \( prACP_\delta(A) \) if the probability of \( \delta \)-transitions is taken into account. On the other hand, one may prefer the "substochastic" interpretation that a process like \( a + \frac{1}{2} \delta \) performs an \( a \)-transition (after which it successfully terminates) with probability \( \frac{1}{2} \), but may also do nothing (deadlock) with probability \( \frac{1}{2} \). However, the process \( \delta 0 (a + \frac{1}{2} \delta) \) never deadlocks and is equivalent to \( a \).

The presence of \( \delta \)-edges requires a new definition of probabilistic bisimulation.

**Definition 3.8** Let \( g_1 = < V_1, r_1, \mu_1 > \), \( g_2 = < V_2, r_2, \mu_2 > \) be probabilistic process graphs. A probabilistic \( \delta \)-bisimulation between \( g_1 \) and \( g_2 \) is an equivalence relation \( \mathcal{R} \subseteq (V_1 \cup V_2) \times (V_1 \cup V_2) \) with the following properties:

- \( \mathcal{R}(r_1, r_2) \)
- \( \forall v \in V_1, w \in V_2 \) such that \( \mathcal{R}(v, w) \):
  - \( \forall a \in A, S \in (V_1 \cup V_2)/\mathcal{R}, \mu_1(v, a, S \cap V_1) = \mu_2(w, a, S \cap V_2) \)
  - \( \mu_1(v, \delta, V_1) = \mu_2(w, \delta, V_2) \)

Graphs \( g_1 \) and \( g_2 \) are probabilistically \( \delta \)-bisimilar, written \( g_1 \models^\mathcal{R} g_2 \), if there exists a probabilistic \( \delta \)-bisimulation between \( g_1 \) and \( g_2 \).

The definition is the same as the earlier definition of probabilistic bisimulation except that probabilistically \( \delta \)-bisimilar nodes must perform the action \( \delta \) with the same total probability, without regard to where the \( \delta \)-transitions lead.

In order to prove that \( \models^\mathcal{R} \) is a congruence in \( prACP_\delta(A) \), we need the following proposition to facilitate our reasoning that \( \models^\mathcal{R} \) respects restriction.

**Proposition 3.5** Let \( g_1 \models^\mathcal{R} g_2 \) and let \( \mathcal{R} \) be a probabilistic \( \delta \)-bisimulation between \( g_1 \) and \( g_2 \) with \( (v_1, v_2) \in \mathcal{R} \). Then:

1. \( \mu_1(v_1, a, V_1) = \mu_2(v_2, a, V_2), a \in A_\delta \)}
2. initials($v_1$) = initials($v_2$)

3. $\nu_H(v_1) = \nu_H(v_2)$, $H \subseteq A$.

**Proof:** For $\alpha = \delta$, result (1) is immediate from Definition 3.8. For $\alpha \neq \delta$, (1) is easily deduced from Definition 3.8 as $\mu_1(v_1, a, S \cap V_1) = \mu_2(v_2, a, S \cap V_2)$ for all equivalence classes $S$ of the partition of $V_1 \cup V_2$ induced by $R$. Results (2) and (3) are simple consequences of (1).

**Proposition 3.6** If $g_1 \equiv_\delta g_2$, then $g \parallel_\delta g_1 \equiv_\delta g \parallel_\delta g_2$, $g \parallel_\delta g_1 \equiv_\delta g_2 \parallel_\delta g_1$, $g_1 \parallel_\delta g_2 \equiv_\delta g_2 \parallel_\delta g_1$, $g \parallel_\delta g_1 \equiv_\delta g_\parallel_\delta g_2$ for all $H \subseteq A$, and $I(g_1) = I(g_2)$.

**Proof:** The proof for $\parallel_\delta$ is similar to the proof for $\parallel$ in Proposition 3.4. Let $\alpha \neq \delta$. The $\alpha$-transitions of non-zero probability stemming from $(v, v_1)$ are now of the form:

1. $\mu_1^\parallel((v, v_1), a, [(v', v_1)]_R) = \sigma \cdot \theta \cdot \mu(v, a, v')$
2. $\mu_1^\parallel((v, v_1), a, [(v', v_1)]_R) = (1 - \sigma) \cdot \theta \cdot \mu_1(v_1, a, [v'_1]_R)$
3. $\mu_1^\parallel((v, v_1), a, [(v', v_1)]_R) = (1 - \theta) \cdot \sum_{b, c. b \parallel_\delta c} \sigma \mu(v, b, v') \cdot \mu_1(v_1, c, [v'_1]_R)$
4. $\mu_1^\parallel((v, v_1), \delta, V \times V_1) = \sigma \cdot \theta \cdot \mu(v, \delta, V) + (1 - \theta) \cdot \sum_{b, c. b \parallel_\delta c} \sigma \mu(v, b, V) \cdot \mu_1(v_1, c, V_1) + (1 - \sigma) \cdot \theta \cdot \mu_1(v_1, \delta, V_1)$

The argument for the first two types of transitions is virtually identical to the argument set forth in Proposition 3.4. For the third type, since $g_1 \equiv_\delta g_2$, $\mu_1^\parallel((v, v_1), a, [(v', v_1)]_R) = \mu_2^\parallel((v, v_1), a, [(v', v_1)]_R)$. The arguments for the first three cases collectively are sufficient for the fourth case and we are done.

As in Proposition 3.4, the argument is similar if $v_1$ or $v$ is an endpoint.

Again, as in Proposition 3.4, the proofs for $\parallel_\delta$ and $\parallel_\delta$ follow reasoning similar to, if not simpler than, the proof of $\parallel$. In particular, there are fewer transitions of non-zero probability from $(r, r_1)$ and $(r, r_2)$ since such transitions can come from $g$ only, in the case of probabilistic left-merge, and from communications between $g, g_1$ or $g, g_2$ only, in the case of probabilistic communication merge.

For the case of restriction, assume $g_1 \equiv_\delta g_2$ and let $R$ be a probabilistic $\delta$-bisimulation between $g_1$ and $g_2$. We show that $R$ is also a probabilistic $\delta$-bisimulation between $\partial_H(g_1)$ and $\partial_H(g_2)$, $H \subseteq A$. Let $(v_1, v_2) \in R$ and let $\mu_1^\delta$ be the tdf of $\partial_H(g_1)$, $i = 1, 2$. If initials($v_1$) $\subseteq H_1$, then, by Proposition 3.5. initials($v_2$) $\subseteq H_2$ and therefore $\mu_1^\delta(v_1, \mathcal{E}, V_1)$. $\mu_2^\delta(v_2, \mathcal{E}, V_2) = 1$. Otherwise, $\mu_1^\parallel(v_1, a, V_1), \mu_2^\parallel(v_2, a, V_2) = 0$, if $a \in H_2$; and for all $S \in (V_1 \cup V_2) \mathrm{N}$. $\mu_2^\parallel((v_1, a, S - V_1), \mu_2^\parallel((v_2, a, S - V_2) = \mu_1(v_1, a, S \cap V_1) / \nu_H(v_1)$, if $a \notin H_2$. This last step is a consequence of the fact that $(v_1, v_2) \in R$ and Proposition 3.5. part (3).

That $\equiv_\delta$ respects operator $I$ follows directly from part (3) of Proposition 3.5.

**Theorem 3.3**

1. $\mathsf{prF} \equiv_\delta \mathsf{prACP}_\delta(A)$

2. For all closed expressions $p, q$ over $\Sigma(\mathsf{prACP}_\delta(A))$:

$$\mathsf{prF} \equiv_\delta p = q \implies \mathsf{prACP}_\delta(A) \vdash p = q.$$
Proof: For part 1, the proof of soundness of axiom prA7 is a simple extension of the soundness argument for A7 (Theorem 2.2). Axioms prCl-3 are merely postulated about the communication merge |p,q. The soundness of the rest of the axioms of prACP1(A) rests on probabilistic process graph isomorphism arguments (the remarks given in the soundness part of the proofs of Theorems 2.2 and 3.3 are relevant with the obvious extensions).

Note that the condition to prD3.1 implies that \( \nu_H(r_x) = 0 \) and the condition to prD3.2 implies that \( \nu_H(r_x + s_y) = 1 \) and \( \nu_H(r_x), \nu_H(r_y) = 1 \). The soundness of these axioms now easily follows. As alluded to in Section 2.3, unlike D3.2, prD3.2 is not sound under the weaker condition

\[
I(x) - H_\emptyset \neq \emptyset \text{ and } I(y) - H_\emptyset \neq \emptyset
\]

(for example, consider \( x = a + \frac{1}{2} b, y = c, H = \{a\} \), and \( \pi = \frac{2}{3} \)). This situation is closely related to the fact that the equivalence induced on the stratified model of probabilistic processes via abstraction to the generative model is not a congruence; in particular, it fails to respect restriction [vGSST90].

For part 2, the proof is analogous to the completeness proof of ACP1(A).

- The definition of a probabilistic basic term uses +, instead of +.

- The term rewriting system prRACP1(A) uses the probabilistic counterparts of the rules in RACP1(A) and the normal form is defined analogously as well. For example, prRACP1(A) contains the rule prC0'

\[
a |_{\sigma,\delta} b = c \implies a |_{\sigma,\delta} b \rightarrow c
\]

- The proof that a probabilistic normal form is also a probabilistic basic term proceeds as before – no rule in prRACP1(A) is conditional with respect to any probability.

- prRACP1(A) is strongly normalizing modulo prA1, prA2, prA2': take a prRACP1(A) reduction and erase all probability subscripts. One obtains a valid RACP1(A) reduction.

- The "elimination theorem" for prACP1(A) is also similar. Let \( p \) be a closed prACP1(A) term and let \( \overline{p} \) be the closed ACP1(A) term obtained by erasing all probability subscripts. Now let

\[
p = t_0 \rightarrow t_1 \rightarrow \cdots \rightarrow t_n
\]

be a normalizing reduction of \( \overline{p} \). This reduction can be decorated appropriately with probabilities to obtain a prRACP1(A) normalization of \( p \).

\[ \equiv \]

4 ACP1 as an Abstraction of prACP1

In this section we demonstrate that ACP1(A) can be considered an abstraction of prACP1(A) at both the level of the graph model and at the level of the equational theory. For the former, we exhibit a homomorphism \( \phi \) from probabilistic process graphs to non-probabilistic process graphs that preserves the structure of the bisimulation congruence classes. For the latter, we exhibit a homomorphism \( \Phi \) from prACP1(A) terms to ACP1(A) terms that preserves the validity of equational reasoning.
4.1 Graph Model Homomorphism

The homomorphism $\phi : pr\mathcal{G} \rightarrow \mathcal{G}$, from probabilistic process graphs to non-probabilistic process graphs, simply "forgets" probabilities.

**Definition 4.1** Let $g = \langle V, r, \mu \rangle$ be a probabilistic process graph. Then $\phi(g) = \langle V, r, \rightarrow \rangle$ has the same states and start state as $g$ and $\rightarrow$ is such that

$$v_1 \xrightarrow{a} v_2 \iff \mu(v_1, a, v_2) > 0$$

**Proposition 4.1** Let $g_1, g_2$ be probabilistic process graphs.

$$\phi(a) = a, a \in A_\delta$$

$$\phi(g_1 \cdot_\mu g_2) = \phi(g_1) \cdot \phi(g_2)$$

$$\phi(g_1 +_\mu g_2) = \phi(g_1) + \phi(g_2)$$

$$\phi(g_1 |_\mu g_2) = \phi(g_1) | \phi(g_2)$$

$$\phi(g_1 \parallel_\mu g_2) = \phi(g_1) \parallel \phi(g_2)$$

$$\phi(\hat{\delta}_H(g_1)) = \hat{\delta}_H(\phi(g_1))$$

**Proposition 4.2** The homomorphism $\phi$ preserves the structure of the bisimulation congruence classes. That is,

$$g_1 \equiv_\delta g_2 \implies \phi(g_1) \equiv_\delta \phi(g_2)$$

**Proof:** Let $g_1 = \langle V_1, r_1, \mu_1 \rangle, g_2 = \langle V_2, r_2, \mu_2 \rangle$ be probabilistic process graphs, and let $\phi(g_1) = \langle V_1, r_1, \rightarrow_1 \rangle$ and $\phi(g_2) = \langle V_2, r_2, \rightarrow_2 \rangle$ be their homomorphic images under $\phi$. Further, let $\mathcal{R} \subseteq V_1 \times V_2$ be a $\delta$-probabilistic bisimulation containing $(r_1, r_2)$. That is, $g_1 \equiv_\delta g_2$. Now let $(v, w)$ be an arbitrary pair in $\mathcal{R}$ and assume for some $v' \in V_1, a \in A$ that $\mu_1(v, a, v') > 0$. By Definition 4.1, $v \xrightarrow{a} v'$. Then $\mu_1(v, a, [v']) > 0$ where $[v'] = \{u \in V_1 \cup V_2 | (u, v') \in \mathcal{R} \} \in (V_1 \cup V_2)/\mathcal{R}$. Since $(v, w) \in \mathcal{R}$, then there exists a $w' \in [v']$ with $\mu_2(r_2, a, w') > 0$; i.e., $\mathcal{R}(v', w')$ and, by Definition 4.1 again, $r_2 \xrightarrow{a} w'$. By a symmetric argument and by considering the case $a = \delta$ (which is simpler), we have as desired that $g_2 \equiv_\delta g_2 \implies \phi(g_1) \equiv_\delta \phi(g_2)$.

The converse of this result is clearly not true, e.g., $a + b \equiv_\delta b + a$ but $a + \frac{1}{2} b \not\equiv_\delta \frac{1}{2} b + \frac{1}{2} a$. Thus, the graph model $\mathcal{F}/=_\delta$ of $ACP_\mathcal{F}(A)$ is strictly more abstract than the probabilistic graph model $pr\mathcal{F}/=_\delta$ of $prACP_\mathcal{F}(A)$.

4.2 Equational Theory Homomorphism

Let $L(E)$ be the language of all terms, open and closed, generated by the signature of the equational specification $E$. The homomorphism $\Phi : L(prACP_\mathcal{F}(A)) \rightarrow L(ACP_\mathcal{F}(A))$ from $prACP_\mathcal{F}(A)$ terms to $ACP_\mathcal{F}(A)$ terms, is defined as follows:
\( \Phi(a) = a, a \in A_\delta \)
\( \Phi(x) = x \)
\( \Phi(x \cdot y) = \Phi(x) \cdot \Phi(y) \)
\( \Phi(x \div y) = \Phi(x) + \Phi(y) \)
\( \Phi(z \mid \sigma, \theta y) = \Phi(x) | \Phi(y) \)
\( \Phi(z \parallel \sigma, \theta y) = \Phi(x) \parallel \Phi(y) \)
\( \Phi(\delta_H(x)) = \delta_H(\Phi(x)) \)

The following proposition states that any valid proof of \( \text{prACP}_T(A) \) can be mapped into a valid proof of \( \text{ACP}_T(A) \) using the homomorphism \( \Phi \).

**Proposition 4.3** Let \( t_1, t_2 \) be terms of \( \text{prACP}_T(A) \), i.e., \( t_1, t_2 \in L(\text{prACP}_T(A)) \).

\[
\frac{\text{prACP}_T(A) \vdash t_1 = t_2}{\text{ACP}_T(A) \vdash \Phi(t_1) = \Phi(t_2)}
\]

**Proof:** The proof is by induction on the length of the \( \text{prACP}_T(A) \) proof, using the observation that, for every \( \text{prACP}_T(A) \) axiom of the form \( c \Rightarrow t_1 = t_2 \), its homomorphic image \( \Phi(c) \Rightarrow \Phi(t_1) = \Phi(t_2) \) is an \( \text{ACP}_T(A) \) axiom. Here \( c \) is a possibly empty condition on the validity of the \( \text{prACP}_T(A) \) axiom, and the fact that \( \Phi(c) \) is equal to the condition of the corresponding \( \text{ACP}_T(A) \) axiom means that no axiom of \( \text{prACP}_T(A) \) is conditional on a probability appearing within an \( \text{prACP}_T(A) \) term. \( \square \)

Note that the converse of the result does not hold, e.g., \( a + b = b + a \) but \( a + \frac{1}{2} b \neq b + \frac{1}{2} a \). Thus, \( \text{ACP}_T(A) \) is a strictly more abstract theory than \( \text{prACP}_T(A) \).

### 5 Comments on an Internal Probabilistic Choice Operator

In this section we consider the question whether it is possible to add a probabilistic internal choice operator to \( \text{prACP}_T(A) \). Such an operator \( \lor_p : \text{prP} \times \text{prP} \to \text{prP} \) should have the following properties (similar to \( \cap \) of CSP [HoA85]):

1. \( x \lor_p y \) denotes a process that equals \( x \) with probability \( \pi \) and equals \( y \) with probability \( 1 - \pi \).
2. \( \lor_p \) distributes over the operators of \( \text{prACP}_T(A) \), e.g., for all \( \sqcap, \div, \parallel, [\sigma, \theta, \|\sigma, \theta] \):
   
   \[
   (x \sqcap (y \lor_p z)) = (x \sqcap y) \lor_p (x \sqcap z)
   \]
   
   \[
   (x \lor_p y) \sqcap x = (x \sqcap z) \lor_p (y \sqcap z)
   \]

3. \( z \lor_p y = y \lor_{\pi=1-x} z \) and \( z \lor_p (y \lor_p z) = (z \lor_p (y \lor_{\pi=\rho} z)) \lor_{\pi=\rho=\pi+\rho} z \)

Each of these properties is very plausible. Nevertheless, we observe a difficulty that suggests that the setup with \( \lor_p \) must be flawed. It follows that if an internal probabilistic choice is to be added, at least one of properties (1) - (3) must be removed. But, as stated before, these requirements are needed to simplify any setting simultaneously involving \( +_p \) and \( \lor_p \).

The difficulty with \( \lor_p \) comes about as follows.
Proposition 5.1 \( \text{prACP}_T(A) + (1) - (3) \vdash a \lor \frac{1}{2} b = a \lor \frac{1}{4} (b \lor \frac{1}{2} (a + \frac{1}{2} b)) \)

Proof:

\[
a \lor \frac{1}{2} b = (a \lor \frac{1}{2} b) + \frac{1}{2} (a \lor \frac{1}{2} b)
\]

\[
= \ldots
\]

\[
= a \lor \frac{1}{4} (b \lor \frac{1}{2} (a + \frac{1}{2} b))
\]

Next we introduce a probability measure on traces.

Probabilities of Traces

We define \( \text{Pr} : \text{prP} \times A^* \rightarrow (0,1] \) as follows:

\[
\text{Pr}(z \rightarrow \epsilon) = 1
\]

\[
\text{Pr}(a \rightarrow b) = \begin{cases} 1 & \text{if } a = b \\ 0 & \text{if } a \neq b \end{cases}
\]

\[
\text{Pr}(a \cdot b \cdot c \cdot \sigma) = 0
\]

\[
\text{Pr}(a \cdot z \rightarrow b \cdot \sigma) = \text{Pr}(a \rightarrow b) \cdot \text{Pr}(z \rightarrow \sigma)
\]

\[
\text{Pr}(z \cdot y \rightarrow \sigma) = \pi \cdot \text{Pr}(z \rightarrow \sigma) + (1 - \pi) \cdot \text{Pr}(y \rightarrow \sigma)
\]

\[
\text{Pr}(z \lor y \rightarrow \sigma) = \pi \cdot \text{Pr}(z \rightarrow \sigma) + (1 - \pi) \cdot \text{Pr}(y \rightarrow \sigma).
\]

Given this meaning, it seems clear that one must require:

\[
\text{prACP}_T(A) + (1) - (3) \vdash p = q \quad \forall \sigma \in A^* \quad \text{Pr}(p \rightarrow \sigma) = \text{Pr}(q \rightarrow \sigma)
\]

Now consider the following example:

\[
A = \{a, b, \text{guess}(a), \text{guess}(b), \text{success}(a), \text{success}(b), \text{fail}\}
\]

\[
a_{\sigma, \text{guess}(a)} = \text{success}(a), \forall \sigma, \theta \in (0, 1)
\]

\[
b_{\sigma, \text{guess}(b)} = \text{success}(b), \forall \sigma, \theta \in (0, 1)
\]

\[
a_{\sigma, \text{guess}(b)} = b_{\sigma, \text{guess}(a)} = \text{fail}, \forall \sigma, \theta \in (0, 1)
\]

All other communications are \( \ell \). Let \( H = \{a, b, \text{guess}(a), \text{guess}(b)\} \), and let us write \( \| \) for \( \|_{\frac{1}{2}, \frac{1}{2}} \).

Now, using Proposition 5.1, we find

\[
\text{prACP}_T(A) + (1) - (3) \vdash \theta_H(\text{guess}(a) \| (a \lor \frac{1}{2} b)) = \theta_H(\text{guess}(a) \| a \lor \frac{1}{4} (b \lor \frac{1}{2} (a + \frac{1}{2})))
\]

But

\[
\text{Pr}(\theta_H(\text{guess}(a) \| a \lor \frac{1}{2} b) \rightarrow \text{success}(a))
\]

\[
= \text{Pr}(\theta_H(\text{guess}(a) \| a) \lor \frac{1}{2} \theta_H(\text{guess}(a) \| b) \rightarrow \text{success}(a))
\]

\[
= \text{Pr}(\text{success}(a) \lor \frac{1}{2} \text{fail} \rightarrow \text{success}(a))
\]

\[
= \frac{1}{2}
\]
and

\[ Pr(\partial_H(guess(a)) || a \triangleright (b \triangleright \frac{1}{3} (a + \frac{1}{2} b))) \rightarrow success(a) \]

\[ = \quad Pr(\partial_H(guess(a)) || a) \triangleright \frac{1}{3} (\partial_H(guess(a)) || b) \triangleright \frac{1}{3} \partial_H(guess(a)) || (a + \frac{1}{2} b)) \rightarrow success(a) \]

\[ = \quad Pr(success(a) \triangleright \frac{1}{3} (fail \triangleright \frac{1}{3} success(a)) \rightarrow success(a)) \]

\[ = \quad \frac{1}{4} + \frac{3}{2} \cdot \frac{3}{3} = \frac{3}{4} \]

This calculation indicates a definite problem for combining a probabilistic alternative composition \(+_\pi\) with probabilistic internal choice \(\triangleright\).

It follows that a generalization to a probabilistic setting of CSP that features both composition mechanisms (\(\square\) and \(\triangledown\)) cannot be done along the same lines.

If an internal choice must be added, the authors feel that the mentioned difficulty is best remedied by:

1. adding a sort of state distribution \(SD\) and an embedding \(i : prP \rightarrow SD\) turning a process into a state distribution.
2. Then, \(\triangleright\) can have functionality \(SD \times SD \rightarrow SD\).

6 Conclusions

In this paper, we have presented complete axiomatizations of probabilistic processes within the context of the process algebra ACP. Given that axiom A6 of ACP \((x + \delta = x)\) does not have a plausible interpretation in the generative model of probabilistic computation, we introduced the somewhat weaker theory \(ACP^r\), in which A6 is rejected. \(ACP^r\) is, in essence, a minor alteration of ACP expressing almost the same process identities on finite processes.

Our end-result is the axiom system \(prACP^r\), which can be seen as a probabilistic extension of \(ACP^r\) for generative probabilistic processes. In particular, \(ACP^r\) is homomorphically derivable from \(prACP^r\). As desired, we showed that \(prACP^r\) constitutes a complete axiomatization of Larsen and Skou's probabilistic bisimulation for finite processes.

Several directions for future work can be identified. First, we are interested in adding certain important features to the model, such as recursion and unobservable \(\tau\) actions. Secondly, we desire also to completely axiomatize the reactive and stratified models of probabilistic processes \(vGSSST90\). In the stratified model, which is well-suited for reasoning about probabilistic "fair" scheduling, distinctions are made between processes based on the branching structure of their purely probabilistic choices. We conjecture that by eliminating axiom \(prA2\) (probabilistic alternative composition is not associative in the stratified model!) and weakening the condition to \(prD3.2\) as discussed in the soundness part of the proof of Theorem 3.3, the desired axiomatization can be obtained.

Acknowledgements

The authors gratefully acknowledge Rob van Glabbeek, Chi-Chang Jou, and Bernhard Steffen for valuable discussions.
References


A Proofs of Lemmas 3.1 and 3.2

Lemma 3.1 For any permutation \( \xi \) of \( \{1, \cdots, n\}, n \geq 2, \)

\[
\text{prBPA}(A) \vdash \sum_{i=1}^{n} [\pi_i] z_i = \sum_{i=1}^{n} [\pi_{\xi(i)}] z_{\xi(i)}
\]

Proof: The proof is by induction on \( n \). All non-annotated steps are assumed to follow directly from the definition of summation form notation.

- Basis: \( n = 2 \)
  We prove the non-trivial case where \( \xi(1) = 2, \xi(2) = 1. \)
  
  
  \[
  \sum_{i=1}^{2} [\pi_i] z_i = z_1 + \pi_1 \sum_{i=1}^{1} \left[ \frac{\pi_{i+1}}{1 - \pi_1} \right] z_{i+1} \\
  = z_1 + \pi_1 z_2 \\
  = z_2 + \pi_2 z_1 \quad (\text{prA1}) \\
  = z_2 + \pi_3 \sum_{i=1}^{1} \left[ \frac{\pi_i}{1 - \pi_2} \right] z_i \\
  = \sum_{i=1}^{2} [\pi_{\xi(i)}] z_{\xi(i)}
  \]

- Hypothesis: suppose the lemma holds for \( n \leq k. \)

- Induction: \( n = k + 1 \)
  If \( \xi(1) = 1, \) then we have
  
  \[
  \sum_{i=1}^{k+1} [\pi_i] z_i = z_1 + \pi_1 \sum_{i=1}^{k} \left[ \frac{\pi_{i+1}}{1 - \pi_1} \right] z_{i+1} \\
  = z_1 + \pi_1 \sum_{i=1}^{k} \left[ \frac{\pi_{\xi(i+1)}}{1 - \pi_1} \right] z_{\xi(i+1)} \quad (\text{induction}) \\
  = \sum_{i=1}^{k+1} [\pi_{\xi(i)}] z_{\xi(i)}
  \]

  If \( \xi(1) = j \neq 1, \) then
  
  \[
  \sum_{i=1}^{k+1} [\pi_i] z_i = z_1 + \pi_1 \sum_{i=1}^{k} \left[ \frac{\pi_{i+1}}{1 - \pi_1} \right] z_{i+1} \\
  = z_1 + \pi_1 \sum_{i=1}^{k} \left[ \frac{\pi_{\xi(i+1)}}{1 - \pi_1} \right] z_{\xi'(i+1)} \quad (\text{induction})
  \]

  where \( \xi' \) is any permutation from 2 to \( n + 1 \)
  with \( \xi'(2) = j \)
\[
\begin{align*}
C(i+2) &= x_j + \pi_j \left( \sum_{i=1}^{k-1} \left[ \frac{\pi \xi(i+2)}{1 - \pi_i - \pi_j} \right] x_{\xi(i+2)}(i) \right) \\
&= x_j + \pi_j \left( \sum_{i=1}^{k-1} \left[ \frac{\pi \xi(i+2)}{1 - \pi_i - \pi_j} \right] x_{\xi(i+2)}(i) + 1 - \pi_j \pi_j \right) \quad (prA1) \\
&= \left( x_j + \pi_j \sum_{i=1}^{k-1} \left[ \frac{\pi \xi(i+2)}{1 - \pi_i - \pi_j} \right] x_{\xi(i+2)}(i) \right) + 1 - \pi_j \pi_j \quad (prA2) \\
&= x_j + \pi_j \left( y_1 + \rho_1 \sum_{i=1}^{k-1} \left[ \frac{\pi \xi(i+2)}{1 - \pi_i - \pi_j} \right] y_{\xi(i+2)}(i) \right) \quad (prA1) \\
&= x_j + \pi_j \left( y_1 + \rho_1 \sum_{i=1}^{k-1} \left[ \frac{\pi \xi(i+2)}{1 - \pi_i - \pi_j} \right] y_{\xi'(i+2)} \right) \\
&= x_j + \pi_j \left( y_1 + \rho_1 \sum_{i=1}^{k-1} \left[ \frac{\pi \xi(i+2)}{1 - \pi_i - \pi_j} \right] y_{\xi(i+2)} \right) \\
&= x_j + \pi_j \left( y_1 + \rho_1 \sum_{i=1}^{k-1} \left[ \frac{\pi \xi(i+2)}{1 - \pi_i - \pi_j} \right] y_{\xi'(i+2)} \right) \\
&= x_j + \pi_j \left( y_1 + \rho_1 \sum_{i=1}^{k-1} \left[ \frac{\pi \xi(i+2)}{1 - \pi_i - \pi_j} \right] y_{\xi'(i+2)} \right)
\end{align*}
\]

where \( y_1 = x_1, \rho_1 = \frac{\pi \xi(i+2)}{1 - \pi_j} \)

for \( 1 \leq i \leq k-1, y_{i+1} = x_{\xi(i+2)}(i), \rho_{i+1} = \frac{\pi \xi(i+2)}{1 - \pi_j} \)

\[
= x_j + \pi_j \sum_{i=1}^{k} [\rho_i] y_i
\]

\[
= x_j + \pi_j \sum_{i=1}^{k} \left[ \rho \xi(i) \right] y_{\xi(i)} \quad \text{(induction)}
\]

where \( \xi'' \) is the permutation of 1 to 2 with

\[
y_{\xi''(i)}(i) = x_{\xi(i+1)}(i) \quad \text{and} \quad \rho_{\xi''(i)}(i) = \frac{\pi \xi(i+1)}{1 - \pi_j}
\]

\[
= x_j + \pi_j \sum_{i=1}^{k} \left[ \frac{\pi \xi(i+1)}{1 - \pi_j} \right] x_{\xi(i+1)}
\]

\[
= \sum_{i=1}^{k+1} \left[ \frac{\pi \xi(i)}{1 - \pi_i} \right] x_{\xi(i)}
\]

\[
\square
\]
Lemma 3.2 In the summation form $\sum_{i=1}^{n+1} [\pi_i] z_i$, let $z_1$ and $z_2$ be syntactically identical. Then

$$\text{prBPA}(A) \vdash \sum_{i=1}^{n+1} [\pi_i] z_i = \sum_{i=1}^{n} [\rho_i] y_i$$

where $\rho_1 = \pi_1 + \pi_2$, $y_1 = z_1$, and $\rho_i = \pi_{i+1}$, $y_i = z_{i+1}$, $2 \leq i \leq n$.

Proof: There are two cases; all non-annotated steps are assumed to follow directly from the definition of summation form notation. If $n = 1$, then we have:

$$\sum_{i=1}^{2} [\pi_i] z_i = z_1 + \pi_1 \sum_{i=1}^{1} \left[ \frac{\pi_{i+1}}{1 - \pi_1} \right] z_{i+1}$$

$$= z_1 + \pi_1 \sum_{i=1}^{1} \left[ \frac{1 - \pi_1}{1 - \pi_1} \right] z_1$$

$$= z_1 + \pi_1 z_1$$

$$= \sum_{i=1}^{1} [\rho_i] y_i$$

($prA3$)

If $n \geq 2$, then we have:

$$\sum_{i=1}^{n+1} [\pi_i] z_i = z_1 + \pi_1 \sum_{i=1}^{n} \left[ \frac{\pi_{i+1}}{1 - \pi_1} \right] z_{i+1}$$

$$= z_1 + \pi_1 \left( z_2 + \pi_2 \sum_{i=1}^{n-1} \left[ \frac{\pi_{i+2}}{1 - \pi_1 - \pi_2} \right] z_{i+2} \right)$$

$$= (z_1 + \pi_1, z_2) + \pi_2 \sum_{i=1}^{n-1} \left[ \frac{\pi_{i+2}}{1 - \pi_1 - \pi_2} \right] z_{i+2}$$

($prA2$)

$$= y_2 - \rho_2 \sum_{i=1}^{n-1} \left[ \frac{\pi_{i+2}}{1 - \pi_1 - \pi_2} \right] z_{i+2}$$

($prA3$)

$$= \sum_{i=1}^{n} [\rho_i] y_i$$

($Given\ condition$)
In this series appeared:

Formal methods and tools for the development of distributed and real-time systems, p. 17.

90/2  K.M. van Hee-P.M.P. Rambags  
Dynamic process creation in high-level Petri nets, pp. 19.

90/3  R. Gerth  
Foundations of Compositional Program Refinement - safety properties -, p. 38.

90/4  A. Peeters  
Decomposition of delay-insensitive circuits, p. 25.

90/5  J.A. Brzozowski-J.C. Ebergen  
On the delay-sensitivity of gate networks, p. 23.

90/6  A.J.J.M. Marcelis  

90/7  A.J.J.M. Marcelis  
A logic for one-pass, one-attributed grammars, p. 14.

90/8  M.B. Josephs  
Receptive Process Theory, p. 16.

90/9  A.T.M. Aerts-P.M.E. De Bra-K.M. van Hee  
Combining the functional and the relational model, p. 15.

90/10  M.J. van Diepen-K.M. van Hee  
A formal semantics for Z and the link between Z and the relational algebra, p. 30. (Revised version of CSNotes 89/17).

90/11  P. America-F.S. de Boer  
A proof system for process creation, p. 84.

90/12  P. America-F.S. de Boer  
A proof theory for a sequential version of POOL, p. 110.

Proving termination of Parallel Programs, p. 7.

90/14  F.S. de Boer  
A proof system for the language POOL, p. 70.

90/15  F.S. de Boer  
Compositionality in the temporal logic of concurrent systems, p. 17.

90/16  F.S. de Boer-C. Palamidessi  
A fully abstract model for concurrent logic languages, p. 23.

90/17  F.S. de Boer-C. Palamidessi  
On the asynchronous nature of communication in logic languages: a fully abstract model based on sequences, p. 29.
Design and implementation aspects of remote procedure calls, p. 15.

Two Case Studies in ExSpect, p. 24.

The Nature of Delay-Insensitive Computing, p. 18.

Data, Process and Behaviour Modelling in an integrated specification framework, p. 37.


Implication. A survey of the different logical analyses "if...then...", p. 26.

Parallel Programs for the Recognition of $P$-invariant Segments, p. 16.

Performance Analysis of VLSI Programs, p. 31.

An Implementation Model for GOOD, p. 18.

SPECIFICATIEMETHODEN, een overzicht, p. 20.

CPO-models for second order lambda calculus with recursive types and subtyping, p. 49.

Terminology and Paradigms for Fault Tolerance, p. 25.

Interval Timed Petri Nets and their analysis, p. 53.

POLYNOMIAL RELATORS, p. 52.

Relational Catamorphism, p. 31.


A note on Extensionality, p. 21.

The PDB Hypermedia Package. Why and how it was built, p. 63.
<table>
<thead>
<tr>
<th>Page</th>
<th>Authors</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>91/16</td>
<td>A.I.J.M. Marcelis</td>
<td>An example of proving attribute grammars correct: the representation of arithmetical expressions by DAGs, p. 25.</td>
</tr>
<tr>
<td>91/18</td>
<td>Rik van Geldrop</td>
<td>Transformational Query Solving, p. 35.</td>
</tr>
<tr>
<td>91/19</td>
<td>Erik Poll</td>
<td>Some categorical properties for a model for second order lambda calculus with subtyping, p. 21.</td>
</tr>
<tr>
<td>91/23</td>
<td>K.M. van Hee, L.J. Somers, M. Voorhoeve</td>
<td>Z and high level Petri nets, p. 16.</td>
</tr>
<tr>
<td>91/24</td>
<td>A.T.M. Aerts, D. de Reus</td>
<td>Formal semantics for BRM with examples, p. 25.</td>
</tr>
<tr>
<td>91/25</td>
<td>P. Zhou, J. Hooman, R. Kuiper</td>
<td>A compositional proof system for real-time systems based on explicit clock temporal logic: soundness and completeness, p. 52.</td>
</tr>
<tr>
<td>91/27</td>
<td>F. de Boer, C. Palamidessi</td>
<td>Embedding as a tool for language comparison: On the CSP hierarchy, p. 17.</td>
</tr>
<tr>
<td>91/28</td>
<td>F. de Boer</td>
<td>A compositional proof system for dynamic process creation, p. 24.</td>
</tr>
<tr>
<td>91/30</td>
<td>J.C.M. Baeten, F.W. Vaandrager</td>
<td>An Algebra for Process Creation, p. 29.</td>
</tr>
<tr>
<td>91/31</td>
<td>H. ten Eikelder</td>
<td>Some algorithms to decide the equivalence of recursive types, p. 26.</td>
</tr>
</tbody>
</table>
91/32 P. Struik

91/33 W. v.d. Aalst
The modelling and analysis of queueing systems with QNM-ExSpect, p. 23.

91/34 J. Coenen
Specifying fault tolerant programs in deontic logic, p. 15.

91/35 F.S. de Boer
Asynchronous communication in process algebra, p. 20.

92/01 J. Coenen
A note on compositional refinement, p. 27.

92/02 J. Coenen
A compositional semantics for fault tolerant real-time systems, p. 18.

92/03 J.C.M. Baeten
Real space process algebra, p. 42.

92/04 J.P.H.W.v.d.Eijnde
Program derivation in acyclic graphs and related problems, p. 90.

92/05 J.P.H.W.v.d.Eijnde
Conservative fixpoint functions on a graph, p. 25.

92/06 J.C.M. Baeten
Discrete time process algebra, p.45.

92/07 R.P. Nederpelt
The fine-structure of lambda calculus, p. 110.

92/08 R.P. Nederpelt
On stepwise explicit substitution, p. 30.

92/09 R.C. Backhouse

92/10 P.M.P. Rambags
Composition and decomposition in a CPN model, p. 55.

92/11 R.C. Backhouse
Demonic operators and monotype factors, p. 29.

92/12 F. Kamareddine

92/13 F. Kamareddine
Set theory and nominalisation, Part II, p.22.

92/14 J.C.M. Baeten
The total order assumption, p. 10.

92/15 F. Kamareddine
A system at the cross-roads of functional and logic programming, p.36.

92/16 R.R. Seljée
Integrity checking in deductive databases; an exposition, p.32.

92/17 W.M.P. van der Aalst
Interval timed coloured Petri nets and their analysis, p. 20.
<table>
<thead>
<tr>
<th>Page</th>
<th>Authors</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>92/18</td>
<td>R. Nederpelt</td>
<td>A unified approach to Type Theory through a refined lambda-calculus, p. 30.</td>
</tr>
<tr>
<td></td>
<td>F. Kamareddine</td>
<td></td>
</tr>
<tr>
<td>92/19</td>
<td>J.C.M. Baeten</td>
<td>Axiomatizing Probabilistic Processes:</td>
</tr>
<tr>
<td></td>
<td>J.A. Bergstra</td>
<td>ACP with Generative Probabilities, p. 36.</td>
</tr>
<tr>
<td></td>
<td>S.A. Smolka</td>
<td></td>
</tr>
<tr>
<td>92/20</td>
<td>F. Kamareddine</td>
<td>Are Types for Natural Language? P. 32.</td>
</tr>
<tr>
<td>92/21</td>
<td>F. Kamareddine</td>
<td>Non well-foundedness and type freeness can unify the interpretation of functional application, p. 16.</td>
</tr>
<tr>
<td>92/22</td>
<td>R. Nederpelt</td>
<td>A useful lambda notation, p. 17.</td>
</tr>
<tr>
<td></td>
<td>F. Kamareddine</td>
<td></td>
</tr>
<tr>
<td>92/23</td>
<td>F. Kamareddine</td>
<td>Nominalization, Predication and Type Containment, p. 40.</td>
</tr>
<tr>
<td></td>
<td>E. Klein</td>
<td></td>
</tr>
<tr>
<td>92/24</td>
<td>M. Codish</td>
<td>Bottom-up Abstract Interpretation of Logic Programs, p. 33.</td>
</tr>
<tr>
<td></td>
<td>D. Dams</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Eyal Yardeni</td>
<td></td>
</tr>
<tr>
<td>92/25</td>
<td>E. Poll</td>
<td>A Programming Logic for Fo, p. 15.</td>
</tr>
</tbody>
</table>