Preferential Choice and Coordination Conditions

J.A. Bergstra[1,2] and C.A. Middelburg[3]

[1] Programming Research Group, University of Amsterdam,
P.O. Box 41882, 1009 DB Amsterdam, the Netherlands
janb@science.uva.nl
[2] Department of Philosophy, Utrecht University,
P.O. Box 80126, 3508 TC Utrecht, the Netherlands
janb@phil.uu.nl
[3] Computing Science Department, Eindhoven University of Technology,
P.O. Box 513, 5600 MB Eindhoven, the Netherlands
keesm@win.tue.nl

Abstract. We present a process algebra with conditional expressions of which the conditions concern the enabledness of actions in the context in which a process is placed. With those conditions, it becomes easy to model preferential choices. A preferential choice of a process is a choice whereby certain alternatives are excluded if at least one of the other alternatives is permitted by the context in which the process is placed. Preferential choices are usually modelled rather indirectly using a priority mechanism.

Keywords: coordination conditions, preferential choice, splitting bisimulation, process algebra, Boolean algebras.

1998 CR Categories: D.1.3, D.2.1, D.2.4, F.1.2, F.3.1, F.3.2.

1 Introduction

In [11], we started an investigation into the potentialities of conditional expressions in the setting of the algebraic theory about processes known as ACP [8, 7]. The primary intention of the investigation is to find basic ways to increase expressiveness. The main extensions of ACP introduced in [11] are $\mathrm{ACP}^{\mathrm{c}}$ (ACP with conditional expressions), $\mathrm{ACP}^{\mathrm{cs}}$ ($\mathrm{ACP}^{\mathrm{c}}$ with signals) and $\mathrm{ACP}^{\mathrm{cr}}$ ($\mathrm{ACP}^{\mathrm{c}}$ with retrospective conditions).

In the current paper, we proceed with the investigation started in [11]. We present $\mathrm{ACP}^{\mathrm{cc}}$, a variant of $\mathrm{ACP}^{\mathrm{c}}$ in which the conditions concern the enabledness of actions in the context in which a process is placed. Such conditions are called coordination conditions in this paper because they are primarily intended for coordination of processes that proceed in parallel. Like in $\mathrm{ACP}^{\mathrm{c}}$, the conditions in $\mathrm{ACP}^{\mathrm{cc}}$ are taken from a Boolean algebra. This looks to be important in the case of $\mathrm{ACP}^{\mathrm{cc}}$ where the context in which a process is placed determines whether a condition holds.

The merit of $\mathrm{ACP}^{\mathrm{cc}}$ is primarily the following. If a process has different alternatives to proceed, some may be preferred in the sense that the others should be excluded if at least one of the preferred alternatives is permitted by the context in which the process is placed. Such preferential choices are usually modelled rather indirectly using a priority mechanism, see e.g. [17], but can easily be modelled in $\mathrm{ACP}^{\mathrm{cc}}$. See [4] for an extension of ACP with a priority mechanism. Unease about the solution to model preferential choices with a priority mechanism forms part of our motivation to develop $\mathrm{ACP}^{\mathrm{cc}}$.

We present the main models of $\mathrm{ACP}^{\mathrm{cc}}$ as well. They are based on labelled transition systems of which the labels consist of a condition and an action, called conditional transition systems, and a variant of bisimilarity in which a transition of one of the related transition systems may be simulated by a set of transitions of the other transition system, called splitting bisimilarity. The presented models cover finitely branching processes as well as infinitely branching processes.

We also extend $\mathrm{ACP}^{\mathrm{cc}}$ with a preferential choice operator to show that it is easy to give defining equations for a preferential choice operator in the presence of conditional expressions of which the conditions are coordination conditions. This extension of $\mathrm{ACP}^{\mathrm{cc}}$ can be viewed as an application of $\mathrm{ACP}^{\mathrm{cc}}$ that remains entirely within the domain of process algebra.

The preferential choice operator appears to generalize the priority choice operator added to CCS in [16]. The preferential choice operator is defined for all possible operands, whereas syntactic restrictions are imposed on the operands of the priority choice operator from [16]. Because of those restrictions, the priority choice operator violates the basics of an algebraic approach. Therefore, we consider the preferential choice operator a notable improvement on the priority choice operator.

The structure of this paper is as follows. First of all, we introduce $\mathrm{PA}_\delta^{\mathrm{cc}}$, an important subtheory of $\mathrm{ACP}^{\mathrm{cc}}$ that does not support communication (Section 2). After that, we introduce conditional transition systems and splitting bisimilarity of conditional transition systems (Section 3) and the full splitting bisimulation models of $\mathrm{PA}_\delta^{\mathrm{cc}}$, the main models of $\mathrm{PA}_\delta^{\mathrm{cc}}$ (Section 4). Following this, we have a closer look at splitting bisimilarity based on structural operational semantics (Section 5). Next, we extend $\mathrm{PA}_\delta^{\mathrm{cc}}$ to $\mathrm{ACP}^{\mathrm{cc}}$ (Section 6) and adapt the full splitting bisimulation models of $\mathrm{PA}_\delta^{\mathrm{cc}}$ to full splitting bisimulation models of $\mathrm{ACP}^{\mathrm{cc}}$ (Section 7). Then, we extend $\mathrm{ACP}^{\mathrm{cc}}$ with guarded recursion (Section 8). Thereupon, we extend $\mathrm{ACP}^{\mathrm{cc}}$ with the preferential choice operator (Section 9) and give an example of its use (Section 10). Finally, we make some remarks about related work and mention some options for future work (Section 11).

Some familiarity with Boolean algebras is desirable. The definitions of all notions concerning Boolean algebras that are used in this paper can, for example, be found in [22].

2 PA_δ with Coordination Conditions

PA_δ is a subtheory of ACP that does not support communication (see e.g. [7]). In this section, we present an extension of PA_δ with encapsulation, pre-abstraction and guarded commands, called $\mathrm{PA}_\delta^{\mathrm{cc}}$. Encapsulation was originally incorporated in ACP to encapsulate actions of a process from communication with actions from the outside (see [8]). Pre-abstraction was added to ACP in [5] as a limited kind of abstraction: the actions from which is abstracted are identified, but they remain observable as internal actions.[1] Guarded commands are conditional expressions of the form $\zeta :\rightarrow p$, where ζ and p are expressions representing a condition and a process, respectively. Guarded commands were added to ACP for the first time in [6].

In $\mathrm{PA}_\delta^{\mathrm{cc}}$, just as in PA_δ or ACP extended with pre-abstraction, it is assumed that a fixed but arbitrary finite set of *actions* A, with $\delta \notin \mathsf{A}$ and $t \in \mathsf{A}$, has been given. Action t is the internal action that replaces all occurrences of actions from which is abstracted by means of pre-abstraction. Henceforth, we write A_δ for $\mathsf{A} \cup \{\delta\}$.

If it is permitted by the context in which a process is placed to perform an action a, we say that a is enabled in that context. In $\mathrm{PA}_\delta^{\mathrm{cc}}$, we consider conditions concerning the enabledness of actions. More precisely, conditions are taken from the quotient algebra of the free Boolean algebra over $\{\mathcal{E}_a \mid a \in \mathsf{A}\}$ by the equivalence induced by the equation $\mathcal{E}_t = \top$. Henceforth, we write $\mathcal{C}$ for this algebra and also for the domain of this algebra. The intuition is that $\mathcal{E}_a$ holds if action a is enabled in the context present. Because it is internal, the action t is enabled in any context. The elements of $\mathcal{C}$ are called *coordination* conditions.

The algebraic theory $\mathrm{PA}_\delta^{\mathrm{cc}}$ has two sorts:

- the sort $\mathbf{P}$ of *processes*;
- the sort $\mathbf{C}$ of *conditions*.

The algebraic theory $\mathrm{PA}_\delta^{\mathrm{cc}}$ has the following constants and operators to build terms of sort $\mathbf{P}$:

- the *deadlock* constant $\delta : \mathbf{P}$;
- for each $a \in \mathsf{A}$, the *action* constant $a : \mathbf{P}$;
- the binary *alternative composition* operator $+ : \mathbf{P} \times \mathbf{P} \rightarrow \mathbf{P}$;
- the binary *sequential composition* operator $\cdot : \mathbf{P} \times \mathbf{P} \rightarrow \mathbf{P}$;
- the binary *guarded command* operator $:\rightarrow : \mathbf{C} \times \mathbf{P} \rightarrow \mathbf{P}$;
- the binary *parallel composition* operator $\parallel : \mathbf{P} \times \mathbf{P} \rightarrow \mathbf{P}$;
- the binary *left merge* operator $⌊\!\!⌊ : \mathbf{P} \times \mathbf{P} \rightarrow \mathbf{P}$;
- for each $H \subseteq \mathsf{A} \setminus \{t\}$, the unary *encapsulation* operator $\partial_H : \mathbf{P} \rightarrow \mathbf{P}$;
- for each $I \subseteq \mathsf{A}$, the unary *pre-abstraction* operator $t_I : \mathbf{P} \rightarrow \mathbf{P}$.

The algebraic theory $\mathrm{PA}_\delta^{\mathrm{cc}}$ has the following constants and operators to build terms of sort $\mathbf{C}$:

- the *bottom* constant $\bot : \mathbf{C}$;
- the *top* constant $\top : \mathbf{C}$;
- for each $a \in \mathsf{A}$, the *enabledness* constant $\mathcal{E}_a : \mathbf{C}$;
- the unary *complement* operator $- : \mathbf{C} \rightarrow \mathbf{C}$;

[1] This limited kind of abstraction was not given a name in [5]. The name pre-abstraction originates from [1].

– the binary *join* operator $\sqcup : \mathbf{C} \times \mathbf{C} \to \mathbf{C}$;
– the binary *meet* operator $\sqcap : \mathbf{C} \times \mathbf{C} \to \mathbf{C}$;
– for each $H \subseteq \mathsf{A} \setminus \{t\}$, the unary *encapsulation* operator $\partial_H : \mathbf{C} \to \mathbf{C}$;
– for each $I \subseteq \mathsf{A}$, the unary *pre-abstraction* operator $t_I : \mathbf{C} \to \mathbf{C}$.

We use infix notation for the binary operators. The following precedence conventions are used to reduce the need for parentheses. The operators to build terms of sort $\mathbf{C}$ bind stronger than the operators to build terms of sort $\mathbf{P}$. The operator $\cdot$ binds stronger than all other binary operators to build terms of sort $\mathbf{P}$ and the operator $+$ binds weaker than all other binary operators to build terms of sort $\mathbf{P}$.

Let p and q be closed terms of sort $\mathbf{P}$, ζ and ξ be closed term of sort $\mathbf{C}$, $a \in \mathsf{A}$, $H \subseteq \mathsf{A} \setminus \{t\}$, and $I \subseteq \mathsf{A}$. Intuitively, the constants and operators to build terms of sort $\mathbf{P}$ can be explained as follows:

– δ can neither perform an action nor terminate successfully;
– a first performs action a unconditionally and then terminates successfully;
– $p + q$ behaves either as p or as q, but not both;
– $p \cdot q$ first behaves as p, but when p terminates successfully it continues by behaving as q;
– $\zeta :\to p$ behaves as p under condition ζ;
– $p \parallel q$ behaves as the process that proceeds with p and q in parallel;
– $p \mathbin{⌊\!⌊} q$ behaves the same as $p \parallel q$, except that it starts with performing an action of p;
– $\partial_H(p)$ behaves the same as p, except that actions from H are blocked.
– $t_I(p)$ behaves the same as p, except that it performs the internal action t whenever p would perform an action in I.

Intuitively, the constants and operators to build terms of sort $\mathbf{C}$ can be explained as follows:

– $\mathcal{E}_a$ is a condition that holds if action a is enabled in the context present;
– $\bot$ is a condition that never holds;
– $\top$ is a condition that always holds;
– $-\zeta$ is the opposite of ζ;
– $\zeta \sqcup \xi$ is either ζ or ξ;
– $\zeta \sqcap \xi$ is both ζ and ξ;
– $\partial_H(\zeta)$ is ζ with all enabledness conditions $\mathcal{E}_a$ with $a \in H$ replaced by $\bot$;
– $t_I(\zeta)$ is ζ with all enabledness conditions $\mathcal{E}_a$ with $a \in I$ replaced by $\top$.

Some earlier extensions of ACP include conditional expressions of the form $p \lhd \zeta \rhd q$; see e.g. [2]. This notation with triangles originates from [19]. We treat conditional expressions of the form $p \lhd \zeta \rhd q$, where p and q are terms of sort $\mathbf{P}$ and ζ is a term of sort $\mathbf{C}$, as abbreviations. That is, we write $p \lhd \zeta \rhd q$ for $\zeta :\to p + -\zeta :\to q$.

We will use the sum notation $\sum_{i \in \mathcal{I}} p_i$, where $\mathcal{I} = \{i_1, \ldots, i_n\}$ and $p_{i_1}, \ldots, p_{i_n}$ are terms of sort $\mathbf{P}$, for $p_{i_1} + \ldots + p_{i_n}$. The convention is that $\sum_{i \in \mathcal{I}} p_i$ stands for δ if $\mathcal{I} = \emptyset$.

Table 1. Axioms of Boolean algebras

$\phi \sqcup \bot = \phi$	BA1	$\phi \sqcap \top = \phi$	BA5
$\phi \sqcup -\phi = \top$	BA2	$\phi \sqcap -\phi = \bot$	BA6
$\phi \sqcup \psi = \psi \sqcup \phi$	BA3	$\phi \sqcap \psi = \psi \sqcap \phi$	BA7
$\phi \sqcup (\psi \sqcap \chi) = (\phi \sqcup \psi) \sqcap (\phi \sqcup \chi)$	BA4	$\phi \sqcap (\psi \sqcup \chi) = (\phi \sqcap \psi) \sqcup (\phi \sqcap \chi)$	BA8

Table 2. Axioms of $\mathrm{PA}_\delta^{\mathrm{cc}}$

$x + y = y + x$		A1	$\top :\rightarrow x = x$		GC1
$(x + y) + z = x + (y + z)$		A2	$\bot :\rightarrow x = \delta$		GC2
$x + x = x$		A3	$\phi :\rightarrow \delta = \delta$		GC3
$(x + y) \cdot z = x \cdot z + y \cdot z$		A4	$\phi :\rightarrow (x + y) = \phi :\rightarrow x + \phi :\rightarrow y$		GC4
$(x \cdot y) \cdot z = x \cdot (y \cdot z)$		A5	$\phi :\rightarrow x \cdot y = (\phi :\rightarrow x) \cdot y$		GC5
$x + \delta = x$		A6	$\phi :\rightarrow (\psi :\rightarrow x) = (\phi \sqcap \psi) :\rightarrow x$		GC6
$\delta \cdot x = \delta$		A7	$(\phi \sqcup \psi) :\rightarrow x = \phi :\rightarrow x + \psi :\rightarrow x$		GC7
			$(\phi :\rightarrow x) \lfloor\!\lfloor y = \phi :\rightarrow (x \lfloor\!\lfloor y)$		GC8
			$\partial_H(\phi :\rightarrow x) = \partial_H(\phi) :\rightarrow \partial_H(x)$		GC11E
$x \parallel y = x \lfloor\!\lfloor y + y \lfloor\!\lfloor x$		M1	$t_I(\phi :\rightarrow x) = t_I(\phi) :\rightarrow t_I(x)$		GC12E
$a \lfloor\!\lfloor x = a \cdot x$		M2			
$a \cdot x \lfloor\!\lfloor y = a \cdot (x \parallel y)$		M3	$\partial_H(\bot) = \bot$		ED1
$(x + y) \lfloor\!\lfloor z = x \lfloor\!\lfloor z + y \lfloor\!\lfloor z$		M4	$\partial_H(\top) = \top$		ED2
			$\partial_H(\mathcal{E}_a) = \mathcal{E}_a$	if $a \notin H$	ED3
$\partial_H(a) = a$	if $a \notin H$	D1	$\partial_H(\mathcal{E}_a) = \bot$	if $a \in H$	ED4
$\partial_H(a) = \delta$	if $a \in H$	D2	$\partial_H(-\phi) = -\partial_H(\phi)$		ED5
$\partial_H(x + y) = \partial_H(x) + \partial_H(y)$		D3	$\partial_H(\phi \sqcup \psi) = \partial_H(\phi) \sqcup \partial_H(\psi)$		ED6
$\partial_H(x \cdot y) = \partial_H(x) \cdot \partial_H(y)$		D4	$\partial_H(\phi \sqcap \psi) = \partial_H(\phi) \sqcap \partial_H(\psi)$		ED7
$t_I(a) = a$	if $a \notin I$	I1	$t_I(\bot) = \bot$		EI1
$t_I(a) = t$	if $a \in I$	I2	$t_I(\top) = \top$		EI2
$t_I(x + y) = t_I(x) + t_I(y)$		I3	$t_I(\mathcal{E}_a) = \mathcal{E}_a$	if $a \notin I$	EI3
$t_I(x \cdot y) = t_I(x) \cdot t_I(y)$		I4	$t_I(\mathcal{E}_a) = \top$	if $a \in I$	EI4
			$t_I(-\phi) = -t_I(\phi)$		EI5
			$t_I(\phi \sqcup \psi) = t_I(\phi) \sqcup t_I(\psi)$		EI6
$\mathcal{E}_t = \top$		ET	$t_I(\phi \sqcap \psi) = t_I(\phi) \sqcap t_I(\psi)$		EI7

The axioms of $\mathrm{PA}_\delta^{\mathrm{cc}}$ are the axioms of Boolean Algebras (BA) given in Table 1 and the additional axioms given in Table 2. Several axioms given in Table 2 are actually axiom schemas: a and b stand for arbitrary constants of sort **P** (i.e. $a, b \in \mathsf{A}_\delta$), H stands for an arbitrary subset of $\mathsf{A} \setminus \{t\}$, and I stands for an arbitrary subset of A.

The axioms of BA given in Table 1 have been taken from [18]. Several alternatives for this axiomatization can be found in the literature. If we use basic laws of BA other than axioms BA1–BA8 in a step of a derivation, we will refer to them as applications of BA and not give their derivation from axioms BA1–BA8. The terms of sort **C** are interpreted in $\mathcal{C}$ as usual.

The axioms of $\mathrm{PA}_\delta^{\mathrm{cc}}$ include the axioms of PA_δ, the usual axioms for encapsulation and the usual axioms for pre-abstraction (see e.g. [1]). Axioms GC1–GC7 have been taken from [2], but with the axiom $x \cdot z \lhd \phi \rhd y \cdot z = (x \lhd \phi \rhd y) \cdot z$ (CO5) replaced by $\phi :\to x \cdot y = (\phi :\to x) \cdot y$ (GC5).

An interesting subtheory of $\mathrm{PA}_\delta^{\mathrm{cc}}$ is $\mathrm{BPA}_\delta^{\mathrm{cc}}$. This subtheory is obtained by removing the parallel composition operator, the left merge operator, the encapsulation operators and the pre-abstraction operators from the signature of $\mathrm{PA}_\delta^{\mathrm{cc}}$ and removing all axioms in which these operator occur from the axioms of $\mathrm{PA}_\delta^{\mathrm{cc}}$ – in other words, the axioms of $\mathrm{BPA}_\delta^{\mathrm{cc}}$ are BA1–BA8, A1–A7, GC1–GC7, ET.

To prove a statement for all closed terms of sort **P** from the language of $\mathrm{BPA}_\delta^{\mathrm{cc}}$, it is sufficient to prove it for all basic terms. The set $\mathcal{B}$ of *basic terms* is inductively defined by the following rules:

- $\delta \in \mathcal{B}$;
- if ζ is a closed term of sort **C** and $a \in \mathsf{A}$, then $\zeta :\to a \in \mathcal{B}$;
- if ζ is a closed term of sort **C**, $a \in \mathsf{A}$ and $p \in \mathcal{B}$, then $\zeta :\to a \cdot p \in \mathcal{B}$;
- if $p, q \in \mathcal{B}$, then $p + q \in \mathcal{B}$.

The basic terms are exactly the closed terms of sort **P** from the language of $\mathrm{BPA}_\delta^{\mathrm{cc}}$ of the form

$$\sum_{i<n} \zeta_i :\to a_i \cdot p_i + \sum_{i<m} \xi_i :\to b_i \ ,$$

where $a_0, \ldots, a_{n-1}, b_0, \ldots, b_{m-1} \in \mathsf{A}$, $\zeta_0, \ldots, \zeta_{n-1}, \xi_0, \ldots, \xi_{m-1}$ are closed term of sort **C** and $p_0, \ldots, p_{n-1}$ are basics terms ($n, m \geq 0$). We can prove that all closed terms of sort **P** from the language of $\mathrm{BPA}_\delta^{\mathrm{cc}}$ are derivably equal to a basic term.

Lemma 2.1 (Elimination for $\mathbf{BPA}_\delta^{\mathbf{cc}}$). *For all closed terms p of sort* **P** *from the language of* $\mathrm{BPA}_\delta^{\mathrm{cc}}$*, there exists a basic term $q \in \mathcal{B}$ such that* $\mathrm{BPA}_\delta^{\mathrm{cc}} \vdash p = q$.

Proof. The term rewriting system consisting of axioms A4–A7 and GC1–GC7 oriented from left to right is strongly normalizing. This can be proved by using the method of lexicographical path ordering of Kamin and Lévy (see e.g. [20]), making the signature one-sorted, taking the ordering $:\to \; > \cdot > +$, $:\to \; > \delta, \sqcap$, and giving the lexicographical status for the first argument to $\cdot$ and the lexicographical status for the second argument to $:\to$. Moreover, it is easy to see that each normal form is a basic term. □

We can also prove that all closed terms of sort **P** from the language of $\mathrm{PA}_\delta^{\mathrm{cc}}$ are derivably equal to a basic term.

Lemma 2.2 (Elimination for $\mathbf{PA}_\delta^{\mathrm{cc}}$). *For all closed terms p of sort $\mathbf{P}$ from the language of $\mathrm{PA}_\delta^{\mathrm{cc}}$, there exists a basic term $q \in \mathcal{B}$ such that $\mathrm{PA}_\delta^{\mathrm{cc}} \vdash p = q$.*

Proof. The term rewriting system consisting of all axioms of $\mathrm{PA}_\delta^{\mathrm{cc}}$, except BA1–BA8 and A1–A3, oriented from left to right is strongly normalizing. This can be proved by using the method of lexicographical path ordering of Kamin and Lévy, making the signature one-sorted, ranking the operators $\|$ and $\lfloor\!\lfloor$ as in [9], taking the ordering $\ldots > \|_3 > \lfloor\!\lfloor_3 > \|_2 > \lfloor\!\lfloor_2 > :\!\rightarrow > \cdot > +$, $:\!\rightarrow > \delta, \sqcap$, $\partial_H > :\!\rightarrow, \bot, -, \sqcup$ for all $H \subseteq \mathsf{A} \setminus \{t\}$, $t_I > :\!\rightarrow, t, \top, -, \sqcup$ for all $I \subseteq \mathsf{A}$, and giving the lexicographical status for the first argument to $\cdot$ and the lexicographical status for the second argument to $:\!\rightarrow$. Moreover, it is easy to see that each normal from is a basic term. □

Moreover, we can prove that $\mathrm{PA}_\delta^{\mathrm{cc}}$ is a conservative extension of $\mathrm{BPA}_\delta^{\mathrm{cc}}$.

Lemma 2.3 (Conservative extension). *If p and q are closed terms of sort $\mathbf{P}$ from the language of $\mathrm{BPA}_\delta^{\mathrm{cc}}$, then $\mathrm{BPA}_\delta^{\mathrm{cc}} \vdash p = q$ iff $\mathrm{PA}_\delta^{\mathrm{cc}} \vdash p = q$.*

Proof. The implication from left to right follows immediately from the fact that the axioms of $\mathrm{BPA}_\delta^{\mathrm{cc}}$ are included in the axioms of $\mathrm{PA}_\delta^{\mathrm{cc}}$. The implication from right to left is proved as follows. Let p and q be closed terms of sort $\mathbf{P}$ from the language of $\mathrm{BPA}_\delta^{\mathrm{cc}}$ such that $\mathrm{PA}_\delta^{\mathrm{cc}} \vdash p = q$. Because it is left-linear and non-overlapping (see e.g. [20]), the term rewriting system used in the proof of Lemma 2.2 is confluent modulo axioms A1–A3. Consequently, the reductions of p and q by means of this term rewriting system yields the same normal form modulo axioms A1–A3. Moreover, the reductions of p and q only use axioms A4–A7 and GC1–GC7, oriented from left to right, because the additional operators of $\mathrm{PA}_\delta^{\mathrm{cc}}$ do not occur in p and q, and no rewrite rule introduces occurrences of those operators that were not already there in its left-hand side. Hence, the reduction of p into its normal form followed by the reverse of the reduction of q into its normal form is a proof of $p = q$ in $\mathrm{BPA}_\delta^{\mathrm{cc}}$. □

The preceding three lemmas will be useful in the completeness proof of $\mathrm{PA}_\delta^{\mathrm{cc}}$ for the full splitting bisimulation models of $\mathrm{PA}_\delta^{\mathrm{cc}}$ that will be introduced in Section 4.

We proceed to the presentation of the structural operational semantics of $\mathrm{PA}_\delta^{\mathrm{cc}}$. The following relations on closed terms of sort $\mathbf{P}$ from the language of $\mathrm{PA}_\delta^{\mathrm{cc}}$ are used:

- for each $\ell \in (\mathcal{C} \setminus \{\bot\}) \times \mathsf{A}$, a binary relation $\xrightarrow{\ell}$;
- for each $\ell \in (\mathcal{C} \setminus \{\bot\}) \times \mathsf{A}$, a unary relation $\xrightarrow{\ell} \surd$.

We write $p \xrightarrow{[\alpha]\,a} q$ instead of $(p, q) \in \xrightarrow{(\alpha,a)}$ and $p \xrightarrow{[\alpha]\,a} \surd$ instead of $p \in \xrightarrow{(\alpha,a)} \surd$. The relations $\xrightarrow{\ell} \surd$ and $\xrightarrow{\ell}$ can be explained as follows:

- $p \xrightarrow{[\alpha]\,a} \surd$: p is capable of performing action a under condition α and then terminating successfully;
- $p \xrightarrow{[\alpha]\,a} q$: p is capable of performing action a under condition α and then proceeding as q.

Table 3. Transition rules for $\mathrm{PA}_{\delta}^{\mathrm{cc}}$

$$\frac{}{a \xrightarrow{[\top]\,a} \surd}$$

$$\frac{x \xrightarrow{[\phi]\,a} \surd}{x + y \xrightarrow{[\phi]\,a} \surd} \quad \frac{y \xrightarrow{[\phi]\,a} \surd}{x + y \xrightarrow{[\phi]\,a} \surd} \quad \frac{x \xrightarrow{[\phi]\,a} x'}{x + y \xrightarrow{[\phi]\,a} x'} \quad \frac{y \xrightarrow{[\phi]\,a} y'}{x + y \xrightarrow{[\phi]\,a} y'}$$

$$\frac{x \xrightarrow{[\phi]\,a} \surd}{x \cdot y \xrightarrow{[\phi]\,a} y} \quad \frac{x \xrightarrow{[\phi]\,a} x'}{x \cdot y \xrightarrow{[\phi]\,a} x' \cdot y}$$

$$\frac{x \xrightarrow{[\phi]\,a} \surd}{\psi :\to x \xrightarrow{[\phi \sqcap \psi]\,a} \surd}\ \phi \sqcap \psi \neq \bot \quad \frac{x \xrightarrow{[\phi]\,a} x'}{\psi :\to x \xrightarrow{[\phi \sqcap \psi]\,a} x'}\ \phi \sqcap \psi \neq \bot$$

$$\frac{x \xrightarrow{[\phi]\,a} \surd}{x \parallel y \xrightarrow{[\phi]\,a} y} \quad \frac{y \xrightarrow{[\phi]\,a} \surd}{x \parallel y \xrightarrow{[\phi]\,a} x} \quad \frac{x \xrightarrow{[\phi]\,a} x'}{x \parallel y \xrightarrow{[\phi]\,a} x' \parallel y} \quad \frac{y \xrightarrow{[\phi]\,a} y'}{x \parallel y \xrightarrow{[\phi]\,a} x \parallel y'}$$

$$\frac{x \xrightarrow{[\phi]\,a} \surd}{x \mathbin{\|\!\_} y \xrightarrow{[\phi]\,a} y} \quad \frac{x \xrightarrow{[\phi]\,a} x'}{x \mathbin{\|\!\_} y \xrightarrow{[\phi]\,a} x' \parallel y}$$

$$\frac{x \xrightarrow{[\phi]\,a} \surd}{\partial_H(x) \xrightarrow{[\partial_H(\phi)]\,a} \surd}\ a \notin H,\, \partial_H(\phi) \neq \bot \quad \frac{x \xrightarrow{[\phi]\,a} x'}{\partial_H(x) \xrightarrow{[\partial_H(\phi)]\,a} \partial_H(x')}\ a \notin H,\, \partial_H(\phi) \neq \bot$$

$$\frac{x \xrightarrow{[\phi]\,a} \surd}{t_I(x) \xrightarrow{[t_I(\phi)]\,a} \surd}\ a \notin I,\, t_I(\phi) \neq \bot \quad \frac{x \xrightarrow{[\phi]\,a} x'}{t_I(x) \xrightarrow{[t_I(\phi)]\,a} t_I(x')}\ a \notin I,\, t_I(\phi) \neq \bot$$

$$\frac{x \xrightarrow{[\phi]\,a} \surd}{t_I(x) \xrightarrow{[t_I(\phi)]\,t} \surd}\ a \in I,\, t_I(\phi) \neq \bot \quad \frac{x \xrightarrow{[\phi]\,a} x'}{t_I(x) \xrightarrow{[t_I(\phi)]\,t} t_I(x')}\ a \in I,\, t_I(\phi) \neq \bot$$

The structural operational semantics of $\mathrm{PA}_{\delta}^{\mathrm{cc}}$ is described by the transition rules given in Table 3. We will return to this structural operational semantics in Section 5.

3 Transition Systems and Splitting Bisimilarity for $\mathrm{PA}_{\delta}^{\mathrm{cc}}$

In this section, we introduce conditional transition systems and splitting bisimilarity of conditional transition systems. In Section 4, we will make use of conditional transition systems and splitting bisimilarity of conditional transition systems to construct models of $\mathrm{PA}_{\delta}^{\mathrm{cc}}$. In Section 5, we will show that the structural operational semantics presented in Section 2 induces a conditional transition system for each closed term of sort **P** from the language of $\mathrm{PA}_{\delta}^{\mathrm{cc}}$.

Conditional transition systems are labelled transition systems of which the labels consist of a condition different from $\bot$ and an action. Labels of this kind are sometimes called *guarded actions*. Henceforth, we write $\mathcal{C}^-$ for $\mathcal{C} \setminus \{\bot\}$.

A *conditional transition system* T consists of the following:

- a set S of *states*;

– a set $\xrightarrow{\ell} \subseteq S \times S$, for each $\ell \in \mathcal{C}^- \times \mathsf{A}$;
– a set $\xrightarrow{\ell}\surd \subseteq S$, for each $\ell \in \mathcal{C}^- \times \mathsf{A}$;
– an *initial state* $s^0 \in S$.

If $(s, s') \in \xrightarrow{\ell}$ for some $\ell \in \mathcal{C}^- \times \mathsf{A}$, then we say that there is a *transition* from s to s'. We usually write $s \xrightarrow{[\alpha]\,a} s'$ instead of $(s, s') \in \xrightarrow{(\alpha,a)}$ and $s \xrightarrow{[\alpha]\,a} \surd$ instead of $s \in \xrightarrow{(\alpha,a)} \surd$. Furthermore, we write $\rightarrow$ for the family of sets $(\xrightarrow{\ell})_{\ell \in \mathcal{C}^- \times \mathsf{A}}$ and $\rightarrow\surd$ for the family of sets $(\xrightarrow{\ell}\surd)_{\ell \in \mathcal{C}^- \times \mathsf{A}}$.

The relations $\xrightarrow{\ell}\surd$ and $\xrightarrow{\ell}$ can be explained as follows:

– $s \xrightarrow{[\alpha]\,a} \surd$: in state s, it is possible to perform action a under condition α, and by doing so to terminate successfully;
– $s \xrightarrow{[\alpha]\,a} s'$: in state s, it is possible to perform action a under condition α, and by doing so to make a transition to state s'.

A conditional transition system may have states that are not reachable from its initial state by a number of transitions. Connected conditional transition systems are transition systems without unreachable states.

Let $T = (S, \rightarrow, \rightarrow\surd, s^0)$ be a conditional transition system. Then the *reachability* relation of T is the smallest relation $\twoheadrightarrow \subseteq S \times S$ such that:

– $s \twoheadrightarrow s$;
– if $s \xrightarrow{\ell} s'$ and $s' \twoheadrightarrow s''$, then $s \twoheadrightarrow s''$.

We write $\mathrm{RS}(T)$ for $\{s \in S \mid s^0 \twoheadrightarrow s\}$. T is called a *connected* conditional transition system if $S = \mathrm{RS}(T)$. Henceforth, we will only consider connected conditional transition systems. This often calls for extraction of the connected part of a conditional transition system that is composed of connected conditional transition systems.

Let $T = (S, \rightarrow, \rightarrow\surd, s^0)$ be a conditional transition system that is not necessarily connected. Then the *connected part* of T, written $\Gamma(T)$, is defined as follows:

$$\Gamma(T) = (S', \rightarrow', \rightarrow\surd', s^0) \;,$$

where

$$S' = \mathrm{RS}(T) \;,$$

and for every $\ell \in \mathcal{C}^- \times \mathsf{A}$:

$$\begin{aligned} \xrightarrow{\ell}{}' \;&= \xrightarrow{\ell} \cap\, (S' \times S') \;, \\ \xrightarrow{\ell}\surd' &= \xrightarrow{\ell}\surd \cap S' \;. \end{aligned}$$

It is assumed that for each infinite cardinal κ a fixed but arbitrary set $\mathcal{S}_\kappa$ with the following properties has been given:

– the cardinality of $\mathcal{S}_\kappa$ is greater than or equal to κ;

– if $S_1, S_2 \subseteq \mathcal{S}_\kappa$, then $S_1 \uplus S_2 \subseteq \mathcal{S}_\kappa$ and $S_1 \times S_2 \subseteq \mathcal{S}_\kappa$.[2]

Let κ be an infinite cardinal. Then $\mathbb{CTS}_\kappa$ is the set of all connected conditional transition systems $T = (S, \rightarrow, \rightarrow\surd, s^0)$ such that $S \subset \mathcal{S}_\kappa$ and the branching degree of T is less than κ, i.e. for all $s \in S$, the cardinality of the set $\{(\ell, s') \in (\mathcal{C}^- \times \mathsf{A}) \times S \mid (s, s') \in \xrightarrow{\ell}\} \cup \{\ell \in \mathcal{C}^- \times \mathsf{A} \mid s \in \xrightarrow{\ell}\surd\}$ is less than κ.[3]

The condition $S \subset \mathcal{S}_\kappa$ guarantees that $\mathbb{CTS}_\kappa$ is indeed a set.

A conditional transition system is said to be *finitely branching* if its branching degree is less than $\aleph_0$. Otherwise, it is said to be *infinitely branching*.

Conditional transition system that differ only with respect to the identity of the states are isomorphic.

Let $T_1 = (S_1, \rightarrow_1, \rightarrow\surd_1, s_1^0)$ and $T_2 = (S_2, \rightarrow_2, \rightarrow\surd_2, s_2^0)$ be conditional transition systems. Then T_1 and T_2 are *isomorphic*, written $T_1 \cong T_2$, if there exists a bijective function $b : S_1 \rightarrow S_2$ such that:

– $b(s_1^0) = s_2^0$;
– $s_1 \xrightarrow{\ell}_1 s_1'$ iff $b(s_1) \xrightarrow{\ell}_2 b(s_1')$;
– $s \xrightarrow{\ell}\surd_1$ iff $b(s) \xrightarrow{\ell}\surd_2$.

Henceforth, we will always consider two conditional transition systems essentially the same if they are isomorphic.

Remark 3.1. The set $\mathbb{CTS}_\kappa$ is independent of $\mathcal{S}_\kappa$. By that we mean the following. Let $\mathbb{CTS}_\kappa$ and $\mathbb{CTS}_\kappa'$ result from different choices for $\mathcal{S}_\kappa$. Then there exists a bijection $b : \mathbb{CTS}_\kappa \rightarrow \mathbb{CTS}_\kappa'$ such that for all $T \in \mathbb{CTS}_\kappa$, $T \cong b(T)$.

Bisimilarity has to be adapted to the setting with guarded actions. In the definition given below, we use a well-known notion from the field of Boolean algebras: the partial order relation $\sqsubseteq$ on $\mathcal{C}$ defined by

$$\alpha \sqsubseteq \beta \ \text{ iff } \ \alpha \sqcup \beta = \beta \ .$$

Moreover, we use the notation $\bigsqcup A$, where $A = \{\alpha_1, \ldots, \alpha_n\} \subseteq \mathcal{C}$, for $\alpha_1 \sqcup \ldots \sqcup \alpha_n$.

Let $T_1 = (S_1, \rightarrow_1, \rightarrow\surd_1, s_1^0) \in \mathbb{CTS}_\kappa$ and $T_2 = (S_2, \rightarrow_2, \rightarrow\surd_2, s_2^0) \in \mathbb{CTS}_\kappa$ (for an infinite cardinal κ). Then a *splitting bisimulation* B between T_1 and T_2 is a binary relation $B \subseteq S_1 \times S_2$ such that $B(s_1^0, s_2^0)$ and for all s_1, s_2 such that $B(s_1, s_2)$:

– if $s_1 \xrightarrow{[\alpha]\,a}_1 s_1'$, then there is a set $CS_2' \subseteq \mathcal{C}^- \times S_2$ such that $\alpha \sqsubseteq \bigsqcup \mathrm{dom}(CS_2')$ and for all $(\alpha', s_2') \in CS_2'$, $s_2 \xrightarrow{[\alpha']\,a}_2 s_2'$ and $B(s_1', s_2')$;

[2] We write $A \uplus B$ for the disjoint union of sets A and B, i.e. $A \uplus B = (A \times \{\emptyset\}) \cup (B \times \{\{\emptyset\}\})$. We write μ_1 and μ_2 for the associated injections $\mu_1 : A \rightarrow A \uplus B$ and $\mu_2 : B \rightarrow A \uplus B$, defined by $\mu_1(a) = (a, \emptyset)$ and $\mu_2(b) = (b, \{\emptyset\})$.

[3] In [11], the definition of $\mathbb{CTS}_\kappa$ is given for an arbitrary set of *atomic conditions*. In the case where the set $\{\mathcal{E}_a \mid a \in \mathsf{A}\}$ is taken as the set of atomic conditions, that definition and the definition given here are essentially the same.

– if $s_2 \xrightarrow{[\alpha]\,a}_2 s_2'$, then there is a set $CS_1' \subseteq \mathcal{C}^- \times S_1$ such that $\alpha \sqsubseteq \bigsqcup \mathrm{dom}(CS_1')$ and for all $(\alpha', s_1') \in CS_1'$, $s_1 \xrightarrow{[\alpha']\,a}_1 s_1'$ and $B(s_1', s_2')$;
– if $s_1 \xrightarrow{[\alpha]\,a} \surd_1$, then there is a set $C' \subseteq \mathcal{C}^-$ such that $\alpha \sqsubseteq \bigsqcup C'$ and for all $\alpha' \in C'$, $s_2 \xrightarrow{[\alpha']\,a} \surd_2$;
– if $s_2 \xrightarrow{[\alpha]\,a} \surd_2$, then there is a set $C' \subseteq \mathcal{C}^-$ such that $\alpha \sqsubseteq \bigsqcup C'$ and for all $\alpha' \in C'$, $s_1 \xrightarrow{[\alpha']\,a} \surd_1$.

Two conditional transition systems $T_1, T_2 \in \mathbb{CTS}_\kappa$ are *splitting bisimilar*, written $T_1 \Leftrightarrow T_2$, if there exists a splitting bisimulation B between T_1 and T_2. Let B be a splitting bisimulation between T_1 and T_2. Then we say that B is a splitting bisimulation *witnessing* $T_1 \Leftrightarrow T_2$.

The name splitting bisimulation is used because a transition of one of the related transition systems may be simulated by a set of transitions of the other transition system. Splitting bisimulation should not be confused with split bisimulation [15].

It is easy to see that $\Leftrightarrow$ is an equivalence on $\mathbb{CTS}_\kappa$. Let $T \in \mathbb{CTS}_\kappa$. Then we write $[T]_{\Leftrightarrow}$ for $\{T' \in \mathbb{CTS}_\kappa \mid T \Leftrightarrow T'\}$, i.e. the $\Leftrightarrow$-equivalence class of T. We write $\mathbb{CTS}_\kappa/{\Leftrightarrow}$ for the set of equivalence classes $\{[T]_{\Leftrightarrow} \mid T \in \mathbb{CTS}_\kappa\}$.

In Section 4, we will use $\mathbb{CTS}_\kappa/{\Leftrightarrow}$ as the domain of a structure that is a model of $\mathrm{PA}_\delta^{\mathrm{cc}}$. As the domain of a structure, $\mathbb{CTS}_\kappa/{\Leftrightarrow}$ must be a set. That is the case because $\mathbb{CTS}_\kappa$ is a set. The latter is guaranteed by considering only conditional transition systems of which the set of states is a subset of $\mathcal{S}_\kappa$.

Remark 3.2. The question arises whether $\mathcal{S}_\kappa$ is large enough if its cardinality is greater than or equal to κ. This question can be answered in the affirmative. Let $T = (S, \rightarrow, \rightarrow\surd, s^0)$ be a connected conditional transition system of which the branching degree is less than κ. Then there exists a connected conditional transition system $T' = (S', \rightarrow', \rightarrow\surd', s^{0\prime})$ of which the branching degree is less than κ such that $T \Leftrightarrow T'$ and the cardinality of S' is less than κ.

It is easy to see that, if we would consider conditional transition systems with unreachable states as well, each conditional transition system would be splitting bisimilar to its connected part. This justifies the choice to consider only connected conditional transition systems. It is easy to see that isomorphic conditional transition systems are splitting bisimilar. This justifies the choice to consider conditional transition systems essentially the same if they are isomorphic.

4 Full Splitting Bisimulation Models of $\mathrm{PA}_\delta^{\mathrm{cc}}$

In this section, we introduce the full splitting bisimulation models of $\mathrm{PA}_\delta^{\mathrm{cc}}$. They are models of which the domain consists of equivalence classes of conditional transition systems modulo splitting bisimilarity. The qualification "full" originates from [10]. It expresses that there exist other splitting bisimulation models, but each of them is isomorphically embedded in a full splitting bisimulation model.

The models of $\mathrm{PA}_\delta^{\mathrm{cc}}$ are structures that consist of the following:

- a non-empty set $\mathcal{D}$, called the *domain* of the model;
- for each constant of $\mathrm{PA}_\delta^{\mathrm{cc}}$, an element of $\mathcal{D}$;
- for each n-ary operator of $\mathrm{PA}_\delta^{\mathrm{cc}}$, an n-ary operation on $\mathcal{D}$.

In the full splitting bisimulation models of $\mathrm{PA}_\delta^{\mathrm{cc}}$ that are introduced in this section, the domain is $\mathbb{CTS}_\kappa/\underline{\leftrightarrow}$ for some infinite cardinal κ. We obtain the models concerned by associating certain elements of $\mathbb{CTS}_\kappa/\underline{\leftrightarrow}$ with the constants of $\mathrm{PA}_\delta^{\mathrm{cc}}$ and certain operations on $\mathbb{CTS}_\kappa/\underline{\leftrightarrow}$ with the operators of $\mathrm{PA}_\delta^{\mathrm{cc}}$. We begin by associating elements of $\mathbb{CTS}_\kappa$ and operations on $\mathbb{CTS}_\kappa$ with the constants and operators. The result of this is subsequently lifted to $\mathbb{CTS}_\kappa/\underline{\leftrightarrow}$.

It is assumed that for each infinite cardinal κ a fixed but arbitrary function $\mathrm{ch}_\kappa : (\mathcal{P}(\mathcal{S}_\kappa) \setminus \emptyset) \to \mathcal{S}_\kappa$ such that for all $S \in \mathcal{P}(\mathcal{S}_\kappa) \setminus \emptyset$, $\mathrm{ch}_\kappa(S) \in S$ has been given.

We will use the abbreviation $s \xrightarrow{a} s' \wr s''$ for $s \xrightarrow{a} s' \lor (s \xrightarrow{a} \surd \land s' = s'')$.

We associate with each constant c of $\mathrm{PA}_\delta^{\mathrm{cc}}$ an element $\widehat{c}$ of $\mathbb{CTS}_\kappa$ and with each operator f of $\mathrm{PA}_\delta^{\mathrm{cc}}$ an operation $\widehat{f}$ on $\mathbb{CTS}_\kappa$ as follows.

- $$\widehat{\delta} = (\{s^0\}, \emptyset, \emptyset, s^0)\;,$$
  where
  $$s^0 = \mathrm{ch}_\kappa(\mathcal{S}_\kappa)\;.$$
- $$\widehat{a} = (\{s^0\}, \emptyset, \to\surd, s^0)\;,$$
  where
  $$\begin{aligned} s^0 &= \mathrm{ch}_\kappa(\mathcal{S}_\kappa)\;,\\ \xrightarrow{(\top,a)}\surd &= \{s^0\}\;,\end{aligned}$$
  and for every $(\alpha, a') \in (\mathcal{C}^- \times \mathsf{A}) \setminus \{(\top, a)\}$:
  $$\xrightarrow{(\alpha,a')}\surd = \emptyset\;.$$
- Let $T_i = (S_i, \to_i, \to\surd_i, s_i^0) \in \mathbb{CTS}_\kappa$ for $i = 1, 2$. Then
  $$T_1 \mathbin{\widehat{+}} T_2 = \Gamma(S, \to, \to\surd, s^0)\;,$$
  where
  $$\begin{aligned} s^0 &= \mathrm{ch}_\kappa(\mathcal{S}_\kappa \setminus (S_1 \uplus S_2))\;,\\ S &= \{s^0\} \cup (S_1 \uplus S_2)\;,\end{aligned}$$
  and for every $(\alpha, a) \in \mathcal{C}^- \times \mathsf{A}$:
  $$\begin{aligned} \xrightarrow{(\alpha,a)} = {} & \{(s^0, \mu_1(s)) \mid s_1^0 \xrightarrow{[\alpha]\,a}_1 s\}\\ & \cup \{(s^0, \mu_2(s)) \mid s_2^0 \xrightarrow{[\alpha]\,a}_2 s\}\\ & \cup \{(\mu_1(s), \mu_1(s')) \mid s \xrightarrow{[\alpha]\,a}_1 s'\}\\ & \cup \{(\mu_2(s), \mu_2(s')) \mid s \xrightarrow{[\alpha]\,a}_2 s'\}\;,\end{aligned}$$

$$\begin{aligned}\xrightarrow{(\alpha,a)}\surd = {} & \{s^0 \mid s_1^0 \xrightarrow{[\alpha]\,a} \surd_1\} \\ & \cup \{s^0 \mid s_2^0 \xrightarrow{[\alpha]\,a} \surd_2\} \\ & \cup \{\mu_1(s) \mid s \xrightarrow{[\alpha]\,a} \surd_1\} \\ & \cup \{\mu_2(s) \mid s \xrightarrow{[\alpha]\,a} \surd_2\}\ .\end{aligned}$$

– Let $T_i = (S_i, \rightarrow_i, \rightarrow\surd_i, s_i^0) \in \mathbb{CTS}_\kappa$ for $i = 1, 2$. Then

$$T_1 \mathbin{\widehat{\cdot}} T_2 = \Gamma(S, \rightarrow, \rightarrow\surd, s_1^0)\ ,$$

where

$$S = S_1 \uplus S_2\ ,$$

and for every $(\alpha, a) \in \mathcal{C}^- \times \mathsf{A}$:

$$\begin{aligned}\xrightarrow{(\alpha,a)} \quad & = \{(\mu_1(s), \mu_1(s')) \mid s \xrightarrow{[\alpha]\,a}_1 s'\} \\ & \cup \{(\mu_1(s), \mu_2(s_2^0)) \mid s \xrightarrow{[\alpha]\,a} \surd_1\} \\ & \cup \{(\mu_2(s), \mu_2(s')) \mid s \xrightarrow{[\alpha]\,a}_2 s'\}\ , \\ \xrightarrow{(\alpha,a)}\surd & = \{\mu_2(s) \mid s \xrightarrow{[\alpha]\,a} \surd_2\}\ .\end{aligned}$$

– Let $T = (S, \rightarrow, \rightarrow\surd, s^0) \in \mathbb{CTS}_\kappa$. Then

$$\alpha \mathbin{\widehat{:\rightarrow}} T = \Gamma(S, \rightarrow', \rightarrow\surd', s^0)\ ,$$

where for every $(\alpha', a) \in \mathcal{C}^- \times \mathsf{A}$:

$$\begin{aligned}\xrightarrow{(\alpha',a)}{}' \quad & = \{(s^0, s') \mid \exists\beta \bullet s^0 \xrightarrow{[\beta]\,a} s' \ \wedge\ \alpha' = \alpha \sqcap \beta\} \\ & \cup \{(s, s') \mid s \xrightarrow{[\alpha']\,a} s' \ \wedge\ s \neq s^0\}\ , \\ \xrightarrow{(\alpha',a)}\surd' & = \{s^0 \mid \exists\beta \bullet s^0 \xrightarrow{[\beta]\,a} \surd \ \wedge\ \alpha' = \alpha \sqcap \beta\} \\ & \cup \{s \mid s \xrightarrow{[\alpha']\,a} \surd \ \wedge\ s \neq s^0\}\ .\end{aligned}$$

– Let $T_i = (S_i, \rightarrow_i, \rightarrow\surd_i, s_i^0) \in \mathbb{CTS}_\kappa$ for $i = 1, 2$. Then

$$T_1 \mathbin{\widehat{\parallel}} T_2 = (S, \rightarrow, \rightarrow\surd, s^0)\ ,$$

where

$$\begin{aligned} s^0 & = (s_1^0, s_2^0)\ , \\ s^{\checkmark} & = \mathrm{ch}_\kappa(\mathcal{S}_\kappa \setminus (S_1 \cup S_2))\ , \\ S & = ((S_1 \cup \{s^{\checkmark}\}) \times (S_2 \cup \{s^{\checkmark}\})) \setminus \{(s^{\checkmark}, s^{\checkmark})\}\ ,\end{aligned}$$

and for every $(\alpha, a) \in \mathcal{C}^- \times \mathsf{A}$:

$$\begin{aligned}\xrightarrow{(\alpha,a)} \quad & = \{((s_1, s_2), (s_1', s_2)) \mid (s_1', s_2) \in S \ \wedge\ s_1 \xrightarrow{[\alpha]\,a}_1 s_1' \wr s^{\checkmark}\} \\ & \cup \{((s_1, s_2), (s_1, s_2')) \mid (s_1, s_2') \in S \ \wedge\ s_2 \xrightarrow{[\alpha]\,a}_2 s_2' \wr s^{\checkmark}\}\ , \\ \xrightarrow{(\alpha,a)}\surd & = \{(s_1, s^{\checkmark}) \mid s_1 \xrightarrow{[\alpha]\,a} \surd_1\} \\ & \cup \{(s^{\checkmark}, s_2) \mid s_2 \xrightarrow{[\alpha]\,a} \surd_2\}\ .\end{aligned}$$

– Let $T_i = (S_i, \rightarrow_i, \rightarrow\surd_i, s_i^0) \in \mathbb{CTS}_\kappa$ for $i = 1, 2$. Suppose that $T_1 \mathbin{\widehat{\|}} T_2 = (S, \rightarrow, \rightarrow\surd, s^0)$ where $S = ((S_1 \cup \{s^\checkmark\}) \times (S_2 \cup \{s^\checkmark\})) \setminus \{(s^\checkmark, s^\checkmark)\}$ and $s^\checkmark = \mathrm{ch}_\kappa(\mathcal{S}_\kappa \setminus (S_1 \cup S_2))$. Then

$$T_1 \mathbin{\widehat{\lfloor\!\lfloor}} T_2 = \Gamma(S', \rightarrow', \rightarrow\surd, s^{0\prime}) ,$$

where

$$s^{0\prime} = \mathrm{ch}_\kappa(\mathcal{S}_\kappa \setminus S) ,$$

$$S' = \{s^{0\prime}\} \cup S ,$$

and for every $(\alpha, a) \in \mathcal{C}^- \times \mathsf{A}$:

$$\xrightarrow{(\alpha,a)}{}' = \{(s^{0\prime}, (s, s_2^0)) \mid s_1^0 \xrightarrow{[\alpha]\, a}_1 s \wr s^\checkmark\} \cup \xrightarrow{(\alpha,a)} .$$

– Let $T = (S, \rightarrow, \rightarrow\surd, s^0) \in \mathbb{CTS}_\kappa$. Then

$$\widehat{\partial_H}(T) = \Gamma(S, \rightarrow', \rightarrow\surd', s^0) ,$$

where for every $(\alpha, a) \in \mathcal{C}^- \times (\mathsf{A} \setminus H)$:

$$\begin{aligned} \xrightarrow{(\alpha,a)}{}' &= \{(s, s') \mid \exists \alpha' \bullet s \xrightarrow{[\alpha']\, a} s' \wedge \alpha = \partial_H(\alpha')\} , \\ \xrightarrow{(\alpha,a)}\surd' &= \{s \mid \exists \alpha' \bullet s \xrightarrow{[\alpha']\, a} \surd \wedge \alpha = \partial_H(\alpha')\} , \end{aligned}$$

and for every $(\alpha, a) \in \mathcal{C}^- \times H$:

$$\begin{aligned} \xrightarrow{(\alpha,a)}{}' &= \emptyset , \\ \xrightarrow{(\alpha,a)}\surd' &= \emptyset . \end{aligned}$$

– Let $T = (S, \rightarrow, \rightarrow\surd, s^0) \in \mathbb{CTS}_\kappa$. Then

$$\widehat{t_I}(T) = \Gamma(S, \rightarrow', \rightarrow\surd', s^0) ,$$

where for every $(\alpha, a) \in \mathcal{C}^- \times ((\mathsf{A} \setminus I) \setminus \{t\})$:

$$\begin{aligned} \xrightarrow{(\alpha,a)}{}' &= \{(s, s') \mid \exists \alpha' \bullet s \xrightarrow{[\alpha']\, a} s' \wedge \alpha = t_I(\alpha')\} , \\ \xrightarrow{(\alpha,a)}\surd' &= \{s \mid \exists \alpha' \bullet s \xrightarrow{[\alpha']\, a} \surd \wedge \alpha = t_I(\alpha')\} , \end{aligned}$$

and for every $\alpha \in \mathcal{C}^-$:

$$\begin{aligned} \xrightarrow{(\alpha,t)}{}' &= \{(s, s') \mid \exists \alpha', a \bullet s \xrightarrow{[\alpha']\, a} s' \wedge a \in I \cup \{t\} \wedge \alpha = t_I(\alpha')\} , \\ \xrightarrow{(\alpha,t)}\surd' &= \{s \mid \exists \alpha', a \bullet s \xrightarrow{[\alpha']\, a} \surd \wedge a \in I \cup \{t\} \wedge \alpha = t_I(\alpha')\} , \end{aligned}$$

and for every $(\alpha, a) \in \mathcal{C}^- \times (I \setminus \{t\})$:

$$\begin{aligned} \xrightarrow{(\alpha,a)}{}' &= \emptyset , \\ \xrightarrow{(\alpha,a)}\surd' &= \emptyset . \end{aligned}$$

In the definition of alternative composition on $\mathbb{CTS}_\kappa$, the connected part of a conditional transition system is extracted because the initial states of the conditional transition systems T_1 and T_2 may be unreachable from the new initial state. The new initial state is introduced because, in T_1 and/or T_2, there may exist a transition back to the initial state. In the definition of sequential composition on $\mathbb{CTS}_\kappa$, the connected part of a conditional transition system is extracted because the initial state of the conditional transition system T_2 may be unreachable from the initial state of the conditional transition system T_1 – due to absence of termination in T_1.

Remark 4.1. The elements of $\mathbb{CTS}_\kappa$ and the operations on $\mathbb{CTS}_\kappa$ defined above are independent of ch_κ. Different choices for ch_κ lead for each constant of $\mathrm{PA}_\delta^{\mathrm{cc}}$ to isomorphic elements of $\mathbb{CTS}_\kappa$ and lead for each operator of $\mathrm{PA}_\delta^{\mathrm{cc}}$ to operations on $\mathbb{CTS}_\kappa$ with isomorphic results.

We can easily show that splitting bisimilarity is a congruence with respect to alternative composition, sequential composition, guarded command, parallel composition, left merge, encapsulation and pre-abstraction.

Lemma 4.1 (Congruence). *Let κ be an infinite cardinal. Then for all $T_1, T_2, T_1', T_2' \in \mathbb{CTS}_\kappa$ and $\alpha \in \mathcal{C}$, $T_1 \mathrel{\underline{\leftrightarrow}} T_1'$ and $T_2 \mathrel{\underline{\leftrightarrow}} T_2'$ imply $T_1 \mathbin{\widehat{+}} T_2 \mathrel{\underline{\leftrightarrow}} T_1' \mathbin{\widehat{+}} T_2'$, $T_1 \mathbin{\widehat{\cdot}} T_2 \mathrel{\underline{\leftrightarrow}} T_1' \mathbin{\widehat{\cdot}} T_2'$, $\alpha \mathbin{\widehat{:\rightarrow}} T_1 \mathrel{\underline{\leftrightarrow}} \alpha \mathbin{\widehat{:\rightarrow}} T_1'$, $T_1 \mathbin{\widehat{\|}} T_2 \mathrel{\underline{\leftrightarrow}} T_1' \mathbin{\widehat{\|}} T_2'$, $T_1 \mathbin{\widehat{\lfloor\!\lfloor}} T_2 \mathrel{\underline{\leftrightarrow}} T_1' \mathbin{\widehat{\lfloor\!\lfloor}} T_2'$, $\widehat{\partial_H}(T_1) \mathrel{\underline{\leftrightarrow}} \widehat{\partial_H}(T_1')$ and $\widehat{t_I}(T_1) \mathrel{\underline{\leftrightarrow}} \widehat{t_I}(T_1')$.*

Proof. Let $T_i = (S_i, \rightarrow_i, \rightarrow\surd_i, s_i^0)$ and $T_i' = (S_i', \rightarrow_i', \rightarrow\surd_i', s_i^{0\prime})$ for $i = 1, 2$. Let R_1 and R_2 be splitting bisimulations witnessing $T_1 \mathrel{\underline{\leftrightarrow}} T_1'$ and $T_2 \mathrel{\underline{\leftrightarrow}} T_2'$, respectively. Then we construct relations $R_{\widehat{+}}$, $R_{\widehat{\cdot}}$, $R_{\widehat{:\rightarrow}}$, $R_{\widehat{\|}}$, $R_{\widehat{\lfloor\!\lfloor}}$, $R_{\widehat{\partial_H}}$ and $R_{\widehat{t_I}}$, as follows:

- $R_{\widehat{+}} = (\{(s^0, s^{0\prime})\} \cup \mu_1(R_1) \cup \mu_2(R_2)) \cap (S \times S')$, where S and S' are the sets of states of $T_1 \mathbin{\widehat{+}} T_2$ and $T_1' \mathbin{\widehat{+}} T_2'$, respectively, and s^0 and $s^{0\prime}$ are the initial states of $T_1 \mathbin{\widehat{+}} T_2$ and $T_1' \mathbin{\widehat{+}} T_2'$, respectively;
- $R_{\widehat{\cdot}} = (\mu_1(R_1) \cup \mu_2(R_2)) \cap (S \times S')$, where S and S' are the sets of states of $T_1 \mathbin{\widehat{\cdot}} T_2$ and $T_1' \mathbin{\widehat{\cdot}} T_2'$, respectively;
- $R_{\widehat{:\rightarrow}} = R_1 \cap (S \times S')$, where S and S' are the sets of states of $\alpha \mathbin{\widehat{:\rightarrow}} T_1$ and $\alpha \mathbin{\widehat{:\rightarrow}} T_1'$, respectively;
- $R_{\widehat{\|}} = \{((s_1, s_2), (s_1', s_2')) \mid (s_1, s_1') \in R_1 \cup R^{\surd}, (s_2, s_2') \in R_2 \cup R^{\surd}\} \setminus R^{\surd}$, where $R^{\surd} = \{(\mathrm{ch}_\kappa(\mathcal{S}_\kappa \setminus (S_1 \cup S_2)), \mathrm{ch}_\kappa(\mathcal{S}_\kappa \setminus (S_1' \cup S_2')))\}$;
- $R_{\widehat{\lfloor\!\lfloor}} = (\{(s^0, s^{0\prime})\} \cup R_{\widehat{\|}}) \cap (S \times S')$, where S and S' are the sets of states of $T_1 \mathbin{\widehat{\lfloor\!\lfloor}} T_2$ and $T_1' \mathbin{\widehat{\lfloor\!\lfloor}} T_2'$, respectively, and s^0 and $s^{0\prime}$ are the initial states of $T_1 \mathbin{\widehat{\lfloor\!\lfloor}} T_2$ and $T_1' \mathbin{\widehat{\lfloor\!\lfloor}} T_2'$, respectively;
- $R_{\widehat{\partial_H}} = R_1 \cap (S \times S')$, where S and S' are the sets of states of $\widehat{\partial_H}(T_1)$ and $\widehat{\partial_H}(T_1')$, respectively;
- $R_{\widehat{t_I}} = R_1 \cap (S \times S')$, where S and S' are the sets of states of $\widehat{t_I}(T_1)$ and $\widehat{t_I}(T_1')$, respectively.

Here, we write $\mu_i(R_i)$ for $\{(\mu_i(s), \mu_i(s')) \mid R_i(s, s')\}$, where μ_i is used to denote both the injection of S_i into $S_1 \uplus S_2$ and the injection of S'_i into $S'_1 \uplus S'_2$. Given the definitions of alternative composition, sequential composition, guarded command, parallel composition, left merge, encapsulation and pre-abstraction, it is easy to see that $R_{\widehat{+}}$, $R_{\widehat{\cdot}}$, $R_{\widehat{:\rightarrow}}$, $R_{\widehat{\parallel}}$, $R_{\widehat{\lfloor\!\lfloor}}$, $R_{\widehat{\partial_H}}$ and $R_{\widehat{t_I}}$ are splitting bisimulations witnessing $T_1 \mathbin{\widehat{+}} T_2 \mathrel{\underline{\Leftrightarrow}} T'_1 \mathbin{\widehat{+}} T'_2$, $T_1 \mathbin{\widehat{\cdot}} T_2 \mathrel{\underline{\Leftrightarrow}} T'_1 \mathbin{\widehat{\cdot}} T'_2$, $\alpha \mathbin{\widehat{:\rightarrow}} T_1 \mathrel{\underline{\Leftrightarrow}} \alpha \mathbin{\widehat{:\rightarrow}} T'_1$, $T_1 \mathbin{\widehat{\parallel}} T_2 \mathrel{\underline{\Leftrightarrow}} T'_1 \mathbin{\widehat{\parallel}} T'_2$, $T_1 \mathbin{\widehat{\lfloor\!\lfloor}} T_2 \mathrel{\underline{\Leftrightarrow}} T'_1 \mathbin{\widehat{\lfloor\!\lfloor}} T'_2$, $\widehat{\partial_H}(T_1) \mathrel{\underline{\Leftrightarrow}} \widehat{\partial_H}(T'_1)$ and $\widehat{t_I}(T_1) \mathrel{\underline{\Leftrightarrow}} \widehat{t_I}(T'_1)$, respectively. □

The *full splitting bisimulation models* $\mathfrak{P}^{\mathrm{cc}}_\kappa$, one for each infinite cardinal κ, consist of the following:

- a set $\mathcal{P}$, called the domain of $\mathfrak{P}^{\mathrm{cc}}_\kappa$;
- for each constant c of $\mathrm{PA}^{\mathrm{cc}}_\delta$, an element $\widetilde{c}$ of $\mathcal{P}$;
- for each n-ary operator f of $\mathrm{PA}^{\mathrm{cc}}_\delta$, an n-ary operation $\widetilde{f}$ on $\mathcal{P}$;

where those ingredients are defined as follows:

$$
\begin{aligned}
&\mathcal{P} &&= \mathbb{CTS}_\kappa/\underline{\Leftrightarrow}\,, &\quad& \alpha \mathbin{\widetilde{:\rightarrow}} [T_1]_{\underline{\Leftrightarrow}} &&= [\alpha \mathbin{\widehat{:\rightarrow}} T_1]_{\underline{\Leftrightarrow}}\,,\\
&\widetilde{\delta} &&= [\,\widehat{\delta}\,]_{\underline{\Leftrightarrow}}\,, && [T_1]_{\underline{\Leftrightarrow}} \mathbin{\widetilde{\parallel}} [T_2]_{\underline{\Leftrightarrow}} &&= [T_1 \mathbin{\widehat{\parallel}} T_2]_{\underline{\Leftrightarrow}}\,,\\
&\widetilde{a} &&= [\,\widehat{a}\,]_{\underline{\Leftrightarrow}}\,, && [T_1]_{\underline{\Leftrightarrow}} \mathbin{\widetilde{\lfloor\!\lfloor}} [T_2]_{\underline{\Leftrightarrow}} &&= [T_1 \mathbin{\widehat{\lfloor\!\lfloor}} T_2]_{\underline{\Leftrightarrow}}\,,\\
&[T_1]_{\underline{\Leftrightarrow}} \mathbin{\widetilde{+}} [T_2]_{\underline{\Leftrightarrow}} &&= [T_1 \mathbin{\widehat{+}} T_2]_{\underline{\Leftrightarrow}}\,, && \widetilde{\partial_H}([T_1]_{\underline{\Leftrightarrow}}) &&= [\widehat{\partial_H}(T_1)]_{\underline{\Leftrightarrow}}\,,\\
&[T_1]_{\underline{\Leftrightarrow}} \mathbin{\widetilde{\cdot}} [T_2]_{\underline{\Leftrightarrow}} &&= [T_1 \mathbin{\widehat{\cdot}} T_2]_{\underline{\Leftrightarrow}}\,, && \widetilde{t_I}([T_1]_{\underline{\Leftrightarrow}}) &&= [\widehat{t_I}(T_1)]_{\underline{\Leftrightarrow}}\,.
\end{aligned}
$$

The operations alternative composition, sequential composition, guarded command, parallel composition, left merge, encapsulation and pre-abstraction on $\mathbb{CTS}_\kappa/\underline{\Leftrightarrow}$ are well-defined because $\underline{\Leftrightarrow}$ is a congruence with respect to the corresponding operations on $\mathbb{CTS}_\kappa$.

The structures $\mathfrak{P}^{\mathrm{cc}}_\kappa$ are models of $\mathrm{PA}^{\mathrm{cc}}_\delta$.

Theorem 4.1 (Soundness). *For each infinite cardinal κ, we have $\mathfrak{P}^{\mathrm{cc}}_\kappa \models \mathrm{PA}^{\mathrm{cc}}_\delta$.*

Proof. Because $\underline{\Leftrightarrow}$ is a congruence, it is sufficient to show that all axioms are sound. The soundness of all axioms follows easily from the definitions of the ingredients of $\mathfrak{P}^{\mathrm{cc}}_\kappa$. □

The axioms of $\mathrm{PA}^{\mathrm{cc}}_\delta$ constitute a complete axiomatization of the full splitting bisimulation models.

Theorem 4.2 (Completeness). *Let κ be an infinite cardinal. Then we have, for all closed terms p and q of sort* **P** *from the language of $\mathrm{PA}^{\mathrm{cc}}_\delta$, $\mathfrak{P}^{\mathrm{cc}}_\kappa \models p = q$ implies $\mathrm{PA}^{\mathrm{cc}}_\delta \vdash p = q$.*

Proof. By Lemma 2.2, for each closed term p of sort **P** from the language of $\mathrm{PA}^{\mathrm{cc}}_\delta$, there is a closed term q of sort **P** from the language of $\mathrm{BPA}^{\mathrm{cc}}_\delta$ such that $p = q$ is derivable from the axioms of $\mathrm{PA}^{\mathrm{cc}}_\delta$; and by Lemma 2.3, all equations between closed terms of sort $\mathbf{P}$ from the language of $\mathrm{BPA}_\delta^{\mathrm{cc}}$ that can be derived from the axioms of $\mathrm{PA}_\delta^{\mathrm{cc}}$ can be derived from the axioms of $\mathrm{BPA}_\delta^{\mathrm{cc}}$. Therefore, it is sufficient to prove that the axioms of $\mathrm{BPA}_\delta^{\mathrm{cc}}$ constitute a complete axiomatization of the restrictions of the full splitting bisimulation models of $\mathrm{PA}_\delta^{\mathrm{cc}}$ to the constants and operators of $\mathrm{BPA}_\delta^{\mathrm{cc}}$. In [9], a proof of completeness of the axioms of BPA_τ for the graph models of BPA_τ is given. That proof introduces a strongly normalising graph rewriting system with the following two properties: (i) two process graphs are bisimilar iff their normal forms are isomorphic and (ii) every rewriting step corresponds to a proof step in BPA_τ. The proof from [9] can easily be adapted to conditional transition systems, splitting bisimilarity and $\mathrm{BPA}_\delta^{\mathrm{cc}}$. The one-step reductions are in this case *sharing* of double states as in [9], and two types of *joining* of transitions: (a) replacing $s \xrightarrow{[\alpha]\,a} s''$ and $s \xrightarrow{[\beta]\,a} s''$ by $s \xrightarrow{[\alpha \sqcup \beta]\,a} s''$ and (b) replacing $s \xrightarrow{[\alpha]\,a} \surd$ and $s \xrightarrow{[\beta]\,a} \surd$ by $s \xrightarrow{[\alpha \sqcup \beta]\,a} \surd$. □

As to be expected, the full splitting bisimulation models are related by isomorphic embeddings.

Theorem 4.3 (Isomorphic Embedding). *Let κ and κ' be infinite cardinals such that $\kappa < \kappa'$. Then $\mathfrak{P}_\kappa^{\mathrm{cc}}$ is isomorphically embedded in $\mathfrak{P}_{\kappa'}^{\mathrm{cc}}$.*

Proof. The proof is analogous to the proof of the corresponding property for the full splitting bisimulation models of $\mathrm{ACP}^{\mathrm{c}}$ given in [11]. □

5 SOS-Based Splitting Bisimilarity for $\mathbf{PA}_\delta^{\mathbf{cc}}$

It is customary to associate transition systems with closed terms (of sort $\mathbf{P}$) from the language of an ACP-like theory about processes by means of structural operational semantics and to identify closed terms if their associated transition systems are equivalent by a bisimilarity-based notion of equivalence.

The structural operational semantics of $\mathrm{PA}_\delta^{\mathrm{cc}}$ presented in Section 2 determines a conditional transition system for each closed term of sort $\mathbf{P}$ from the language of $\mathrm{PA}_\delta^{\mathrm{cc}}$. These transition systems are special in the sense that their states are closed terms of sort $\mathbf{P}$ from the language of $\mathrm{PA}_\delta^{\mathrm{cc}}$.

Let p be a closed term of sort $\mathbf{P}$ from the language of $\mathrm{PA}_\delta^{\mathrm{cc}}$. Then the transition system of p *induced by* the structural operational semantics of $\mathrm{PA}_\delta^{\mathrm{cc}}$, written $\mathrm{CTS}(p)$, is the connected conditional transition system $\Gamma(S, \rightarrow, \rightarrow\surd, s^0)$, where:

- S is the set of closed terms of sort $\mathbf{P}$ from the language of $\mathrm{PA}_\delta^{\mathrm{cc}}$;
- $\xrightarrow{(\alpha,a)} \subseteq S \times S$ and $\xrightarrow{(\alpha,a)}\surd \subseteq S$ for each $\alpha \in \mathcal{C} \setminus \{\bot\}$ and $a \in \mathsf{A}$ are the smallest subsets of $S \times S$ and S, respectively, for which the transition rules from Table 3 hold;
- $s^0 \in S$ is the closed term p.

Let p and q be closed terms of sort $\mathbf{P}$ from the language of $\mathrm{PA}_\delta^{\mathrm{cc}}$. Then we say that p and q are *splitting bisimilar*, written $p \Leftrightarrow q$, if $\mathrm{CTS}(p) \Leftrightarrow \mathrm{CTS}(q)$.

Clearly, the structural operational semantics does not give rise to infinitely branching conditional transition systems. For each closed term p of sort $\mathbf{P}$ from the language of $\mathrm{PA}_\delta^{\mathrm{cc}}$, there exists a $T \in \mathbb{CTS}_{\aleph_0}$ such that $\mathrm{CTS}(p) \cong T$. In Section 4, it has been shown that it is possible to consider infinitely branching conditional transition systems as well.

6 ACP with Coordination Conditions

In order to support communication, we generalize the parallel composition operator of $\mathrm{PA}_\delta^{\mathrm{cc}}$, resulting in $\mathrm{ACP}^{\mathrm{cc}}$.

Just as in $\mathrm{PA}_\delta^{\mathrm{cc}}$, it is assumed that a fixed but arbitrary finite set of *actions* A, with $\delta \notin \mathsf{A}$ and $t \in \mathsf{A}$, has been given. In $\mathrm{ACP}^{\mathrm{cc}}$, it is further assumed that a fixed but arbitrary commutative and associative *communication* function $\mid : \mathsf{A}_\delta \times \mathsf{A}_\delta \to \mathsf{A}_\delta$, such that $\delta \mid a = \delta$ and $t \mid a = \delta$ for all $a \in \mathsf{A}_\delta$, has been given. The function $\mid$ is regarded to give the result of synchronously performing any two actions for which this is possible, and to be δ otherwise.

The theory $\mathrm{ACP}^{\mathrm{cc}}$ is an extension of $\mathrm{PA}_\delta^{\mathrm{cc}}$. It has the constants and operators of $\mathrm{PA}_\delta^{\mathrm{cc}}$ and in addition:

- the binary *communication merge* operator $\mid : \mathbf{P} \times \mathbf{P} \to \mathbf{P}$;
- for each $U \subseteq \mathsf{A}$ with $t \in U$, the unary *enabledness update* operator $\Psi_U : \mathbf{C} \to \mathbf{C}$.

We use infix notation for the the communication merge operator as well.

Let p and q be closed terms of sort $\mathbf{P}$, ζ be a closed term of sort $\mathbf{C}$, and $U \subseteq \mathsf{A}$ with $t \in U$. Intuitively, the additional operators can be explained as follows:

- $p \mid q$ behaves the same as $p \parallel q$, except that it starts with performing an action of p and an action of q synchronously;
- $\Psi_U(\zeta)$ is ζ with all enabledness conditions $\mathcal{E}_a$ with $a \notin U$ replaced by $\bot$ and all enabledness conditions $\mathcal{E}_a$ with $a \in U$ replaced by $\top$.

The axioms of $\mathrm{ACP}^{\mathrm{cc}}$ are the axioms of $\mathrm{PA}_\delta^{\mathrm{cc}}$ with axioms M1–M4 replaced by axioms CM1–CM4, CME, C1–C3 and EU1–EU7 from Table 4. In axiom schema CME, the a_i, b_i, a'_j and b'_j stand for arbitrary action constants, and the ζ_i, ξ_i, ζ'_j and ξ'_j stand for arbitrary closed terms of sort $\mathbf{C}$. Moreover, for every $U, V, W \subseteq \mathsf{A}$ with $t \in U, V, W$, $\Phi_{U,V,W}$ iff for all $u \in \mathsf{A} \setminus \{t\}$:

$$\begin{aligned} u \in V \;\Leftrightarrow\; & \exists j < n' \bullet (\Psi_W(\zeta'_j) = \top \,\wedge\, u \mid a'_j \in U) \\ & \vee\; \exists j < m' \bullet (\Psi_W(\xi'_j) = \top \,\wedge\, u \mid b'_j \in U) \end{aligned}$$

and

$$\begin{aligned} u \in W \;\Leftrightarrow\; & \exists i < n \bullet (\Psi_V(\zeta_i) = \top \,\wedge\, a_i \mid u \in U) \\ & \vee\; \exists i < m \bullet (\Psi_V(\xi_i) = \top \,\wedge\, b_i \mid u \in U)\,. \end{aligned}$$

Axioms CM2–CM4 of $\mathrm{ACP}^{\mathrm{cc}}$ are in fact the same as axioms M2–M4 of $\mathrm{PA}_\delta^{\mathrm{cc}}$. The axioms of $\mathrm{ACP}^{\mathrm{cc}}$ do not include axioms CM5–CM9 of ACP (see e.g. [8]), i.e.

Table 4. Axioms for $\mathrm{ACP^{cc}}$ $(a, b, c \in \mathsf{A}_\delta)$

$x \parallel y = x \mathbin{\|\!\!\lfloor} y + y \mathbin{\|\!\!\lfloor} x + x \mid y$	CM1	$a \mid b = b \mid a$	C1
$a \mathbin{\|\!\!\lfloor} x = a \cdot x$	CM2	$(a \mid b) \mid c = a \mid (b \mid c)$	C2
$a \cdot x \mathbin{\|\!\!\lfloor} y = a \cdot (x \parallel y)$	CM3	$\delta \mid a = \delta$	C3
$(x + y) \mathbin{\|\!\!\lfloor} z = x \mathbin{\|\!\!\lfloor} z + y \mathbin{\|\!\!\lfloor} z$	CM4		

$$
\begin{aligned}
&\Big(\sum_{i<n} \zeta_i :\to a_i \cdot x_i + \sum_{i<m} \xi_i :\to b_i\Big) \;\Big|\; \Big(\sum_{j<n'} \zeta'_j :\to a'_j \cdot x'_j + \sum_{j<m'} \xi'_j :\to b'_j\Big) = \\
&\sum_{\substack{t \in U \subseteq \mathsf{A}, t \in V \subseteq \mathsf{A}, t \in W \subseteq \mathsf{A},\\ \Phi_{U,V,W}}} \textstyle\bigsqcap_{u \in U} \mathcal{E}_u \sqcap \bigsqcap_{u \in \mathsf{A} \setminus U} -\mathcal{E}_u :\to \\
&\qquad \Big(\sum_{i<n, j<n'} \Psi_V(\zeta_i) \sqcap \Psi_W(\zeta'_j) :\to (a_i \mid a'_j) \cdot (x_i \parallel y_j) + \\
&\qquad \sum_{i<m, j<n'} \Psi_V(\xi_i) \sqcap \Psi_W(\zeta'_j) :\to (b_i \mid a'_j) \cdot y_j + \\
&\qquad \sum_{i<n, j<m'} \Psi_V(\zeta_i) \sqcap \Psi_W(\xi'_j) :\to (a_i \mid b'_j) \cdot x_i + \\
&\qquad \sum_{i<m, j<m'} \Psi_V(\xi_i) \sqcap \Psi_W(\xi'_j) :\to (b_i \mid b'_j)\Big) \qquad \text{CME}
\end{aligned}
$$

$\Psi_U(\bot) = \bot$		EU1	$\Psi_U(-\phi) = -\Psi_U(\phi)$	EU5
$\Psi_U(\top) = \top$		EU2	$\Psi_U(\phi \sqcup \psi) = \Psi_U(\phi) \sqcup \Psi_U(\psi)$	EU6
$\Psi_U(\mathcal{E}_a) = \bot$	if $a \notin U$	EU3	$\Psi_U(\phi \sqcap \psi) = \Psi_U(\phi) \sqcap \Psi_U(\psi)$	EU7
$\Psi_U(\mathcal{E}_a) = \top$	if $a \in U$	EU4		

the axioms of ACP for communication merge, and axioms GC9–GC10 of $\mathrm{ACP^c}$ (see e.g. [11]), i.e. the additional axioms of $\mathrm{ACP^c}$ for communication merge.

CME, the axiom schema that replaces CM5–CM9 and GC9–GC10 in $\mathrm{ACP^{cc}}$, needs further explanation. Consider processes p, q and $p \mid q$. The process p and the context in which $p \mid q$ is placed form parts of the context in which q is placed; and the process q and the context in which $p \mid q$ is placed form parts of the context in which p is placed. Any subset of A that includes t may be the set of all actions that are enabled in the context in which $p \mid q$ is placed. Suppose that $U \subseteq \mathsf{A}$ with $t \in U$ is the set of all actions that are enabled in the context in which $p \mid q$ is placed. Furthermore, suppose that $V \subseteq \mathsf{A}$ with $t \in V$ and $W \subseteq \mathsf{A}$ with $t \in W$ are the sets of all actions that are enabled in the contexts in which p and q, respectively, are placed. Then the following must hold for all $a \in \mathsf{A} \setminus \{t\}$:

- $a \in V$ iff q, with exactly the actions in W enabled, can perform an action b such that $a \mid b \in U$ (because a is enabled in q together with the context in which $p \mid q$ is placed);
- $a \in W$ iff p, with exactly the actions in V enabled, can perform an action b such that $a \mid b \in U$ (because a is enabled in p together with the context in which $p \mid q$ is placed).

V and W determine whether the conditions under which p and q can perform their initial actions evaluate to $\top$ or $\bot$. In the case of p, for example, the basis is that $\mathcal{E}_a$ evaluates to $\top$ if $a \in V$, and $\bot$ otherwise. All this is made precise in axiom schema CME. Notice that V and W need not exist for each U: the behaviour of p and q may inhibit proper mutual enabling of actions for $p \mid q$. In such cases, $p \mid q$ is considered to be incapable of doing anything. In other words, if V and W do not exist for some U, $p \mid q$ is considered to behave the same as δ in the event of U being the set of all actions that are enabled in the context in which $p \mid q$ is placed.

We proceed with giving a few examples. In the examples, we take A such that $\mathsf{A} = A \cup \{\overline{a} \mid a \in A\} \cup \{t\}$ for some set A. Moreover, we take $\mid : \mathsf{A}_\delta \times \mathsf{A}_\delta \to \mathsf{A}_\delta$ such that, for all $a \in A$ and $b \in \mathsf{A}$, $a \mid \overline{a} = t$, $a \mid b = \delta$ if $b \neq \overline{a}$, $\overline{a} \mid b = \delta$ if $b \neq a$, and $t \mid b = \delta$. We start with giving an example of processes p and q of which the behaviour inhibits proper mutual enabling of actions for $p \mid q$, whatever actions are enabled in the context in which $p \mid q$ is placed. Consider the processes $p \equiv \mathcal{E}_a :\to b$ and $q \equiv -\mathcal{E}_{\overline{b}} :\to \overline{a}$. The behaviour of these processes inhibits proper mutual enabling of actions for $p \mid q$ whatever actions are enabled in the context in which $p \mid q$ is placed: if a is enabled by q, then b is enabled by p and consequently a is not enabled by q; and if a is not enabled by q, then b is not enabled by p and consequently a is enabled by q. Hence, we have

$$(\mathcal{E}_a :\to b) \mid (-\mathcal{E}_{\overline{b}} :\to \overline{a}) = \delta \ .$$

We proceed with giving an example of processes p and q of which the behaviour does not inhibit proper mutual enabling of actions for $p \mid q$, whatever actions are enabled in the context in which $p \mid q$ is placed. Consider the processes $p \equiv \mathcal{E}_a :\to b + \overline{c}$ and $q \equiv \overline{a} + \mathcal{E}_{\overline{b}} :\to c$. There is a unique proper mutual enabling, viz. the mutual enabling in which exactly the actions a and $\overline{c}$ are enabled by q, and exactly the actions $\overline{b}$ and c are enabled by p. Hence, we have

$$(\mathcal{E}_a :\to b + \overline{c}) \mid (\overline{a} + \mathcal{E}_{\overline{b}} :\to c) = t \ .$$

Finally, we give an example of processes p and q of which the behaviour is such that there are two proper mutual enablings of actions for $p \mid q$, whatever actions are enabled in the context in which $p \mid q$ is placed. Consider the processes $p \equiv \mathcal{E}_a :\to b + -\mathcal{E}_a :\to c$ and $q \equiv \mathcal{E}_{\overline{b}} :\to \overline{a} + -\mathcal{E}_{\overline{b}} :\to \overline{c}$. There are two proper mutual enablings, viz. the mutual enabling in which only the action a is enabled by q and only the action $\overline{b}$ is enabled by p, and the mutual enabling in which only the action c is enabled by q and only the action $\overline{c}$ is enabled by p. Hence, we have

$$(\mathcal{E}_a :\to b + -\mathcal{E}_a :\to c) \mid (\mathcal{E}_{\overline{b}} :\to \overline{a} + -\mathcal{E}_{\overline{b}} :\to \overline{c}) = \delta + t = t \ .$$

Axioms similar to the axioms of ACP and $\mathrm{ACP}^{\mathrm{c}}$ for communication merge are too much to expect for $\mathrm{ACP}^{\mathrm{cc}}$: the mutual enabling of actions involved in the communication merge of two processes is a matter which can only be resolved by looking at the processes as a whole.

Like in the case of $\mathrm{PA}_\delta^{\mathrm{cc}}$, we can prove that all closed terms of sort $\mathbf{P}$ from the language of $\mathrm{ACP}^{\mathrm{cc}}$ are derivably equal to a basic term.

Lemma 6.1 (Elimination for ACP^{cc}). *For all closed terms p of sort* $\mathbf{P}$ *from the language of* ACP^{cc}*, there exists a basic term $q \in \mathcal{B}$ such that* ACP$^{cc} \vdash p = q$.

Proof. Let CME$'$ be CME with the conditions $\bigsqcap_{u \in U} \mathcal{E}_u \sqcap \bigsqcap_{u \in \mathsf{A} \setminus U} -\mathcal{E}_u$ moved inward (applying GC4) and combined with the conditions $\Psi_V(\zeta_i) \sqcap \Psi_W(\zeta'_j)$, $\Psi_V(\xi_i) \sqcap \Psi_W(\zeta'_j)$, $\Psi_V(\zeta_i) \sqcap \Psi_W(\xi'_j)$ and $\Psi_V(\xi_i) \sqcap \Psi_W(\xi'_j)$ (applying GC6). The term rewriting system consisting of all axioms of ACP^{cc}, except BA1–BA8 and A1–A3 and with CME replaced by CME$'$, oriented from left to right is strongly normalizing. This can be proved by using the method of lexicographical path ordering of Kamin and Lévy, making the signature one-sorted, ranking the operators $\|$, $\lfloor\!\lfloor$ and $|$ as in [9], taking the ordering $\ldots > \|_3 > \lfloor\!\lfloor_3, |_3 > \|_2 > \lfloor\!\lfloor_2, |_2 > {:\rightarrow} > \cdot > +$, ${:\rightarrow} > \delta, \sqcap$, $\partial_H > {:\rightarrow}, \bot, -, \sqcup$ for all $H \subseteq \mathsf{A} \setminus \{t\}$, $t_I > {:\rightarrow}, t, \top, -, \sqcup$ for all $I \subseteq \mathsf{A}$, $|_2 > a, \mathcal{E}_a, -, \Psi_U$ for all $a \in \mathsf{A}$ and $U \subseteq \mathsf{A}$, and giving the lexicographical status for the first argument to $\cdot$ and the lexicographical status for the second argument to ${:\rightarrow}$. Moreover, it is easy to see that each normal from is a basic term. □

We can also prove that ACP^{cc} is a conservative extension of BPA$_\delta^{cc}$.

Lemma 6.2 (Conservative extension). *If p and q are closed terms of sort* $\mathbf{P}$ *from the language of* BPA$_\delta^{cc}$*, then* BPA$_\delta^{cc} \vdash p = q$ *iff* ACP$^{cc} \vdash p = q$.

Proof. The proof is analogous to the proof of Lemma 2.3. □

The preceding two lemmas will be useful in the completeness proof of ACP^{cc} for the full splitting bisimulation models of ACP^{cc} that will be introduced below.

Associating transition systems with closed terms of sort $\mathbf{P}$ from the language of ACP^{cc} in the style of structural operational semantics looks to be impossible. The problem is to define by means of transition rules, for every $U, V, W \subseteq \mathsf{A}$ with $t \in U, V, W$, a binary relation on closed terms of sort $\mathbf{P}$ that corresponds to $\Phi'_{U,V,W}$ as in the definition of $\widehat{\|}'$ in Section 7.

7 Full Splitting Bisimulation Models of ACP^{cc}

In this section, we adapt the full splitting bisimulation models of PA$_\delta^{cc}$ to ACP^{cc}. In order to cover communication, the operations on $\mathbb{CTS}_\kappa/\Leftrightarrow$ associated with the operators $\|$ and $\lfloor\!\lfloor$ have to be adapted and an operation on $\mathbb{CTS}_\kappa/\Leftrightarrow$ associated with the operator $|$ has to be added.

Like before, we begin by associating operations $\widehat{\|}'$, $\widehat{\lfloor\!\lfloor}'$ and $\widehat{|}'$ on $\mathbb{CTS}_\kappa$ with the operators $\|$, $\lfloor\!\lfloor$ and $|$.

- Let $T_i = (S_i, \rightarrow_i, \rightarrow\surd_i, s_i^0) \in \mathbb{CTS}_\kappa$ for $i = 1, 2$. Then

$$T_1 \mathbin{\widehat{\|}'} T_2 = \Gamma(S, \rightarrow, \rightarrow\surd, s^0) \;,$$

where

$$s^0 = (s_1^0, s_2^0) \;,$$

$$s^{\surd} = \mathrm{ch}_\kappa(\mathcal{S}_\kappa \setminus (S_1 \cup S_2)) \;,$$

$$S = ((S_1 \cup \{s^{\surd}\}) \times (S_2 \cup \{s^{\surd}\})) \setminus \{(s^{\surd}, s^{\surd})\} \;,$$

and for every $(\alpha, a) \in \mathcal{C}^- \times \mathsf{A}$:

$$\begin{aligned}
\xrightarrow{(\alpha,a)} \; &= \{((s_1,s_2),(s_1',s_2)) \mid (s_1',s_2) \in S \;\wedge\; s_1 \xrightarrow{[\alpha]\,a}_1 s_1' \wr s^{\checkmark}\} \\
&\cup \{((s_1,s_2),(s_1,s_2')) \mid (s_1,s_2') \in S \;\wedge\; s_2 \xrightarrow{[\alpha]\,a}_2 s_2' \wr s^{\checkmark}\} \\
&\cup \{((s_1,s_2),(s_1',s_2')) \mid (s_1',s_2') \in S \;\wedge \\
&\qquad \bigvee_{\substack{\alpha',\beta' \in \mathcal{C}^-,\, a',b' \in \mathsf{A},\\ t \in U,V,W \subseteq \mathsf{A}}} \quad (s_1 \xrightarrow{[\alpha']\,a'}_1 s_1' \wr s^{\checkmark} \;\wedge\; s_2 \xrightarrow{[\beta']\,b'}_2 s_2' \wr s^{\checkmark} \;\wedge \\
&\qquad\qquad \mathit{\Phi}'_{U,V,W}(s_1,s_2) \;\wedge \\
&\qquad\qquad \textstyle\prod_{u \in U} \mathcal{E}_u \sqcap \prod_{u \in \mathsf{A} \setminus U} -\mathcal{E}_u = \alpha \;\wedge \\
&\qquad\qquad \Psi_V(\alpha') \sqcap \Psi_W(\beta') = \top \;\wedge\; a' \mid b' = a)\}\;,
\end{aligned}$$

$$\begin{aligned}
\xrightarrow{(\alpha,a)} \surd &= \{(s_1, s^{\checkmark}) \mid s_1 \xrightarrow{[\alpha]\,a} \surd_1\} \\
&\cup \{(s^{\checkmark}, s_2) \mid s_2 \xrightarrow{[\alpha]\,a} \surd_2\} \\
&\cup \{(s_1,s_2) \mid \\
&\qquad \bigvee_{\substack{\alpha',\beta' \in \mathcal{C}^-,\, a',b' \in \mathsf{A},\\ t \in U,V,W \subseteq \mathsf{A}}} \quad (s_1 \xrightarrow{[\alpha']\,a'} \surd_1 \;\wedge\; s_2 \xrightarrow{[\beta']\,b'} \surd_2 \;\wedge \\
&\qquad\qquad \mathit{\Phi}'_{U,V,W}(s_1,s_2) \;\wedge \\
&\qquad\qquad \textstyle\prod_{u \in U} \mathcal{E}_u \sqcap \prod_{u \in \mathsf{A} \setminus U} -\mathcal{E}_u = \alpha \;\wedge \\
&\qquad\qquad \Psi_V(\alpha') \sqcap \Psi_W(\beta') = \top \;\wedge\; a' \mid b' = a)\}\;.
\end{aligned}$$

and for every $U,V,W \subseteq \mathsf{A}$ with $t \in U,V,W$ and for every $(s_1,s_2) \in S$, $\mathit{\Phi}'_{U,V,W}(s_1,s_2)$ iff for all $u \in \mathsf{A} \setminus \{t\}$:

$$\begin{aligned}
&u \in V \;\Leftrightarrow \\
&\quad \exists \alpha'', a'', s'' \bullet (s_2 \xrightarrow{[\alpha'']\,a''}_2 s'' \;\wedge\; \Psi_W(\alpha'') = \top \;\wedge\; u \mid a'' \in U) \\
&\quad\; \vee\; \exists \alpha'', a'' \bullet (s_2 \xrightarrow{[\alpha'']\,a''} \surd_2 \;\wedge\; \Psi_W(\alpha'') = \top \;\wedge\; u \mid a'' \in U)
\end{aligned}$$

and

$$\begin{aligned}
&u \in W \;\Leftrightarrow \\
&\quad \exists \alpha'', a'', s'' \bullet (s_1 \xrightarrow{[\alpha'']\,a''}_1 s'' \;\wedge\; \Psi_V(\alpha'') = \top \;\wedge\; a'' \mid u \in U) \\
&\quad\; \vee\; \exists \alpha'', a'' \bullet (s_1 \xrightarrow{[\alpha'']\,a''} \surd_1 \;\wedge\; \Psi_V(\alpha'') = \top \;\wedge\; a'' \mid u \in U)\;.
\end{aligned}$$

– Let $T_i = (S_i, \rightarrow_i, \rightarrow\surd_i, s_i^0) \in \mathbb{CTS}_\kappa$ for $i = 1,2$. Suppose that $T_1 \mathbin{\widehat{\|}}' T_2 = (S, \rightarrow, \rightarrow\surd, s^0)$ where $S = ((S_1 \cup \{s^{\checkmark}\}) \times (S_2 \cup \{s^{\checkmark}\})) \setminus \{(s^{\checkmark}, s^{\checkmark})\}$ and $s^{\checkmark} = \mathrm{ch}_\kappa(\mathcal{S}_\kappa \setminus (S_1 \cup S_2))$. Then

  $$T_1 \mathbin{\widehat{[\![}}' T_2 = \Gamma(S', \rightarrow', \rightarrow\surd, s^{0\prime})\;,$$

  where

  $$s^{0\prime} = \mathrm{ch}_\kappa(\mathcal{S}_\kappa \setminus S)\;,$$

  $$S' = \{s^{0\prime}\} \cup S\;,$$

  and for every $(\alpha, a) \in \mathcal{C}^- \times \mathsf{A}$:

  $$\xrightarrow{(\alpha,a)}{}' = \{(s^{0\prime}, (s, s_2^0)) \mid s_1^0 \xrightarrow{[\alpha]\,a}_1 s \wr s^{\checkmark}\} \cup \xrightarrow{(\alpha,a)}\;.$$

– Let $T_i = (S_i, \rightarrow_i, \rightarrow\surd_i, s_i^0) \in \mathbb{CTS}_\kappa$ for $i = 1, 2$. Suppose that $T_1 \mathbin{\widehat{\|}'} T_2 = (S, \rightarrow, \rightarrow\surd, s^0)$ where $S = ((S_1 \cup \{s^\surd\}) \times (S_2 \cup \{s^\surd\})) \setminus \{(s^\surd, s^\surd)\}$ and $s^\surd = \mathrm{ch}_\kappa(\mathcal{S}_\kappa \setminus (S_1 \cup S_2))$. Then

$$T_1 \mathbin{\widehat{|}'} T_2 = \Gamma(S', \rightarrow', \rightarrow\surd', s^{0\prime}) ,$$

where

$$s^{0\prime} = \mathrm{ch}_\kappa(\mathcal{S}_\kappa \setminus S) ,$$

$$S' = \{s^{0\prime}\} \cup S ,$$

and for every $(\alpha, a) \in \mathcal{C}^- \times \mathsf{A}$:

$$\begin{aligned}
\xrightarrow{(\alpha,a)}{}' &= \{(s^{0\prime}, (s_1, s_2)) \mid (s_1, s_2) \in S \wedge \\
&\quad \bigvee_{\substack{\alpha',\beta' \in \mathcal{C}^-,\, a',b' \in \mathsf{A},\\ t \in U,V,W \subseteq \mathsf{A}}} (s_1^0 \xrightarrow{[\alpha']\, a'}_1 s_1 \wr s^\surd \wedge s_2^0 \xrightarrow{[\beta']\, b'}_2 s_2 \wr s^\surd \wedge \\
&\qquad \Phi'_{U,V,W}(s_1^0, s_2^0) \wedge \\
&\qquad \textstyle\bigsqcap_{u \in U} \mathcal{E}_u \sqcap \bigsqcap_{u \in \mathsf{A} \setminus U} -\mathcal{E}_u = \alpha \wedge \\
&\qquad \Psi_V(\alpha') \sqcap \Psi_W(\beta') = \top \wedge a' \mid b' = a)\} \\
&\quad \cup \xrightarrow{(\alpha,a)} ,
\end{aligned}$$

$$\begin{aligned}
\xrightarrow{(\alpha,a)}\surd' &= \{s^{0\prime} \mid \\
&\quad \bigvee_{\substack{\alpha',\beta' \in \mathcal{C}^-,\, a',b' \in \mathsf{A},\\ t \in U,V,W \subseteq \mathsf{A}}} (s_1^0 \xrightarrow{[\alpha']\, a'} \surd_1 \wedge s_2^0 \xrightarrow{[\beta']\, b'} \surd_2 \wedge \\
&\qquad \Phi'_{U,V,W}(s_1^0, s_2^0) \wedge \\
&\qquad \textstyle\bigsqcap_{u \in U} \mathcal{E}_u \sqcap \bigsqcap_{u \in \mathsf{A} \setminus U} -\mathcal{E}_u = \alpha \wedge \\
&\qquad \Psi_V(\alpha') \sqcap \Psi_W(\beta') = \top \wedge a' \mid b' = a)\} \\
&\quad \cup \xrightarrow{(\alpha,a)} \surd .
\end{aligned}$$

and $\Phi'_{U,V,W}(s_1, s_2)$ is as in the definition of $\widehat{\|}'$.

We can show that splitting bisimilarity is a congruence with respect to parallel composition, left merge and communication merge.

Lemma 7.1 (Congruence). *Let κ be an infinite cardinal. Then for all $T_1, T_2, T_1', T_2' \in \mathbb{CTS}_\kappa$, $T_1 \Leftrightarrow T_1'$ and $T_2 \Leftrightarrow T_2'$ imply $T_1 \mathbin{\widehat{\|}'} T_2 \Leftrightarrow T_1' \mathbin{\widehat{\|}'} T_2'$, $T_1 \mathbin{\widehat{\|\!\!\lfloor}'} T_2 \Leftrightarrow T_1' \mathbin{\widehat{\|\!\!\lfloor}'} T_2'$ and $T_1 \mathbin{\widehat{|}'} T_2 \Leftrightarrow T_1' \mathbin{\widehat{|}'} T_2'$.*

Proof. Although parallel composition and left merge as considered in the setting of $\mathrm{ACP}^{\mathrm{cc}}$ differs from parallel composition and left merge as considered in the setting of $\mathrm{PA}_\delta^{\mathrm{cc}}$, witnessing splitting bisimulations can be constructed in the same way as in the proof of Lemma 4.1. For communication merge, the witnessing splitting bisimulation is constructed like for left merge. It is straightforward to show that the constructed relations are splitting bisimulations indeed. However, it is not so easy as in the proof of Lemma 4.1. The most important complication is that we have to verify whether the constructed relation, say R, has the following property: $R((s_1, s_2), (s_1', s_2'))$ implies $\Phi'_{U,V,W}(s_1, s_2)$ iff $\Phi'_{U,V,W}(s_1', s_2')$ for all $U, V, W \subseteq \mathsf{A}$ with $t \in U, V, W$. □

The *full splitting bisimulation models* $\mathfrak{P}^{\mathrm{cc}\prime}_{\kappa}$ of ACP$^{\mathrm{cc}}$, one for each infinite cardinal κ, are the full splitting bisimulation models $\mathfrak{P}^{\mathrm{cc}}_{\kappa}$ of PA$^{\mathrm{cc}}_{\delta}$ with adapted operations $\widetilde{\|}'$ and $\widetilde{\|\!\!\_}'$ on $\mathbb{CTS}_{\kappa}/\underline{\Leftrightarrow}$ for the operators $\|$ and $\|\!\!\_$, and an additional operation $\widetilde{|}'$ on $\mathbb{CTS}_{\kappa}/\underline{\Leftrightarrow}$ for the operator $|$. Those operations are defined as follows:

$$[T_1]_{\underline{\Leftrightarrow}} \mathbin{\widetilde{\|}'} [T_2]_{\underline{\Leftrightarrow}} = [T_1 \mathbin{\widehat{\|}'} T_2]_{\underline{\Leftrightarrow}}\,,$$

$$[T_1]_{\underline{\Leftrightarrow}} \mathbin{\widetilde{\|\!\!\_}'} [T_2]_{\underline{\Leftrightarrow}} = [T_1 \mathbin{\widehat{\|\!\!\_}'} T_2]_{\underline{\Leftrightarrow}}\,,$$

$$[T_1]_{\underline{\Leftrightarrow}} \mathbin{\widetilde{|}'} [T_2]_{\underline{\Leftrightarrow}} = [T_1 \mathbin{\widehat{|}'} T_2]_{\underline{\Leftrightarrow}}\,.$$

Parallel composition, left merge and communication merge on $\mathbb{CTS}_{\kappa}/\underline{\Leftrightarrow}$ are well-defined because $\underline{\Leftrightarrow}$ is a congruence with respect to the corresponding operations on $\mathbb{CTS}_{\kappa}$.

The structures $\mathfrak{P}^{\mathrm{cc}\prime}_{\kappa}$ are models of ACP$^{\mathrm{cc}}$.

Theorem 7.1 (Soundness). *For each infinite cardinal κ, we have $\mathfrak{P}^{\mathrm{cc}\prime}_{\kappa} \models$ ACP$^{\mathrm{cc}}$.*

Proof. Because $\underline{\Leftrightarrow}$ is a congruence, and all changes with respect to $\mathfrak{P}^{\mathrm{cc}}_{\kappa}$ concern the operations associated with the operators $\|$, $\|\!\!\_$ and $|$, it is sufficient to show that all axioms from Table 4 are sound. The soundness of all those axioms follows easily from the definitions of the ingredients of $\mathfrak{P}^{\mathrm{cc}}_{\kappa}$. □

The axioms of ACP$^{\mathrm{cc}}$ constitute a complete axiomatization of the full splitting bisimulation models.

Theorem 7.2 (Completeness). *Let κ be an infinite cardinal. Then we have, for all closed terms p and q of sort* **P** *from the language of* ACP$^{\mathrm{cc}}$*, $\mathfrak{P}^{\mathrm{cc}}_{\kappa} \models p = q$ implies* ACP$^{\mathrm{cc}} \vdash p = q$.

Proof. By Lemma 6.1, for each closed term p of sort **P** from the language of ACP$^{\mathrm{cc}}$, there is a closed term q of sort **P** from the language of BPA$^{\mathrm{cc}}_{\delta}$ such that $p = q$ is derivable from the axioms of ACP$^{\mathrm{cc}}$; and by Lemma 6.2, all equations between closed terms of sort **P** from the language of BPA$^{\mathrm{cc}}_{\delta}$ that can be derived from the axioms of ACP$^{\mathrm{cc}}$ can be derived from the axioms of BPA$^{\mathrm{cc}}_{\delta}$. Therefore, it is sufficient to prove that the axioms of BPA$^{\mathrm{cc}}_{\delta}$ constitute a complete axiomatization of the restrictions of the full splitting bisimulation models of ACP$^{\mathrm{cc}}$ to the constants and operators of BPA$^{\mathrm{cc}}_{\delta}$. In the proof of Theorem 4.2, it is shown that the axioms of BPA$^{\mathrm{cc}}_{\delta}$ constitute a complete axiomatization of the restrictions of the full splitting bisimulation models of PA$^{\mathrm{cc}}_{\delta}$ to the constants and operators of BPA$^{\mathrm{cc}}_{\delta}$. Because the restrictions of the full splitting bisimulation models of PA$^{\mathrm{cc}}_{\delta}$ to the constants and operators of BPA$^{\mathrm{cc}}_{\delta}$ coincide with the restrictions of the full splitting bisimulation models of ACP$^{\mathrm{cc}}$ to the constants and operators of BPA$^{\mathrm{cc}}_{\delta}$, the proof is completed. □

It is easy to see that Theorem 4.3 goes through for $\mathfrak{P}^{\mathrm{cc}\prime}_{\kappa}$.

In this section, the full splitting bisimulation models $\mathfrak{P}^{\mathrm{cc}}_{\kappa}$ of PA$^{\mathrm{cc}}_{\delta}$ have been expanded to obtain the full splitting bisimulation models $\mathfrak{P}^{\mathrm{cc}\prime}_{\kappa}$ of ACP$^{\mathrm{cc}}$. Henceforth, we will loosely write $\mathfrak{P}^{\mathrm{cc}}_{\kappa}$ for $\mathfrak{P}^{\mathrm{cc}\prime}_{\kappa}$.

Table 5. Axioms for recursion

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

8 Guarded Recursion

In order to allow for the description of (potentially) non-terminating processes, we add guarded recursion to $\mathrm{ACP^{cc}}$.

A *recursive specification* over $\mathrm{ACP^{cc}}$ is a set of *recursive* equations $E = \{X = t_X \mid X \in V\}$ where V is a set of variables and each t_X is a term of sort $\mathbf{P}$ from the language of $\mathrm{ACP^{cc}}$ that only contains variables from V. We write $\mathrm{V}(E)$ for the set of all variables that occur on the left-hand side of an equation in E. A *solution* of a recursive specification E is a set of processes (in some model of $\mathrm{ACP^{cc}}$) $\{P_X \mid X \in \mathrm{V}(E)\}$ such that the equations of E hold if, for all $X \in \mathrm{V}(E)$, X stands for P_X.

Let t be a term of sort $\mathbf{P}$ from the language of $\mathrm{ACP^{cc}}$ containing a variable X. We call an occurrence of X in t *guarded* if t has a subterm of the form $a \cdot t'$ containing this occurrence of X. A recursive specification over $\mathrm{ACP^{cc}}$ is called a *guarded* recursive specification if all occurrences of variables in the right-hand sides of its equations are guarded or it can be rewritten to such a recursive specification using the axioms of $\mathrm{ACP^{cc}}$ and the equations of the recursive specification. We are only interested in models of $\mathrm{ACP^{cc}}$ in which guarded recursive specifications have unique solutions.

For each guarded recursive specification E and each variable $X \in \mathrm{V}(E)$, we introduce a constant of sort $\mathbf{P}$ standing for the unique solution of E for X. This constant is denoted by $\langle X|E\rangle$. We often write X for $\langle X|E\rangle$ if E is clear from the context. In such cases, it should also be clear from the context that we use X as a constant.

We will also use the following notation. Let t be a term of sort $\mathbf{P}$ from the language of $\mathrm{ACP^{cc}}$ and E be a guarded recursive specification over $\mathrm{ACP^{cc}}$. Then we write $\langle t|E\rangle$ for t with, for all $X \in \mathrm{V}(E)$, all occurrences of X in t replaced by $\langle X|E\rangle$.

The additional axioms for recursion are given in Table 5. Both RDP and RSP are axiom schemas. Side conditions are added to restrict the variables, terms and guarded recursive specifications for which X, t_X and E stand. The additional axioms for recursion are known as the recursive definition principle (RDP) and the recursive specification principle (RSP). The equations $\langle X|E\rangle = \langle t_X|E\rangle$ for a fixed E express that the constants $\langle X|E\rangle$ make up a solution of E. The conditional equations $E \Rightarrow X = \langle X|E\rangle$ express that this solution is the only one.

In the full splitting bisimulation models of $\mathrm{ACP^{cc}}$, guarded recursive specifications over $\mathrm{ACP^{cc}}$ have unique solutions.

Theorem 8.1 (Unique solutions). *For each infinite cardinal κ, guarded recursive specifications over* $\mathrm{ACP^{cc}}$ *have unique solutions in* $\mathfrak{P}^{\mathrm{cc}}_{\kappa}$.

Table 6. Axioms for preferential choice ($a \in \mathsf{A}$)

$\delta \nrightarrow x = x$	PC1
$a \cdot x \nrightarrow y = a \cdot x + -\mathcal{E}_a :\rightarrow y$	PC2
$(x + y) \nrightarrow z = x \nrightarrow (y \nrightarrow z) + y \nrightarrow (x \nrightarrow z)$	PC3
$(\phi :\rightarrow x) \nrightarrow y = \phi :\rightarrow (x \nrightarrow y) + -\phi :\rightarrow y$	PC4

Proof. The proof is analogous to the proof of the corresponding property for the full splitting bisimulation models of ACP$^{\mathrm{c}}$ given in [11]. □

Thus, the full splitting bisimulation models $\mathfrak{P}^{\mathrm{cc}\prime\prime}_{\kappa}$ of ACP$^{\mathrm{cc}}$ with guarded recursion are simply the expansions of the full splitting bisimulation models $\mathfrak{P}^{\mathrm{cc}}_{\kappa}$ of ACP$^{\mathrm{cc}}$ obtained by associating with each constant $\langle X|E\rangle$ the unique solution of E for X in the full splitting bisimulation model concerned.

9 Preferential Choice

In the presence of conditional expressions of which the conditions concern the enabledness of actions in the context in which a process is placed, it is easy to give defining equations for a preferential choice operator. In this section, we extend ACP$^{\mathrm{cc}}$ with the binary *preferential choice* operator $\nrightarrow : \mathbf{P} \times \mathbf{P} \rightarrow \mathbf{P}$.

Let p and q be closed terms of sort $\mathbf{P}$. Intuitively, the preferential choice operator can be explained as follows:

- $p \nrightarrow q$ behaves as p if the context in which it is placed permits it to behave as p, and as q otherwise.

The additional axioms for preferential choice are the axioms given in Table 6. From the axioms of ACP$^{\mathrm{cc}}$ and axioms PC1–PC4, we can easily derive the equation $(\phi :\rightarrow a \cdot x + \psi :\rightarrow b \cdot y) \nrightarrow z = \phi :\rightarrow a \cdot x + \psi :\rightarrow b \cdot y + -((\phi \sqcap \mathcal{E}_a) \sqcup (\psi \sqcap \mathcal{E}_b)) :\rightarrow z$. The following generalization of this result gives a full picture of the preferential choice operator.

Proposition 9.1 (Characterization). *From the axioms of* ACP$^{\mathrm{cc}}$ *and axioms PC1–PC4, the following is derivable for all* $n, m \geq 0$*, for all* $a_0, \ldots, a_{n-1}$*,* $b_0, \ldots, b_{m-1} \in \mathsf{A}$*:*

$$\begin{aligned}&\Big(\sum_{i<n} \phi_i :\rightarrow a_i \cdot x_i + \sum_{i<m} \psi_i :\rightarrow b_i\Big) \nrightarrow y \\ &\quad = \sum_{i<n} \phi_i :\rightarrow a_i \cdot x_i + \sum_{i<m} \psi_i :\rightarrow b_i \\ &\qquad + -\big(\textstyle\bigsqcup_{i<n} (\phi_i \sqcap \mathcal{E}_{a_i}) \sqcup \bigsqcup_{i<m} (\psi_i \sqcap \mathcal{E}_{b_i})\big) :\rightarrow y \;.\end{aligned}$$

Proof. Easy, by induction on $n + m$. □

A corollary of Proposition 9.1 is that the preferential choice operator is associative.

Corollary 9.1 (Associativity). *For all closed terms p, q, r of sort* **P** *from the language of* ACP$^{\mathrm{cc}}$ *extended with preferential choice,* $p \nrightarrow (q \nrightarrow r) = (p \nrightarrow q) \nrightarrow r$ *is derivable from the axioms of* ACP$^{\mathrm{cc}}$ *and axioms PC1–PC4.*

Another corollary of Proposition 9.1 is that all occurrences of the preferential choice operator can be eliminated from closed terms.

Corollary 9.2 (Elimination). *For all closed terms p of sort* **P** *from the language of* ACP$^{\mathrm{cc}}$ *extended with preferential choice, there exists a closed term q of sort* **P** *from the language of* ACP$^{\mathrm{cc}}$ *such that $p = q$ is derivable from the axioms of* ACP$^{\mathrm{cc}}$ *and axioms PC1–PC4.*

The full bisimulation models of ACP$^{\mathrm{cc}}$ with preferential choice are the expansions of the full bisimulation models $\mathfrak{P}^{\mathrm{cc}}$ of ACP$^{\mathrm{cc}}$ obtained by first associating with the operator $\nrightarrow$ a corresponding operation on $\mathbb{CTS}_\kappa$ and then lifting the result of this to $\mathbb{CTS}_\kappa/{\underline{\leftrightarrow}}$. This calls for extraction of the initial guarded actions of a conditional transition system.

Let $T = (S, \rightarrow, \rightarrow\surd, s^0) \in \mathbb{CTS}_\kappa$. Then the *initial guarded actions* of T, written $\mathrm{I}(T)$, is the set $\{(\alpha, a) \in \mathcal{C}^- \times \mathsf{A} \mid \exists s \in S \bullet s^0 \xrightarrow{[\alpha]\,a} s \;\vee\; s^0 \xrightarrow{[\alpha]\,a} \surd\}$.

We proceed with associating with the operator $\nrightarrow$ an operation $\widehat{\nrightarrow}$ on $\mathbb{CTS}_\kappa$ as follows.

- Let $T_i = (S_i, \rightarrow_i, \rightarrow\surd_i, s_i^0) \in \mathbb{CTS}_\kappa$ for $i = 1, 2$. Then

  $$T_1 \mathbin{\widehat{\nrightarrow}} T_2 = \Gamma(S, \rightarrow, \rightarrow\surd, s^0) \;,$$

  where

  $$\begin{aligned} s^0 &= \mathrm{ch}_\kappa(\mathcal{S}_\kappa \setminus (S_1 \uplus S_2)) \;, \\ S &= \{s^0\} \cup (S_1 \uplus S_2) \;, \end{aligned}$$

  and for every $(\alpha, a) \in \mathcal{C}^- \times \mathsf{A}$:

  $$\begin{aligned} \xrightarrow{(\alpha,a)} \;&= \{(s^0, \mu_1(s)) \mid s_1^0 \xrightarrow{[\alpha]\,a}_1 s\} \\ &\cup \{(s^0, \mu_2(s)) \mid \exists \beta \bullet s_2^0 \xrightarrow{[\beta]\,a}_2 s \;\wedge \\ &\qquad\qquad \alpha = -({\textstyle\bigsqcup_{(\alpha',a') \in \mathrm{I}(T_1)}}(\alpha' \sqcap \mathcal{E}_{a'})) \sqcap \beta\} \\ &\cup \{(\mu_1(s), \mu_1(s')) \mid s \xrightarrow{[\alpha]\,a}_1 s'\} \\ &\cup \{(\mu_2(s), \mu_2(s')) \mid s \xrightarrow{[\alpha]\,a}_2 s'\} \;, \end{aligned}$$

  $$\begin{aligned} \xrightarrow{(\alpha,a)}\surd \;&= \{s^0 \mid s_1^0 \xrightarrow{[\alpha]\,a} \surd_1\} \\ &\cup \{s^0 \mid \exists \beta \bullet s_2^0 \xrightarrow{[\alpha]\,a} \surd_2 \;\wedge \\ &\qquad\qquad \alpha = -({\textstyle\bigsqcup_{(\alpha',a') \in \mathrm{I}(T_1)}}(\alpha' \sqcap \mathcal{E}_{a'})) \sqcap \beta\} \\ &\cup \{\mu_1(s) \mid s \xrightarrow{[\alpha]\,a} \surd_1\} \\ &\cup \{\mu_2(s) \mid s \xrightarrow{[\alpha]\,a} \surd_2\} \;. \end{aligned}$$

We can show that splitting bisimilarity is a congruence with respect to preferential choice.

Lemma 9.1 (Congruence). *Let κ be an infinite cardinal. Then for all $T_1, T_2, T_1', T_2' \in \mathbb{CTS}_\kappa$, $T_1 \Leftrightarrow T_1'$ and $T_2 \Leftrightarrow T_2'$ imply $T_1 \mathbin{\widehat{\not\mapsto}} T_2 \Leftrightarrow T_1' \mathbin{\widehat{\not\mapsto}} T_2'$.*

Proof. Although preferential choice differs from alternative composition (being a non-preferential choice), a witnessing splitting bisimulation can be constructed in the same way as in the proof of Lemma 4.1. It is straightforward to show that the constructed relation is a splitting bisimulation indeed. As compared with alternative composition, all we have to do more is to show that $T_1 \Leftrightarrow T_1'$ implies $\bigsqcup_{(\alpha',a')\in \mathrm{I}(T_1)}(\alpha' \sqcap \mathcal{E}_{a'}) = \bigsqcup_{(\alpha',a')\in \mathrm{I}(T_1')}(\alpha' \sqcap \mathcal{E}_{a'})$. □

The operation $\widetilde{\not\mapsto}$ on $\mathbb{CTS}_\kappa/\Leftrightarrow$ is defined as follows:

$$[\,T_1\,]_{\Leftrightarrow} \mathbin{\widetilde{\not\mapsto}} [\,T_2\,]_{\Leftrightarrow} = [\,T_1 \mathbin{\widehat{\not\mapsto}} T_2\,]_{\Leftrightarrow}\ .$$

10 Example

In this section, we give an example of the use of the preferential choice operator. The example is adapted from [4]. The example concerns a printer with a control panel. The printer will print an infinite sequence $\langle c_0, c_1, c_2, \ldots\rangle$ of characters from a finite set of characters C, but can be interrupted by means of the control panel. The control panel has two buttons, a start button to indicate that the printing must be started and a stop button to indicate that the printing must be stopped. Whenever a button is pushed, a corresponding message is sent to the printer. The printer prints characters from the infinite sequence, but can also receive messages from the control panel. The recursive specification of the control panel is as follows:

$$C = b(\mathit{start}) \cdot s(\mathit{start}) \cdot C + b(\mathit{stop}) \cdot s(\mathit{stop}) \cdot C\ ;$$

and the recursive specification of the printer is as follows ($i \geq 0$):

$$\begin{aligned} P\ &= W_0 \\ W_i &= r(\mathit{start}) \cdot P_i + r(\mathit{stop}) \cdot W_i \\ P_i\ &= (r(\mathit{start}) \cdot P_i + r(\mathit{stop}) \cdot W_i) \not\mapsto p(c_i) \cdot P_{i+1}\ . \end{aligned}$$

In this example, we take $\mid : \mathsf{A}_\delta \times \mathsf{A}_\delta \to \mathsf{A}_\delta$ such that $s(\mathit{start}) \mid r(\mathit{start}) = c(\mathit{start})$ and $s(\mathit{stop}) | r(\mathit{stop}) = c(\mathit{stop})$, and in all other cases it yields δ. The whole system is described by the $\partial_H(C \parallel P)$, where $H = \{s(\mathit{start}), r(\mathit{start}), s(\mathit{stop}), r(\mathit{stop})\}$.

In the recursive specification of the printer, the preferential choice operator is used in the equation for P_i to describe that receiving a message from the control panel must be given preference to printing a character. It follows from the axioms for preferential choice that the preferential choice operator can be eliminated from this equation. This yields the following equation:

$$P_i = r(\mathit{start}) \cdot P_i + r(\mathit{stop}) \cdot W_i + -(\mathcal{E}_{r(\mathit{start})} \sqcup \mathcal{E}_{r(\mathit{stop})}) :\rightarrow p(c_i) \cdot P_{i+1}\ .$$

From the axioms of ACP^{cc}, the additional axioms for preferential choice and RSP, we can derive that $\partial_H(C \parallel P)$ is the solution of the following recursive specification ($i \geq 0$):

$$
\begin{aligned}
S &= S_0^w \\
S_i^w &= b(start) \cdot c(start) \cdot S_i^p + b(stop) \cdot c(stop) \cdot S_i^w \\
S_i^p &= b(start) \cdot c(start) \cdot S_i^p + b(stop) \cdot c(stop) \cdot S_i^w + p(c_i) \cdot S_{i+1}^p \ .
\end{aligned}
$$

This shows that the reaction to button pushing is immediate: printing of characters never takes place between button pushing and the reaction to button pushing.

11 Concluding Remarks

We have presented ACP^{cc}, an extension of ACP with conditional expressions of which the conditions concern the enabledness of actions in the context in which a process is placed, as well as the main models of ACP^{cc}. To the best of our knowledge, there is no other work in the field of process algebra studying conditions of this kind. However, there are several extensions of ACP that include conditional expressions of some kind. As a matter of fact, ACP^{cc} is a variant of a recent extension of ACP with conditional expressions called ACP^{c} [11, 12]. Earlier extensions of ACP that include conditional expressions can be found in [6, 2, 14, 3].

A striking point of ACP^{cc} is that its axioms do not include axioms similar to the axioms of ACP for communication merge (axioms CM5–CM9, see e.g. [7]). Such axioms are too much to expect: the mutual enabling of actions involved in the communication merge of two processes is a matter which can only be resolved by looking at the processes as a whole. The same need of a global approach makes structural operational semantics unsuited for associating transition systems with closed terms.

We do not have a clear notion of the applications of ACP^{cc}. We have treated an application of ACP^{cc} that remains within the domain of process algebra: the extension of ACP^{cc} with a preferential choice operator. This operator appears to generalize the priority choice operator added to CCS in [16].

ACP^{cc} includes pre-abstraction, but not abstraction. Abstraction is usually based on observation equivalence [21] or branching bisimulation equivalence [23], which both abstract from both the structure of finitary internal activity and its presence. Abstraction from its presence is not considered to be well-suited to the setting of ACP^{cc}. Orthogonal bisimulation equivalence [13], an equivalence introduced recently, abstracts from the structure of finitary internal activity, but not from its presence. Therefore, this equivalence looks to be better suited to the setting of ACP^{cc}.

One option for future work is development of an extension of ACP^{cc} with abstraction based on orthogonal bisimulation equivalence. Another option for future work is investigation into ways to combine the kind of conditions considered in ACP^{cc} with other kinds of conditions, in particular with the retrospective conditions of ACP^{cr} [11].

References

1. J. C. M. Baeten and J. A. Bergstra. Global renaming operators in concrete process algebra. *Information and Control*, 78(3):205–245, 1988.
2. J. C. M. Baeten and J. A. Bergstra. Process algebra with signals and conditions. In M. Broy, editor, *Programming and Mathematical Methods*, volume F88 of *NATO ASI Series*, pages 273–323. Springer-Verlag, 1992.
3. J. C. M. Baeten and J. A. Bergstra. Process algebra with propositional signals. *Theoretical Computer Science*, 177:381–405, 1997.
4. J. C. M. Baeten, J. A. Bergstra, and J. W. Klop. Syntax and defining equations for an interrupt mechanism in process algebra. *Fundamenta Informaticae*, 9:127–168, 1986.
5. J. C. M. Baeten, J. A. Bergstra, and J. W. Klop. Conditional axioms and α/β-calculus in process algebra. In M. Wirsing, editor, *Formal Description of Programming Concepts III*, pages 53–75. North-Holland, 1987.
6. J. C. M. Baeten, J. A. Bergstra, S. Mauw, and G. J. Veltink. A process specification formalism based on static COLD. In J. A. Bergstra and L. M. G. Feijs, editors, *Algebraic Methods II: Theory, Tools and Applications*, volume 490 of *Lecture Notes in Computer Science*, pages 303–335. Springer-Verlag, 1991.
7. J. C. M. Baeten and W. P. Weijland. *Process Algebra*, volume 18 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, Cambridge, 1990.
8. J. A. Bergstra and J. W. Klop. Process algebra for synchronous communication. *Information and Control*, 60(1/3):109–137, 1984.
9. J. A. Bergstra and J. W. Klop. Algebra of communicating processes with abstraction. *Theoretical Computer Science*, 37:77–121, 1985.
10. J. A. Bergstra and C. A. Middelburg. Model theory for process algebra. Computer Science Report 04-24, Department of Mathematics and Computer Science, Eindhoven University of Technology, September 2004.
11. J. A. Bergstra and C. A. Middelburg. Splitting bisimulations and retrospective conditions. Computer Science Report 05-03, Department of Mathematics and Computer Science, Eindhoven University of Technology, January 2005.
12. J. A. Bergstra and C. A. Middelburg. Strong splitting bisimulation equivalence. To appear in J.L. Fiadeiro, N. Harman, M. Roggenbach and J. Rutten, editors, *CALCO 2005*, LNCS, Springer-Verlag, 2005. Preliminary version: Computing Science Report 05-04, Department of Mathematics and Computing Science, Eindhoven University of Technology.
13. J. A. Bergstra, A. Ponse, and M.B van der Zwaag. Branching time and orthogonal bisimulation equivalence. *Theoretical Computer Science*, 309:313–355, 2003.
14. J. A. Bergstra, A. Ponse, and J. J. van Wamel. Process algebra with backtracking. In J. W. de Bakker, W. P. de Roever, and G. Rozenberg, editors, *A Decade of Concurrency (Reflections and Perspectives)*, volume 803 of *Lecture Notes in Computer Science*, pages 46–91. Springer-Verlag, 1994.
15. N. Busi, R. J. van Glabbeek, and R. Gorrieri. Axiomatising ST-bisimulation semantics. In R.-R. Olderog, editor, *PROCOMET'94*, volume 56 of *IFIP Transactions A*, pages 169–188. North-Holland, 1994.

16. J. Camilleri and G. Winskel. CCS with priority choice. *Information and Computation*, 116(1):26–37, 1995.
17. R. Cleaveland, G. Lüttgen, and V. Natarajan. Priority in process algebra. In J. A. Bergstra, A. Ponse, and S. A. Smolka, editors, *Handbook of Process Algebra*, pages 711–765. Elsevier, Amsterdam, 2001.
18. P. R. Halmos. *Lectures on Boolean Algebras*. Mathematical Studies. Van Nostrand, Princeton, NJ, 1963.
19. C. A. R. Hoare, I. J. Hayes, He Jifeng, C. C. Morgan, A. W. Roscoe, J. W. Sanders, I. H. Sorensen, J. M. Spivey, and B. A. Sufrin. Laws of programming. *Communications of the ACM*, 30:672–686, 1987.
20. J. W. Klop. Term rewriting systems. In S. Abramsky, D. M. Gabbay, and T. S. E. Maibaum, editors, *Handbook of Logic in Computer Science*, volume II, pages 1–116. Oxford University Press, Oxford, 1992.
21. R. Milner. *Communication and Concurrency*. Prentice-Hall, Englewood Cliffs, 1989.
22. J. D. Monk and R. Bonnet, editors. *Handbook of Boolean Algebras*, volume 1. Elsevier, Amsterdam, 1989.
23. R. J. van Glabbeek and W. P. Weijland. Branching time and abstraction in bisimulation semantics. *Journal of the ACM*, 43(3):555–600, 1996.